Universidad de Buenos Aires
Facultad de Ciencias Exactas y Naturales
Departamento de Computación

Un estudio semántico sobre extensiones avanzadas del λ-cálculo: patrones y operadores de control

Tesis presentada para optar por el título de Doctor de la Universidad de Buenos Aires en el área Ciencias de la Computación

Lic. Andrés Ezequiel Viso

Directores de Tesis: Dr. Eduardo Bonelli y Dra. Delia Kesner
Consejero de Estudios: Dr. Alejandro Ríos

Lugar de trabajo: Departamento de Computación, FCEyN, UBA

Buenos Aires, 22 de Julio de 2020

Resumen

El siguiente trabajo presenta distintos análisis semánticos sobre fundamentos de lenguajes de programación funcional. En particular, se establece como objeto de estudio el λ-*cálculo* para luego focalizar especialmente en dos de sus extensiones, a saber: el *Pure Pattern Calculus* (PPC) y el $\lambda\mu$-*cálculo*.

En un primer lugar se estudian estrategias de evaluación para λ-cálculo por medio de un sistema de tipos intersección (no idempotente). El mismo es capaz de caracterizar (exactamente) los términos normalizantes. Esta herramienta permite establecer una igualdad observacional entre la conocida estrategia *call-by-need* –noción puramente sintáctica– y el concepto semántico de *needed redexes*. Más precisamente, se introduce la reducción *weak-head needed* y se muestra que el sistema de tipos identifica sintácticamente todos los redexes weak-head needed de un término.

PPC, introducido por Kesner y Jay, es una extensión del λ-cálculo con *patrones dinámicos* y *pattern matching* que permite abstraer virtualmente cualquier término. Esto dio lugar a dos nuevas formas de polimorfismo: *path polymorphism* y *pattern polymorphism*. El primero se refiere a la definición de funciones que se aplican uniformemente a estructuras de datos arbitrarias, definidas de manera inductiva. Su esencia recae sobre patrones de la forma $x\,y$ (*i.e.* la aplicación de dos variables), lo que permite descomponer una estructura de datos en sus diferentes partes. Este trabajo se enfoca en path polymorphism restringiendo el análisis al *Calculus of Applicative Patterns* (CAP), un nuevo formalismo que puede ser visto como el fragmento estático de PPC, lo que descarta la forma de *pattern polymorphism*. Se desarrolla un *sistema de tipos estático* para CAP, capaz de capturar path polymorphism mediante una combinación adecuada de herramientas de tipado como: tipos constantes, aplicativos, unión y recursivos. Las propiedades fundamentales de *subject reduction* y *progress* se satisfacen en el sistema propuesto, lo que garantiza el buen comportamiento dinámico del cálculo. Esto recae crucialmente en una novedosa noción de *compatibilidad de patrones*. A su vez, se desarrolla un algoritmo de *type-checking* eficiente mediante la formulación de una variante dirigida por sintaxis del sistema de tipos. Esto involucra algoritmos de chequeo de equivalencia de tipos y sub-tipado, ambos basados en caracterizaciones co-inductivas de estas relaciones.

En una tercera línea de investigación, se consideran programas con operadores de control. El objetivo es identificar aquellos programas con tales constructores cuyas semánticas de reducción se corresponden de manera exacta. Esto se logra introduciendo una relación $\simeq$, definida sobre una presentación alternativa del $\lambda\mu$-cálculo con operadores explícitos llamada ΛM. El resultado se basa en dos ingredientes fundamentales: (1) la factorización de la reducción del $\lambda\mu$-cálculo en pasos multiplicativos y exponenciales mediante el uso de los nuevos operadores explícitos introducidos; y (2) la traducción de los términos del ΛM-cálculo en las *proof-nets polarizadas* (PPNs) de Laurent, de modo que cut-elimination en las PPN simula la reducción en ΛM. La relación $\simeq$ propuesta resulta ser una *bisimulación fuerte* con respecto a la reducción en ΛM, *i.e.* dos términos equivalentes tienen exactamente la misma semántica de reducción, un resultado que falla tanto para la σ-equivalencia de Regnier en el λ-cálculo como para la σ-equivalencia de Laurent en $\lambda\mu$.

Abstract

This work presents different semantical analyses on the foundations of functional programming languages. In particular, λ-*calculus* is set as the object of study and, afterwards, special focus is placed on two of its extensions, namely the *Pure Pattern Calculus* (PPC) and $\lambda\mu$-*calculus*.

First, evaluation strategies for the λ-calculus are studied by means of an (non-idempotent) intersection types system capable to characterise (exactly) the normalising terms. This tool allows to establish an observational equivalence between the so-called *call-by-need* strategy –a purely syntactical notion– with the semantical concept of *needed redexes*. More precisely, the *weak-head needed* reduction is introduced and the type system is shown to syntactically identify all the weak-head needed redexes of a term.

PPC, due to Kesner and Jay, is an extension of the λ-calculus with *dynamic patterns* and *pattern matching* that allows to abstract virtually any term. This gave place to two novel notions of polymorphism: *path polymorphism* and *pattern polymorphism*. The former enables the definition of functions uniformly applicable to arbitrary recursively specified data structures. Its essence relies on patterns of the form $x\,y$ (*i.e.* the application of two variables), which allows to decompose a data structure into its parts. This work focuses on path polymorphism by restricting the analysis to the novel *Calculus of Applicative Patterns* (CAP), which can be seen as the static fragment of PPC, thus ruling out *pattern polymorphism*. A *static type system* for CAP is developed, able to capture the desired form of polymorphism by means of a proper combination of typing tools like constants as types, applicative types, union types and recursive types. The proposed system enjoys the fundamental properties of *subject reduction* and *progress* that guarantee well-behaved dynamics, and relies crucially on a novel notion of *pattern compatibility*. An efficient *type-checking* algorithm is developed as well, by formulating a syntax-directed variant of the type system. This involves algorithms for checking type equivalence and subtyping, both based on coinductive characterisations of those relations.

In a third line of research, programs with control operators are considered. The purpose is to identify programs with such constructors whose reduction semantics are in exact correspondence. This is achieved by introducing a relation $\simeq$, defined over a revised presentation of $\lambda\mu$-calculus with explicit term operators dubbed ΛM. The result builds on two fundamental ingredients: (1) factorisation of $\lambda\mu$-reduction into multiplicative and exponential steps by means of the newly introduced explicit operators of ΛM; and (2) translation of ΛM-terms into Laurent's *Polarized Proof-Nets* (PPN) such that cut-elimination in PPN simulates reduction in ΛM. The proposed relation $\simeq$ happens to be a *strong bisimulation* with respect to reduction in ΛM, *i.e.* two equivalent terms have exactly the same reduction semantics, a result which fails for Regnier's σ-equivalence in λ-calculus as well as for Laurent's σ-equivalence in $\lambda\mu$.

Agradecimientos

Quiero agradecer especialmente a mi familia por el apoyo en este camino, que empezó como una aventura más y terminó resultando bastante más largo de lo esperado. A mis amigos de la infancia, de la facultad, de la música; esos que siempre están cuando uno necesita distraerse y hablar de otra cosa. Voy a evitar dar nombres para no olvidarme de nadie.

También quiero hacer una mención especial al grupo de investigación, comandado en su momento por Delia, Eduardo y Alejandro. Debo reconocer que ingresé a la carrera sin tener muy claro dónde me estaba metiendo, y me encontré con mucho más de lo que un estudiante puede esperar. Un ambiente de trabajo inmejorable, con todos los recursos puestos a disposición del estudiante. Me dieron la posibilidad no solo de formarme como profesional sino también de conocer el mundo en ese proceso. Pero por sobre todas las cosas, me encontré con un grupo humano excelente, con el cual forjamos una relación que trasciende lo profesional.

Esta tesis esta dedicada a mis viejos (Jorge y Marisa), a Jobo y a mi tío Rubén.

Índice general

1. Introducción **1**
- 1.1. Contribuciones . . . 3
  - 1.1.1. Reducciones necesarias en λ-cálculo . . . 3
  - 1.1.2. Calculus of Applicative Patterns . . . 5
  - 1.1.3. Bisimulación fuerte para operadores de control . . . 7
- 1.2. Aspectos técnicos . . . 8
  - 1.2.1. El λ-cálculo . . . 9
  - 1.2.2. El sistema de tipos simples para λ-cálculo . . . 10
  - 1.2.3. Inducción y co-inducción . . . 11
  - 1.2.4. Chequeo de pertenencia . . . 13

2. Reducciones necesarias en λ-cálculo **17**
- 2.1. Preliminares . . . 18
  - 2.1.1. El λ-cálculo call-by-name . . . 18
  - 2.1.2. Reducciones head, weak-head y leftmost . . . 19
- 2.2. Reducciones needed . . . 20
  - 2.2.1. Residuos . . . 20
  - 2.2.2. Nociones de forma normal . . . 20
  - 2.2.3. Nociones de reducción needed . . . 21
- 2.3. El sistema de tipos $\mathcal{N}$ . . . 22
  - 2.3.1. Propiedades del sistema . . . 25
- 2.4. Sustitución y reducción en derivaciones . . . 26
- 2.5. Redexes tipados vs. weak-head needed . . . 28
  - 2.5.1. Redexes weak-head needed son tipados . . . 28
  - 2.5.2. Redexes tipados principalmente son weak-head needed . . . 29
- 2.6. Normalización weak-head needed . . . 30
- 2.7. El λ-cálculo call-by-need . . . 30
- 2.8. Equivalencia observacional . . . 31
- 2.9. Conclusión . . . 32

3. Calculus of Applicative Patterns **33**
- 3.1. Preliminares . . . 35
  - 3.1.1. Pure Pattern Calculus . . . 36
  - 3.1.2. Restricciones de PPC . . . 39
- 3.2. Sintaxis de CAP . . . 40
  - 3.2.1. Relación entre CAP y PPC . . . 42
- 3.3. Sistema de tipos . . . 42
  - 3.3.1. Expresiones de tipo . . . 43

3.3.2. Invertibilidad del sub-tipado . . . 45
3.3.3. Reglas de tipado . . . 51
3.3.4. Compatibilidad . . . 52
3.3.5. Propiedades del sistema . . . 58
3.4. Adecuación del sistema . . . 60
3.5. Normalización fuerte . . . 62
3.6. Chequeo de tipos . . . 63
3.6.1. Chequeo de equivalencia y sub-tipado . . . 63
3.6.2. Algoritmo de chequeo de tipos . . . 68
3.6.3. Hacia un chequeo de tipos eficiente . . . 74
3.6.4. Prototipos de implementación . . . 84
3.7. Conclusión . . . 85

4. Bisimulación fuerte para operadores de control 87
4.1. Preliminares . . . 92
4.1.1. El $\lambda\mu$-cálculo . . . 92
4.1.2. Sistema de tipos simples para $\lambda\mu$-cálculo . . . 93
4.1.3. Proof-nets polarizadas . . . 95
4.1.4. Sigma equivalencia para $\lambda\mu$-cálculo . . . 102
4.2. El ΛM-cálculo . . . 103
4.2.1. Semántica operacional . . . 104
4.3. Sistema de tipos simples para ΛM-cálculo . . . 106
4.4. Traducción a proof-nets . . . 108
4.5. Reducción meaningful para ΛM-cálculo . . . 112
4.6. Equivalencia estructural para ΛM-cálculo . . . 115
4.7. Dos resultados de correspondencia . . . 118
4.8. El resultado de bisimulación fuerte . . . 121
4.9. Conclusión . . . 121

5. Conclusión 123

A. Pruebas: Reducciones necesarias en λ-cálculo 127

B. Pruebas: Calculus of Applicative Patterns 135

C. Pruebas: Bisimulación fuerte para operadores de control 177

Capítulo 1

Introducción

El λ-cálculo, introducido por Church [Chu32] es considerado un lenguaje de programación universal en el que las funciones son ciudadanos de primera clase: se puede aplicar una función a otra función y se puede devolver una función como resultado de la aplicación de otra a su argumento. A pesar de la simplicidad de su sintaxis, el λ-cálculo es suficientemente rico para representar todas las funciones computables. Se trata de un caso particular de *sistema de reescritura* [BN98, BKvO03], que sienta las bases de la Programación Funcional [Pey87], estableciendo los fundamentos teóricos subyacentes a lenguajes como LISP, Miranda, Haskell, o los pertenecientes a la familia ML (Caml, SML, OCaml, etc.). Además, los diferentes sistemas de tipos introducidos para el λ-cálculo a lo largo de los años [Bar92, BDS13] y los distintos formalismos lógicos están estrechamente relacionados a través de la correspondencia de Curry-Howard [SU06]. Ésta establece que un programa es el testigo de la demostración de la fórmula lógica que su tipo representa, dando así lugar al desarrollo de asistentes de prueba (Coq, Nuprl, Isabelle).

Estrategias de reducción. Algunos sistemas de reescritura como el λ-cálculo, si bien no son completos, cuentan con la propiedad de Church–Rosser o confluencia, lo que provee un procedimiento teórico de semi-decisión para la igualdad. Esta garantía teórica, no obstante, deja abiertos algunos interrogantes de índole práctica, ya que el orden en el que se aplican las reglas de reescritura puede impactar considerablemente en el tiempo de ejecución requerido para obtener, en caso de que exista, la forma normal de un término, o sea, el resultado del cómputo que el término codifica. Considerando por ejemplo la función $f(x) = x + x$, hay varias maneras de calcular el resultado final de la expresión $f(2 * 3)$. Dos de ellas son las siguientes:

$$\begin{array}{lllllllll} f(2*3) & \to & 2*3+2*3 & \to & 6+2*3 & \to & 6+6 & \to & 12 \\ f(2*3) & \to & f(6) & \to & 6+6 & \to & 12 & & \end{array} \tag{1.1}$$

La primera secuencia de pasos de reducción, en este caso, sigue una estrategia de evaluación *call-by-name*: el argumento de la función se copia sin evaluar, lo que provoca que se duplique el trabajo de resolver el cómputo de la sub-expresión $2 * 3$. La segunda secuencia de pasos, en cambio, sigue una estrategia de evaluación *call-by-value*: el argumento de la función se evalúa hasta que llega a ser un valor, y recién entonces se producen dos copias de su resultado. En este caso, la primera secuencia de reducción es computacionalmente más onerosa que la segunda. Las estrategias de evaluación son, en general, mecanismos para elegir el orden en el que se aplican las reglas de reescritura. Una estrategia de evaluación podría requerir exponencialmente más trabajo que otra para alcanzar la forma normal de un término. Peor aún, podría darse el caso de que una estrategia de evaluación nunca alcance la forma normal de un término, a pesar de que dicha forma normal exista.

A partir de la década de 1970 surgió interés en la existencia de alguna estrategia de evaluación óptima, es decir, una estrategia que evite duplicar el trabajo computacional, y que realice únicamente los cómputos indispensables para alcanzar la forma normal de un término. Vuillemin estudió el problema de evaluación optimal en el marco de los *recursive program schemes* [Vui74, Vui75]. En su tesis doctoral de 1978 [Lév75], Lévy estudió este problema para el λ-cálculo y dio condiciones suficientes para obtener un mecanismo de evaluación optimal, basado en compartir cómputos que pertenecen a la misma familia, es decir, aquellos cómputos que tienen un origen común. El trabajo de Lévy dio origen a una sub-área de investigación en sí misma [Lam90, GAL92, Lan93, AL95, AM01, GK96, Gue99]. Recientemente, Balabonski estudió el problema de reducción optimal para un cálculo con patrones dinámicos [Bal10] (dichos cálculos serán introducidos a continuación) y demostró que la estrategia de reducción call-by-need es optimal para el caso de reducción débil [Bal13]. Guerrini y Solieri dieron cotas para el costo que introducen las estructuras que implementan el mecanismo de evaluación optimal [GS17].

Cálculos con patrones. Aunque el λ-cálculo puede representar, con codificaciones adecuadas, cualquier estructura de datos, los lenguajes de programación y los asistentes de prueba modernos extienden el lenguaje básico con construcciones primitivas para representar estas estructuras con el fin de evitar la ineficiencia inducida por una codificación puramente funcional, en general de difícil utilización y costosa en términos de recursos. Una de las extensiones clásicas es el mecanismo de *pattern matching*, concepto al que se le ha dedicado especial atención tanto en la programación funcional como en la programación orientada a objetos, y que ha dado lugar a diferentes cálculos con patrones (*pattern calculi*) [vO90, CK98, Kah03, CK04, JK06, Jay04, KvOdV08].

El pattern matching es un modo de abstracción que brinda la posibilidad de definir funciones de acuerdo a la forma de su argumento y que, al momento de aplicar una tal función, provee un mecanismo de análisis sintáctico que permite descomponer el argumento para el posterior uso de sus partes. Un ejemplo paradigmático es la función `sum` que suma los elementos de una lista de enteros, escrita en Haskell:

```
sum :: [Int] -> Int
sum []     = 0
sum (x:xs) = x + sum xs
```

En este ejemplo, para definir `sum` se utilizan dos patrones: `[]` que sólo hace match con la lista vacía, y `(x:xs)` que hace match con cualquier lista no vacía unificando `x` con la cabeza y `xs` con la cola de dicha lista. Los patrones pueden ser de una de dos clases: estáticos o dinámicos. Los primeros son aquellos que se especifican en tiempo de compilación mientras que los segundos pueden ser creados en tiempo de ejecución. El problema fundamental de cómo especificar, en un marco riguroso, patrón, datos y finalmente la correspondencia entre ellos es el objetivo que se persigue con los cálculos con patrones.

Los patrones tienen una fuerte presencia en computación, incluyendo áreas tales como Ingeniería de Software (específicamente diseño de arquitecturas), Inteligencia Artificial, Data Mining y Procesamiento de Imágenes (especialmente en la mecánica de reconocimiento). En lenguajes de programación funcional el uso de patrones para descomponer datos y acceder a sus partes para poder procesarlas es estándar. Además, variantes de ese mecanismo de implementación se están incorporando a lenguajes de programación orientada a objetos (como Scala). El procesamiento de datos semi-estructurados (como documentos XML) también se beneficia del uso de patrones. Recientemente han aparecido varios cálculos con patrones en la literatura [BCKL03, AMR06, KvOdV08, JK09, AMR09, Jay09] como consecuencia del interés mencionado y el hecho de que la interacción entre patrones/pattern matching y otros elementos de un lenguaje de programación dista de ser obvia. Esta situación se ve particularmente exacerbada en el caso de los cálculos con patrones dinámicos en donde los sistemas de tipos se encuentran notablemente ausentes.

Operadores de control. Otras extensiones usuales del λ-cálculo son aquellas que incorporan operadores de control. Estos operadores permiten manipular de manera explícita el contexto bajo el cual un programa es ejecutado. Fueron introducidos originalmente por Landin, con el operador **J** [Lan65a, Lan65b], ante la imposibilidad de transcribir de manera directa los *jumps* y *labels* en su estudio de la correspondencia entre ALGOL 60 y el λ-cálculo. Esto dio lugar posteriormente al estudio de un sinfín de nuevos operadores para manipular el flujo de control en la implementación de lenguajes de programación funcional [Rey70, Ste84, RC86, Rey98, SS98], y a un estilo particular de programación dentro de estos lenguajes conocido como *continuation-passing style* (CPS). En CPS, el control es pasado explícitamente a los programas como un parámetro formal llamado continuación. Por ejemplo, la siguiente función escrita en Haskell computa la suma de los elementos de una lista de enteros (invocando a la función `sum` anterior) y luego llama a la continuación recibida con el resultado:

```
sum' :: [Int] -> (Int -> a) -> a
sum' xs cont = cont (sum xs)
```

Este estilo de programación es capturado de un forma más pura por el $\lambda\mu$-cálculo de Parigot [Par92], el cual extiende al λ-cálculo con solo dos nuevos constructores: *comandos* $[\alpha]\, t$ y *μ-abstracciones* $\mu\alpha.c$: que deben ser entendidos como "llamar a la continuación α con t como argumento" y "guardar la continuación actual y continuar ejecutando c" respectivamente. Esta simple extensión es suficiente para capturar la esencia detrás de los operadores de control, permitiendo codificar a los mismos en términos del $\lambda\mu$-cálculo. Se destacan particularmente el operador **C** de Felleisen [FF87] y el llamado **call-cc** (*call with current continuation*) [Gri90], pues en el marco simplemente tipado resultan ser los testigos, vía el isomorfismo de Curry–Howard, de las fórmulas $\neg\neg A \to A$ y $((A \to B) \to A) \to A$ (Peirce's Law) respectivamente. Esto implica que el $\lambda\mu$-cálculo simplemente tipado es capaz de capturar la semántica no-constructiva de la lógica clásica, en contraposición a la lógica intuicionista isomorfa al λ-cálculo simplemente tipado.

1.1. Contribuciones

El presente trabajo introduce tres estudios semánticos realizados sobre el λ-cálculo y algunas de sus extensiones anteriormente mencionadas. Estas líneas de investigación buscan desarrollar la teoría subyacente a la formalización e implementación de un lenguaje de programación funcional que incorpore características novedosas respecto a los utilizados por la industria actualmente. A continuación se describen brevemente los estudios realizados:

1.1.1. Reducciones necesarias en λ-cálculo

En primer lugar se estudian la estrategia call-by-need [Wad71] del λ-cálculo y la noción semántica de *needed reduction* [BKKS87], con el objetivo de establecer alguna relación entre ambas. A modo ilustrativo, considerar la función $g(x, y) = x + x$, similar a f del ejemplo (1.1) pero tomando un segundo parámetro y que luego es descartado. Es claro entonces que en la expresión $g(2 * 3, f(8))$ no hay necesidad de computar $f(8)$ para obtener el resultado general. Estas sub-expresiones computables se denominan *redexes*. La noción semántica de needed reduction establece que una secuencia de pasos es needed si y solo si computa únicamente redexes necesarios para llegar al resultado. Por ejemplo

$$\begin{array}{lllllllll} g(2*3, f(8)) & \to & 2*3+2*3 & \to & 6+2*3 & \to & 6+6 & \to & 12 \\ g(2*3, f(8)) & \to & g(6, f(8)) & \to & 6+6 & \to & 12 & & \end{array} \tag{1.2}$$

son ambas secuencias needed, mientras que la siguiente contiene pasos espúreos

$$g(2 * 3, f(8)) \to g(6, f(8)) \to g(6, 8 + 8) \to g(6, 16) \to 6 + 6 \to 12$$

Por su parte, call-by-need es una estrategia de reducción determinista que busca computar el resultado de una expresión siempre que éste exista, evitando al mismo tiempo duplicar trabajo innecesariamente. En particular, la segunda secuencia en (1.2) bien podría considerarse la elegida por una estrategia call-by-need adecuada en el marco de este ejemplo. Notar que de ninguna manera debe reducirse el segundo parámetro de g puesto que esto puede resultar en la falla del cómputo en su totalidad: $g(2, 1/0)$.

Existen diversas presentaciones alternativas de la estrategia call-by-need [AFM$^+$95, AF97, MOW98, CF12, ABM14]. Para el alcance del estudio aquí realizado resulta de particular interés la formulación dada en [ABM14], basada en una extensión del λ-cálculo con sustituciones explícitas (ES), por la simpleza en su definición y, especialmente, porque un estudio reciente relaciona esta presentación puntual con la estrategia de reducción call-by-name, mostrando que ambas resultan equivalentes desde el punto de vista observacional [Kes16]. Es decir, un programa termina para la estrategia call-by-need si y solo si termina para la estrategia call-by-name. Cabe destacar, sin embargo, que todas estas presentaciones de call-by-need resultan operacionalmente equivalentes [BBBK17].

Se tiene particular interés en el resultado de [Kes16] puesto que se obtiene a través de un estudio semántico que involucra el uso de tipos intersección no idempotente. Esta clase particular de tipos fue presentada originalmente por Gardner [Gar94] como una variante de los tipos intersección (introducidos por Coppo y Dezani [CD78]) inspirada en la *lógica lineal* [Gir87]. Su relevancia radica en que permite realizar estudios *cuantitativos* [dC07] sobre las expresiones que denotan programas, tanto desde el punto de vista sintáctico como semántico. La literatura sobre aplicaciones puntuales de esta herramienta es extensa [Kfo00, KW04, LLD$^+$04, NM04, PR10a, PR10b, BR12, BL13, KV14, BKV17, KV17, AGK18, BKR18, KV19b, KV19a]. En el presente trabajo se apela a ellos para caracterizar los redexes needed de un término y así derivar distintas propiedades, siguiendo ideas de [Gar94].

En cuanto a la noción semántica de reducción necesaria, fue introducida originalmente por Barendregt et. al. [BKKS87] como parte de los estudios semánticos sobre optimalidad y eficiencia de las estrategias de reducción para el λ-cálculos derivados a partir de la tesis de Lévy [Lév75]. Resulta que caracterizar el conjunto de los redexes needed de una expresión es un problema no-decidible. Sin embargo [BKKS87] presenta aproximaciones computables de su solución mediante la caracterización de los llamados *spine redexes*. En este trabajo se busca relacionar el concepto de needed reduction con las secuencias de reducción resultantes de la estrategia call-by-need antes mencionada. Cabe destacar que esta última se define en el marco de la reducción débil, es decir que no se permite reducir dentro de expresiones que denotan funciones (*i.e.* λ-abstracciones), mientras que el concepto de needed reduction se presenta originalmente para la reducción fuerte (*i.e.* en cualquier posición interna del término). Para hacer posible una comparación entre ambos conceptos fue necesario adaptar esta noción semántica al caso débil. Esto constituye un aporte fundamental del trabajo realizado, puesto que en [Gar94] se presenta una caracterización de redexes needed similar a la aquí propuesta, pero que falla en relacionar los conceptos al no capturar correctamente la noción débil de reducción.

Los resultados obtenidos a partir del estudio realizado se presentan en el Cap. 2 y fueron publicados en [KRV18]. En pos de facilitar la lectura, ciertos detalles técnicos como definiciones o demostraciones auxiliares se postergan al Ap. A. Las principales contribuciones son:

1. La noción de reducción needed es presentada de manera uniforme para los casos de reducción full, head y weak-head (*cf.* Sec. 2.2.3).

2. Se muestra que las secuencias de reducción leftmost, head y weak-head (a la forma normal adecuada) representan una forma canónica de reducción needed para cada caso (Teo. 2.2.6), extendiendo así uno de los principales resultados de [BKKS87] a los casos más débiles con una prueba sencilla e intuitiva.

3. Para caracterizar los redexes weak-head needed mediante el uso de tipos intersección no idempotente es necesario extender la noción de reducción estándar sobre términos a árboles de derivación de tipos. Esto es formalizado de manera precisa en la Sec. 2.4.

4. Se prueba que el sistema de tipos utilizado permite identificar exactamente los redexes weak-head needed de un término en presencia de un *tipado principal* (Teo. 2.5.3 y 2.5.7). Esto lleva a su vez a la correcta caracterización de los términos normalizantes para la reducción weak-head needed (Teo. 2.6.3).

5. Combinando los resultados obtenidos con el de [Kes16] anteriormente mencionado se establece la equivalencia observacional entre la estrategia de reducción call-by-need y la noción semántica de needed reduction (Teo. 2.8.2).

1.1.2. Calculus of Applicative Patterns

En segundo término se pone el foco sobre los cálculos con patrones. Particularmente sobre una noción puntual atrapada por algunos de estos, llamada *path polymorphism*. Esta característica se refiere a la posibilidad de definir funciones capaces de atravesar de manera uniforme estructuras de datos arbitrarias. En otras palabras, permite definir *queries* (como un *update* o *lookup*) de manera genérica, sin importar *a priori* la estructura sobre la cual se esté trabajando. Por ejemplo, dada la representación uniforme estándar de estructuras de datos como aplicaciones binarias, al estilo de Haskell, suponer que se permite hacer pattern matching con un patrón de la forma `x y` que descompone arbitrariamente estas aplicaciones. Luego, podría definirse una función como

```
sum (Pt z) = z
sum (x y)  = (sum x) + (sum y)
sum w      = 0
```

donde `Pt` es un constructor de *puntos* a partir de un número entero. Esta función permite recorrer una estructura arbitraria (listas, árboles, etc.), calculando la suma de los puntos que la componen. Notar que cualquier constructor o término no aplicativo es capturado por la tercera rama, aportando `0` al resultado final.

El objetivo del estudio aquí realizado es desarrollar un sistema de tipos estático para un cálculo adecuado capaz de capturar esta característica. El sistema desarrollado se basa en una combinación no trivial de herramientas de tipado como tipos constantes, aplicativos, unión y recursivos; y en una noción novedosa de *compatibilidad de patrones*, que permite garantizar las buenas propiedades del cálculo, garantizando así el buen comportamiento dinámico del mismo. A su vez, se desarrolla un algoritmo de *type-checking*: dado un programa y una expresión de tipo, se verifica que la asignación de tal tipo al programa es cuestión sea válida en el sistema de tipos propuesto. El desarrollo realizado resulta en un algoritmo eficiente que involucra, a su vez, chequeo de equivalencia de tipos y sub-tipado, ambos basados en caracterizaciones co-inductivas de estas relaciones.

Las distintas etapas de este desarrollo se detallan en el Cap. 3 y fueron publicadas en [VBA15, EVB15, ABEV19]. Al igual que en el capítulo anterior, se postergan los detalles técnicos auxiliares a Ap. B.

La literatura sobre cálculos con patrones (tipados) es extensa. Se mencionan a continuación los trabajos más relevantes en relación al estudio aquí realizado. En [AMR06, AMR09] se propone el λ-*calculus with Constructors*, el cual adopta una noción de pattern matching diferente a la usual: utiliza un constructor de alternativa $\{c_1 \mapsto s_1, \ldots, c_n \mapsto s_n\} \cdot t$ en donde ciertas ocurrencias del constructor c_i en t son reemplazadas por sus correspondientes términos s_i. En [Pet09, Pet11] se estudia una disciplina tipada para el cálculo que garantiza que esta sustitución de constructores no resulta bloqueada por constantes fuera del dominio (*i.e. progress*). Esta disciplina de tipos incorpora polimorfismo paramétrico, sin embargo no considera path polymorphism. Otras dos grandes líneas de investigación ameritan ser comentadas: por un lado el trabajo de Jay y Kesner en el *Pure Pattern Calculus*; y por otro el ρ-cálculo de Kirchner et. al.

En [JK06, JK09] Jay y Kesner introducen el *Pure Pattern Calculus* (PPC): un formalismo donde puede abstraerse virtualmente cualquier término, permitiendo así computar patrones dinámicamente (pueden contener variables libres) y expresar funciones path-polimórficas. En [Jay09] Jay introduce un sistema de tipos para PPC, pero desafortunadamente ninguna de las propiedades fundamentales del mismo es estudiada (*i.e. subject reduction* o *strong normalization*). Más aún, se adopta una disciplina de tipos dinámica, la cual tiene un impacto directo en la semántica de reducción del cálculo. En otras palabras, muchas decisiones relativas al tipado son postergadas (en ocasiones innecesariamente) al tiempo de ejecución, resultando en la modificación de la semántica operacional del cálculo respecto a la variante no tipada. Esto resulta llamativo puesto que la noción de reducción de PPC es particularmente compleja y, al mismo tiempo, existen estudios sobre la factibilidad de definir estrategias normalizantes en el marco no tipado [BKLR17]. También se presenta en [Jay09] un sistema de tipos para una restricción de PPC a patrones estáticos (*Query Calculus*), que descarta el polimorfismo de patrones aunque manteniendo path polymorphism. Es este sistema el que introduce una noción novedosa de *matching* sobre tipos (luego extendida al sistema de PPC) que resulta en una modificación no trivial de la semántica operacional del cálculo. Desafortunadamente, esto conlleva una pérdida de la intuición detrás de la noción de reducción y no es claro hasta que punto el sistema de tipos captura path polymorphism de la manera esperada.

El ρ-cálculo [CK01] es un cálculo de patrones parametrizado sobre una teoría de matching. Si bien en principio permite capturar path polymorphism, ninguna de las distintas extensiones estudiadas en la literatura lo aborda [CKL01a, CKL01b, CKL02, CLW03, LW05, BCKL03]. En efecto, ninguno de los sistemas citados permite patrones de la forma $x\,y$. Esta limitación parece deberse al enfoque alternativo adoptado en los mismos, donde se tipan patrones como $\mathsf{c}\,x$ asignando a la constante c un tipo funcional *fijo*. Este enfoque parece incompatible con path polymorphism tal cual se lo entiende en el presente trabajo, puesto que no hay una forma clara de tipar el patrón $x\,y$, donde x detona una porción arbitraria de información (data) semi-estructurada.

En cuanto a desarrollo algorítmico realizado, existen numerosos trabajos relacionados al chequeo de equivalencia y sub-tipado de tipos recursivos [AC93, KPS95, BH98, JP97], aunque no se contempla la posibilidad de combinarlos con operadores asociativos, conmutativos e idempotentes (ACI) como es el caso de los tipos unión aquí utilizados. Estas ideas, entre otras, se encuentran resumidas en [Pie02], donde se presentan algoritmos genéricos para el chequeo de pertenencia a un conjunto definido coinductivamente y se propone, a su vez, introducir una variante dirigida por sintaxis del sistema de tipos. Por otro lado, en [PZ01] se estudia la posibilidad de incorporar productos asociativos e idempotentes al chequeo de equivalencia, en un enfoque basado en autómatas. Los mismos autores muestran en [Zha02] que este planteo no es extensible a sub-tipado. Finalmente, en [DPR05] se presenta otro algoritmo basado en autómatas capaz de chequear sub-tipado en presencia de productos AC con una complejidad computacional de $\mathcal{O}(n^2m^2d^{5/2})$, donde n y m son los tamaños de los tipos a analizar, y d es una cota para la aridad de los productos involucrados.

El presente trabajo pone el foco entonces sobre la característica path polimórfica de PPC, en pos de desarrollar un disciplina tipos estática capaz de capturar dicha noción garantizando la preservación de la semántica operacional del cálculo y las buenas propiedades dinámicas del sistema de tipos. Se introduce un nuevo formalismo, llamado *Calculus of Applicative Pattern* (CAP), que resulta equivalente en poder expresivo al fragmento estático de PPC, al mismo tiempo que incorpora la alternativa como un constructor nativo del cálculo y elimina el manejo explícito de la falla en la operación de matching. Las contribuciones desarrolladas sobre CAP pueden resumirse como:

1. Un sistema de tipos estático (*cf.* $\mathcal{P}$ en Sec. 3.3.3) que resulta de una combinación adecuada de tipos constantes, aplicativos, unión y recursivos; y satisface las propiedades fundamentales de *subject reduction* (Teo. 3.4.1) y *progress* (Teo. 3.4.3).

2. Relaciones de equivalencia y sub-tipado adecuadas para la combinación de herramientas mencio-

nada (*cf.* Sec. 3.3.1), de modo de garantizar la invertibilidad de esta última (Teo. 3.3.31).

3. Una noción novedosa de *compatibilidad de patrones* (*cf.* Sec. 3.3.4) sobre la cual recae crucialmente la adecuación del sistema $\mathcal{P}$.

4. Una variante dirigida por sintaxis del sistema $\mathcal{P}$, al igual que variantes co-inductivas de las relaciones de equivalencia y sub-tipado, adecuadas para el desarrollo de algoritmos de chequeo correctos y completos (Teo. 3.6.9, 3.6.13 y 3.6.14).

5. Reformulaciones de los algoritmos desarrollados adoptando una representación basada en autómatas para las expresiones de tipo, que resultan en versiones eficientes del chequeo de tipos, equivalencia y sub-tipado (Teo. 3.6.20 y 3.6.25, y Sec. 3.6.3).

1.1.3. Bisimulación fuerte para operadores de control

La tercera línea de investigación surge en el marco de un estudio sobre la factibilidad de definir una estrategia call-by-need (*cf.* Sec. 1.1.1) para el $\lambda\mu$-cálculo inspirada en el aspecto computacional, en contraposición a estudios realizados desde una mirada lógica [AHS11, ADH$^+$12, PS16]. Intuitivamente, desde el punto de vista computacional, una estrategia call-by-need tiene dos objetivos principales: por un lado encontrar la forma normal de un término siempre que ésta exista (tal como lo hace la estrategia call-by-name); y por otro evitar la duplicación innecesaria de trabajo (del mismo modo que la estrategia call-by-value). En este sentido se dice que call-by-need toma lo mejor de ambos mundos. Con tal fin es que se restringe la evaluación de los redexes de una expresión, lo que puede llevar a situaciones indeseadas donde un redex no sea requerido por la estrategia pero se encuentre bloqueando otro que sí lo es. Esto da lugar a una noción de equivalencia estructural que busca identificar aquellos términos que poseen esencialmente la misma semántica de reducción o, informalmente, resultan de permutaciones de los mismos redexes.

En el caso del λ-cálculo, esta noción fue desarrollada por Regnier [Reg94], dando lugar a la relación conocida como *σ-equivalencia*. La intuición subyacente proviene, vía el isomorfismo de Curry–Howard [SU06], de la *lógica lineal* y su representación como *proof-nets* [Gir87]. Más precisamente de su fragmento *multiplicativo/exponencial* (MELL). Una proof-net es una estructura en forma de grafo cuyos nodos denotan inferencias lógicas y sus ejes (o aristas) denotan fórmulas sobre las que estas inferencias operan. Vienen equipadas con su propia noción de equivalencia estructural y una semántica operacional basada en *cut-elimination*. La σ-equivalencia es capaz de identificar aquellos términos del λ-cálculo que son interpretados como la misma proof-net (módulo equivalencia estructural y cuts multiplicativos) [Reg94]. En [AK10] el resultado de Regnier es extendido a una presentación del λ-cálculo con *sustituciones explícitas* (ES), que permite identificar pasos de reducción multiplicativos y exponenciales en los propios términos del cálculo. Este nueva noción de σ-equivalencia, ahora sobre términos con ES, resulta ser una *bisimulación fuerte* respecto a la noción de reducción exponencial [ABKL14]. Formalmente, $\simeq_\sigma$ es simétrica al mismo tiempo que $t \simeq_\sigma t'$ y $t \rightsquigarrow s$ implican la existencia de s' tal que $s \rightsquigarrow s'$ y $t' \simeq_\sigma s'$, donde $\rightsquigarrow$ denota la noción de reducción exponencial sobre términos del λ con ES. Gráficamente:

$$\begin{array}{ccc} t & \simeq_\sigma & s \\ \vdots & & \vdots \\ t' & \simeq_\sigma & s' \end{array}$$

Esto implica que las secuencias de reducción entre términos σ-equivalentes tienen la misma longitud o, en otras palabras, términos equivalentes poseen *exactamente* la misma semántica de reducción.

La noción de equivalencia estructural desarrollada por Regnier fue extendida al $\lambda\mu$-cálculo por Laurent [Lau02, Lau03], utilizando como modelo subyacente las *proof-nets polarizadas* (PPNs) y capturando en consecuencia la *lógica clásica*. Se tiene entonces que términos del $\lambda\mu$-cálculo σ-equivalentes son interpretados por la misma PPN (módulo equivalencia estructural y cuts multiplicativos), pero al igual que el resultado original de Regnier para λ-cálculo, esta nueva relación de equivalencia para operadores de control no resulta en una bisimulación fuerte respecto a la noción de reducción del $\lambda\mu$-cálculo. Es decir, términos σ-equivalentes resultan de permutaciones de esencialmente los mismos redexes, pero no poseen exactamente la misma semántica de reducción. El estudio aquí realizado busca extender estas ideas de [AK12] al caso clásico, intentando refinar la noción de σ-equivalencia de Laurent al mismo tiempo que se busca hacer más granular la relación de reducción del $\lambda\mu$-cálculo, en pos de la correcta formulación de una relación de equivalencia estructural que resulte a su vez en una bisimulación fuerte.

Un último trabajo relacionado merece ser mencionado. En [KV17], el $\lambda\mu$-cálculo es reformulado en el $\lambda\mu\mathtt{r}$-cálculo con operadores explícitos, junto con una semántica operacional *small-step* para *sustituciones y replacements* a distancia. Si bien el objetivo es completamente diferente (los autores buscan formalizar un sistema de tipos intersección no idempotente para $\lambda\mu\mathtt{r}$-cálculo de modo de realizar un estudio sobre el uso de recursos por parte del mismo) el presente trabajo se inspira en la forma en que la operación de replacement del $\lambda\mu$-cálculo fue adaptada a un operador explícito.

La descomposición multiplicativa/exponencial del caso intuicionista aplicada en el marco clásico resulta no ser suficiente para identificar programas con operadores de control cuyas semánticas operacionales se correspondan exactamente. La semántica de reducción del replacement explícito requiere de una descomposición en principio sorpresiva. El análisis que se sigue es sutil, aunque admite una interpretación natural en PPNs. Más aún, permite obtener un resultado novedoso de bisimulación fuerte que dista de ser obvio, resaltando la profunda correspondencia entre las PPNs y los cálculos clásicos. Los resultados de esta línea de investigación se desarrollan en el Cap. 4 y se encuentran disponibles a su vez en [KBV20]. En este caso, los detalles técnicos pueden encontrarse en el Ap. C. Las principales contribuciones pueden resumirse como:

1. Un refinamiento del $\lambda\mu$-cálculo, llamado ΛM-cálculo, que incluye operadores explícitos de sustitución para variables, y replacement y renaming para nombres. Se prueba la confluencia de tal cálculo (Teo.4.2.2).

2. Una interpretación natural del ΛM-cálculo en las PPNs. Más precisamente, la reducción del ΛM-cálculo puede ser implementada por cut-elimination en las PPNs (Teo. 4.4.3), en el sentido que un paso de reducción en el ΛM-cálculo se traduce a una secuencia de reducción en las PPNs.

3. Una noción de equivalencia estructural $\simeq$ para el ΛM-cálculo que: (I) caracteriza las PPNs módulo equivalencia estructural (Teo.4.7.7); (II) es conservativa respecto de la noción de σ-equivalencia de Laurent (Teo. 4.7.8); y (III) es una bisimulación fuerte respecto a los pasos de reducción *meaningful* del ΛM-cálculo (Teo. 4.8.3).

1.2. Aspectos técnicos

Esta sección introduce algunos conceptos y herramientas teóricas que serán utilizadas recurrentemente a lo largo de la presente tesis. Estos constituyen aspecto técnicos generales, comunes a las tres líneas de investigación llevadas adelante. Se posterga un análisis detallado del estado del arte de cada linea particular al capítulo correspondiente.

1.2.1. El λ-cálculo

El λ-cálculo sienta las bases de la programación funcional y constituye el hilo conductor entre los diferentes estudios aquí realizados. Se introduce a continuación en su versión más general [Bar85].

Dado un conjunto infinito numerable de *variables* $\mathbb{V}$ $(x, y, \ldots)$, los conjuntos de *términos* $\mathbb{T}_\lambda$, *valores* y *contextos* se definen mediante la siguiente gramática:

$$\begin{array}{rrcl} \textbf{Términos} & t & ::= & x \in \mathbb{V} \mid t\,t \mid \lambda x.t \\ \textbf{Valores} & v & ::= & \lambda x.t \\ \textbf{Contextos} & \mathtt{C} & ::= & \Box \mid \mathtt{C}\,t \mid t\,\mathtt{C} \mid \lambda x.\mathtt{C} \end{array}$$

Ejemplos recurrentes de términos son $\lambda x.x$, $\lambda x.\lambda y.x$ y $(\lambda x.x\,x)\,(\lambda x.x\,x)$, llamados I, K y Ω respectivamente.

Se denota $\mathtt{C}\langle t\rangle$ al resultado de reemplazar $\Box$ por t en $\mathtt{C}$. Los conjuntos de *variables libres* y *ligadas* de un término t, denotados $\mathsf{fv}(t)$ y $\mathsf{bv}(t)$ respectivamente, se definen inductivamente como:

$$\begin{array}{rcl@{\qquad}rcl} \mathsf{fv}(x) & \triangleq & \emptyset & \mathsf{bv}(x) & \triangleq & \emptyset \\ \mathsf{fv}(t\,u) & \triangleq & \mathsf{fv}(t) \cup \mathsf{fv}(u) & \mathsf{bv}(t\,u) & \triangleq & \mathsf{bv}(t) \cup \mathsf{bv}(u) \\ \mathsf{fv}(\lambda x.t) & \triangleq & \mathsf{fv}(t) \setminus \{x\} & \mathsf{bv}(\lambda x.t) & \triangleq & \mathsf{bv}(t) \cup \{x\} \end{array}$$

Una variable x se dice *fresca* en un término t, notado $x \notin t$, si y solo si $x \notin \mathsf{fv}(t)$ y $x \notin \mathsf{bv}(t)$. Se trabaja con la noción estándar de α-conversión [Bar85], *i.e.* renombre de variables ligadas para las abstracciones; por lo que $\lambda x.x\,y =_\alpha \lambda z.z\,y$.

Una *sustitución* $(\rho, \sigma, \ldots)$ es una función parcial de variables en términos. Se denota $\sigma = \{x_i \backslash u_i\}_{i \in I}$, con I un conjunto de índices, a la sustitución que asigna u_i a x_i (*i.e.* $\sigma(x_i) \triangleq u_i$) para $i \in I$ y cuyo dominio e imagen de definen como $\mathsf{dom}(\sigma) \triangleq \bigcup_{i \in I} \{x_i\}$ y $\mathsf{img}(\sigma) \triangleq \bigcup_{i \in I} \{u_i\}$ respectivamente. Se define $\sigma(x) \triangleq x$ para toda variable $x \notin \mathsf{dom}(\sigma)$. La sustitución vacía es notada $\{\}$ o id.

Un término de la forma $(\lambda x.s)\,u$ es llamado *β-redex* (o simplemente *redex* cuando β es claro por contexto) y $\lambda x.$ es llamado el *ancla* del redex. La *relación de reducción en un paso* $\to_\beta$ está dada por la clausura por contextos $\mathtt{C}$ de la regla de reescritura

$$(\lambda x.s)\,u \quad \mapsto_\beta \quad \{x \setminus u\}s$$

donde $\{x \setminus u\}s$ representa la aplicación de la sustitución $\{x \setminus u\}$ al término t: *i.e.* el reemplazo de todas las ocurrencias libres de x en t por u, definida inductivamente de la siguiente manera:

$$\begin{array}{rcll} \sigma x & \triangleq & \sigma(x) & \\ \sigma(t\,v) & \triangleq & \sigma t\,\sigma v & \\ \sigma(\lambda y.t) & \triangleq & \lambda y.\sigma t & y \;\mathtt{evita}\; \sigma \end{array}$$

donde el predicado y $\mathtt{evita}$ σ expresa que no hay colisión entre y y las variables en σ (*i.e.* $y \notin \mathsf{dom}(\sigma) \cup (\bigcup_{x \in \mathsf{dom}(\sigma)} \mathsf{fv}(\sigma x))$), evitando así la captura indeseada de variables. Esta condición siempre puede satisfacerse apelando a α-conversión, mediante un simple renombre de la variable ligada y.

Notar que pueden darse diferentes secuencias de reducción desde un mismo término. Esta situación se ilustra en el siguiente ejemplo, donde se subraya el redex contraído a cada paso:

$$\begin{array}{lllllllll} \underline{(\lambda x.x\,x)\,(I\,I)} & \to_\beta & (I\,I)\,\underline{(I\,I)} & \to_\beta & \underline{(I\,I)}\,I & \to_\beta & \underline{I\,I} & \to_\beta & I \\ (\lambda x.x\,x)\,\underline{(I\,I)} & \to_\beta & \underline{(\lambda x.x\,x)\,I} & \to_\beta & \underline{I\,I} & \to_\beta & I & & \end{array}$$

Este ejemplo es similar al (1.1) presentado anteriormente, pero expresado ahora en términos del λ-cálculo, donde se puede apreciar la diferencia entre call-by-name y call-by-value. Esta diferencia puede

$$\frac{\Gamma(x) = \tau}{\Gamma \vdash x : \tau}\ (\textsc{var}) \qquad \frac{\Gamma \vdash t : \sigma \to \tau \quad \Delta \vdash u : \sigma}{\Gamma \cup \Delta \vdash t\, u : \tau}\ (\textsc{app}) \qquad \frac{\Gamma; x : \sigma \vdash t : \tau}{\Gamma \vdash \lambda x.t : \sigma \to \tau}\ (\textsc{abs})$$

Figura 1.1: Reglas de tipado del sistema $\mathcal{S}$ para el λ-cálculo.

verse exacerbada al punto tal que una estrategia *normalice* (*i.e.* encuentre un término irreducible) mientras que la otra diverja:

$$\begin{array}{llllll} (\underline{K\, I})\, \Omega & \to_\beta & \underline{(\lambda y.I)\, \Omega} & \to_\beta & I & \\ (\underline{K\, I})\, \Omega & \to_\beta & \underline{(\lambda y.I)\, \underline{\Omega}} & \to_\beta & (\lambda y.I)\, \underline{\Omega} & \to_\beta \quad \ldots \end{array}$$

Notar que $\Omega = (\lambda x.x\, x)\, (\lambda x.x\, x)$ no es un valor, por lo que la estrategia call-by-value buscará reducir el argumento de $\lambda y.I$ hasta encontrar un valor, sin éxito alguno.

Las distintas estrategias y nociones de reducción utilizadas a lo largo del presente trabajo serán introducidas oportunamente en el contexto de cada estudio. Sin embargo hay conceptos que son comunes a todas ellas que se detallan a continuación.

Dada una relación de reducción $\to_{\mathcal{R}}$ se escribe $\twoheadrightarrow_{\mathcal{R}}$ para denotar su clausura reflexiva-transitiva. El resultado esperado de evaluar un programa se especifica a través de nociones adecuadas de formas normales: un término t se dice en *$\mathcal{R}$-forma normal* ($t \in \mathcal{NF}_{\mathcal{R}}$) si y solo si no existe t' tal que $t \to_{\mathcal{R}} t'$. Un término t es *$\mathcal{R}$-normalizante* ($t \in \mathcal{WN}_{\mathcal{R}}$) si y solo si existe $u \in \mathcal{NF}_{\mathcal{R}}$ tal que $t \twoheadrightarrow_{\mathcal{R}} u$. Luego, dado un término $\mathcal{R}$-normalizante, puede definirse el *conjunto de $\mathcal{R}$-formas normales* de t como $\mathsf{nf}_{\mathcal{R}}(t) \triangleq \{t' \mid t \twoheadrightarrow_{\mathcal{R}} t', t' \in \mathcal{NF}_{\mathcal{R}}\}$.

1.2.2. El sistema de tipos simples para λ-cálculo

Si bien el sistema $\mathcal{S}$ de tipos simples para λ-cálculo no es utilizado explícitamente para los estudios aquí realizados, se lo presenta a continuación con el fin de introducir conceptos generales de sistemas de tipos e ilustrar sucintamente el isomorfismo de Curry–Howard.

Dado un conjunto infinito numerable de *variables de tipo* $\mathcal{V}$ ($\alpha, \beta, \gamma, \ldots$), se define el conjunto de *tipos simples* $\mathcal{T}$ mediante la siguiente gramática:

$$\textbf{(Tipos)} \quad \tau \quad ::= \quad \alpha \in \mathcal{V} \mid \tau \to \tau$$

Un *contexto de tipado* ($\Gamma, \Delta, \ldots$) es una función parcial de variables en tipos simples: $\Gamma(x) = A$. La *unión compatible de contextos* $\Gamma \cup \Delta$ se encuentra definida siempre que para todo $x \in \mathsf{dom}(\Gamma) \cap \mathsf{dom}(\Delta)$ se tenga $\Gamma(x) = \Delta(x)$. La *unión disjunta* de contextos de tipado se denota $\Gamma; \Delta$. Le *restricción* de un contexto Γ a un conjunto de variables θ, denotada $\Gamma|_\theta$, se define como la función parcial tal que $\Gamma|_\theta(x) \triangleq \Gamma(x)$ para todo $x \in \mathsf{dom}(\Gamma) \cap \theta$, y está indefinida para toda otra variable.

Un *juicio de tipado* tiene la forma $\Gamma \vdash t : \tau$, donde Γ es un contexto de tipado, t un término y τ un tipo. Las reglas de tipado para el sistema $\mathcal{S}$ del λ-cálculo están dadas en la Fig. 1.1.

El axioma (VAR) establece que una variable es tipable siempre que ésta pertenezca al dominio del contexto de tipado. Esta regla permite tener en el contexto hipótesis que prediquen sobre variables que no necesariamente ocurren libres en el término tipado, resultando en un sistema de tipos con *weakening*. La regla (APP) es aditiva: permite tipar la aplicación $t\, u$ siempre que los contextos de tipado de t y u sean compatibles, y tipo del dominio de t sea exactamente el tipo de u. Existen también variantes multiplicativas de esta regla (*cf.* Sec. 2.3 en el Cap. 2). Por su parte (ABS) tipa abstracciones descartando del contexto la variable abstraída.

Una *derivación de tipo* es un árbol obtenido por la aplicación inductiva de las reglas del sistema $\mathcal{S}$. La notación $\rhd_{\mathcal{S}} \Gamma \vdash t : \tau$ indica que existe una derivación para el juicio $\Gamma \vdash t : \tau$ en el sistema $\mathcal{S}$. Un

$$\frac{}{\Gamma;\tau \vdash \tau}\ (\text{AX}) \qquad \frac{\Gamma \vdash \sigma \to \tau \quad \Delta \vdash \sigma}{\Gamma \cup \Delta \vdash \tau}\ (\to\text{E}) \qquad \frac{\Gamma;\sigma \vdash \tau}{\Gamma \vdash \sigma \to \tau}\ (\to\text{I})$$

Figura 1.2: Cálculo de secuentes proposicional intuicionista.

término t es tipable en $\mathcal{S}$, o $\mathcal{S}$-tipable, si y solo si t es el *sujeto* de alguna derivación, *i.e.* si y solo si existen Γ y τ tal que $\rhd_{\mathcal{S}} \Gamma \vdash t : \tau$. Adicionalmente se utiliza la notación $\pi \rhd_{\mathcal{S}} \Gamma \vdash t : \tau$ para asignar el nombre π a la derivación del juicio de tipado $\Gamma \vdash t : \tau$ en el sistema $\mathcal{S}$. Por conveniencia, usualmente se abrevia con π_t a una derivación con sujeto t para algún contexto y tipo adecuado. Ejemplos de derivaciones válidas son:

$$\pi_I = \frac{\dfrac{}{x : \sigma \vdash x : \sigma}\ (\text{VAR})}{\vdash I : \sigma \to \sigma}\ (\text{ABS}) \qquad \pi_{II} = \frac{\dfrac{\dfrac{}{x : \sigma \to \sigma \vdash x : \sigma \to \sigma}\ (\text{VAR})}{\vdash I : (\sigma \to \sigma) \to (\sigma \to \sigma)}\ (\text{ABS}) \qquad \pi_I}{\vdash I\, I : \sigma \to \sigma}\ (\text{APP})$$

El *isomorfismo de Curry–Howard* [SU06] establece una relación entre estas derivaciones de tipos y los árboles de derivación para fórmulas de la lógica, en formalismos como el *Cálculo de Secuentes* [Gen35a, Gen35b]. Una mirada cercana permite ver que las reglas de la Fig. 1.1 se corresponden con aquellas del fragmento intuicionista de la lógica proposicional (*cf.* Fig. 1.2).

Más aún, el ejemplo ilustra una de las propiedades principales del isomorfismo: la correspondencia entre la normalización de pruebas y la reducción de términos en el λ-cálculo. Notar que ambas derivaciones π_I y π_{II} corresponden al mismo teorema $\sigma \to \sigma$. Sin embargo π_I resulta una derivación más corta y es, de hecho, el caso $I\, I \to_\beta I$. Más aún, I es la β-forma normal de $I\, I$. Es así que el isomorfismo de Curry–Howard establece que los programas (términos) son testigos de las demostraciones de los teoremas de la lógica subyacente. Para cada sistema de tipos posible la lógica asociada vía el isomorfismo resultará diferente, según la expresividad de cada formalismo.

1.2.3. Inducción y co-inducción

Se definen a continuación las nociones de inducción y co-inducción, siguiendo la presentación de [Pie02]. La primera constituye la herramienta principal a utilizar tanto para definir como demostrar propiedades a lo largo del presente trabajo. La segunda, en cambio, resulta esencial para trabajar con elementos infinitos (como al interpretar de μ-tipos contractivos en árboles infinitos en la Sec. 3.3.2 del Cap. 3) y será utilizada oportunamente y de manera más explícita.

Dado un conjunto de elementos $\mathcal{U}$ denominado *universo de discurso*, o simplemente *dominio*, los principios de *inducción* y *co-inducción* se utilizan para identificar sub-conjuntos de interés de $\mathcal{U}$.

Definición 1.2.1. *Una función* $\Phi : \wp(\mathcal{U}) \to \wp(\mathcal{U})$ *es* monótona *si* $\mathcal{X} \subseteq \mathcal{Y}$ *implica* $\Phi(\mathcal{X}) \subseteq \Phi(\mathcal{Y})$.

Las definiciones y propiedades presentadas a continuación asumen Φ monótona en $\wp(\mathcal{U})$ y se referirán a ella como la *función generadora*.

Definición 1.2.2. *Sea* $\mathcal{X} \in \wp(\mathcal{U})$*:*

1. $\mathcal{X}$ *es* Φ-cerrado *si* $\Phi(\mathcal{X}) \subseteq \mathcal{X}$.
2. $\mathcal{X}$ *es* Φ-denso *si* $\mathcal{X} \subseteq \Phi(\mathcal{X})$.
3. $\mathcal{X}$ *es un* punto fijo *de* Φ *si* $\Phi(\mathcal{X}) = \mathcal{X}$.

Intuitivamente, puede entenderse $\mathcal{U}$ como un conjunto de sentencias y Φ como la relación de "justificación" que, dado un conjunto de declaraciones (premisas), retorna nuevas sentencias (conclusiones) que se siguen de ellas. Luego, un conjunto Φ-cerrado es aquel que no puede ser incrementado mediante el agregado de elementos justificados por Φ: ya contiene todas las conclusiones que se siguen de sus elementos. Por otro lado, un conjunto Φ-denso está "auto-justificado": toda aseveración en él esta justificada por elementos del mismo conjunto. Un punto fijo de Φ es un conjunto cerrado y denso al mismo tiempo: incluye (exactamente) todas las justificaciones requeridas por sus elementos y todas las conclusiones que se siguen de ellos.

Considerar por ejemplo el la siguiente función generadora sobre el universo $\mathcal{U} \triangleq \{a, b, c\}$:

$$\begin{array}{rcl@{\qquad}rcl} \Phi_1(\emptyset) & \triangleq & \{c\} & \Phi_1(\{a,b\}) & \triangleq & \{c\} \\ \Phi_1(\{a\}) & \triangleq & \{c\} & \Phi_1(\{a,c\}) & \triangleq & \{b,c\} \\ \Phi_1(\{b\}) & \triangleq & \{c\} & \Phi_1(\{b,c\}) & \triangleq & \{a,b,c\} \\ \Phi_1(\{c\}) & \triangleq & \{b,c\} & \Phi_1(\{a,b,c\}) & \triangleq & \{a,b,c\} \end{array}$$

Φ_1 admite un único conjunto Φ_1-cerrado: $\{a, b, c\}$; mientras que hay cuatro Φ_1-densos: $\emptyset, \{c\}, \{b, c\} y \{a, b, c\}$. Alternativamente, Φ_1 puede representarse de manera compacta como un conjunto de *reglas de inferencia*

$$\frac{}{c} \qquad \frac{c}{b} \qquad \frac{b \quad c}{a}$$

donde cada regla establece que si todos los elementos arriba de la línea perteneces al conjunto de entrada, luego todos los de abajo pertenecen al de salida.

Teorema 1.2.3 (Knaster–Tarski [Tar55])**.**

1. *La intersección de todos los conjuntos Φ-cerrados es el menor punto fijo de Φ.*
2. *La unión de todos los conjuntos Φ-densos es el mayor punto fijo de Φ.*

Definición 1.2.4. *El* menor punto fijo *de Φ es notado $\mu\Phi$. El* mayor punto fijo *de Φ es notado $\nu\Phi$.*

Por ejemplo, para la función generadora Φ_1 anterior $\mu\Phi_1 = \nu\Phi_1 = \{a, b, c\}$.

Notar que $\mu\Phi$ es en sí mismo Φ-cerrado (por lo tanto, el menor conjunto Φ-cerrado) y que $\nu\Phi$ es Φ-denso (por lo tanto, el mayor conjunto Φ-denso), lo que da lugar a los principios de inducción y co-inducción respectivamente. Los mismos son corolarios directos del Teo. 1.2.3 y constituyen las herramientas fundamentales para los razonamientos inductivos y co-inductivos respectivamente.

Corolario 1.2.5.

1. Principio de inducción*: si $\mathcal{X}$ es Φ-cerrado, entonces $\mu\Phi \subseteq \mathcal{X}$.*
2. Principio de co-inducción*: si $\mathcal{X}$ es Φ-denso, entonces $\mathcal{X} \subseteq \nu\Phi$.*

La intuición detrás de estos principios proviene de interpretar $\mathcal{X}$ como un predicado, representado como su conjunto característico (*i.e.* el sub-conjunto de $\mathcal{U}$ para el cual el predicado es verdadero): mostrar que la propiedad $\mathcal{X}$ se satisface para un elemento x es equivalente a mostrar que $x \in \mathcal{X}$. Luego, el principio de inducción establece que cualquier propiedad cuyo conjunto característico sea cerrado por Φ (*i.e.* la propiedad es preservada por Φ) es verdadera para todos los elementos del conjunto inductivamente definido $\mu\Phi$. Por otro lado, El principio de co-inducción provee un método para establecer que un elemento x pertenece al conjunto co-inductivamente definido $\nu\Phi$. Para mostrar que $x \in \nu\Phi$ basta encontrar un conjunto $\mathcal{X}$ tal que $x \in \mathcal{X}$ y $\mathcal{X}$ sea Φ-denso.

$$\begin{array}{lcl} \mathsf{gfp}_0(\mathcal{X}) & \triangleq & \texttt{if } \mathsf{support}_\Phi(\mathcal{X})\uparrow \texttt{ then} \\ & & \quad \texttt{false} \\ & & \texttt{else if } \mathsf{support}_\Phi(\mathcal{X}) \subseteq \mathcal{X} \texttt{ then} \\ & & \quad \texttt{true} \\ & & \texttt{else } \mathsf{gfp}_0(\mathcal{X} \cup \mathsf{support}_\Phi(\mathcal{X})) \end{array}$$

Figura 1.3: Algoritmo co-inductivo ingenuo para chequeo de pertenencia.

A lo largo de la tesis se utiliza la presentación compacta como conjunto de reglas de inferencia para las funciones generadoras de distintas relaciones, distinguiendo aquellas que son interpretadas co-inductivamente mediante el uso de una doble línea en dichas reglas. Por ejemplo, las reglas:

$$\frac{}{c} \qquad \frac{c}{b} \qquad \frac{b \quad c}{a}$$

definen el conjunto $\nu\Phi_1 = \{a, b, c\}$.

1.2.4. Chequeo de pertenencia

Se tiene particular interés en chequear la pertenencia de un elemento a un conjunto $\nu\Phi$ definido co-inductivamente sobre un universo $\mathcal{U}$. A continuación se presenta un método [Pie02] que consiste, dado $x \in \mathcal{U}$, en construir un conjunto Φ-denso que contenga a x, para garantizar su pertenencia a $\nu\Phi$ por el principio de co-inducción (Cor. 1.2.5 (2)).

Se dice que un elemento es *generado* por un conjunto $\mathcal{X} \subseteq \mathcal{U}$ si $x \in \Phi(\mathcal{X})$. En principio x puede ser generado por Φ de muchas maneras distintas, *i.e.* pueden existir muchos conjuntos $\mathcal{X} \subseteq \mathcal{U}$ *generadores* de x. Dado que la función generadora Φ es por definición monótona, cualquier conjunto $\mathcal{X}'$ tal que $\mathcal{X} \subseteq \mathcal{X}'$ es a su vez generador de x, lo que lleva a considerar conjuntos generadores minimales. Más aún, si se consideran funciones generadoras *invertibles* se tiene a lo sumo un conjunto generado minimal.

Definición 1.2.6. *Una función generadora Φ se dice* invertible *si, para todo $x \in \mathcal{U}$, el conjunto $G_x = \{\mathcal{X} \subseteq \mathcal{U} \mid x \in \Phi(\mathcal{X})\}$ es vacío o contiene un único elemento que es sub-conjunto de todos los demás.*

La noción de conjunto soporte de una función generadora invertible permite obtener, dado un $x \in \mathcal{U}$, el conjunto mínimo de elementos cuya pertenencia a $\nu\Phi$ se debe verificar para garantizar la de x.

Definición 1.2.7. *El conjunto* soporte *de una función generadora Φ invertible se define como:*

$$\mathsf{support}_\Phi(x) \quad \triangleq \quad \begin{cases} \mathcal{X} & \textit{si } \mathcal{X} \in G_x \textit{ y } \forall \mathcal{X}' \in G_x . \mathcal{X} \subseteq \mathcal{X}' \\ \uparrow & \textit{si } G_x = \emptyset \end{cases}$$

La definición se extiende a conjuntos como $\mathsf{support}_\Phi(\mathcal{X}) \triangleq \bigcup_{x \in \mathcal{X}} \mathsf{support}_\Phi(x)$*, resultando indefinida si alguno de los elementos de la unión lo está.*

Continuando con el ejemplo de la Sec. 1.2.3, se tiene $\mathsf{support}_{\Phi_1}(\{a, b\}) = \{b, c\}$ y $\mathsf{support}_{\Phi_1}(\{c\}) = \emptyset$.

Estos conceptos sugieren un algoritmo sencillo para verificar si un conjunto de elementos está incluido en el mayor punto fijo (gfp por su sigla en Inglés) de una función generadora Φ invertible, *cf.* Fig. 1.3. Dicho algoritmo consiste esencialmente en "ejecutar Φ al revés", agregando a cada paso el conjunto minimal de premisas necesarias para justificar la pertenencia de los elementos acumulados

$$
\begin{array}{lcl}
\mathsf{gfp}_1(\mathcal{S},\mathcal{X}) & \triangleq & \texttt{if } \mathcal{X} = \emptyset \texttt{ then} \\
& & \quad \texttt{true} \\
& & \texttt{else let } x \in \mathcal{X} \texttt{ in} \\
& & \quad \texttt{if } x \in \mathcal{S} \texttt{ then} \\
& & \quad\quad \mathsf{gfp}_1(\mathcal{S}, \mathcal{X} \setminus \{x\}) \\
& & \quad \texttt{elseif } \mathsf{support}_{\Phi}(x)\uparrow \texttt{ then} \\
& & \quad\quad \texttt{false} \\
& & \quad \texttt{else} \\
& & \quad\quad \mathsf{gfp}_1(\mathcal{S} \cup \{x\}, (\mathcal{X} \cup \mathsf{support}_{\Phi}(x)) \setminus (\mathcal{S} \cup \{x\}))
\end{array}
$$

Figura 1.4: Algoritmo co-inductivo mejorado para chequeo de pertenencia.

$$
\begin{array}{lcl}
\mathsf{gfp}(\mathcal{S},x) & \triangleq & \texttt{if } x \in \mathcal{S} \texttt{ then} \\
& & \quad \mathcal{S} \\
& & \texttt{elseif } \mathsf{support}_{\Phi}(x)\uparrow \texttt{ then} \\
& & \quad \texttt{fail} \\
& & \texttt{else} \\
& & \quad \texttt{let } \{x_1,\ldots,x_n\} = \mathsf{support}_{\Phi}(x) \texttt{ in} \\
& & \quad \texttt{let } \mathcal{S}_0 = \mathcal{S} \cup \{x\} \texttt{ in} \\
& & \quad \texttt{let } \mathcal{S}_1 = \mathsf{gfp}(\mathcal{S}_0, x_1) \texttt{ in} \\
& & \quad \ldots \\
& & \quad \texttt{let } \mathcal{S}_n = \mathsf{gfp}(\mathcal{S}_{n-1}, x_n) \texttt{ in} \\
& & \quad\quad \mathcal{S}_n
\end{array}
$$

Figura 1.5: Algoritmo co-inductivo final para chequeo de pertenencia.

hasta el momento. Al obtener un conjunto de premisas Φ-denso se concluye por principio de co-inducción. De encontrarse indefinido el conjunto soporte para los elementos buscados, se concluye que no es posible justificar el conjunto original. Para el ejemplo de Φ_1, la pertenencia de b a $\nu\Phi_1$ se verifica con la siguiente ejecución:

$$\mathsf{gfp}_0(\{b\}) = \mathsf{gfp}_0(\{b,c\}) = \texttt{true}$$

En [Pie02] se demuestra la corrección y completitud de este algoritmo, al mismo tiempo que se proponen una serie de mejoras incrementales para disminuir la complejidad del mismo. Observar en primer lugar que $\mathsf{support}_{\Phi}(b)$ es computado en ambos llamados a gfp_0 del ejemplo anterior. Puede evitarse recalcular el conjunto soporte innecesariamente si se distingue entre los elementos que ya fueron analizados y los que no. En la segunda versión del algoritmo (gfp_1, *cf.* Fig. 1.4) los conjuntos $\mathcal{S}$ y $\mathcal{X}$ representan respectivamente a estos dos grupos de elementos. A cada paso, al transferir un elemento x de $\mathcal{X}$ a $\mathcal{S}$ se agregan a $\mathcal{X}$ los elementos del soporte de x que aún no fueron examinados. De esta forma se busca explorar únicamente los nuevos elementos necesarios para construir, eventualmente, un conjunto Φ-denso $\mathcal{S}$. Con estas mejoras el seguimiento del ejemplo anterior es ahora:

$$\mathsf{gfp}_1(\emptyset,\{b\}) = \mathsf{gfp}_1(\{b\},\{c\}) = \mathsf{gfp}_1(\{b,c\},\emptyset) = \texttt{true}$$

evitando así recalcular $\mathsf{support}_{\Phi_1}(b)$ innecesariamente.

Una última modificación es propuesta en [Pie02], donde se trabaja sobre un único elemento x a verificar y, en lugar de computar un resultado booleano, se retorna el conjunto Φ-denso $\mathcal{S}$ en caso de éxito, o de lo contrario falla. Esto permite tener un testigo de la pertenencia del elemento x buscado

al conjunto co-inductivo $\nu\Phi$. La versión final de gfp se presenta en la Fig. 1.5, donde se reemplaza el conjunto de trabajo $\mathcal{X}$ por un único elemento x a verificar. Al inspeccionar el soporte de x, se realizan todos los llamados recursivos correspondientes, devolviendo siempre el conjunto de premisas aumentado para ser utilizado en el próximo paso. Así, $\mathsf{gfp}(\emptyset, x) = \mathcal{S}$ no solo garantiza $x \in \nu\Phi$, sino que además se tiene como testigo el conjunto $\mathcal{S} \subseteq \Phi(\mathcal{S})$. En caso se falla se concluye $x \notin \nu\Phi$, pues el algoritmo se demuestra correcto y completo [Pie02].

Capítulo 2

Reducciones necesarias en λ-cálculo

Este capítulo se centra en la primera línea de investigación propuesta: el estudio de estrategias de evaluación para λ-cálculo por medio de un sistema de tipos intersección (no idempotente). En particular, se busca establecer una relación entre la noción semántica de reducción necesaria [BKKS87] y la estrategía de reducción call-by-need [Wad71] formalizada de manera sintáctica.

Dos nociones fundamentales detrás de este capítulo son las de programas reducibles o no reducibles: los primeros son programas (representados por λ-términos) que contienen sub-programas no evaluados, llamados expresiones reducibles (*redexes*), mientras que los últimos pueden ser vistos como resultados definitivos de una computación, llamados *formas normales*. Todo programa reducible contiene una especie particular de redexes llamados *needed* o, en otras palabras, todo λ-términos que no se encuentra en forma normal contiene un redex needed. Un redex r se dice *needed* en un λ-término t si r tiene que ser contraído (*i.e.* evaluado) tarde o temprano al reducir t a forma normal, o, informalmente, no hay forma de evitar r para alcanzar una forma normal.

La estrategia necesaria, que siempre contrae redexes needed, es normalizante [BKKS87], *i.e.* si un término puede ser reducido (de cualquier manera) a una forma normal, entonces la contracción de redexes needed necesariamente termina. Éste resulta ser un excelente punto de partida para diseñar estrategias de evaluación pero, desafortunadamente, la *neededness* de un redex no es decidible [BKKS87]. Como consecuencia, implementaciones reales de lenguajes de programación deben apelar a formas indirectas de garantizar que los redexes contraídos al evaluar un programa son efectivamente necesarios para el cómputo.

El objetivo de este trabajo, sin embargo, es establecer una clara conexión entre la noción semántica de *neededness* y diferentes implementaciones de lenguajes funcionales lazy (*e.g.* Miranda o Haskell). Tales implementaciones se basan en cálculos *call-by-need*, promovidos por Wadsworth [Wad71], y extensivamente estudiados en [AFM$^+$95]. Precisamente, los cálculos call-by-need llenan el espacio entre la conocida semántica operacional del λ-cálculo call-by-name y las implementaciones concretas de lenguajes funcionales lazy. Mientras call-by-name re-evalúa un argumento cada vez que éste es usado –una operación que puede resultar costosa– call-by-need puede ser visto como una versión de call-by-name con *memoization*, donde el valor de un argumento es guardado la primera vez que éste es evaluado para sus usos subsiguientes. Por ejemplo, si $t = \Delta\,(I\,I)$, donde $\Delta = \lambda x.x\,x$ e $I = \lambda x.x$, entonces call-by-name duplica el argumento $I\,I$, mientras que los lenguajes lazy primero reducen $I\,I$ al valor I de modo que futuros usos de este argumento no requieren volver a evaluarlo.

Mientras que la noción de reducción needed está definida respecto a formas normales, los cálculos call-by-need evalúan programas a valores especiales llamados *formas normales weak-head*: *i.e.* abstracciones o bien aplicaciones encabezadas por una variables (términos de la forma $x\,t_1 \ldots t_n$ donde $t_1, \ldots, t_n$ son términos arbitrarios). Para sobrepasar este déficit, se adapta la noción de redex needed

para el caso de la reducción weak-head. Por eso, informalmente, un redex $\mathtt{r}$ es *weak-head needed* en un término t si $\mathtt{r}$ debe ser contraído tarde o temprano al reducir t a una forma normal weak-head. La noción de estrategia derivada es llamada *estrategia weak-head needed*, y siempre contrae redexes weak-head needed.

El presente trabajo introduce dos resultados independientes sobre weak-head neededness, ambos obtenidos por medio de tipos intersección (no idempotente) [Gar94, dC07, BKV17]. Se considera, en particular, el sistema de tipos $\mathcal{N}$ [Kes16] (llamado originalmente $\mathcal{V}$) y se muestra que éste permite identificar todos los redexes weak-head needed de un término weak-head normalizante. Esto se hace adaptando la noción clásica de *tipado principal* [Roc88] y probando que un redex en un término weak-head normalizante t es weak-head needed si y solo si es tipado en una derivación de tipado principal para t en $\mathcal{N}$. El segundo objetivo es mostrar la equivalencia observacional entre call-by-need y la reducción weak-head needed. Dos términos son observacionalmente equivalentes cuando todas los cómputos empíricamente testeables sobre ellos son idénticos. Esto implica que un término t puede ser evaluado a una forma normal weak-head usando la maquinaria call-by-need si y solo si t normaliza con la reducción weak-head needed.

Mediante el sistema $\mathcal{N}$ mencionado anteriormente, se utiliza una técnica para razonar sobre equivalencia observacional que es flexible, general y fácil de verificar o incluso certificar. En efecto, el sistema $\mathcal{N}$ provee un argumento semántico: primero mostrando que un término t es tipable en $\mathcal{N}$ si y solo si t es normalizante para la estrategia weak-head needed ($t \in \mathcal{WN}_{\mathtt{whnd}}$); luego, apelando a algunos resultados de [Kes16], mostrando que el sistema $\mathcal{N}$ es correcto y completo para call-by-name: *i.e.* un término t es tipable en $\mathcal{N}$ si y solo si t es normalizante para call-by-name ($t \in \mathcal{WN}_{\mathtt{name}}$); y por último mostrando que t es normalizante para call-by-name si y solo si t es normalizante para call-by-need ($t \in \mathcal{WN}_{\mathtt{need}}$). Completando de este modo la siguiente cadena de equivalencias:

$$t \in \mathcal{WN}_{\mathtt{whnd}} \iff t \text{ tipable en } \mathcal{N} \iff t \in \mathcal{WN}_{\mathtt{name}} \iff t \in \mathcal{WN}_{\mathtt{need}}$$

Esto lleva a la equivalencia observacional entre call-by-need, call-by-name y la reducción weak-head needed.

2.1. Preliminares

Esta sección introduce algunas definiciones y nociones estándar relacionadas a las estrategias de reducción estudiadas en este capítulo, a saber: call-by-name, reducción head y weak-head, y neededness. Esta última se basa en la *teoría de residuos* [Bar85].

2.1.1. El λ-cálculo call-by-name

El λ-cálculo call-by-name consiste en restringir, de manera sintáctica, la aplicación de la regla de reescritura β (*cf.* Sec. 1.2.1) a los llamados *contextos name*:

$$\textbf{Contextos name} \quad \mathtt{E} \quad ::= \quad \Box \mid \mathtt{E}\, t$$

De este modo, la *relación de reducción en un paso* $\to_{\mathtt{name}}$ está dada por la clausura por contextos $\mathtt{E}$ de la regla de reescritura β. Una consecuencia inmediata de esta definición es que call-by-name prohíbe la reducción dentro de argumentos y abstracciones, *e.g.*

$$\begin{array}{lll} (\lambda x.I\, I)\,(\underline{I\, I}) & \to_\beta & (\lambda x.I\, I)\, I \\ (\lambda x.\underline{I\, I})\,(I\, I) & \to_\beta & (\lambda x.I)\, I\, I \end{array}$$

mientras que

$$\begin{array}{lll} (\lambda x.I\, I)\,(\underline{I\, I}) & \not\to_{\mathtt{name}} & (\lambda x.I\, I)\, I \\ (\lambda x.\underline{I\, I})\,(I\, I) & \not\to_{\mathtt{name}} & (\lambda x.I)\, I\, I \end{array}$$

2.1.2. Reducciones head, weak-head y leftmost

Para introducir adecuadamente diferentes nociones de reducción, se formaliza primero un mecanismo general que consiste en contraer un redex en una posición específica. Una *posición* es una palabra finita sobre el alfabeto $\{0, 1\}$. Se denota con ϵ la palabra vacía y con a^n la concatenación de $n \in \mathbb{N}$ copias de la palabra a.

Definición 2.1.1. *El conjunto de* posiciones de un término t, *notado* $\mathsf{pos}(t)$, *se define como:*

$$\begin{array}{rcl} \mathsf{pos}(x) & \triangleq & \{\epsilon\} \\ \mathsf{pos}(s\,u) & \triangleq & \{\epsilon\} \cup \{0\mathsf{p} \mid \mathsf{p} \in \mathsf{pos}(s)\} \cup \{1\mathsf{p} \mid \mathsf{p} \in \mathsf{pos}(u)\} \\ \mathsf{pos}(\lambda x.s) & \triangleq & \{\epsilon\} \cup \{0\mathsf{p} \mid \mathsf{p} \in \mathsf{pos}(s)\} \end{array}$$

Dadas dos posiciones p y q, se dice que p es *prefijo* de q, notado $\mathsf{p} \leq \mathsf{q}$, si existe p' tal que $\mathsf{pp}' = \mathsf{q}$. El *sub-término* de t en posición p se denota $t|_{\mathsf{p}}$ y se define inductivamente como:

$$\begin{array}{rcl} t|_{\epsilon} & \triangleq & t \\ (t\,u)|_{0\mathsf{q}} & \triangleq & t|_{\mathsf{q}} \\ (t\,u)|_{1\mathsf{q}} & \triangleq & u|_{\mathsf{q}} \\ (\lambda x.t)|_{0\mathsf{q}} & \triangleq & t|_{\mathsf{q}} \end{array}$$

Por ejemplo, $((\lambda x.y)\,z)|_{00} = y$.

Definición 2.1.2. *El conjunto de* posiciones de redexes *de un término t se define como* $\mathsf{rpos}(t) \triangleq \{\mathsf{p} \in \mathsf{pos}(t) \mid t|_{\mathsf{p}} = (\lambda x.s)\,u\}$.

Se usa la notación $\mathsf{r} : t \to_\beta t'$ para expresar que $\mathsf{r} \in \mathsf{rpos}(t)$ y t reduce a t' *contrayendo* el redex en posición r, *e.g.* $000 : (\lambda x.(\lambda y.y)\,x\,x)\,z \to_\beta (\lambda x.x\,x)\,z$. Esta noción se extiende intuitivamente a secuencias de reducción, notado $\rho : t \twoheadrightarrow_\beta t'$, donde ρ es una lista de todas las posiciones de redexes contraídos a lo largo de la secuencia. Por ejemplo, $0, 1, \epsilon : (\underline{I\,I})\,(I\,I) \to_\beta I\,(\underline{I\,I}) \to_\beta \underline{I\,I} \to_\beta I$. Se usa *nil* para denotar la secuencia de reducción vacía, de modo que $nil : t \twoheadrightarrow_\beta t$ se satisface para todo término t.

Todo término t del λ-cálculo tiene exactamente una de las siguientes formas: $\lambda x_1 \ldots \lambda x_n.y\,t_1 \ldots t_m$ o $\lambda x_1 \ldots \lambda x_n.(\lambda y.s)\,u\,t_1 \ldots t_m$ con $n, m \geq 0$. En el segundo caso se dice que $(\lambda y.s)\,u$ es el *head redex* de t, mientras que en el primero no hay head redex. Más aún, si $n = 0$, se dice que $(\lambda y.s)\,u$ es el *weak-head redex* de t. En términos de posiciones, el head redex de t es la posición de redex de la forma 0^n minimal, con $n \geq 0$. En particular, si éste satisface que $t|_{0^k}$ no es una abstracción para todo $k < n$, entonces es el weak-head redex de t. Una secuencia de reducción que contrae a cada paso el head redex (weak-head redex) del término correspondiente es llamada una *reducción head* (*reducción weak-head* respectivamente).

Dadas dos posiciones de redexes $\mathsf{r}, \mathsf{r}' \in \mathsf{rpos}(t)$, se dice que r está *a la izquierda* de r' si el ancla de r se encuentra más a la izquierda que el ancla de r' en la escritura del término t. Por ejemplo, la posición de redex 0 está a la izquierda de 1 en $(I\,x)\,(I\,y)$, mientras que ϵ está a la izquierda de 00 en $(\lambda x.(I\,I))\,z$. Alternativamente, la relación "a la izquierda" puede ser entendida como el orden de diccionario entre posiciones de redexes, es decir: r está a la izquierda de r' si $\mathsf{r}' = \mathsf{rq}$ con $\mathsf{q} \neq \epsilon$ (*i.e.* r es prefijo de r'); o bien $\mathsf{r} = \mathsf{p}0\mathsf{q}'$ y $\mathsf{r}' = \mathsf{p}1\mathsf{q}$ (*i.e.* comparten un prefijo común y r está en la rama izquierda de una aplicación, mientras que r' está en la rama derecha). Notar que en cualquier caso esto implica $\mathsf{r}' \not\leq \mathsf{r}$. Dado que esta noción define un orden total entre redexes, todo término no en forma normal tiene un único *leftmost redex*. El término t *reduce leftmost* a t' si t reduce a t' y el paso de reducción contrae el leftmost redex de t. Por ejemplo, $(I\,x)\,(I\,y)$ reduce leftmost a $x\,(I\,y)$ y $(\lambda x.(I\,I))\,z$ reduce leftmost a $I\,I$. Esta noción se extiende a secuencias de reducción como es esperado.

2.2. Reducciones needed

La reducción needed está basada en dos nociones fundamentales: la de residuos, que describe cómo un redex dado es rastreado a lo largo de una secuencia de reducción, y la de forma normal, que caracteriza los resultados esperados luego de una secuencia de reducción. En esta sección se extiende la noción estándar de reducción needed [BKKS87] a los casos head y weak-head needed.

2.2.1. Residuos

La noción de residuo se basa en la de descendientes de una posición, que permite hacer un seguimiento sobre cómo dicha posición es borrada, duplicada o modificada al contraer un redex.

Definición 2.2.1. *Los* descendientes *de una posición* p *luego de contraer un redex* r *en un término* t*, notado* p/r*, es el conjunto de posiciones definido como:*

$$\begin{array}{rl} \emptyset & \textit{si } \mathsf{p} = \mathsf{r} \textit{ o } \mathsf{p} = \mathsf{r}0 \\ \{\mathsf{p}\} & \textit{si } \mathsf{r} \not\leq \mathsf{p} \\ \{\mathsf{rq}\} & \textit{si } \mathsf{p} = \mathsf{r}00\mathsf{q} \\ \{\mathsf{rkq} \mid s|_{\mathsf{k}} = x\} & \textit{si } \mathsf{p} = \mathsf{r}1\mathsf{q} \textit{ con } t|_{\mathsf{r}} = (\lambda x.s)\, u \end{array}$$

Por ejemplo, dado $t = (\lambda x.(\lambda y.x)\, x)\, z$ se tienen $\mathsf{pos}(t) = \{\epsilon, 0, 1, 00, 000, 001, 0000\}$, $\mathsf{rpos}(t) = \{\epsilon, 00\}$, $00/00 = \emptyset$, $\epsilon/00 = \{\epsilon\}$, $00/\epsilon = \{\epsilon\}$ y $1/\epsilon = \{1, 000\}$.

Notar que $\mathsf{p}/\mathsf{r} \subseteq \mathsf{pos}(t')$ donde $\mathsf{r} : t \to_\beta t'$. Más aún, si p es la posición de un redex en t (*i.e.* $\mathsf{p} \in \mathsf{pos}(t)$), entonces $\mathsf{p}/\mathsf{r} \subseteq \mathsf{rpos}(t')$, y cada posición en p/r es llamada un *residuo* de p luego de reducir r. Esta noción se extiende a conjuntos de posiciones de redexes de modo que los *residuos* de $\mathcal{P}$ luego de r en t son $\mathcal{P}/\mathsf{r} \triangleq \bigcup_{\mathsf{p}\in\mathcal{P}} \mathsf{p}/\mathsf{r}$. En particular, $\emptyset/\mathsf{r} = \emptyset$. Dada la secuencia de reducción $\rho : t \to_\beta t'$ y el conjunto $\mathcal{P} \subseteq \mathsf{rpos}(t)$, los *residuos* de $\mathcal{P}$ luego de la secuencia ρ son: $\mathcal{P}/nil \triangleq \mathcal{P}$ y $\mathcal{P}/\mathsf{r}\rho' \triangleq (\mathcal{P}/\mathsf{r})/\rho'$.

La estabilidad de la relación "a la izquierda" hace uso de la noción de residuo:

Lema 2.2.2. *Sean* $\mathsf{l}, \mathsf{r}, \mathsf{s} \in \mathsf{rpos}(t)$ *tal que* l *está a la izquierda de* r*,* $\mathsf{s} \not\leq \mathsf{l}$ *y* $\mathsf{s} : t \to_\beta t'$*. Entonces,* $\mathsf{l} \in \mathsf{rpos}(t')$ *y* l *está a la izquierda de* r' *para toda* $\mathsf{r}' \in \mathsf{r}/\mathsf{s}$*.*

Notar que este resultado no solo implica que el leftmost redex se preserva por reducción de otros redexes, sino que además los residuos del leftmost redex ocurren en exactamente las mismas posiciones que el original.

Corolario 2.2.3. *Sea* $\mathsf{l} \in \mathsf{rpos}(t)$ *el leftmost redex de* t*. Si la reducción* $\rho : t \to_\beta t'$ *no contrae a* l *ni a ninguno de sus residuos, entonces* $\mathsf{l} \in \mathsf{rpos}(t')$ *es el leftmost redex de* t'*.*

2.2.2. Nociones de forma normal

Volviendo sobre la definición de forma normal introducida en la Sec. 1.2.1 para una relación de reducción arbitraria $\mathcal{R}$ en el λ-cálculo, se presentan aquí nociones de forma normal específicas para las reducciones weak-head, head y full.

En particular, resulta ser que un término en *β-forma normal weak-head* ($\mathcal{WHNF}_\beta$) es de la forma $x\, t_1 \ldots t_n$ ($n \geq 0$) o $\lambda x.t$, donde $t, t_1, \ldots, t_n$ son términos arbitrarios, *i.e.* el término no tiene weak-head redex. El conjunto de *β-formas normales weak-head* de t es entonces $\mathsf{whnf}_\beta(t) \triangleq \{t' \mid t \twoheadrightarrow_\beta t', t' \in \mathcal{WHNF}_\beta\}$.

Así mismo, un término en *β-forma normal head* ($\mathcal{HNF}_\beta$) resulta ser de la forma $\lambda x_1 \ldots \lambda x_n . x\, t_1 \ldots t_m$ ($n, m \geq 0$), *i.e.* no tiene head redex. El *conjunto de β-formas normales head* de t está dado por $\mathsf{hnf}_\beta(t) \triangleq \{t' \mid t \twoheadrightarrow_\beta t', t' \in \mathcal{HNF}_\beta\}$.

Finalmente, todo término en *β-forma normal (full)* ($\mathcal{NF}_\beta$) tiene la forma $\lambda x_1 \ldots \lambda x_n . x\, t_1 \ldots t_m$ $(n, m \geq 0)$ donde $t_1, \ldots, t_m$ son a su vez β-formas normales. Es sabido, por confluencia $\to_\beta$ [Bar85], que dado un término normalizante t, el conjunto $\mathsf{nf}_\beta(t)$ es unitario. Luego, se utilizará esta notación para referirse tanto al conjunto como a su único elemento según convenga.

Cabe destacar que $\mathcal{NF}_\beta \subseteq \mathcal{HNF}_\beta \subseteq \mathcal{WHNF}_\beta$. De hecho, las inclusiones son estrictas: por ejemplo $\lambda x.(\lambda y.y)\, z$ está en β-forma normal weak-head pero no head, mientas que $x\,((\lambda y.y)\, x)\, z$ está en β-forma normal head pero no full.

2.2.3. Nociones de reducción needed

Las diferentes nociones de forma normal consideradas en la Sec. 2.2.2 sugieren diferentes nociones de reducción needed, más allá de la estándar en la literatura [BKKS87].

Definición 2.2.4. *Sean* $\mathsf{r} \in \mathsf{rpos}(t)$ *y* ρ *una secuencia de reducción desde* t*. Se dice que* ρ usa r *si y solo si* ρ *reduce* r *o alguno de sus residuos. Luego,*

1. r *es* needed *en* t *si toda secuencia de reducción desde* t *a* β*-forma normal (full) usa* r*;*
2. r *es* head needed *en* t *si toda secuencia de reducción desde* t *a una* β*-forma normal head usa* r*;*
3. r *es* weak-head needed *en* t *si toda secuencia de reducción desde* t *a una* β*-forma normal weak-head usa* r*.*

Notar, en particular, que $\mathsf{nf}_\beta(t) = \emptyset$ ($\mathsf{hnf}_\beta(t) = \emptyset$ o $\mathsf{whnf}_\beta(t) = \emptyset$) implica que todo redex en t es needed (head needed o weak-head needed respectivamente).

Una paso de reducción $\to_\beta$ es *needed* (*head needed* o *weak-head needed*), notado $\to_{\mathtt{nd}}$ ($\to_{\mathtt{hnd}}$ o $\to_{\mathtt{whnd}}$ respectivamente), si el redex contraído es needed (head needed o weak-head needed respectivamente). Análogamente, una secuencia de reducción $\twoheadrightarrow_\beta$ es *needed* (*head needed* o *weak-head needed*), notada $\twoheadrightarrow_{\mathtt{nd}}$ ($\twoheadrightarrow_{\mathtt{hnd}}$ o $\twoheadrightarrow_{\mathtt{whnd}}$ respectivamente), si todo paso de reducción en la secuencia es needed (head needed o weak-head needed respectivamente).

Por ejemplo, considerar la secuencia de reducción:

$$(\lambda y.\lambda x.I\, x\, (\underline{I\, I}_{\mathsf{r}_1}))\, (I\, I) \to_{\mathtt{nd}} (\lambda y.\lambda x.\underline{I\, x}_{\mathsf{r}_2}\, I)\, (I\, I) \to_{\mathtt{nd}} \underline{(\lambda y.\lambda x.x\, I)\, (I\, I)}_{\mathsf{r}_3} \to_{\mathtt{nd}} \lambda x.x\, I$$

que es needed pero no head needed dado que el redex r_1 no necesariamente debe ser contraído para obtener una forma normal head:

$$(\lambda y.\lambda x.\underline{I\, x}_{\mathsf{r}_2}\, (I\, I))\, (I\, I) \to_{\mathtt{hnd}} \underline{(\lambda y.\lambda x.x\, (I\, I))\, (I\, I)}_{\mathsf{r}_3} \to_{\mathtt{hnd}} \lambda x.x\, (I\, I)$$

Más aún, esta segunda secuencia de reducción es head needed pero no weak-head needed dado que solo el redex r_3 es necesario para llegar a una forma normal weak-head:

$$\underline{(\lambda y.\lambda x.I\, x\, (I\, I))\, (I\, I)}_{\mathsf{r}_3} \to_{\mathtt{whnd}} \lambda x.I\, x\, (I\, I)$$

Notar que las siguientes igualdades se satisfacen: $\mathcal{NF}_{\mathtt{nd}} = \mathcal{NF}_\beta$, $\mathcal{NF}_{\mathtt{hnd}} = \mathcal{HNF}_\beta$ y $\mathcal{NF}_{\mathtt{whnd}} = \mathcal{WHNF}_\beta$.

Los leftmost redexes y las secuencias de reducción leftmost son de hecho needed:

Lema 2.2.5. *El leftmost redex de cualquier término no en forma normal (head o weak-head) es needed (head needed o weak-head needed respectivamente).*

Teorema 2.2.6. *Sean* $\mathsf{r} \in \mathsf{rpos}(t)$ *y* $\rho : t \twoheadrightarrow_\beta t'$ *la reducción leftmost (reducción head o reducción weak-head respectivamente) tal que* $t' = \mathsf{nf}_\beta(t)$ *(*$t' \in \mathsf{hnf}_\beta(t)$ *o* $t' \in \mathsf{whnf}_\beta(t)$ *respectivamente). Luego,* r *es needed (head needed o weak-head needed respectivamente) en* t *sii* ρ *usa* r*.*

Demostración. $\Rightarrow$) Inmediato por Def. 2.2.4.

$\Leftarrow$) Sea $\rho = \rho'\mathsf{r}'\rho''$ con $\mathsf{r}' \in \mathsf{r}/\rho'$. Por hipótesis r' es el leftmost redex de su correspondiente término. Por Lem. 2.2.5, r' es needed (head needed o weak-head needed respectivamente). Notar que, dado un redex s no needed en t, se sigue de la Def. 2.2.4 que ningún residuo de s es needed. Por lo tanto, r' needed (head needed o weak-head needed respectivamente) implica r needed (head needed o weak-head needed respectivamente) también. □

Notar que la reducción weak-head es un prefijo de la reducción head, que es a su vez un prefijo de la reducción leftmost a forma normal. Como consecuencia, es inmediato ver que todo redex weak-head needed es en particular head needed, y todo head needed es needed también. Por ejemplo, considerar:

$$\underline{(\lambda y.\lambda x.\overline{I\,x}^{\mathsf{r}_2}\,(\overline{I\,I}^{\mathsf{r}_1}))\,(\overline{I\,I}^{\mathsf{r}_4})}_{\mathsf{r}_3}$$

donde r_1 es needed pero no head needed ni weak-head needed. Sin embargo, r_2 es tanto needed como head needed, mientras que r_3 es el único redex weak-head needed del término, y r_4 no es necesario en ningún caso.

2.3. El sistema de tipos $\mathcal{N}$

En esta sección se presenta el sistema de tipos intersección (no idempotente) $\mathcal{N}$, introducido originalmente en [Kes16] y utilizado aquí para caracterizar los términos normalizantes respecto a la reducción weak-head needed. Más precisamente, se muestra que un término t es tipable en el sistema $\mathcal{N}$ si y solo si t es normalizante contrayendo únicamente redexes weak-head needed. Esta caracterización es usada en la Sec. 2.8 para concluir que la reducción weak-head needed es observacionalmente equivalente al λ-cálculo call-by-need (introducido en la Sec. 2.7).

Dado un tipo constante a para denotar *respuestas* (a*nswers*) y un conjunto infinito numerable de *variables de tipo* $\mathcal{V}$ $(\alpha, \beta, \gamma, \ldots)$, se define el conjunto de *tipos* $\mathcal{T}$ mediante la siguiente gramática:

$$\begin{array}{rrcl} \textbf{(Tipos)} & \tau, \sigma & ::= & \mathsf{a} \mid \alpha \in \mathcal{V} \mid \mathcal{M} \to \tau \\ \textbf{(Tipos multiset)} & \mathcal{M}, \mathcal{N} & ::= & \{\!\{\tau_i\}\!\}_{i \in I} \end{array}$$

donde I es un conjunto finito de índices.

El multi-conjunto vacío es denotado $\{\!\{\}\!\}$. Notar que los tipos son estrictos [vB92], *i.e.* el lado derecho de un tipo funcional nunca es un tipo multiset. Por lo tanto, la forma general de un tipo es $\mathcal{M}_1 \to \ldots \to \mathcal{M}_n \to \tau$ $(n \geq 0)$ con τ un tipo base (constante o variable).

Un *contexto de tipado* $(\Gamma, \Delta, \ldots)$ es una función de variables en tipos multiset que asigna el multi-conjunto vacío a todas las variables salvo a un conjunto finito de éstas. El *dominio* de Γ está dado por $\mathsf{dom}(\Gamma) \triangleq \{x \mid \Gamma(x) \neq \{\!\{\}\!\}\}$. La unión de contextos, notada $\Gamma + \Delta$, se define como $(\Gamma + \Delta)(x) \triangleq \Gamma(x) \sqcup \Delta(x)$, donde $\sqcup$ denota la unión de multi-conjuntos. A modo ilustrativo, considerar el ejemplo, $(x : \{\!\{\sigma\}\!\}; y : \{\!\{\tau\}\!\}) + (x : \{\!\{\sigma\}\!\}; z : \{\!\{\tau\}\!\}) = (x : \{\!\{\sigma, \sigma\}\!\}; y : \{\!\{\tau\}\!\}; z : \{\!\{\tau\}\!\})$. Esta noción es extendida a muchos contextos de la manera esperada, de modo que $+_{i \in I}\, \Gamma_i$ denota la unión finita de contextos (si $I = \emptyset$ esta notación es entendida como el contexto vacío). Se escribe $\Gamma \setminus\!\setminus x$ para el contexto tal que $(\Gamma \setminus\!\setminus x)(x) = \{\!\{\}\!\}$ y $(\Gamma \setminus\!\setminus x)(y) = \Gamma(y)$ si $y \neq x$.

Un *juicio de tipado* tiene la forma $\Gamma \vdash t : \tau$, donde Γ es un contexto de tipado, t un término y τ un tipo. Las reglas de tipado para el sistema $\mathcal{N}$ del λ-cálculo están dadas en la Fig. 2.1.

El tipo constante a en la regla (VAL) es usado para tipar valores. El axioma (AX) es relevante (no hay weakening) y la regla (APP) es multiplicativa, *i.e.* los distintos tipos asignados a una misma

$$\frac{}{x : \{\!\{\tau\}\!\} \vdash x : \tau}\ (\text{AX}) \qquad \frac{\Gamma \vdash t : \tau}{\Gamma \setminus\!\setminus x \vdash \lambda x.t : \Gamma(x) \to \tau}\ (\text{ABS})$$

$$\frac{}{\vdash \lambda x.t : \mathtt{a}}\ (\text{VAL}) \qquad \frac{\Gamma \vdash t : \{\!\{\sigma_i\}\!\}_{i\in I} \to \tau \quad (\Delta_i \vdash u : \sigma_i)_{i\in I}}{\Gamma +_{i\in I} \Delta_i \vdash t\, u : \tau}\ (\text{APP})$$

Figura 2.1: Reglas de tipado del sistema $\mathcal{N}$ para el λ-cálculo.

variable en los contextos Γ y $(\Delta_i)_{i\in I}$ se combinan en un único multi-conjunto en la conclusión, en contraposición a la regla aditiva del λ-cálculo simplemente tipado que apela a la unión compatible de conjuntos (*cf.* Sec. 1.2.2). Notar que el argumento de una aplicación es tipado $\#(I)$ veces por las premisas de la regla (APP). Un caso particular es cuando $I = \emptyset$: el sub-término u del término tipado $t\,u$ resulta no tipado.

El siguiente ejemplo ilustra el uso de los multi-conjuntos para llevar cuenta de la cantidad de veces que un argumento es utilizado por una función:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{x : \{\!\{\{\!\{\mathtt{a}\}\!\} \to \mathtt{a}\}\!\} \vdash x : \{\!\{\mathtt{a}\}\!\} \to \mathtt{a}}\ (\text{AX}) \quad \dfrac{}{x : \{\!\{\mathtt{a}\}\!\} \vdash x : \mathtt{a}}\ (\text{AX})}{x : \{\!\{\{\!\{\mathtt{a}\}\!\} \to \mathtt{a}, \mathtt{a}\}\!\} \vdash x\, x : \mathtt{a}}\ (\text{APP})}{\vdash \lambda x.x\, x : \{\!\{\{\!\{\mathtt{a}\}\!\} \to \mathtt{a}, \mathtt{a}\}\!\} \to \mathtt{a}}\ (\text{ABS}) \quad \dfrac{\dfrac{}{x : \{\!\{\mathtt{a}\}\!\} \vdash x : \mathtt{a}}\ (\text{AX})}{\vdash I : \{\!\{\mathtt{a}\}\!\} \to \mathtt{a}}\ (\text{ABS}) \quad \dfrac{}{\vdash I : \mathtt{a}}\ (\text{VAL})}{\vdash (\lambda x.x\, x)\, I : \mathtt{a}}\ (\text{APP})$$

Notar como se requiere tipar I dos veces para poder realizar la aplicación.

Se denota con $\pi \rhd_{\mathcal{N}} \Gamma \vdash t : \tau$ una *derivación de tipo* para el término t en $\mathcal{N}$, asignándole el nombre π. El *tamaño* de una derivación π, notado $\mathsf{sz}(\pi)$, se define como el número de nodos de su correspondiente árbol de derivación. Se escribe $\mathsf{rule}(\pi) \in \{(\text{AX}), (\text{VAL}), (\text{ABS}), (\text{APP})\}$ para acceder a la última regla aplicada en la derivación π. Del mismo modo, $\mathsf{prem}(\pi)$ es el multi-conjunto de sub-derivaciones propias maximales de π. Por ejemplo, dada

$$\pi = \frac{\pi_t \quad (\pi_u^i)_{i\in I}}{\Gamma \vdash t\, u : \tau}\ (\text{APP})$$

se tiene $\mathsf{rule}(\pi) = (\text{APP})$ y $\mathsf{prem}(\pi) = \{\!\{\pi_t\}\!\} \sqcup \{\!\{\pi_u^i \mid i \in I\}\!\}$. Adicionalmente, se usan las funciones $\mathsf{ctxt}(\pi)$, $\mathsf{subj}(\pi)$ y $\mathsf{type}(\pi)$ para acceder respectivamente al contexto, sujeto y tipo del juicio de tipado a la raíz del árbol de derivación π. A modo de abreviatura, se escribe $\pi(x)$ para denotar el tipo asociado a la variable x en el contexto de tipado de la conclusión de π (*i.e.* $\pi(x) \triangleq \mathsf{ctxt}(\pi)(x)$).

Los sistemas de tipos intersección pueden ser vistos como modelos [CD80], *i.e.* el tipado es estable por β-conversión: si t es tipable y $t =_\beta t'$, entonces t' también es tipable. Esta propiedad puede separarse en dos enunciados distintos, conocidos como *subject reduction* y *subject expansion* respectivamente. El primero garantiza la estabilidad del tipado por reducción, mientras que el segundo por expansión. En el caso particular de los tipos no idempotentes, subject reduction puede refinarse a una noción de *weighted subject reduction*, estableciendo que no solo el tipado es estable por reducción, sino que también el tamaño de la derivación de tipo decrece. Más aún, el decremento es estricto si la reducción se produce en ciertas posiciones de redex especiales, llamadas *posiciones tipadas*. A continuación se formalizan estos conceptos.

Definición 2.3.1. *El conjunto de* posiciones tipadas *de una derivación de tipo π, notado $\mathsf{tpos}(\pi)$, se define por inducción en $\mathsf{rule}(\pi)$:*

1. *Si* $\mathsf{rule}(\pi) \in \{(\text{AX}), (\text{VAL})\}$, *entonces se define* $\mathsf{tpos}(\pi) \triangleq \{\epsilon\}$.
2. *Si* $\mathsf{rule}(\pi) = (\text{ABS})$ *con* $\mathsf{subj}(\pi) = \lambda x.t$ *y* $\mathsf{prem}(\pi) = \{\!\{\pi_t\}\!\}$, *entonces se define* $\mathsf{tpos}(\pi) \triangleq \{\epsilon\} \cup \{\mathsf{0p} \mid \mathsf{p} \in \mathsf{tpos}(\pi_t)\}$.
3. *Si* $\mathsf{rule}(\pi) = (\text{APP})$ *con* $\mathsf{subj}(\pi) = t\,u$ *y* $\mathsf{prem}(\pi) = \{\!\{\pi_t\}\!\} \sqcup \{\!\{\pi_u^i \mid i \in I\}\!\}$, *entonces se define* $\mathsf{tpos}(\pi) \triangleq \{\epsilon\} \cup \{\mathsf{0p} \mid \mathsf{p} \in \mathsf{tpos}(\pi_t)\} \cup (\bigcup_{i\in I} \{\mathsf{1p} \mid \mathsf{p} \in \mathsf{tpos}(\pi_u^i)\})$.

En particular se tiene $\mathsf{tpos}(\pi_t) \subseteq \mathsf{pos}(t)$. Notar que hay dos tipos de posiciones no tipadas: dentro de argumentos no tipados y en el cuerpo no tipado de abstracciones. Por ejemplo, considerar las siguientes derivaciones:

$$\pi_K = \dfrac{\dfrac{\dfrac{}{x : \{\!\{\mathsf{a}\}\!\} \vdash x : \mathsf{a}}\,(\text{AX})}{x : \{\!\{\mathsf{a}\}\!\} \vdash \lambda y.x : \{\!\{\}\!\} \to \mathsf{a}}\,(\text{ABS})}{\vdash K : \{\!\{\mathsf{a}\}\!\} \to \{\!\{\}\!\} \to \mathsf{a}}\,(\text{ABS}) \qquad \pi_{KI} = \dfrac{\pi_K \qquad \dfrac{}{\vdash I : \mathsf{a}}\,(\text{VAL})}{\vdash K\,I : \{\!\{\}\!\} \to \mathsf{a}}\,(\text{APP}) \qquad \pi_{KI\Omega} = \dfrac{\pi_{KI} \qquad \pi_\Omega}{\vdash K\,I\,\Omega : \mathsf{a}}\,(\text{APP}) \tag{2.1}$$

donde $\mathsf{tpos}(\pi_{KI\Omega}) = \{\epsilon, \mathsf{0}, \mathsf{00}, \mathsf{01}, \mathsf{000}, \mathsf{0000}\} \subseteq \mathsf{pos}(K\,I\,\Omega)$.

Nota 2.3.2. *El weak-head redex de un término ocurre siempre en posición tipada.*

Alternativamente, se denotan derivaciones de tipo de forma más compacta con los siguientes predicados:

- $\mathsf{AX}(x, \tau)$ denota el resultado de aplicar (AX) con sujeto x y tipo τ:

$$\mathsf{AX}(x, \tau) \triangleq \dfrac{}{x : \{\!\{\tau\}\!\} \vdash x : \tau}\,(\text{AX})$$

- $\mathsf{VAL}(x, t)$ denota el resultado de aplicar (VAL) abstrayendo la variable x en el término t:

$$\mathsf{VAL}(x, t) \triangleq \dfrac{}{\vdash \lambda x.t : \mathsf{a}}\,(\text{VAL})$$

- $\mathsf{ABS}(x, \pi_t)$ denota el resultado de aplicar (ABS) con la premisa π_t, abstrayendo la variable x en el término t:

$$\mathsf{ABS}(x, \pi_t) \triangleq \dfrac{\pi_t}{\mathsf{ctxt}(\pi_t) \setminus\!\setminus x \vdash \lambda x.t : \pi_t(x) \to \mathsf{type}(\pi_t)}\,(\text{ABS})$$

- $\mathsf{APP}(\pi_t, u, (\pi_u^i)_{i\in I})$ denota el resultado de aplicar (APP) con las premisas π_t y $(\pi_u^i)_{i\in I}$ y el argumento u (u es no tipado cuando $I = \emptyset$):

$$\mathsf{APP}(\pi_t, u, (\pi_u^i)_{i\in I}) \triangleq \dfrac{\pi_t \qquad (\pi_u^i)_{i\in I}}{\mathsf{ctxt}(\pi_t) +_{i\in I} \mathsf{ctxt}(\pi_u^i) \vdash t\,u : \tau}\,(\text{APP})$$

  Notar que la aplicación es válida siempre que $\mathsf{type}(\pi_t) = \{\!\{\sigma_i\}\!\}_{i\in I} \to \tau$ y $(\mathsf{type}(\pi_u^i) = \sigma_i)_{i\in I}$.

En la Sec. 2.4 la noción de reducción es extendida de términos a derivaciones de tipos. Para esto es necesario poder acceder a las sub-derivaciones de una derivación y, eventualmente, reemplazarlas por otras adecuadas. Las siguientes definiciones se introducen con tal fin.

Definición 2.3.3. *El multi-conjunto de* sub-derivaciones *de* π *en posición* $\mathsf{p} \in \mathsf{tpos}(\pi)$, *notado* $\pi|_{\mathsf{p}}$, *se define inductivamente como:*

1. *Si* $\mathsf{p} = \epsilon$, *entonces se define* $\pi|_{\mathsf{p}} \triangleq \{\!\{\pi\}\!\}$.
2. *Si* $\mathsf{p} = 0\mathsf{p}'$, i.e. $\mathsf{rule}(\pi) \in \{(\text{ABS}), (\text{APP})\}$ *con* $\mathsf{subj}(\pi) \in \{\lambda x.t, t\,u\}$ *y* $\pi_t \in \mathsf{prem}(\pi)$, *entonces se define* $\pi|_{\mathsf{p}} \triangleq \pi_t|_{\mathsf{p}'}$.
3. *Si* $\mathsf{p} = 1\mathsf{p}'$, i.e. $\mathsf{rule}(\pi) = (\text{APP})$ *con* $\mathsf{subj}(\pi) = t\,u$ *y* $\mathsf{prem}(\pi) = \{\!\{\pi_t\}\!\} \sqcup \{\!\{\pi_u^i \mid i \in I\}\!\}$, *entonces se define* $\pi|_{\mathsf{p}} \triangleq \bigsqcup_{i\in I} \pi_u^i|_{\mathsf{p}'}$.

Definición 2.3.4. *El* reemplazo de sub-derivaciones *de* π *en posición* $\mathsf{p} \in \mathsf{tpos}(\pi)$ *por* $(\pi_s^i)_{i\in I}$, *notado* $\pi\langle(\pi_s^i)_{i\in I}\rangle_{\mathsf{p}}$, *se define inductivamente asumiendo* $\#(\pi|_{\mathsf{p}}) = \#(I)$:

1. *Si* $\mathsf{p} = \epsilon$, *entonces se define* $\pi\langle(\pi_s^i)_{i\in I}\rangle_{\mathsf{p}} \triangleq \pi_s^{i_0}$ *con* $I = \{i_0\}$.
2. *Si* $\mathsf{p} = 0\mathsf{p}'$, *luego:*
   a) *Si* $\mathsf{rule}(\pi) = (\text{ABS})$ *con* $\mathsf{subj}(\pi) = \lambda x.t$ *y* $\mathsf{prem}(\pi) = \{\!\{\pi_t\}\!\}$, *entonces se define* $\pi\langle(\pi_s^i)_{i\in I}\rangle_{\mathsf{p}} \triangleq \mathsf{ABS}(x, \pi_t\langle(\pi_s^i)_{i\in I}\rangle_{\mathsf{p}'})$.
   b) *Si* $\mathsf{rule}(\pi) = (\text{APP})$ *con* $\mathsf{subj}(\pi) = t\,u$ *y* $\mathsf{prem}(\pi) = \{\!\{\pi_t\}\!\} \sqcup \{\!\{\pi_u^j \mid j \in J\}\!\}$, *entonces se define* $\pi\langle(\pi_s^i)_{i\in I}\rangle_{\mathsf{p}} \triangleq \mathsf{APP}(\pi_t\langle(\pi_s^i)_{i\in I}\rangle_{\mathsf{p}'}, u, (\pi_u^j)_{j\in J})$.
3. *Si* $\mathsf{p} = 1\mathsf{p}'$, i.e. $\mathsf{rule}(\pi) = (\text{APP})$ *con* $\mathsf{subj}(\pi) = t\,u$ *y* $\mathsf{prem}(\pi) = \{\!\{\pi_t\}\!\} \sqcup \{\!\{\pi_u^j \mid j \in J\}\!\}$ *con* $I = \biguplus_{i\in J} I_j$, *entonces se define*

$$\pi\langle(\pi_s^i)_{i\in I}\rangle_{\mathsf{p}} \triangleq \mathsf{APP}(\pi_t, u\langle s\rangle_{\mathsf{p}'}, (\pi_u^j\langle(\pi_s^{i'})_{i'\in I_j}\rangle_{\mathsf{p}'})_{j\in J})$$

   donde s *es el sujeto de* π_s^i *para todo* $i \in I$, *y* $r\langle r'\rangle_{\mathsf{q}}$ *denota el reemplazo del sub-término* $r|_{\mathsf{q}}$ *por* r' *en* r *(la captura de variables está permitida). Notar que la descomposición de* I *en conjuntos* $(I_j)_{i\in J}$ *es no determinista, por lo que el reemplazo de sub-derivaciones resulta ser a su vez una operación no determinista.*

A modo ilustrativo, considerar la derivación $\pi_{KI\Omega}$ del ejemplo (2.1) y $\pi_{\lambda x.t} \rhd_{\mathcal{N}} \vdash \lambda x.t : \mathsf{a}$ obtenida a partir de (VAL). Luego,

$$\pi_{KI\Omega}\langle\pi_{\lambda x.t}\rangle_{01} = \dfrac{\dfrac{\pi_K \quad \pi_{\lambda x.t}}{\vdash K\,(\lambda x.t) : \{\!\{\}\!\} \to \mathsf{a}}\,(\text{APP})}{\vdash K\,(\lambda x.t)\,\Omega : \mathsf{a}}\,(\text{APP})$$

2.3.1. Propiedades del sistema

A continuación se presentan las propiedades principales del sistema $\mathcal{N}$: *weighted subject reduction* y *subject expansion*; las cuales dependen respectivamente de los lemas de sustitución y anti-sustitución. Estos resultados son presentados originalmente en [Kes16], donde se introduce el sistema de tipos $\mathcal{N}$ para estudiar la semántica de call-by-name, pero se reproducen a continuación para completitud de la presentación.

Lema 2.3.5 (Sustitución)**.** *Sean* $\pi_t \rhd_{\mathcal{N}} \Gamma; x : \{\!\{\sigma_i\}\!\}_{i\in I} \vdash t : \tau$ *y* $(\pi_u^i \rhd_{\mathcal{N}} \Delta_i \vdash u : \sigma_i)_{i\in I}$. *Luego, existe una derivación* $\pi_{\{x\setminus u\}t} \rhd_{\mathcal{N}} \Gamma +_{i\in I} \Delta_i \vdash \{x \setminus u\}t : \tau$ *tal que* $\mathsf{sz}(\pi_{\{x\setminus u\}t}) = \mathsf{sz}(\pi_t) +_{i\in I} \mathsf{sz}(\pi_u^i) - |I|$.

Teorema 2.3.6 (Weighted Subject Reduction)**.** *Sea* $\pi_t \rhd_{\mathcal{N}} \Gamma \vdash t : \tau$. *Si* $\mathsf{r} : t \to_\beta t'$, *entonces existe* $\pi_{t'} \rhd_{\mathcal{N}} \Gamma \vdash t' : \tau$. *Más aún,*

1. *Si* $\mathsf{r} \in \mathsf{tpos}(\pi_t)$, *entonces* $\mathsf{sz}(\pi_t) > \mathsf{sz}(\pi_{t'})$.

2. *Si* $\mathsf{r} \notin \mathsf{tpos}(\pi_t)$, *entonces* $\mathsf{sz}(\pi_t) = \mathsf{sz}(\pi_{t'})$.

Demostración. Por inducción en $\mathsf{r} \in \mathsf{tpos}(\pi_t)$ apelando al Lem. 2.3.5. Detalles en el Ap. A. □

Lema 2.3.7 (Anti-sustitución)**.** *Sea* $\pi_{\{x \setminus u\}t} \rhd_{\mathcal{N}} \Gamma \vdash \{x \setminus u\}t : \tau$. *Luego, existen derivaciones* $\pi_t \rhd_{\mathcal{N}} \Gamma'; x : \{\!\{\sigma_i\}\!\}_{i \in I} \vdash t : \tau$ *y* $(\pi_u^i \rhd_{\mathcal{N}} \Delta_i \vdash u : \sigma_i)_{i \in I}$ *tal que* $\Gamma = \Gamma' +_{i \in I} \Delta_i$.

Teorema 2.3.8 (Subject Expansion)**.** *Sea* $\pi_{t'} \rhd_{\mathcal{N}} \Gamma \vdash t' : \tau$. *Si* $t \to_\beta t'$, *entonces existe* $\pi_t \rhd_{\mathcal{N}} \Gamma \vdash t : \tau$.

Demostración. Por definición $t \to_\beta t'$ implica $t = \mathtt{C}\langle l \rangle$ y $t' = \mathtt{C}\langle r \rangle$ con $l \mapsto_\beta r$. La demostración es por inducción en $\mathtt{C}$ apelando al Lem. 2.3.7. Detalles en el Ap. A. □

2.4. Sustitución y reducción en derivaciones

Para poder relacionar adecuadamente las posiciones de redex entre términos β-convertibles, se extiende la noción de β-reducción a árboles de derivación de tipo, haciendo uso de la sustitución tipada. En contraposición con la sustitución y la β-reducción sobre términos, estas nuevas operaciones son no deterministas sobre árboles de derivación (una extensa discusión y ejemplos al respecto puede encontrarse en [Via17]).

Definición 2.4.1. *La* sustitución *de* x *por* $(\pi_s^i)_{i \in I}$ *en* π, *notada* $\{x \setminus (\pi_s^i)_{i \in I}\}\pi$ *por abuso de notación, es una derivación de tipo definida inductivamente sobre* π, *siempre que* $\pi(x) = \{\!\{\mathsf{type}(\pi_s^i)\}\!\}_{i \in I}$:

1. *Si* $\pi = \mathsf{AX}(y, \tau)$, *entonces*

$$\{x \setminus (\pi_s^i)_{i \in I}\}\pi \triangleq \begin{cases} \pi_s^{i_0} & \text{si } y = x \text{ con } I = \{i_0\} \\ \pi & \text{si } y \neq x \end{cases}$$

2. *Si* $\pi = \mathsf{VAL}(y, t)$, *entonces*

$$\{x \setminus (\pi_s^i)_{i \in I}\}\pi \triangleq \mathsf{VAL}(y, \{x \setminus s\}t)$$

si no hay captura de la variable y.

3. *Si* $\pi = \mathsf{ABS}(y, \pi_t)$, *entonces*

$$\{x \setminus (\pi_s^i)_{i \in I}\}\pi \triangleq \mathsf{ABS}(y, \{x \setminus (\pi_s^i)_{i \in I}\}\pi_t)$$

si no hay captura de la variable y.

4. *Si* $\pi = \mathsf{APP}(\pi_t, u, (\pi_u^j)_{j \in J})$ *con* $I = I' \uplus (\biguplus_{j \in J} I_j)$, *donde se tiene* $\pi_t(x) = \{\!\{\mathsf{type}(\pi_s^i)\}\!\}_{i \in I'}$ *y* $(\pi_u^j(x) = \{\!\{\mathsf{type}(\pi_s^i)\}\!\}_{i \in I_j})_{j \in J}$, *entonces*

$$\{x \setminus (\pi_s^i)_{i \in I}\}\pi \triangleq \mathsf{APP}(\{x \setminus (\pi_s^i)_{i \in I'}\}\pi_t, \{x \setminus u\}t, (\{x \setminus (\pi_s^{i'})_{i' \in I_j}\}\pi_u^j)_{j \in J})$$

Notar que la descomposición de I *en conjuntos* I' *e* $(I_j)_{i \in J}$ *es no determinista, por lo que la sustitución de árboles de derivación resulta ser a su vez una operación no determinista.*

Intuitivamente, la sustitución tipada reemplaza ocurrencias de x en posiciones tipadas por su correspondiente derivación π_u^i del mismo tipo, donde tal apareo es seleccionado de manera no determinista. Más aún, también sustituye todas las ocurrencias de x en posiciones no tipadas, donde esta operación

no tipada es completamente determinista. Por ejemplo, considerar la siguiente sustitución, donde $\pi_{K\Omega}$ es tomado del ejemplo (2.1) en la Sec. 2.3:

$$\{x \setminus \pi_{K\Omega}\}\left(\dfrac{\dfrac{}{x : \{\!\{\{\!\{\}\!\} \to \mathsf{a}\}\!\} \vdash x : \{\!\{\}\!\} \to \mathsf{a}}\,(\text{AX})}{x : \{\!\{\{\!\{\}\!\} \to \mathsf{a}\}\!\} \vdash x\,x : \mathsf{a}}\,(\text{APP})\right) = \dfrac{\pi_{K\Omega}}{\vdash (K\,\Omega)\,(K\,\Omega) : \mathsf{a}}\,(\text{APP})$$

donde la ocurrencia a derecha de x está en posición no tipada, al igual que la ocurrencia a derecha de $K\,\Omega$ en el resultado.

El siguiente lema relaciona las posiciones tipadas de los árboles que componen la sustitución con las posiciones tipadas del árbol sustituido:

Lema 2.4.2. *Sean π_t y $(\pi^i_u)_{i\in I}$ derivaciones tal que $\{x \setminus (\pi^i_u)_{i\in I}\}\pi_t$ está definida, y $\mathsf{p} \in \mathsf{pos}(t)$. Entonces,*

1. $\mathsf{p} \in \mathsf{tpos}(\pi_t)$ *sii* $\mathsf{p} \in \mathsf{tpos}(\{x \setminus (\pi^i_u)_{i\in I}\}\pi_t)$.
2. $\mathsf{q} \in \mathsf{tpos}(\pi^k_u)$ *para algún* $k \in I$ *sii existe* $\mathsf{p}' \in \mathsf{tpos}(\pi_t)$ *tal que* $t|_{\mathsf{p}'} = x$ *y* $\mathsf{p}'\mathsf{q} \in \mathsf{tpos}(\{x \setminus (\pi^i_u)_{i\in I}\}\pi_t)$.

Basado en esta noción de sustitución sobre árboles de derivación, se introduce ahora la reducción (no determinista) sobre los mismos. La *relación de reducción* $\to_\beta$ sobre árboles de derivación se define considerando en primer lugar las siguientes reglas de reescritura:

1. Para β-redexes tipados:

$$\dfrac{\dfrac{\pi_t \rhd_{\mathcal{N}} \Gamma; x : \{\!\{\sigma_i\}\!\}_{i\in I} \vdash t : \tau}{\Gamma \vdash \lambda x.t : \{\!\{\sigma_i\}\!\}_{i\in I} \to \tau} \qquad (\pi^i_u \rhd_{\mathcal{N}} \Delta_i \vdash u : \sigma_i)_{i\in I}}{\Gamma +_{i\in I} \Delta_i \vdash (\lambda x.t)\,u : \tau} \;\mapsto_\beta\; \{x \setminus (\pi^i_u)_{i\in I}\}\pi_t$$

2. Para β-redexes en posiciones no tipadas, con $u \to_\beta u'$:

$$\dfrac{\Gamma \vdash t : \{\!\{\}\!\} \to \tau}{\Gamma \vdash t\,u : \tau} \;\mapsto_\nu\; \dfrac{\Gamma \vdash t : \{\!\{\}\!\} \to \tau}{\Gamma \vdash t\,u' : \tau} \qquad \dfrac{}{\vdash \lambda x.u : \mathsf{a}} \;\mapsto_\xi\; \dfrac{}{\vdash \lambda x.u' : \mathsf{a}}$$

Como en el caso de la β-reducción estándar del λ-cálculo, donde las reglas son clausuradas por *contextos de términos*, es necesario clausurar la relación presentada anteriormente por *contextos de árboles de derivación*. Sin embargo, un paso de reducción en un sub-término dado (en posición p) puede ocasionar varios pasos *simultáneos* en el correspondiente árbol de derivación (recordar que $\pi|_{\mathsf{p}}$ se define como un multi-conjunto). Informalmente, dado un redex r de t, una derivación de tipo π para t y un multi-conjunto de sub-derivaciones minimales de π que contienen r, notado $\mathscr{M}$, se aplican las reglas de reescritura $\mapsto_{\beta,\nu,\xi}$ a todos los elementos de $\mathscr{M}$, obteniendo así un multi-conjunto $\mathscr{M}'$, y se reconstruye con él la derivación de tipo del reducto de t.

Para formalizar esta idea, se introduce el concepto de *prefijo tipado maximal*:

Definición 2.4.3. *El* prefijo tipado maximal *de una posición $\mathsf{p} \in \mathsf{pos}(\mathsf{subj}(\pi))$ en una derivación de tipo π, notado $\mathsf{mtp}_\pi(\mathsf{p})$ es el único prefijo de p que satisface:*

$$\mathsf{mtp}_\pi(\mathsf{p}) = \mathsf{q} \overset{\triangle}{\iff} \mathsf{q} \in \mathsf{tpos}(\pi) \wedge \forall \mathsf{q}' \in \mathsf{tpos}(\pi).\mathsf{q}' \leq \mathsf{p} \implies \mathsf{q}' \leq \mathsf{q}$$

Notar que el multi-conjunto de sub-derivaciones de π en posición $\mathsf{mtp}_\pi(\mathsf{p})$ (*i.e.* $\pi|_{\mathsf{mtp}_\pi(\mathsf{p})}$) se corresponde con el multi-conjunto de sub-derivaciones minimales de π que contienen a p. Por ejemplo, considerar la derivación de tipo $\pi_{KI\Omega}$ del ejemplo (2.1) en la Sec. 2.3:

$$\pi_{KI} = \dfrac{\pi_K \qquad \dfrac{}{\vdash I : \mathsf{a}}\,(\textsc{val})}{\vdash K\,I : \{\!\{\} \to \mathsf{a}}\,(\textsc{app})$$
$$\pi_{KI\Omega} = \dfrac{\vdash K\,I : \{\!\{\} \to \mathsf{a} \qquad \vdash \Omega : \mathsf{a}}{\vdash K\,I\,\Omega : \mathsf{a}}\,(\textsc{app})$$

donde $\mathsf{tpos}(\pi_{KI\Omega}) = \{\epsilon, \mathsf{0}, \mathsf{00}, \mathsf{01}, \mathsf{000}, \mathsf{0000}\}$. Se tiene $\mathsf{mtp}_{\pi_{KI\Omega}}(\mathsf{010}) = \mathsf{01}$ y la sub-derivación minimal de $\pi_{KI\Omega}$ que contiene la posición $\mathsf{010}$ es, de hecho, $\pi_{KI\Omega}|_{\mathsf{01}} = \{\!\{\pi_I\}\!\}$. Además, $\mathsf{mtp}_{\pi_{KI\Omega}}(\mathsf{1p}) = \epsilon$ para toda $\mathsf{1p} \in \mathsf{pos}(K\,I\,\Omega)$, y $\pi_{KI\Omega}|_\epsilon = \{\!\{\pi_{KI\Omega}\}\!\}$.

Finalmente, se dice que π *reduce en un paso* a π', notado $\pi \to_\beta \pi'$, si y solo si existe $\mathsf{r} \in \mathsf{rpos}(\mathsf{subj}(\pi))$ tal que

$$\pi|_{\mathsf{mtp}_\pi(\mathsf{r})} \rightsquigarrow_{\beta,\nu,\xi} \mathscr{M} \quad \text{y} \quad \pi' = \pi\langle\mathscr{M}\rangle_\mathsf{r}$$

donde $\rightsquigarrow_{\beta,\nu,\xi}$ es el *lifting* a multi-conjunto de las reglas de reescritura $\mapsto_{\beta,\nu,\xi}$ presentadas anteriormente, aplicando la misma regla a todos los elementos del multi-conjunto.

Esto define la *relación de reducción* $\to_\beta$ sobre árboles de derivación. Una secuencia de reducción sobre árboles de derivación que contrae únicamente redexes en posición tipada se denomina *secuencia de reducción tipada*.

Notar que las reducciones tipadas son normalizantes por Teo. 2.3.6, resultando en una clase especial de derivaciones, a saber: aquellas que no contiene redexes tipados. Dada una derivación $\pi \rhd_\mathcal{N} \Gamma \vdash t : \tau$, se dice que π es *normal* si y solo si $\mathsf{tpos}(\pi) \cap \mathsf{rpos}(t) = \emptyset$. De hecho, la reducción sobre árboles induce una reducción sobre términos, pues siempre que $\rho : \pi \twoheadrightarrow_\beta \pi'$ se tiene $\mathsf{subj}(\pi) \twoheadrightarrow_\beta \mathsf{subj}(\pi')$. Por abuso de notación, se denotaran ambas secuencias con el mismo nombre ρ.

2.5. Redexes tipados vs. weak-head needed

En esta sección se presenta uno de los resultados principales de este estudio. Se establece aquí una conexión entre los redexes weak-head needed y aquellos en posición tipada. Más precisamente, se muestra en la Sec. 2.5.1 que todo redex weak-head needed ocurre en una posición tipada, para cualquier derivación de tipo del término en cuestión. Sin embargo, el recíproco no necesariamente es cierto. Para ello, se muestra en la Sec. 2.5.2 que todo redex en posición tipada para una clase especial de derivaciones (llamadas principales) es de hecho un redex weak-head needed. Primero un resultado técnico:

Lema 2.5.1. *Sean* $\mathsf{r} : \pi_t \to_\beta \pi_{t'}$ *y* $\mathsf{p} \in \mathsf{pos}(t)$ *tal que* $\mathsf{p} \neq \mathsf{r}$ *y* $\mathsf{p} \neq \mathsf{0r}$. *Entonces,* $\mathsf{p} \in \mathsf{tpos}(\pi_t)$ *sii existe* $\mathsf{p'} \in \mathsf{p/r}$ *tal que* $\mathsf{p'} \in \mathsf{tpos}(\pi_{t'})$.

2.5.1. Redexes weak-head needed son tipados

Para mostrar que todo redex weak-head needed ocurre en posición tipada, cabe destacar primero que una posición tipada no puede provenir de una no tipada:

Lema 2.5.2. *Sean* $\rho : \pi_t \twoheadrightarrow_\beta \pi_{t'}$ *y* $\mathsf{p} \in \mathsf{pos}(t)$. *Si existe* $\mathsf{p'} \in \mathsf{p}/\rho$ *tal que* $\mathsf{p'} \in \mathsf{tpos}(\pi_{t'})$, *entonces* $\mathsf{p} \in \mathsf{tpos}(\pi_t)$.

Luego, el resultado buscado se formaliza de la siguiente manera:

Teorema 2.5.3. *Sea* r *un weak-head redex de* t *y* π_t *una derivación de tipo para* t. *Luego,* $\mathsf{r} \in \mathsf{tpos}(\pi_t)$.

Demostración. Por Teo. 2.2.6, r es usado en la reducción weak-head desde t a $t' \in \mathcal{WHNF}_\beta$. Por Nota 2.3.2, la reducción weak-head contrae unicamente redexes tipados. Por lo tanto, r o alguno de sus residuos está en posición tipada en su correspondiente árbol de derivación. Finalmente, se concluye por Lem 2.5.2. □

2.5.2. Redexes tipados principalmente son weak-head needed

Como se mencionó anteriormente, el recíproco del Teo. 2.5.3 no es necesariamente cierto: dada una derivación de tipo arbitraria, puede haber redexes en posiciones tipadas que no se correspondan con redexes weak-head needed. Esto es ilustrado en el siguiente ejemplo (recordar $\pi_{KI\Omega}$ del ejemplo (2.1) en la Sec. 2.3):

$$\dfrac{\dfrac{\pi_{KI\Omega}}{\vdash \lambda x.K\,I\,\Omega : \{\!\{\}\!\} \to \mathsf{a}}\,(\text{ABS})}{\vdash \lambda x.K\,I\,\Omega : \{\!\{\{\!\{\}\!\} \to \mathsf{a}\}\!\} \to \mathsf{a}}\,(\text{ABS}) \qquad \dfrac{\dfrac{}{x : \{\!\{\{\!\{\}\!\} \to \mathsf{a}\}\!\} \vdash x : \{\!\{\}\!\} \to \mathsf{a}}\,(\text{AX}) \quad \pi_{KI\Omega}}{x : \{\!\{\{\!\{\}\!\} \to \mathsf{a}\}\!\} \vdash x\,(K\,I\,\Omega) : \mathsf{a}}\,(\text{ABS})$$

donde las posiciones $\mathsf{0}$ en el término $\lambda x.K\,I\,\Omega$ y $\mathsf{1}$ en $x\,(K\,I\,\Omega)$ corresponden a redexes tipados que no son weak-head needed, pues ambos términos ya se encuentran en forma normal weak-head. Afortunadamente, el tipado relaciona este tipo de redexes si se restringen las derivaciones a las llamadas *principales*. En este marco, se entiende por tipado principal a la derivación de menor tamaño posible para un término dado. Se define la noción a partir de las formas normales weak-head y se generaliza a término normalizantes mediante subject expansion, apelando al Teo. 2.3.8.

Definición 2.5.4. *Sea $s \in \mathcal{WHNF}_\beta$. La derivación $\pi_s \rhd_{\mathcal{N}} \Gamma \vdash s : \tau$ se dice* principal normal *sii:*

1. *$s = x\,s_1 \dots s_n$ ($n \geq 0$), $\Gamma = \{x : \{\!\{\overbrace{\{\!\{\}\!\} \to \dots \to \{\!\{\}\!\}}^{n \text{ veces}} \to \tau\}\!\}\}$ y τ es una variable de tipo α (*i.e. *ninguno de los s_i está tipado); o*
2. *$s = \lambda x.s'$, $\Gamma = \emptyset$ y $\tau = \mathsf{a}$.*

Sea t un término weak-head normalizante tal que $\pi_t \rhd_{\mathcal{N}} \Gamma \vdash t : \tau$. π_t se dice principal *sii $\pi_t \twoheadrightarrow_\beta \pi'_t$ con $t' \in \mathcal{WHNF}_\beta$ y $\pi_{t'}$ principal normal.*

Notar, en particular, que la definición anterior no depende de la forma normal weak-head t' elegida: suponer que $t'' \in \mathsf{whnf}_\beta(t)$ y $t'' \neq t'$, por confluencia del λ-cálculo [Bar85] $t' =_\beta t''$, por lo que t' es normal principalmente tipable si y solo si t'' también lo es, por Teo. 2.3.6 y 2.3.8.

Lema 2.5.5. *Sean π_t una derivación con sujeto t, $\mathsf{r} \in \mathsf{rpos}(t) \cap \mathsf{tpos}(\pi_t)$ y $\rho : \pi_t \twoheadrightarrow_\beta \pi_{t'}$. Si $\pi_{t'}$ es principal normal, entonces ρ usa r.*

Las nociones de reducción leftmost y weak-head needed sobre términos (no tipados) se extienden de manera natural a reducciones tipadas sobre árboles de derivación. Por lo tanto, se tiene el siguiente resultado:

Lema 2.5.6. *Sean t un término weak-head normalizante y π_t un tipado principal. Luego, toda secuencia de reducción tipada leftmost desde π_t es weak-head needed.*

Teorema 2.5.7. *Sean t un término weak-head normalizante, π_t un tipado principal y $\mathsf{r} \in \mathsf{rpos}(t) \cap \mathsf{tpos}(\pi_t)$. Luego, r es un redex weak-head needed en t.*

Demostración. Sea $\rho : \pi_t \twoheadrightarrow_\beta \pi_{t'}$ la secuencia de reducción tipada leftmost con $\pi_{t'}$ normal. Notar que $\pi_{t'}$ existe por Def. 2.5.4. Por Lem. 2.5.6, ρ es una secuencia de reducción weak-head needed. Más aún, por Lem. 2.5.5, ρ usa r. Por lo tanto, r es weak-head needed en t. □

Una consecuencia inmediata de los Teo. 2.5.3 y 2.5.7 es que, dado un término weak-head normalizante t, los redexes en posición tipada en la derivación de tipo principal de t (que siempre existe) se corresponden con los redexes weak-head needed de t. Por lo tanto, el sistema $\mathcal{N}$ permite identificar todos los redexes weak-head needed de un término weak-head normalizante.

2.6. Normalización weak-head needed

Una de las contribuciones principales del presente capítulo es la equivalencia observacional entre call-by-need y la reducción weak-head needed. En esta sección se introduce una pieza clave para tal contribución: caracterización mediante el sistema $\mathcal{N}$ de los términos weak-head normalizantes. De hecho, se relaciona el tipado con la noción de weak-head neededness mostrando que todo término es tipable en el sistema $\mathcal{N}$ si y solo si es normalizante para la reducción weak-head needed. Esta caracterización pone en evidencia el poder expresivo de los tipos intersección.

Lema 2.6.1. *Sea $\pi_t \rhd_{\mathcal{N}} \Gamma \vdash t : \tau$. Luego, π_t normal implica $t \in \mathcal{WHNF}_\beta$.*

Sea $\rho : t_1 \twoheadrightarrow_\beta t_n$, se dice que ρ es una secuencia de reducción *left-to-right* si y solo si para todo $i < n$, si $\mathsf{r}_i : t_i \to_\beta t_{i+1}$ y l_i está a la izquierda de r_i, entonces para todo $j > i$ tal que $\mathsf{r}_j : t_j \to_\beta t_{j+1}$ se tiene $\mathsf{r}_j \notin \mathsf{l}_i/\rho_{ij}$, donde $\rho_{ij} : t_i \twoheadrightarrow_\beta t_j$ es la correspondiente sub-secuencia de reducción de ρ. En otras palabras, para todo j y todo $i < j$, r_j no es un residuo de un redex a la izquierda de r_i (relativo a la secuencia de reducción de t_i a t_j dada) [Bar85].

Las reducciones left-to-right definen, en particular, estrategias estándar de reducción, las cuales establecen una forma canónica de construir secuencias de reducción entre términos:

Teorema 2.6.2 ([Bar85])**.** *Sea $t \in \mathbb{T}_\lambda$. Si $t \twoheadrightarrow_\beta t'$, entonces existe una reducción left-to-right de t a t'.*

Teorema 2.6.3. *Sea $t \in \mathbb{T}_\lambda$. Luego, $\pi_t \rhd_{\mathcal{N}} \Gamma \vdash t : \tau$ sii $t \in \mathcal{WN}_{\mathtt{whnd}}$.*

Demostración. $\Rightarrow$) Por Teo. 2.3.6 se tiene que la estrategia que contrae únicamente redexes tipados es normalizante, *i.e.* existe t' tal que $t \twoheadrightarrow_\beta t'$ y $\pi_{t'} \rhd_{\mathcal{N}} \Gamma \vdash t' : \tau$ es normal. Luego, por Lem. 2.6.1, $t' \in \mathcal{WHNF}_\beta$. Por Teo. 2.6.2, existe una reducción left-to-right $\rho : t \twoheadrightarrow_\beta t'$. Sea $\rho : t = t_1 \twoheadrightarrow_\beta t_n \twoheadrightarrow_\beta t'$ con $n \geq 0$ tal que $t_1, \dots t_{n-1} \notin \mathcal{WHNF}_\beta$ y $t_n \in \mathcal{WHNF}_\beta$.

Se muestra a continuación que todos los pasos de reducción en $t_1 \twoheadrightarrow_\beta t_n$ son leftmost. Asumir en pos de una contradicción que existe $k < n$ tal que $\mathsf{r} : t_k \to_\beta t_{k+1}$ y r no es el leftmost redex de t_k (notado l_k). Como ρ es una reducción left-to-right, ningún residuo de l_k es contraído luego del k-ésimo paso. Por lo tanto, se tiene una secuencia de reducción desde $t_k \notin \mathcal{WHNF}_\beta$ a $t_n \in \mathcal{WHNF}_\beta$ tal que l_k no es usado. Esto lleva a una contradicción pues l_k es weak-head needed por Lem. 2.2.5.

Como consecuencia, todos los pasos de reducción en $t_1 \twoheadrightarrow_\beta t_n$ son leftmost. Más aún, por Lem. 2.2.5, $t \twoheadrightarrow_\beta t_n \in \mathcal{WHNF}_\beta = \mathcal{NF}_{\mathtt{whnd}}$. Por lo tanto, $t \in \mathcal{WN}_{\mathtt{whnd}}$.

$\Leftarrow$) Considerar la secuencia de reducción $\rho : t \twoheadrightarrow_{\mathtt{whnd}} t'$ con $t' \in \mathsf{whnf}_\beta(t)$. Sea $\pi_{t'} \rhd_{\mathcal{N}} \Gamma \vdash t' : \tau$ una derivación de tipo principal normal para t' construida como en la Def. 2.5.4. Finalmente, se concluye por inducción en ρ aplicando Teo. 2.3.8, $\pi_t \rhd_{\mathcal{N}} \Gamma \vdash t : \tau$. □

2.7. El λ-cálculo call-by-need

Esta sección describe la sintaxis y semántica operacional del λ-cálculo call-by-need introducido en [ABM14]. Éste se introduce como una variante del λ-cálculo con *sustituciones explícitas* (λ_{ES}), y es más conciso que otras presentaciones anteriores de call-by-need [AFM$^+$95, AF97, MOW98, CF12], aunque es equivalente a éstas desde el punto de vista operacional [BBBK17] y resulta más conveniente para presentar los resultados de este estudio.

Dado un conjunto infinito numerable de *variables* $\mathbb{V}$ $(x, y, \ldots)$, los conjuntos de *términos* $\mathbb{T}_{\lambda_{\text{ES}}}$, *valores*, *respuestas* y *contextos* se definen mediante la siguiente gramática:

$$\begin{array}{rrcl} \textbf{Términos} & t & ::= & x \in \mathbb{V} \mid t\,t \mid \lambda x.t \mid t[x \setminus t] \\ \textbf{Valores} & v & ::= & \lambda x.t \\ \textbf{Contextos lista} & \mathtt{L} & ::= & \Box \mid \mathtt{L}[x \setminus t] \\ \textbf{Respuestas} & a & ::= & \mathtt{L}\langle \lambda x.t \rangle \\ \textbf{Contextos need} & \mathtt{N} & ::= & \Box \mid \mathtt{N}\,t \mid \mathtt{N}[x \setminus t] \mid \mathtt{N}\langle\!\langle x \rangle\!\rangle [x \setminus \mathtt{N}] \end{array}$$

Los términos de la forma $t[x \setminus u]$ se llaman *clausuras*, y $[x \setminus u]$ se denomina una *sustitución explícita* (ES). El sub-conjunto de términos en $\mathbb{T}_{\lambda_{\text{ES}}}$ sin ES es exactamente $\mathbb{T}_{\lambda}$ (el conjunto de términos del λ-cálculo). Las nociones de variables *libres* y *ligadas* se definen de la manera esperada, extendiendo la definición dada para el λ-cálculo, para contemplar el operador de sustitución explícita en particular $\mathsf{fv}(t[x \setminus u]) \triangleq (\mathsf{fv}(t) \setminus \{x\}) \cup \mathsf{fv}(u)$ y $\mathsf{bv}(t[x \setminus u]) \triangleq \mathsf{bv}(t) \cup \{x\} \cup \mathsf{bv}(u)$. También se extiende la noción estándar de α-conversión a ES.

Se usa la notación especial $\mathtt{N}\langle\!\langle u \rangle\!\rangle$ o $\mathtt{L}\langle\!\langle u \rangle\!\rangle$ cuando las variables libres de u no son capturadas por el contexto, *i.e.* no hay ninguna abstracción o sustitución explícita en el contexto que ligue variables libres de u. Por ejemplo, dado $\mathtt{N} = (\Box\, x)[x \setminus z]$, se tiene $\mathtt{N}\langle y \rangle = (y\, x)[x \setminus z] = \mathtt{N}\langle\!\langle y \rangle\!\rangle$, pero $\mathtt{N}\langle x \rangle = (x\, x)[x \setminus z]$ no puede ser denotado $\mathtt{N}\langle\!\langle x \rangle\!\rangle$. Notar el uso de esta notación especial en el último caso de contexto need. Un ejemplo del tal contexto es $(x\, y)[y \setminus z][x \setminus \Box]$.

El *λ-cálculo call-by-need*, introducido en [ABM14], está dado por el conjunto de términos $\mathbb{T}_{\lambda_{\text{ES}}}$ y la *relación de reducción* $\to_{\mathtt{need}}$, definida como la unión de $\to_{\mathtt{dB}}$ y $\to_{\mathtt{lsv}}$, *i.e.* las clausuras por *contextos need* de las respectivas reglas de reescritura:

$$\begin{array}{rcl} \mathtt{L}\langle \lambda x.t \rangle\, u & \mapsto_{\mathtt{dB}} & \mathtt{L}\langle t[x \setminus u] \rangle \\ \mathtt{N}\langle\!\langle x \rangle\!\rangle [x \setminus \mathtt{L}\langle v \rangle] & \mapsto_{\mathtt{lsv}} & \mathtt{L}\langle \mathtt{N}\langle\!\langle v \rangle\!\rangle [x \setminus v] \rangle \end{array}$$

Estas reglas evitan la captura de variables. Un ejemplo de secuencia de reducción `need` es el siguiente, donde se resalta el redex a cada paso para mayor claridad:

$$\begin{array}{lll} \underline{(\lambda x_0.I\,(x_0\,I))\,(\lambda y.I\,y)} & \to_{\mathtt{dB}} & \underline{(I\,(x_0\,I))[x_0 \setminus \lambda y.I\,y]} \to_{\mathtt{dB}} \\ \underline{x_1[x_1 \setminus x_0\,I][x_0 \setminus \lambda y.I\,y]} & \to_{\mathtt{lsv}} & x_1[x_1 \setminus \underline{(\lambda x_2.I\,x_2)\,I}][x_0 \setminus \lambda y.I\,y] \to_{\mathtt{dB}} \\ x_1[x_1 \setminus \underline{(I\,x_2)}[x_2 \setminus I]][x_0 \setminus \lambda y.I\,y] & \to_{\mathtt{dB}} & x_1[x_1 \setminus \underline{x_3[x_3 \setminus x_2][x_2 \setminus I]}][x_0 \setminus \lambda y.I\,y] \to_{\mathtt{lsv}} \\ x_1[x_1 \setminus x_3[x_3 \setminus \underline{I}][x_2 \setminus I]][x_0 \setminus \lambda y.I\,y] & \to_{\mathtt{lsv}} & \underline{x_1[x_1 \setminus I[x_3 \setminus I][x_2 \setminus I]]}[x_0 \setminus \lambda y.I\,y] \to_{\mathtt{lsv}} \\ I[x_1 \setminus I][x_3 \setminus I][x_2 \setminus I][x_0 \setminus \lambda y.I\,y] & & \end{array}$$

Al igual que para call-by-name, la reducción preserva las variables libres, *i.e.* $t \to_{\mathtt{need}} t'$ implica $\mathsf{fv}(t) \subseteq \mathsf{fv}(t')$. Notar que la reducción call-by-need también es weak, de modo que las respuestas (*answers*) no son `need`-reducibles.

2.8. Equivalencia observacional

Los resultados en la Sec. 2.6 son usados aquí para probar la corrección y completitud de call-by-need respecto a la reducción weak-head needed. Más precisamente, la estrategia de reducción call-by-need termina de evaluar en un valor si y solo si la reducción weak-head needed termina en un valor. Eso significa que call-by-need y weak-head needed son observacionalmente equivalentes.

Formalmente, dada una reducción $\mathcal{R}$ en un lenguaje de términos $\mathbb{T}$, y una noción asociada de contexto para $\mathbb{T}$, se dice que t es *observacionalmente equivalente* a u, notado $t \cong_{\mathcal{R}} u$, si y solo si $\mathtt{C}\langle t \rangle \in \mathcal{WN}_{\mathcal{R}} \iff \mathtt{C}\langle u \rangle \in \mathcal{WN}_{\mathcal{R}}$ para todo contexto $\mathtt{C}$.

Para mostrar el resultado final de este estudio, se apela al siguiente teorema:

Teorema 2.8.1 ([Kes16])**.**

1. *Sea* $t \in \mathbb{T}_\lambda$*. Luego,* $\pi \rhd_{\mathcal{N}} \Gamma \vdash t : \tau$ *sii* $t \in \mathcal{WN}_{\mathtt{name}}$*.*
2. *Sean* $t, u \in \mathbb{T}_\lambda$*. Luego,* $t \cong_{\mathtt{name}} u$ *sii* $t \cong_{\mathtt{need}} u$*.*

Teorema 2.8.2. *Sean* $t, u \in \mathbb{T}_\lambda$*. Luego,* $t \cong_{\mathtt{whnd}} u$ *sii* $t \cong_{\mathtt{need}} u$*.*

Demostración. Por Teo. 2.8.1 (2) basta probar $t \cong_{\mathtt{whnd}} u$ sii $t \cong_{\mathtt{name}} u$. Luego,

$$
\begin{array}{rcll}
t \cong_{\mathtt{name}} u & & \text{sii} & \text{(definición)} \\
\mathtt{C}\langle t\rangle \in \mathcal{WN}_{\mathtt{name}} & \iff & \mathtt{C}\langle u\rangle \in \mathcal{WN}_{\mathtt{name}} & \text{sii} \quad \text{(Teo. 2.8.1 (1))} \\
\mathtt{C}\langle t\rangle \ \mathcal{N}\text{-tipable} & \iff & \mathtt{C}\langle u\rangle \ \mathcal{N}\text{-tipable} & \text{sii} \quad \text{(Teo. 2.6.3)} \\
\mathtt{C}\langle t\rangle \in \mathcal{WN}_{\mathtt{whnd}} & \iff & \mathtt{C}\langle u\rangle \in \mathcal{WN}_{\mathtt{whnd}} & \text{sii} \quad \text{(definición)} \\
t \cong_{\mathtt{whnd}} u & & &
\end{array}
$$

□

2.9. Conclusión

El estudio realizado en el presente capítulo permitió establecer una conexión entre la noción semántica de reducción necesaria y el concepto sintáctico de estrategia call-by-need, al demostrar la equivalencia observacional entre ambas. Más aún, se prueba mediante el uso de tipos intersección no idempotente –una técnica poderosa capaz de caracterizar diferentes propiedades operacionales– que la estrategia call-by-need reduce únicamente redexes needed.

Los tipos intersección no idempotente proveen una herramienta simple y natural para mostrar la equivalencia observacional entre estas dos nociones. Otra técnica de demostración aplicable a este fin (que no involucran tipos intersección) puede encontrarse en [Bal12]. Sin embargo, ésta fue ideada para probar que nociones sintácticas de evaluación lazy resultan optimales [Lév75], por lo que es considerablemente más compleja que la técnica aquí presentada, puesto que involucra la definición de un *sistema de reducción combinatorio* (CRS) [Klo80, KvOvR93] y la adaptación de criterios de reducción optimal para sistemas de reescritura ortogonal de primer orden [Kha93] en presencia de sharing [Lév75] al CRS propuesto. Esto resulta en un sistema de reescritura de orden superior [BKvO03]. Por otro lado, la técnica utilizada en [Bal12] se sabe no aplicable al λ-cálculo en su generalidad [Gue96].

Una extensión interesante (y a la vez simple) del resultado presentado en la Sec. 2.5 es ver si la reducción call-by-need (definida sobre términos con sustituciones explícitas) contrae únicamente `dB`-redexes weak-head needed, para una noción apropiada de weak-head needed sobre términos con sustituciones explícitas. Una herramienta técnica para obtener tal resultado sería el sistema $\mathcal{A}$ [Kes16], una adaptación directa del sistema $\mathcal{N}$ a la sintaxis del λ-cálculo call-by-need.

Una posible línea de trabajo futuro, relacionada a los resultados obtenidos, surge a partir de la reciente formulación de *strong call-by-need* [BBBK17], que describe una estrategia call-by-need determinista a forma normal (full). Resulta natural buscar extender la técnica aquí presentada para obtener un resultado de equivalencia observacional entre la noción estándar de reducción necesaria [BKKS87] y la estrategia strong call-by-need.

Capítulo 3

Calculus of Applicative Patterns

El presente capítulo se centra en el estudio de propiedades sobre cálculos con patrones. Principalmente en *path polymorphism*, el cual involucra, entre otros aspectos, el uso de patrones estáticos y funciones recursivas. El objetivo del estudio aquí realizado es desarrollar un sistema de tipos estático para un cálculo adecuado capaz de capturar esta característica, como puntapié inicial al desarrollo de un prototipo de lenguaje funcional. Se busca realizar un análisis completamente estático de los programas (*i.e.* en tiempo de compilación), diferenciando el estudio aquí realizado del presentado en [Jay09] donde se relegan decisiones de tipado relativas a *path polymorphism* al tiempo de ejecución, modificando así la semántica operacional del cálculo tipado respecto al no tipado.

En el λ-cálculo las funciones son representadas con expresiones de la forma $\lambda x.t$, donde x es el parámetro formal y t el cuerpo. Dicha función puede ser aplicada a cualquier término, independientemente de su forma. Esto es capturado por la regla de β-reducción: $(\lambda x.t)\, s \mapsto_\beta \{x \setminus s\}t$. La regla de β-reducción del λ-cálculo no impone condiciones sobre la forma de s, puede ser cualquier término. Los *cálculos con patrones* [CK01, Kah04, JK06, KvOdV08, JK09] son generalizaciones del λ-cálculo en los cuales las abstracciones $\lambda x.t$ son reemplazadas por términos de la forma $\lambda p.t$, donde p es llamado el *patrón*. Por ejemplo, $\lambda\langle x, y\rangle.x$ donde p es $\langle x, y\rangle$, representando un par de términos; esta función proyecta la primera componente del par. La aplicación más externa en una expresión de la forma $(\lambda\langle x, y\rangle.x)\, s$ solo podrá ser contraída si s es de la forma $\langle s_0, s_1\rangle$, para expresiones s_0, s_1 cualesquiera; de lo contrario el cómputo deberá concentrarse en s.

El agregado de constantes al λ-cálculo permite, en particular, expresar estructuras de datos, y también patrones, como una combinación de constructores aplicados a términos (la llamada *notación aplicativa*). Por ejemplo, la lista $[1, 2]$ es codificada en el siguiente término l:

$$\texttt{cons}\,(\texttt{l}\,1)\,(\texttt{cons}\,(\texttt{l}\,2)\,\texttt{nil})$$

construido a partir de las constantes `cons`, `nil` y `l`. Notar el uso de la constante `l`(eaf) para inyectar valores de tipos estándar, en este caso *integers*, en la estructura. Otro ejemplo es el término t capturando un árbol binario:

$$\texttt{node}\,(\texttt{l}\,1)\,(\texttt{node}\,(\texttt{l}\,2)\,\texttt{nil}\,\texttt{nil})\,(\texttt{node}\,(\texttt{l}\,3)\,\texttt{nil}\,\texttt{nil})$$

Esta notación se aplica a patrones también: $\lambda\langle x, y\rangle.x$ puede escribirse $\lambda\texttt{p}\,x\,y.x$, donde `p` es una constante. Notar que `p` es aplicada a la variable x y el término resultante es, a su vez, aplicado a la variable y.

Considerar la siguiente función para actualizar los valores de cualquiera de las dos estructuras al

aplicar una función f provista por el usuario:

$$\begin{array}{lll} \mathsf{upd} = f \to & (\ \mathtt{l}\, z & \to \mathtt{l}\,(f\, z) \\ & \mid x\, y & \to (\mathsf{upd}\, f\, x)\,(\mathsf{upd}\, f\, y) \\ & \mid w & \to w \qquad\qquad\qquad\quad)\end{array} \tag{3.1}$$

Sea $(+1)$ la función sucesor. La expresión $\mathsf{upd}\,(+1)$ es aplicable tanto a l como a t, retornando una estructura actualizada donde todos los números en sus hojas fueron incrementados en uno. La expresión a la derecha del "=" es llamada *abstracción* y consiste de una única rama, la cual esta formada por un patrón (la variable f) y un cuerpo (en este caso una nueva abstracción que consiste de tres ramas). El argumento de una abstracción es comparado contra los patrones, en el orden que estos están escritos, y el cuerpo apropiado es seleccionado para continuar la ejecución. De particular interés es el patrón $x\, y$, el cual captura la esencia del *path polymorphism*: abstrae un camino en una estructura que debe ser "separado".

El presente capítulo se propone estudiar en profundidad este fenómeno, mediante el desarrollo de un sistema de tipos estático adecuado para un cálculo que permita expresar ejemplos como (3.1). Se denominará a este formalismo *Calculus of Applicative Patterns* (CAP) y a continuación se discuten los desafíos para desarrollar tal sistema de tipos. La motivación se hará mediante el uso de ejemplos que, al mismo tiempo, servirán como introducción a la sintaxis del cálculo. Para una definición completa y formal véase la Sec. 3.2.

Considerar las siguientes expresiones:

$$(\mathtt{nil} \to 0)\,\mathtt{cons} \qquad (\mathtt{l}\, x \to_{\{x:\mathsf{Nat}\}} x + 1)\,(\mathtt{l}\,\mathtt{true})$$

que en principio deberían ser rechazadas por el sistema de tipos a desarrollar. En el primer caso, la abstracción no es capaz de aceptar la constante `cons`. Para esto se incorporan *tipos constantes*, de modo que el patrón `nil` será asignado un tipo $\mathbb{nil}$ y, análogamente, el argumento `cons` será tipado con $\mathbb{cons}$ para luego comparar estas constantes. En el segundo caso, la variable x en el patrón $\mathtt{l}\, x$ precisa ser de tipo Nat para preservar el tipo de la expresión al reducir, pues la expresión resultante $x + 1$ así lo requiere. Sin embargo, el argumento de `l` en $\mathtt{l}\,\mathtt{true}$ es Bool. Esta situación será filtrada introduciendo *tipos aplicativos* [Pet11]: $\mathtt{l}\, x$ será asignado un tipo de la forma $\mathbb{l} \mathbin{@} \mathsf{Nat}$, mientras que $\mathtt{l}\,\mathtt{true}$ será asignado $\mathbb{l} \mathbin{@} \mathsf{Bool}$. Una simple comparación de las expresiones de tipos dictará que estos no son compatibles y, por lo tanto, el término resultante será rechazado por el sistema.

Considerar ahora la siguiente variante del ejemplo (3.1) donde las variables de los patrones tiene anotaciones de tipo:

$$\begin{array}{lll} \mathsf{upd} = f \to_{\{f:A \supset B\}} & (\ \mathtt{l}\, z \to_{\{z:A\}} & \mathtt{l}\,(f\, z) \\ & \mid x\, y \to_{\{x:C, y:D\}} & (\mathsf{upd}\, f\, x)\,(\mathsf{upd}\, f\, y) \\ & \mid w \to_{\{w:E\}} & w)\end{array} \tag{3.2}$$

En CAP, el usuario proveerá tales anotaciones de tipo. Por ejemplo, la declaración para f establece que ésta denota una función de A en B, por lo que z debe ser a su vez de tipo A. El caso de x e y requiere un análisis más en profundidad, para poder establecer qué forma deberían tener sus tipos C y D respectivamente, y así deducir el tipo del patrón $x\, y$. Tal patrón puede ser instanciado en diferentes términos en cada llamado recursivo a la función `upd`. Por ejemplo, considerar $\mathsf{upd}\,(+1)\, l$ (*i.e.* la lista $[1, 2]$ definida anteriormente). La siguiente tabla ilustra algunos de los términos con los cuales x e y son instanciadas durante la evaluación de $\mathsf{upd}\,(+1)\, l$:

	x	y
$\mathsf{upd}\,(+1)\, l$	$\mathtt{cons}\,(\mathtt{l}\,1)$	$\mathtt{cons}\,(\mathtt{l}\,2)\,\mathtt{nil}$
$\mathsf{upd}\,(+1)\,(\mathtt{cons}\,(\mathtt{l}\,1))$	`cons`	$\mathtt{l}\,1$
$\mathsf{upd}\,(+1)\,(\mathtt{cons}\,(\mathtt{l}\,2)\,\mathtt{nil})$	$\mathtt{cons}\,(\mathtt{l}\,2)$	`nil`

El tipo asignado a x (y a y) debe abarcar todos los términos en su respectiva columna. Esto sugiere la adopción de una disciplina de *tipos unión*. Asumiendo que el usuario provee una cobertura exhaustiva, el tipo de x en upd es:

$$F_A \triangleq \mu\alpha.(\mathbb{I} \mathbin{@} A) \oplus (\alpha \mathbin{@} \alpha) \oplus (\mathtt{nil} \oplus \mathtt{cons} \oplus \mathtt{node})$$

donde se agrega el tipo constante $\mathtt{node}$ para considerar también el caso $\mathsf{upd}\,(+1)\,t$. En esta expresión, μ es el constructor de *tipos recursivos*, mientras que $\oplus$ denota *tipos unión*. La variable y será asignada el mismo tipo F_A, resultando el patrón $x\,y$ de tipo $F_A \mathbin{@} F_A$. Por el análisis anterior, el primer patrón de la abstracción interna tendrá tipo $\mathbb{I} \mathbin{@} A$. Luego, bastará asignar $E \triangleq \mathtt{nil} \oplus \mathtt{cons} \oplus \mathtt{node}$ al patrón w y considerar el tipo del dominio de una abstracción como la unión del de sus patrones (en este caso $(\mathbb{I} \mathbin{@} A) \oplus (F_A \mathbin{@} F_A) \oplus (\mathtt{nil} \oplus \mathtt{cons} \oplus \mathtt{node})$) para terminar asignando a la función upd el tipo $(A \supset B) \supset (F_A \supset F_B)$, apelando a una noción adecuada de equivalencia para tipos recursivos [BDS13]: $\mu V.A \simeq \{V \setminus \mu V.A\}A$. Notar que el tipo de la imagen de la abstracción interna también resulta en la unión del tipo de las respectivas ramas pues, *a priori*, no es posible saber qué cuerpo será efectivamente evaluado al reducir.

Como se mencionó, $\mathsf{upd}\,(+1)$ debe ser aplicable tanto a la lista l como al árbol t. Para eso, se debe asegurar que los tipos de l y t son adecuados. La combinación de herramientas propuesta (*i.e.* tipos constantes, aplicativos, unión y recursivos) permite expresar *datatypes* de manera uniforme para estructuras algebraicas como listas y árboles. En este caso, l y t serán asignados respectivamente los tipos:

$$\mathsf{List}_A \triangleq \mu\alpha.\mathtt{nil} \oplus (\mathtt{cons} \mathbin{@} A \mathbin{@} \alpha) \qquad \mathsf{Tree}_A \triangleq \mu\alpha.\mathtt{nil} \oplus (\mathtt{node} \mathbin{@} A \mathbin{@} \alpha \mathbin{@} \alpha)$$

Una noción adecuada de *sub-tipado* permitirá aplicar $\mathsf{upd}\,(+1)$ a ambas estructuras. En efecto, ambos tipos $\mathsf{List}_{\mathsf{Nat}}$ y $\mathsf{Tree}_{\mathsf{Nat}}$ resultarán sub-tipos de F_{Nat}. Será la correcta formulación de esta relación de sub-tipado una de las claves para poder garantizar las propiedades de adecuación del sistema de tipos: *i.e. subject reduction* y *progress*. La primera garantiza la estabilidad del tipado por reducción, mientras que la segunda asegura que la reducción siempre es posible hasta tanto se consiga un resultado o *valor* (*i.e.* no se estanca en un término mal formado).

Aunque el sub-tipado es importante para el resultado de adecuación, una noción novedosa de *compatibilidad de patrones* será crucial a la hora de probar estas propiedades. Considerar el siguiente ejemplo:

$$(\mathtt{l}\,x \to_{\{x:\mathsf{Bool}\}} \mathtt{if}\ x\ \mathtt{then}\ 1\ \mathtt{else}\ 0) \mid (\mathtt{l}\,y \to_{\{y:\mathsf{Nat}\}} y + 1) \tag{3.3}$$

Si bien hay una rama capaz de manipular correctamente el argumento $\mathtt{l}\,4$ (la segunda), CAP evaluará los casos de izquierda a derecha siguiendo las practicas estándar de los lenguajes de programación funcional. Dado que la operación de match también tiene éxito al evaluar el término $\mathtt{l}\,4$ con el patrón $\mathtt{l}\,x$, se obtendrá incorrectamente el reducto $\mathtt{if}\ 4\ \mathtt{then}\ 1\ \mathtt{else}\ 0$. Por lo tanto, es necesario relacionar los patrones $\mathtt{l}\,x$ y $\mathtt{l}\,y$ para evitar la pérdida de la propiedad de subject reduction. En particular, $\mathtt{l}\,y$ es una instancia de $\mathtt{l}\,x$, por lo que se requerirá que el tipo de la segunda rama sea sub-tipo del de la primera (*i.e.* $\mathbb{I} \mathbin{@} \mathsf{Nat} \preceq \mathbb{I} \mathbin{@} \mathsf{Bool}$), puesto que ésta última siempre toma precedencia. En este caso, la condición de sub-tipado falla dado que $\mathsf{Nat} \not\preceq \mathsf{Bool}$, por lo que el ejemplo (3.3) será rechazado por el sistema de tipos. Ejemplos como (3.2) requieren de un análisis más detallado (*cf.* Sec. 3.3.4). A modo de resumen, la compatibilidad se focaliza en las posiciones de mismatch entre los patrones, imponiendo una condición de sub-tipado entre los tipos asociados siempre que haya un solapamiento estructural entre sus potenciales argumentos.

3.1. Preliminares

Esta sección introduce principalmente el *Pure Pattern Calculus*, herramienta que inspira el presente trabajo.

3.1.1. Pure Pattern Calculus

El *Pure Pattern Calculus* (PPC) [JK06, JK09, Jay09] se introdujo como una extensión del λ-cálculo con soporte para *pattern matching* que permite abstraer términos arbitrarios. Esto transforma a los patrones en "ciudadanos de primera clase", que pueden ser pasados por parámetro, evaluados y devueltos como resultados de una función. Este novedoso cálculo introduce a su vez dos nuevas formas de polimorfismo: *path polymorphism* y *pattern polymorphism*. El primero se refiere a la capacidad de definir funciones que recorran recursivamente estructuras de datos arbitrarias, mientras que el segundo permite tratar los patrones como parámetros que pueden ser generados dinámicamente. Su desarrollo conlleva una serie de dificultades técnicas para garantizar las buenas propiedades de la semántica operacional, esencialmente la preservación de la confluencia, la cual se basa en la definición de una *operación de matching* adecuada.

En particular se considera el PPC con símbolos *matchable* introducido en [JK09]. Dado un conjunto infinito numerable de *símbolos* $\mathbb{V}$ $(x, y, \ldots)$, la sintaxis de PPC se define como:

$$\begin{array}{rrcl} \textbf{Términos} & t & ::= & x \mid \widehat{x} \mid t\,t \mid t \to_\theta t \\ \textbf{Contextos} & \mathtt{C} & ::= & \Box \mid \mathtt{C}\,t \mid t\,\mathtt{C} \mid \mathtt{C} \to_\theta t \mid t \to_\theta \mathtt{C} \end{array}$$

Un término de la forma x es llamado una *variable*, mientras que se denomina a $\widehat{x}$ un *matchable*. A su vez se tiene la clásica *aplicación* $t\,u$ y la *abstracción* es denotada $p \to_\theta s$, donde p es el *patrón*, s el *cuerpo* y θ es un conjunto de símbolos utilizado para definir los conjuntos de *variables libres* ($\mathsf{fv}(\_)$) y *matchables libres* ($\mathsf{fm}(\_)$) de un término

$$\begin{array}{rcl@{\qquad}rcl} \mathsf{fv}(x) & \triangleq & \{x\} & \mathsf{fm}(x) & \triangleq & \emptyset \\ \mathsf{fv}(\widehat{x}) & \triangleq & \emptyset & \mathsf{fm}(\widehat{x}) & \triangleq & \{x\} \\ \mathsf{fv}(t\,u) & \triangleq & \mathsf{fv}(t) \cup \mathsf{fv}(u) & \mathsf{fm}(t\,u) & \triangleq & \mathsf{fm}(t) \cup \mathsf{fm}(u) \\ \mathsf{fv}(p \to_\theta t) & \triangleq & \mathsf{fv}(p) \cup (\mathsf{fv}(t) \setminus \theta) & \mathsf{fm}(p \to_\theta t) & \triangleq & (\mathsf{fm}(p) \setminus \theta) \cup \mathsf{fm}(t) \end{array}$$

Intuitivamente, dado $x \in \theta$, la abstracción $p \to_\theta s$ liga las ocurrencias libres de x como variable en s y sus ocurrencias libres como matchable en p.

Un término es *cerrado* si no tiene ocurrencias libres de variables. Notar que se permiten ocurrencias libres de matchables, las cuales juegan el rol de constantes o constructores, utilizadas para definir *data structures*[1]. El patrón p en la abstracción $p \to_\theta s$ se dice *lineal* si cada símbolo $x \in \theta$ tiene una única ocurrencia como matchable libre en p. Sin pérdida de generalidad, se trabaja en PPC con patrones lineales [JK09], para evitar así la falla innecesaria de la operación de matching (introducida más adelante).

Como es usual al trabajar con λ-cálculo y sus extensiones, se utiliza la convención de nombres de Barendregt [Bar85] (α-conversión), teniendo en cuenta que al renombrar el símbolo ligado x por uno fresco y en $p \to_\theta s$, se sustituye x por y en θ, $\widehat{x}$ por $\widehat{y}$ en p, y x por y en s.

Un *match* $(\mu, \nu, \ldots)$ puede ser exitoso (retornando una sustitución σ), fallar (en cuyo caso retorna un símbolo especial `fail`) o estar indeterminado (retornando el símbolo especial `wait`). Los casos de éxito y falla se dicen *decididos*. A su vez se extienden los conceptos y notaciones asociados a sustituciones para el caso del match. El dominio de `fail` es el conjunto vacío mientas que el de `wait` es indefinido. Los conjuntos de variables y matchables libres de σ se definen como la unión de los $\mathsf{fv}(\sigma x)$ y los $\mathsf{fm}(\sigma x)$ respectivamente, para toda $x \in \mathsf{dom}(\sigma)$, mientras que $\mathsf{fv}(\mathtt{fail}) \triangleq \mathsf{fm}(\mathtt{fail}) \triangleq \emptyset$ y estos se encuentran indefinidos para `wait`. El conjunto de símbolos de un match se define entonces como $\mathsf{sym}(\mu) \triangleq \mathsf{dom}(\mu) \cup \mathsf{fv}(\mu) \cup \mathsf{fm}(\mu)$. Se usa el predicado x `evita` μ para denotar $x \notin \mathsf{sym}(\mu)$. El mismo

[1] En general, se utiliza la sintaxis `c` para denotar constantes arbitrarias en PPC, que pueden verse como matchables $\widehat{c}$ ($c \in \mathbb{V}$) que no son capturados por ningún ligador.

se extiende a conjuntos de la manera esperada, θ `evita` μ. En particular, si θ `evita` μ entonces μ debe estar decidido.

El resultado de aplicar una sustitución σ a un término t se define sobre términos de PPC como:

$$\begin{array}{rcll} \sigma x & \triangleq & \sigma(x) & \\ \sigma\widehat{x} & \triangleq & \widehat{x} & \\ \sigma(t\,u) & \triangleq & (\sigma t)\,(\sigma u) & \\ \sigma(p \to_\theta s) & \triangleq & \sigma p \to_\theta \sigma s & \theta \texttt{ evita } \sigma \end{array}$$

La restricción en la definición de $\sigma(p \to_\theta s)$ es necesaria para evitar capturas indeseadas de variables o matchables, que podrían cambiar la semántica del término. Notar que siempre puede satisfacerse dicha condición apelando a α-conversión.

El resultado de aplicar un match μ a un término t es notado $\mu\,t$ y de define como: si μ es una sustitución σ, luego $\mu\,t \triangleq \sigma t$; si $\mu = \texttt{fail}$, entonces $\mu\,t$ se define como la constante distinguida `NoMatch`[2]; si $\mu = \texttt{wait}$, entonces $\mu\,t$ está indefinido.

La *composición* $\sigma_1 \circ \sigma_0$ de sustituciones σ_0 y σ_1 se define como $(\sigma_1 \circ \sigma_0)x \triangleq \sigma_1(\sigma_0 x)$. Más aún, la noción se extiende a matches μ_0 y μ_1 definiendo $\mu_1 \circ \mu_0 \triangleq \texttt{fail}$ cuando al menos uno de los dos es `fail`. De lo contrario, si uno de los dos es `wait`, entonces $\mu_1 \circ \mu_0 \triangleq \texttt{wait}$. En particular $\texttt{fail} \circ \texttt{wait} = \texttt{fail}$.

La *unión disjunta* $\mu_0 \uplus \mu_1$ de matches μ_0 y μ_1 se define de la siguiente manera: si uno de los dos es `fail` o la intersección de sus dominios no es vacía, entonces $\mu_0 \uplus \mu_1 \triangleq \texttt{fail}$; de lo contrario, si uno de los dos es `wait`, entonces $\mu_0 \uplus \mu_1 \triangleq \texttt{wait}$; en cualquier otro caso μ_0 y μ_1 son sustituciones tal que $\mathsf{dom}(\mu_0) \cap \mathsf{dom}(\mu_1) = \emptyset$, entonces se retorna la sustitución dada por:

$$(\mu_0 \uplus \mu_1)x = \begin{cases} \mu_0 x & \text{si } x \in \mathsf{dom}(\mu_0) \\ \mu_1 x & \text{si } x \in \mathsf{dom}(\mu_1) \\ x & \text{si no} \end{cases}$$

Se usa unión disjunta para garantizar que la operación de matching sea determinista.

Por último, antes de introducir la operación de matching es necesario motivar el concepto de *matchable form*. El patrón $\widehat{x}\,\widehat{y}$ permite, en principio, descomponer aplicaciones arbitrarias, lo que puede resultar en la pérdida de confluencia del cálculo

$$\begin{array}{lcl} (\widehat{x}\,\widehat{y} \to_{\{x,y\}} y)\,((\widehat{w} \to_{\{w\}} \widehat{z_0}\,\widehat{z_1})\,\widehat{z_0}) & \to & (\widehat{x}\,\widehat{y} \to_{\{x,y\}} y)\,(\widehat{z_0}\,\widehat{z_1}) \quad \to \quad \widehat{z_1} \\ (\widehat{x}\,\widehat{y} \to_{\{x,y\}} y)\,((\widehat{w} \to_{\{w\}} \widehat{z_0}\,\widehat{z_1})\,\widehat{z_0}) & \to & \widehat{z_0} \end{array}$$

El problema surge al permitir el match entre el patrón $\widehat{x}\,\widehat{y}$ y una aplicación que todavía puede ser reducida como $(\widehat{w} \to_{\{w\}} \widehat{z_0}\,\widehat{z_1})\,\widehat{z_0}$. Para evitar esta situación se requiere que el match solo pueda estar decidido si el argumento se encuentra lo suficientemente evaluado. Un problema similar ocurre si el patrón es reducible, por lo que ambos (patrón y argumento) deben ser *matchable forms*. Los conjuntos de *data structures* y *matchable forms* están dados por la siguiente gramática:

$$\begin{array}{rrcl} \textbf{Data structures} & d & ::= & \widehat{x} \mid d\,t \\ \textbf{Matchable forms} & m & ::= & d \mid t \to_\theta t \end{array}$$

La *operación de matching* $\{\!\{p \setminus^\theta u\}\!\}$ de un patrón p contra un término u relativa a un conjunto de

[2]Esta variante de la definición es tomada de [Jay09]. Cabe destacar que en [JK09] se define $\texttt{fail}\,t \triangleq \widehat{x} \to_{\{x\}} x$, al mismo tiempo que se muestra que puede seleccionarse cualquier combinador sin afectar las buenas propiedades del cálculo (*e.g.* confluencia).

símbolos θ se define como la aplicación de las siguientes ecuaciones en orden:

$$\begin{array}{rcll} \{\!\{\widehat{x} \setminus^{\theta} u\}\!\} & \triangleq & \{x \setminus u\} & \text{si } x \in \theta \\ \{\!\{\widehat{x} \setminus^{\theta} \widehat{x}\}\!\} & \triangleq & \{\} & \text{si } x \notin \theta \\ \{\!\{p\,q \setminus^{\theta} t\,u\}\!\} & \triangleq & \{\!\{p \setminus^{\theta} t\}\!\} \uplus \{\!\{q \setminus^{\theta} u\}\!\} & \text{si } t\,u \text{ y } p\,q \text{ son } \textit{matchable forms} \\ \{\!\{p \setminus^{\theta} u\}\!\} & \triangleq & \texttt{fail} & \text{si } u \text{ y } p \text{ son } \textit{matchable forms} \\ \{\!\{p \setminus^{\theta} u\}\!\} & \triangleq & \texttt{wait} & \end{array}$$

Adicionalmente, se verifica que $\mathsf{dom}(\{\!\{p \setminus^{\theta} u\}\!\}) = \theta$. De lo contrario se define $\{\!\{p \setminus^{\theta} u\}\!\} \triangleq \texttt{fail}$. Esta última condición es necesaria para evitar que variables ligadas se salgan de *scope* al reducir. Para garantizarla basta pedir, dada la abstracción $p \to_{\theta} s$, que $\theta \subseteq \mathsf{fm}(p)$.

La relación de reducción $\to_{\mathtt{PPC}}$ para términos de PPC está dada por la clausura por contextos $\mathtt{C}$ de la regla:

$$(p \to_{\theta} s)\, u \mapsto_{\mathtt{PPC}} \{\!\{p \setminus^{\theta} u\}\!\}\, s$$

Para ilustrar esta relación, considerar el término $(\widehat{x}\,\widehat{y} \to_{\{x,y\}} y)\,((\widehat{w} \to_{\{w\}} \widehat{z_0}\,\widehat{z_1})\,\widehat{z_0})$ dado anteriormente. El matching entre el patrón $\widehat{x}\,\widehat{y}$ y el argumento $(\widehat{w} \to_{\{w\}} \widehat{z_0}\,\widehat{z_1})\,\widehat{z_0}$ no se encuentra aún decidido. Luego, el término orginal contiene un único redex, dado que $\{\!\{\widehat{w} \setminus \widehat{z_0}\}\!\} = \{w \setminus \widehat{z_0}\}$. Se tiene entonces la siguiente secuencia de reducción:

$$(\widehat{x}\,\widehat{y} \to_{\{x,y\}} y)\,((\widehat{w} \to_{\{w\}} \widehat{z_0}\,\widehat{z_1})\,\widehat{z_0}) \quad \to \quad (\widehat{x}\,\widehat{y} \to_{\{x,y\}} y)\,(\widehat{z_0}\,\widehat{z_1}) \quad \to \quad \widehat{z_1}$$

Más aún, puede garantizarse la buena definición de la relación basada en el operación de matching definida anteriormente.

Teorema 3.1.1 ([JK09])**.** *La relación de reducción $\to_{\mathtt{PPC}}$ es confluente (CR).*

La prueba de esta propiedad se basa en la *condición de matching rígido* (RMC) introducida en [JK09], para la cual se define una noción de reducción simultanea $\Rightarrow_{\mathtt{PPC}}$ sobre PPC. Se dice que una operación de matching $\{\!\{p \setminus^{\theta} u\}\!\}$ satisface RMC si para todas las reducciones simultaneas $u \Rightarrow_{\mathtt{PPC}} u'$, $p \Rightarrow_{\mathtt{PPC}} p'$ y $s \Rightarrow_{\mathtt{PPC}} s'$, se tiene $\{\!\{p \setminus^{\theta} u\}\!\}\, s \Rightarrow_{\mathtt{PPC}} \{\!\{p' \setminus^{\theta} u'\}\!\}\, s'$. Luego, puede verse que si $\{\!\{p \setminus^{\theta} u\}\!\}$ satisface RMC, entonces $\Rightarrow_{\mathtt{PPC}}$ tiene la propiedad del diamante, lo que implica la confluencia de $\to_{\mathtt{PPC}}$. En particular, la definición de $\{\!\{p \setminus^{\theta} u\}\!\}$ dada anteriormente satisface RMC.

Dos aspectos de interés para poder realizar un estudio acabado de las propiedades de PPC son: (1) poder definir funciones por casos o *branches*; y (2) poder codificar funciones recursivas.

Para el primero, se introduce la sintaxis de *alternativa* $p \to_{\theta} s \mid r$, codificada en PPC con el término

$$p \to_{\theta} s \mid r \triangleq \widehat{x} \to_{\{x\}} (\mathtt{NoMatch}\,\widehat{y} \to_{\{y\}} y)\,((p \to_{\theta} \widehat{z} \to_{\{z\}} \mathtt{NoMatch}\,s)\,x\,(r\,x))$$

donde $x, y, z \in \mathbb{V}$ son símbolos frescos, *i.e.* sin ocurrencias libres ni ligadas en p, s o r. Este operador se asume asociativo a derecha, por lo que $p_0 \to_{\theta_0} s_0 \mid p_1 \to_{\theta_1} s_1 \mid \ldots \mid p_n \to_{\theta_n} s_n$ es $p_0 \to_{\theta_0} s_0 \mid (p_1 \to_{\theta_1} s_1 \mid (\ldots \mid p_n \to_{\theta_n} s_n))$. Notar que la rama izquierda toma precedencia pues, dado $(p \to_{\theta} s \mid r)\,u$, la ejecución continua con $\{\!\{p \setminus^{\theta} u\}\!\}\, s$ si el match $\{\!\{p \setminus^{\theta} u\}\!\}$ es exitoso, siendo el sub-término $r\,u$ descartado al unificar con el símbolo $z \notin \mathsf{fv}(s)$. Caso contrario, si $\{\!\{p \setminus^{\theta} u\}\!\} = \texttt{fail}$, se obtiene $\mathtt{NoMatch}\,(r\,u)$ y la ejecución continua por la rama r. En particular, dado el término $(p_0 \to_{\theta_0} s_0 \mid p_1 \to_{\theta_1} s_1 \mid \ldots \mid p_n \to_{\theta_n} s_n)\,u$ la ejecución continuara por la primer rama (de izquierda a derecha) cuyo match $\{\!\{p_i \setminus^{\theta_i} u\}\!\}$ sea exitoso, emulando la semántica operacional de lenguajes de programación con pattern matching como OCaml, Haskell, etc.

En cuanto al segundo aspecto, es de vital importancia dada la naturaleza recursiva de *path polymorphism*. Para codificar este tipo de funciones puede apelarse a un operador de *punto fijo* como el *paradoxical combinator* de Curry [Bar85], el cual es expresado en PPC como

$$\mathsf{Y} \triangleq \widehat{f} \to_{\{f\}} (\widehat{x} \to_{\{x\}} f\,(x\,x))\,(\widehat{x} \to_{\{x\}} f\,(x\,x))$$

Luego, dado un término t y un símbolo f tal que $f \in \mathsf{fv}(t)$, puede definirse la función recursiva $f = t$ como $\mathsf{Y}\,(\widehat{f} \to_{\{f\}} t)$. Por ejemplo,

$$\begin{array}{lll} \mathsf{upd} = \widehat{x} \to_{\{x\}} \widehat{f} \to_{\{f\}} & (\; x\,\widehat{w} & \to_{\{w\}} \; x\,(f\,w) \\ & \mid \widehat{y}\,\widehat{z} & \to_{\{y,z\}} \; (\mathsf{upd}\,x\,f\,y)\,(\mathsf{upd}\,x\,f\,z) \\ & \mid \widehat{g} & \to_{\{g\}} \; g \qquad\qquad\qquad\qquad) \end{array}$$

Este término define una función de *update* genérica, que al ser aplicada a un constructor `pt`, una función f y un data structure d, reemplaza sub-términos de d de la forma $\mathtt{pt}\,t$ por $\mathtt{pt}\,(f\,t)$. Notar que la condición $f \in \mathsf{fv}(t)$ no es estrictamente necesaria para definir la sintaxis "$=$", de no satisfacerse simplemente se obtiene una función no recursiva:

$$\begin{array}{lll} \mathsf{eq} = \widehat{x} \to_{\{x\}} & (\; x & \to_{\{\}} \; \mathtt{true} \\ & \mid \widehat{y} & \to_{\{y\}} \; \mathtt{false} \;) \end{array}$$

3.1.2. Restricciones de PPC

Adicionalmente, en [JK09] los autores distinguen dos fragmentos propios de PPC en un estudio de cómo otros cálculos con patrones de la literatura son capturados por este formalismo. Dichos fragmentos resultan de restringir la expresividad de los patrones a *patrones estáticos* y *patrones algebraicos* respectivamente.

El primero descarta la forma de *pattern polymorphism* al no permitir la presencia de variables libres en los patrones. Esto conlleva la simplificación de la sintaxis puesto que tampoco se permiten abstracciones en los patrones y los símbolos matchable pasan a jugar el rol de constantes únicamente. A su vez, pueden evitarse los conjuntos θ pues el ligado es implícito: *i.e.* todas las variables libres del patrón son capturadas por la abstracción. Más aún, esto permite evitar la distinción entre matchables y variables, manteniendo únicamente a estas últimas e incorporando un conjunto infinito numerable de *constantes* $\mathbb{C}$ ($\mathtt{c}, \mathtt{d}, \ldots$). Luego, los conjuntos de *patrones* $\mathbb{P}_{\mathsf{PPC}}$, *términos* $\mathbb{T}_{\mathsf{PPC}}$, *data structures* $\mathbb{D}_{\mathsf{PPC}}$ y *matchable forms* $\mathbb{M}_{\mathsf{PPC}}$ del fragmento estático de PPC están dados entonces por la siguiente gramática:

$$\begin{array}{rrcl} \textbf{Patrones} & p & ::= & x \in \mathbb{V} \mid \mathtt{c} \in \mathbb{C} \mid p\,p \\ \textbf{Términos} & t & ::= & x \in \mathbb{V} \mid \mathtt{c} \in \mathbb{C} \mid t\,t \mid p \to t \\ \textbf{Data structures} & d & ::= & \mathtt{c} \in \mathbb{C} \mid d\,t \\ \textbf{Matchable forms} & m & ::= & d \mid p \to t \end{array}$$

Esto también permite simplificar la *operación de matching*

$$\begin{array}{rcll} \{\!\{x \setminus u\}\!\} & \triangleq & \{x \setminus u\} & \\ \{\!\{\mathtt{c} \setminus \mathtt{c}\}\!\} & \triangleq & \{\} & \\ \{\!\{p\,q \setminus t\,u\}\!\} & \triangleq & \{\!\{p \setminus t\}\!\} \uplus \{\!\{q \setminus u\}\!\} & \text{si } t\,u \text{ es una } \textit{matchable form} \\ \{\!\{p \setminus u\}\!\} & \triangleq & \mathtt{fail} & \text{si } u \text{ es una } \textit{matchable form} \\ \{\!\{p \setminus u\}\!\} & \triangleq & \mathtt{wait} & \end{array}$$

La característica distintiva de este fragmento es la posibilidad de expresar aplicaciones arbitrarias entre los patrones, lo que permite capturar *path polymorphism*. Por ejemplo, puede expresarse la función de *update* presentada anteriormente, fijando de antemano el constructor `pt`:

$$\begin{array}{lll} \mathsf{upd} = f \to & (\; \mathtt{pt}\,w & \to \; \mathtt{pt}\,(f\,w) \\ & \mid y\,z & \to \; (\mathsf{upd}\,f\,y)\,(\mathsf{upd}\,f\,z) \\ & \mid g & \to \; g \qquad\qquad) \end{array} \tag{3.4}$$

El segundo fragmento mencionado implica restringir aún más la sintaxis de los patrones, forzando a que todo patrón *no trivial* sea una estructura de datos algebraica (*i.e.* encabezada por una constante o constructor)

$$\begin{array}{rrcl} \textbf{Patrones algebraicos} & p & ::= & x \in \mathbb{V} \mid q \\ \textbf{Patrones no triviales} & q & ::= & \mathtt{c} \in \mathbb{C} \mid q\,p \end{array}$$

Este fragmento descarta a su vez la forma de *path polymorphism*. Ejemplos como (3.4) ya no pueden codificarse ante la imposibilidad de expresar el patrón $y\,z$. Sin embargo, el combinador Y sí es capturado por este fragmento, permitiendo definir funciones recursivas como:

$$\begin{array}{rl} \mathsf{plus} = & (\ \mathtt{zero} \ \to\ y\ \to\ y \\ & \mid\ \mathtt{succ}\,x\ \to\ y\ \to\ \mathtt{succ}\,(\mathsf{plus}\,x\,y)\ ) \end{array}$$

Notar que la definición de la alternativa $p \to s \mid r$ presentada anteriormente para el caso general de PPC es válida tanto para el fragmento estático como algebraico, solo restringiendo p, s y r a la sintaxis correspondiente.

3.2. Sintaxis de CAP

En la presente sección se intruce la sintáxis y semántica operacional del *Calculus of Applicative Patterns* (CAP), el cual constituye esencialmente el fragmento estático de PPC. El objetivo de introducir esta herramienta es poder focalizar el estudio en *path polymorphism*, dejando de lado fundamentalmente dos dificultades adicionales que PPC propone: (1) los patrones dinámicos (*pattern polymorphism*); y (2) el manejo explícito de la falla de la operación de matching. Para ello sencillamente se divide la sintaxis en dos categorías principales: *patrones* $\mathbb{P}_{\mathsf{CAP}}$ $(p, q, \ldots)$ y *términos* $\mathbb{T}_{\mathsf{CAP}}$ $(s, t, \ldots)$, incorporando la alternativa como constructor de términos del cálculo, lo que implica a su vez la modificación de la regla de reducción.

Dados conjuntos infinito numerables de *variables* $\mathbb{V}$ $(x, y, \ldots)$ y *constantes* $\mathbb{C}$ $(\mathtt{c}, \mathtt{d}, \ldots)$, los conjuntos de *patrones* $\mathbb{P}_{\mathsf{CAP}}$ y *términos* $\mathbb{T}_{\mathsf{CAP}}$ se definen mediante la siguiente gramática:

$$\begin{array}{rrcl} \textbf{Patrones} & p & ::= & x \in \mathbb{V} \mid \mathtt{c} \in \mathbb{C} \mid p\,p \\ \textbf{Términos} & t & ::= & x \in \mathbb{V} \mid \mathtt{c} \in \mathbb{C} \mid t\,t \mid (p \to t \mid \ldots \mid p \to t) \\ \textbf{Contextos} & \mathtt{C} & ::= & \Box \mid \mathtt{C}\,t \mid t\,\mathtt{C} \mid (p \to s \mid \ldots \mid p \to \mathtt{C} \mid \ldots \mid p \to s) \end{array}$$

Notar que los patrones comprenden un sub-conjunto propio de los términos, y resultan ser exactamente los mismo que aquellos del fragmento estático de PPC. Las *abstracciones* son aquí extendidas a varias ramas, de modo que la alternativa es en CAP un constructor nativo del cálculo. Esta decisión de diseño se relaciona con la noción de *compatibilidad* (*cf.* Sec. 3.3.4), la cual establece una relación entre los patrones que participan de una misma alternativa y sus tipos asociados, de modo de garantizar el buen comportamiento dinámico del cálculo. Por conveniencia, abstracciones de la forma $p_0 \to t_0 \mid \ldots \mid p_{n-1} \to t_{n-1}$, con $n > 0$, se abrevian $(p_i \to t_i)_{i<n}$. Así mismo, contextos de la forma $p_0 \to s_0 \mid \ldots \mid p_k \to \mathtt{C} \mid \ldots \mid p_{n-1} \to s_{n-1}$, con $n > 0$ y $k < n$, se abrevian $(p_i \to \mathtt{C}_k)_{i<n}$ siempre que sea posible, indicando que el contexto $\mathtt{C}$ se encuentra en la rama k.

Los conceptos de *data structures* $\mathbb{D}_{\mathsf{CAP}}$ $(d, e, \ldots)$ y *matchable forms* $\mathbb{M}_{\mathsf{CAP}}$ $(m, n, \ldots)$ se preservan respecto a PPC, solo que ahora debe tenerse en cuenta que las abstracciones pueden contener varias ramas en su definición.

$$\begin{array}{rrcl} \textbf{Data structures} & d & ::= & \mathtt{c} \mid d\,t \\ \textbf{Matchable forms} & m & ::= & d \mid (p \to t \mid \ldots \mid p \to t) \end{array}$$

Notar que la intersección entre los conjuntos de patrones y data structures es no vacía. En particular, todo patrón encabezado por una constante (o *constructor*) es un data structure.

Se introducen a continuación conceptos preliminares necesarios para la correcta presentación de la semántica operacional del cálculo.

Los conjuntos de *variables libres* de un término o patrón ($\mathsf{fv}(\_)$) se define inductivamente como:

$$\begin{array}{rcl} \mathsf{fv}(x) & \triangleq & \{x\} \\ \mathsf{fv}(\mathsf{c}) & \triangleq & \emptyset \\ \mathsf{fv}(t\,u) & \triangleq & \mathsf{fv}(t) \cup \mathsf{fv}(u) \\ \mathsf{fv}((p_i \to t_i)_{i<n}) & \triangleq & \bigcup_{i<n} (\mathsf{fv}(t_i) \setminus \mathsf{fv}(p_i)) \end{array}$$

La definición es extendida a conjuntos de términos de manera inmediata, simplemente tomando la unión de las variables libres de cada elemento del conjunto.

El resultado de aplicar una sustitución σ a un término t se define sobre términos de CAP como:

$$\begin{array}{rcll} \sigma x & \triangleq & \sigma(x) & \\ \sigma(\mathsf{c}) & \triangleq & \mathsf{c} & \\ \sigma(t\,u) & \triangleq & (\sigma t)\,(\sigma u) & \\ \sigma(p_i \to t_i)_{i<n} & \triangleq & (p_i \to \sigma t_i)_{i<n} & (\mathsf{fv}(p_i)\ \mathtt{evita}\ \sigma)_{i<n} \end{array}$$

La condición adicional $(\mathsf{fv}(p_i)\ \mathtt{evita}\ \sigma)_{i<n}$ garantiza que no se producen capturas indeseadas y puede satisfacerse en cualquier situación por α-conversión.

En cuanto a la operación de matching para CAP, se utiliza la misma definición que para el fragmento estático de PPC (*cf.* Sec. 3.1.2) considerando el conjunto de matchable forms de CAP, anteriormente definido. Al igual que en PPC, se asume una condición de linealidad en los patrones para evitar casos de falla triviales en la operación de matching.

La *semántica operacional* de CAP está dada por un único axioma de reducción, que extiende de manera natural la regla de β-reducción del λ-cálculo:

$$\frac{(\{\!\{p_i \setminus u\}\!\} = \mathtt{fail})_{i<k} \quad \{\!\{p_k \setminus u\}\!\} = \sigma \quad k < n}{(p_i \to s_i)_{i<n}\, u \mapsto_{\mathtt{CAP}} \sigma s_k}$$

Intuitivamente, se busca hacer coincidir el argumento u contra los patrones de $(p_i \to s_i)_{i<n}$, evaluándolos de izquierda a derecha y seleccionando la primera rama de éxito, siempre que las anteriores ya se encuentren decididas. La relación de reducción $\to_{\mathtt{CAP}}$ para CAP se define entonces como la clausura por contextos $\mathtt{C}$ de la regla $\mapsto_{\mathtt{CAP}}$.

El siguiente ejemplo ilustra la aplicación de la regla de reducción y la operación de matching:

$$\begin{array}{ll} & (\mathtt{true} \to t \mid \mathtt{false} \to s)\,((\mathtt{true} \to \mathtt{false} \mid \mathtt{false} \to \mathtt{true})\,\mathtt{true}) \\ \to_{\mathtt{CAP}} & (\mathtt{true} \to t \mid \mathtt{false} \to s)\,\mathtt{false} \\ \to_{\mathtt{CAP}} & s \end{array}$$

Notar que en $(\mathtt{true} \to t \mid \mathtt{false} \to s)\,\mathtt{false}$ la segunda rama es seleccionada dado que $\{\!\{\mathtt{true} \setminus \mathtt{false}\}\!\} = \mathtt{fail}$. Adicionalmente, considerar el siguiente ejemplo que ilustra como un data structure es descompuesto por la operación de matching:

$$\begin{array}{l} (\mathtt{nil} \to \mathtt{nothing} \mid \mathtt{cons}\,(\mathtt{l}\,x)\,y \to \mathtt{just}\,x)\,(\mathtt{cons}\,(\mathtt{l}\,t)\,(\mathtt{cons}\,(\mathtt{l}\,s)\,\mathtt{nil})) \\ \quad \to_{\mathtt{CAP}} \mathtt{just}\,t \end{array}$$

donde

$$\begin{array}{rcl} \{\!\{\mathtt{nil} \setminus \mathtt{cons}\,(\mathtt{l}\,t)\,(\mathtt{cons}\,(\mathtt{l}\,s)\,\mathtt{nil})\}\!\} & = & \mathtt{fail} \\ \{\!\{\mathtt{cons}\,(\mathtt{l}\,x)\,y \setminus \mathtt{cons}\,(\mathtt{l}\,t)\,(\mathtt{cons}\,(\mathtt{l}\,s)\,\mathtt{nil})\}\!\} & = & \{x \setminus t, y \setminus \mathtt{cons}\,(\mathtt{l}\,s)\,\mathtt{nil}\} \end{array}$$

Como es esperado, la relación de reducción de CAP resulta confluente.

Teorema 3.2.1. *La relación de reducción* $\to_{\mathsf{CAP}}$ *es confluente (CR).*

Demostración. La propiedad se prueba mediante una adaptación directa de la técnica de Tait–Martin-Löf [Tak95] a la sintaxis y semántica operacional de CAP. Todos los detalles se encuentran en el Ap. B. □

3.2.1. Relación entre CAP y PPC

CAP puede demostrarse equivalente al fragmento estático de PPC, en el sentido que ambos formalismos capturan los mismos programas y pueden simularse mutuamente. Formalmente, se definen funciones de traducción entre los cálculos y se demuestra que para toda reducción en uno de ellos existe una contra-parte en el otro vía la traducción.

Definición 3.2.2. *La* traducción $\_^{PPC}$ *de términos de CAP en términos del fragmento estático de PPC se define como*

$$\begin{array}{rcl} x^{PPC} & \triangleq & x \\ \mathsf{c}^{PPC} & \triangleq & \mathsf{c} \\ (r\,u)^{PPC} & \triangleq & r^{PPC}\,u^{PPC} \\ (p_0 \to s_0 \mid \ldots \mid p_n \to s_n)^{PPC} & \triangleq & p_0 \to {s_0}^{PPC} \mid (\ldots \mid p_n \to {s_n}^{PPC}) \end{array}$$

donde se utiliza la definición de alternativa para PPC introducida anteriormente.

Lema 3.2.3. *Sea* $t \in \mathbb{T}_{CAP}$*. Si* $t \to_{\mathsf{CAP}} t'$*, entonces* ${t}^{PPC} \twoheadrightarrow_{\mathsf{PPC}} {t'}^{PPC}$*.*

Al interpretar términos de PPC en CAP de debe considerar un mecanismo para capturar el manejo explícito que se hace de la falla en la operación de matching en PPC. Para esto, las abstracciones son traducidas con un caso adicional donde el patrón es trivial (*i.e.* la operación de matching siempre tiene éxito) y el cuerpo es la constante distinguida `NoMatch` (*cf.* Sec. 3.1.1).

Definición 3.2.4. *La* traducción $\_^{CAP}$ *de términos del fragmento estático de PPC en términos de CAP se define como*

$$\begin{array}{rcl} x^{CAP} & \triangleq & x \\ \mathsf{c}^{CAP} & \triangleq & \mathsf{c} \\ (r\,u)^{CAP} & \triangleq & r^{CAP}\,u^{CAP} \\ (p \to s)^{CAP} & \triangleq & p \to {s}^{CAP} \mid x \to \texttt{NoMatch} \end{array}$$

Lema 3.2.5. *Sea* $t \in \mathbb{T}_{PPC}$*. Si* $t \to_{\mathsf{PPC}} t'$*, entonces* ${t}^{CAP} \to_{\mathsf{CAP}} {t'}^{CAP}$*.*

3.3. Sistema de tipos

Esta sección introduce las expresiones de tipo utilizadas para dar semántica a los términos de CAP. De particular interés resultan los μ-tipos (o tipos recursivos), expresiones finitas que denotan árboles binarios regulares de altura infinita, y sus nociones de equivalencia y sub-tipado asociadas. Estos conceptos juegan un rol crucial a la hora de probar las buenas propiedades del sistema de tipos, pues garantizar la *invertibilidad* de la relación de sub-tipado resulta ser un paso clave para la prueba de *subject reduction* (*cf.* Sec. 3.4).

3.3.1. Expresiones de tipo

Para garantizar que patrones de la forma $x\,y$ descomponen unicamente data structures en lugar de aplicaciones arbitrarias, se separan las expresiones de tipo en categorías: los *tipos* generales y los *datatypes*, estando esta última incluida en la primera.

Dados conjuntos infinitos numerables de *variables de datatype* $\mathcal{V}_D$ $(\alpha, \beta, \ldots)$, *variables de tipo* $\mathcal{V}_A$ $(X, Y, \ldots)$ y *tipos constantes* $\mathcal{C}$ $(\mathbb{c}, \mathbb{d}, \ldots)$, los conjuntos de μ*-datatypes* $\mathcal{D}$ y μ*-tipos* $\mathcal{T}$ $(A, B, \ldots)$ se definen recursivamente como:

$$\begin{array}{rrcl} \mu\textbf{-datatypes} & D & ::= & \alpha \in \mathcal{V}_D \mid \mathbb{c} \in \mathcal{C} \mid D \mathbin{@} A \mid D \oplus D \mid \mu\alpha.D \\ \mu\textbf{-tipos} & A & ::= & X \in \mathcal{V}_A \mid D \mid A \supset A \mid A \oplus A \mid \mu X.A \end{array}$$

donde @ denota el constructor de tipos aplicativos (y una expresión de la forma $D \mathbin{@} A$ es denominada *compound*), $\supset$ el de tipos funcionales, $\oplus$ es el constructor de uniones y μ el de tipos recursivos. Se define $\mathcal{V} \triangleq \mathcal{V}_D \cup \mathcal{V}_A$ y se usan las meta-variables $V, W, \ldots$ para denotar elementos arbitrarios del conjunto. Del mismo modo, se usan $a, b, \ldots$ para elementos en $\mathcal{V} \cup \mathcal{C}$.

Nota 3.3.1. *Un tipo de la forma $\mu\alpha.A$ es excluido de la sintaxis ya que puede producir desdoblamientos inválidos (*cf. *Sec. 3.3.2). Por ejemplo, $\mu\alpha.\alpha \supset \alpha \simeq (\mu\alpha.\alpha \supset \alpha) \supset (\mu\alpha.\alpha \supset \alpha)$, donde α es una variable de datatype mientras que el tipo funcional no es un datatype. Por otro lado, tipos de la forma $\mu X.D$ no son necesarios pues denotan la solución a la ecuación $X = D$ [BDS13], por lo que X representa un datatype (*i.e. *$X = \alpha$).*

Se asume que el constructor de tipos unión $\oplus$ liga más fuerte que el funcional $\supset$, mientras que el aplicativo @ liga más fuerte que $\oplus$. Por lo tanto, $D \mathbin{@} A \oplus A' \supset B$ se interpreta como $((D \mathbin{@} A) \oplus A') \supset B$. Adicionalmente, se utiliza la notación $\bigoplus_{i \in I} A_i$ para una secuencia finita de tipos unión consecutivos $A_{i_1} \oplus \ldots \oplus A_{i_n}$, donde I puede ser una secuencia, un conjunto o un multi-conjunto finito de índices, según el contexto. Por conveniencia, cuando I es el intervalo de números naturales $[0, n)$ se escribe simplemente $\bigoplus_{i<n} A_i$. Esta notación es ambigua en tanto no explicita cómo se asocian las sub-expresiones. En la siguiente sección se presenta una relación de equivalencia sobre μ-tipos que identificará todas estas posibles asociaciones en una misma clase (*cf.* Sec. 3.3.2). Por otro lado, se escribe $\mu V.A$ para referirse indistintamente a tipos de la forma $\mu\alpha.D$ o $\mu X.A$.

Definición 3.3.2. *Un tipo* no-unión *A es un μ-tipo de alguna de las siguientes formas: α, $\mathbb{c}$, $D \mathbin{@} A'$, $A' \supset A''$ o $\mu V.B$ con B un tipo no-unión.*

Los μ-tipos se asumen *contractivos*: A es contractivo si para toda sub-expresión de la forma $\mu V.B$ en A, V ocurre en B únicamente bajo un constructor $\supset$ o @. De este modo, se re-define de aquí en adelante $\mathcal{T}$ como el conjunto de μ-tipos contractivos. Este conjunto viene equipado con nociones de equivalencia $\simeq_\mu$ y sub-tipado $\preceq_\mu$.

Definición 3.3.3 (Equivalencia y Sub-tipado para μ-tipos contractivos)**.** *La relación de* equivalencia *$\simeq_\mu$ de μ-tipos contractivos es la menor congruencia generada por las reglas de la Fig. 3.1. La relación de* sub-tipado *$\preceq_\mu$ de μ-tipos contractivos es el menor pre-orden generado por las reglas de la Fig. 3.2, donde un* contexto de sub-tipado *Σ es un conjunto de asunciones sobre variables $V, W \in \mathcal{V}$ de la forma $V \preceq_\mu W$.*

En particular los esquemas de regla (E-REC), (E-FOLD) y (E-CONTR) codifican dos reglas cada uno, una para μ-datatypes ($\mu\alpha.D$) y otra para μ-tipos ($\mu X.A$). La relación resultante de quitar (E-CONTR) [AK96, BH98] es conocida como *equivalencia débil* [Car92] y se sabe insuficiente para capturar la semántica de *árboles infinitos regulares* (*cf.* Sec. 3.3.2), necesaria para garantizar la invertibilidad de la relación de sub-tipado (Teo. 3.3.31). Por ejemplo, los tipos $\mu X.(X \supset A) \supset A$ y $\mu X.X \supset A$ no pueden ser equiparados en dicha equivalencia.

$$\frac{}{\vdash A \simeq_\mu A}\ (\text{E-REFL}) \qquad \frac{\vdash A \simeq_\mu B \quad \vdash B \simeq_\mu C}{\vdash A \simeq_\mu C}\ (\text{E-TRANS}) \qquad \frac{\vdash A \simeq_\mu B}{\vdash B \simeq_\mu A}\ (\text{E-SYMM})$$

$$\frac{\vdash D \simeq_\mu D' \quad \vdash A \simeq_\mu A'}{\vdash D \mathbin{@} A \simeq_\mu D' \mathbin{@} A'}\ (\text{E-COMP}) \qquad \frac{\vdash A \simeq_\mu A' \quad \vdash B \simeq_\mu B'}{\vdash A \supset B \simeq_\mu A' \supset B'}\ (\text{E-FUNC})$$

$$\frac{}{\vdash A \oplus A \simeq_\mu A}\ (\text{E-UNION-IDEM}) \qquad \frac{}{\vdash A \oplus B \simeq_\mu B \oplus A}\ (\text{E-UNION-COMM})$$

$$\frac{}{\vdash A \oplus (B \oplus C) \simeq_\mu (A \oplus B) \oplus C}\ (\text{E-UNION-ASSOC})$$

$$\frac{\vdash A \simeq_\mu A' \quad \vdash B \simeq_\mu B'}{\vdash A \oplus B \simeq_\mu A' \oplus B'}\ (\text{E-UNION}) \qquad \frac{\vdash A \simeq_\mu B}{\vdash \mu V.A \simeq_\mu \mu V.B}\ (\text{E-REC})$$

$$\frac{}{\vdash \mu V.A \simeq_\mu \{V \setminus \mu V.A\}A}\ (\text{E-FOLD}) \qquad \frac{\vdash A \simeq_\mu \{V \setminus A\}B \quad \mu V.B \text{ contractivo}}{\vdash A \simeq_\mu \mu V.B}\ (\text{E-CONTR})$$

Figura 3.1: Equivalencia fuerte de μ-tipos contractivos.

$$\frac{}{\Sigma; V \preceq_\mu W \vdash V \preceq_\mu W}\ (\text{S-HYP}) \qquad \frac{\vdash A \simeq_\mu B}{\Sigma \vdash A \preceq_\mu B}\ (\text{S-EQ}) \qquad \frac{}{\Sigma \vdash A \preceq_\mu A}\ (\text{S-REFL})$$

$$\frac{\Sigma \vdash A \preceq_\mu B \quad \Sigma \vdash B \preceq_\mu C}{\Sigma \vdash A \preceq_\mu C}\ (\text{S-TRANS}) \qquad \frac{\Sigma \vdash D \preceq_\mu D' \quad \Sigma \vdash A \preceq_\mu A'}{\Sigma \vdash D \mathbin{@} A \preceq_\mu D' \mathbin{@} A'}\ (\text{S-COMP})$$

$$\frac{\Sigma \vdash A \preceq_\mu A' \quad \Sigma \vdash B \preceq_\mu B'}{\Sigma \vdash A' \supset B \preceq_\mu A \supset B'}\ (\text{S-FUNC}) \qquad \frac{\Sigma \vdash A \preceq_\mu C \quad \Sigma \vdash B \preceq_\mu C}{\Sigma \vdash A \oplus B \preceq_\mu C}\ (\text{S-UNION-L})$$

$$\frac{\Sigma \vdash A \preceq_\mu B}{\Sigma \vdash A \preceq_\mu B \oplus C}\ (\text{S-UNION-R1}) \qquad \frac{\Sigma \vdash A \preceq_\mu C}{\Sigma \vdash A \preceq_\mu B \oplus C}\ (\text{S-UNION-R2})$$

$$\frac{\Sigma; V \preceq_\mu W \vdash A \preceq_\mu B \quad W \notin \mathsf{fv}(A) \quad V \notin \mathsf{fv}(B)}{\Sigma \vdash \mu V.A \preceq_\mu \mu W.B}\ (\text{S-REC})$$

Figura 3.2: Sub-tipado de μ-tipos contractivos

Respecto a la relación de sub-tipado, se adoptan las reglas para la unión de [Vou04]. Cabe destacar que una versión ingenua del esquema de regla (S-REC), en la que $\Sigma \vdash \mu V.A \preceq_\mu \mu V.B$ se deduce de $\Sigma \vdash A \preceq_\mu B$, se sabe incorrecta [AC93]. Usualmente se abrevia $\vdash A \preceq_\mu B$ como simplemente $A \preceq_\mu B$ (del mismo modo para la equivalencia). La sintaxis $\Sigma; V \preceq_\mu W$ en la regla (S-REC) denota unión disjunta.

Nota 3.3.4. *Para todo $A \in \mathcal{T}$ existen $(A_i)_{i<n} \in \mathcal{T}$ no-unión tal que $A \simeq_\mu \oplus_{i<n} A_i$. Más aún, si $A \in \mathcal{D}$, entonces $(A_i)_{i<n} \in \mathcal{D}$.*

El siguiente lema introduce un esquema de regla admisible para los tipos unión que será utilizado luego para relacionar $\simeq_\mu$ con su caracterización co-inductiva. En particular, notar que en este caso no es necesario que los tipos A_j, B_j del enunciado sean no-unión.

Lema 3.3.5. *Sean $A = \oplus_{i<n} A_i$, $B = \oplus_{j<m} B_j$ μ-tipos contractivos con funciones $f : [0,n) \to [0,m)$ y $g : [0,m) \to [0,n)$ tal que $(A_i \simeq_\mu B_{f(i)})_{i<n}$ y $(A_{g(j)} \simeq_\mu B_j)_{j<m}$. Entonces, $A \simeq_\mu B$.*

3.3.2. Invertibilidad del sub-tipado

El resultado de adecuación para el sistema de tipos buscado, estudiado en la Sec. 3.4, se basa fuertemente en el hecho de que la relación de sub-tipado $\preceq_\mu$ tiene la propiedad de ser *invertible* para tipos no-unión (*cf.* Teo. 3.3.31).

Para probar esto se apela a una interpretación estándar de μ-tipos contractivos en árboles (infinitos), y formulaciones co-inductivas de equivalencia ($\simeq_\mathfrak{T}$) y sub-tipado ($\preceq_\mathfrak{T}$) sobre los mismos, que resultan isomorfas a las introducidas para μ-tipos contractivos. La invertibilidad de $\preceq_\mathfrak{T}$ es demostrada co-inductivamente (*cf.* Lem. 3.3.19), obteniendo el Teo. 3.3.31 vía el isomorfismo.

Considerar los *constructores de tipo* @ y $\supset$ junto con el *conectivo* binario $\oplus$ y el alfabeto $\mathfrak{L} \triangleq \{a^0 \mid a \in \mathcal{V} \cup \mathcal{C}\} \cup \{@^2, \supset^2, \oplus^2\}$. Se define el conjunto de *árboles finitos* $\mathfrak{D}^{fin}$ y $\mathfrak{T}^{fin}$ sobre el alfabeto $\mathfrak{L}$ dados, respectivamente, por la siguiente gramática:

$$\begin{array}{rrcl} \textbf{Datatypes finitos} & \mathcal{D} & ::= & \alpha \in \mathcal{V}_D \mid \mathbb{c} \in \mathcal{C} \mid \mathcal{D} \mathbin{@} \mathcal{A} \mid \mathcal{D} \oplus \mathcal{D} \\ \textbf{Árboles finitos} & \mathcal{A} & ::= & X \in \mathcal{V}_A \mid \mathcal{D} \mid \mathcal{A} \supset \mathcal{A} \mid \mathcal{A} \oplus \mathcal{A} \end{array}$$

Una *posición* p es una palabra finita sobre el alfabeto $\{\mathsf{0}, \mathsf{1}\}$, denotando con ϵ a la palabra vacía. Se utilizan las posiciones para acceder a los nodos de un árbol $\mathcal{A}$ mediante *indexación*, notada $\mathcal{A}(\mathsf{p})$ y definida inductivamente se la siguiente manera:

$$\begin{array}{rcll} a(\epsilon) & \triangleq & a & \text{para } a \in \mathcal{V} \cup \mathcal{C} \\ (\mathcal{A}_0 \star \mathcal{A}_1)(\epsilon) & \triangleq & \star & \text{para } \star \in \{@, \supset, \oplus\} \\ (\mathcal{A}_0 \star \mathcal{A}_1)(i\mathsf{p}) & \triangleq & \mathcal{A}_i(\mathsf{p}) & \text{para } \star \in \{@, \supset, \oplus\} \end{array}$$

Se denomina *dominio* de un árbol $\mathsf{dom}(\mathcal{A})$ al conjunto de posiciones definidas en él. La *profundidad* de una posición $\mathsf{p} \in \mathsf{dom}(\mathcal{A})$ se define como:

$$\begin{array}{rcll} \mathit{depth}(\mathcal{A}, \epsilon) & \triangleq & 0 & \\ \mathit{depth}(\mathcal{A}_0 \star \mathcal{A}_1, i\mathsf{p}) & \triangleq & 1 + \mathit{depth}(\mathcal{A}_i, \mathsf{p}) & \text{para } \star \in \{@, \supset\} \\ \mathit{depth}(\mathcal{A}_0 \oplus \mathcal{A}_1, i\mathsf{p}) & \triangleq & \mathit{depth}(\mathcal{A}_i, \mathsf{p}) & \end{array}$$

La *distancia* entre dos árboles $\mathcal{A}$ y $\mathcal{B}$ queda luego definida como:

$$d(\mathcal{A}, \mathcal{B}) \triangleq 2^{-\min\{\mathit{depth}(\mathcal{A},\mathsf{p}) \mid \mathcal{A}(\mathsf{p}) \neq \mathcal{B}(\mathsf{p})\}}$$

donde, por convención, $\min \emptyset \triangleq \infty$ y $2^{-\infty} \triangleq 0$. Puede verificarse que d es efectivamente un función de distancia[3] entre elementos de $\mathfrak{T}^{fin}$ (y por lo tanto $\mathfrak{D}^{fin}$), por lo que $\langle \mathfrak{T}^{fin}, d \rangle$ (resp. $\langle \mathfrak{D}^{fin}, d \rangle$) constituye un espacio métrico[4]. Luego, se define el conjunto de *árboles (posiblemente) infinitos* $\mathfrak{T}$ como la compleción métrica de dicho espacio. Esta construcción es estándar [Cou83, BKvO03].

Por último, se considera un sub-conjunto propio de $\mathfrak{T}$ conocido como los árboles regulares. Dado un árbol $\mathcal{A} \in \mathfrak{T}$ y una posición $\mathsf{p} \in \mathsf{dom}(\mathcal{A})$, se define $\mathcal{A}|_{\mathsf{p}}$ como el *sub-árbol* de $\mathcal{A}$ en posición p, de manera inductiva:

$$\begin{array}{rcl} \mathcal{A}|_{\epsilon} & \triangleq & \mathcal{A} \\ \mathcal{A}_0 \star \mathcal{A}_1|_{i\mathsf{p}} & \triangleq & \mathcal{A}_i|_{\mathsf{p}} \quad \text{para } \star \in \{@, \supset, \oplus\} \end{array}$$

Luego, el conjunto de *árboles regulares* $\mathfrak{T}^{reg}$ se define como aquellos elementos de $\mathfrak{T}$ para los cuales su conjunto de sub-árboles es finito. Análogamente para $\mathfrak{D}$, se obtiene $\mathfrak{D}^{reg}$. Los árboles finitos son, por definición, regulares, por lo que se tiene entonces $\mathfrak{T}^{fin} \subset \mathfrak{T}^{reg} \subset \mathfrak{T}$ y $\mathfrak{D}^{fin} \subset \mathfrak{D}^{reg} \subset \mathfrak{D}$ respectivamente. Se utilizan las meta-variables $\mathcal{A}, \mathcal{B}, \ldots$ para detonar elementos arbitrarios de $\mathfrak{T}$. Al igual que con los μ-tipos contractivos, cuando no haya ambigüedad se llamará a los árboles infinitos simplemente tipos.

Dado un símbolo $\star \in \mathfrak{L}$, se escribe $\mathcal{A} \neq \star$ para expresar $\mathcal{A}(\epsilon) \neq \star$. Por ejemplo, $\mathcal{A} \neq \oplus$ significa que $\mathcal{A}$ es un tipo no-unión. Cabe destacar que el hecho de que el conectivo $\oplus$ no aporte a la profundidad de una posición en un árbol tiene como consecuencia que los elemento de $\mathfrak{T}$ no pueden tener ramas infinitas compuestas únicamente de nodos $\oplus$ [BKvO03].

Definición 3.3.6. *Se denomina* tipo unión-maximal *a una expresión de la forma* $\bigoplus_{i \in I} \mathcal{A}_i$ *con* $\mathcal{A}_i \neq \oplus$ *para todo* $i \in I$.

Nota 3.3.7. *Todo tipo* $\mathcal{A}$ *puede ser expresado como un tipo unión-maximal* $\mathcal{A} = \bigoplus_{i \in I} \mathcal{A}_i$*, independientemente de cómo se encuentren asociados los distintos* $\mathcal{A}_i$*. Todas estas posibles asociaciones son equivalentes para la noción introducida a continuación.*

La meta-operación de sustitución de variables en árboles se define inductivamente por posiciones:

Definición 3.3.8. *La* sustitución $\{V \setminus \mathcal{B}\}\mathcal{A}$ *de una variable* V *por un tipo (infinito)* $\mathcal{B}$ *en otro* $\mathcal{A}$ *se define como*

$$\begin{array}{rcll} (\{V \setminus \mathcal{B}\}\mathcal{A})(\mathsf{p}) & \triangleq & \mathcal{A}(\mathsf{p}) & \textit{si } \mathcal{A}(\mathsf{p}) \textit{ y } \mathcal{A}(\mathsf{p}) \neq V \\ (\{V \setminus \mathcal{B}\}\mathcal{A})(\mathsf{pq}) & \triangleq & \mathcal{B}(\mathsf{q}) & \textit{si } \mathcal{A}(\mathsf{p}) \textit{ y } \mathcal{A}(\mathsf{p}) = V \end{array}$$

El siguiente lema permite caracterizar la sustitución de una manera más conveniente, mostrando que el árbol resultante de sustituir es el esperado:

Lema 3.3.9.

1. $\{V \setminus \mathcal{B}\}V = \mathcal{B}$.
2. $\{V \setminus \mathcal{B}\}a = a$ *si* $V \neq a \in \mathcal{V} \cup \mathcal{C}$.
3. $\{V \setminus \mathcal{B}\}(\mathcal{A}_0 \star \mathcal{A}_1) = \{V \setminus \mathcal{B}\}\mathcal{A}_0 \star \{V \setminus \mathcal{B}\}\mathcal{A}_1$ *para* $\star \in \{@, \supset, \oplus\}$.

[3] Una función $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ es una *función de distancia en el dominio* $\mathcal{X}$ si $\forall x, y, z \in \mathcal{X}$ se satisface: (I) $d(x, y) \geq 0$; (II) $d(x, y) = 0$ sii $x = y$; (III) $d(x, y) = d(y, x)$; y (IV) $d(x, z) \leq d(x, y) + d(y, z)$.

[4] Un *espacio métrico* es un par ordenado $\langle \mathcal{X}, d \rangle$ donde $\mathcal{X}$ es un conjunto, llamado *dominio*, y d es una función de distancia en $\mathcal{X}$. Un espacio métrico se dice *completo* si toda secuencia de Cauchy converge.

$$\frac{}{a \simeq_{\mathfrak{T}} a}\ (\text{E-REFL-T}) \qquad \frac{\mathcal{D} \simeq_{\mathfrak{T}} \mathcal{D}' \quad \mathcal{A} \simeq_{\mathfrak{T}} \mathcal{A}'}{\mathcal{D} @ \mathcal{A} \simeq_{\mathfrak{T}} \mathcal{D}' @ \mathcal{A}'}\ (\text{E-COMP-T}) \qquad \frac{\mathcal{A} \simeq_{\mathfrak{T}} \mathcal{A}' \quad \mathcal{B} \simeq_{\mathfrak{T}} \mathcal{B}'}{\mathcal{A} \supset \mathcal{B} \simeq_{\mathfrak{T}} \mathcal{A}' \supset \mathcal{B}'}\ (\text{E-FUNC-T})$$

$$\frac{\begin{array}{lll}(\mathcal{A}_i \simeq_{\mathfrak{T}} \mathcal{B}_{f(i)})_{i<n} & f : [0,n) \to [0,m) & (\mathcal{A}_i \neq \oplus)_{i<n} \\ (\mathcal{A}_{g(j)} \simeq_{\mathfrak{T}} \mathcal{B}_j)_{j<m} & g : [0,m) \to [0,n) & (\mathcal{B}_j \neq \oplus)_{j<m}\end{array} \quad n+m>2}{\bigoplus_{i<n}\mathcal{A}_i \simeq_{\mathfrak{T}} \bigoplus_{j<m}\mathcal{B}_j}\ (\text{E-UNION-T})$$

Figura 3.3: Equivalencia de tipos infinitos.

Otra operación conveniente sobre árboles infinitos es el truncado, que permite considerar el prefijo de un árbol resultante de cortar al mismo desde su raíz hasta una profundidad dada. Al igual que antes, los nodos $\oplus$ no aportan al calculo de dicha profundidad.

Se nota con $\#_{\oplus}(\mathcal{A})$ el máximo número de nodos unión adyacentes a partir de la raíz del árbol $\mathcal{A}$:

$$\#_{\oplus}(\mathcal{A}) \triangleq \begin{cases} 0 & \text{si } \mathcal{A} \neq \oplus \\ 1 + \#_{\oplus}(\mathcal{A}_0) + \#_{\oplus}(\mathcal{A}_1) & \text{si } \mathcal{A} = \mathcal{A}_0 \oplus \mathcal{A}_1 \end{cases}$$

Recordar que, por definición de $\mathfrak{T}$, un árbol no puede consistir de infinitas ocurrencias consecutivas de $\oplus$. Por lo tanto, la definición anterior resulta bien fundada, al igual que la siguiente:

Definición 3.3.10. *El* truncado *de un tipo (infinito) $\mathcal{A}$ a profundidad k, notación $\mathcal{A}\rfloor_k$, se define inductivamente como:*

$$\begin{array}{rcll} \mathcal{A}\rfloor_0 & \triangleq & \varepsilon & \\ a\rfloor_{k+1} & \triangleq & a & \text{para } a \in \mathcal{V} \cup \mathcal{C} \\ (\mathcal{A}_0 \star \mathcal{A}_1)\rfloor_{k+1} & \triangleq & \mathcal{A}_0\rfloor_k \star \mathcal{A}_1\rfloor_k & \text{para } \star \in \{@, \supset\} \\ (\mathcal{A}_0 \oplus \mathcal{A}_1)\rfloor_{k+1} & \triangleq & \mathcal{A}_0\rfloor_{k+1} \oplus \mathcal{A}_1\rfloor_{k+1} & \end{array}$$

donde $\varepsilon \in \mathcal{C}$ es una constante de tipo distinguida para identificar los nodos donde se produjo un corte.

Nota 3.3.11. *Dado un tipo unión-maximal $\bigoplus_{i<n}\mathcal{A}_i$, se tienen $(\bigoplus_{i<n}\mathcal{A}_i)\rfloor_{k+1} = \bigoplus_{i<n}(\mathcal{A}_i\rfloor_{k+1})$.*

Equivalencia de tipos infinitos

Definición 3.3.12. *La relación de* equivalencia $\simeq_{\mathfrak{T}}$ *de tipos infinitos se define como la interpretación co-inductiva de los esquemas de regla de la Fig. 3.3.*

Notar que (E-UNION-T) es en realidad un esquema de regla, representando todas las posibles asociaciones dentro de los tipos unión-maximal $\mathcal{A} = \bigoplus_{i<n}\mathcal{A}_i$ y $\mathcal{B} = \bigoplus_{j<m}\mathcal{B}_j$. Cada instancia del esquema expresa que cada $\mathcal{A}_i$ debe ser equivalente a algún $\mathcal{B}_j$ vía la función $f : [0,n) \to [0,m)$ y vice versa (con $g : [0,m) \to [0,n)$ respectivamente). Notar que este esquema implica que el conectivo $\oplus$ no solo es asociativo, sino también conmutativo e idempotente.

Formalmente, sea $\Phi_{\simeq_{\mathfrak{T}}} : \wp(\mathfrak{T} \times \mathfrak{T}) \to \wp(\mathfrak{T} \times \mathfrak{T})$ la función generadora asociada a las reglas de la Fig. 3.3, definido como

$$\begin{array}{rcl} \Phi_{\simeq_{\mathfrak{T}}}(\mathcal{X}) & \triangleq & \{\langle a, a\rangle \mid a \in \mathcal{V} \cup \mathcal{C}\} \\ & \cup & \{\langle \mathcal{D} @ \mathcal{A}, \mathcal{D}' @ \mathcal{A}'\rangle \mid \langle \mathcal{D}, \mathcal{D}'\rangle, \langle \mathcal{A}, \mathcal{A}'\rangle \in \mathcal{X}\} \\ & \cup & \{\langle \mathcal{A} \supset \mathcal{B}, \mathcal{A}' \supset \mathcal{B}'\rangle \mid \langle \mathcal{A}, \mathcal{A}'\rangle, \langle \mathcal{B}, \mathcal{B}'\rangle \in \mathcal{X}\} \\ & \cup & \{\langle \bigoplus_{i<n}\mathcal{A}_i, \bigoplus_{j<m}\mathcal{B}_j\rangle \mid \mathcal{A}_i, \mathcal{B}_j \neq \oplus, n+m>2 \\ & & \quad \exists f : [0,n) \to [0,m).\langle \mathcal{A}_i, \mathcal{B}_{f(i)}\rangle \in \mathcal{X}, \\ & & \quad \exists g : [0,m) \to [0,n).\langle \mathcal{A}_{g(j)}, \mathcal{B}_j\rangle \in \mathcal{X}\} \end{array}$$

$$\frac{}{a \preceq_{\mathfrak{T}} a}\ (\text{S-REFL-T}) \qquad \frac{\mathcal{D} \preceq_{\mathfrak{T}} \mathcal{D}' \quad \mathcal{A} \preceq_{\mathfrak{T}} \mathcal{A}'}{\mathcal{D} \mathbin{@} \mathcal{A} \preceq_{\mathfrak{T}} \mathcal{D}' \mathbin{@} \mathcal{A}'}\ (\text{S-COMP-T}) \qquad \frac{\mathcal{A}' \preceq_{\mathfrak{T}} \mathcal{A} \quad \mathcal{B} \preceq_{\mathfrak{T}} \mathcal{B}'}{\mathcal{A} \supset \mathcal{B} \preceq_{\mathfrak{T}} \mathcal{A}' \supset \mathcal{B}'}\ (\text{S-FUNC-T})$$

$$\frac{(\mathcal{A}_i \preceq_{\mathfrak{T}} \mathcal{B}_{f(i)})_{i<n} \quad f : [0,n) \to [0,m) \quad (\mathcal{A}_i \neq \oplus)_{i<n}, (\mathcal{B}_j \neq \oplus)_{j<m} \quad n+m>2}{\bigoplus_{i<n} \mathcal{A}_i \preceq_{\mathfrak{T}} \bigoplus_{j<m} \mathcal{B}_j}\ (\text{S-UNION-T})$$

Figura 3.4: Sub-tipado de tipos infinitos.

Luego, $\simeq_{\mathfrak{T}} \triangleq \nu\Phi_{\simeq_{\mathfrak{T}}}$. Puede verse que esto define efectivamente una relación de equivalencia. Más aún, tal relación resulta invertible para tipos no-unión. Los detalles de éstas propiedades pueden encontrarse en el Ap. B.

Lema 3.3.13. *$\simeq_{\mathfrak{T}}$ es una relación de equivalencia (*i.e. *reflexiva, simétrica y transitiva).*

Lema 3.3.14. *Sean $\mathcal{A}, \mathcal{B} \in \mathfrak{T}$ tipos no-unión tal que $\mathcal{A} \simeq_{\mathfrak{T}} \mathcal{B}$.*

1. *Si $\mathcal{A} = a$, entonces $\mathcal{B} = a$.*
2. *Si $\mathcal{A} = \mathcal{D} \mathbin{@} \mathcal{A}'$, entonces $\mathcal{B} = \mathcal{D}' \mathbin{@} \mathcal{B}'$ con $\mathcal{D} \simeq_{\mathfrak{T}} \mathcal{D}'$ y $\mathcal{A}' \simeq_{\mathfrak{T}} \mathcal{B}'$.*
3. *Si $\mathcal{A} = \mathcal{A}' \supset \mathcal{A}''$, entonces $\mathcal{B} = \mathcal{B}' \supset \mathcal{B}''$ con $\mathcal{A}' \simeq_{\mathfrak{T}} \mathcal{B}'$ y $\mathcal{A}'' \simeq_{\mathfrak{T}} \mathcal{B}''$.*

Por otro lado, la sustitución de tipos (infinitos) equivalentes resulta en expresiones a su vez equivalentes.

Lema 3.3.15. *Sean $\mathcal{A} \simeq_{\mathfrak{T}} \mathcal{A}'$ y $\mathcal{B} \simeq_{\mathfrak{T}} \mathcal{B}'$. Luego, $\{V \setminus \mathcal{B}\}\mathcal{A} \simeq_{\mathfrak{T}} \{V \setminus \mathcal{B}'\}\mathcal{A}'$.*

Por último, resulta conveniente la caracterización de la equivalencia mediante la comparación de los truncados finitos para toda profundidad posible.

Lema 3.3.16. *$\mathcal{A} \simeq_{\mathfrak{T}} \mathcal{B}$ sii $\forall k \in \mathbb{N}. \mathcal{A}\rfloor_k \simeq_{\mathfrak{T}} \mathcal{B}\rfloor_k$.*

Sub-tipado de tipos infinitos

Del mismo modo, se tiene una caracterización co-inductiva del sub-tipado para tipos infinitos.

Definición 3.3.17. *La relación de* sub-tipado *$\preceq_{\mathfrak{T}}$ de tipos infinitos se define como la interpretación co-inductiva de los esquemas de regla de la Fig. 3.4.*

Aquí se destaca la regla (S-UNION-T), donde un tipo unión-maximal $\oplus_{i<n}\mathcal{A}_i$ es sub-tipo de otro unión-maximal $\oplus_{j<m}\mathcal{B}_j$ si al menos uno de los dos tiene una ocurrencia del conectivo $\oplus$ (*i.e.* $n+m>2$) y existe una función $f : [0,n) \to [0,m)$ tal que $\mathcal{A}_i \preceq_{\mathfrak{T}} \mathcal{B}_{f(i)}$ para todo $i < n$. Cabe destacar que las reglas se derivan de aquellas en la Fig. 3.2. Más precisamente, (S-UNION-T) surge de (S-UNION-R1), (S-UNION-R2), (S-UNION-L) y el hecho de que siempre pueden permutarse (S-UNION-R1) y (S-UNION-R2) con (S-UNION-L).

Al igual que para la equivalencia, puede definirse formalmente la relación de sub-tipado mediante la función generadora $\Phi_{\preceq_{\mathfrak{T}}} : \wp(\mathfrak{T} \times \mathfrak{T}) \to \wp(\mathfrak{T} \times \mathfrak{T})$, asociada a las reglas de la Fig. 3.4

$$\begin{array}{rcl}
\Phi_{\preceq_{\mathfrak{T}}}(\mathcal{X}) & \triangleq & \{\langle a, a\rangle \mid a \in \mathcal{V} \cup \mathcal{C}\} \\
& \cup & \{\langle \mathcal{D} \mathbin{@} \mathcal{A}, \mathcal{D}' \mathbin{@} \mathcal{A}'\rangle \mid \langle \mathcal{D}, \mathcal{D}'\rangle, \langle \mathcal{A}, \mathcal{A}'\rangle \in \mathcal{X}\} \\
& \cup & \{\langle \mathcal{A} \supset \mathcal{B}, \mathcal{A}' \supset \mathcal{B}'\rangle \mid \langle \mathcal{A}', \mathcal{A}\rangle, \langle \mathcal{B}, \mathcal{B}'\rangle \in \mathcal{X}\} \\
& \cup & \{\langle \oplus_{i<n}\mathcal{A}_i, \oplus_{j<m}\mathcal{B}_j\rangle \mid \mathcal{A}_i, \mathcal{B}_j \neq \oplus, n+m>2 \\
& & \quad \exists f : [0,n) \to [0,m). \langle \mathcal{A}_i, \mathcal{B}_{f(i)}\rangle \in \mathcal{X}\}
\end{array}$$

Luego, $\preceq_{\mathfrak{T}} \triangleq \nu\Phi_{\preceq_{\mathfrak{T}}}$. A continuación se presentan algunas propiedades de la relación de sub-tipado.

Lema 3.3.18. *$\preceq_{\mathfrak{T}}$ es un pre-orden (*i.e. *reflexivo y transitivo).*

Al igual que la equivalencia presentada anteriormente, la relación de sub-tipado para tipos infinitos es invertible en el caso de los tipos no-unión.

Lema 3.3.19. *Sean $\mathcal{A}, \mathcal{B} \in \mathfrak{T}$ tipos no-unión tal que $\mathcal{A} \preceq_{\mathfrak{T}} \mathcal{B}$.*

1. *$\mathcal{A} = a$ sii $\mathcal{B} = a$.*
2. *$\mathcal{A} = \mathcal{D} \mathbin{@} \mathcal{A}'$ sii $\mathcal{B} = \mathcal{D}' \mathbin{@} \mathcal{B}'$, con $\mathcal{D} \preceq_{\mathfrak{T}} \mathcal{D}'$ y $\mathcal{A}' \preceq_{\mathfrak{T}} \mathcal{B}'$.*
3. *$\mathcal{A} = \mathcal{A}' \supset \mathcal{A}''$ sii $\mathcal{B} = \mathcal{B}' \supset \mathcal{B}''$, con $\mathcal{B}' \preceq_{\mathfrak{T}} \mathcal{A}'$ y $\mathcal{A}'' \preceq_{\mathfrak{T}} \mathcal{B}''$.*

Adicionalmente puede verse que la relación de sub-tipado subsume a la de equivalencia.

Lema 3.3.20. *Si $\mathcal{A} \simeq_{\mathfrak{T}} \mathcal{B}$, entonces $\mathcal{A} \preceq_{\mathfrak{T}} \mathcal{B}$.*

Lema 3.3.21. *$\mathcal{A} \preceq_{\mathfrak{T}} \mathcal{B}$ sii $\forall k \in \mathbb{N}. \mathcal{A}\rfloor_k \preceq_{\mathfrak{T}} \mathcal{B}\rfloor_k$.*

Correspondencia con μ-tipos contractivos

El objetivo de esta sección es probar la correspondencia entre las relaciones de equivalencia y sub-tipado sobre tipos infinitos y sus contra-partes definidas anteriormente para μ-tipos contractivos. De este modo, se concluye la invertibilidad de la relación $\simeq_\mu$, una propiedad clave para garantizar al adecuación del sistema de tipos a definir.

Los μ-tipos contractivos caracterizan [Cou83, AC93, BH98, Pie02] un sub-conjunto propio de $\mathfrak{T}$, conocido como *árboles regulares* ($\mathfrak{T}^{reg}$): aquellos árboles cuyos conjunto de sub-árboles distintos es finito. Dado un μ-tipo contractivo A, $[\![A]\!]^{\mathfrak{T}}$ denota el árbol regular obtenido al desdoblar completamente todas las ocurrencias de $\mu V.B$ en A. La Def. 3.3.22 a continuación extiende la dada en [Pie02] incorporando tipos aplicativos y unión. Puede verificarse su buena fundación apelando al orden lexicográfico $\langle |\mathsf{p}|, \#_\mu(A)\rangle$, donde $\#_\mu(A)$ es el número de ocurrencias del constructor de tipos μ a la cabeza del tipo A. Notar que la contractividad de los μ-tipos garantiza $\#_\mu(\mu V.A) > \#_\mu(\{V \setminus \mu V.A\}A)$.

Definición 3.3.22. *La* traducción *de μ-tipos contractivos a tipos infinitos regulares $[\![\_]\!]^{\mathfrak{T}} : \mathcal{T} \to \mathfrak{T}^{reg}$, se define inductivamente de la siguiente manera:*

$$
\begin{array}{rcll}
[\![a]\!]^{\mathfrak{T}}(\epsilon) & \triangleq & a & \text{para } a \in \mathcal{V} \cup \mathcal{C} \\
[\![A_0 \star A_1]\!]^{\mathfrak{T}}(\epsilon) & \triangleq & \star & \text{para } \star \in \{@, \supset, \oplus\} \\
[\![A_0 \star A_1]\!]^{\mathfrak{T}}(i\mathsf{p}) & \triangleq & [\![A_i]\!]^{\mathfrak{T}}(\mathsf{p}) & \text{para } \star \in \{@, \supset, \oplus\} \\
[\![\mu V.A]\!]^{\mathfrak{T}}(\mathsf{p}) & \triangleq & [\![\{V \setminus \mu V.A\}A]\!]^{\mathfrak{T}}(\mathsf{p}) &
\end{array}
$$

La conmutación entre la traducción y las sustituciones es la esperada.

Lema 3.3.23. $[\![\{V \setminus B\}A]\!]^{\mathfrak{T}} = \{V \setminus [\![B]\!]^{\mathfrak{T}}\}[\![A]\!]^{\mathfrak{T}}$.

El desdoblamiento finito de un μ-tipo contractivo $A = \mu V.A'$ consiste en reemplazar recursivamente todas las ocurrencias libres de V en A' por A mismo, una cantidad finita de veces. Este resulta ser una herramienta útil para demostrar la correspondencia entre las relaciones de equivalencia (sub-tipado) vía la traducción definida anteriormente. Formalmente:

Definición 3.3.24. *El* desdoblamiento finito *de un μ-tipo contractivo $A = \mu V.A'$ relativo a un μ-tipo contractivo B y una sustitución σ, notado $A^k_{B,\sigma}$, se define como:*

$$A^0_{B,\sigma} \triangleq B \qquad A^{k+1}_{B,\sigma} \triangleq (\sigma \uplus \{V \setminus A^k_{B,\sigma}\})A'$$

Lema 3.3.25. *Sean $A = \mu V.A'$ y B μ-tipos contractivos, y σ sustitución. Luego, $\forall k \in \mathbb{N}.\lfloor[\![A^k_{B,\sigma}]\!]^{\mathfrak{T}}\rfloor_k \simeq_{\mathfrak{T}} \lfloor[\![\sigma A]\!]^{\mathfrak{T}}\rfloor_k$.*

Nota 3.3.26. *Se sigue inmediatamente del resultado anterior que para todo $n \geq k$, $\lfloor[\![A^n_{B,\sigma}]\!]^{\mathfrak{T}}\rfloor_k \simeq_{\mathfrak{T}} \lfloor[\![\sigma A]\!]^{\mathfrak{T}}\rfloor_k$.*

Uno de los resultados principales de la presente sección es la correspondencia entre las relaciones $\simeq_\mu$ y $\simeq_{\mathfrak{T}}$ vía la función $[\![\_]\!]^{\mathfrak{T}}$. Para lograrlo se apela al siguiente lema, que relaciona dos μ-tipos equivalentes con los truncados arbitrarios de sus respectivos árboles:

Lema 3.3.27. *$A \simeq_\mu B$ sii $\forall k \in \mathbb{N}.\lfloor[\![A]\!]^{\mathfrak{T}}\rfloor_k \simeq_{\mathfrak{T}} \lfloor[\![B]\!]^{\mathfrak{T}}\rfloor_k$.*

Teorema 3.3.28. *$A \simeq_\mu B$ sii $[\![A]\!]^{\mathfrak{T}} \simeq_{\mathfrak{T}} [\![B]\!]^{\mathfrak{T}}$.*

Demostración. El resultado es consecuencia inmediata de los Lem. 3.3.16 y 3.3.27: $A \simeq_\mu B$ sii $\forall k \in \mathbb{N}.\lfloor[\![A]\!]^{\mathfrak{T}}\rfloor_k \simeq_{\mathfrak{T}} \lfloor[\![B]\!]^{\mathfrak{T}}\rfloor_k$ sii $[\![A]\!]^{\mathfrak{T}} \simeq_{\mathfrak{T}} [\![B]\!]^{\mathfrak{T}}$. □

Para demostrar la correspondencia entre las relaciones de sub-tipado se requiere verificar primero que todas las asunciones sobre variables en el contexto de sub-tipado pueden ser sustituidas por μ-tipos contractivos convenientes previo a la aplicación de $[\![\_]\!]^{\mathfrak{T}}$.

Lema 3.3.29. *Sean $\Sigma = \{V_i \preceq_\mu W_i\}_{i<n}$ un contexto de sub-tipado y σ una sustitución tal que $\mathsf{dom}(\sigma) = \{V_i, W_i\}_{i<n}$, $\sigma(V_i) = A_i$ y $\sigma(W_i) = B_i$ con $\mathsf{dom}(\sigma) \cap \mathsf{fv}(\{A_i, B_i\}_{i<n}) = \emptyset$, $[\![A_i]\!]^{\mathfrak{T}} \preceq_{\mathfrak{T}} [\![B_i]\!]^{\mathfrak{T}}$ y $A_i, B_i \in \mathcal{T}$ para todo $i < n$. Si $\Sigma \vdash A \simeq_\mu B$, entonces $[\![\sigma A]\!]^{\mathfrak{T}} \simeq_{\mathfrak{T}} [\![\sigma B]\!]^{\mathfrak{T}}$.*

Finalmente, se demuestra la correspondencia entre $\preceq_\mu$ y $\preceq_{\mathfrak{T}}$ vía la función $[\![\_]\!]^{\mathfrak{T}}$, apelando al Lem. 3.3.29.

Teorema 3.3.30. *$A \preceq_\mu B$ sii $[\![A]\!]^{\mathfrak{T}} \preceq_{\mathfrak{T}} [\![B]\!]^{\mathfrak{T}}$.*

Demostración. La ida es inmediata por Lem. 3.3.29 con la sustitución identidad. Para la vuelta se apela al truncado de tipos infinitos, al estilo del Lem. 3.3.27 para equivalencia, y se concluye por Lem. 3.3.21. Detalles en el Ap. B. □

Por último, se tiene la invertibilidad de la relación de sub-tipado $\simeq_\mu$ para tipos no-unión. Un resultado clave para el desarrollo de la prueba de adecuación del sistema $\mathcal{P}$.

Teorema 3.3.31. *Sean $A, B \in \mathcal{T}$ tipos no-unión tal que $A \preceq_\mu B$.*

1. *$A = a$ sii $B = a$.*
2. *$A = D \mathbin{@} A'$ sii $B = D' \mathbin{@} B'$, con $D \preceq_\mu D'$ y $A' \preceq_\mu B'$.*
3. *$A = A' \supset A''$ sii $B = B' \supset B''$, con $B' \preceq_\mu A'$ y $A'' \preceq_\mu B''$.*

Demostración. El resultado es inmediato por Lem. 3.3.19 y Teo. 3.3.30. □

Reglas de tipado para patrones

$$\frac{}{x : A \vdash_{\mathsf{p}} x : A}\ (\textsc{p-match}) \qquad \frac{}{\vdash_{\mathsf{p}} \mathsf{c} : \mathbb{c}}\ (\textsc{p-const}) \qquad \frac{\theta \vdash_{\mathsf{p}} p : D \quad \theta' \vdash_{\mathsf{p}} q : A}{\theta;\theta' \vdash_{\mathsf{p}} p\, q : D \mathbin{@} A}\ (\textsc{p-comp})$$

Reglas de tipado para términos

$$\frac{\Gamma(x) = A}{\Gamma \vdash x : A}\ (\textsc{t-var}) \qquad \frac{}{\Gamma \vdash \mathsf{c} : \mathbb{c}}\ (\textsc{t-const}) \qquad \frac{\Gamma \vdash r : D \quad \Gamma \vdash u : A}{\Gamma \vdash r\, u : D \mathbin{@} A}\ (\textsc{t-comp})$$

$$\frac{(\theta_i \vdash_{\mathsf{p}} p_i : A_i)_{i<n} \quad \mathsf{cmp}([p_i : A_i]_{i<n}) \quad (\Gamma;\theta_i \vdash s_i : B)_{i<n}}{\Gamma \vdash (p_i \to_{\theta_i} s_i)_{i<n} : \bigoplus_{i<n} A_i \supset B}\ (\textsc{t-abs})$$

$$\frac{\Gamma \vdash r : \bigoplus_{i<n} A_i \supset B \quad \Gamma \vdash u : A_k \quad k < n}{\Gamma \vdash r\, u : B}\ (\textsc{t-app}) \qquad \frac{\Gamma \vdash s : A \quad \vdash A \preceq_\mu A'}{\Gamma \vdash s : A'}\ (\textsc{t-subs})$$

Figura 3.5: Reglas de tipado del sistema $\mathcal{P}$ para CAP.

Propiedades adicionales de los μ-tipos contractivos

Los siguientes resultados muestran que μ-tipos unión-maximal en relación de equivalencia o subtipado pueden ser descompuestos, al igual que los tipos infinitos, siguiendo la lógica de las reglas (E-UNION-T) y (S-UNION-T) respectivamente. Estas propiedades resultan cruciales a la hora de probar la adecuación del sistema de tipos.

Lema 3.3.32. *Sean $(A_i)_{i<n}, (B_j)_{j<m} \in \mathcal{T}$ tipos no-unión. Luego, $\bigoplus_{i<n} A_i \simeq_\mu \bigoplus_{j<m} B_j$ sii existen funciones $f : [0,n) \to [0,m)$ y $g : [0,m) \to [0,n)$ tales que $(A_i \simeq_\mu B_{f(i)})_{i<n}$ y $(A_{g(j)} \simeq_\mu B_j)_{j<m}$.*

Lema 3.3.33. *Sean $(A_i)_{i<n} \in \mathcal{T}$ tipos no-unión. Luego, $\bigoplus_{i<n} A_i \preceq_\mu \bigoplus_{j<m} B_j$ sii existe una función $f : [0,n) \to [0,m)$ tal que $(A_i \preceq_\mu B_{f(i)})_{i<n}$.*

3.3.3. Reglas de tipado

El sistema de tipos $\mathcal{P}$ para CAP introducido a continuación consta de dos clases de juicios de tipado. Por un lado se denota de manera tradicional los juicios para términos, $\Gamma \vdash t : A$, y por otro se usan juicios de tipado especiales para patrones, $\theta \vdash_{\mathsf{p}} p : A$. Además, se extiende la noción a sustituciones: $\Gamma \vdash \sigma : \theta$ sii $\mathsf{dom}(\sigma) = \mathsf{dom}(\theta)$ y para toda $x \in \mathsf{dom}(\sigma)$ vale $\Gamma \vdash \sigma(x) : \theta(x)$. Las reglas del sistema $\mathcal{P}$ se muestran en la Fig. 3.5, donde se tienen conjuntos de reglas distintos para patrones (arriba) y términos (abajo). Se utiliza la notación $\rhd_{\mathcal{P}} \Gamma \vdash t : A$ para expresar que el juicio de tipado $\Gamma \vdash t : A$ es derivable en $\mathcal{P}$ (análogamente para $\rhd_{\mathcal{P}} \theta \vdash_{\mathsf{p}} p : A$). Un término se dice *$\mathcal{P}$-tipable* en el sistema si existe una derivación que lo tiene como *sujeto*, *i.e.* existen Γ y A tal que $\pi \rhd_{\mathcal{P}} \Gamma \vdash t : A$, siendo π el nombre de la derivación.

Las reglas para patrones son simples e intuitivas: buscan replicar en el tipo la estructura del patrón: *i.e.* si $\rhd_{\mathcal{P}} \theta \vdash_{\mathsf{p}} p : A$, entonces $\mathsf{pos}(p) \subseteq \mathsf{pos}(A)$. Cabe destacar, sin embargo, que las variables en el contexto de tipado son manipuladas de manera lineal: *i.e.* $\mathsf{dom}(\theta) = \mathsf{fv}(p)$ siempre que $\rhd_{\mathcal{P}} \theta \vdash_{\mathsf{p}} p : A$ (notar la unión disjunta de contextos en la regla (P-COMP)). Esto garantiza la linealidad de los patrones tipables, una característica necesaria para el buen comportamiento de la operación de matching (*cf.* Sec. 3.2).

Respecto a las reglas de tipado para términos, si bien no se cuenta con reglas explícitas de *weakening* y *strengthening*, estas son admisibles en el sistema (*cf.* Lem. 3.3.50). De particular interés es la regla

(T-ABS) y el hecho de que se tienen dos reglas para la aplicación: (T-COMP) y (T-APP). En cuanto a estas últimas, la primera busca tipar aplicaciones que no generan redexes, sino que pertenecen a estructuras de datos y pueden, eventualmente, ser descompuestas por la operación de match; mientras que la segunda requiere un tipo funcional para el sub-término izquierdo, constituyendo potencialmente un redex. Cabe destacar el hecho de que en (T-APP) los A_i no necesariamente son tipos no-unión, permitiendo así tipar ejemplos como

$$(\mathtt{true} \to 1 \mid \mathtt{false} \to 0)\,((\mathtt{true} \to \mathtt{false} \mid \mathtt{false} \to \mathtt{true})\,\mathtt{true})$$

donde 0 y 1 son abreviaturas para `zero` y `suc zero` respectivamente. En este caso, la instancia más externa de (T-APP) es con $n = 1$ y $A_0 = \mathsf{Bool} = \mathbb{true} \oplus \mathbb{false}$. En cuanto a (T-ABS), requiere una serie de condiciones. En primer lugar, cada patrón p_i debe ser tipable bajo un contexto θ_i propio. Por otro lado, el cuerpo s_i de cada rama debe ser tipable bajo el contexto Γ extendido con el correspondiente θ_i. Más notable es la condición $\mathsf{cmp}([p_i : A_i]_{i<n})$: *i.e.* que la lista $[p_i : A_i]_{i<n}$ sea *compatible*, una noción desarrollada para garantizar el buen comportamiento dinámico de sistema, la cual es discutida en la Sec 3.3.4. Los contextos θ_i son introducidos en la sintaxis de las abstracciones para llevar registro explícito de los tipos asignados a las variables abstraídas. Esto resulta en una sintaxis muy similar a la original de PPC, pero donde los θ_i deben interpretarse como conjuntos de asignaciones de la forma $x : A$ en lugar de simples conjuntos de variables (símbolos). En particular, si $\mathsf{fv}(p) = \emptyset$ se escribe simplemente $p \to s$.

3.3.4. Compatibilidad

El objetivo de esta sección es estudiar la definición del predicado $\mathsf{cmp}(\_)$ presente en la regla (T-ABS): *i.e.* establecer las condiciones necesaria para garantizar que, dada una abstracción de la forma $(p_i \to_{\theta_i} s_i)_{i<n}$, un argumento destinado a una rama k no sea accidentalmente capturado por otra rama $l \neq k$. Como consideración preliminar, el hecho de que la regla de reducción $\mapsto_\beta$ evalúe los patrones de izquierda a derecha garantiza que todo potencial l conflictivo satisface $l < k$ (*i.e.* basta analizar las ramas precedentes). Dicho en otros términos, se busca establecer una noción de compatibilidad entre los patrones de una abstracción y sus tipos asignados de modo que, de haber solapamiento entre las ramas, los respectivos tipos de los patrones estén adecuadamente relacionados para garantizar que el tipo general de una expresión se preserve al reducir.

Se dice que un patrón p *subsume* a otro q, notado $p \lhd q$, si existe una sustitución σ tal que $\sigma p = q$. Para ilustrar los conceptos detrás de la noción de compatibilidad, considerar la abstracción binaria $(p \to_\theta s \mid q \to_{\theta'} t)$ y los juicios de tipado $\theta \vdash_{\mathsf{p}} p : A$ y $\theta' \vdash_{\mathsf{p}} q : B$. A continuación se analizan las distintas posibilidades según si p subsume a q o no, para luego generalizar el análisis a una cantidad arbitraria de ramas.

Si p subsume a q, en particular, todo argumento de la rama $q \to_{\theta'} t$ será capturado por $p \to_\theta s$. En efecto, dado $u : B$ en matchable form, $\{\!\{p \setminus u\}\!\}$ será exitoso, reduciendo a $\{\!\{p \setminus u\}\!\}\, s$ y quedando la rama $q \to_{\theta'} t$ inutilizable. Puesto que para tipar s se asume $\theta \vdash_{\mathsf{p}} p : A$, esto lleva a imponer la condición $B \preceq_\mu A$ para evitar situaciones como

$$(x \to_{\{x:\mathsf{Nat}\}} x \mid y \to_{\{y:\mathsf{Bool}\}} 0)\,\mathtt{true} \to_\beta \mathtt{true}$$

Si p no subsume a q (*i.e.* $p \not\lhd q$), se analiza la causa por la que esto sucede para poder determinar si es necesario imponer condiciones adicionales sobre A y B. En algunos casos no lo es, por ejemplo

$$f \to_{\{f:C \supset C'\}} \begin{array}{l} (\ \mathtt{c}\,z \to_{\{z:C\}} \mathtt{c}\,(f\,z) \\ \mid\ \mathtt{d}\,y \to_{\{y:E\}} \mathtt{d}\,y \quad\ ) \end{array} \tag{3.5}$$

No es necesario establecer una relación entre $A = \mathtt{c} \mathbin{@} C$ y $B = \mathtt{d} \mathbin{@} E$ puesto que las ramas son mutuamente excluyentes. Sin embargo, hay situaciones que requieren un análisis más refinado, como

$$f \to_{\{f:C \supset C'\}} \begin{array}{ll} (\ \mathtt{c}\, z & \to_{\{z:C\}} \ \mathtt{c}\,(f\, z) \\ \mid \ x\, y & \to_{\{x:D, y:E\}} \ x\, y \quad) \end{array} \tag{3.6}$$

donde, en particular, $q \lhd p$; o en

$$f \to_{\{f:C \supset C'\}} \begin{array}{ll} (\ \mathtt{c}\, z & \to_{\{z:C\}} \ \mathtt{c}\,(f\, z) \\ \mid \ x\, \mathtt{e} & \to_{\{x:D\}} \ x\, \mathtt{e} \quad) \end{array}$$

donde $q \not\lhd p$. El problema en estos últimos dos casos se presenta si se permite que x tome el valor $\mathtt{c}$. En término de tipos, si $\mathtt{c} \preceq_\mu D$ entonces se quiere $B = D \mathbin{@} E \preceq_\mu \mathtt{c} \mathbin{@} C = A$ (ídem para $B = D \mathbin{@} \mathtt{e}$). Sin embargo, esta condición no siempre se puede expresar en términos de sub-tipado. Por ejemplo,

$$f \to_{\{f:C \supset C'\}} \begin{array}{ll} (\ x\, y & \to_{\{x:D, y:C\}} \ f\, y \\ \mid \ z & \to_{\{z:C'\}} \ z \quad) \end{array} \tag{3.7}$$

donde puede darse $C' \simeq_\mu D' \mathbin{@} E$, siendo necesario pedir $B = C' \preceq_\mu A$ sin que se de necesariamente $D \preceq_\mu D'$ o $C \preceq_\mu E$ (*i.e.* el simple hecho de ser C' equivalente a un tipo aplicativo implica imponer la condición $B \preceq_\mu A$, independientemente de la relación entre sus partes). Por otro lado, notar que no sería correcto imponer siempre la condición $B \preceq_\mu A$: por ejemplo, si $C' = C_0 \supset C_1$, un argumento para la segunda rama es (en principio) una abstracción, en cuyo caso no es necesario imponer restricciones adicionales.

Para formalizar este criterio, se asume $p \not\lhd q$ y se analizan las posiciones de mismatch entre los patrones (*cf.* Def. 3.3.34), relacionándolas con los constructores admitidos (*cf.* Def. 3.3.41) en la respectiva posición de sus tipos asociados A y B. Recordar que, por construcción, el tipado de patrones replica en el tipo la estructura del patrón. La notación $t|_{\mathsf{p}}$ denota el sub-término en posición p de t (*cf.* Sec. 1.2.1), extendido a patrones de CAP de forma inmediata puesto que estos se componen únicamente de átomos y aplicaciones.

Definición 3.3.34. *El sub-conjunto de* posiciones maximales *de un conjunto de posiciones P se define como:*

$$\mathsf{maxpos}(P) \triangleq \{\mathsf{p} \in P \mid \nexists \mathsf{q} \in P. \mathsf{q} = \mathsf{pp}', \mathsf{p}' \neq \epsilon\}$$

El conjunto de posiciones de mismatch *entre dos patrones se define como:*

$$\mathsf{mmpos}(p, q)\{\mathsf{p} \mid \mathsf{p} \in \mathsf{maxpos}(\mathsf{pos}(p) \cap \mathsf{pos}(q)), p|_{\mathsf{p}} \not\lhd q|_{\mathsf{p}}\}$$

En particular, si $p \lhd q$, entonces $\mathsf{mmpos}(p, q) = \emptyset$. Por otro lado, si $p \not\lhd q$, las posiciones de mismatch entre p y q corresponden a nodos hoja en el árbol sintáctico de alguno de los dos patrones.

Sea p una posición de mismatch entre p y q, la siguiente tabla ilustra las distintas situaciones de interés:

	$p\|_{\mathsf{p}}$	$q\|_{\mathsf{p}}$	
(a)	$\mathtt{c}$	y	puede requerir restricción
(b)	$\mathtt{c}$	$\mathtt{d}$	no hay solapamiento ($q \not\lhd p$)
(c)	$\mathtt{c}$	$q_0\, q_1$	no hay solapamiento
(d)	$p_0\, p_1$	y	puede requerir restricción
(e)	$p_0\, p_1$	$q_0\, q_1$	no hay solapamiento

En los casos (b), (c) y (e) no es necesario imponer condiciones adicionales sobre los tipos de p y q porque los respectivos conjuntos de argumentos posibles son disjuntos (*cf.* ejemplo (3.5)). Los casos en donde deben examinarse A y B para imponer una condición (de ser necesario) son (a) y (d), ejemplificados en (3.6) y (3.7) respectivamente.

Se introducen a continuación algunas definiciones y propiedades auxiliares que respaldan la formalización de la noción de compatibilidad propuesta.

Definición 3.3.35. *El conjunto de* posiciones inspeccionables *de un tipo infinito $\mathcal{A}$, notado* $\mathsf{ipos}(\mathcal{A})$*, se define como:*

$$\begin{array}{rcll} \mathsf{ipos}(a) & \triangleq & \{\epsilon\} & \text{para } a \in \mathcal{V} \cup \mathcal{C} \\ \mathsf{ipos}(\mathcal{A}_0 \mathbin{@} \mathcal{A}_1) & \triangleq & \{\epsilon\} \cup \left(\bigcup_{i\in[0,1]} \{i\mathsf{p} \mid \mathsf{p} \in \mathsf{ipos}(\mathcal{A}_i)\}\right) & \\ \mathsf{ipos}(\mathcal{A}_0 \supset \mathcal{A}_1) & \triangleq & \{\epsilon\} & \\ \mathsf{ipos}(\mathcal{A}_0 \oplus \mathcal{A}_1) & \triangleq & \bigcup_{i\in[0,1]} \mathsf{ipos}(\mathcal{A}_i) & \end{array}$$

A su vez, dado un μ-tipo contractivo A, se define $\mathsf{ipos}(A) \triangleq \mathsf{ipos}(\llbracket A \rrbracket^{\mathfrak{T}})$.

En particular, los nodos unión son colapsados en un mismo conjunto, mientras que no se permite inspeccionar por debajo de tipos funcionales. La intuición detrás de esto es que los patrones permiten descomponer data structures, cuyos tipos pueden estar definidos de manera recursiva como la unión de compounds y constructores atómicos, *e.g.* $\mathsf{Nat} \triangleq \mu\alpha.\mathtt{zero} \oplus \mathtt{suc} \mathbin{@} \alpha$. Sin embargo, los patrones no pueden descomponer funciones (la operación de matching solo tiene éxito con una abstracción como parámetro cuando el patrón es trivial, *i.e.* $p = x$), por lo que no es necesario inspeccionar internamente sus tipos.

Dada una derivación de tipo para un patrón, las posiciones de este último resultan ser un subconjunto de las posiciones inspeccionables del tipo asociado.

Lema 3.3.36. *Sea $\pi_p \rhd_{\mathcal{P}} \theta \vdash_{\mathsf{p}} p : A$. Luego,* $\mathsf{pos}(p) \subseteq \mathsf{ipos}(A)$.

Más aún, la relación de equivalencia preserva las posiciones inspeccionables.

Lema 3.3.37. *Sean $\mathcal{A} \simeq_{\mathfrak{T}} \mathcal{B}$. Luego,* $\mathsf{ipos}(\mathcal{A}) = \mathsf{ipos}(\mathcal{B})$.

Corolario 3.3.38. *Sean $A \simeq_{\mu} B$. Luego,* $\mathsf{ipos}(A) = \mathsf{ipos}(B)$.

A su vez, el sub-tipado preserva el orden de inclusión del los conjuntos de posiciones inspeccionables subyacentes.

Lema 3.3.39. *Sean $\mathcal{A} \preceq_{\mathfrak{T}} \mathcal{B}$. Luego,* $\mathsf{ipos}(\mathcal{A}) \subseteq \mathsf{ipos}(\mathcal{B})$.

Corolario 3.3.40. *Sean $A \preceq_{\mu} B$. Luego,* $\mathsf{ipos}(A) \subseteq \mathsf{ipos}(B)$.

Estos resultados justifican la buena definición de la noción de constructores de tipo admitidos en una posición dada de un patrón.

Definición 3.3.41. *El conjunto de* constructores admitidos *por un tipo infinito $\mathcal{A}$ en posición* $\mathsf{p} \in \mathsf{ipos}(\mathcal{A})$*, notado $\mathcal{A}{\downarrow_{\circ}}_{\mathsf{p}}$, se define como:*

$$\begin{array}{rcll} \mathcal{A}{\downarrow_{\circ}}_{\epsilon} & \triangleq & \{\mathcal{A}(\epsilon)\} & \text{si } \mathcal{A}(\epsilon) \neq \oplus \\ \mathcal{A}{\downarrow_{\circ}}_{i\mathsf{p}} & \triangleq & \emptyset & \text{si } \mathcal{A}(\epsilon) \notin \{@, \oplus\},\ i \in [0,1] \\ (\mathcal{A}_0 \mathbin{@} \mathcal{A}_1){\downarrow_{\circ}}_{i\mathsf{p}} & \triangleq & \mathcal{A}_i{\downarrow_{\circ}}_{\mathsf{p}} & \text{para } i \in [0,1] \\ (\mathcal{A}_0 \oplus \mathcal{A}_1){\downarrow_{\circ}}_{\mathsf{p}} & \triangleq & \bigcup_{i\in[0,1]} \mathcal{A}_i{\downarrow_{\circ}}_{\mathsf{p}} & \end{array}$$

A su vez, dado un μ-tipo contractivo A, se define $A{\downarrow_{\circ}}_{\mathsf{p}} \triangleq \llbracket A \rrbracket^{\mathfrak{T}}{\downarrow_{\circ}}_{\mathsf{p}}$.

En particular, dado $\rhd_{\mathcal{P}} \theta \vdash_{\mathsf{p}} p : A$, el conjunto $A\downarrow_{\diamond \mathsf{p}}$ está bien definido para toda posición $\mathsf{p} \in \mathsf{pos}(p)$ pues $\mathsf{pos}(p) \subseteq \mathsf{ipos}(A)$.

Como es esperado, la equivalencia de tipos preserva el conjunto de constructores admitidos en una posición inspeccionable dada.

Lema 3.3.42. *Sean $\mathcal{A} \simeq_{\mathfrak{T}} \mathcal{B}$ y $\mathsf{p} \in \mathsf{ipos}(\mathcal{A})$. Luego, $\mathcal{A}\downarrow_{\diamond \mathsf{p}} = \mathcal{B}\downarrow_{\diamond \mathsf{p}}$.*

Corolario 3.3.43. *Sean $A \simeq_{\mu} B$ y $\mathsf{p} \in \mathsf{ipos}(A)$. Luego, $A\downarrow_{\diamond \mathsf{p}} = B\downarrow_{\diamond \mathsf{p}}$.*

También es el caso para sub-tipado, el cual preserva el orden de inclusión de los respectivos conjuntos de constructores admitidos.

Lema 3.3.44. *Sean $\mathcal{A} \preceq_{\mathfrak{T}} \mathcal{B}$ y $\mathsf{p} \in \mathsf{ipos}(\mathcal{A})$. Luego, $\mathcal{A}\downarrow_{\diamond \mathsf{p}} \subseteq \mathcal{B}\downarrow_{\diamond \mathsf{p}}$.*

Corolario 3.3.45. *Sean $A \preceq_{\mu} B$ y $\mathsf{p} \in \mathsf{ipos}(A)$. Luego, $A\downarrow_{\diamond \mathsf{p}} \subseteq B\downarrow_{\diamond \mathsf{p}}$.*

El siguiente predicado informa cuándo los conjuntos de argumentos posibles entre dos patrones se solapan, basándose no solo en la estructura de los patrones, sino también en su información de tipo asociada.

Definición 3.3.46. *La asignación de tipo $p : A$ se* solapa *con $q : B$, notado $\mathsf{ovl}(p : A, q : B)$, sii $\forall \mathsf{p} \in \mathsf{mmpos}(p, q).A\downarrow_{\diamond \mathsf{p}} \cap B\downarrow_{\diamond \mathsf{p}} \neq \emptyset$.*

Aquellos casos de solapamiento entre conjuntos de argumentos son precisamente los que requieren de la restricción sobre los tipos asociados a los patrones. Esto se expresa en la noción de *compatibilidad*.

Definición 3.3.47. *La asignación de tipo $p : A$ es* compatible *con $q : B$, notado $\mathsf{cmp}(p : A, q : B)$, sii $\mathsf{ovl}(p : A, q : B) \implies B \preceq_{\mu} A$. Una lista de asignaciones $[p_i : A_i]_{i<n}$ se dice* compatible *si todo elemento es compatible con sus sucesores:* i.e. $\mathsf{cmp}([p_i : A_i]_{i<n})$ *sii* $\forall i, j < n.i < j \implies \mathsf{cmp}(p_i : A_i, p_j : A_j)$.

Ejemplos de derivaciones

Una vez presentada la noción de compatibilidad se cuenta con todas las herramientas para ilustrar cómo se tipan o rechazan los términos anteriormente analizados.

En particular, el ejemplo (3.3) es rechazado por el sistema puesto que los patrones $\mathtt{l}\, x$ e $\mathtt{l}\, y$ se solapan, pero sus tipos asociados $\mathbb{I} \mathbin{@} \mathsf{Nat}$ y $\mathbb{I} \mathbin{@} \mathsf{Bool}$ no se encuentran en relación de sub-tipado.

Más interesante aún resulta el caso de la función upd (*cf.* ejemplo (3.2)), la cual constituye el ejemplo paradigmático de path polymorphism. En particular, tipar upd implica tipar el operador de *punto fijo* Y, lo cual es posible gracias a la expresividad de los tipos recursivos. En primer lugar es necesario tipar una variable aplicada a sí misma, para lo que se apela al tipo recursivo $\mu X.X \supset A$ con $X \notin A$. Notar que se tiene entonces $\mu X.X \supset A \simeq_{\mu} (\mu X.X \supset A) \supset A$. Luego, la siguiente derivación π_{xx} es válida:

$$\dfrac{\dfrac{\dfrac{}{x : \mu X.X \supset A \vdash x : \mu X.X \supset A}\,(\text{T-VAR})}{x : \mu X.X \supset A \vdash x : (\mu X.X \supset A) \supset A}\,(\text{T-SUBS}) \qquad \dfrac{}{x : \mu X.X \supset A \vdash x : \mu X.X \supset A}\,(\text{T-VAR})}{x : \mu X.X \supset A \vdash x\, x : A}\,(\text{T-APP})$$

Más aún, es posible aplicar f y abstraer x, obteniendo la derivación $\pi_{x \to f(xx)}$:

$$\dfrac{\dfrac{\dfrac{}{f : A \supset A \vdash f : A \supset A}\,(\text{T-VAR}) \qquad \pi_{xx}}{f : A \supset A; x : \mu X.X \supset A \vdash f\,(x\, x) : A}\,(\text{T-APP})}{f : A \supset A \vdash x \to_{\{x:\mu X.X \supset A\}} f\,(x\, x) : (\mu X.X \supset A) \supset A}\,(\text{T-ABS})$$

Finalmente, apelando nuevamente a equivalencia y sub-tipado se obtiene la siguiente derivación π_{Y} para el operador Y con un tipo A arbitrario:

$$\dfrac{\dfrac{}{f : A \supset A \vdash f : A \supset A}\,(\text{P-MATCH}) \qquad \dfrac{\pi_{x \to f(xx)} \qquad \dfrac{\pi_{x \to f(xx)}}{f : A \supset A \vdash x \to_{\{x:\mu X.X \supset A\}} f\,(x\,x) : \mu X.X \supset A}\,(\text{T-SUBS})}{f : A \supset A \vdash (x \to_{\{x:\mu X.X \supset A\}} f\,(x\,x))\,(x \to_{\{x:\mu X.X \supset A\}} f\,(x\,x)) : A}\,(\text{T-APP})}{\vdash f \to_{\{f:A \supset A\}} (x \to_{\{x:\mu X.X \supset A\}} f\,(x\,x))\,(x \to_{\{x:\mu X.X \supset A\}} f\,(x\,x)) : (A \supset A) \supset A}\,(\text{T-ABS})$$

Basta ver entonces que se puede derivar el tipo adecuado para el término t_{upd} dado por

$$upd \to_{\{upd:(A \supset B) \supset F_A \supset F_B\}} f \to_{\{f:A \supset B\}} \begin{array}{ll} (\ \mathtt{1}\,z \to_{\{z:A\}} & \mathtt{1}\,(f\,z) \\ \mid x\,y \to_{\{x:F_A, y:F_A\}} & (upd\,f\,x)\,(upd\,f\,y) \\ \mid w \to_{\{w:\mathbb{nil} \oplus \mathbb{cons} \oplus \mathbb{node}\}} & w \qquad\qquad\ \ ) \end{array}$$

de modo de obtener upd al aplicar el operador Y (*cf.* definición de Y y funciones recursivas en Sec. 3.1.1). Recordar que se busca asignar $\mathsf{upd} : (A \supset B) \supset F_A \supset F_B$, donde F_X se define como el tipo recursivo $\mu\alpha.(\mathbb{I} \mathbin{@} X) \oplus (\alpha \mathbin{@} \alpha) \oplus (\mathbb{nil} \oplus \mathbb{cons} \oplus \mathbb{node})$. Entonces, es necesario obtener una derivación para el juicio $\vdash t_{\mathsf{upd}} : ((A \supset B) \supset F_A \supset F_B) \supset (A \supset B) \supset F_A \supset F_B$ para luego aplicar correctamente el operador Y.

Por un lado, es posible derivar los juicios de tipado esperados para los patrones de la abstracción más interna de upd, obteniendo $\pi_{\mathtt{1}z}$, π_{xy} y π_w respectivamente:

$$\dfrac{\dfrac{}{\vdash_{\mathsf{p}} \mathtt{1} : \mathbb{I}}\,(\text{P-CONST}) \qquad \dfrac{}{z : A \vdash_{\mathsf{p}} z : A}\,(\text{P-MATCH})}{z : A \vdash_{\mathsf{p}} \mathtt{1}\,z : \mathbb{I} \mathbin{@} A}\,(\text{P-COMP})$$

$$\dfrac{\dfrac{}{x : F_A \vdash_{\mathsf{p}} x : F_A}\,(\text{P-MATCH}) \qquad \dfrac{}{y : F_A \vdash_{\mathsf{p}} y : F_A}\,(\text{P-MATCH})}{x : F_A; y : F_A \vdash_{\mathsf{p}} x\,y : F_A \mathbin{@} F_A}\,(\text{P-COMP})$$

$$\dfrac{}{w : \mathbb{nil} \oplus \mathbb{cons} \oplus \mathbb{node} \vdash_{\mathsf{p}} w : \mathbb{nil} \oplus \mathbb{cons} \oplus \mathbb{node}}\,(\text{P-MATCH})$$

Notar que estas asignaciones de tipo a los patrones son compatibles, pues el predicado de solapamiento resulta falso en los tres casos a comparar:

1. $\mathsf{cmp}(\mathtt{1}\,z : \mathbb{I} \mathbin{@} A, x\,y : F_A \mathbin{@} F_A)$: se tiene el conjunto de posiciones $\mathsf{mmpos}(\mathtt{1}\,z, x\,y) = \{1\}$, y al inspeccionar resulta $(\mathbb{I} \mathbin{@} A)\downarrow_1 \cap\, (F_A \mathbin{@} F_A)\downarrow_1 = \{\mathbb{I}\} \cap \{@, \mathbb{nil}, \mathbb{cons}, \mathbb{node}\} = \emptyset$.

2. $\mathsf{cmp}(\mathtt{1}\,z : \mathbb{I} \mathbin{@} A, w : \mathbb{nil} \oplus \mathbb{cons} \oplus \mathbb{node})$: se tiene el conjunto de posiciones $\mathsf{mmpos}(\mathtt{1}\,z, w) = \{\epsilon\}$, y al inspeccionar resulta $(\mathbb{I} \mathbin{@} A)\downarrow_\epsilon \cap\, (\mathbb{nil} \oplus \mathbb{cons} \oplus \mathbb{node})\downarrow_\epsilon = \{@\} \cap \{\mathbb{nil}, \mathbb{cons}, \mathbb{node}\} = \emptyset$.

3. $\mathsf{cmp}(x\,y : F_A \mathbin{@} F_A, w : \mathbb{nil} \oplus \mathbb{cons} \oplus \mathbb{node})$: se tiene el conjunto de posiciones $\mathsf{mmpos}(x\,y, w) = \{\epsilon\}$, y al inspeccionar resulta $(F_A \mathbin{@} F_A)\downarrow_\epsilon \cap\, (\mathbb{nil} \oplus \mathbb{cons} \oplus \mathbb{node})\downarrow_\epsilon = \{@\} \cap \{\mathbb{nil}, \mathbb{cons}, \mathbb{node}\} = \emptyset$.

Por lo que no es necesario aplicar restricciones sobre los tipos asociados.

Por otro lado, considerar la siguiente derivación $\pi_{upd\,f}$ para el $upd\,f$:

$$\dfrac{\dfrac{}{upd : (A \supset B) \supset F_A \supset F_B \vdash upd : (A \supset B) \supset F_A \supset F_B}\,(\text{T-VAR}) \qquad \dfrac{}{f : A \supset B \vdash f : A \supset B}\,(\text{T-VAR})}{upd : (A \supset B) \supset F_A \supset F_B; f : A \supset B \vdash upd\,f : F_A \supset F_B}\,(\text{T-APP})$$

que permite derivar $\pi_{upd\,f\,x}$ y $\pi_{upd\,f\,y}$ respectivamente:

$$\dfrac{\pi_{upd\,f} \qquad \dfrac{}{x : F_A \vdash x : F_A}\,(\text{T-VAR})}{upd : (A \supset B) \supset F_A \supset F_B; f : A \supset B; x : F_A \vdash upd\,f\,x : F_B}\,(\text{T-APP})$$

$$\dfrac{\pi_{upd\,f} \qquad \dfrac{}{y : F_A \vdash y : F_A}\,(\text{T-VAR})}{upd : (A \supset B) \supset F_A \supset F_B; f : A \supset B; y : F_A \vdash upd\,f\,y : F_B}\,(\text{T-APP})$$

Se apela a sub-tipado para derivar el tipo F_B correcto para los cuerpos de las ramas de la abstracción interna. Notar que $F_X \simeq_\mu (\mathbb{I} \mathbin{@} X) \oplus (F_X \mathbin{@} F_X) \oplus (\mathbb{nil} \oplus \mathbb{cons} \oplus \mathbb{node})$ por (E-FOLD). Luego, se tienen $\mathbb{I} \mathbin{@} B \preceq_\mu F_B$, $F_B \mathbin{@} F_B \preceq_\mu F_B$ y $\mathbb{nil} \oplus \mathbb{cons} \oplus \mathbb{node} \preceq_\mu F_B$ por (S-UNION-R1), (S-UNION-R1) y (S-EQ). Por lo tanto, es posible derivar $\pi'_{\mathtt{l}z}$, π'_{xy} y π'_w respectivamente (para simplificar la lectura se nombra las derivaciones con los patrones de sus respectivas ramas):

$$\dfrac{\dfrac{\dfrac{}{\vdash \mathtt{l} : \mathbb{I}}\,(\text{T-CONST}) \qquad \dfrac{\dfrac{}{f : A \supset B \vdash f : A \supset B}\,(\text{T-VAR}) \qquad \dfrac{}{z : A \vdash z : A}\,(\text{T-VAR})}{f : A \supset B; z : A \vdash f\,z : B}\,(\text{T-APP})}{f : A \supset B; z : A \vdash \mathtt{l}\,(f\,z) : \mathbb{I} \mathbin{@} B}\,(\text{T-COMP})}{f : A \supset B; z : A \vdash \mathtt{l}\,(f\,z) : F_B}\,(\text{T-SUBS})$$

$$\dfrac{\dfrac{\pi_{upd\,f\,x} \qquad \pi_{upd\,f\,y}}{upd : (A \supset B) \supset F_A \supset F_B; f : A \supset B; x : F_A; y : F_A \vdash (upd\,f\,x)\,(upd\,f\,y) : F_B \mathbin{@} F_B}\,(\text{T-COMP})}{upd : (A \supset B) \supset F_A \supset F_B; f : A \supset B; x : F_A; y : F_A \vdash (upd\,f\,x)\,(upd\,f\,y) : F_B}\,(\text{T-SUBS})$$

$$\dfrac{\dfrac{}{w : \mathbb{nil} \oplus \mathbb{cons} \oplus \mathbb{node} \vdash w : \mathbb{nil} \oplus \mathbb{cons} \oplus \mathbb{node}}\,(\text{T-VAR})}{w : \mathbb{nil} \oplus \mathbb{cons} \oplus \mathbb{node} \vdash w : F_B}\,(\text{T-SUBS})$$

Se está entonces en condiciones de aplicar (T-ABS) y (T-SUBS) para obtener:

$$\dfrac{\pi_{\mathtt{l}z} \quad \pi_{xy} \quad \pi_w \qquad \pi'_{\mathtt{l}z} \quad \pi'_{xy} \quad \pi'_w}{upd : (A \supset B) \supset F_A \supset F_B; f : A \supset B \vdash \begin{array}{lll} (\ \mathtt{l}\,z \to_{\{z:A\}} & \mathtt{l}\,(f\,z) & \\ \mid x\,y \to_{\{x:F_A, y:F_A\}} & (upd\,f\,x)\,(upd\,f\,y) & : F_A \supset F_B \\ \mid w \to_{\{w:\mathbb{nil}\oplus\mathbb{cons}\oplus\mathbb{node}\}} & w & \ ) \end{array}}\,(\text{T-ABS})/(\text{T-SUBS})$$

Finalmente basta aplicar (T-ABS) dos veces para abstraer las variables upd y f, obteniendo la derivación $\rhd_{\mathcal{P}} \vdash t_{\mathsf{upd}} : ((A \supset B) \supset F_A \supset F_B) \supset (A \supset B) \supset F_A \supset F_B$, y luego concluir por (T-APP) con π_{Y}, $\pi_{\mathsf{upd}} \rhd_{\mathcal{P}} \vdash \mathsf{Y}\,t_{\mathsf{upd}} : (A \supset B) \supset F_A \supset F_B$.

Como ejemplo adicional, suponer que se quiere aplicar la función `upd` a una estructura que contiene valores de diferentes tipos: como ser números prefijados con la constante `l` y funciones de naturales en naturales prefijadas con la constante `f`. Notar que `upd` no puede ser tipada tal cual es. La razón es que su última rama deberá soportar adicionalmente valores de tipo funcional, recibiendo el tipo $\mathbb{nil} \oplus \mathbb{cons} \oplus \mathbb{node} \oplus (\mathsf{Nat} \supset \mathsf{Nat})$. Esta expresión no resulta ser un datatype, por contener un componente funcional en la unión. Como consecuencia, el patrón $x\,y$ de la segunda rama de `upd` no puede ser tipado, dado que requiere un tipo aplicativo @ cuya primera componente debe ser necesariamente un datatype. Una posible solución a esta situación es considerar una rama adicional capaz de manejar

valores prefijados por $\mathtt{f}$:

$$\begin{array}{l}\mathsf{upd}' = f \to_{\{f:A\supset B\}} g \to_{\{g:(A_0\supset A_1)\supset B\}} \\ \qquad\qquad \begin{array}{lll}(\ \mathtt{l}\, z & \to_{\{z:A\}} & \mathtt{l}\,(f\, z) \\ \mid \mathtt{f}\, z & \to_{\{z:A_0\supset A_1\}} & \mathtt{f}\,(g\, z) \\ \mid x\, y & \to_{\{x:C,y:D\}} & (\mathsf{upd}\, f\, x)\,(\mathsf{upd}\, f\, y) \\ \mid w & \to_{\{w:E\}} & w \qquad\qquad\qquad\qquad\qquad\qquad)\end{array}\end{array}$$

El tipo de upd' es entonces $(A \supset B) \supset ((A_0 \supset A_1) \supset B) \supset (F'_{A,A_0\supset A_1} \supset F'_{B,B})$ con el tipo recursivo $F'_{X,Y} \triangleq \mu\alpha.(\mathbb{l} \mathbin{@} X) \oplus (\mathbb{f} \mathbin{@} Y) \oplus (\alpha \mathbin{@} \alpha) \oplus (\mathbb{nil} \oplus \mathbb{cons} \oplus \mathbb{node})$. Aunque a primera vista esto puede resultar una limitación del sistema, es en realidad bastante natural. El sistema de tipos establece una clara distinción entre data semi-estructurada, propensa a ser tratada con path polymorphism, y data no estructurada, representada aquí por los tipos base y funcionales.

3.3.5. Propiedades del sistema

Antes de abordar la adecuación del sistema (*i.e. subject reduction* y *progress* en Sec. 3.4) se presentan una serie de propiedades adicionales, entre las que se destacan el clásico *lema de sustitución* (Lem. 3.3.51) y el novedoso *lema de compatibilidad* (Lem. 3.3.55).

Los siguientes dos lemas son adaptaciones directas del clásico Lema de Generación al sistema de tipos de CAP. Permiten descomponer el tipado de un patrón o un término basándose en la forma de estos últimos respectivamente.

Lema 3.3.48 (Generación de patrones)**.** *Sea $\pi_p \rhd_{\mathcal{P}} \theta \vdash_{\mathsf{p}} p : A$. Luego,*

1. *Si $p = x$, entonces $\theta(x) = A$.*
2. *Si $p = \mathsf{c}$, entonces $A = \mathbb{c}$.*
3. *Si $p = p_0\, p_1$, entonces $\theta = \theta_0; \theta_1$ y $A = D \mathbin{@} A'$ tal que $\pi_{p_0} \rhd_{\mathcal{P}} \theta_0 \vdash_{\mathsf{p}} p_0 : D$ y $\pi_{p_1} \rhd_{\mathcal{P}} \theta_1 \vdash_{\mathsf{p}} p_1 : A'$.*

Lema 3.3.49 (Generación)**.** *Sea $\pi_t \rhd_{\mathcal{P}} \Gamma \vdash t : A$. Luego,*

1. *Si $t = x$, entonces $\Gamma(x) = A'$ y $A' \preceq_\mu A$.*
2. *Si $t = \mathsf{c}$, entonces $\mathbb{c} \preceq_\mu A$.*
3. *Si $t = r\, u$, entonces:*
   a) *$\exists D, A' \in \mathcal{T}$ tal que $D \mathbin{@} A' \preceq_\mu A$, $\pi_r \rhd_{\mathcal{P}} \Gamma \vdash r : D$ y $\pi_u \rhd_{\mathcal{P}} \Gamma \vdash u : A'$; o bien*
   b) *$\exists (A_i)_{i<n}, A' \in \mathcal{T}$ tal que $A' \preceq_\mu A$, $\pi_r \rhd_{\mathcal{P}} \Gamma \vdash r : \oplus_{i<n} A_i \supset A'$ y $\pi_u \rhd_{\mathcal{P}} \Gamma \vdash u : A_k$ para algún $k < n$.*
4. *Si $t = (p_i \to s_i)_{i<n}$, entonces $\exists (A_i)_{i<n}, B \in \mathcal{T}$ tal que $\oplus_{i<n} A_i \supset B \preceq_\mu A$, $(\pi_{p_i} \rhd_{\mathcal{P}} \theta_i \vdash_{\mathsf{p}} p_i : A_i)_{i<n}$, $\mathsf{cmp}([p_i : A_i]_{i<n})$ y $(\pi_{s_i} \rhd_{\mathcal{P}} \Gamma; \theta_i \vdash s_i : B)_{i<n}$.*

El siguiente lema relaciona las variables libres de un término con las asunciones del contexto de tipado. En particular, permite ver que las clásicas reglas de *weakening* y *strengthening* son admisibles en el sistema.

Lema 3.3.50. *Sea $\pi \rhd_{\mathcal{P}} \Gamma \vdash t : A$. Luego,*

1. *Sea Δ un contexto de tipado tal que $\Gamma \subseteq \Delta$, entonces $\pi' \rhd_{\mathcal{P}} \Delta \vdash t : A$.*

2. $\mathsf{fv}(t) \subseteq \mathsf{dom}(\Gamma)$.

3. $\pi' \rhd_{\mathcal{P}} \Gamma|_{\mathsf{fv}(t)} \vdash t : A$.

Un resultado auxiliar clásico para la prueba de *subject reduction* es el *lema de sustitución*. Recordar que $\Gamma \vdash \sigma : \theta$ denota $\mathsf{dom}(\sigma) = \mathsf{dom}(\theta)$ y para toda $x \in \mathsf{dom}(\sigma)$ vale $\Gamma \vdash \sigma(x) : \theta(x)$. Luego, el lema adaptado al sistema de tipos de CAP establece:

Lema 3.3.51 (Sustitución)**.** *Sean $\pi_s \rhd_{\mathcal{P}} \Gamma; \theta \vdash s : A$ y $\pi_\sigma \rhd_{\mathcal{P}} \Gamma \vdash \sigma : \theta$. Luego, $\pi_{\sigma s} \rhd_{\mathcal{P}} \Gamma \vdash \sigma s : A$.*

El siguiente lema permite deducir la forma del tipo asignado a un término cuando este último es un data structure. Esencialmente muestra que a todo data structure tipable puede asignársele, a su vez, un datatype no-unión.

Lema 3.3.52. *Sea $\pi_d \rhd_{\mathcal{P}} \Gamma \vdash d : A$ con $d \in \mathbb{D}_{\mathsf{CAP}}$. Luego, existe $D \in \mathcal{D}$ no-unión tal que $D \preceq_\mu A$ y $\pi'_d \rhd_{\mathcal{P}} \Gamma \vdash d : D$. Más aún,*

1. *Si $d = \mathsf{c}$, entonces $D = \mathbb{c}$.*

2. *Si $d = d'\, t$, entonces $D = D' \mathbin{@} A'$ tal que $\pi'_{d'} \rhd_{\mathcal{P}} \Gamma \vdash d' : D'$ y $\pi'_t \rhd_{\mathcal{P}} \Gamma \vdash t : A'$.*

El siguiente lema muestra que la falla en la operación de match es suficiente para garantizar que los tipos del patrón y el argumento no están relacionados por sub-tipado.

Lema 3.3.53. *Sean $\pi_u \rhd_{\mathcal{P}} \Gamma \vdash u : B$, $\pi_p \rhd_{\mathcal{P}} \theta \vdash_{\mathsf{p}} p : A$ y $\{\!\{p \setminus u\}\!\} = \mathtt{fail}$. Luego, $B \not\preceq_\mu A$.*

En contraposición al lema anterior, en caso de éxito en la operación de match puede derivarse el tipo de la sustitución resultante, como se expresa en el siguiente resultado.

Lema 3.3.54. *Sean $\pi_u \rhd_{\mathcal{P}} \Gamma \vdash u : A$, $\pi_p \rhd_{\mathcal{P}} \theta \vdash_{\mathsf{p}} p : A$ y $\{\!\{p \setminus u\}\!\} = \sigma$. Luego, $\pi_\sigma \rhd_{\mathcal{P}} \Gamma \vdash \sigma : \theta$.*

La compatibilidad debe ser interpretada en el contexto de un β-redex. Asumir que un argumento u de tipo B es pasado a una función donde hay (al menos) dos ramas, definidas por los patrones p y q, donde el último tiene el mismo tipo que u. Si el argumento es capturado por el primer patrón de un tipo A (potencialmente) distinto, entonces $\mathsf{ovl}(p : A, q : B)$ debe satisfacerse. De ser así, como los patrones de una abstracción tipada son compatibles y p precede a q, se tiene $B \preceq_\mu A$ y, por lo tanto puede darse tipo A a u también. El *lema de compatibilidad* garantiza que efectivamente el predicado de solapamiento se satisface en tal situación.

Lema 3.3.55 (Compatibilidad)**.** *Sean $\pi_u \rhd_{\mathcal{P}} \Gamma \vdash u : B$, $\pi_p \rhd_{\mathcal{P}} \theta \vdash_{\mathsf{p}} p : A$, $\pi_q \rhd_{\mathcal{P}} \theta' \vdash_{\mathsf{p}} q : B$ y $\{\!\{p \backslash u\}\!\} = \sigma$. Luego, $\mathsf{ovl}(p : A, q : B)$.*

Demostración. Por inducción en p. Notar que $p \lhd q$ implica $\mathsf{mmpos}(p, q) = \emptyset$, lo que a su vez implica $\mathsf{ovl}(p : A, q : B)$ trivialmente. Se analizan únicamente lo casos $p \not\lhd q$.

- $p = \mathsf{c} \neq q$. De $\{\!\{p \setminus u\}\!\} = \sigma$ se tiene $u = \mathsf{c}$ y, por Lem. 3.3.52 (1), $\mathbb{c} \preceq_\mu B$. Luego, por Cor. 3.3.45, $\mathbb{c}\!\downarrow_\epsilon = \{\mathbb{c}\} \subseteq B\!\downarrow_\epsilon$. Más aún, por Lem. 3.3.48 (2), $A = \mathbb{c}$, lo que implica $A\!\downarrow_\epsilon = \{\mathbb{c}\}$. Se concluye pues $\mathsf{mmpos}(p, q) = \{\epsilon\}$ y $A\!\downarrow_\epsilon \cap B\!\downarrow_\epsilon \neq \emptyset$.

- $p = p_0\, p_1$. De $\{\!\{p \setminus u\}\!\} = \sigma$ se tienen $u = d\, u'$ y $\sigma = \sigma_0 \uplus \sigma_1$ con $\sigma_0 = \{\!\{p_0 \setminus d\}\!\}$ y $\sigma_1 = \{\!\{p_1 \setminus u'\}\!\}$. Por Lem. 3.3.48 (3), $\theta = \theta_0; \theta_1$ y $A = D \mathbin{@} A'$ tal que $\pi_{p_0} \rhd_{\mathcal{P}} \theta_0 \vdash_{\mathsf{p}} p_0 : D$ y $\pi_{p_1} \rhd_{\mathcal{P}} \theta_1 \vdash_{\mathsf{p}} p_1 : A'$. Más aún, por Lem. 3.3.52 (2), $\exists D', B' \in \mathcal{T}$ tal que $D' \mathbin{@} B' \preceq_\mu B$, $\pi_d \rhd_{\mathcal{P}} \Gamma \vdash d : D'$ y $\pi_{u'} \rhd_{\mathcal{P}} \Gamma \vdash u' : B'$. Se procede analizando la forma de q:

- $q = y$. Luego, $\mathsf{mmpos}(p, q) = \{\epsilon\}$. Por Cor. 3.3.45, $D' \mathbin{@} B'{\downarrow_\diamond}_\epsilon = \{@\} \subseteq B{\downarrow_\diamond}_\epsilon$. Luego, se concluye pues $A{\downarrow_\diamond}_\epsilon = \{@\}$ y por lo tanto $A{\downarrow_\diamond}_\epsilon \cap B{\downarrow_\diamond}_\epsilon \neq \emptyset$.
- $q = \mathtt{d}$. Por Lem. 3.3.48 (2), $B = \Bbbd$, lo que lleva a una contradicción con $D' \mathbin{@} B' \preceq_\mu B$ por Teo. 3.3.31. Luego, este caso no es posible.
- $q = q_0\, q_1$. Por Lem. 3.3.48 (3), $\theta' = \theta'_0; \theta'_1$ y $B = D'' \mathbin{@} B''$ tal que $\pi_{q_0} \rhd_\mathcal{P} \theta'_0 \vdash_\mathsf{p} q_0 : D''$ y $\pi_{q_1} \rhd_\mathcal{P} \theta'_1 \vdash_\mathsf{p} q_1 : B''$. De $D' \mathbin{@} B' \preceq_\mu B$, por Teo. 3.3.31 (2), se tienen $D' \preceq_\mu D''$ y $B' \preceq_\mu B''$. Más aún, por (T-SUBS) con π_d y $\pi_{u'}$, se obtienen $\rhd_\mathcal{P} \Gamma \vdash d : D''$ y $\rhd_\mathcal{P} \Gamma \vdash u' : B''$. Luego, por *h.i.* se satisfacen $\mathsf{ovl}(p_0 : D, q_0 : D'')$ y $\mathsf{ovl}(p_1 : A', q_1 : B'')$. Por último, como ambos patrones son aplicativos toda posición de mismatch es interna, por lo que se concluye $\mathsf{ovl}(p : A, q : B)$.

□

3.4. Adecuación del sistema

La presente sección aborda la adecuación del sistema. Esto se refiere particularmente a dos propiedades: *subject reduction* y *progress*.

Primero de trata *subject reduction*. Es decir, que el tipo de un término se preserva por reducción.

Teorema 3.4.1 (Subject Reduction)**.** *Sea* $\pi \rhd_\mathcal{P} \Gamma \vdash t : A$. *Si* $t \to t'$, *entonces existe* $\pi' \rhd_\mathcal{P} \Gamma \vdash t' : A$.

Demostración. Por definición, $t \to t'$ implica $t = \mathtt{C}\langle l \rangle$ y $t' = \mathtt{C}\langle r \rangle$ con $l \mapsto_\beta r$. La demostración es por inducción en $\mathtt{C}$.

- $\mathtt{C} = \Box$. Luego, $t = (p_i \to_{\theta_i} s_i)_{i<n}\, u$ con $(\{\!\{p_i \setminus u\}\!\} = \mathtt{fail})_{i<k}$, $\{\!\{p_k \setminus u\}\!\} = \sigma$ y $t' = \sigma s_k$ para algún $k < n$. Por Lem. 3.3.49 (3) hay dos posibles casos:

  a) $\exists D, A' \in \mathcal{T}$ tal que $D \mathbin{@} A' \preceq_\mu A$, $\rhd_\mathcal{P} \Gamma \vdash (p_i \to_{\theta_i} s_i)_{i<n} : D$ y $\rhd_\mathcal{P} \Gamma \vdash u : A'$. Luego, por Lem. 3.3.49 (4), $\exists (A_i)_{i<n}, B \in \mathcal{T}$ tal que $\oplus_{i<n} A_i \supset B \preceq_\mu D$. Por Nota 3.3.4, $D \simeq_\mu \oplus_{j<m} D_j$ con $D_j \in \mathcal{D}$ no-unión para todo $j < m$. Más aún, por (S-EQ) y (S-TRANS), $\oplus_{i<n} A_i \supset B \preceq_\mu \oplus_{j<m} D_j$ y, por Lem. 3.3.33, $\oplus_{i<n} A_i \supset B \preceq_\mu D_l$ para algún $l < m$. Esto lleva a una contracción con Teo. 3.3.31 y el hecho de que $D_l \in \mathcal{D}$. Luego, este caso no es posible.

  b) $\exists (B_j)_{j<m}, B' \in \mathcal{T}$ tal que $B' \preceq_\mu A$, $\rhd_\mathcal{P} \Gamma \vdash (p_i \to_{\theta_i} s_i)_{i<n} : \oplus_{j<m} B_j \supset B'$ y $\rhd_\mathcal{P} \Gamma \vdash u : B_l$ para algún $l < m$. Por Lem. 3.3.49 (4), $\exists (A_i)_{i<n}, A' \in \mathcal{T}$ tal que $\oplus_{i<n} A_i \supset A' \preceq_\mu \oplus_{j<m} B_j \supset B'$, $(\pi_{p_i} \rhd_\mathcal{P} \theta_i \vdash_\mathsf{p} p_i : A_i)_{i<n}$, $\mathsf{cmp}([p_i : A_i]_{i<n})$ y $(\pi_{s_i} \rhd_\mathcal{P} \Gamma; \theta_i \vdash s_i : A')_{i<n}$. Por Teo. 3.3.31 (3), $\oplus_{j<m} B_j \preceq_\mu \oplus_{i<n} A_i$ y $A' \preceq_\mu B'$. Se quiere ver que puede derivarse $\Gamma \vdash u : A_k$, para lo que se consideran dos posibles situaciones:

     a) Si u es una matchable form, entonces
     - u es un data structure. Luego, por Lem. 3.3.52, existe $D \in \mathcal{D}$ no-unión tal que $D \preceq_\mu B_l$ y $\rhd_\mathcal{P} \Gamma \vdash u : D$.
     - u es una abstracción. Luego, por Lem. 3.3.49 (4), $\exists C', C'' \in \mathcal{T}$ tal que $C' \supset C'' \preceq_\mu B_l$ y $\rhd_\mathcal{P} \Gamma \vdash u : C' \supset C''$.

     En ambos casos se tiene $C \in \mathcal{T}$ no-unión tal que $C \preceq_\mu B_l$ y $\rhd_\mathcal{P} \Gamma \vdash u : C$. Luego, $C \preceq_\mu \oplus_{j<m} B_j$ y, por (S-TRANS), $C \preceq_\mu \oplus_{i<n} A_i$. Más aún, por Lem. 3.3.33, $C \preceq_\mu A_{k'}$ para algún $k' < n$ y, por (T-SUBS), $\pi_u \rhd_\mathcal{P} \Gamma \vdash u : A_{k'}$. Si $k = k'$, listo. Sino (*i.e.* $k \neq k'$), por Lem. 3.3.53 con π_u, π_{p_i} y $\{\!\{p_i \setminus u\}\!\} = \mathtt{fail}$, se tiene $A_{k'} \not\preceq_\mu A_i$ para todo $i < k$. Luego, necesariamente es el caso $k < k'$. Por Lem. 3.3.55 con π_u, π_{p_k}, $\pi_{p_{k'}}$ y $\{\!\{p_k \setminus u\}\!\} = \sigma$, se tiene $\mathsf{ovl}(p_k : A_k, p_{k'} : A_{k'})$. Entonces, de $\mathsf{cmp}([p_i : A_i]_{i<n})$ se tiene $\mathsf{cmp}(p_k : A_k, p_{k'} : A_{k'})$ y, por lo tanto, $A_{k'} \preceq_\mu A_k$. Luego, por (T-SUBS), $\pi'_u \rhd_\mathcal{P} \Gamma \vdash u : A_k$.

b) Si u no es una matchable form, entonces $p_k = x$. Más aún, como $\{\!\{p_i \backslash u\}\!\} = \texttt{fail}$ implica u en matchable form, necesariamente se tiene $k = 1$ (*i.e.* no puede haber ramas precedentes que fallen). Por otro lado, $x \lhd p_i$ para todo $i < n$, por lo que $\mathsf{cmp}([p_i : A_i]_{i<n})$ garantiza $A_i \preceq_\mu A_k$ para todo $i < n$. Luego, $\oplus_{i<n} A_i \preceq_\mu A_k$. Más aún, de $B_l \preceq_\mu \oplus_{j<m} B_j$ y $\oplus_{j<m} B_j \preceq_\mu \oplus_{i<n} A_i$ se obtiene $B_l \preceq_\mu A_k$ por (S-TRANS). Entonces, por (T-SUBS), $\pi'_u \rhd_\mathcal{P} \Gamma \vdash u : A_k$.

Finalmente, en ambos casos se tiene $\pi'_u \rhd_\mathcal{P} \Gamma \vdash u : A_k$. Por Lem 3.3.54 con π_{p_k}, se tiene $\rhd_\mathcal{P} \Gamma \vdash \sigma : \theta_k$ y, por Lem. 3.3.51 con π_{s_k}, $\rhd_\mathcal{P} \Gamma \vdash t' : A'$. Luego, de $A' \preceq_\mu B'$ y $B' \preceq_\mu A$, se concluye por (T-SUBS), $\pi' \rhd_\mathcal{P} \Gamma \vdash t' : A$.

- $\mathtt{C} = \mathtt{C}'\, u$. Luego, $t = s\, u$ y $t' = s'\, u$ con $s \to s'$. Por Lem. 3.3.49 (3) se tiene dos posibles casos:

  a) $\exists D, A' \in \mathcal{T}$ tal que $D \mathbin{@} A' \preceq_\mu A$, $\pi_s \rhd_\mathcal{P} \Gamma \vdash s : D$ y $\rhd_\mathcal{P} \Gamma \vdash u : A'$. Por *h.i.* se tiene $\pi_{s'} \rhd_\mathcal{P} \Gamma \vdash s' : D$. Se concluye por (T-COMP) y (T-SUBS).

  b) $\exists (A_i)_{i<n}, A' \in \mathcal{T}$ tal que $A' \preceq_\mu A$, $\pi_s \rhd_\mathcal{P} \Gamma \vdash s : \oplus_{i<n} A_i \supset A'$ y $\rhd_\mathcal{P} \Gamma \vdash u : A_k$ para algún $k < n$. Por *h.i.* $\pi_{s'} \rhd_\mathcal{P} \Gamma \vdash s' : \oplus_{i<n} A_i \supset A'$. Se concluye por (T-APP) y (T-SUBS).

- $\mathtt{C} = s\, \mathtt{C}'$. Luego, $t = s\, u$ y $t' = s\, u'$ con $u \to u'$. Por Lem. 3.3.49 (3) se tiene dos posibles casos:

  a) $\exists D, A' \in \mathcal{T}$ tal que $D \mathbin{@} A' \preceq_\mu A$, $\rhd_\mathcal{P} \Gamma \vdash s : D$ y $\pi_u \rhd_\mathcal{P} \Gamma \vdash u : A'$. Por *h.i.* se tiene $\pi_{u'} \rhd_\mathcal{P} \Gamma \vdash u' : A'$. Se concluye por (T-COMP) y (T-SUBS).

  b) $\exists (A_i)_{i<n}, A' \in \mathcal{T}$ tal que $A' \preceq_\mu A$, $\rhd_\mathcal{P} \Gamma \vdash s : \oplus_{i<n} A_i \supset A'$ y $\pi_u \rhd_\mathcal{P} \Gamma \vdash u : A_k$ para algún $k < n$. Por *h.i.* se tiene $\pi_{u'} \rhd_\mathcal{P} \Gamma \vdash u' : A_k$. Se concluye por (T-APP) y (T-SUBS).

- $\mathtt{C} = (p_i \to_{\theta_i} C'_k)_{i<n}$. Luego, $t = (p_i \to_{\theta_i} s_i)_{i<n}$ y $t' = (p_i \to_{\theta_i} s'_i)_{i<n}$ con $s_k \to s'_k$ y $s'_i = s_i$ para todo $i \neq k < n$. Lem. 3.3.49 (4), $\exists (A_i)_{i<n}, A' \in \mathcal{T}$ tal que $\oplus_{i<n} A_i \supset A' \preceq_\mu A$, $(\rhd_\mathcal{P} \theta_i \vdash_\mathsf{p} p_i : A_i)_{i<n}$, $\mathsf{cmp}([p_i : A_i]_{i<n})$ y $(\rhd_\mathcal{P} \Gamma; \theta_i \vdash s_i : A')_{i<n}$. Por *h.i.* $\rhd_\mathcal{P} \Gamma; \theta_k \vdash s'_k : A'$. Más aún, como $s'_i = s_i$ para todo $i \neq k < n$, $(\rhd_\mathcal{P} \Gamma; \theta_i \vdash s'_i : A')_{i<n}$. Se concluye por (T-ABS) y (T-SUBS).

□

Para formalizar la propiedad de *progress* es necesario introducir previamente la noción de *valor* para CAP, dada por la siguiente gramática:

$$v \quad ::= \quad x\, v \ldots v \mid \mathtt{c}\, v \ldots v \mid (p_i \to_{\theta_i} s_i)_{i<n}$$

El siguiente resultado auxiliar garantiza que la operación de matching tiene éxito cuando el argumento es un valor cerrado (*i.e.* sin variables libres) y tipado. Notar en particular que los valores se encuentran en matchable form.

Lema 3.4.2. *Sean $\pi_v \rhd_\mathcal{P} \vdash v : A$ y $\pi_p \rhd_\mathcal{P} \theta \vdash_\mathsf{p} p : A$. Luego, $\{\!\{p \setminus v\}\!\} = \sigma$.*

Finalmente, *progress* establece que la reducción en términos cerrados no queda estancada hasta no alcanzar un valor.

Teorema 3.4.3 (Progress)**.** *Sea $\pi \rhd_\mathcal{P} \vdash t : A$. Si t no es un valor, entonces $t \to t'$.*

Demostración. Por inducción en t.

- $t = x$. Este caso no es posible pues, por Lem. 3.3.50 (2), $\mathsf{fv}(t) = \{x\} \subseteq \emptyset$.
- $t = c$. El resultado es inmediato pues t es un valor.

- $t = r\,u$. Hay tres posibles casos:

  1. r no es un valor. Por Lem. 3.3.49 (3), $\exists A', A'' \in \mathcal{T}$ tal que $\pi_r \rhd_{\mathcal{P}} \vdash r : A'$ y $\rhd_{\mathcal{P}} \vdash u : A''$. Por *h.i.* $r \to r'$. Se concluye con $t' = r'\,u$.
  2. r es un valor y u no. Por Lem. 3.3.49 (3), $\exists A', A'' \in \mathcal{T}$ tal que $\rhd_{\mathcal{P}} \vdash r : A'$ y $\pi_u \rhd_{\mathcal{P}} \vdash u : A''$. Por *h.i.* $u \to u'$. Se concluye con $t' = r\,u'$.
  3. r y u son valores. Si r es un data structure el resultado es inmediato pues t también es un valor. Suponer entonces $r = (p_i \to_{\theta_i} s_i)_{i<n}$. Por Lem. 3.3.49 (3) hay dos casos a considerar:

     a) $\exists D, A' \in \mathcal{T}$ tal que $D \mathbin{@} A' \preceq_\mu A$, $\rhd_{\mathcal{P}} \vdash (p_i \to_{\theta_i} s_i)_{i<n} : D$ y $\rhd_{\mathcal{P}} \vdash u : A'$. Luego, por Lem. 3.3.49 (4), $\exists (A_i)_{i<n}, B \in \mathcal{T}$ tal que $\oplus_{i<n} A_i \supset B \preceq_\mu D$. Por Nota 3.3.4, $D \simeq_\mu \oplus_{j<m} D_j$ con $D_j \in \mathcal{D}$ no-unión para todo $j < m$. Más aún, por (S-EQ) y (S-TRANS), $\oplus_{i<n} A_i \supset B \preceq_\mu \oplus_{j<m} D_j$ y, por Lem. 3.3.33, $\oplus_{i<n} A_i \supset B \preceq_\mu D_l$ para algún $l < m$. Esto lleva a una contracción con Teo. 3.3.31 y el hecho de que $D_l \in \mathcal{D}$. Luego, este caso no es posible.

     b) $\exists (B_j)_{j<m}, B' \in \mathcal{T}$ tal que $B' \preceq_\mu A$, $\rhd_{\mathcal{P}} \vdash (p_i \to_{\theta_i} s_i)_{i<n} : \oplus_{j<m} B_j \supset B'$ y $\pi_u \rhd_{\mathcal{P}} \vdash u : B_l$ para algún $l < m$. Por Lem. 3.3.49 (4), $\exists (A_i)_{i<n}, A' \in \mathcal{T}$ tal que $\oplus_{i<n} A_i \supset A' \preceq_\mu \oplus_{j<m} B_j \supset B'$, $(\pi_{p_i} \rhd_{\mathcal{P}} \theta_i \vdash_{\mathsf{p}} p_i : A_i)_{i<n}$, $\mathsf{cmp}([p_i : A_i]_{i<n})$ y $(\pi_{s_i} \rhd_{\mathcal{P}} \vdash \theta_i s_i : A')_{i<n}$. Luego, por Teo. 3.3.31 (3), $\oplus_{j<m} B_j \preceq_\mu \oplus_{i<n} A_i$ y $A' \preceq_\mu B'$. Por otro lado, el Lem. 3.3.50 (2) con π_u garantiza que u es un valor cerrado, *i.e.* un data structure o una abstracción. Por lo tanto, u es una matchable form y $\{\!\{p_i \setminus u\}\!\}$ está decidido para todo $i < n$. Luego, se tienen dos casos posibles:

     - u es un data structure. Luego, por Lem. 3.3.52, existe $D \in \mathcal{D}$ no-unión tal que $D \preceq_\mu B_l$ y $\rhd_{\mathcal{P}} \vdash u : D$.
     - u es una abstracción. Luego, por Lem. 3.3.49 (4), $\exists C', C'' \in \mathcal{T}$ tal que $C' \supset C'' \preceq_\mu B_l$ y $\rhd_{\mathcal{P}} \vdash u : C' \supset C''$.

     En ambos casos se tiene $C \in \mathcal{T}$ no-unión tal que $C \preceq_\mu B_l$ y $\rhd_{\mathcal{P}} \vdash u : C$. Luego, $C \preceq_\mu \oplus_{j<m} B_j$ y, por (S-TRANS), $C \preceq_\mu \oplus_{i<n} A_i$. Más aún, por Lem. 3.3.33, $C \preceq_\mu A_{k'}$ para algún $k' < n$ y, por (T-SUBS), $\pi'_u \rhd_{\mathcal{P}} \vdash u : A_{k'}$. Sea $\mathcal{S} \triangleq \{i < n \mid \{\!\{p_i \setminus u\}\!\} = \rho\}$. Por Lem. 3.4.2 con π'_u y $\pi_{p_{k'}}$, se tiene $\{\!\{p_{k'} \setminus u\}\!\} = \sigma$. Es decir, $\mathcal{S} \neq \emptyset$. Finalmente, basta tomar $k = \min \mathcal{S}$ para concluir con $t' = \{\!\{p_k \setminus u\}\!\} s_k$.

- $t = (p_i \to_{\theta_i} s_i)_{i<n}$. El resultado es inmediato pues t es un valor.

□

3.5. Normalización fuerte

Los tipos recursivos permiten tipar términos no normalizantes, lo que puede ejemplificarse fácilmente apelando a la derivación π_{Y} dada anteriormente para el operador de punto fijo Y:

$$\dfrac{\pi_{\mathsf{Y}} \qquad \dfrac{\dfrac{}{x : A \vdash_{\mathsf{p}} x : A}\,(\text{P-MATCH}) \qquad \dfrac{}{x : A \vdash x : A}\,(\text{T-VAR})}{\vdash x \to_{\{x:A\}} x : A \supset A}\,(\text{T-ABS})}{\vdash \mathsf{Y}\,I : A}\,(\text{T-APP})$$

Es inmediato ver que $\mathsf{Y}\,I$ reduce a Ω expresado en términos de CAP. Luego, por subject reduction se tiene $\pi_\Omega \rhd_{\mathcal{P}} \vdash \Omega : A$ para todo μ-tipo contractivo A.

Existen en la literatura distintas restricciones [Men87, Cop98] que permiten caracterizar normalización fuerte para un sub-conjunto de los términos tipables. Estas restricciones se basan en un análisis

sobre la positividad de las variables ligadas en las expresiones de tipo. Sin embargo, las mismas solo son aplicables en sistemas que apelan a equivalencia débil de tipos recursivos [Cop98, BDS13], en contraposición a la equivalencia fuerte aquí considerada (*cf.* Sec. 3.3.1). El motivo es que la equivalencia fuerte es capaz de igualar tipos positivos con negativos: *e.g.* $\mu X.(X \supset A) \supset A \simeq_\mu \mu X.X \supset A$. Más aún, no parece haber una formulación natural de dichas restricciones para sistemas en presencia de equivalencia fuerte de tipos recursivos [BDS13, pág. 515]. De hecho, si bien se han logrado avances al respecto en el marco de esta investigación [BV17], no fue posible aún dar con la restricción adecuada (independientemente del análisis de positividad sobre las variables) que permita garantizar normalización fuerte para un sub-conjunto de los términos del sistema sin descartar por completo el uso de tipos recursivos.

3.6. Chequeo de tipos

Como primer paso hacia una implementación de un lenguaje de programación que incorpore path polymorphism de manera nativa, se desarrolló un algoritmo de chequeo de tipos adecuado para el sistema propuesto. Hay dos aspectos a tener en cuenta para tal desarrollo: por un lado, las reglas del sistema presentado en la Sec. 3.3.3 no son fácilmente invertibles, en especial por la presencia de la regla (T-SUBS), lo que queda de manifiesto en el Lem. 3.3.49 y dificulta la definición de un algoritmo; por otro, es necesario chequear sub-tipado, lo que a su vez lleva a un chequeo de equivalencia (regla (S-EQ)). El primer aspecto motiva la introducción de una variante del sistema dirigida por sintaxis, la cual es desarrollada en la Sec. 3.6.2, mientras que el segundo lleva a un estudio sobre la factibilidad y complejidad de chequear equivalencia y sub-tipado sobre μ-tipos contractivos, los cuales denotan árboles regulares infinitos. La siguiente sección se centra en este segundo aspecto.

3.6.1. Chequeo de equivalencia y sub-tipado

Existen, a grandes rasgos, dos enfoques para implementar el chequeo de equivalencia y sub-tipado en presencia de tipos recursivos, uno basado en teoría de autómatas y otro basado en una caracterización co-inductiva de las relaciones asociadas. El primero lleva a algoritmos eficientes [PZ01] mientras que el último, aunque no conduce a algoritmos tan eficientes, es más abstracto en naturaleza y por lo tanto más cercano al formalismo en sí mismo. En el caso de sub-tipado de tipos recursivos en presencia de operadores asociativos, conmutativos e idempotentes (ACI), no es claro si puede aplicarse el enfoque basado en autómatas; en particular, el presentado en [PZ01] se sabe que no es aplicable [Zha02]. Sin embargo, el enfoque co-inductivo, desarrollado a continuación, lleva a un algoritmo correcto. Posteriormente, en la Sec. 3.6.3, se explora una alternativa al enfoque basado en autómatas inspirada en [DPR05], adaptándolo para soportar operadores ACI.

Las caracterizaciones co-inductivas de sub-conjuntos de $\mathcal{T} \times \mathcal{T}$ cuya función de generación Φ es *invertible* admiten un algoritmo simple (aunque no necesariamente eficiente) para verificar la pertenencia de un elemento, el cual consisten en "ejecutar Φ al revés" (*cf.* Sec. 1.2.4). Esta estrategia se sustenta en el hecho de que los μ-tipos contractivos garantizan un espacio finito de estados a explorar (*i.e.* sus desdoblamientos infinitos resultan árboles regulares, cuyo conjunto de sub-árboles es finito). La invertibilidad de Φ garantiza además que existe solo una manera de verificar la pertenencia de un elemento a $\nu\Phi$: si $\Phi : \wp(\mathcal{T} \times \mathcal{T}) \to \wp(\mathcal{T} \times \mathcal{T})$ es invertible, entonces para todo par $\langle A, B\rangle$ el conjunto $\{\mathcal{X} \in \wp(\mathcal{T} \times \mathcal{T}) \mid \langle A, B\rangle \in \Phi(\mathcal{X})\}$ es vacío o bien contiene un único elemento que es sub-conjunto de todos los demás.

$$\frac{}{a \simeq_{\vec{\mu}} a}\ (\text{E-REFL-AL}) \qquad \frac{D \simeq_{\vec{\mu}} D' \quad A \simeq_{\vec{\mu}} A'}{D \mathbin{@} A \simeq_{\vec{\mu}} D' \mathbin{@} A'}\ (\text{E-COMP-AL}) \qquad \frac{A \simeq_{\vec{\mu}} A' \quad B \simeq_{\vec{\mu}} B'}{A \supset B \simeq_{\vec{\mu}} A' \supset B'}\ (\text{E-FUNC-T})$$

$$\frac{\mathscr{C}\langle\{V \setminus \mu V.A\}A\rangle \simeq_{\vec{\mu}} B}{\mathscr{C}\langle\mu V.A\rangle \simeq_{\vec{\mu}} B}\ (\text{E-REC-L-AL}) \qquad \frac{A \simeq_{\vec{\mu}} \mathscr{D}\langle\{W \setminus \mu W.B\}B\rangle \quad A \neq \mathscr{C}\langle\mu V.C\rangle}{A \simeq_{\vec{\mu}} \mathscr{D}\langle\mu W.B\rangle}\ (\text{E-REC-R-AL})$$

$$\frac{\begin{array}{lll}(A_i \simeq_{\vec{\mu}} B_{f(i)})_{i<n} & f : [0,n) \to [0,m) & (A_i \neq \mu, \oplus)_{i<n} \\ (A_{g(j)} \simeq_{\vec{\mu}} B_j)_{j<m} & g : [0,m) \to [0,n) & (B_j \neq \mu, \oplus)_{j<m}\end{array} \quad n+m>2}{\bigoplus_{i<n} A_i \simeq_{\mathfrak{T}} \bigoplus_{j<m} B_j}\ (\text{E-UNION-AL})$$

Figura 3.6: Equivalencia algorítmica de μ-tipos contractivos.

Chequeo de equivalencia

Para chequear equivalencia de μ-tipos contractivos con el enfoque actual es necesario dar una caracterización co-inductiva de la relación, al estilo de $\simeq_{\mathfrak{T}}$ pero expresada sobre elementos de $\mathcal{T}$.

Definición 3.6.1. *La relación de* equivalencia algorítmica $\simeq_{\vec{\mu}}$ *de μ-tipos contractivos se define como la interpretación co-inductiva de los esquemas de regla de la Fig. 3.6.*

Las reglas (E-REFL-AL), (E-COMP-AL) y (E-FUNC-AL) son estándar, mientras que (E-UNION-AL) hace uso de las funciones f y g para capturar la naturaleza ACI de la unión. Cabe destacar el uso de una nueva clase de contextos en las reglas (E-REC-L-AL) y (E-REC-R-AL), denominado $\oplus$-*contexto left-to-right* $(\mathscr{C}, \mathscr{D}, \ldots)$, dados por la gramática:

$$\begin{array}{rcll} \mathscr{C} & ::= & \Box & \\ & | & A \oplus \mathscr{C} & \text{si } A \neq \mu, \oplus \\ & | & \mathscr{C} \oplus A & \text{si } A \neq \oplus \end{array}$$

Intuitivamente, $\mathscr{C}\langle A\rangle$ con $A \neq \oplus$ denota un tipo unión $\bigoplus_{i<n} A_i$ tal que $(A_i \neq \oplus)_{i<n}$, $A_k = A$ para algún $k < n$ y $(A_i \neq \mu)_{i<k}$ (*i.e.* todos los elementos a la izquierda de A son tipos no-unión). Estos contextos ayudan a identificar la primera ocurrencia del constructor μ dentro de una unión. A su vez, esto permite garantizar la invertibilidad de la función generadora de $\simeq_{\vec{\mu}}$, pues las condiciones de las reglas (E-UNION-AL), (E-REC-L-AL) y (E-REC-R-AL) resultan mutuamente excluyentes.

Formalmente, sea $\Phi_{\simeq_{\vec{\mu}}} : \wp(\mathfrak{T} \times \mathfrak{T}) \to \wp(\mathfrak{T} \times \mathfrak{T})$ la función generadora asociada a las reglas de la Fig. 3.6, definido como

$$\begin{array}{rcl} \Phi_{\simeq_{\vec{\mu}}}(\mathcal{X}) & \triangleq & \{\langle a, a\rangle \mid a \in \mathcal{V} \cup \mathcal{C}\} \\ & \cup & \{\langle D \mathbin{@} A, D' \mathbin{@} A'\rangle \mid \langle D, D'\rangle, \langle A, A'\rangle \in \mathcal{X}\} \\ & \cup & \{\langle A \supset B, A' \supset B'\rangle \mid \langle A, A'\rangle, \langle B, B'\rangle \in \mathcal{X}\} \\ & \cup & \{\langle \mathscr{C}\langle\mu V.A\rangle, B\rangle \mid \langle \mathscr{C}\langle\{V \setminus \mu V.A\}A\rangle, B\rangle \in \mathcal{X}\} \\ & \cup & \{\langle A, \mathscr{D}\langle\mu W.B\rangle\rangle \mid A \neq \mathscr{C}\langle\mu V.C\rangle, \langle A, \mathscr{D}\langle\{W \setminus \mu W.B\}B\rangle\rangle \in \mathcal{X}\} \\ & \cup & \{\langle \bigoplus_{i<n} A_i, \bigoplus_{j<m} B_j\rangle \mid A_i, B_j \neq \mu, \oplus, n+m>2 \\ & & \quad \exists f : [0,n) \to [0,m).\langle A_i, B_{f(i)}\rangle \in \mathcal{X}, \\ & & \quad \exists g : [0,m) \to [0,n).\langle A_{g(j)}, B_j\rangle \in \mathcal{X}\} \end{array}$$

Luego, $\simeq_{\vec{\mu}} \triangleq \nu\Phi_{\simeq_{\vec{\mu}}}$. Más aún, $\simeq_{\vec{\mu}}$ coincide con $\simeq_{\mu}$ (*cf.* Teo. 3.6.9), lo que permite definir un algoritmo de chequeo para esta última basado en la invertibilidad de $\Phi_{\simeq_{\vec{\mu}}}$. La prueba de esta propiedad recae sobre algunos lemas técnicos y nociones preliminares que se presentan a continuación.

Se definen los $\mu\oplus$-*contextos multi-hole* $(\mathcal{A}, \mathcal{B}, \ldots)$ mediante la gramática:

$$\mathcal{A} ::= \Box \mid \mathcal{A} \oplus \mathcal{A} \mid \mu V.\mathcal{A}$$

Cuando $\mathcal{A}$ consiste únicamente de constructores μ (resp. $\oplus$) se lo denomina μ-*contexto* (resp. $\oplus$-*contexto*). Al igual que con los tipos, se denota con $\bigoplus_{i<n}\mathcal{A}_i$ un $\mu\oplus$-contexto independientemente de cómo estén asociadas las uniones (binarias) en él.

Nota 3.6.2. *Todo μ-tipo contractivo A puede ser expresado unívocamente como un $\mu\oplus$-*contexto maximal *$A = \mathcal{A}\langle A_0, \ldots, A_{n-1}\rangle$, donde $(A_i \neq \mu, \oplus)_{i<n}$.*

Esta caracterización unívoca de μ-tipos contractivos como $\mu\oplus$-contextos maximales aplicados de la forma $\mathcal{A}\langle A_0, \ldots, A_{n-1}\rangle$ (abreviado $\mathcal{A}\langle\vec{A}\rangle$ para simplificar la lectura) resulta conveniente a la hora de probar el isomorfismo entre ambas relaciones de equivalencia $\simeq_\mu$ y $\simeq_{\vec{\mu}}$. En particular, resulta de interés proyectar las sub-expresiones de tipo que ocupan posiciones de $\Box$ en $\mathcal{A}$, para lo cual se definen a continuación la noción de posición proyectable y la operación de proyección.

Definición 3.6.3. *El conjunto de* posiciones proyectables *de un μ-tipo contractivo A se define como* $\mathsf{ppos}(A) \triangleq \{\mathsf{p} \mid A = \mathcal{A}\langle\vec{A}\rangle, \mathcal{A}|_\mathsf{p} = \Box\}$.

Notar que no se requiere maximalidad sobre el $\mu\oplus$-contexto en la definición de posiciones proyectables, por lo cualquier prefijo de una posición proyectable en el $\mu\oplus$-contexto maximal que caracteriza a un μ-tipo contractivo A es a su vez proyectable. Luego, la operación de proyección consiste en retornar la sub-expresión de tipo en la posición (proyectable) deseada, desdoblando previamente aquellos constructores de tipos recursivos que se atraviesen en el camino, evitando así liberar variables ligadas.

Definición 3.6.4. *La* proyección *de un μ-tipo contractivo A en posición $\mathsf{p} \in \mathsf{ppos}(A)$, notada $A{\downarrow_\epsilon}$, se define como:*

$$\begin{array}{rcl} A{\downarrow_\epsilon} & \triangleq & A \\ (\mu V.A'){\downarrow_{\mathsf{0p}}} & \triangleq & \{V \setminus \mu V.A'\}A'{\downarrow_\mathsf{p}} \\ (A_0 \oplus A_1){\downarrow_{i\mathsf{p}}} & \triangleq & A_i{\downarrow_\mathsf{p}} \end{array}$$

Por último, resulta de interés eliminar constructores recursivos de $\mu\oplus$-contextos, obteniendo así $\oplus$-contextos.

Definición 3.6.5. *El* borrado *de constructores μ en un $\mu\oplus$-contexto $\mathcal{A}$, notado $\mathcal{A}^{\setminus\mu}$, se define como:*

$$\begin{array}{rcl} \Box^{\setminus\mu} & \triangleq & \Box \\ (\mu V.\mathcal{A}')^{\setminus\mu} & \triangleq & \mathcal{A}'^{\setminus\mu} \\ (\mathcal{A}_0 \oplus \mathcal{A}_1)^{\setminus\mu} & \triangleq & \mathcal{A}_0{}^{\setminus\mu} \oplus \mathcal{A}_1{}^{\setminus\mu} \end{array}$$

Lema 3.6.6. *Sean $A = \mathcal{A}\langle\vec{A}\rangle$ y $\mathsf{p}_0, \ldots, \mathsf{p}_{n-1} \in \mathsf{ppos}(A)$ las posiciones de $\Box$ en $\mathcal{A}$ (enumeradas de izquierda a derecha). Luego, $[\![A]\!]^\mathfrak{T} = \mathcal{A}^{\setminus\mu}\langle[\![A{\downarrow_{\mathsf{p}_0}}]\!]^\mathfrak{T}, \ldots, [\![A{\downarrow_{\mathsf{p}_{n-1}}}]\!]^\mathfrak{T}\rangle$.*

El siguiente lema permite deducir la forma de un μ-tipo contractivo cuya interpretación como árbol (infinito) regular sea una unión-maximal.

Lema 3.6.7. *Sea A un μ-tipo contractivo tal que $[\![A]\!]^\mathfrak{T} = \bigoplus_{i<n}\mathcal{A}_i$ con $(\mathcal{A}_i \neq \oplus)_{i<n}$. Luego, existen $n' \leq n$, $\oplus$-contextos maximales $\mathcal{A}$ y $(\mathcal{A}_l)_{l<n'}$, μ-tipos contractivos $(A_l \neq \oplus)_{l<n'}$ y funciones $b, e : [0, n') \to [0, n)$ tales que $A = \mathcal{A}\langle\vec{A}\rangle$ y $([\![A_l]\!]^\mathfrak{T} = \mathcal{A}_l\langle\mathcal{A}_{b(l)}, \ldots, \mathcal{A}_{e(l)}\rangle)_{l<n'}$.*

Estos resultados auxiliares sirven para demostrar que $\simeq_{\vec{\mu}}$ puede ser determinado mediante $\simeq_{\mathfrak{T}}$ para todos los truncados finitos de los árboles asociados.

Lema 3.6.8. *$A \simeq_{\vec{\mu}} B$ sii $\forall k \in \mathbb{N}. \llbracket A \rrbracket^{\mathfrak{T}} \rfloor_k \simeq_{\mathfrak{T}} \llbracket B \rrbracket^{\mathfrak{T}} \rfloor_k$.*

Finalmente, esto lleva al resultado de isomorfismo buscado entre $\simeq_{\mu}$ y $\simeq_{\vec{\mu}}$.

Teorema 3.6.9. *$A \simeq_{\mu} B$ sii $A \simeq_{\vec{\mu}} B$.*

Demostración. El resultado es consecuencia inmediata de los Lem. 3.3.27 y 3.6.8: $A \simeq_{\mu} B$ sii $\forall k \in \mathbb{N}. \llbracket A \rrbracket^{\mathfrak{T}} \rfloor_k \simeq_{\mathfrak{T}} \llbracket B \rrbracket^{\mathfrak{T}} \rfloor_k$ sii $A \simeq_{\vec{\mu}} B$. □

Por lo tanto, puede chequearse $\simeq_{\mu}$ apelando a la invertibilidad de la función generadora $\Phi_{\simeq_{\vec{\mu}}}$. El algoritmo resultante es presentado en la Fig. 3.7, donde $\texttt{seq}\ e_0, \ldots, e_n$ es un comando que evalúa de manera secuencial cada uno de sus argumentos retornando el valor del primero que tiene éxito. La evaluación de la expresión $\texttt{eqtype}(\emptyset, A, B)$ tiene dos resultados posibles: o bien falla (retornando `fail`), lo que significa $A \not\simeq_{\vec{\mu}} B$; o retorna un conjunto $\Phi_{\simeq_{\vec{\mu}}}$-denso $S \in \wp(\mathcal{T} \times \mathcal{T})$ tal que $\langle A, B \rangle \in S$, lo que garantiza $A \simeq_{\vec{\mu}} B$ y, por lo tanto, $A \simeq_{\mu} B$.

Chequeo de sub-tipado

El enfoque para chequeo de sub-tipado es similar al presentado para la equivalencia. En este caso se define una relación co-inductiva de sub-tipado sobre μ-tipos contractivos, que resulta de adaptar de manera directa la relación $\simeq_{\mathfrak{T}}$ (presentada anteriormente sobre árboles infinitos) para manipular el constructor adicional μ.

Definición 3.6.10. *La relación de* sub-tipado algorítmica $\preceq_{\vec{\mu}}$ *de μ-tipos contractivos se define como la interpretación co-inductiva de los esquemas de regla de la Fig. 3.8.*

Formalmente, sea $\Phi_{\preceq_{\vec{\mu}}} : \wp(\mathfrak{T} \times \mathfrak{T}) \to \wp(\mathfrak{T} \times \mathfrak{T})$ la función generadora asociada a las reglas de la Fig. 3.8, definido como

$$
\begin{array}{rcl}
\Phi_{\preceq_{\vec{\mu}}}(\mathcal{X}) & \triangleq & \{\langle a, a\rangle \mid a \in \mathcal{V} \cup \mathcal{C}\} \\
& \cup & \{\langle D \mathbin{@} A, D' \mathbin{@} A'\rangle \mid \langle D, D'\rangle, \langle A, A'\rangle \in \mathcal{X}\} \\
& \cup & \{\langle A \supset B, A' \supset B'\rangle \mid \langle A', A\rangle, \langle B, B'\rangle \in \mathcal{X}\} \\
& \cup & \{\langle \mu V.A, B\rangle \mid \langle \{V \setminus \mu V.A\}A, B\rangle \in \mathcal{X}\} \\
& \cup & \{\langle A, \mu W.B\rangle \mid A \neq \mu, \langle A, \{W \setminus \mu W.B\}B\rangle \in \mathcal{X}\} \\
& \cup & \{\langle \bigoplus_{i<n} A_i, B\rangle \mid A_i \neq \oplus, B \neq \mu, n > 1, \langle A_i, B\rangle \in \mathcal{X}\} \\
& \cup & \{\langle A, \bigoplus_{j<m} B_j\rangle \mid A \neq \mu, \oplus, B_j \neq \oplus, m > 1 \\
& & \quad \exists k < m. \langle A, B_k\rangle \in \mathcal{X}\}
\end{array}
$$

Luego, $\preceq_{\vec{\mu}} \triangleq \nu\Phi_{\preceq_{\vec{\mu}}}$. Para mostrar el isomorfismo entre $\preceq_{\mu}$ y $\preceq_{\vec{\mu}}$ se apela al Teo. 3.3.30 y el Lem. 3.3.21 junto con los siguientes resultados auxiliares.

El siguiente lema permite relacionar diferentes proyecciones no-unión de tipos unión-maximal que están relacionados en un conjunto $\Phi_{\preceq_{\vec{\mu}}}$-denso.

Lema 3.6.11. *Sean $\langle A, B\rangle \in \mathcal{S}$ con $\mathcal{S} \subseteq \Phi_{\preceq_{\vec{\mu}}}(\mathcal{S})$, $(A_i)_{i<n}, (B_j)_{j<m}$ μ-tipos contractivos y $\mathcal{A}, \mathcal{B}$ $\mu\oplus$-contextos maximales tal que $A = \mathcal{A}\langle\vec{A}\rangle$ y $B = \mathcal{B}\langle\vec{B}\rangle$. Luego, existe una función $f : [0, n) \to [0, m)$ tal que $(\langle A{\downarrow_{\mathsf{p}_i}}, B{\downarrow_{\mathsf{q}_{f(i)}}}\rangle \in \mathcal{S})_{i<n}$, donde $\mathsf{p}_0, \ldots, \mathsf{p}_{n-1} \in \mathsf{ppos}(A)$ y $\mathsf{q}_0, \ldots, \mathsf{q}_{m-1} \in \mathsf{ppos}(B)$ son las posiciones de $\Box$ en $\mathcal{A}$ y $\mathcal{B}$ respectivamente.*

Lema 3.6.12. *$A \preceq_{\vec{\mu}} B$ sii $\forall k \in \mathbb{N}. \llbracket A \rrbracket^{\mathfrak{T}} \rfloor_k \preceq_{\mathfrak{T}} \llbracket B \rrbracket^{\mathfrak{T}} \rfloor_k$.*

```
eqtype(S, A, B) ≜
    if ⟨A, B⟩ ∈ S
      then S
      else let S_0 = S ∪ {⟨A, B⟩} in
        case ⟨A, B⟩ of
          ⟨a, a⟩ →
            S_0
          ⟨A' @ A'', B' @ B''⟩ →
            if A', B' son datatypes
              then let S_1 = eqtype(S_0, A', B') in
                eqtype(S_1, A'', B'')
              else fail
          ⟨A' ⊃ A'', B' ⊃ B''⟩ →
            let S_1 = eqtype(S_0, A', B') in
              eqtype(S_1, A'', B'')
          ⟨𝒞⟨μV.A'⟩, B⟩ →
            eqtype(S_0, 𝒞⟨{V \ μV.A'}A'⟩, B)
          ⟨A, 𝒟⟨μW.B'⟩⟩ →
            eqtype(S_0, A, 𝒟⟨{W \ μW.B'}B'⟩)
          ⟨⊕_{i<n} A_i, ⊕_{j<m} B_j⟩ →
            let S_1 = (seq eqtype(S_0, A_0, B_0), ..., eqtype(S_0, A_0, B_{m-1})) in
            ...
            let S_n = (seq eqtype(S_{n-1}, A_{n-1}, B_0), ..., eqtype(S_{n-1}, A_{n-1}, B_{m-1})) in
            let S_{n+1} = (seq eqtype(S_n, A_0, B_0), ..., eqtype(S_n, A_{n-1}, B_0)) in
            ...
            let S_{n+m-1} = (seq eqtype(S_{n+m-2}, A_0, B_{m-2}), ..., eqtype(S_{n+m-2}, A_{n-1}, B_{m-2})) in
              seq eqtype(S_{n+m-1}, A_0, B_{m-1}), ..., eqtype(S_{n+m-1}, A_{n-1}, B_{m-1})
          otherwise →
            fail
```

Figura 3.7: Algoritmo co-inductivo para chequeo de equivalencia.

$$\frac{}{a \preceq_{\vec{\mu}} a}\ (\textsc{s-refl-al})$$

$$\frac{D \preceq_{\vec{\mu}} D' \quad A \preceq_{\vec{\mu}} A'}{D \mathbin{@} A \preceq_{\vec{\mu}} D' \mathbin{@} A'}\ (\textsc{s-comp-al}) \qquad \frac{A' \preceq_{\vec{\mu}} A \quad B \preceq_{\vec{\mu}} B'}{A \supset B \preceq_{\vec{\mu}} A' \supset B'}\ (\textsc{s-func-al})$$

$$\frac{\{V \setminus \mu V.A\}A \preceq_{\vec{\mu}} B}{\mu V.A \preceq_{\vec{\mu}} B}\ (\textsc{s-rec-l-al}) \qquad \frac{A \preceq_{\vec{\mu}} \{W \setminus \mu W.B\}B \quad A \neq \mu}{A \preceq_{\vec{\mu}} \mu W.B}\ (\textsc{s-rec-r-al})$$

$$\frac{(A_i \preceq_{\vec{\mu}} B)_{i<n} \quad n > 1 \quad B \neq \mu \quad (A_i \neq \oplus)_{i<n}}{\bigoplus_{i<n} A_i \preceq_{\vec{\mu}} B}\ (\textsc{s-union-l-al})$$

$$\frac{A \preceq_{\vec{\mu}} B_k \quad k < m \quad m > 1 \quad A \neq \mu, \oplus \quad (B_j \neq \oplus)_{j<m}}{A \preceq_{\vec{\mu}} \bigoplus_{j<m} B_j}\ (\textsc{s-union-r-al})$$

Figura 3.8: Sub-tipado algorítmico de μ-tipos contractivos.

Finalmente, esto lleva al resultado de isomorfismo buscado entre $\preceq_\mu$ y $\preceq_{\vec{\mu}}$.

Teorema 3.6.13. *$A \preceq_\mu B$ sii $A \preceq_{\vec{\mu}} B$.*

Demostración. Por Teo. 3.3.30, Lem. 3.3.21 y 3.6.12: $A \preceq_\mu B$ sii $[\![A]\!]^{\mathfrak{T}} \preceq_{\mathfrak{T}} [\![B]\!]^{\mathfrak{T}}$ sii $\forall k \in \mathbb{N}. \lfloor [\![A]\!]^{\mathfrak{T}} \rfloor_k \preceq_{\mathfrak{T}} \lfloor [\![B]\!]^{\mathfrak{T}} \rfloor_k$ sii $A \preceq_{\vec{\mu}} B$. □

Desafortunadamente la función generadora $\Phi_{\preceq_{\vec{\mu}}}$ no es invertible: notar que (S-UNION-R-AL) se solapa consigo misma. Por ejemplo, $\mathbb{c} \preceq_{\vec{\mu}} (\mathbb{c} \oplus \mathbb{d}) \oplus (\mathbb{e} \oplus \mathbb{c})$ pertenece a dos conjuntos $\Phi_{\preceq_{\vec{\mu}}}$-densos:

$$\mathcal{X}_0 = \{\langle \mathbb{c}, (\mathbb{c} \oplus \mathbb{d}) \oplus (\mathbb{e} \oplus \mathbb{c})\rangle, \langle \mathbb{c}, \mathbb{c} \oplus \mathbb{d}\rangle, \langle \mathbb{c}, \mathbb{c}\rangle\}$$
$$\mathcal{X}_1 = \{\langle \mathbb{c}, (\mathbb{c} \oplus \mathbb{d}) \oplus (\mathbb{e} \oplus \mathbb{c})\rangle, \langle \mathbb{c}, \mathbb{e} \oplus \mathbb{c}\rangle, \langle \mathbb{c}, \mathbb{c}\rangle\}$$

Sin embargo, dado que ésta es la única causa de pérdida de la invertibilidad, puede fácilmente adaptarse el enfoque para obtener un algoritmo de chequeo de sub-tipado adecuado: en el caso de la regla (S-UNION-R-AL) simplemente se verifican todas las combinaciones posibles. El pseudo-código de tal algoritmo es presentado en la Fig. 3.9. Al igual que $\mathsf{eqtype}(\emptyset, A, B)$, la evaluación de $\mathsf{subtype}(\emptyset, A, B)$ puede fallar (*i.e.* $A \not\preceq_{\vec{\mu}} B$) o retornar un conjunto $\Phi_{\preceq_{\vec{\mu}}}$-denso que garantiza $A \preceq_{\vec{\mu}} B$, y por lo tanto $A \preceq_\mu B$.

3.6.2. Algoritmo de chequeo de tipos

Esta sección introduce una variante dirigida por sintaxis del sistema de tipos $\mathcal{P}$. La misma se obtiene esencialmente tomando las regla de la Fig. 3.5 y eliminando (T-SUBS). Esto conlleva, a su vez, la adecuada modificación de las reglas (T-ABS) y (T-APP), para preservar la expresividad del sistema.

La relación de sub-tipado cumple dos roles fundamentales en $\mathcal{P}$. Por un lado el de compatibilizar los tipos del dominio de una función y sus argumentos, de modo de poder aplicar adecuadamente ambas expresiones. Por otro lado, en (T-ABS) se requiere que el cuerpo de todas las ramas de una abstracción tengan el mismo tipo, lo cual no es una restricción en presencia de sub-tipado, pues siempre se puede generalizar el tipo individual de cada rama mediante uniones. Ambas situaciones se ilustran en la derivación de tipo π_{upd} dada en Sec. 3.3.4, donde a cada rama de la abstracción interna se le asigna un tipo propio ($\mathbb{l} \mathbin{@} B$, $F_B \mathbin{@} F_B$ y $\mathbb{nil} \oplus \mathbb{cons} \oplus \mathbb{node}$ resp.) y todos estos son igualados a F_B por sub-tipado;

$$
\begin{array}{l}
\mathsf{subtype}(\mathcal{S}, A, B) \triangleq \\
\quad \texttt{if } \langle A, B\rangle \in \mathcal{S} \\
\quad\quad \texttt{then } S \\
\quad\quad \texttt{else let } \mathcal{S}_0 = \mathcal{S} \cup \{\langle A, B\rangle\} \texttt{ in} \\
\quad\quad\quad \texttt{case } \langle A, B\rangle \texttt{ of} \\
\quad\quad\quad\quad \langle a, a\rangle \rightarrow \\
\quad\quad\quad\quad\quad \mathcal{S}_0 \\
\quad\quad\quad\quad \langle A' \mathbin{@} A'', B' \mathbin{@} B''\rangle \rightarrow \\
\quad\quad\quad\quad\quad \texttt{if } A', B' \text{ son datatypes} \\
\quad\quad\quad\quad\quad\quad \texttt{then let } \mathcal{S}_1 = \mathsf{subtype}(\mathcal{S}_0, A', A'') \texttt{ in} \\
\quad\quad\quad\quad\quad\quad\quad \mathsf{subtype}(\mathcal{S}_1, B', B'') \\
\quad\quad\quad\quad\quad\quad \texttt{else fail} \\
\quad\quad\quad\quad \langle A' \supset A'', B' \supset B''\rangle \rightarrow \\
\quad\quad\quad\quad\quad \texttt{let } \mathcal{S}_1 = \mathsf{subtype}(\mathcal{S}_0, B', A') \texttt{ in} \\
\quad\quad\quad\quad\quad\quad \mathsf{subtype}(\mathcal{S}_1, A'', B'') \\
\quad\quad\quad\quad \langle \mu V.A', B\rangle \rightarrow \\
\quad\quad\quad\quad\quad \mathsf{subtype}(\mathcal{S}_0, \{V \setminus \mu V.A'\}A', B) \\
\quad\quad\quad\quad \langle A, \mu W.B'\rangle \rightarrow \\
\quad\quad\quad\quad\quad \mathsf{subtype}(\mathcal{S}_0, A, \{W \setminus \mu W.B'\}B') \\
\quad\quad\quad\quad \langle \oplus_{i<n} A_i, B\rangle \rightarrow \\
\quad\quad\quad\quad\quad \texttt{let } \mathcal{S}_1 = \mathsf{subtype}(\mathcal{S}_0, A_0, B) \texttt{ in} \\
\quad\quad\quad\quad\quad \texttt{let } \mathcal{S}_2 = \mathsf{subtype}(\mathcal{S}_1, A_1, B) \texttt{ in} \\
\quad\quad\quad\quad\quad \ldots \\
\quad\quad\quad\quad\quad \texttt{let } \mathcal{S}_{n-1} = \mathsf{subtype}(\mathcal{S}_{n-2}, A_{n-2}, B) \texttt{ in} \\
\quad\quad\quad\quad\quad\quad \mathsf{subtype}(\mathcal{S}_{n-1}, A_{n-1}, B) \\
\quad\quad\quad\quad \langle A, \oplus_{j<m} B_j\rangle \rightarrow \\
\quad\quad\quad\quad\quad \mathsf{seq}\ \mathsf{subtype}(\mathcal{S}_0, A, B_0), \ldots, \mathsf{subtype}(\mathcal{S}_0, A, B_{m-1}) \\
\quad\quad\quad\quad \texttt{otherwise} \rightarrow \\
\quad\quad\quad\quad\quad \texttt{fail}
\end{array}
$$

Figura 3.9: Algoritmo co-inductivo para chequeo de sub-tipado.

Reglas de tipado para patrones

$$\frac{}{x : A \vdash_{\mathsf{p}} x : A}\ (\textsc{p-match-al}) \qquad \frac{}{\vdash_{\mathsf{p}} \mathsf{c} : \mathbb{c}}\ (\textsc{p-const-al}) \qquad \frac{\theta \vdash_{\mathsf{p}} p : D \quad \theta' \vdash_{\mathsf{p}} q : A}{\theta; \theta' \vdash_{\mathsf{p}} p\, q : D \mathbin{@} A}\ (\textsc{p-comp-al})$$

Reglas de tipado para términos

$$\frac{\Gamma(x) = A}{\Gamma \vdash x : A}\ (\textsc{t-var-al}) \qquad \frac{}{\Gamma \vdash \mathsf{c} : \mathbb{c}}\ (\textsc{t-const-al}) \qquad \frac{\Gamma \vdash r : D \quad \Gamma \vdash u : A}{\Gamma \vdash r\, u : D \mathbin{@} A}\ (\textsc{t-comp-al})$$

$$\frac{(\theta_i \vdash_{\mathsf{p}} p_i : A_i)_{i<n} \quad \mathsf{cmp}([p_i : A_i]_{i<n}) \quad (\Gamma; \theta_i \vdash s_i : B_i)_{i<n}}{\Gamma \vdash (p_i \to_{\theta_i} s_i)_{i<n} : \bigoplus_{i<n} A_i \supset \bigoplus_{i<n} B_i}\ (\textsc{t-abs-al})$$

$$\frac{\begin{array}{l}\Gamma \vdash r : A \quad A \simeq_\mu \bigoplus_{i<n}(A_i \supset B_i) \\ \Gamma \vdash u : C \quad (\vdash C \preceq_\mu A_i)_{i<n}\end{array}}{\Gamma \vdash r\, u : \bigoplus_{i<n} B_i}\ (\textsc{t-app-al})$$

Figura 3.10: Reglas de tipado del sistema $\mathcal{C}$ para CAP.

del mismo modo que el dominio de esta abstracción es F_A, pero resulta aplicable tanto a List_A como a Tree_A apelando, nuevamente, al sub-tipado.

Bajo estas consideraciones se define el *sistema de tipos algorítmico* de CAP, llamado $\mathcal{C}$, con las reglas presentadas en la Fig. 3.10.

Las reglas (p-match-al), (p-const-al), (p-comp-al), (t-var-al), (t-const-al) y (t-comp-al) son idénticas a sus contra-partes en el sistema $\mathcal{P}$, mientras que (t-abs-al) y (t-app-al) introducen las modificaciones necesarias para contemplar las situaciones descritas anteriormente. Cabe destacar que para (t-abs-al) basta con tomar la unión de los tipos individuales del cuerpo de cada rama, sin necesidad de apelar a la relación de sub-tipado. Sin embargo, (t-app-al) involucra el chequeo tanto por sub-tipado como por equivalencia. La complejidad de esta regla se debe a que, desafortunadamente, la variante ingenua de (t-app)

$$\frac{\Gamma \vdash r : (\bigoplus_{i<n} A_i) \supset B \quad \Gamma \vdash u : A' \quad \vdash A' \preceq_\mu A_k \quad k < n}{\Gamma \vdash r\, u : B}$$

no resulta completa. En otras palabras, existen derivaciones de la forma $\rhd_{\mathcal{P}} \Gamma \vdash t : A$ para las cuales no existe $A' \preceq_\mu A$ tal que $\Gamma \vdash t : A'$ pueda derivarse con esta variante. Por ejemplo, tomando $\Gamma = \{x : (\mathbb{c} \oplus \mathbb{e} \supset \mathbb{d}) \oplus (\mathbb{c} \oplus \mathbb{f} \supset \mathbb{d})\}$, $t = x\,\mathsf{c}$ y $A = \mathbb{d}$. La derivación $\rhd_{\mathcal{P}} \Gamma \vdash x\,\mathsf{c} : \mathbb{d}$ apela al hecho de que $(\mathbb{c} \oplus \mathbb{e} \supset \mathbb{d}) \oplus (\mathbb{c} \oplus \mathbb{f} \supset \mathbb{d}) \preceq_\mu \mathbb{c} \supset \mathbb{d}$ y, por (t-subs), se tiene $\rhd_{\mathcal{P}} \Gamma \vdash x : \mathbb{c} \supset \mathbb{d}$, concluyendo así con (t-app). Sin embargo, en $\Gamma \vdash x : (\mathbb{c} \oplus \mathbb{e} \supset \mathbb{d}) \oplus (\mathbb{c} \oplus \mathbb{f} \supset \mathbb{d})$ no puede aplicarse la regla ingenua dado que ésta requiere un tipo funcional para x, el cual no puede obtenerse en la ausencia de (t-subs). Más aún, en general de $\Gamma \vdash r : A$ y $A \preceq_\mu \bigoplus_{i<n} A_i \supset B$ no puede inferirse que A sea un tipo funcional por la posible presencia de uniones. Volviendo al ejemplo, de $\rhd_{\mathcal{C}} \Gamma \vdash x : (\mathbb{c} \oplus \mathbb{e} \supset \mathbb{d}) \oplus (\mathbb{c} \oplus \mathbb{f} \supset \mathbb{d})$ y $\rhd_{\mathcal{C}} \Gamma \vdash \mathsf{c} : \mathbb{c}$ sí puede deducirse $\rhd_{\mathcal{C}} \Gamma \vdash x\,\mathsf{c} : \mathbb{d}$ por (t-app-al), pues $\mathbb{c} \preceq_\mu \mathbb{c} \oplus \mathbb{e}$ y $\mathbb{c} \preceq_\mu \mathbb{c} \oplus \mathbb{f}$.

El sistema $\mathcal{C}$ se demuestra correcto y completo con respecto a $\mathcal{P}$. Cabe destacar, sin embargo, la forma de completitud derivada, también conocida como *tipado minimal* [Pie02]. El sistema de tipos $\mathcal{P}$ original permite asignar distintas expresiones de tipo a un mismo término, mientras que el sistema algorítmico $\mathcal{C}$ asigna a lo sumo una expresión. Por lo tanto, es claro que la completitud clásica no puede satisfacerse. Se muestra en su lugar que si un término tiene un tipo A en $\mathcal{P}$, entonces tiene

un tipo *más específico* A' en $\mathcal{C}$, en el sentido de que $A' \preceq_\mu A$. En otras palabras, el sistema de tipos algorítmico asigna a cada término tipable su menor tipo posible. Es llamado alternativamente tipado minimal dado que, junto al resultado de corrección, permite mostrar que todo término tipado en $\mathcal{P}$ tiene un tipo mínimo.

Teorema 3.6.14.

1. *Si* $\pi \rhd_{\mathcal{C}} \Gamma \vdash t : A$*, entonces existe* $\pi' \rhd_{\mathcal{P}} \Gamma \vdash t : A$*.*
2. *Si* $\pi \rhd_{\mathcal{P}} \Gamma \vdash t : A$*, entonces existe* $A' \preceq_\mu A$ *tal que* $\pi' \rhd_{\mathcal{C}} \Gamma \vdash t : A'$*.*

Demostración. Notar que para el caso de los patrones se tiene $\pi \rhd_{\mathcal{C}} \theta \vdash_{\mathsf{p}} p : A$ sii $\pi' \rhd_{\mathcal{P}} \theta \vdash_{\mathsf{p}} p : A$ trivialmente, pues las reglas de cada sistema se corresponden una a una.

1. Por inducción en $\pi \rhd_{\mathcal{C}} \Gamma \vdash t : A$ analizando la última regla aplicada.
   - (T-VAR-AL). Luego, $t = x$ y $\Gamma(x) = A$. Se concluye por (T-VAR).
   - (T-CONST-AL). Luego, $t = \mathsf{c}$ y $A = \mathbb{c}$. Se concluye por (T-CONST).
   - (T-COMP-AL). Luego, $t = r\,u$ y $A = D \mathbin{@} A'$ con $\pi_r \rhd_{\mathcal{C}} \Gamma \vdash r : D$ y $\pi_u \rhd_{\mathcal{C}} \Gamma \vdash u : A'$. Por *h.i.* existen $\pi'_r \rhd_{\mathcal{P}} \Gamma \vdash r : D$ y $\pi'_u \rhd_{\mathcal{P}} \Gamma \vdash u : A'$. Luego, se concluye por (T-COMP).
   - (T-ABS-AL). Luego, $t = (p_i \to_{\theta_i} s_i)_{i<n}$ y $A = \bigoplus_{i<n} A_i \supset \bigoplus_{i<n} B_i$ con $(\pi_{p_i} \rhd_{\mathcal{C}} \theta_i \vdash_{\mathsf{p}} p_i : A_i)_{i<n}$, $\mathsf{cmp}([p_i : A_i]_{i<n})$ y $(\pi_{s_i} \rhd_{\mathcal{C}} \Gamma; \theta_i \vdash s_i : B_i)_{i<n}$. Por *h.i.* se tiene $(\pi'_{s_i} \rhd_{\mathcal{P}} \Gamma; \theta_i \vdash s_i : B_i)_{i<n}$. Más aún, por (S-UNION-R1) y (S-UNION-R2), $(B_i \preceq_\mu \bigoplus_{i<n} B_i)_{i<n}$. Luego, por (T-SUBS), se tiene $(\pi'_{s_i} \rhd_{\mathcal{P}} \Gamma; \theta_i \vdash s_i : \bigoplus_{i<n} B_i)_{i<n}$. Finalmente, se concluye por (T-ABS).
   - (T-APP-AL). Luego, $t = r\,u$ y $A = \bigoplus_{i<n} B_i$ con $\pi_r \rhd_{\mathcal{C}} \Gamma \vdash r : A'$, $\pi_u \rhd_{\mathcal{C}} \Gamma \vdash u : C$, $A' \simeq_\mu \bigoplus_{i<n}(A_i \supset B_i)$ y $(C \preceq_\mu A_i)_{i<n}$. Por *h.i.* existen $\pi'_r \rhd_{\mathcal{P}} \Gamma \vdash r : A'$ y $\pi'_u \rhd_{\mathcal{P}} \Gamma \vdash u : C$. Más aún, por (S-UNION-R1) y (S-UNION-R2), se tiene $(B_j \preceq_\mu \bigoplus_{i<n} B_i)_{j<n}$ y, por (S-FUNC), se tiene $(A_j \supset B_j \preceq_\mu C \supset \bigoplus_{i<n} B_i)_{j<n}$. Luego, por (S-EQ) y (S-UNION-L), $A' \preceq_\mu \bigoplus_{i<n}(A_i \supset B_i) \preceq_\mu C \supset \bigoplus_{i<n} B_i$. Finalmente, por (T-SUBS), $\pi'_r \rhd_{\mathcal{P}} \Gamma \vdash r : C \supset \bigoplus_{i<n} B_i$ y se concluye por (T-APP).
2. Por inducción en $\pi \rhd_{\mathcal{P}} \Gamma \vdash t : A$ analizando la última regla aplicada.
   - (T-VAR). Luego, $t = x$ y $\Gamma(x) = A$. Se concluye por (T-VAR-AL).
   - (T-CONST). Luego, $t = \mathsf{c}$ y $A = \mathbb{c}$. Se concluye por (T-CONST-AL).
   - (T-COMP). Luego, $t = r\,u$ y $A = D \mathbin{@} A'$ con $\pi_r \rhd_{\mathcal{P}} \Gamma \vdash r : D$ y $\pi_u \rhd_{\mathcal{P}} \Gamma \vdash u : A'$. Por *h.i.* existe $D' \preceq_\mu D$ y $A'' \preceq_\mu A'$ tal que $\pi'_r \rhd_{\mathcal{C}} \Gamma \vdash r : D'$ y $\pi'_u \rhd_{\mathcal{C}} \Gamma \vdash u : A''$. Más aún, por (S-COMP), $D' \mathbin{@} A'' \preceq_\mu A$. Luego, se concluye por (T-COMP-AL).
   - (T-ABS). Luego, $t = (p_i \to_{\theta_i} s_i)_{i<n}$ y $A = \bigoplus_{i<n} A_i \supset B$ con $(\pi_{p_i} \rhd_{\mathcal{P}} \theta_i \vdash_{\mathsf{p}} p_i : A_i)_{i<n}$, $\mathsf{cmp}([p_i : A_i]_{i<n})$ y $(\pi_{s_i} \rhd_{\mathcal{P}} \Gamma; \theta_i \vdash s_i : B)_{i<n}$. Por *h.i.* existen tipos $(B_i \preceq_\mu B)_{i<n}$ tal que $(\pi'_{s_i} \rhd_{\mathcal{C}} \Gamma; \theta_i \vdash s_i : B_i)_{i<n}$. Luego, se concluye por (T-ABS-AL) con $A' = \bigoplus_{i<n} A_i \supset \bigoplus_{i<n} B_i$. Notar que, por (S-UNION-L), $\bigoplus_{i<n} B_i \preceq_\mu B$ y, por (S-FUNC), $A' \preceq_\mu A$.
   - (T-APP). Luego, $t = r\,u$ con $\pi_r \rhd_{\mathcal{P}} \Gamma \vdash r : \bigoplus_{i<n} A_i \supset A$ y $\pi_u \rhd_{\mathcal{P}} \Gamma \vdash u : A_k$ para algún $k < n$. Por *h.i.* existe $B \preceq_\mu \bigoplus_{i<n} A_i \supset A$ y $C \preceq_\mu A_k$ tal que $\pi'_r \rhd_{\mathcal{C}} \Gamma \vdash r : B$ y $\pi'_u \rhd_{\mathcal{C}} \Gamma \vdash u : C$. Por Nota. 3.3.4, existen $(B_j)_{j<m}$ no-unión tal que $B \simeq_\mu \bigoplus_{j<m} B_j$. Más aún, por (S-EQ) y (S-TRANS), $\bigoplus_{j<m} B_j \preceq_\mu \bigoplus_{i<n} A_i \supset A$ y, por Teo. 3.6.9 y (S-UNION-L-AL), $(B_j \preceq_\mu \bigoplus_{i<n} A_i \supset A)_{j<m}$. Luego, por Teo. 3.3.31 (3), se tienen $(B_j = B'_j \supset B''_j)_{j<m}$ tal que $(\bigoplus_{i<n} A_i \preceq_\mu B'_j)_{j<m}$ y $(B''_j \preceq_\mu A)_{j<m}$. Notar, en particular, que $B \simeq_\mu \bigoplus_{j<m}(B'_j \supset B''_j)$. Por otro lado, $A_k \preceq_\mu \bigoplus_{i<n} A_i$ por (S-UNION-R1) y (S-UNION-R2). Entonces, por (S-TRANS) se tiene también $(C \preceq_\mu B'_j)_{j<m}$. Finalmente, se concluye por (T-APP-AL) con $A' = \bigoplus_{j<m} B''_j$. Notar que $A' \preceq_\mu A$ por (S-UNION-L).

$$
\begin{array}{rcl}
\mathsf{tcp}(\Gamma, x) & \triangleq & \texttt{if } x \in \mathsf{dom}(\Gamma) \texttt{ then } \Gamma(x) \texttt{ else fail} \\
\mathsf{tcp}(\Gamma, \mathsf{c}) & \triangleq & \mathbb{c} \\
\mathsf{tcp}(\Gamma, p\,q) & \triangleq & \texttt{let } A = \mathsf{tcp}(\Gamma, p), B = \mathsf{tcp}(\Gamma, q) \texttt{ in} \\
 & & \quad \texttt{if } A \text{ es un datatype } \texttt{then } A \mathbin{@} B \texttt{ else fail}
\end{array}
$$

$$
\begin{array}{rcl}
\mathsf{tc}(\Gamma, x) & \triangleq & \texttt{if } x \in \mathsf{dom}(\Gamma) \texttt{ then } \Gamma(x) \texttt{ else fail} \\
\mathsf{tc}(\Gamma, \mathsf{c}) & \triangleq & \mathbb{c} \\
\mathsf{tc}(\Gamma, (p_i \to_{\theta_i} s_i)_{i<n}) & \triangleq & \texttt{let } A_i = \mathsf{tcp}(\theta_i, p_i), B_i = \mathsf{tc}(\Gamma, \theta_i, s_i) \texttt{ in} \\
 & & \quad \texttt{if } \forall i < n. \forall j \in [i+1, n]. \mathsf{compat}(p_i : A_i, p_j : A_j) \\
 & & \quad\quad \texttt{then } \oplus_{i<n} A_i \supset \oplus_{i<n} B_i \texttt{ else fail} \\
\mathsf{tc}(\Gamma, r\,u) & \triangleq & \texttt{let } A = \mathsf{tc}(\Gamma, r), C = \mathsf{tc}(\Gamma, u) \texttt{ in} \\
 & & \quad \texttt{if } A \text{ es un datatype } \texttt{then } A \mathbin{@} C \\
 & & \quad\quad \texttt{else let } \oplus_{i<n}(A_i \supset B_i) = \mathsf{unfold}(A) \texttt{ in} \\
 & & \quad\quad\quad \texttt{if } \forall i < n. \mathsf{subtype}(\emptyset, C, A_i) \\
 & & \quad\quad\quad\quad \texttt{then } \oplus_{i<n} B_i \texttt{ else fail}
\end{array}
$$

Figura 3.11: Algoritmo de chequeo de tipos para CAP.

- (T-SUBS). Luego, $\pi_t \rhd_{\mathcal{P}} \Gamma \vdash t : B$ con $B \preceq_\mu A$. Por *h.i.* existe $A' \preceq_\mu B$ tal que $\pi'_t \rhd_{\mathcal{C}} \Gamma \vdash t : A'$. Luego, se concluye por (S-TRANS).

□

Este resultado permite obtener funciones de chequeo de tipos simples tanto para patrones ($\mathsf{tcp}(\theta, p)$) como para términos ($\mathsf{tc}(\Gamma, t)$), presentadas en la Fig. 3.11, de modo que $\mathsf{tc}(\Gamma, t) = A$ si y solo si existe $A' \simeq_\mu A$ tal que $\rhd_{\mathcal{C}} \Gamma \vdash t : A$. Es interesante destacar el caso de la aplicación, donde el sistema no es estrictamente dirigido por sintaxis si se consideran unicamente los sujetos de las reglas: hay solapamiento entre (T-COMP-AL) y (T-APP-AL). Sin embargo, el resultado del llamado recursivo sobre r en el término $r\,u$ permite eliminar esa ambigüedad al analizar sintácticamente el tipo resultante. Si el tipo de r es un datatype, se está necesariamente en el caso (T-COMP-AL), mientras que si éste puede ser reescrito (mediante equivalencia) como una unión de tipos funcionales, se construye el tipo de $r\,u$ como la unión de los co-dominios, tal como lo indica (T-APP-AL). La expresión $\mathsf{unfold}(A)$ permite reescribir el tipo de r apelando a las reglas (E-REC-L-AL) y (E-REC-R-AL) hasta obtener un tipo unión-maximal $\oplus_{i<n} A_i$ tal que $(A_i \neq \mu, \oplus)_{i<n}$:

$$
\begin{array}{rcl}
\mathsf{unfold}(A \supset B) & \triangleq & A \supset B \\
\mathsf{unfold}(\oplus_{i<n} A_i) & \triangleq & \texttt{let } \oplus_{j<m_i}(A_{ij} \supset B_{ij}) = \mathsf{unfold}(A_i) \texttt{ foreach } i < n \texttt{ in} \\
 & & \quad \oplus_{\substack{i<n \\ j<m_i}} (A_{ij} \supset B_{ij}) \qquad \text{si } n > 1 \text{ y } (A_i \neq \oplus)_{i<n} \\
\mathsf{unfold}(\mu V.A) & \triangleq & \mathsf{unfold}(\{V \setminus \mu V.A\} A) \\
\mathsf{unfold}(\_) & \triangleq & \texttt{fail}
\end{array}
$$

La terminación de la función unfold está garantizada por contractividad. Notar que el uso de esta función, cuya corrección se justifica mediante reglas de equivalencia, resulta en la no necesidad de utilizar el algoritmo de chequeo de equivalencia en sí mismo.

Por su parte, los predicados de compatibilidad y solapamiento (*cf.* Sec. 3.3.4) admiten implemen-

taciones directas dado el algoritmo para chequeo de sub-tipado presentado anteriormente:

$$\begin{array}{rcl}
\mathsf{compat}(p : A, q : B) & \triangleq & (\texttt{not}\ \mathsf{overlap}(p : A, q : B))\ \texttt{or}\ \mathsf{subtype}(\emptyset, B, A) \\
\\
\mathsf{overlap}(p_0\, p_1 : A, q_0\, q_1 : B) & \triangleq & \texttt{let}\ A = A_0 \mathbin{@} A_1, B = B_0 \mathbin{@} B_1\ \texttt{in} \\
& & \quad \mathsf{overlap}(p_0 : A_0, q_0 : B_0)\ \texttt{and}\ \mathsf{overlap}(p_1 : A_1, q_1 : B_1) \\
\mathsf{overlap}(p : A, q : B) & \triangleq & (p = x)\ \texttt{or}\ (p = q = \mathtt{c})\ \texttt{or}\ (A{\downarrow_\circ}_\epsilon \cap B{\downarrow_\circ}_\epsilon \neq \emptyset)
\end{array}$$

En particular, overlap asume que las asignaciones $p : A$ y $q : B$ ya fueron chequeadas, *i.e.* existen θ, θ' tales que $\rhd_{\mathcal{C}} \theta \vdash p : A$ y $\rhd_{\mathcal{C}} \theta' \vdash q : B$. Luego, si ambos patrones son compounds, sus tipos asociados son necesariamente aplicativos, mientras que los tipos unión solo pueden ser asignados a variables (*i.e.* en posición de hoja en el árbol sintáctico de los patrones). Esta correspondencia es aprovechada para recorrer simultáneamente patrones y tipos de manera lineal, lo que implica que el orden de complejidad para chequear compatibilidad queda gobernado por el algoritmo de chequeo de sub-tipado.

Análisis de complejidad

Durante la ejecución de tc se realiza un recorrido sobre la estructura completa del término, resultando de particular interés los casos de la aplicación y la abstracción, pues incurren en verificaciones adicionales que rompen el orden lineal de este recorrido.

Por un lado, al analizar una aplicación $r\, u$ se realizan los correspondientes llamados recursivos y, de no ser el tipo asociado a r un datatype, se utiliza la función unfold para verificar que se trate de una unión de tipos funcionales y obtener así sus componentes. Luego, se realiza un llamado a subtype por cada tipo de la unión maximal resultante. Por otra parte, para términos de la forma $(p_i \to_{\theta_i} s_i)_{i<n}$ es necesario verificar la compatibilidad de la lista de asignaciones de tipo a los patrones $[p_i : A_i]_{i<n}$ (siempre que $n > 1$), lo cual resulta en $O(n^2)$ llamados a compat. Esta función auxiliar realiza un recorrido lineal sobre los patrones examinados al evaluar overlap y eventualmente efectúa un llamado a subtype.

Observar, en primer lugar, que al utilizar la función unfold puede ser necesario desdoblar expresiones de la forma $\mu V.A$ para evaluar si se trata de un tipo funcional. Dependiendo de su estructura, el tamaño del resultado de dicha sustitución podría crecer exponencialmente con respecto al de la expresión original. Considerar, por ejemplo, la secuencia de desdoblamientos del tipo $\mu X_1 \ldots \mu X_n . \oplus_{i<n} (X_i \supset \mathtt{c})$:

$$\begin{array}{rcl}
A_0 & = & \mu X_0 \ldots \mu X_n . \oplus_{i<n} (X_i \supset \mathtt{c}) \\
A_1 & = & \mu X_1 \ldots \mu X_n . \oplus_{i\in[1,n)} (X_i \supset \mathtt{c}) \oplus (A_0 \supset \mathtt{c}) \\
A_2 & = & \mu X_2 \ldots \mu X_n . \oplus_{i\in[2,n)} (X_i \supset \mathtt{c}) \oplus (A_0 \supset \mathtt{c}) \oplus (A_1 \supset \mathtt{c}) \\
 & \vdots & \\
A_n & = & \oplus_{i<n} (A_i \supset \mathtt{c})
\end{array}$$

Aquí, en cada reescritura se remplaza una variable por la expresión de tipo completa, duplicando el tamaño de la misma en cada paso, obteniendo un resultado final con tamaño de orden 2^n.

Un problema similar surge en la función subtype, donde se deben realizar las sustituciones adecuadas para desdoblar construcciones recursivas. Adicionalmente, en [Pie02] se argumenta que el esquema gfp (*cf.* Sec. 1.2.4), en el cual se basa el algoritmo de sub-tipado desarrollado, produce una cantidad cuadrática de llamados recursivos, producto de explorar las combinaciones de los sub-términos del par a verificar. Esta justificación está basada fuertemente en el hecho de que nunca se descarta el conjunto de hipótesis confirmadas hasta el momento. Sin embargo, este no es el caso de la adaptación aquí propuesta. En efecto, al verificar tipos de la forma $A \preceq_\mu \oplus_{i<m} B_j$ (con $m > 1$, $A \neq \mu, \oplus$ y $B_j \neq \oplus$) pueden efectuarse eventualmente m llamados recursivos realizando un *backtracking* en cada

paso, producto de descartar el conjunto de hipótesis tras cada prueba del comando `seq`. Más aún, el hecho de que la cantidad de sub-problemas dependa de la estructura de los tipos involucrados dificulta un análisis detallado sobre la cantidad total de operaciones en un algoritmo de naturaleza recursiva como el presentado.

Si bien los algoritmos presentados anteriormente son claros e intuitivos respecto a la semántica de las relaciones que verifican, el análisis expuesto sugiere la realización de un estudio más acabado respecto a dos aspectos puntuales: (I) la adopción de una representación adecuada para las expresiones de tipo que evite el crecimiento exponencial de la representación algebraica actual al desdoblar los tipos recursivos; (II) el desarrollo de un esquema algorítmico donde sea posible realizar un análisis de complejidad correcto en presencia de operadores asociativos, conmutativos e idempotentes (ACI).

3.6.3. Hacia un chequeo de tipos eficiente

El algoritmo presentado anteriormente es claro en cuanto a su correspondencia con el sistema $\mathcal{C}$ y las relaciones de equivalencia de tipos y sub-tipado asociadas, pero resulta ineficiente en términos de complejidad computacional. En particular, el número de llamados recursivos en eqtype y subtype depende del tamaño de los tipos en cuestión, y desdoblar estos tipos puede incrementar exponencialmente su tamaño. En la presente sección se desarrolla un nuevo algoritmo de chequeo de tipos que busca atender estos puntos débiles, tomando ideas de los siguientes trabajos:

[JP97] donde se aborda, entre otros temas, el problema de chequeo de sub-tipado en presencia de tipos recursivos.

[PZ01] donde se propone una representación para tipos recursivos sobre *term automata*, la cual resulta más conveniente que la algebraica adoptada en la sección anterior.

[DPR05] donde se aborda el problema de chequeo de sub-tipado para tipos recursivos en presencia de productos asociativos y conmutativos (AC).

Estas ideas son combinadas y adaptadas al caso de operadores idempotentes (los tipos unión son ACI) con el fin de mejorar el orden de complejidad de dos pasos clave en el algoritmo tc, a saber: subtype y unfold; este último particularmente sensible a la representación adoptada. El resultado final es un algoritmo de complejidad $\mathcal{O}(n^7 d)$ donde n es el tamaño de la entrada (*i.e.* Γ más t) y d es una cota a la aridad de las uniones (maximales) que ocurren en Γ y t. Notar que toda la información necesaria para tipar t se encuentra en el contexto o anotada en el mismo término. Por lo tanto, puede establecerse una relación lineal entre el tamaño de la entrada y el del tipo resultante, lo que permite considerar a n como el tamaño de este último.

Term automata

Los μ-tipos contractivos pueden ser interpretados como grafos dirigidos dado que su desdoblamiento infinito produce árboles (infinitos) regulares. Como una simplificación adicional para una adecuada representación, se reemplazan las uniones binarias utilizadas hasta el momento por constructores n-arios, colapsando múltiples uniones consecutivas en un único nodo de la aridad adecuada. Sea $\mathfrak{L}_{\mathfrak{n}} \triangleq \{a^0 \mid a \in \mathcal{V} \cup \mathcal{C}\} \cup \{@^2, \supset^2\} \cup \{\oplus^n \mid n > 1\}$, se define $\mathfrak{T}_{\mathfrak{n}}$ como el conjunto de *árboles n-arios (infinitos)* cuyos nodos son símbolos de $\mathfrak{L}_{\mathfrak{n}}$. La construcción es análoga a la presentada en la Sec. 3.3.2 para árboles binarios y, al igual que entonces, se distinguen los sub-conjuntos propios de *árboles finitos* $\mathfrak{T}_{\mathfrak{n}}^{fin}$ y *árboles regulares* $\mathfrak{T}_{\mathfrak{n}}^{reg}$ [Cou83]. En este caso, lo μ-tipos contractivos son interpretados en $\mathfrak{T}_{\mathfrak{n}}^{reg}$ colapsando los tipos unión-maximal en un único nodo unión n-aria. Para eso se apela a la Def. 3.3.22 y a una nueva función de traducción de tipos infinitos (árboles binarios) en árboles n-arios.

Definición 3.6.15. *La* traducción *de tipos infinitos en árboles n-arios infinitos regulares* $[\![\_]\!]^{\mathfrak{n}} : \mathfrak{T} \to \mathfrak{T}_{\mathfrak{n}}$, *se define inductivamente de la siguiente manera:*

$$\begin{array}{rcll} [\![a]\!]^{\mathfrak{n}}(\epsilon) & \triangleq & a & \textit{para } a \in \mathcal{V} \cup \mathcal{C} \\ [\![\mathcal{A}_0 \star \mathcal{A}_1]\!]^{\mathfrak{n}}(\epsilon) & \triangleq & \star & \textit{para } \star \in \{@, \supset\} \\ [\![\mathcal{A}_0 \star \mathcal{A}_1]\!]^{\mathfrak{n}}(i\mathsf{p}) & \triangleq & [\![A_i]\!]^{\mathfrak{n}}(\mathsf{p}) & \textit{para } \star \in \{@, \supset\} \\ [\![\oplus_{i<n}\mathcal{A}_i]\!]^{\mathfrak{n}}(\epsilon) & \triangleq & \oplus^n & \\ [\![\oplus_{i<n}\mathcal{A}_i]\!]^{\mathfrak{n}}(i\mathsf{p}) & \triangleq & [\![\mathcal{A}_i]\!]^{\mathfrak{n}}(\mathsf{p}) & \end{array}$$

A su vez, dado un μ-tipo contractivo A, se define $[\![A]\!]^{\mathfrak{n}} \triangleq [\![[\![A]\!]^{\mathfrak{T}}]\!]^{\mathfrak{n}}$.

Estos árboles n-arios serán entonces representados como *term automata* [AC93].

Definición 3.6.16. *Un* term automaton *es una tupla* $\mathcal{M} = \langle \mathcal{Q}, \Sigma, q_0, \delta, \ell \rangle$ *donde:*

1. $\mathcal{Q}$ *es un conjunto finito de estados.*
2. Σ *es un alfabeto donde cada símbolo tiene un aridad asociada.*
3. q_0 *es el estado inicial.*
4. $\delta : \mathcal{Q} \times \mathbb{N} \to \mathcal{Q}$ *es una función parcial de transiciones entre estados, definida sobre la aridad del símbolo asociado a por* ℓ *a cada estado.*
5. $\ell : \mathcal{Q} \to \Sigma$ *es una función total de etiquetado sobre los estados.*

Se denota con $\mathcal{M}_A$ el autómata asociado al tipo A, que permite recorrer todos los caminos desde la raíz de A a cualquiera de sus sub-expresiones. La Fig. 3.12 ilustra el árbol n-ario infinito y el autómata asociados al tipo $\mathsf{List}_A \triangleq \mu\alpha.\mathbb{nil} \oplus (\mathbb{cons} \mathbin{@} A \mathbin{@} \alpha)$, *i.e.* $[\![\mathsf{List}_A]\!]^{\mathfrak{T}}$ y $\mathcal{M}_{\mathsf{List}_A}$ respectivamente. Si q_0 es el estado inicial de $\mathcal{M}_{\mathsf{List}_A}$ y $\widehat{\delta}$ denota la extensión natural de δ a secuencias de símbolos, luego $\ell(\widehat{\delta}(q_0, 100)) = \mathbb{cons}$. Como se mencionó anteriormente, la estructura regular de los árboles resultantes de traducir μ-tipos contractivos lleva a que le autómata tenga una cantidad finita de estados y, por lo tanto, quede bien definido.

Chequeo de sub-tipado

Se presenta en primer lugar un algoritmo eficiente para chequeo de sub-tipado, postergando el caso de la equivalencia, dado que esta última no es invocada finalmente por el algoritmo de chequeo de tipos tc. Sin embargo, el desarrollo que sigue a continuación es general y puede adaptarse a distintas relaciones sobre árboles n-arios. Se utilizará el caso de la equivalencia para ilustrar esta situación.

Se introduce a continuación una noción co-inductiva de sub-tipado sobre $\mathfrak{T}_{\mathfrak{n}}$ *up-to* un conjunto de hipótesis.

Definición 3.6.17. *La relación de* sub-tipado $\preceq^{\mathcal{R}}_{\mathfrak{n}}$ *de árboles n-arios infinitos* up-to *un conjunto de hipótesis* $\mathcal{R}$ *se define como la interpretación co-inductiva de los esquemas de regla de la Fig. 3.13.*

Formalmente, sea $\Phi_{\preceq^{\mathcal{R}}_{\mathfrak{n}}} : \wp(\mathfrak{T} \times \mathfrak{T}) \to \wp(\mathfrak{T} \times \mathfrak{T})$ la función generadora asociada a las reglas de la

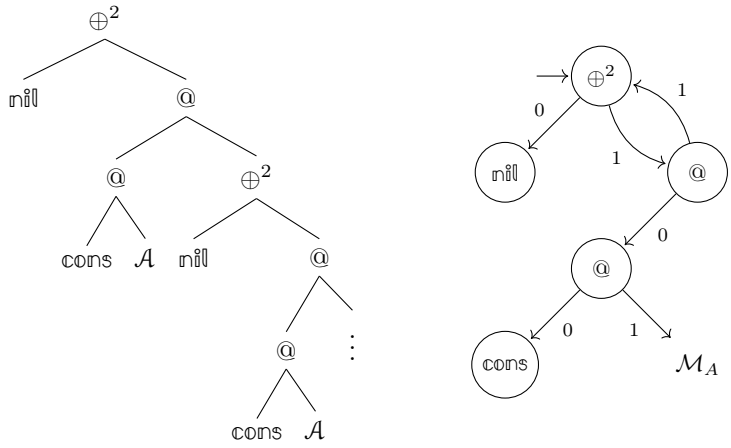


Figura 3.12: Representación como árbol (infinito) n-ario y term automaton del tipo List_A.

$$\frac{}{a \preceq_{\mathtt{n}}^{\mathcal{R}} a}\ (\textsc{s-refl-up})$$

$$\frac{\mathcal{D}\ (\preceq_{\mathtt{n}}^{\mathcal{R}} \cup \mathcal{R})\ \mathcal{D}' \quad \mathcal{A}\ (\preceq_{\mathtt{n}}^{\mathcal{R}} \cup \mathcal{R})\ \mathcal{A}'}{\mathcal{D} \mathbin{@} \mathcal{A} \preceq_{\mathtt{n}}^{\mathcal{R}} \mathcal{D}' \mathbin{@} \mathcal{A}'}\ (\textsc{s-comp-up}) \qquad \frac{\mathcal{A}'\ (\preceq_{\mathtt{n}}^{\mathcal{R}} \cup \mathcal{R})\ \mathcal{A} \quad \mathcal{B}\ (\preceq_{\mathtt{n}}^{\mathcal{R}} \cup \mathcal{R})\ \mathcal{B}'}{\mathcal{A} \supset \mathcal{B} \preceq_{\mathtt{n}}^{\mathcal{R}} \mathcal{A}' \supset \mathcal{B}'}\ (\textsc{s-func-up})$$

$$\frac{(\mathcal{A}_i\ (\preceq_{\mathtt{n}}^{\mathcal{R}} \cup \mathcal{R})\ \mathcal{B})_{i<n} \quad (\mathcal{A}_i \neq \oplus)_{i<n} \quad \mathcal{B} \neq \oplus}{\oplus_i^n \mathcal{A}_i \preceq_{\mathtt{n}}^{\mathcal{R}} \mathcal{B}}\ (\textsc{s-union-l-up})$$

$$\frac{\mathcal{A}\ (\preceq_{\mathtt{n}}^{\mathcal{R}} \cup \mathcal{R})\ \mathcal{B}_k \quad k < m \quad \mathcal{A} \neq \oplus \quad (\mathcal{B}_j \neq \oplus)_{j<m}}{\mathcal{A} \preceq_{\mathtt{n}}^{\mathcal{R}} \oplus_j^m \mathcal{B}_j}\ (\textsc{s-union-r-up})$$

$$\frac{(\mathcal{A}_i\ (\preceq_{\mathtt{n}}^{\mathcal{R}} \cup \mathcal{R})\ \mathcal{B}_{f(i)})_{i<n} \quad f : [0,n) \to [0,m) \quad (\mathcal{A}_i \neq \oplus)_{i<n} \quad (\mathcal{B}_j \neq \oplus)_{j<m}}{\oplus_i^n \mathcal{A}_i \preceq_{\mathtt{n}}^{\mathcal{R}} \oplus_j^m \mathcal{B}_j}\ (\textsc{s-union-up})$$

Figura 3.13: Sub-tipado *up-to* $\mathcal{R}$ para árboles n-arios.

Fig. 3.13, definido como

$$\begin{array}{rcl}
\Phi_{\preceq^{\mathcal{R}}_{\mathfrak{n}}}(\mathcal{X}) & \triangleq & \{\langle a, a\rangle \mid a \in \mathcal{V} \cup \mathcal{C}\} \\
& \cup & \{\langle \mathcal{D} \mathbin{@} \mathcal{A}, \mathcal{D}' \mathbin{@} \mathcal{A}'\rangle \mid \langle \mathcal{D}, \mathcal{D}'\rangle, \langle \mathcal{A}, \mathcal{A}'\rangle \in \mathcal{X} \cup \mathcal{R}\} \\
& \cup & \{\langle \mathcal{A} \supset \mathcal{B}, \mathcal{A}' \supset \mathcal{B}'\rangle \mid \langle \mathcal{A}', \mathcal{A}\rangle, \langle \mathcal{B}, \mathcal{B}'\rangle \in \mathcal{X} \cup \mathcal{R}\} \\
& \cup & \{\langle \oplus_i^n \mathcal{A}_i, \mathcal{B}\rangle \mid \mathcal{A}_i, \mathcal{B} \neq \oplus, \langle \mathcal{A}_i, \mathcal{B}\rangle \in \mathcal{X} \cup \mathcal{R}\} \\
& \cup & \{\langle \mathcal{A}, \oplus_j^m \mathcal{B}_j\rangle \mid \mathcal{A}, \mathcal{B}_j \neq \oplus, \\
& & \quad \exists k < m. \langle \mathcal{A}, \mathcal{B}_k\rangle \in \mathcal{X} \cup \mathcal{R}\} \\
& \cup & \{\langle \oplus_i^n \mathcal{A}_i, \oplus_j^m \mathcal{B}_j\rangle \mid \mathcal{A}_i, \mathcal{B}_j \neq \oplus, \\
& & \quad \exists f : [0, n) \to [0, m). \langle \mathcal{A}_i, \mathcal{B}_{f(i)}\rangle \in \mathcal{X} \cup \mathcal{R}\}
\end{array}$$

Luego, $\preceq^{\mathcal{R}}_{\mathfrak{n}} \triangleq \nu\Phi_{\preceq^{\mathcal{R}}_{\mathfrak{n}}}$. Esta noción resulta conveniente para el posterior desarrollo y verificación de un algoritmo eficiente para chequeo de sub-tipado. Intuitivamente, $\mathcal{A} \preceq^{\mathcal{R}}_{\mathfrak{n}} \mathcal{B}$ expresa que $\mathcal{A}$ puede demostrarse sub-tipo de $\mathcal{B}$ siempre y cuando se asuman como válidas las hipótesis en el conjunto de pares $\mathcal{R}$. Para $\mathcal{R} = \emptyset$, $\preceq^{\mathcal{R}}_{\mathfrak{n}}$ coincide con $\preceq_{\mathfrak{T}}$ vía la traducción de tipos infinitos a árboles n-arios.

Lema 3.6.18. *$\mathcal{A} \preceq_{\mathfrak{T}} \mathcal{B}$ sii $\mathcal{A} \preceq^{\emptyset}_{\mathfrak{n}} \mathcal{B}$.*

Por lo tanto, $\preceq^{\emptyset}_{\mathfrak{n}}$ también se corresponde con la noción de sub-tipado principal del sistema $\preceq_{\mu}$ a través de la traducción de μ-contractivos a tipos infinitos.

Teorema 3.6.19. *$A \preceq_{\mu} B$ sii $[\![A]\!]^{\mathfrak{n}} \preceq^{\emptyset}_{\mathfrak{n}} [\![B]\!]^{\mathfrak{n}}$.*

Demostración. Por Teo. 3.3.30 y Lem. 3.6.18: $A \preceq_{\mu} B$ sii $[\![A]\!]^{\mathfrak{T}} \preceq_{\mathfrak{T}} [\![B]\!]^{\mathfrak{T}}$ sii $[\![A]\!]^{\mathfrak{n}} \preceq^{\emptyset}_{\mathfrak{n}} [\![B]\!]^{\mathfrak{n}}$. □

Descripción del algoritmo

El algoritmo de chequeo para esta nueva noción de sub-tipado se desarrolla a partir de ideas en [DPR05]. La presentación dada a continuación es general, pudiendo ser no solo aplicada a sub-tipado sino también a otras relaciones, como se hará con la equivalencia más adelante. Un par $p \in \mathfrak{T}_{\mathfrak{n}} \times \mathfrak{T}_{\mathfrak{n}}$ se dice *válido* si $p \in \preceq^{\emptyset}_{\mathfrak{n}}$. El algoritmo consiste en dos fases. El objetivo de la primera es construir un universo $\mathcal{U} \subseteq \mathfrak{T}_{\mathfrak{n}} \times \mathfrak{T}_{\mathfrak{n}}$ que será luego refinado para obtener un conjunto comprendido únicamente por pares válidos. Se comienza con un par inicial (*cf.* Fig. 3.14, buildUniverse) y luego se exploran los pares de sub-términos de ambos tipos en el par descomponiendo el constructor en cuestión (*cf.* Fig. 3.14, children). Notar que, dado p, el algoritmo puede en principio agregar pares inválidos al universo. La segunda fase es la encargada de eliminar estos pares no deseados. Notar también que la primera fase puede adaptarse fácilmente a otras relaciones simplemente re-definiendo la función children. $\mathcal{U}$ puede ser interpretado como un grafo dirigido donde un eje p a $q \in \mathsf{children}(p)$ es agregado para denotar que q puede pertenecer al conjunto soporte de p (*cf.* Def. 1.2.7) en la relación final $\preceq^{\emptyset}_{\mathfrak{n}}$. En este caso se dice que p es un *padre* de q. Dado que los tipos representados como autómatas pueden contener ciclos, un par puede ser agregado a $\mathcal{U}$ más de una vez y, por lo tanto, tener más de un padre. Sea $\mathsf{deg}(p)$ el *grado de entrada* de p, *i.e.* su número de padres.

Durante la segunda fase del algoritmo (Fig. 3.15, $\mathsf{gfp}^{\preceq}$) se mantienen los siguiente conjuntos, los cuales conforman una partición de $\mathcal{U}$:

- $\mathcal{W}$: pares cuya validez debe ser verificada.
- $\mathcal{S}$: pares considerados válidos condicionalmente.
- $\mathcal{F}$: pares inválidos.

El algoritmo consiste en iterar sobre el conjunto de trabajo $\mathcal{W}$ y transferir el par seleccionado en cada iteración a $\mathcal{S}$ si su validez puede ser probada asumiendo válidos todos los pares que no fueron

```
buildUniverse(p_0) :
    U = ∅
    W = {p_0}
    while W ≠ ∅ :
      p := takeOne(W)
      if p ∉ U
        insert(p, U)
        foreach q ∈ children(p)
          insert(q, W)
    return U
```

```
children(p) :
    case p of
      ⟨D @ A, D' @ B⟩ →
          {⟨D, D'⟩, ⟨A, B⟩}
      ⟨A' ⊃ A'', B' ⊃ B''⟩ →
          {⟨B', A'⟩, ⟨A'', B''⟩}
      ⟨⊕_i^n A_i, ⊕_j^m B_j⟩ →
          {⟨A_i, B_j⟩ | i < n, j < m}
      ⟨⊕_i^n A_i, B⟩, B ≠ ⊕ →
          {⟨A_i, B⟩ | i < n}
      ⟨A, ⊕_j^m B_j⟩, A ≠ ⊕ →
          {⟨A, B_j⟩ | j < m}
      otherwise →
          ∅
```

Figura 3.14: Pseudo-código del la primera fase: construcción del universo.

descartados hasta el momento (*i.e.* los pares en $\mathcal{S} \cup \mathcal{W}$). De no ser posible, p es transferido a $\mathcal{F}$ y todos sus padres en $\mathcal{S}$ deben ser re-evaluados pues su validez *up-to* $\mathcal{W}$ pudo verse modificada. Por lo tanto, estos padres son insertados nuevamente en $\mathcal{W}$ (Fig. 3.15, invalidate). Intuitivamente, $\mathcal{S}$ contiene elementos en $\preceq_{\mathrm{n}}^{\mathcal{W}}$. El proceso termina cuando $\mathcal{W}$ se vacía. El único aspecto de esta segunda fase específico de la relación $\preceq_{\mathrm{n}}^{\mathcal{W}}$ es la función check, la cual puede ser re-definida para contemplar otras posibles relaciones *up-to*.

Corrección del algoritmo

La corrección del algoritmo se basa en el hecho de que $\mathcal{S}$ puede ser considerado un conjunto de pares válidos *asumiendo la validez de los elementos de* $\mathcal{W}$. Más precisamente:

Teorema 3.6.20. *El algoritmo* $\mathsf{gfp}^{\preceq}$ *preserva la siguiente invariante:*

- $\langle \mathcal{W}, \mathcal{S}, \mathcal{F} \rangle$ *es una partición de* $\mathcal{U}$*.*
- $\mathcal{F}$ *consiste únicamente de pares inválidos.*
- $\mathcal{S} \subseteq \Phi_{\preceq_{\mathrm{n}}^{\mathcal{W}}}(\mathcal{S})$

Demostración. Es inmediato ver que las tres condiciones se satisfacen al inicio de la segunda fase del algoritmo, donde $\mathcal{W} = \mathcal{U}$ y $\mathcal{S} = \mathcal{F} = \emptyset$. Más aún, a lo largo del proceso los elementos son transferidos entre estos conjuntos preservando la primera condición.

Notar que en cada iteración, la decisión de invalidar un par p o moverlo a $\mathcal{S}$ se toma analizando si hay suficientes elementos en $\mathcal{S} \cup \mathcal{W}$ para probar su validez (*cf.* Fig. 3.15, check). Esto se muestra analizando la estructura de p:

- $p = \langle a, a \rangle$. Como se ve en (S-REFL-UP), la validez de p no depende de ningún otro par (su soporte es el conjunto vacío). Como es esperado, el algoritmo transfiere directamente p a $\mathcal{S}$, sin chequeo adicional alguno.
- $p = \langle \mathcal{D} \mathbin{@} \mathcal{A}, \mathcal{D}' \mathbin{@} \mathcal{B} \rangle$. Por (S-COMP-UP), p es válido si y solo si los pares $\langle \mathcal{D}, \mathcal{D}' \rangle$ y $\langle \mathcal{A}, \mathcal{B} \rangle$ también lo son. Esos son exactamente los pares evaluados para decidir si invalidar p o no.

```
gfp^≺(p) :
    W = buildUniverse(p)
    S = ∅
    F = ∅
    while W ≠ ∅ :
      q := takeOne(W)
      if check(q, F)
        then insert(q, S)
        else invalidate(q, S, F, W)
    return p ∈ S

invalidate(p, S, F, W) :
    insert(p, F)
    foreach q ∈ parents(p) ∩ S
      move(q, S, W)
```

```
check(p, F) :
    case p of
      ⟨a, a⟩ →
          true
      ⟨D @ A, D' @ B⟩ →
          ⟨D, D'⟩ ∉ F and ⟨A, B⟩ ∉ F
      ⟨A' ⊃ A'', B' ⊃ B''⟩ →
          ⟨B', A'⟩ ∉ F and ⟨A'', B''⟩ ∉ F
      ⟨⊕_i^n A_i, B⟩, B ≠ ⊕ →
          ∀i.⟨A_i, B⟩ ∉ F
      ⟨A, ⊕_j^m B_j⟩, A ≠ ⊕ →
          ∃m.⟨A, B_m⟩ ∉ F
      ⟨⊕_i^n A_i, ⊕_j^m B_j⟩ →
          ∀i.∃j.⟨A_i, B_j⟩ ∉ F
      otherwise →
          false
```

Figura 3.15: Pseudo-código del la segunda fase: refinamiento.

- $p = \langle \mathcal{A}' \supset \mathcal{A}'', \mathcal{B}' \supset \mathcal{B}'' \rangle$. Este caso corresponde a la regla (S-FUNC-UP): p será trasferido a $\mathcal{S}$ si y solo si los pares $\langle \mathcal{B}', \mathcal{A}' \rangle$ y $\langle \mathcal{A}'', \mathcal{B}'' \rangle$ no fueron descartados aún.
- $p = \langle \oplus_i^n \mathcal{A}_i, \mathcal{B} \rangle$ con $\mathcal{B} \neq \oplus$. Como indica (S-UNION-L-UP), el proceso verifica que ningún par $(\langle \mathcal{A}_i, \mathcal{B} \rangle)_{i<n}$ haya sido invalidado aún.
- $p = \langle \mathcal{A}, \oplus_j^m \mathcal{B}_j \rangle$ con $\mathcal{A} \neq \oplus$. Por (S-UNION-R-UP), es suficiente con encontrar un par $\langle \mathcal{A}, \mathcal{B}_k \rangle$ para algún $k \in [0, m)$ no invalidado para mover p al conjunto de pares condicionalmente válidos.
- $p = \langle \oplus_i^n \mathcal{A}_i, \oplus_j^m \mathcal{B}_j \rangle$. El algoritmo verifica que para cada $\mathcal{A}_i$ exista un $\mathcal{B}_j$ aún no invalidado. Estás son exactamente las hipótesis de (S-UNION-UP).

Si no hay suficientes elementos en $\mathcal{S} \cup \mathcal{W}$ para considerar al par p válido, entonces éste es efectivamente inválido para la relación $\preceq_{\mathrm{n}}^{\mathcal{W}}$ y por lo tanto movido a $\mathcal{F}$, preservando la segunda condición invariante.

Respecto a la tercera y última condición, considerar el siguiente análisis por inducción en el número de iteraciones del ciclo principal del algoritmo. Como se comentó anteriormente, el caso base es inmediato con $\mathcal{S}_0 = \emptyset$. Se usan indices i sobre los conjuntos para identificar su estado al inicio de la i-ésima iteración. Hay dos casos a considerar según la decisión tomada sobre p:

- Si p es transferido a $\mathcal{S}$ (*i.e.* todos los elementos necesarios para probar su validez están en $\mathcal{S}_i \cup \mathcal{W}_i$) se tienen $\mathcal{W}_{i+1} = \mathcal{W}_i \setminus \{p\}$, $\mathcal{S}_{i+1} = \mathcal{S}_i \cup \{p\}$ y $\mathcal{F}_{i+1} = \mathcal{F}_i$. Por *h.i.* , $\mathcal{S}_i \subseteq \Phi_{\preceq_{\mathrm{n}}^{\mathcal{W}_i}}(\mathcal{S}_i)$, por lo tanto $\mathcal{S}_i \cup \{p\} \subseteq \Phi_{\preceq_{\mathrm{n}}^{\mathcal{W}_i}}(\mathcal{S}_i) \cup \{p\}$. Notar que $\mathcal{S}_{i+1} \cup \mathcal{W}_{i+1} = \mathcal{S}_i \cup \mathcal{W}_i$. Luego, por definición de $\Phi_{\preceq_{\mathrm{n}}^{\mathcal{R}}}$ se tiene $\Phi_{\preceq_{\mathrm{n}}^{\mathcal{W}_i}}(\mathcal{S}_i) = \Phi_{\preceq_{\mathrm{n}}^{\mathcal{W}_{i+1}}}(\mathcal{S}_{i+1})$. Por lo tanto, $\mathcal{S}_i \cup \{p\} \subseteq \Phi_{\preceq_{\mathrm{n}}^{\mathcal{W}_{i+1}}}(\mathcal{S}_{i+1}) \cup \{p\}$. El conjunto a la izquierda de la desigualdad es exactamente $\mathcal{S}_{i+1}$, Más aún, dado que los elementos necesarios para considerar válido a p se encuentran en $\mathcal{S}_{i+1} \cup \mathcal{W}_{i+1}$, necesariamente se tiene $p \in \Phi_{\preceq_{\mathrm{n}}^{\mathcal{W}_{i+1}}}(\mathcal{S}_{i+1})$. Finalmente, se concluye $\mathcal{S}_{i+1} \subseteq \Phi_{\preceq_{\mathrm{n}}^{\mathcal{W}_{i+1}}}(\mathcal{S}_{i+1})$.
- Si p es transferido a $\mathcal{F}$ se tienen $\mathcal{W}_{i+1} = (\mathcal{W}_i \setminus \{p\}) \cup \mathcal{Q}$, $\mathcal{S}_{i+1} = \mathcal{S}_i \setminus \{p\}$ y $\mathcal{F}_{i+1} = \mathcal{F}_i \cup \{p\}$, donde $\mathcal{Q}$ contiene los padres de p que pertenecen a $\mathcal{S}_i$. Asumir en pos de una contradicción que existe $s \in \mathcal{S}_{i+1}$ tal que $s \notin \Phi_{\preceq_{\mathrm{n}}^{\mathcal{W}_{i+1}}}(\mathcal{S}_{i+1})$. Luego, s no puede tener un conjunto soporte vacío. Por otro lado, la *h.i.* establece que $\mathcal{S}_i$ es $\Phi_{\preceq_{\mathrm{n}}^{\mathcal{W}_i}}$-denso, por lo que existe al menos un elemento

q en el soporte de s tal que $q \in \mathcal{S}_i \cup \mathcal{W}_i$ pero $q \notin \mathcal{S}_{i+1} \cup \mathcal{W}_{i+1}$. Luego, por definición de $\mathcal{W}_{i+1}$ y $\mathcal{S}_{i+1}$, necesariamente es el caso $q = p$, lo que implica que s es padre de p. Dado que el algoritmo transfiere a $\mathcal{W}_{i+1}$ todos los padres de p que pertenecen a $\mathcal{S}_i$, y $\mathcal{S}_{i+1} \subseteq \mathcal{S}_i$ por definición, necesariamente se tiene $s \in \mathcal{Q}$, lo que contradice $s \in \mathcal{S}_{i+1}$. La contradicción proviene de asumir la existencia de tal elemento s, por lo que se concluye $\mathcal{S}_{i+1} \subseteq \Phi_{\preceq_{\mathtt{n}}^{\mathcal{W}_{i+1}}}(\mathcal{S}_{i+1})$.

□

Cuando el ciclo principal de la función $\mathsf{gfp}^{\preceq}$ termina, el conjunto $\mathcal{W}$ se encuentra vacío, por lo que se tiene $\mathcal{S} \subseteq \Phi_{\preceq_{\mathtt{n}}^{\emptyset}}(\mathcal{S})$. Luego, por principio de co-inducción (Cor. 1.2.5 (2)) $\mathcal{S} \subseteq \preceq_{\mathtt{n}}^{\emptyset}$ (*i.e.* todo par en $\mathcal{S}$ es válido) por lo que basta verificar si el par original pertenece a $\mathcal{S}$ (caso contrario, se encuentra en $\mathcal{F}$ por invariante).

Corolario 3.6.21. $\mathcal{A} \preceq_{\mathtt{n}}^{\emptyset} \mathcal{B}$ *sii* $\mathsf{gfp}^{\preceq}(\langle \mathcal{A}, \mathcal{B} \rangle) = \mathtt{true}$.

Basta entonces reemplazar los llamados a $\mathsf{subtype}$ por $\mathsf{gfp}^{\preceq}$ en el algoritmo de chequeo de tipos tc.

Complejidad del algoritmo

La primera fase del algoritmo consiste en identificar pares de sub-términos relevantes en ambos tipos a analizar. Sean N y M los tamaños de tales tipos (considerando nodos y ejes en sus representaciones como *term automata*). Luego, el tamaño y el costo de construir el universo $\mathcal{U}$ puede ser acotado por $\mathcal{O}(NM)$. Como se verá a continuación, el costo total del algoritmo está gobernado por las operaciones de la segunda fase.

Como se indica en [DPR05], dado que un par p solo puede ser invalidado una vez (en cuyo caso se transfieren $\mathsf{deg}(p)$ nodos a $\mathcal{W}$ para su re-consideración) el número de iteraciones en la etapa de refinamiento se encuentra acotado por

$$\sum_{p \in \mathcal{U}} 1 + \sum_{p \in \mathcal{U}} \mathsf{deg}(p) = \sum_{p \in \mathcal{U}} (1 + \mathsf{deg}(p)) = \mathsf{sz}(\mathcal{U})$$

Asumiendo que las operaciones sobre conjuntos pueden realizarse en tiempo constante, el costo de evaluar cada par en el ciclo principal del algoritmo es el de decidir si suspender o invalidar un nodo y, en el última caso, el costo del proceso de invalidación en sí mismo. La decisión de a dónde transferir el par actual se computa en la función check, que siempre realiza un número constante de operaciones sobre tipos no-unión. El peor caso involucra verificar pares de la forma $\langle \oplus_i^n \mathcal{A}_i, \oplus_j^m B_j \rangle$, lo que puede ser resuelto manteniendo en cada nodo una tabla indicando, para cada $\mathcal{A}_i$, el número de pares $\langle \mathcal{A}_i, \mathcal{B}_j \rangle$ que aún no fueron invalidados. Utilizando este enfoque, la verificación puede ser realizada en $\mathcal{O}(d)$ operaciones, donde d es una cota para el tamaño de ambas uniones. Siempre que un par es invalidado, las tablas presentes en sus padres (inmediatos) se actualizan adecuadamente.

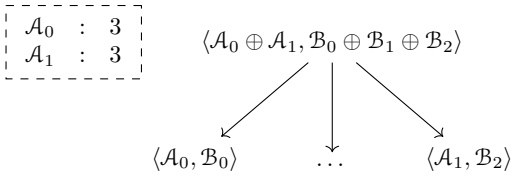


Figura 3.16: Verificación de descendientes invalidados.

Finalmente, el siguiente argumento, presentado en [DPR05], justifica que el costo de invalidar un elemento puede entenderse como $\mathcal{O}(1)$ amortizado. Una nueva iteración se ejecuta por cada uno de los

$$\frac{}{a \simeq_{\mathrm{n}}^{\mathcal{R}} a}\ (\textsc{e-refl-up})$$

$$\frac{\mathcal{D}\ (\simeq_{\mathrm{n}}^{\mathcal{R}} \cup\, \mathcal{R})\ \mathcal{D}' \quad \mathcal{A}\ (\simeq_{\mathrm{n}}^{\mathcal{R}} \cup\, \mathcal{R})\ \mathcal{A}'}{\mathcal{D} \mathbin{@} \mathcal{A} \simeq_{\mathrm{n}}^{\mathcal{R}} \mathcal{D}' \mathbin{@} \mathcal{A}'}\ (\textsc{e-comp-up}) \qquad \frac{\mathcal{A}\ (\simeq_{\mathrm{n}}^{\mathcal{R}} \cup\, \mathcal{R})\ \mathcal{A}' \quad \mathcal{B}\ (\simeq_{\mathrm{n}}^{\mathcal{R}} \cup\, \mathcal{R})\ \mathcal{B}'}{\mathcal{A} \supset \mathcal{B} \simeq_{\mathrm{n}}^{\mathcal{R}} \mathcal{A}' \supset \mathcal{B}'}\ (\textsc{e-func-up})$$

$$\frac{(\mathcal{A}_i\ (\simeq_{\mathrm{n}}^{\mathcal{R}} \cup\, \mathcal{R})\ \mathcal{B})_{i<n} \quad (\mathcal{A}_i \neq \oplus)_{i<n} \quad \mathcal{B} \neq \oplus}{\oplus_i^n \mathcal{A}_i \simeq_{\mathrm{n}}^{\mathcal{R}} \mathcal{B}}\ (\textsc{e-union-l-up})$$

$$\frac{(\mathcal{A}\ (\simeq_{\mathrm{n}}^{\mathcal{R}} \cup\, \mathcal{R})\ \mathcal{B}_j)_{j<m} \quad \mathcal{A} \neq \oplus \quad (\mathcal{B}_j \neq \oplus)_{j<m}}{\mathcal{A} \simeq_{\mathrm{n}}^{\mathcal{R}} \oplus_j^m \mathcal{B}_j}\ (\textsc{e-union-r-up})$$

$$\frac{\begin{array}{lll}(\mathcal{A}_i\ (\simeq_{\mathrm{n}}^{\mathcal{R}} \cup\, \mathcal{R})\ \mathcal{B}_{f(i)})_{i<n} & f : [0,n) \to [0,m) & (\mathcal{A}_i \neq \oplus)_{i<n} \\ (\mathcal{A}_{g(j)}\ (\simeq_{\mathrm{n}}^{\mathcal{R}} \cup\, \mathcal{R})\ \mathcal{B}_j)_{j<m} & g : [0,m) \to [0,n) & (\mathcal{B}_j \neq \oplus)_{j<m}\end{array}}{\oplus_i^n \mathcal{A}_i \simeq_{\mathrm{n}}^{\mathcal{R}} \oplus_j^m \mathcal{B}_j}\ (\textsc{e-union-up})$$

Figura 3.17: Equivalencia *up-to* $\mathcal{R}$ para árboles n-arios.

$\mathsf{deg}(p)$ pares agregados a $\mathcal{W}$ al invalidar p. Dado que se asume que agregar un elemento a un conjunto tiene costo $\mathcal{O}(1)$, puede entenderse que el costo de la invalidación se cuenta como una operación más en la iteración que verificara a cada uno de los elementos en cuestión en la continuidad del algoritmo. Luego, el costo total de cada iteración resulta ser $\mathcal{O}(d)$ amortizado, por lo expuesto anteriormente sobre el la función check. Esto da un costo global de $\mathcal{O}(\mathsf{sz}(\mathcal{U})d)$ para el chequeo de sub-tipado, *i.e.* $\mathcal{O}(NMd)$ en términos del tamaño de los autómatas de entrada.

Sean n y m el número de constructores en los μ-tipos contractivos A y B respectivamente. Los tamaños N y M de los autómatas que representan a estos tipos pueden considerarse acotados por $\mathcal{O}(n^2)$ y $\mathcal{O}(m^2)$ respectivamente. Por lo tanto, la complejidad del algoritmo de chequeo de sub-tipado es $\mathcal{O}(n^2m^2d)$ en términos del tamaño de su entrada.

Chequeo de equivalencia

Si bien el algoritmo de chequeo de equivalencia no es invocado directamente por tc, se ilustra a continuación cómo puede adaptarse el esquema general desarrollado para sub-tipado, mostrando así la versatilidad del mismo y obteniendo finalmente una presentación completa de algoritmos para el sistema de tipos propuesto.

Análogamente a lo presentado anteriormente, se introduce una noción de equivalencia co-inductiva sobre árboles (infinitos) n-arios, que se demuestra isomorfa a la relación $\simeq_\mu$ presentada en la Sec. 3.3.1.

Definición 3.6.22. *La relación de* equivalencia $\simeq_{\mathrm{n}}^{\mathcal{R}}$ *de árboles n-arios infinitos* up-to *un conjunto de hipótesis $\mathcal{R}$ se define como la interpretación co-inductiva de los esquemas de regla de la Fig. 3.17.*

Formalmente, sea $\Phi_{\simeq_{\mathrm{n}}^{\mathcal{R}}} : \wp(\mathfrak{T} \times \mathfrak{T}) \to \wp(\mathfrak{T} \times \mathfrak{T})$ la función generadora asociada a las reglas de la

Fig. 3.17, definido como

$$\begin{array}{rcl}
\Phi_{\simeq_{\mathtt{n}}^{\mathcal{R}}}(\mathcal{X}) & \triangleq & \{\langle a, a\rangle \mid a \in \mathcal{V} \cup \mathcal{C}\} \\
& \cup & \{\langle \mathcal{D} \mathbin{@} \mathcal{A}, \mathcal{D}' \mathbin{@} \mathcal{A}'\rangle \mid \langle \mathcal{D}, \mathcal{D}'\rangle, \langle \mathcal{A}, \mathcal{A}'\rangle \in \mathcal{X} \cup \mathcal{R}\} \\
& \cup & \{\langle \mathcal{A} \supset \mathcal{B}, \mathcal{A}' \supset \mathcal{B}'\rangle \mid \langle \mathcal{A}, \mathcal{A}'\rangle, \langle \mathcal{B}, \mathcal{B}'\rangle \in \mathcal{X} \cup \mathcal{R}\} \\
& \cup & \{\langle \oplus_i^n \mathcal{A}_i, \mathcal{B}\rangle \mid \mathcal{A}_i, \mathcal{B} \neq \oplus, \langle \mathcal{A}_i, \mathcal{B}\rangle \in \mathcal{X} \cup \mathcal{R}\} \\
& \cup & \{\langle \mathcal{A}, \oplus_j^m \mathcal{B}_j\rangle \mid \mathcal{A}, \mathcal{B}_j \neq \oplus, \langle \mathcal{A}, \mathcal{B}_j\rangle \in \mathcal{X} \cup \mathcal{R}\} \\
& \cup & \{\langle \oplus_i^n \mathcal{A}_i, \oplus_j^m \mathcal{B}_j\rangle \mid \mathcal{A}_i, \mathcal{B}_j \neq \oplus, \\
& & \quad \exists f : [0,n) \to [0,m).\langle \mathcal{A}_i, \mathcal{B}_{f(i)}\rangle \in \mathcal{X} \cup \mathcal{R}, \\
& & \quad \exists g : [0,m) \to [0,n).\langle \mathcal{A}_{g(j)}, \mathcal{B}_j\rangle \in \mathcal{X} \cup \mathcal{R}\}
\end{array}$$

Luego, $\simeq_{\mathtt{n}}^{\mathcal{R}} \triangleq \nu\Phi_{\simeq_{\mathtt{n}}^{\mathcal{R}}}$. Notar que esta relación resulta de reemplazar las reglas (S-FUNC-UP), (S-UNION-R-UP) y (S-UNION-UP) de la Fig. 3.13 por sus respectivas contra-partes de la Fig. 3.17, manteniendo las restantes reglas inalteradas. Al igual que antes, la relación *up-to* el conjunto de hipótesis vacío resulta isomorfa a la relación de equivalencia co-inductiva sobre tipos infinitos (*i.e.* árboles binarios) vía la traducción.

Lema 3.6.23. *$\mathcal{A} \simeq_{\mathfrak{T}} \mathcal{B}$ sii $\mathcal{A} \simeq_{\mathtt{n}}^{\emptyset} \mathcal{B}$.*

Por lo tanto, $\simeq_{\mathtt{n}}^{\emptyset}$ también se corresponde con la noción de equivalencia principal del sistema $\simeq_{\mu}$ a través de la traducción de μ-contractivos a tipos infinitos.

Teorema 3.6.24. *$A \simeq_{\mu} B$ sii $[\![A]\!]^{\mathtt{n}} \simeq_{\mathtt{n}}^{\emptyset} [\![B]\!]^{\mathtt{n}}$.*

Demostración. Por Teo. 3.3.28 y Lem. 3.6.23: $A \simeq_{\mu} B$ sii $[\![A]\!]^{\mathfrak{T}} \simeq_{\mathfrak{T}} [\![B]\!]^{\mathfrak{T}}$ sii $[\![A]\!]^{\mathtt{n}} \simeq_{\mathtt{n}}^{\emptyset} [\![B]\!]^{\mathtt{n}}$. □

Asimismo, el algoritmo $\mathsf{gfp}^{\simeq}$ es el resultado de adaptar el esquema presentado para sub-tipado a la nueva relación $\simeq_{\mathtt{n}}^{\mathcal{R}}$. Esto se logra simplemente re-definiendo las funciones children y check de la primera y segunda fase respectivamente (*cf.* Fig. 3.18). Para el primer caso la única diferencia se da en la regla (E-FUNC-UP), donde el nuevo par $\langle \mathcal{A}', \mathcal{B}'\rangle$ es agregado al conjunto en lugar del original $\langle \mathcal{B}', \mathcal{A}'\rangle$ utilizado para sub-tipado. Esto podría haber sido evitado apelando a la simetría de la relación de equivalencia, pero busca enfatizar el hecho de que la primera fase del algoritmo puede ser adaptada fácilmente de acuerdo a la necesidad de cada relación. Para la fase de refinamiento es necesario verificar adecuadamente las premisas de las nuevas reglas (E-FUNC-UP), (E-UNION-R-UP) y (E-UNION-UP), siendo esta última la más interesante, mientras que las restantes no requieren modificaciones al pseudo-código.

Bajo estas consideraciones es inmediato ver que, en cada iteración, el conjunto $\mathcal{S}$ consiste de pares en la relación $\simeq_{\mathtt{n}}^{\mathcal{W}}$, obteniendo $\mathcal{S} \subseteq \simeq_{\mathtt{n}}^{\emptyset}$ al final del proceso.

Teorema 3.6.25. *El algoritmo $\mathsf{gfp}^{\simeq}$ preserva la siguiente invariante:*

- *$\langle \mathcal{W}, \mathcal{S}, \mathcal{F}\rangle$ es una partición de $\mathcal{U}$.*
- *$\mathcal{F}$ consiste únicamente de pares inválidos.*
- *$\mathcal{S} \subseteq \Phi_{\simeq_{\mathtt{n}}^{\mathcal{W}}}(\mathcal{S})$*

Demostración. El análisis es exactamente igual al caso de sub-tipado. La única diferencia es al demostrar la segunda condición, donde se necesita garantizar que los pares son considerados válidos de acuerdo a las reglas de la Fig. 3.17. Se muestra a continuación aquellos casos que difieren respecto a $\preceq_{\mathtt{n}}^{\mathcal{W}}$:

- $p = \langle \mathcal{A}' \supset \mathcal{A}'', \mathcal{B}' \supset \mathcal{B}''\rangle$. Este caso corresponde a (E-FUNC-UP) y el algoritmo modificado verifica $\langle \mathcal{A}', \mathcal{B}'\rangle$ en lugar de $\langle \mathcal{B}', \mathcal{A}'\rangle$. Esa es la razón por la que children fue re-definida (*cf.* Fig. 3.18).

```
children(p) :
    case p of
      ⟨D @ A, D' @ B⟩ →
          {⟨D, D'⟩, ⟨A, B⟩}
      ⟨A' ⊃ A'', B' ⊃ B''⟩ →
          {⟨A', B'⟩, ⟨A'', B''⟩}
      ⟨⊕_i^n A_i, ⊕_j^m B_j⟩ →
          {⟨A_i, B_j⟩ | i < n, j < m}
      ⟨⊕_i^n A_i, B⟩, B ≠ ⊕ →
          {⟨A_i, B⟩ | i < n}
      ⟨A, ⊕_j^m B_j⟩, A ≠ ⊕ →
          {⟨A, B_j⟩ | j < m}
      otherwise →
          ∅
```

```
check(p, F) :
    case p of
      ⟨a, a⟩ →
          true
      ⟨D @ A, D' @ B⟩ →
          ⟨D, D'⟩ ∉ F and ⟨A, B⟩ ∉ F
      ⟨A' ⊃ A'', B' ⊃ B''⟩ →
          ⟨A', B'⟩ ∉ F and ⟨A'', B''⟩ ∉ F
      ⟨⊕_i^n A_i, B⟩, B ≠ ⊕ →
          ∀i.⟨A_i, B⟩ ∉ F
      ⟨A, ⊕_j^m B_j⟩, A ≠ ⊕ →
          ∀j.⟨A, B_j⟩ ∉ F
      ⟨⊕_i^n A_i, ⊕_j^m B_j⟩ →
          ∀i.∃j.⟨A_i, B_j⟩ ∉ F and ∀j.∃i.⟨A_i, B_j⟩ ∉ F
      otherwise →
          false
```

Figura 3.18: Pseudo-código de las modificaciones necesarias para equivalencia.

- $\langle \mathcal{A}, \oplus_j^m \mathcal{B}_j \rangle$. Por (E-UNION-R-UP), es necesario que $(\langle \mathcal{A}, \mathcal{B}_j \rangle \in \mathcal{S} \cup \mathcal{W})_{j<m}$ para mover p al conjunto de pares condicionalmente válidos.

- $p = \langle \oplus_i^n \mathcal{A}_i, \oplus_j^m \mathcal{B}_j \rangle$. En el caso de sub-tipado, el algoritmo verifica que para cada $\mathcal{A}_i$ exista un $\mathcal{B}_j$ tal que el par $\langle \mathcal{A}_i, \mathcal{B}_j \rangle$ no haya sido invalidado aún (*cf.* la función f en (S-UNION-UP)). Al extender el chequeo a equivalencia es necesario verificar también el sentido opuesto, *i.e.* para cada $\mathcal{B}_j$ existe un $\mathcal{A}_i$ tal que el par $\langle \mathcal{A}_i, \mathcal{B}_j \rangle$ no haya sido invalidado aún, garantizando así la existencia de la función g en (E-UNION-UP).

□

La corrección de la presentación algorítmica de la equivalencia de tipos se sigue tal como en el caso de sub-tipado (Cor. 3.6.21), pues $\mathsf{gfp}^{\simeq}$ se comporta de modo similar a $\mathsf{gfp}^{\preceq}$ excepto que la función de chequeo fue adaptada para la equivalencia.

Corolario 3.6.26. $\mathcal{A} \simeq_{\mathsf{n}}^{\emptyset} \mathcal{B}$ *sii* $\mathsf{gfp}^{\simeq}(\langle \mathcal{A}, \mathcal{B} \rangle) = \mathtt{true}$.

Para el análisis de complejidad, recordar que N y M denotan el tamaño de los autómatas que representan los tipos a chequear. Notar que el tamaño del universo construido es el mismo que antes, por lo que el costo total del algoritmo sigue estando gobernado por la segunda fase, que tiene a lo sumo $\mathcal{O}(NM)$ iteraciones. Para el costo de cada iteración es suficiente con analizar la complejidad de check, dado que las funciones restantes no han sido modificadas. Tal como se remarcó anteriormente, la única diferencia significativa en check entre sub-tipado y equivalencia es en los casos que involucran uniones. Allí, el peor caso es cuando se chequea (E-UNION-UP) pues requiere la existencia de dos funciones f y g que relacionan las sub-expresiones de tipo. Esto se puede realizar en tiempo lineal manteniendo una tabla con contadores de pares de descendientes no invalidados, como se detalló anteriormente. Por lo tanto, el costo de una iteración es $\mathcal{O}(d)$, resultando en un costo total de $\mathcal{O}(NMd)$ al igual que antes.

Chequeo de tipos

Volviendo sobre el algoritmo de chequeo de tipos (tc), como se discutió anteriormente, éste recorre de manera lineal el término de entrada siendo las cláusulas más costosas aquellas que resuelven los casos de la abstracción y la aplicación. Estos casos involucran llamados a $\mathsf{gfp}^{\preceq}$. Notar que estos llamados no dependen directamente de la entrada a la función tc. Sin embargo, puede establecerse una relación lineal entre el tamaño de los tipos considerados al invocar $\mathsf{gfp}^{\preceq}$ y el tamaño de la entrada al algoritmo de chequeo de tipos, dado que estas expresiones se construyen a partir de elementos en Γ (el contexto de entrada) o anotaciones de tipo en el propio término t. Considerar por ejemplo $\mathsf{gfp}^{\preceq}(\langle A, B\rangle)$ con a y b los tamaños de cada tipo en cuestión. La complejidad de este llamado es $\mathcal{O}(a^2b^2d)$ y, por lo discutido anteriormente, se puede considerar equivalente a $\mathcal{O}(n^4d)$, donde n es el tamaño de la entrada a tc (*i.e.* Γ más t).

Por otro lado, el cambio de representación a *term automata* simplifica notablemente la función unfold: ya no es necesario aplicar sustituciones para desdoblar el tipo, sino que basta con verificar que el estado inicial del autómata que representa al tipo A se encuentre etiquetado con un tipo funcional, o sea una unión n-aria cuyos hijos sean todos tipos funcionales. En el peor caso esto involucra una cantidad lineal de chequeos y, dado que el tamaño de A está en relación lineal con el tamaño de la entrada, se puede decir que la complejidad de unfold es $\mathcal{O}(n)$.

Considerar en detalle los casos de la abstracción y la aplicación en tc:

Abstracción En primer lugar se realizan tantos llamados a tcp (el algoritmo de chequeo de tipos para patrones) como ramas tiene la abstracción. Notar que tcp tiene complejidad lineal en el tamaño de sus entradas y, además, estas llamadas son realizadas con argumentos p_i y θ_i que ocurren en el término original. Todas estas llamadas juntas pueden considerarse, por lo tanto, de una complejidad temporal lineal con respecto a la entrada de tc. Luego se realiza una cantidad cuadrática (en el número de ramas de la abstracción) de chequeos de compatibilidad. Como se analizó anteriormente, la compatibilidad involucra, en el peor caso, una llamada a $\mathsf{gfp}^{\preceq}$. Si se asume una cantidad lineal de ramas en la abstracción, con respecto al tamaño de la entrada, se obtiene entonces una complejidad $\mathcal{O}(n^6d)$ para este caso.

Aplicación En primer lugar se realiza un chequeo lineal sobre el tipo para verificar que se trata de un datatype. Si es el caso, se retorna. Si no, un segundo chequeo lineal es necesario (unfold) para luego realizar tantas llamadas a $\mathsf{gfp}^{\preceq}$ como elementos haya en la unión de tipos funcionales resultante. Esto resulta en una complejidad local de $\mathcal{O}(n^4d)$.

Finalmente, la complejidad total del algoritmo tc queda gobernada por el caso de las abstracciones, resultando en un orden de complejidad $\mathcal{O}(n^7d)$ (en el peor caso habría una cantidad lineal de llamados recursivos sobre abstracciones).

3.6.4. Prototipos de implementación

Existen dos prototipos del algoritmo de chequeo de tipos aquí propuesto, ambos realizados por Juan Edi como parte de su Tesis de Licenciatura [Edi15]. Un primer prototipo implementa en lenguaje Haskell la versión ingenua de tc utilizando una representación algebraica de los μ-tipos contractivos e incluyendo los algoritmos para chequeo de equivalencia y sub-tipado propuestos en la Sec. 3.6.1. Una segunda implementación en lenguaje Scala se centra en la versión eficiente de la Sec. 3.6.3 con la representación de tipos recursivos como *term automata*. Adicionalmente, este último prototipo incorpora una serie de optimizaciones avanzadas, sugeridas en [DPR05], entre la que se destaca un algoritmo para la selección del próximo par de $\mathcal{W}$ a ser analizado en $\mathsf{gfp}^{\simeq}$ y $\mathsf{gfp}^{\preceq}$, basado en la detección de componentes fuertemente conexas en los autómatas, utilizando el algoritmo de Tarjan [DST80] de costo lineal y ordenando dichas componentes en orden topológico inverso. En la ausencia de ciclos,

este orden de selección resulta en la evaluación de un par solo una vez que todos sus descendientes ya fueron considerados. En presencia de ciclos, los pares para los cuales no puede determinarse un orden son encapsulados dentro de la misma componente fuertemente conexa.

3.7. Conclusión

En el presente capítulo se introduce el *Calculus of Applicative Patterns* (CAP) como una variante del fragmento estático de PPC donde la alternativa es una operación nativa (*i.e.* un constructor de términos del cálculo) capaz, a su vez, de capturar path polymorphism. Se desarrolla un sistema de tipos para el mismo donde términos *path-polimórficos* son tipados estáticamente, garantizando el buen comportamiento dinámico de los mismos: subject reduction y progress. El sistema combina distintas herramientas como tipos constantes, aplicativos, unión y recursivos, al mismo tiempo que introduce una noción novedosa de compatibilidad entre patrones de la cual depende crucialmente la adecuación del sistema.

Adicionalmente, se estudia la posibilidad de chequear algorítmicamente la asignación de tipos, lo cual requiere de una reformulación dirigida por sintaxis del sistema. A su vez, es necesario desarrollar algoritmos para el chequeo de equivalencia de tipos y sub-tipado, lo que lleva a introducir formulaciones co-inductivas e invertibles de dichas relaciones. Un primer enfoque es desarrollado derivando en un algoritmo correcto pero ineficiente en términos de complejidad computacional. En una segunda etapa se detectan los puntos débiles del algoritmo obtenido anteriormente y se desarrolla a partir de este estudio un algoritmo eficiente. Este último se logra principalmente mediante un adecuado cambio de representación, dejando de lado la versión algebraica de la primera etapa e interpretando los μ-tipos contractivos como *term automata*. Finalmente, el algoritmo obtenido resulta tener complejidad polinomial respecto al tamaño de su entrada.

Respecto a posibles líneas de trabajo futuro relacionadas a esta investigación, se busca principalmente enriquecer la expresividad del sistema con vista a la implementación de un prototipo de lenguaje de programación funcional basado en él:

- Una primera extensión busca incorporar polimorfismo paramétrico al desarrollo, en el estilo de *System* $F_{<:}$ [CMMS91, Pie02, CG05], lo que implica adaptar los resultados obtenidos a la presencia de tipos polimórficos teniendo, *a priori*, un impacto directo en la noción de compatibilidad.

- En los últimos años se han introducido estrategias de reducción normalizantes para PPC [BKLR12, BKLR17], las cuales deberían ser adaptadas al marco de CAP para el adecuado desarrollo de un prototipo funcional. Cabe destacar que estos dos cálculos comparten la misma operación de matching, la cual requiere de un chequeo en paralelo en el caso de la aplicación (*cf.* Sec. 3.1.2, $\{\!\{p\,q \setminus t\,u\}\!\} \triangleq \{\!\{p \setminus t\}\!\} \uplus \{\!\{q \setminus u\}\!\}$) para evitar situaciones de no terminación innecesarias producto de la evaluación secuencial de los sub-términos (*cf.* Sec. 3.1.1, $\mu \uplus \texttt{fail} \triangleq \texttt{fail} \uplus \mu \triangleq \texttt{fail}$).

- Es usual utilizar índices de de Bruijn [Bar85, BKvO03] al representar términos del λ-cálculo, para garantizar que no se realizan capturas indeseadas de variables libres producto de un mal manejo de sus nombres. Recientemente se desarrolló una versión *à la* de Bruijn de PPC [MRV20], la cual podría ser adaptada a CAP para obtener una representación adecuada de sus términos en el eventual prototipo.

- Garantizar normalización fuerte en presencia de tipos recursivos con equivalencia fuerte constituye un problema abierto en el área por más de 20 años [BDS13, pág. 515]. Si bien se han logrado algunos avances al respecto en el marco de CAP, no se ha llegado aún a un resultado definitivo.

- Una extensión más ambiciosa es la de incorporar *pattern polymorphism* al desarrollo. Esto implica reincorporar patrones dinámicos, tal cual lo hace PPC, manteniendo al mismo tiempo las buenas

propiedades del sistema, que hoy en día recaen en la noción meramente estática de compatibilidad.

Capítulo 4

Bisimulación fuerte para operadores de control

Un aspecto importante del estudio de la teoría de los lenguajes de programación es el de identificar similitudes estructurales en las expresiones que denotan programas. Esto se conoce comúnmente como *equivalencia estructural*: expresiones diferentes que se comportan exactamente de la misma manera. Los cálculos de procesos son una fuente inagotable de ejemplos. En el Calculus of Communicating Systems (CCS) [Mil80] las expresiones denotan procesos en un sistema concurrente. Por ejemplo, $P \parallel Q$ denota la composición paralela de los procesos P y Q. La equivalencia estructural en tal marco incluye ecuaciones para identificar las expresiones $P \parallel Q$ y $Q \parallel P$. Este mínimo reordenamiento de las sub-expresiones tiene poco impacto (sino nulo) en el comportamiento de la expresión general: la equivalencia estructural es una *bisimulación fuerte* para la reducción de procesos. El presente capítulo estudia estas nociones de reordenamiento de expresiones en el marco del *λ-cálculo con operadores de control*. La noción de equivalencia estructural inducida, denotada $\simeq$, debe identificar términos que tengan exactamente la misma semántica de reducción, esto es, debe ser una bisimulación fuerte con respecto a la reducción en el cálculo propuesto. En otras palabras, $\simeq$ debe ser simétrica al mismo tiempo que $o \simeq o'$ y $o \rightsquigarrow p$ implican la existencia de p' tal que $p \rightsquigarrow p'$ y $o' \simeq p'$, donde $\rightsquigarrow$ denota alguna noción de reducción dada para el cálculo con operadores de control. Gráficamente

$$\begin{array}{ccc} o & \simeq & p \\ \downsquigarrow & & \vdots \\ o' & \simeq & p' \end{array}$$

La formulación de tales equivalencias estructurales en el λ-cálculo se encuentra obstaculizada por la orientación secuencial en que las expresiones son escritas (de izquierda a derecha). Considerar por ejemplo los términos $(\lambda x.(\lambda y.t)\, u)\, v$ y $(\lambda x.\lambda y.t)\, v\, u$. Podría decirse que estos términos contienen los mismos redexes, solo que permutados; algo similar a lo capturado por las ecuaciones CCS mencionadas anteriormente. Sin embargo, un análisis en detalle revela que esto no es completamente cierto. El primer término contiene dos redexes mientras que el segundo solo uno, resaltados a continuación:

$$\underline{(\lambda x.\overline{(\lambda y.t)\, u})\, v} \qquad \text{y} \qquad \underline{(\lambda x.(\lambda y.t))\, v\, u} \tag{4.1}$$

El redex indicado con la línea superior en el primer término no se encuentra visible en el segundo; reaparecerá, sin embargo, como un redex *creado* al reducir el subrayado. Más allá del hecho de que la sintaxis se interpone, Regnier [Reg94] muestra que estos términos se comportan *esencialmente* de la misma manera. Más precisamente, introduce una noción de equivalencia sobre λ-términos, conocida

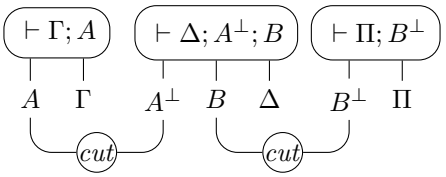
como *σ-equivalencia*, y prueba que términos σ-equivalentes tiene reducciones head, leftmost, perpetual y, más generalmente, maximales de la misma longitud. Sin embargo, no se obtiene un bisimulación fuerte como puede apreciarse en el ejemplo (4.1). Esta bisimulación parece estar subyacente pero no se manifiesta debido a la poco conveniente notación secuencial. Resulta ser que, mediante la intuición gráfica provista por las *proof-nets* de lógica lineal [Gir87], es posible definir un λ-cálculo con sintaxis enriquecida que permite desvelar la bisimulación subyacente para el caso intuicionista [ABKL14]. El presente trabajo apela a esta misma intuición para explorar hasta qué punto es posible obtener una bisimulación fuerte para una noción adecuada de equivalencia estructural en el desafiante marco de la lógica clásica. Es por esto que no solo se busca capturar la equivalencia estructural sobre funciones puras, sino que también sobre programas con *operadores de control*. En este caso, el criterio semántico estará guiado por las *proof-nets polarizadas* (PPN) [Lau02]. A continuación se presentan las proof-nets, para luego argumentar cómo ayudan a formalizar una bisimulación fuerte para la equivalencia estructural del λ-cálculo. Luego se exponen los desafíos encontrados al estudiar el caso clásico.

Proof-nets

Una *proof-net* es una estructura en forma de grafo cuyos nodos denotan inferencias lógicas y sus ejes o aristas denotan las formulas sobre las que estas inferencias operan (*cf.* Sec. 4.1.3). Fueron originalmente introducidas en el marco de la *lógica lineal* [Gir87], la cual provee un mecanismo para controlar explícitamente el uso de recursos al restringir la aplicación de las reglas *estructurales* de *weakening* y *contraction*. Las proof-nets tienen su propia semántica operacional, especificada como reglas de transformación de grafos, la cual captura la noción de *cut-elimination* del cálculo de secuentes. Las reglas de cut-elimination resultantes sobre proof-nets se dividen en dos grupos: *multiplicativas*, que esencialmente reconfiguran (linealmente) los ejes del grafo; y *exponenciales*, que son las únicas capaces de borrar o duplicar (sub-)proof-nets. Estas últimas se considera que introducen cómputo *significativo* (*meaningful*). Es interesante destacar que las proof-nets abstraen el orden en que ciertas reglas ocurren en una derivación de secuentes. Por ejemplo, asumir tres derivaciones de los juicios $\vdash \Gamma; A$, $\vdash \Delta; A^{\perp}; B$ y $\vdash \Pi; B^{\perp}$ respectivamente. Luego, en el cálculo de secuentes pueden generarse las siguiente derivaciones para $\vdash \Gamma; \Delta; \Pi$ mediante *cuts*:

$$\dfrac{\vdash \Gamma; A \qquad \dfrac{\vdash \Delta; A^{\perp}; B \quad \vdash \Pi; B^{\perp}}{\vdash \Delta; A^{\perp}; \Pi}}{\vdash \Gamma; \Delta; \Pi} \qquad \dfrac{\dfrac{\vdash \Gamma; A \quad \vdash \Delta; A^{\perp}; B}{\vdash \Gamma; \Delta; B} \qquad \vdash \Pi; B^{\perp}}{\vdash \Gamma; \Delta; \Pi}$$

Sin embargo, estas diferentes derivaciones son abstraídas en una única proof-net:



En otras palabras, *diferentes* términos/derivaciones son representados por la *misma* proof-net. De este modo, la similitud estructural a veces oculta entre distintos términos puede ser estudiada traduciéndolos a proof-net. Más aún, mediante el isomorfismo de Curry–Howard, que relaciona la computación y la lógica, esta correspondencia puede ser extendida no solo a los mismos términos [DR95, CKP00, KL05, AK12] sino también a su comportamiento dinámico [Acc18]. Este trabajo, sin embargo, se concentra en identificar aquellas derivaciones diferentes de la lógica clásica que pueden ser traducidas a la misma representación como grafo. Como es estándar en la literatura, la noción de igualdad de proof-nets adoptada incluye equivalencias simples como la *asociatividad* para nodos de contracción entre otras similares (*cf.* noción de equivalencia estructural para proof-nets en la Sec. 4.1.3).

$$\begin{array}{rcll} (\lambda x.\lambda y.t)\, u & \simeq_{\sigma_1} & \lambda y.(\lambda x.t)\, u & y \notin u \\ (\lambda x.t\, v)\, u & \simeq_{\sigma_2} & (\lambda x.t)\, u\, v & x \notin v \end{array}$$

Figura 4.1: σ-equivalencia para términos del λ-cálculo.

$$\begin{array}{rcl} (\lambda x.t)[x_1 \setminus v_1] \ldots [x_n \setminus v_n]\, u & \mapsto_{\mathtt{dB}} & t[x \setminus u][x_1 \setminus v_1] \ldots [x_n \setminus v_n] \\ t[x \setminus u] & \mapsto_{\mathtt{S}} & \{x \setminus u\}t \end{array}$$

Figura 4.2: Descomposición de β.

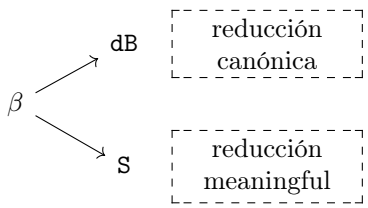
Sigma-equivalencia intuicionista

Como se mencionó anteriormente, Regnier introduce una noción de σ-equivalencia, notada $\simeq_\sigma$, sobre términos del λ-cálculo (*cf.* Fig. 4.1) y muestra que términos σ-equivalentes se comportan esencialmente de la misma manera. Esta relación de equivalencia involucra la permutación de ciertos redexes y fue descubierta mediante el estudio de las proof-nets. En particular, siguiendo la codificación de Girard de la lógica intuicionista en la lógica lineal [Gir87], términos σ-equivalentes son mapeados a la misma proof-net (módulo cuts multiplicativos y equivalencia estructural).

La razón por la cual el resultado de Regnier no es inmediato es que algunos redexes presentes a un lado de la igualdad pueden no estarlo del otro, como se ilustra en el ejemplo (4.1). Puede replantearse esta observación estableciendo que $\simeq_\sigma$ *no* es una bisimulación fuerte sobre el conjunto de términos del λ-cálculo. Si lo fuese, establecer que dos términos σ-equivalentes se comportan esencialmente de la misma manera sería trivial.

Adoptando una mirada más refinada sobre el λ-cálculo, como sugiere la propia lógica lineal al separar el proceso de *cut-elimination* sobre derivaciones en pasos multiplicativos y exponenciales, resulta en una descomposición de la β-reducción en pasos multiplicativos/exponenciales sobre términos. La teoría de *sustituciones explícitas* (ES) [Kes09] proporciona una sintaxis conveniente para reflejar estos pasos a nivel de términos. De hecho, la β-reducción puede descomponerse en dos pasos tal cual ilustra en la Fig. 4.2, a saber: `dB` (por `d`istant `B`eta), que actúa *a distancia* [AK12] en el sentido que una abstracción y su argumento pueden estar separados por un número arbitrario de sustituciones explícitas; y `S` (por `S`ubstitution).

La aplicación de la regla `dB` genera una *sustitución explicita* $t[x \setminus u]$ donde la variable x esta ligada en el cuerpo t de la sustitución y u es el argumento de la misma, tal cual sucede en $(\lambda x.t)\, u$. De modo que `dB` simplemente reconfigura símbolos. De hecho, `dB` se entiende como un cut multiplicativo al traducirla a *proof-nets*. La regla `S` en cambio, ejecuta la sustitución al reemplazar todas las ocurrencias libres de x en t por u, por lo que es `S` la que realiza el cómputo *significativo* (o *meaningful*) y es de hecho interpretada como cuts exponenciales en las proof-nets. Un término sin sub-términos de la forma $(\lambda x.t)[x_1 \setminus v_1] \ldots [x_n \setminus v_n]\, u$ es llamado una `dB`-*forma normal*. Alternativamente se denomina a los mismos *formas canónicas*. Este criterio de descomposición de la β-reducción puede resumirse gráficamente como:



Esto lleva a reemplazar $\simeq_\sigma$ (Fig. 4.1) por una nueva relación $\simeq_{\sigma^B}$ (Fig. 4.3). Esta última resulta esencialmente de tomar la `dB`-forma normal a cada lado de las ecuaciones de $\simeq_\sigma$, agregando la ecuación

$$\begin{array}{rll} (\lambda y.t)[x \setminus u] & \simeq_{\sigma_1^B} & \lambda y.t[x \setminus u] \qquad y \notin u \\ (t\,v)[x \setminus u] & \simeq_{\sigma_2^B} & t[x \setminus u]\,v \qquad x \notin v \\ t[y \setminus v][x \setminus u] & \simeq_{\sigma_3^B} & t[x \setminus u][y \setminus v] \quad x \notin v, y \notin u \end{array}$$

Figura 4.3: σ^B-equivalencia para términos del λ-cálculo con ES.

adicional $\simeq_{\sigma_3^B}$ que permite conmutar sustituciones ortogonales (independientes). Notar, sin embargo, que la dB-expansión de $\simeq_{\sigma_3^B}$ resulta en términos σ-equivalentes, pues $t[y \setminus v][x \setminus u] \simeq_{\sigma^B} t[x \setminus u][y \setminus v]$, con $x \notin v$ y $y \notin u$, dB-expande a $(\lambda y.(\lambda x.t)\,u)\,v \simeq_\sigma (\lambda x.(\lambda y.t)\,v)\,u$, σ-equivalentes por $\simeq_{\sigma_1}$ y $\simeq_{\sigma_2}$.

Dado que la dB-reducción corresponde únicamente a cuts multiplicativos en las proof-nets, la traducción de términos σ^B-equivalentes simplemente tipados también resulta en proof-nets *estructuralmente equivalentes*. En otras palabras, clases de términos del λ-cálculo con ES σ^B-equivalentes en dB-forma normal están en correspondencia uno a uno con las proof-nets de la lógica lineal intuicionista [AK12]. Más aún, $\simeq_{\sigma^B}$ es una bisimulación fuerte con respecto a la noción de reducción meaningful (*i.e.* S-reducción) sobre el conjunto de términos extendido que incluye sustituciones explícitas [AK12, ABKL14]. En efecto, $\simeq_{\sigma^B}$ es simétrica y para todas dB-formas normales $u \simeq_{\sigma^B} v$ y $u \to_{\mathtt{S}} u'$ implican la existencia de un término v' tal que $u' \simeq_{\sigma^B} v'$ y $v \to_{\mathtt{S}} v'$.

$$\begin{array}{ccc} u & \simeq_{\sigma^B} & v \\ {\scriptstyle \mathtt{S}}\downarrow & & {\scriptstyle \mathtt{S}}\,\vdots \\ u' & \simeq_{\sigma^B} & v' \end{array}$$

Notar que las dB-formas normales de los términos en (4.1) son $\simeq_{\sigma^B}$-equivalentes, reparando así el contra-ejemplo.

Sigma-equivalencia clásica

Como ya se mencionó anteriormente, el objetivo de este trabajo es estudiar la equivalencia estructural para el λ-*cálculo con operadores de control*. Existen diversas extensiones posibles del λ-cálculo con tales operadores, que permiten manipular el contexto en el cual un programa es ejecutado [Lan65a, Lan65b, Rey70, FF87, Gri90]. En esta tesis el foco se pone sobre el $\lambda\mu$-cálculo de Parigot [Par92], que extiende al λ-cálculo con dos nuevos operadores: $[\alpha]\,t$ (*named term*) y $\mu\alpha.c$ (μ-*abstracción*). El primero puede entenderse como "llamar a la continuación α con t como argumento", mientras que el segundo como "guardar la continuación actual y continuar ejecutando c". La reducción en el $\lambda\mu$-cálculo consiste en la regla β junto con:

$$(\mu\alpha.c)\,u \quad \mapsto_\mu \quad \mu\alpha.\{\!\{\alpha \setminus u\}\!\}c$$

donde $\{\!\{\alpha \setminus u\}\!\}c$, llamado *replacement*, reemplaza todas las sub-expresiones de la forma $[\alpha]\,t$ por $[\alpha]\,(t\,u)$ en c. La noción de Regnier de σ-equivalencia para términos del λ-cálculo fue extendida al $\lambda\mu$-cálculo por Laurent [Lau03] (*cf.* Fig. 4.10 en Sec. 4.1.4). Como ejemplo de términos relacionados por esta extensión, considerar los siguientes casos, donde se resaltan a su vez los redexes presentes:

$$(\underline{(\lambda x.\mu\alpha.[\gamma]\,u)}\,w)\,v \simeq_\sigma \overline{(\mu\alpha.[\gamma]\,\underline{(\lambda x.u)\,w})\,v}$$

Una vez más, el hecho de que la permutación de redexes es inofensiva para la semántica operaciones dista de ser obvio. El término a la derecha tiene dos redexes presentes (μ y β respectivamente), pero el de la izquierda solo uno (β). Otro ejemplo, más sutil, de términos relacionados por la extensión de Laurent sugiere claramente que la indistinguibilidad operacional no puede recaer simplemente en

$$\begin{array}{rcl}(\mu\alpha.c)[x_1 \setminus v_1]\ldots[x_n \setminus v_n]\,u & \mapsto_{\mathtt{dM}} & (\mu\alpha'.c[\![\alpha \setminus^{\alpha'} u]\!])[x_1 \setminus v_1]\ldots[x_n \setminus v_n] \\ c[\![\alpha \setminus^{\alpha'} u]\!] & \mapsto_{\mathtt{R}} & \{\!\{\alpha \setminus^{\alpha'} u\}\!\}c\end{array}$$

Figura 4.4: Descomposición de β.

relacionar μ-redexes arbitrarios; el μ-redex subrayado a la izquierda no aparece (ni será generado) a la derecha:

$$\underline{(\mu\alpha.[\alpha]\,x)\,y} \simeq_\sigma x\,y \tag{4.2}$$

Es claro entonces que la σ-equivalencia para términos del $\lambda\mu$-cálculo falla en ser una bisimulación fuerte. Sin embargo, Laurent prueba propiedades para $\simeq_\sigma$ en $\lambda\mu$-cálculo similares a aquellas dadas por Regnier para $\simeq_\sigma$ en λ-cálculo. Parece ser entonces que es posible refinar el planteo explicado anteriormente en el caso intuicionista (*cf.* Sec. 4) para obtener una bisimulación fuerte basada en $\simeq_\sigma$ para el $\lambda\mu$-cálculo apelando a la semántica de proof-nets.

Hacia una bisimulación fuerte para operadores de control

Se busca formular una noción de equivalencia estructural para términos del $\lambda\mu$-cálculo, capaz de lidiar con permutaciones inofensivas de redexes (posiblemente involucrando operadores de control) y al mismo tiempo inducir una bisimulación fuerte con respecto a una noción de reducción meaningful adecuada para el cálculo. Tal como establece el isomorfismo de Curry–Howard, la normalización de pruebas en la lógica clásica se corresponde con el cómputo en λ-cálculo con operadores de control [Gri90, Par92]. Más aún, dado que la lógica clásica puede ser interpretada en las *proof-nets polarizadas* (PPNs) [Lau02, Lau03], se utilizará esta semántica para guiar el desarrollo del presente trabajo. Un primer paso hacia el objetivo involucra descomponer la regla de reducción μ al igual que se hace con β con la reglas de la Fig.4.2: esto produce una regla `dM` (por `d`istant `Mu`), que introduce un *replacement explícito* $[\![\alpha \setminus^{\alpha'} s]\!]$, que actuará a distancia; y otra regla `R` (por `R`eplacement), que ejecutará los replacements explícitos. La Fig. 4.4 ilustra esta descomposición, donde $\{\!\{\alpha \setminus^{\alpha'} u\}\!\}c$ reemplaza cada sub-expresión de la forma $[\alpha]\,t$ por $[\alpha']\,t\,u$ en c. Análogamente al caso de β, parece ser que la computación meaningful es llevada acabo por `R` más que por `dM`. Esta observación es respaldada por el hecho de que ambos lados de la regla `dM` serán traducidos en exactamente la misma PPN (*cf.* Sec. 4.4).

Por lo tanto, se fija tentativamente la noción de reducción meaningful en $\mathtt{S} \cup \mathtt{R}$ sobre el conjunto de formas canónicas, definido por lo pronto como $\mathcal{NF}_{\mathtt{dBdM}}$. Sin embargo, al contrario del caso intuicionista donde la descomposición de β en la regla multiplicativa `dB` y la exponencial `S` es suficiente para desvelar la bisimulación fuerte subyacente a la σ-equivalencia de Regnier, resulta ser que la descomposición de la regla μ en `dM` y `R` no es lo suficientemente refinada para el caso clásico. Hay numerosos ejemplos, que serán desarrollados a lo largo del capítulo, que ilustran por qué esto no resulta en una bisimulación fuerte. Uno de ellos surge al considerar la `dBdM`-forma normal de los términos del ejemplo (4.2):

$$\begin{array}{ccc}\mu\alpha'.([\alpha]\,x)[\![\alpha \setminus^{\alpha'} y]\!] & \simeq & x\,y \\ \downarrow{\scriptstyle \mathtt{R}} & & \downarrow{\scriptstyle \mathtt{R}} \\ \mu\alpha'.[\alpha']\,x\,y & \simeq & x\,y\end{array}$$

Este uso particular de `R` a la izquierda parece inocuo. De hecho, se verá que en el cálculo propuesto y su correspondiente traducción a PPNs, ambos términos $\mu\alpha'.([\alpha]\,x)[\![\alpha \setminus^{\alpha'} y]\!]$ y $\mu\alpha'.[\alpha']\,x\,y$ denotan PPNs estructuralmente equivalentes (*cf.* Sec. 4.4 para una discusión detallada). En cualquier caso, este ejemplo lleva a indagar más sobre la estructura de `R`. En particular, se verá (*cf.* Sec. 4.5) que la regla `R` debe ser descompuesta a su vez en varias nociones independientes, cada una de ellas con un rol diferente respecto a la semántica de las PPNs y, por lo tanto, a la bisimulación fuerte $\simeq$

buscada. Identificar estas nociones y su interacción para exponer la bisimulación fuerte detrás de la σ-equivalencia de Laurent es el principal desafío del presente trabajo.

4.1. Preliminares

Se introducen a continuación el $\lambda\mu$-cálculo y su relación de σ-equivalencia asociada, junto con las nociones estándar que serán referenciadas recurrentemente a lo largo del desarrollo.

4.1.1. El $\lambda\mu$-cálculo

Dado conjuntos infinito numerables de variables $\mathbb{V}$ $(x, y, \ldots)$ y nombres $\mathbb{N}$ $(\alpha, \beta, \ldots)$, los conjuntos de *objetos* $\mathbb{O}_{\lambda\mu}$, *términos* $\mathbb{T}_{\lambda\mu}$, *comandos* $\mathbb{C}_{\lambda\mu}$ y *contextos* del $\lambda\mu$-cálculo se definen mediante la siguiente gramática:

$$\begin{array}{rlll}
\textbf{Objetos} & o & ::= & t \mid c \\
\textbf{Términos} & t & ::= & x \in \mathbb{V} \mid t\,t \mid \lambda x.t \mid \mu\alpha.c \\
\textbf{Comandos} & c & ::= & [\alpha]\,t \\
\textbf{Contextos} & \mathtt{O} & ::= & \mathtt{T} \mid \mathtt{C} \\
\textbf{Contextos de término} & \mathtt{T} & ::= & \Box \mid \mathtt{T}\,t \mid t\,\mathtt{T} \mid \lambda x.\mathtt{T} \mid \mu\alpha.\mathtt{C} \\
\textbf{Contextos de comando} & \mathtt{C} & ::= & \boxdot \mid [\alpha]\,\mathtt{T}
\end{array}$$

La gramática extiende los términos de λ-cálculo con dos nuevos constructores: *comandos* $[\alpha]\,t$ y *μ-abstracciones* $\mu\alpha.c$. Respecto a los contextos, se tiene dos tipos de agujeros $\Box$ y $\boxdot$ para las clases de los *término* ($\mathtt{t}$) y *comandos* ($\mathtt{c}$) respectivamente. Usualmente se decoran los contextos o funciones sobre expresiones con su clase $\mathtt{t}$ o $\mathtt{c}$ para clarificar. Por ejemplo, $\mathtt{O}_\mathtt{t}$ es un contexto $\mathtt{O}$ con agujero de clase *término*. El sub-índice es omitido cuando no hay ambigüedad.

Los conjuntos de *variables libres* y *ligadas* de un objeto se definen como es esperado, en particular $\mathsf{fv}(\mu\alpha.c) \triangleq \mathsf{fv}(c)$ y $\mathsf{fv}([\alpha]\,t) \triangleq \mathsf{fv}(t)$. El conjunto de *nombres libres* y *ligados* de un objeto se definen como:

$$\begin{array}{rcl@{\qquad}rcl}
\mathsf{fn}(x) & \triangleq & \emptyset & \mathsf{bn}(x) & \triangleq & \emptyset \\
\mathsf{fn}(t\,u) & \triangleq & \mathsf{fn}(t) \cup \mathsf{fn}(u) & \mathsf{bn}(t\,u) & \triangleq & \mathsf{bn}(t) \cup \mathsf{bn}(u) \\
\mathsf{fn}(\lambda x.t) & \triangleq & \mathsf{fn}(t) & \mathsf{bn}(\lambda x.t) & \triangleq & \mathsf{bn}(t) \\
\mathsf{fn}(\mu\alpha.c) & \triangleq & \mathsf{fn}(c) \setminus \{\alpha\} & \mathsf{bn}(\mu\alpha.c) & \triangleq & \mathsf{bn}(c) \cup \{\alpha\} \\
\mathsf{fn}([\alpha]\,t) & \triangleq & \mathsf{fn}(t) \cup \{\alpha\} & \mathsf{bn}([\alpha]\,t) & \triangleq & \mathsf{bn}(t)
\end{array}$$

Se usan $\mathsf{fv}_x(o)$ y $\mathsf{fn}_\alpha(o)$ para denotar el número de ocurrencias libres de la variable x y del nombre α en el objeto o respectivamente. Adicionalmente, se denota con $x \notin o$ $(\alpha \notin o)$ el hecho de que $x \notin \mathsf{fv}(o) \cup \mathsf{bv}(o)$ (respectivamente $\alpha \notin \mathsf{fn}(o) \cup \mathsf{bn}(o)$). Esta noción es extendida a contextos de manera natural.

Como es estándar, se trabaja bajo la noción de α-conversión, *i.e.* módulo renombre de variables y nombre ligados. Luego, por ejemplo, $[\delta]\,(\mu\alpha.[\alpha]\,\lambda x.x)\,z =_\alpha [\delta]\,(\mu\beta.[\beta]\,\lambda y.y)\,z$. En particular, cuando se usan dos símbolos diferentes para denotar variables o nombres ligados, se asume que estos son de hecho distintos sin mencionarlo explícitamente.

La *aplicación de una sustitución* $\{x \setminus u\}$ a un objeto o, denotada $\{x \setminus u\}o$, puede requerir α-

conversión para evitar capturas indeseadas de variables/nombres libres, y se define como:

$$\begin{array}{rcll} \{x \setminus u\}x & \triangleq & u & \\ \{x \setminus u\}y & \triangleq & y & y \neq x \\ \{x \setminus u\}(t\, v) & \triangleq & \{x \setminus u\}t\, \{x \setminus u\}v & \\ \{x \setminus u\}(\lambda y.t) & \triangleq & \lambda y.\{x \setminus u\}t & \\ \{x \setminus u\}(\mu\alpha.c) & \triangleq & \mu\alpha.\{x \setminus u\}c & \\ \{x \setminus u\}([\alpha]\, t) & \triangleq & [\alpha]\, \{x \setminus u\}t & \end{array}$$

La *aplicación de un replacement* $\{\!\{\alpha \setminus^{\alpha'} u\}\!\}$ a un objeto o donde $\alpha \neq \alpha'$, denotada $\{\!\{\alpha \setminus^{\alpha'} u\}\!\}o$, pasa como argumento el término u a todo sub-comando de o de la forma $[\alpha]\, t$ y reemplaza α por α'. Esta operación también se define módulo α-conversión para evitar la captura de variables/nombres libres. Formalmente:

$$\begin{array}{rcll} \{\!\{\alpha \setminus^{\alpha'} u\}\!\}x & \triangleq & x & \\ \{\!\{\alpha \setminus^{\alpha'} u\}\!\}(t\, v) & \triangleq & \{\!\{\alpha \setminus^{\alpha'} u\}\!\}t\, \{\!\{\alpha \setminus^{\alpha'} u\}\!\}v & \\ \{\!\{\alpha \setminus^{\alpha'} u\}\!\}(\lambda x.t) & \triangleq & \lambda x.\{\!\{\alpha \setminus^{\alpha'} u\}\!\}t & x \notin u \\ \{\!\{\alpha \setminus^{\alpha'} u\}\!\}(\mu\beta.c) & \triangleq & \mu\beta.\{\!\{\alpha \setminus^{\alpha'} u\}\!\}c & \beta \notin u, \beta \neq \alpha, \beta \neq \alpha' \\ \{\!\{\alpha \setminus^{\alpha'} u\}\!\}([\alpha]\, c) & \triangleq & [\alpha']\, (\{\!\{\alpha \setminus^{\alpha'} u\}\!\}c\, u) & \\ \{\!\{\alpha \setminus^{\alpha'} u\}\!\}([\beta]\, c) & \triangleq & [\beta]\, \{\!\{\alpha \setminus^{\alpha'} u\}\!\}c & \beta \neq \alpha \end{array}$$

Por ejemplo, sea el término $I = \lambda z.z$, entonces se tienen $\{x \setminus I\}((\mu\alpha.[\alpha]\, x)\, (\lambda z.z\, x)) = (\mu\alpha.[\alpha]\, I)\, (\lambda z.z\, I)$, y $\{\!\{\alpha \setminus^{\alpha'} I\}\!\}([\alpha]\, x\, (\mu\beta.[\alpha]\, y)) = [\alpha']\, x\, (\mu\beta.[\alpha']\, y\, I)\, I$.

La *relación de reducción en un paso* $\to_{\lambda\mu}$ se define como la clausura por contextos $\mathsf{O_t}$ de las reglas de reescritura β y μ a continuación, *i.e.* $\to_{\lambda\mu} \triangleq \mathsf{O_t}\langle \mapsto_\beta \cup \mapsto_\mu \rangle$:

$$\begin{array}{rcl} (\lambda x.t)\, u & \mapsto_\beta & \{x \setminus u\}t \\ (\mu\alpha.c)\, u & \mapsto_\mu & \mu\alpha'.\{\!\{\alpha \setminus^{\alpha'} u\}\!\}c \end{array}$$

Cabe destacar que la regla μ del $\lambda\mu$-cálculo de Parigot [Par92] apela a una operación binaria de replacement $\{\!\{\alpha \setminus u\}\!\}$ que asigna $[\alpha]\, (\{\!\{\alpha \setminus u\}\!\}t)\, u$ a cada sub-expresión de la forma $[\alpha]\, t$ (y por lo tanto no modifica el nombre del comando). Notar que $\mu\alpha.\{\!\{\alpha \setminus u\}\!\}c =_\alpha \mu\alpha'.\{\!\{\alpha \setminus^{\alpha'} u\}\!\}c$; por lo tanto *e.g.* $\mu\alpha.\{\!\{\alpha \setminus u\}\!\}([\alpha]\, x) = \mu\alpha.[\alpha]\, x\, u =_\alpha \mu\gamma.[\gamma]\, x\, u = \mu\gamma.\{\!\{\alpha \setminus^{\gamma} u\}\!\}([\alpha]\, x)$. Se adopta aquí la presentación ternaria [KV19b] dado que ésta extiende naturalmente la utilizada para el ΛM-cálculo en la Sec. 4.2.

El $\lambda\mu$-cálculo permite expresar diversos operadores de control [dG94, Lau03]. Un ejemplo típico de la expresividad del $\lambda\mu$-cálculo es el operador **call-cc** [Gri90] dado por el término $\lambda x.\mu\alpha.[\alpha]\, x\, (\lambda y.\mu\delta.[\alpha]\, y)$. Esto da lugar a secuencias de reducción como la siguiente:

$$\begin{array}{ll} \underline{\textbf{call-cc}\, t\, u_0 \ldots u_n} & \to_\beta \\ \underline{(\mu\alpha.[\alpha]\, t\, (\lambda y.\mu\delta.[\alpha]\, y))\, u_0} \ldots u_n & \to_\mu \\ \underline{(\mu\alpha.[\alpha]\, t\, u_0\, (\lambda y.\mu\delta.[\alpha]\, y\, u_0))\, u_1} \ldots u_n & \twoheadrightarrow_\mu \\ \mu\alpha.[\alpha]\, t\, u_0 \ldots u_n\, (\lambda y.\mu\delta.[\alpha]\, y\, u_0 \ldots u_n) & \end{array}$$

4.1.2. Sistema de tipos simples para $\lambda\mu$-cálculo

Se introduce a continuación el sistema de tipos simples $\mathcal{M}$ para $\lambda\mu$-cálculo. El conjunto de *tipos* $\mathcal{T}$ se define mediante la siguiente gramática:

$$\textbf{Tipos simples} \quad A \quad ::= \quad \iota \mid A \to A$$

$$\frac{}{x : A \vdash x : A \mid \emptyset}\ (\textsc{var}) \qquad \frac{\Gamma \vdash t : A \to B \mid \Delta \quad \Gamma' \vdash u : A \mid \Delta'}{\Gamma \cup \Gamma' \vdash t\, u : B \mid \Delta \cup \Delta'}\ (\textsc{app})$$

$$\frac{\Gamma; (x : A)^{\leq 1} \vdash t : B \mid \Delta}{\Gamma \vdash \lambda x.t : A \to B \mid \Delta}\ (\textsc{abs}) \qquad \frac{\Gamma \vdash c \mid \Delta; (\alpha : A)^{\leq 1}}{\Gamma \vdash \mu\alpha.c : A \mid \Delta}\ (\textsc{cont})$$

$$\frac{\Gamma \vdash t : A \mid \Delta; (\alpha : A)^{\leq 1}}{\Gamma \vdash [\alpha]\, t \mid \Delta; \alpha : A}\ (\textsc{name})$$

Figura 4.5: Reglas de tipado del sistema $\mathcal{M}$ para el $\lambda\mu$-cálculo.

donde ι es un tipo base. Como es estándar, el tipo funcional $\to$ se asume asociativo a derecha. Se utilizan dos clases de *contextos de tipado* distintas: para *variables* (Γ) y para *nombres* (Δ). Las nociones de *unión compatible*, *unión disjunta* y *restricción* se extienden trivialmente a contextos de nombres (*cf.* Sec. 1.2.2).

Las reglas de tipado para el sistema $\mathcal{M}$ del $\lambda\mu$-cálculo están dadas en la Fig. 4.5. Hay dos clases de *juicios de tipado*: $\Gamma \vdash t : A \mid \Delta$ para términos y $\Gamma \vdash c \mid \Delta$. La notación $\Gamma; (x : A)^{\leq 1}$ se usa para denotar Γ o bien $\Gamma; x : A$, *i.e.* la asunción $x : A$ ocurre a lo sumo una vez en $\Gamma; (x : A)^{\leq 1}$. Análogamente para contextos de nombres. Los comandos no tiene tipo asociado, *cf.* regla (NAME).

Se denotan de forma genérica $\Gamma \vdash o : T \mid \Delta$ los dos tipos de juicios, resultando $o = t$ y $T = A$ en el caso de los términos; y $o = c$ y T ignorado para los comandos. Así mismo se usa $\pi \rhd_{\mathcal{M}} \Gamma \vdash o : T \mid \Delta$ para indicar que el juicio $\Gamma \vdash o : T \mid \Delta$ es derivable en $\mathcal{M}$ y nombrar a la derivación π. En tal caso, se dice que o es el *sujeto* de π. Por conveniencia, usualmente se abrevia con π_o a una derivación con sujeto o para contextos y tipo adecuados.

El $\lambda\mu$-cálculo simplemente tipado resulta isomorfo a la lógica clásica. Esto permite tipar, por ejemplo, al operador **call-cc**, al cual se le asigna el tipo $((A \to B) \to A) \to A$ (Peirce's Law):

$$\dfrac{\dfrac{\dfrac{}{x : (A \to B) \to A \vdash x : (A \to B) \to A \mid \emptyset}\ (\textsc{var}) \quad \dfrac{\dfrac{\dfrac{\dfrac{}{y : A \vdash y : A \mid \emptyset}\ (\textsc{var})}{y : A \vdash [\alpha]\, y \mid \alpha : A}\ (\textsc{name})}{y : A \vdash \mu\delta.[\alpha]\, y : B \mid \alpha : A}\ (\textsc{cont})}{\emptyset \vdash \lambda y.\mu\delta.[\alpha]\, y : A \to B \mid \alpha : A}\ (\textsc{abs})}{x : (A \to B) \to A \vdash x\, (\lambda y.\mu\delta.[\alpha]\, y) : A \mid \alpha : A}\ (\textsc{app})}{x : (A \to B) \to A \vdash [\alpha]\, x\, (\lambda y.\mu\delta.[\alpha]\, y) \mid \alpha : A}\ (\textsc{name})}{x : (A \to B) \to A \vdash \mu\alpha.[\alpha]\, x\, (\lambda y.\mu\delta.[\alpha]\, y) : A \mid \emptyset}\ (\textsc{cont})}{\emptyset \vdash \lambda x.\mu\alpha.[\alpha]\, x\, (\lambda y.\mu\delta.[\alpha]\, y) : ((A \to B) \to A) \to A \mid \emptyset}\ (\textsc{abs})$$

El sistema de tipos $\mathcal{M}$ posee las siguiente propiedades principales.

Lema 4.1.1 (Relevance [Par92])**.** *Sea $o \in \mathbb{O}_{\lambda\mu}$. Si $\pi \rhd_{\mathcal{M}} \Gamma \vdash o : T \mid \Delta$, entonces $\mathsf{dom}(\Gamma) = \mathsf{fv}(o)$ y $\mathsf{dom}(\Delta) = \mathsf{fn}(o)$.*

Lema 4.1.2 (Subject Reduction [Par92])**.** *Sea $o \in \mathbb{O}_{\lambda\mu}$. Si $\pi \rhd_{\mathcal{M}} \Gamma \vdash o : T \mid \Delta$ y $o \to_{\Lambda M} o'$, entonces existen $\Gamma' \subseteq \Gamma$ y $\Delta' \subseteq \Delta$ tal que $\pi' \rhd_{\mathcal{M}} \Gamma' \vdash o' : T \mid \Delta'$.*

Notar que las variables y nombres libres de un objeto pueden disminuir si el paso de reducción aplicado borra el argumento. Por ejemplo, $(\lambda x.y)\, z \to y$ o $(\mu\alpha.[\gamma]\, x)\, z \to \mu\alpha.[\gamma]\, x$.

En adelante, siempre que $o \to_{\lambda\mu} o'$ se hará referencia a π_o y $\pi_{o'}$ como dos derivaciones de tipos *relacionadas*, *i.e.* las obtenidas a partir de π_o por la prueba del Lem. 4.1.2.

4.1.3. Proof-nets polarizadas

La *lógica lineal polarizada* (PLL) [Lau02] es un sistema de pruebas basado en las fórmulas de la lógica lineal, que incorpora polaridad a las mismas:

$$\begin{array}{rrcl} \textbf{Fórmulas negativas} & N & ::= & \iota \mid N \parr N \mid ?P \\ \textbf{Fórmulas positivas} & P & ::= & \iota^\perp \mid P \otimes P \mid !N \end{array}$$

donde ι representa una fórmula atómica. PLL extiende el uso de reglas estructurales, usualmente definidas únicamente sobre ?, a todas las fórmulas negativas. Tiene asociada una noción de *proof-nets polarizadas* (PPN) que da lugar a un criterio de corrección simple.

Una característica particularmente interesante de PLL es la posibilidad de capturar la lógica clásica interpretando $A \to B$ como $!A \multimap B$, obteniendo así una traducción que resulta ser una extensión directa del mapeo de la lógica intuicionista en PLL y, por lo tanto, capturando también la interpretación del λ-cálculo en PLL. De hecho, esta traducción extiende el mapeo intuicionista a la lógica lineal estándar, *i.e.* no polarizada. En efecto, siguiendo la traducción de Girard para fórmulas clásicas pueden interpretarse los tipos simples de la Sec. 4.3 como fórmulas de PLL:

$$\begin{array}{rcl} \iota^\diamond & \triangleq & \iota \\ (A \to B)^\diamond & \triangleq & !(A^\diamond) \multimap B^\diamond \equiv ?(A^{\diamond\perp}) \parr B^\diamond \end{array}$$

La imagen de esta traducción es un sub-conjunto propio del de fórmulas polarizadas, llamado *fórmulas output*, generado por la siguiente gramática junto a su noción dual de *fórmulas anti-output*:

$$\begin{array}{rrcl} \textbf{Fórmulas output} & O & ::= & \iota \mid ?Q \parr O \\ \textbf{Fórmulas anti-output} & Q & ::= & \iota^\perp \mid !O \otimes Q \end{array}$$

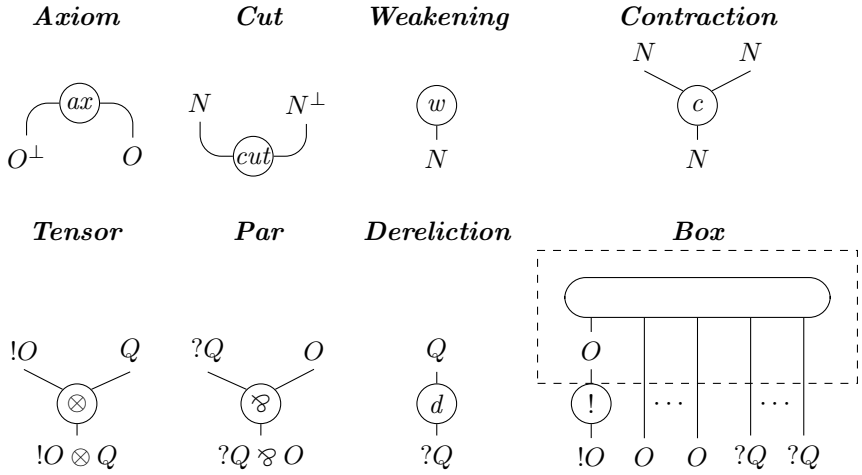
A su vez, fórmulas de la forma $?Q$ son llamadas *fórmulas input*, mientras que aquellas de la forma $!O$ se denominan *fórmulas anti-input*. Luego, la gramática completa del conjunto de *fórmulas polarizadas* [Lau03] se resume como:

$$\begin{array}{rrcl} \textbf{Fórmulas polarizadas} & F & ::= & N \mid P \\ \textbf{Fórmulas negativas} & N & ::= & O \mid ?Q \\ \textbf{Fórmulas positivas} & P & ::= & Q \mid !O \\ \textbf{Fórmulas output} & O & ::= & \iota \mid ?Q \parr O \\ \textbf{Fórmulas anti-output} & Q & ::= & \iota^\perp \mid !O \otimes Q \end{array}$$

La negación es involutiva (*i.e.* $\iota^{\perp\perp} = \iota$) con $(?Q \parr O)^\perp = !Q^\perp \otimes O^\perp$ y $(?Q)^\perp = !Q^\perp$.

Definición 4.1.3. *Una* proof-structure *es un grafo dirigido acíclico finito construido sobre el alfabeto*

de nodos (donde la dirección de las aristas es top-down):



Notar que cada eje se etiqueta con una fórmula. En particular, las conclusiones de un box tienen una única fórmula positiva $!O$, siendo las restantes fórmulas negativas. Para abreviar, se usa como meta-notación el etiquetado de un eje con un conjunto de fórmulas Γ para indicar que hay $|\Gamma|$ ejes, uno por cada fórmula en el conjunto. Esto es particularmente útil a la hora de definir la semántica de reducción de proof-nets y la interpretación de los términos en ellas.

Dada una proof-structure, su *grafo de corrección* se obtiene orientando hacia arriba las aristas positivas, hacia abajo las negativas, y eliminando las boxes (manteniendo únicamente el nodo ! correspondiente). Luego, una *proof-net polarizada* (PPN) es una proof-structure que satisface el siguiente criterio de corrección: (I) su grafo de corrección es acíclico; (II) el número de conclusiones positivas más nodos dereliction es uno; y (III) las boxes son recursivamente proof-structures correctas.

Definición 4.1.4. *La* equivalencia estructural $\equiv$ *de proof-nets polarizadas resulta de clausurar por contextos las identidades de la Fig. 4.6, donde en las últimas dos reglas se asume que el nodo weakening es* final *(*i.e. *no es atrapado por ningún contexto de clausura).*

La primera ecuación define la asociatividad de los nodos de contracción, la segunda axiomatiza la permeabilidad de la contracción respecto a las boxes, la tercera especifica la neutralidad del weakening respecto a la contracción, la cuarta permite extraer nodos weakening finales al top-level, mientras que la quinta permite removerlos. En adelante, se asume igualdad de proof-nets módulo equivalencia estructural, al igual que en [Lau03].

Por su parte, la *relación de reducción* $\to_{\mathtt{PPN}}$ de PPNs está dada por el conjunto de reglas de *cut-elimination* presentado en las Fig. 4.7, 4.8 y 4.9. Las primeras dos figuras contienen las reglas de cut-elimination estándar para MELL [Gir87], separadas en multiplicativas y exponenciales respectivamente, mientras que la tercera contiene las reglas de cut específicas de PPNs [Lau03] que también resultan exponenciales. Las reglas de la Fig. 4.9 contemplan los cuts entre un nodo weakening/contraction/box y un $\otimes$-*tree*: *i.e.* una estructura recursiva cuyo caso base es un axioma, y sus nodos internos son tensores con premisas box a izquierda y $\otimes$-tree a derecha. Los $\otimes$-trees serán utilizados a continuación para interpretar stacks (*cf.* Sec. 4.4). Notar que las reglas $\mathtt{C}(\otimes, w)$, $\mathtt{C}(\otimes, c)$ y $\mathtt{C}(\otimes, !)$ son análogas a $\mathtt{C}(!, w)$, $\mathtt{C}(!, c)$ y $\mathtt{C}(!, !)$ respectivamente, intercambiando tensor por box.

La relación de reducción $\to_{\mathtt{PPN}}$ es confluente (CR) [Lau03, Cor. 14] y fuertemente normalizante (SN) [Lau03, Cor. 15]. Más aún, la relación resultante de considerar el sub-conjunto de reglas

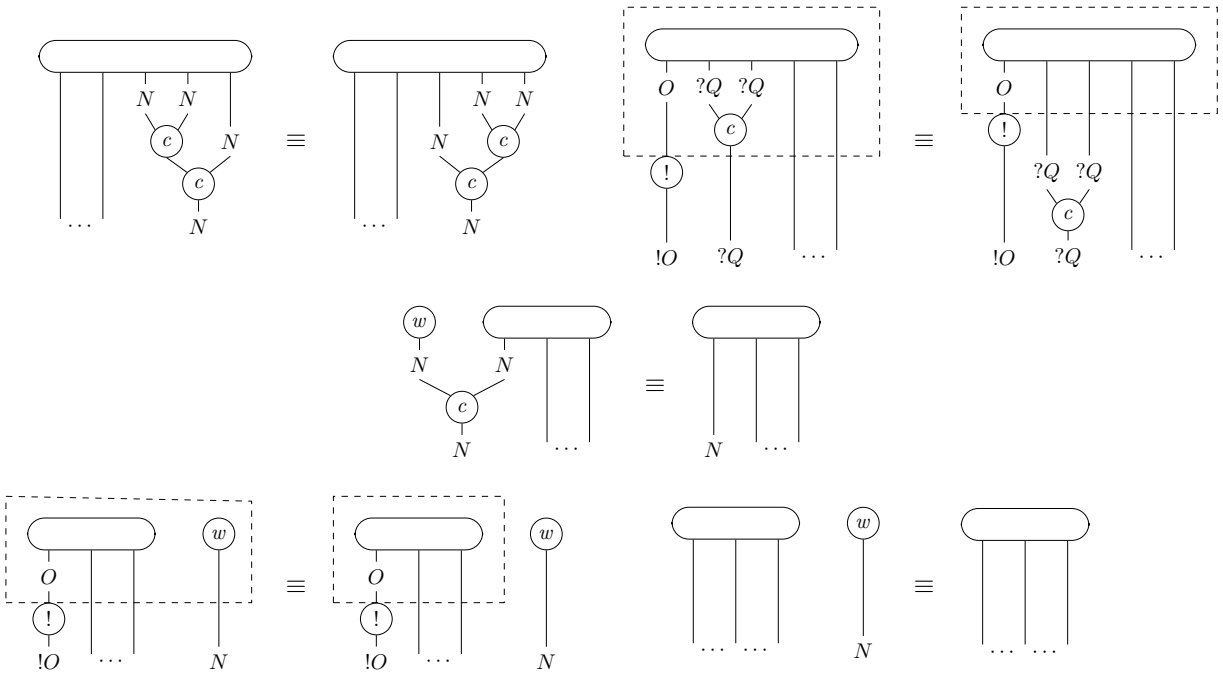


Figura 4.6: Equivalencia estructural de proof-nets polarizadas.

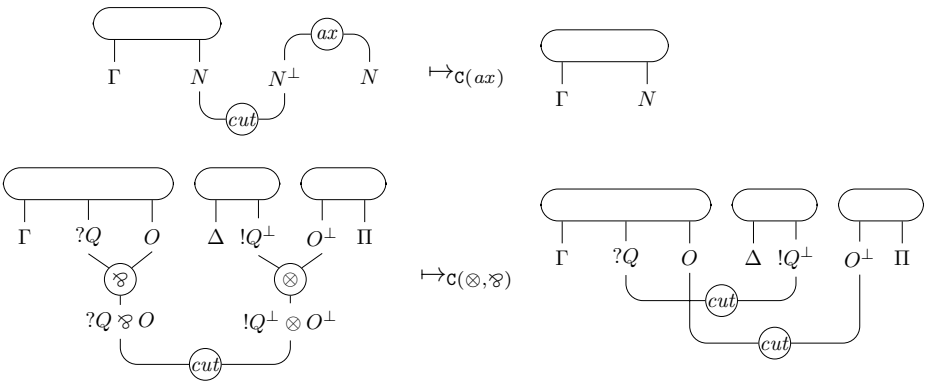


Figura 4.7: Cut-elimination para PPNs: cuts multiplicativos (1/3)

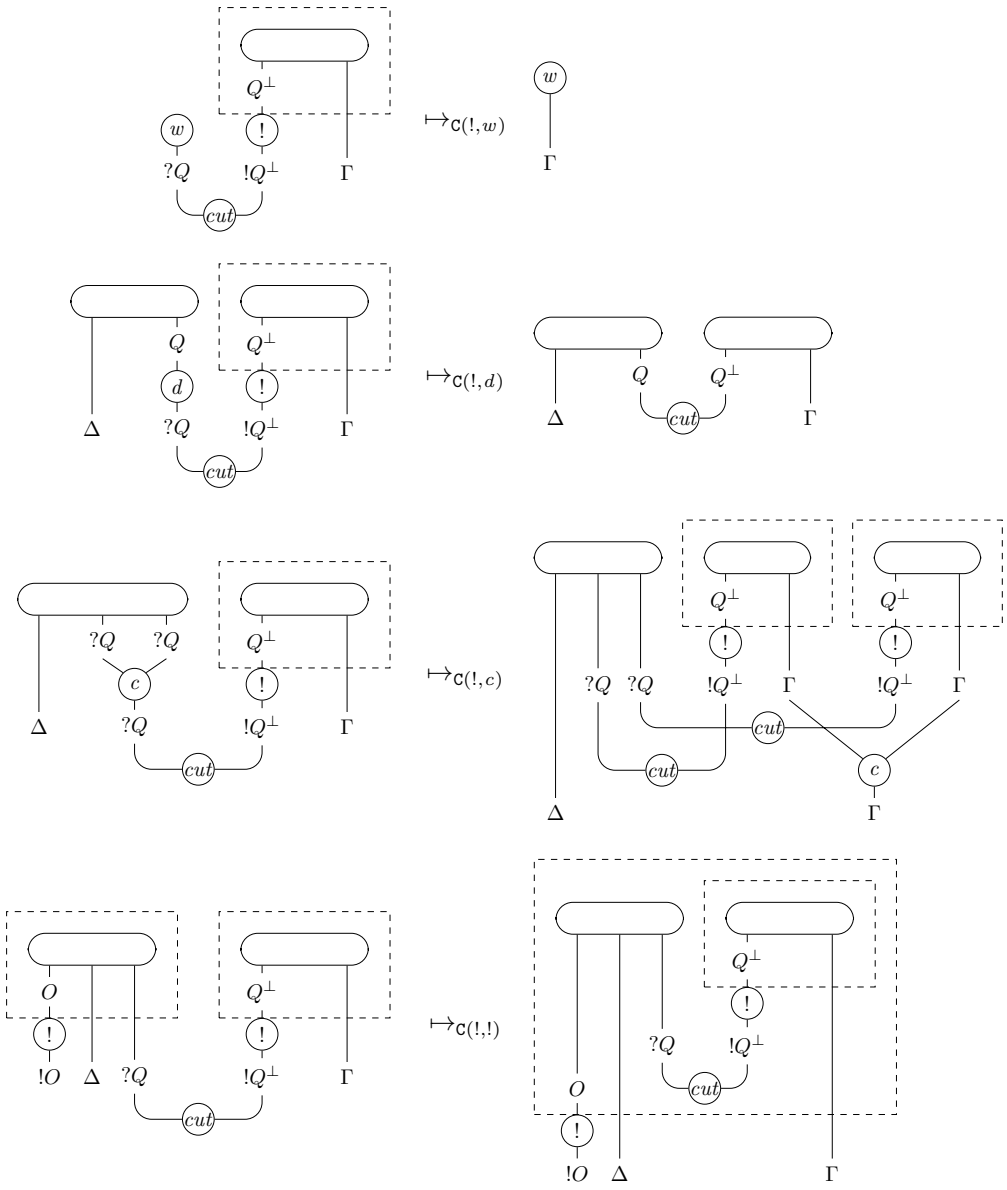


Figura 4.8: Cut-elimination para PPNs: cuts exponenciales (2/3)

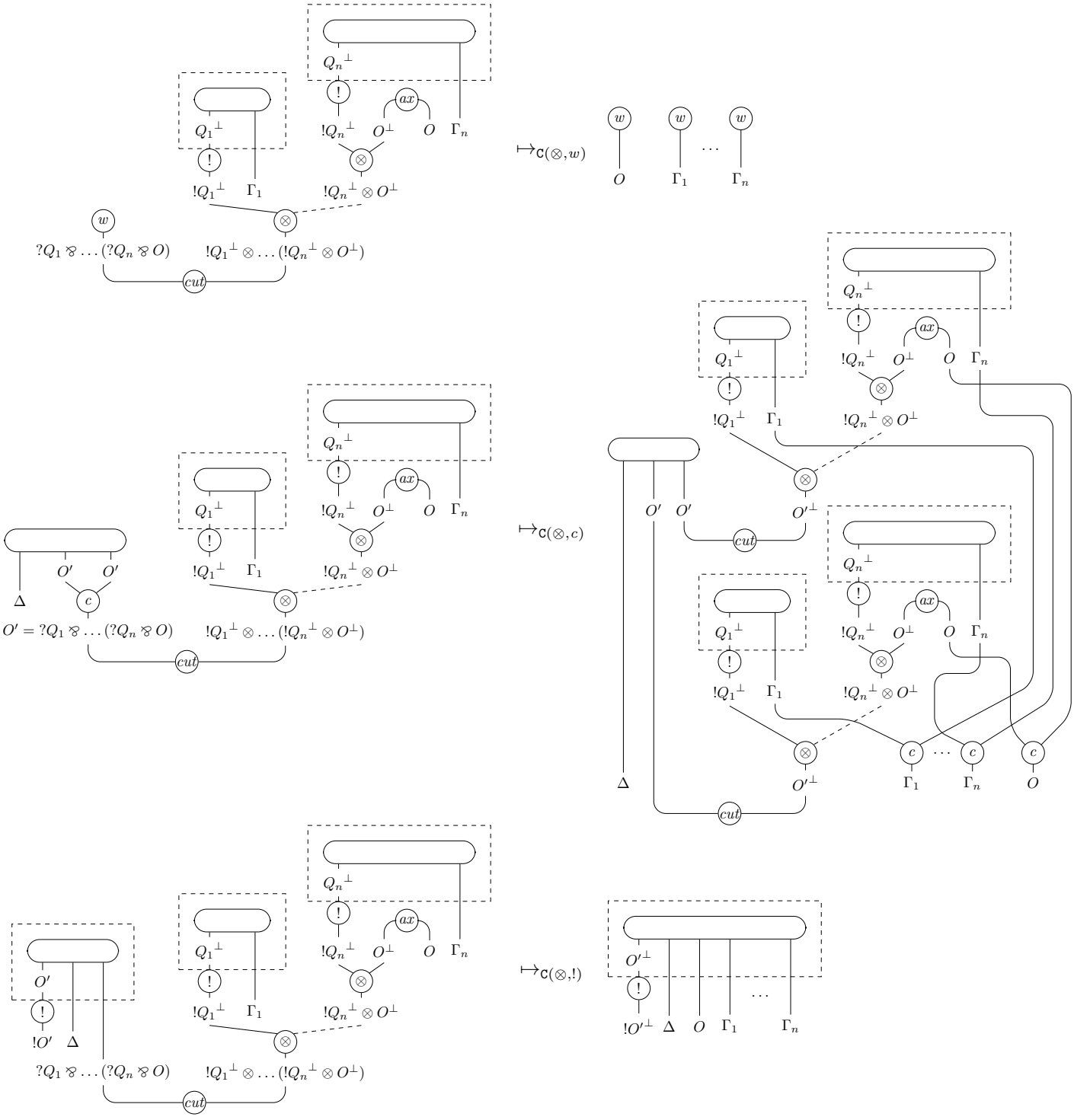


Figura 4.9: Cut-elimination para PPNs: cuts exponenciales (3/3)

$\{\mathtt{C}(ax), \mathtt{C}(\otimes, \parr)\}$ también es CR y SN, y sus formas normales se denominan *formas normales multiplicativas*. Esto constituye una técnica utilizada a continuación para definir la traducción de objetos tipados del ΛM-cálculo a PPNs.

Traducción a proof-nets

Las proof-nets polarizadas son usadas en [Lau02, Lau03] para interpretar los objectos del $\lambda\mu$-cálculo y así motivar las reglas de la σ-equivalencia (*cf.* Sec. 4.1.4) identificando aquellos programas cuyas interpretaciones como proof-nets son estructuralmente equivalentes módulo cuts multiplicativos.

La traducción se define sobre términos simplemente tipados puesto que estos son, vía el isomorfismo de Curry–Howard, testigos de las demostraciones de los teoremas de la lógica clásica y están, por lo tanto, estrechamente relacionados con las proof-nets polarizadas.

En primer lugar se define la traducción del conjunto de tipos $\mathcal{T}$ a fórmulas polarizadas, más precisamente en el sub-conjunto de fórmulas output.

Definición 4.1.5. *La* traducción de un tipo simple A *en una fórmula output polarizada, notada* $A^\diamond$*, se define inductivamente como:*

$$\begin{array}{rcl} \iota^\diamond & \triangleq & \iota \\ (A \to B)^\diamond & \triangleq & ?({A^\diamond}^\perp) \parr B^\diamond \end{array}$$

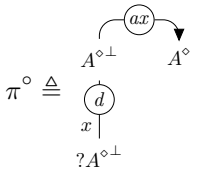
Esta definición también se extiende a contextos simplemente ignorando los sujetos de las asunciones y traduciendo uno a uno sus tipos asociados: *i.e.* dados los contextos $\Gamma = \{x_1 : A_1, \ldots, x_n : A_n\}$ y $\Delta = \{\alpha_1 : B_1, \ldots, \alpha_m : B_m\}$, entonces $\Gamma^\diamond = \{{A_1}^\diamond, \ldots, {A_n}^\diamond\}$ y $\Delta^\diamond = \{{B_1}^\diamond, \ldots, {B_m}^\diamond\}$. Además, se abrevia $?(\Gamma^{\diamond\perp})$ el conjunto $\{?({A_1}^{\diamond\perp}), \ldots, ?({A_n}^{\diamond\perp})\}$.

La traducción de objetos tipados del $\lambda\mu$-cálculo a proof-nets polarizadas se define inductivamente sobre derivaciones de tipos, considerando las reglas de tipado de la Fig. 4.5. Dado $\pi \rhd_{\mathcal{M}} \Gamma \vdash o : T \mid \Delta$, la proof-net polarizada resultante tendrá una conclusión para cada fórmula en $?(\Gamma^{\diamond\perp})$ y $\Delta^\diamond$, cada una de ellas etiquetada con la correspondiente variable/nombre en Γ/Δ. Se denota entre paréntesis una variable/nombre borrada/o del paso inductivo durante la traducción. Además, la proof-net resultante de traducir un término tiene una conclusión *distinguida* para T, indicada con $\rightarrow\!\!\blacktriangleright$, que resulta en una fórmula output. La conclusión distinguida del paso inductivo (de existir) se denota con $\rightarrow\!\!\triangleright$. En resumen:

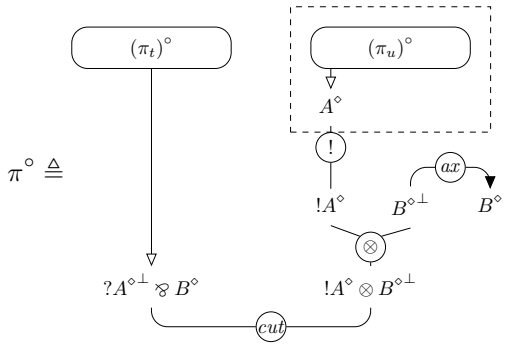
Nota 4.1.6. *Para las derivaciones de tipo basadas en dos (o más) premisas la traducción agrega nodos de contracción en las variables y nombres compartidos, pero se omite escribir estos nodos explícitamente para facilitar la lectura. Más aún, solo se muestran los ejes que participan activamente de la construcción, omitiendo así las conclusiones que no son relevantes para la traducción en sí misma.*

Definición 4.1.7. *La* traducción de una derivación de tipo $\pi \rhd_{\mathcal{M}} \Gamma \vdash o : T \mid \Delta$ *(para un término o comando) en una proof-net polarizada, notada* π°*, se define inductivamente como:*

- (VAR)*:* i.e. $\pi \rhd_{\mathcal{M}} x : A \vdash x : A \mid \emptyset$.



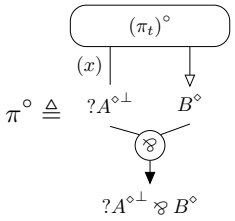
- (APP): i.e. $\pi \triangleright_{\mathcal{M}} \Gamma \cup \Gamma' \vdash t\, u : B \mid \Delta \cup \Delta'$ *con premisas* $\pi_t \triangleright_{\mathcal{M}} \Gamma \vdash t : A \to B \mid \Delta$ *y* $\pi_u \triangleright_{\mathcal{M}} \Gamma' \vdash u : A \mid \Delta'$.



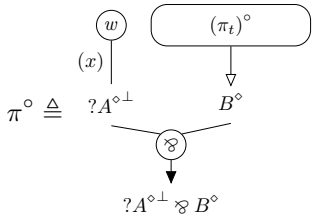
Se agregan nodos de contracción para las conclusiones correspondientes a variables/nombres comunes entre t *y* u.

- (ABS): i.e. $\pi \triangleright_{\mathcal{M}} \Gamma \vdash \lambda x.t : A \to B \mid \Delta$ *con premisa* π_t*:*

  - *si* $x \in \mathsf{fv}(t)$*, entonces* $\pi_t \triangleright_{\mathcal{M}} \Gamma; x : A \vdash t : B \mid \Delta$.

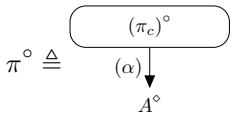


  - *si* $x \notin \mathsf{fv}(t)$*, entonces* $\pi_t \triangleright_{\mathcal{M}} \Gamma \vdash t : B \mid \Delta$ *con* $x \notin \mathsf{dom}(\Gamma)$.

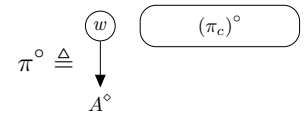


- (CONT): i.e. $\pi \triangleright_{\mathcal{M}} \Gamma \vdash \mu\alpha.c : A \mid \Delta$ *con premisa* π_c*:*

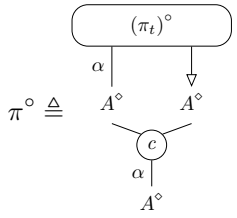
  - *si* $\alpha \in \mathsf{fn}(c)$*, entonces* $\pi_c \triangleright_{\mathcal{M}} \Gamma \vdash c \mid \Delta; \alpha : A$.



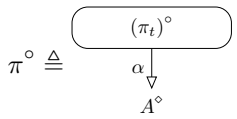
  - *si* $\alpha \notin \mathsf{fn}(c)$*, entonces* $\pi_t \triangleright_{\mathcal{M}} \Gamma \vdash c \mid \Delta$ *con* $x \notin \mathsf{dom}(\Delta)$.



- (NAME): i.e. $\pi \rhd_{\mathcal{M}} \Gamma \vdash [\alpha]\, t \mid \Delta; \alpha : A$ *con premisa* π_t*:*
  - *si* $\alpha \in \mathsf{fn}(t)$*, entonces* $\pi_t \rhd_{\mathcal{M}} \Gamma \vdash t : A \mid \Delta; \alpha : A$.



  - *si* $\alpha \notin \mathsf{fn}(t)$*, entonces* $\pi_t \rhd_{\mathcal{M}} \Gamma \vdash t : A \mid \Delta$ *con* $x \notin \mathsf{dom}(\Delta)$.



Notar que la única regla de tipado que introduce cuts al ser traducida es (APP), y resulta ser un cut multiplicativo $\mathtt{C}(\otimes, \parr)$.

Esta traducción es capaz de simular la reducción del $\lambda\mu$-cálculo por medio de cut-elimination.

Teorema 4.1.8 (Simulación [Lau03])**.** *Sean* $o, p \in \mathbb{O}_{\lambda\mu}$ *tipados. Si* $o \to_{\lambda\mu} p$ *y* π_o, π_p *son las derivaciones de tipos relacionadas correspondientes, entonces* $(\pi_o)^\circ \twoheadrightarrow_{\mathtt{PPN}} (\pi_p)^\circ$.

Adicionalmente se define una segunda traducción para juicios de tipado, notada $\_^\bullet$, como la forma normal multiplicativa de $\_^\circ$. Recordar que la reducción para PPNs es CR y SN (*cf.* Sec. 4.1.3). Esta traducción permite probar lo anteriormente mencionado: objectos σ-equivalentes son interpretados como proof-nets estructuralmente equivalentes módulo cuts multiplicativos (*cf.* Sec. 4.1.4).

4.1.4. Sigma equivalencia para $\lambda\mu$-cálculo

Al igual que para el λ-cálculo, la equivalencia estructural en $\lambda\mu$ captura permutaciones inocuas de redexes, pero en este caso involucrando operadores de control. La noción de σ-equivalencia para términos del $\lambda\mu$-cálculo introducida por Laurent [Lau03] es presentada en la Fig. 4.10. Notar que las primeras dos ecuaciones son aquellas de la σ-equivalencia de Regnier para λ-cálculo, por lo cual $\simeq_\sigma$ del $\lambda\mu$-cálculo extiende estrictamente a $\simeq_\sigma$ del λ-cálculo. Las restantes ecuaciones involucran la interacción entre los operadores de control y los demás constructores del cálculo. Resulta de particular interés destacar la regla $\simeq_{\sigma_7}$ en la cual aparece una nueva operación de *renaming*, *i.e.* la sustitución del nombre libre α por β en el comando c, la cual no interviene en la semántica operacional del $\lambda\mu$-cálculo pero resulta necesaria para la correcta formalización de la equivalencia estructural.

Laurent muestra propiedades para $\simeq_\sigma$ similares a aquellas dadas por Regnier. Más precisamente, $t \simeq_\sigma s$ implica que t normaliza si y solo si s normaliza (tanto para las noción débil de normalización como para la head y la fuerte) [Lau03, Prop. 3]. También muestra la preservación de los tipos y la semántica de proof-nets polarizadas basándose en la codificación de la lógica clásica en la lógica lineal dada por Girard [Gir87].

Lema 4.1.9 ([Lau03])**.** *Sea* $o \in \mathbb{O}_{\lambda\mu}$*. Si* $\pi \rhd_{\mathcal{M}} \Gamma \vdash o : T \mid \Delta$ *y* $o \simeq_\sigma o'$*, entonces existe* $\pi' \rhd_{\mathcal{M}} \Gamma \vdash o' : T \mid \Delta'$.

$$
\begin{array}{rcll}
(\lambda x.\lambda y.t)\,u & \simeq_{\sigma_1} & \lambda y.(\lambda x.t)\,u & y \notin u \\
(\lambda x.t\,v)\,u & \simeq_{\sigma_2} & (\lambda x.t)\,u\,v & x \notin v \\
(\lambda x.\mu\beta.[\alpha]\,t)\,u & \simeq_{\sigma_3} & \mu\beta.[\alpha]\,(\lambda x.t)\,u & \beta \notin u \\
[\alpha']\,(\mu\alpha.[\beta']\,(\mu\beta.c)\,v)\,u & \simeq_{\sigma_4} & [\beta']\,(\mu\beta.[\alpha']\,(\mu\alpha.c)\,u)\,v & \alpha \notin v, \beta \notin u, \beta \neq \alpha', \alpha \neq \beta' \\
[\alpha']\,(\mu\alpha.[\beta']\,\lambda y.\mu\beta.c)\,u & \simeq_{\sigma_5} & [\beta']\,\lambda y.\mu\beta.[\alpha']\,(\mu\alpha.c)\,u & y \notin u, \beta \notin u, \beta \neq \alpha', \alpha \neq \beta' \\
[\alpha']\,\lambda x.\mu\alpha.[\beta']\,\lambda y.\mu\beta.c & \simeq_{\sigma_6} & [\beta']\,\lambda y.\mu\beta.[\alpha']\,\lambda x.\mu\alpha.c & \beta \neq \alpha', \alpha \neq \beta' \\
[\beta]\,\mu\alpha.c & \simeq_{\sigma_7} & \{\alpha \setminus \beta\}c & \\
\mu\alpha.[\alpha]\,t & \simeq_{\sigma_8} & t & \alpha \notin t
\end{array}
$$

Figura 4.10: σ-equivalencia para términos del $\lambda\mu$-cálculo.

Al igual que para subject reduction (*cf.* Lem. 4.1.2), siempre que $o \simeq_\sigma o'$ (u $o \to_{\lambda\mu} o'$), se hace referencia a π_o y $\pi_{o'}$ como dos derivaciones de tipos *relacionadas*, en este caso por equivalencia estructural, obtenidas a partir de π_o por la prueba del Lem. 4.1.9. Luego, la preservación de la semántica de proof-nets se enucia de la siguiente manera.

Teorema 4.1.10 ([Lau03])**.** *Sean $o, p \in \mathbb{O}_{\lambda\mu}$ tipados. Luego, $o \simeq_\sigma p$ sii $(\pi_o)^\bullet \equiv (\pi_p)^\bullet$, donde π_o y π_p son las correspondientes derivaciones de tipo relacionadas.*

Estos resultados distan de ser triviales dado que los objetos a cada lado de una misma ecuación en la Fig. 4.10 no necesariamente tiene los mismos β/μ-redexes. Por ejemplo, los términos $(\mu\alpha.[\alpha]\,x)\,y$ y $x\,y$ están relacionados por la ecuación $\simeq_{\sigma_8}$ pero sin embargo el primero contiene un μ-redex (*lineal*) mientras que el segundo no tiene ningún redex. En efecto, como se menciona anteriormente en la Sec. 4 (*cf.* ejemplo (4.2)), $\simeq_\sigma$ no es una bisimulación fuerte con respecto a la reducción en el $\lambda\mu$-cálculo:

$$
\begin{array}{ccc}
(\mu\alpha.[\alpha]\,x)\,y & \simeq_{\sigma_8} & x\,y \\
{\scriptstyle\mu}\downarrow & & \downarrow{\scriptstyle\mu} \\
\mu\alpha.[\alpha]\,x\,y & \simeq_{\sigma_8} & x\,y
\end{array}
$$

Existen numerosos ejemplos que ilustran la insatisfacibilidad de la bisimulación fuerte (*cf.* Sec. 4.6). Resulta de particular interés el siguiente, generado a partir de la regla $\simeq_{\sigma_7}$:

$$
\begin{array}{ccc}
(\mu\alpha.[\alpha]\,\mu\beta.c)\,u & \simeq_{\sigma_7} & (\mu\alpha.\{\beta \setminus \alpha\}c)\,u \\
{\scriptstyle\mu}\downarrow & & {\scriptstyle\mu}\downarrow \\
\mu\alpha'.[\alpha']\,(\mu\beta.\{\!\{\alpha \setminus^{\alpha'} u\}\!\}c)\,u & \not\simeq_\sigma & \mu\alpha'.\{\!\{\alpha \setminus^{\alpha'} u\}\!\}\{\beta \setminus \alpha\}c
\end{array}
\tag{4.3}
$$

donde es necesario ejecutar un segundo paso de μ-reducción del lado izquierdo para obtener términos σ-equivalentes. En este caso la operación de renaming interviene, eliminando todo rastro de la continuación β al lado derecho de $\simeq_{\sigma_7}$. Sin embargo, el renaming no participa de la semántica operacional del $\lambda\mu$-cálculo (*i.e.* no es introducido por las reglas β y μ), lo que lleva a preguntarse hasta qué punto es factible, tal como en el caso intuicionista, formalizar una noción más refinada de reducción para el $\lambda\mu$-cálculo que permita salvar esta situación.

4.2. El ΛM-cálculo

Como primer paso para refinar el planteo en busca de nociones adecuadas de σ-equivalencia y reducción meaningful para el λ-cálculo con operadores de control, se extiende la sintaxis del $\lambda\mu$-cálculo con operadores explícitos de sustitución, replacement y renaming.

Dado conjuntos infinito numerables de variables $\mathbb{V}$ $(x, y, \ldots)$ y nombres $\mathbb{N}$ $(\alpha, \beta, \ldots)$, los conjuntos de *objetos* $\mathbb{O}_{\Lambda M}$, *términos* $\mathbb{T}_{\Lambda M}$, *comandos* $\mathbb{C}_{\Lambda M}$, *stacks* y *contextos* del ΛM-cálculo se definen mediante la siguiente gramática:

$$
\begin{array}{rrcl}
\textbf{Objetos} & o & ::= & t \mid c \mid s \\
\textbf{Términos} & t & ::= & x \in \mathbb{V} \mid t\,t \mid \lambda x.t \mid \mu\alpha.c \mid t[x \setminus t] \\
\textbf{Comandos} & c & ::= & [\alpha]\,t \mid c[\![\alpha \setminus^{\alpha'} s]\!] \mid c[\alpha \setminus \beta] \\
\textbf{Stacks} & s & ::= & t \mid t \cdot s \\
\textbf{Contextos} & \mathtt{O} & ::= & \mathtt{T} \mid \mathtt{C} \mid \mathtt{S} \\
\textbf{Contextos de término} & \mathtt{T} & ::= & \Box \mid \mathtt{T}\,t \mid t\,\mathtt{T} \mid \lambda x.\mathtt{T} \mid \mu\alpha.\mathtt{C} \mid \mathtt{T}[x \setminus t] \mid t[x \setminus \mathtt{T}] \\
\textbf{Contextos de comando} & \mathtt{C} & ::= & \boxdot \mid [\alpha]\,\mathtt{T} \mid \mathtt{C}[\![\alpha \setminus^{\alpha'} s]\!] \mid c[\![\alpha \setminus^{\alpha'} \mathtt{S}]\!] \mid \mathtt{C}[\alpha \setminus \beta] \\
\textbf{Contextos de stack} & \mathtt{S} & ::= & \mathtt{T} \mid \mathtt{T} \cdot s \mid t \cdot \mathtt{S} \\
\textbf{Contextos de sustitución} & \mathtt{L} & ::= & \Box \mid \mathtt{L}[x \setminus t] \\
\textbf{Contextos de repl./ren.} & \mathtt{R} & ::= & \boxdot \mid \mathtt{R}[\![\alpha \setminus^{\alpha'} s]\!] \mid \mathtt{R}[\alpha \setminus \beta]
\end{array}
$$

Los términos del $\lambda\mu$-cálculo son enriquecidos con *sustituciones explícitas* de la forma $[x \setminus u]$, mientras que a los comandos se les agregan los operadores de *replacement explícito* $[\![\alpha \setminus^{\alpha'} s]\!]$ (donde $\alpha \neq \alpha'$ y α' es llamado el *nombre del replacement*) y *renaming explícito* $[\alpha \setminus \beta]$ (donde α es el nombre reemplazado y β el introducido en su lugar). Los stacks son esencialmente pilas no vacías de términos que solo pueden aparecer en el contexto de un replacement explícito.

La concatenación de stacks se denota $s \cdot s'$, sobrecargando así el constructor de *push*, y se define de la manera esperada: si $s = t_0 \cdot \ldots \cdot t_n$, entonces $s \cdot s' \triangleq t_0 \cdot \ldots \cdot t_n \cdot s'$; resultando la operación asociativa. Dado un término u, se denota con $u :: s$ al término que resulta de aplicar u a todos los elementos del stack s: si $s = t_0 \cdot \ldots \cdot t_n$, entonces $u :: s \triangleq u\,t_0 \ldots t_n$. Recordar que la aplicación es asociativa a izquierda por convención, por lo que esta operación también lo es; es decir $u :: s :: s'$ se interpreta $(u :: s) :: s'$.

Los conjuntos de *variables libres* y *ligadas* de un objeto del ΛM-cálculo se definen de la manera esperada, siendo la sustitución explícita un ligador: *i.e.* $\mathsf{fv}(t[x \setminus u]) \triangleq (\mathsf{fv}(t) \setminus \{x\}) \cup \mathsf{fv}(u)$ y $\mathsf{bv}(t) = \mathsf{bv}(t) \cup \mathsf{bv}(u) \cup \{x\}$. En cuanto a los conjuntos de *nombres libres* y *ligados*, cabe destacar que *no* se liga el nombre del replacement en un replacement explícito:

$$
\begin{array}{rcl@{\qquad}rcl}
\mathsf{fn}(c[\![\alpha \setminus^{\alpha'} s]\!]) & \triangleq & (\mathsf{fn}(c) \setminus \{\alpha\}) \cup \mathsf{fn}(s) \cup \{\alpha'\} & \mathsf{bn}(c[\![\alpha \setminus^{\alpha'} s]\!]) & \triangleq & \mathsf{bn}(c) \cup \mathsf{bn}(s) \cup \{\alpha\} \\
\mathsf{fn}(c[\alpha \setminus \beta]) & \triangleq & (\mathsf{fn}(c) \setminus \{\alpha\}) \cup \{\beta\} & \mathsf{bn}(c[\alpha \setminus \beta]) & \triangleq & \mathsf{bn}(c) \cup \{\alpha\}
\end{array}
$$

Como es usual, se trabaja módulo α-conversión, de modo que variables y nombres ligados pueden ser renombrados libremente. Luego, *e.g.* $x[x \setminus u] =_\alpha y[y \setminus u]$, $([\gamma]\,x)[\![\gamma \setminus^{\alpha} u]\!] =_\alpha ([\beta]\,x)[\![\beta \setminus^{\alpha} u]\!]$ y $([\gamma]\,x)[\gamma \setminus \alpha] =_\alpha ([\beta]\,x)[\beta \setminus \alpha]$. En particular, se asume por α-conversión, que $x \notin \mathsf{fv}(u)$ en $t[x \setminus u]$, $\alpha \notin \mathsf{fn}(s)$ en $c[\![\alpha \setminus^{\alpha'} s]\!]$, y $\alpha \neq \beta$ en $c[\alpha \setminus \beta]$.

Las nociones de variables y nombres libres/ligados se extienden a contextos considerando los casos de $\Box$ y $\boxdot$ definidos como el conjunto vacío. Luego, por ejemplo, x está ligada en $\lambda x.\Box$ y $(\lambda x.x)\,\Box$, y α está ligado en $\boxdot[\![\alpha \setminus^{\alpha'} s]\!]$. Un objeto o de dice *libre para un contexto* $\mathtt{O}$, notado $\mathsf{fc}(o, \mathtt{O})$, si las variables y nombres ligados de $\mathtt{O}$ no ocurren libres en o, *i.e.* $(\mathsf{bv}(\mathtt{O}) \cup \mathsf{bn}(\mathtt{O})) \cap (\mathsf{fv}(o) \cup \mathsf{fn}(o)) = \emptyset$. Por ejemplo, $\mathsf{fc}(z\,y, \lambda x.\Box[x' \setminus w])$ se satisface mientras que $\mathsf{fc}(x\,y, \lambda x.\Box)$ no. Esta noción es extendida de manera natural a conjuntos de objetos, *i.e.* $\mathsf{fc}(\mathcal{S}, \mathtt{O})$ significa que las variables y nombres ligados de $\mathtt{O}$ no ocurren libres en ningún elemento de $\mathcal{S}$.

4.2.1. Semántica operacional

La *aplicación de una sustitución* $\{x \setminus u\}$ a un objeto o de ΛM se extiende a objetos del $\lambda\mu$-cálculo de manera natural, quedando definida módulo α-conversión para evitar capturas de variables/nombres

libres, al igual que en la Sec. 4.1.1. En cuanto a las operaciones de aplicación de un replacement o un renaming, las mismas se formalizan a continuación. Para el caso del replacement puede asumirse, por α-conversión, $\alpha \notin \mathsf{fn}(s)$ y $\alpha \neq \alpha'$:

$$
\begin{array}{rcll}
\{\!|\alpha \setminus^{\alpha'} s|\!\}x & \triangleq & x & \\
\{\!|\alpha \setminus^{\alpha'} s|\!\}(t\,u) & \triangleq & \{\!|\alpha \setminus^{\alpha'} s|\!\}t\,\{\!|\alpha \setminus^{\alpha'} s|\!\}u & \\
\{\!|\alpha \setminus^{\alpha'} s|\!\}(\lambda x.t) & \triangleq & \lambda x.\{\!|\alpha \setminus^{\alpha'} s|\!\}t & x \notin s \\
\{\!|\alpha \setminus^{\alpha'} s|\!\}(\mu\beta.c) & \triangleq & \mu\beta.\{\!|\alpha \setminus^{\alpha'} s|\!\}c & \beta \notin s, \beta \neq \alpha, \beta \neq \alpha' \\
\{\!|\alpha \setminus^{\alpha'} s|\!\}(t[x \setminus u]) & \triangleq & (\{\!|\alpha \setminus^{\alpha'} s|\!\}t)[x \setminus \{\!|\alpha \setminus^{\alpha'} s|\!\}u] & x \notin s \\
\{\!|\alpha \setminus^{\alpha'} s|\!\}([\alpha]\,t) & \triangleq & [\alpha']\,(\{\!|\alpha \setminus^{\alpha'} s|\!\}t) :: s & \\
\{\!|\alpha \setminus^{\alpha'} s|\!\}([\beta]\,t) & \triangleq & [\beta]\,\{\!|\alpha \setminus^{\alpha'} s|\!\}t & \alpha \neq \beta \\
\{\!|\alpha \setminus^{\alpha'} s|\!\}(c[\![\gamma \setminus^{\alpha} s']\!]) & \triangleq & (\{\!|\alpha \setminus^{\alpha'} s|\!\}c)[\![\gamma \setminus^{\alpha'} \{\!|\alpha \setminus^{\alpha'} s|\!\}s' \cdot s]\!] & \gamma \notin s, \gamma \neq \alpha, \gamma \neq \alpha' \\
\{\!|\alpha \setminus^{\alpha'} s|\!\}(c[\![\gamma \setminus^{\beta} s']\!]) & \triangleq & (\{\!|\alpha \setminus^{\alpha'} s|\!\}c)[\![\gamma \setminus^{\beta} \{\!|\alpha \setminus^{\alpha'} s|\!\}s']\!] & \alpha \neq \beta, \gamma \notin s, \gamma \neq \alpha, \gamma \neq \alpha' \\
\{\!|\alpha \setminus^{\alpha'} s|\!\}(c[\gamma \setminus \alpha]) & \triangleq & (\{\!|\alpha \setminus^{\alpha'} s|\!\}c)[\![\gamma \setminus^{\beta} s]\!][\beta \setminus \alpha'] & \beta \text{ fresca}, \gamma \notin s, \gamma \neq \alpha, \gamma \neq \alpha' \\
\{\!|\alpha \setminus^{\alpha'} s|\!\}(c[\gamma \setminus \beta]) & \triangleq & (\{\!|\alpha \setminus^{\alpha'} s|\!\}c)[\gamma \setminus \beta] & \alpha \neq \beta, \gamma \notin s, \gamma \neq \alpha, \gamma \neq \alpha' \\
\{\!|\alpha \setminus^{\alpha'} s|\!\}(t \cdot s') & \triangleq & \{\!|\alpha \setminus^{\alpha'} s|\!\}t \cdot \{\!|\alpha \setminus^{\alpha'} s|\!\}s' &
\end{array}
$$

La mayoría de los casos de la definición son directos. Se comentan a continuación los más interesantes. Cuando un (meta-)replacement afecta a un operador de replacement explícito, *i.e.* el caso $\{\!|\alpha \setminus^{\alpha'} s|\!\}(c[\![\gamma \setminus^{\alpha} s']\!])$, este último se encuentra *bloqueando* la operación de replacement sobre γ. Es decir, γ y α denotan el mismo comando, pero los argumentos de α no deben ser pasados a γ aún. Es por esto que se acumulan en el replacement explícito el parámetro adicional s de la meta-operación. De ahí la necesidad de utilizar *stacks*. Por su parte, cuando un (meta-)replacement afecta a un operador de renaming explícito, *i.e.* el caso $\{\!|\alpha \setminus^{\alpha'} s|\!\}(c[\gamma \setminus \alpha])$, el renaming $[\gamma \setminus \alpha]$ está bloqueando el replacement de modo similar al caso anterior. En esta situación, se *memoriza* la necesidad de pasar los argumentos s al comando γ mediante la creación de un nuevo replacement explícito $[\![\gamma \setminus^{\beta} s]\!]$ sobre un nombre fresco β que luego es, a su vez, renombrado explícitamente a α' (*i.e.* el nombre del replacement sobre α). Vale destacar que se preserva el renaming explícito en pos de lograr el resultado de bisimulación fuerte buscado (*cf.* Sec. 4.6). Por ejemplo, valen $\{\!|\alpha \setminus^{\gamma} y_0 \cdot y_1|\!\}([\alpha]\,x) = [\gamma]\,x\,y_0\,y_1$, y $\{\!|\alpha \setminus^{\gamma} y_0|\!\}(([\alpha]\,x)[\![\beta \setminus^{\alpha} z_0]\!]) = ([\gamma]\,x\,y_0)[\![\beta \setminus^{\gamma} z_0 \cdot y_0]\!]$, mientras que se tiene $\{\!|\alpha \setminus^{\gamma} y_0 \cdot y_1|\!\}(([\alpha]\,x)[\beta \setminus \alpha]) = ([\gamma]\,x\,y_0\,y_1)[\![\beta \setminus^{\gamma} y_0 \cdot y_1]\!][\gamma' \setminus \gamma]$.

Por su parte, la *aplicación de un renaming* se define como:

$$
\begin{array}{rcll}
\{\alpha \setminus \beta\}x & \triangleq & x & \\
\{\alpha \setminus \beta\}(t\,u) & \triangleq & \{\alpha \setminus \beta\}t\,\{\alpha \setminus \beta\}u & \\
\{\alpha \setminus \beta\}(\lambda x.t) & \triangleq & \lambda x.\{\alpha \setminus \beta\}t & \\
\{\alpha \setminus \beta\}(\mu\gamma.c) & \triangleq & \mu\gamma.\{\alpha \setminus \beta\}c & \gamma \neq \alpha, \gamma \neq \beta \\
\{\alpha \setminus \beta\}(t[x \setminus u]) & \triangleq & (\{\alpha \setminus \beta\}t)[x \setminus \{\alpha \setminus \beta\}u] & \\
\{\alpha \setminus \beta\}([\alpha]\,t) & \triangleq & [\beta]\,\{\alpha \setminus \beta\}t & \\
\{\alpha \setminus \beta\}([\delta]\,t) & \triangleq & [\delta]\,\{\alpha \setminus \beta\}t & \alpha \neq \delta \\
\{\alpha \setminus \beta\}(c[\![\gamma \setminus^{\alpha} s]\!]) & \triangleq & (\{\alpha \setminus \beta\}c)[\![\gamma \setminus^{\beta} \{\alpha \setminus \beta\}s]\!] & \gamma \neq \alpha, \gamma \neq \beta \\
\{\alpha \setminus \beta\}(c[\![\gamma \setminus^{\delta} s]\!]) & \triangleq & (\{\alpha \setminus \beta\}c)[\![\gamma \setminus^{\delta} \{\alpha \setminus \beta\}s]\!] & \alpha \neq \delta, \gamma \neq \alpha, \gamma \neq \beta \\
\{\alpha \setminus \beta\}(c[\gamma \setminus \alpha]) & \triangleq & (\{\alpha \setminus \beta\}c)[\gamma \setminus \beta] & \gamma \neq \alpha, \gamma \neq \beta \\
\{\alpha \setminus \beta\}(c[\gamma \setminus \delta]) & \triangleq & (\{\alpha \setminus \beta\}c)[\gamma \setminus \delta] & \alpha \neq \delta, \gamma \neq \alpha, \gamma \neq \beta \\
\{\alpha \setminus \beta\}(t \cdot s) & \triangleq & \{\alpha \setminus \beta\}t \cdot \{\alpha \setminus \beta\}s &
\end{array}
$$

Las tres operaciones $\{x \setminus u\}$, $\{\!|\alpha \setminus^{\alpha'} s|\!\}$ y $\{\alpha \setminus \beta\}$ se extienden a contextos como es esperado.

La *relación de reducción en un paso* $\to_{\Lambda M}$ se define como la clausura por contextos $\mathtt{O_t}$ de las reglas de reescritura `dB`, `S` y `dM`, y por contextos $\mathtt{O_c}$ de la regla `R` a continuación, *i.e.* $\to_{\Lambda M} \triangleq$

$\mathtt{O_t}\langle \mapsto_{\mathtt{dB}} \cup \mapsto_{\mathtt{S}} \cup \mapsto_{\mathtt{dM}} \rangle \cup \mathtt{O_c}\langle \mapsto_{\mathtt{R}} \rangle$:

$$\begin{array}{rcl@{\qquad\qquad}rcl} \mathtt{L}\langle \lambda x.t\rangle\, u & \mapsto_{\mathtt{dB}} & \mathtt{L}\langle t[x \setminus u]\rangle & \mathtt{L}\langle \mu\alpha.c\rangle\, u & \mapsto_{\mathtt{dM}} & \mathtt{L}\langle \mu\alpha'.c[\![\alpha \setminus^{\alpha'} u]\!]\rangle \\ t[x \setminus u] & \mapsto_{\mathtt{S}} & \{x \setminus u\}t & c[\![\alpha \setminus^{\alpha'} s]\!] & \mapsto_{\mathtt{R}} & \{\!\{\alpha \setminus^{\alpha'} s\}\!\}c \end{array}$$

donde $\mapsto_{\mathtt{dB}}$ y $\mapsto_{\mathtt{dM}}$ aplican bajo la condición adicional $\mathtt{fc}(u, \mathtt{L})$, y $\mapsto_{\mathtt{dM}}$ requiere a su vez que α' sea un nombre fresco (*i.e.* $\alpha' \neq \alpha$, $\alpha' \notin c$ y $\alpha' \notin \mathtt{L}$), de modo que ambas reglas extraen el contexto $\mathtt{L}$ evitando la captura indeseada de variables/nombres libres en u. Dado $X \in \{\mathtt{dB}, \mathtt{S}, \mathtt{dM}, \mathtt{R}\}$, se denota $\to_X$ la clausura por contextos de $\mapsto_X$.

Notar que $\mathtt{dB}$ y $\mathtt{dM}$, presentadas anteriormente de manera informal, operan *a distancia* [AK12], en el sentido que una abstracción y su argumento pueden estar separados por un número arbitrario de sustituciones explícitas; una característica consistente con el hecho de que el desarrollo semántico esté guiado por las proof-nets. Además, siguiendo la presentación de Parigot [Par92], podría quererse simplificar la sintaxis apelando a un constructor binario de replacement, obteniendo $\mathtt{L}\langle \mu\alpha.c[\![\alpha \setminus u]\!]\rangle$ al lado derecho de la regla $\mathtt{dM}$. Esto resulta inadecuado en una presentación con operadores explícitos dado que, si bien la regla $\mathtt{R}$ dispara un reemplazo de todas las ocurrencias de α en c, podría quererse eventualmente realizar reemplazos individuales, al estilo de la regla $\mathtt{lsv}$ del λ-cálculo call-by-need (*cf.* Sec. 2.7), utilizando entonces el nombre del replacement (*i.e.* α' en el operador ternario) para marcar aquellos comandos que ya fueron reemplazados, preservando así la correcta semántica operacional [KV19b]. Por ejemplo, considerar

$$([\alpha]\, \mu\beta.[\alpha]\, t)[\![\alpha \setminus^{\alpha'} u]\!] \to ([\alpha]\, \mu\beta.[\alpha']\, t\, u)[\![\alpha \setminus^{\alpha'} u]\!]$$

donde el reemplazo se realiza únicamente sobre el comando interno, re-nombrando a éste último por α', al mismo tiempo que se preserva el operador explícito que aún liga la ocurrencia más externa de α.

Cabe destacar también la inclusión del operador de renaming explícito en la sintaxis de cálculo. Como se mencionó anteriormente, el (meta-)renaming no interviene en la semántica operacional del $\lambda\mu$-cálculo, por lo que el renaming explícito no es introducido ni ejecutado por las reglas del ΛM-cálculo. Sin embargo, se incorpora este último a la sintaxis para salvar la situación ilustrada en el ejemplo (4.3) como se verá al formalizar la relación $\simeq$ en la Sec. 4.6.

El ΛM-cálculo implementa al $\lambda\mu$-cálculo realizando pasos más atómicos.

Lema 4.2.1. *Sea $o \in \mathbb{O}_{\lambda\mu}$. Si $o \to_{\lambda\mu} o'$, entonces $o \twoheadrightarrow_{\Lambda M} o'$.*

Tal como el $\lambda\mu$-cálculo, el ΛM resulta confluente.

Teorema 4.2.2. *La relación de reducción $\to_{\Lambda M}$ es confluente (CR).*

Demostración. La propiedad se prueba usando el método de interpretación de Hardin [CHL96], proyectando el ΛM-cálculo en una variante del $\lambda\mu$-cálculo con renamings explícitos que, a su vez, se demuestra confluente. Detalles en el Ap. C. □

4.3. Sistema de tipos simples para ΛM-cálculo

Se introduce a continuación el sistema de tipos simples para ΛM-cálculo, que extiende al de $\lambda\mu$-cálculo incorporando reglas para los operadores explícitos. El conjunto de *tipos* $\mathcal{T}$ es esencialmente el mismo que el presentado en la Sec. 4.1.2. Simplemente se incorpora como categoría sintáctica adicional los *tipos stack*, los cuales serán utilizados para dar semántica a los stacks introducidos en ΛM:

$$\textbf{Tipos stacks} \quad S \quad ::= \quad A \mid A \cdot S$$

El constructor de tipos $\cdot$ debe ser entendido como una conjunción no-conmutativa, que se interpreta como el tensor de la lógica lineal (*cf.* Sec. 4.1.3). El rol de los stacks en el ΛM-cálculo es el de acumular

$$\frac{\Gamma; (x : B)^{\leq 1} \vdash t : A \mid \Delta \quad \Gamma' \vdash u : B \mid \Delta'}{\Gamma \cup \Gamma' \vdash t[x \setminus u] : A \mid \Delta \cup \Delta'} \text{(SUBS)}$$

$$\frac{\Gamma \vdash c \mid \Delta; (\alpha : S \to B)^{\leq 1}; (\alpha' : B)^{\leq 1} \quad \Gamma' \vdash s : S \mid \Delta'; (\alpha' : B)^{\leq 1}}{\Gamma \cup \Gamma' \vdash c[\![\alpha \setminus^{\alpha'} s]\!] \mid \Delta \cup \Delta'; \alpha' : B} \text{(REPL)}$$

$$\frac{\Gamma \vdash c \mid \Delta; (\alpha : A)^{\leq 1}; (\beta : A)^{\leq 1}}{\Gamma \vdash c[\alpha \setminus \beta] \mid \Delta; \beta : A} \text{(REN)} \qquad \frac{\Gamma \vdash t : A \mid \Delta \quad \Gamma' \vdash s : S \mid \Delta'}{\Gamma \cup \Gamma' \vdash t \cdot s : A \cdot S \mid \Delta \cup \Delta'} \text{(STK)}$$

Figura 4.11: Reglas de tipado adicionales del sistema $\mathcal{E}$ para el ΛM-cálculo.

argumentos que serán eventualmente aplicados a un término de tipo funcional. Por lo tanto, se usa la notación $A_1 \cdot \ldots \cdot A_n \to B$ como abreviatura del tipo $A_1 \to \ldots \to A_n \to B$.

Las reglas de tipado para el sistema $\mathcal{E}$ del ΛM-cálculo están dadas por aquellas del sistema $\mathcal{M}$ (*cf.* Fig. 4.5) más las presentadas en la Fig. 4.11. Estas nuevas reglas involucran los constructores de objectos introducidos en ΛM para lo que, en particular, se introduce una tercera clase de *juicio de tipado* $\Gamma \vdash s : S \mid \Delta$ para stacks. Notar que todo juicio de tipado para términos es, en particular, un juicio de tipado para stacks (de un solo elemento), actuando de caso base para la construcción de estos últimos. Al igual que en el sistema $\mathcal{M}$, los comandos no tiene tipo asociado, *cf.* reglas (REPL) y (REN); mientras que los stacks son tipados con tipos stack, resultando esencialmente en listas heterogéneas, *i.e.* cada elemento del stack puede tener asociado un tipo diferente. Es interesante destacar la regla (REPL) que se corresponde con la regla lógica de *modus ponens*, donde el nombre α' puede, en principio, estar presente en el tipado del comando c o el stack s.

Se denotan de forma genérica $\Gamma \vdash o : T \mid \Delta$ los tres tipos de juicios, resultando $o = t$ y $T = A$ en el caso de los términos; $o = c$ y T ignorado para los comandos; y $o = s$ y $T = S$ para los stacks. Así mismo se usa $\pi \rhd_{\mathcal{E}} \Gamma \vdash o : T \mid \Delta$ para indicar que el juicio $\Gamma \vdash o : T \mid \Delta$ es derivable en $\mathcal{E}$ y nombrar a la derivación π. En tal caso, se dice que o es el *sujeto* de π. Por conveniencia, usualmente se abrevia con π_o a una derivación con sujeto o para contextos y tipo adecuados.

El sistema de tipos $\mathcal{E}$ posee las siguiente propiedades principales.

Lema 4.3.1 (Relevance)**.** *Sea* $o \in \mathbb{O}_{\Lambda M}$*. Si* $\pi \rhd_{\mathcal{E}} \Gamma \vdash o : T \mid \Delta$*, entonces* $\mathsf{dom}(\Gamma) = \mathsf{fv}(o)$ *y* $\mathsf{dom}(\Delta) = \mathsf{fn}(o)$*.*

Teorema 4.3.2 (Subject Reduction)**.** *Sea* $o \in \mathbb{O}_{\Lambda M}$*. Si* $\pi \rhd_{\mathcal{E}} \Gamma \vdash o : T \mid \Delta$ *y* $o \to_{\Lambda M} o'$*, entonces existe* $\pi' \rhd_{\mathcal{E}} \Gamma' \vdash o' : T \mid \Delta'$ *tal que* $\Gamma' \subseteq \Gamma$ *y* $\Delta' \subseteq \Delta$*.*

Demostración. Por inducción en $o \to_{\Lambda M} o'$ apelando a lemas de sustitución y replacement adecuados. Detalles en el Ap. C. □

Adicionalmente, puede extenderse de manera directa el resultado de preservación de tipos para la relación $\simeq_\sigma$ del sistema $\mathcal{M}$ (*cf.* Lem. 4.1.9) a $\mathcal{E}$.

Lema 4.3.3. *Sea* $o \in \mathbb{O}_{\lambda\mu}$*. Si* $\pi \rhd_{\mathcal{E}} \Gamma \vdash o : T \mid \Delta$ *y* $o \simeq_\sigma o'$*, entonces existe* $\pi' \rhd_{\mathcal{E}} \Gamma \vdash o' : T \mid \Delta'$*.*

En adelante, siempre que $o \to_{\Lambda M} o'$ (u $o \simeq_\sigma o'$), se hará referencia a π_o y $\pi_{o'}$ como dos derivaciones de tipos *relacionadas*, *i.e.* las obtenidas a partir de π_o por la prueba del Teo. 4.3.2 (o Lem. 4.3.3 respectivamente).

4.4. Traducción a proof-nets

Esta sección presenta la traducción de objetos de ΛM-cálculo a PPNs, que resulta de extender la introducida anteriormente para el $\lambda\mu$-cálculo (*cf.* Sec. 4.1.3).

En primer lugar se extiende la traducción de tipos para contemplar el caso de los tipos stack. En este caso, se hace considerando un tipo simple B como parámetro adicional que representa el co-dominio del funcional que consume el stack (*cf.* regla (REPL)), obteniendo así una fórmula output válida.

Definición 4.4.1. *La* traducción de un tipo stack S relativa a un tipo simple B *en una fórmula output polarizada, notada* $S^{\diamond}_{B}$*, se define como:*

$$\begin{array}{rcl} A^{\diamond}_{B} & \triangleq & (A \to B)^{\diamond} \\ (A \cdot S)^{\diamond}_{B} & \triangleq & ?(A^{\diamond\perp}) \parr S^{\diamond}_{B} \end{array}$$

La traducción de objetos tipados del ΛM-cálculo a proof-nets polarizadas extiende a la presentada en la Def. 4.1.7 incorporando los operadores explícitos de ΛM. Al igual que en el caso de los términos, la proof-net resultante de traducir un stack tiene una conclusión *distinguida* que resulta ahora en una fórmula anti-output. Además, del mismo modo que para las fórmulas, las derivaciones de tipo para stacks se traducen relativas a un tipo simple B. En resumen:

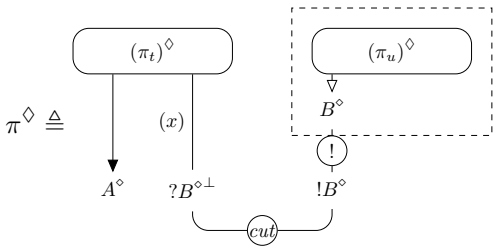
Término Una derivación de tipo $\pi \rhd_{\mathcal{E}} \Gamma \vdash t : A \mid \Delta$ se traduce a una proof-net con conclusiones $?(\Gamma^{\diamond\perp})$, $A^{\diamond}$ y $\Delta^{\diamond}$, donde $A^{\diamond}$ (una fórmula output) es la distinguida.

Comando Una derivación de tipo $\pi \rhd_{\mathcal{E}} \Gamma \vdash c \mid \Delta$ se traduce a una proof-net con conclusiones $?(\Gamma^{\diamond\perp})$ y $\Delta^{\diamond}$. No hay conclusión distinguida.

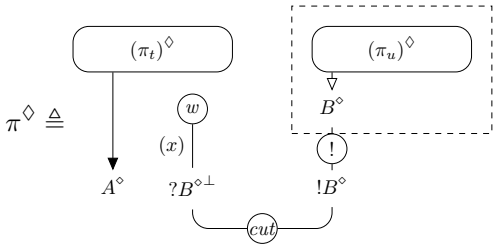
Stack Una derivación de tipo $\pi \rhd_{\mathcal{E}} \Gamma \vdash s : S \mid \Delta$ se traduce, relativa a un tipo simple B, a una proof-net con conclusiones $?(\Gamma^{\diamond\perp})$, $(S \to B)^{\diamond\perp}$, $B^{\diamond}$ y $\Delta^{\diamond}$, donde $(S \to B)^{\diamond\perp}$ (una fórmula anti-output) es la distinguida.

Definición 4.4.2. *La* traducción de una derivación de tipo $\pi \rhd_{\mathcal{E}} \Gamma \vdash o : T \mid \Delta$ *(para un término o comando) en una proof-net polarizada, notada* $\pi^{\Diamond}$*, se define inductivamente como* π° *para las reglas comunes a ambos sistemas, incorporando los siguientes casos para las reglas de la Fig. 4.11:*

- (SUBS)*:* i.e. $\pi \rhd_{\mathcal{E}} \Gamma \cup \Gamma' \vdash t[x \setminus u] : A \mid \Delta \cup \Delta'$ *con premisas* π_t *y* $\pi_u \rhd_{\mathcal{E}} \Gamma' \vdash u : B \mid \Delta'$*:*
  - *si* $x \in \mathsf{fv}(t)$*, entonces* $\pi_t \rhd_{\mathcal{E}} \Gamma; x : B \vdash t : A \mid \Delta$.



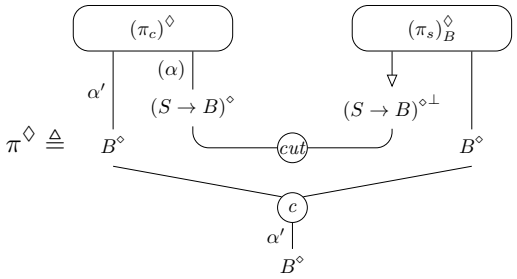
- *si* $x \notin \mathsf{fv}(t)$, *entonces* $\pi_t \rhd_{\mathcal{E}} \Gamma \vdash t : A \mid \Delta$ *con* $x \notin \mathsf{dom}(\Gamma)$.



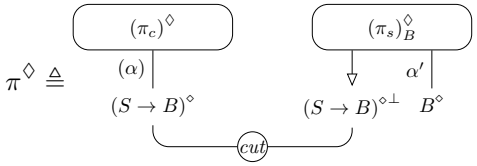
Se agregan nodos de contracción para las conclusiones correspondientes a variables/nombres comunes entre t *y* u.

- (REPL)*:* i.e. $\pi \rhd_{\mathcal{E}} \Gamma \cup \Gamma' \vdash c[\![\alpha \setminus^{\alpha'} s]\!] \mid \Delta \cup \Delta'; \alpha' : B$ *con premisas* π_c *y* $\pi_s \rhd_{\mathcal{E}} \Gamma' \vdash s : S \mid \Delta'; (\alpha' : B)^{\leq 1}$. *Se detallan los casos donde* $\alpha' \notin \mathsf{fn}(s)$ *y, por lo tanto,* $\alpha' \notin \mathsf{dom}(\Delta')$*:*

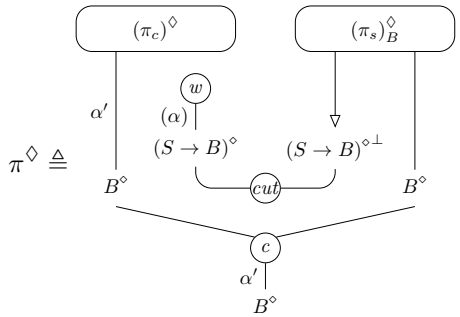
  - *si* $\alpha, \alpha' \in \mathsf{fn}(c)$, *entonces* $\pi_c \rhd_{\mathcal{E}} \Gamma \vdash c \mid \Delta; \alpha : S \to B; \alpha' : B$.



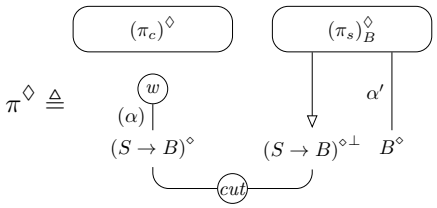
  - *si* $\alpha \in \mathsf{fn}(c)$ *y* $\alpha' \notin \mathsf{fn}(c)$, *entonces* $\pi_c \rhd_{\mathcal{E}} \Gamma \vdash c \mid \Delta; \alpha : S \to B$ *con* $\alpha' \notin \mathsf{dom}(\Delta)$.



  - *si* $\alpha \notin \mathsf{fn}(c)$ *y* $\alpha' \in \mathsf{fn}(c)$, *entonces* $\pi_c \rhd_{\mathcal{E}} \Gamma \vdash c \mid \Delta; \alpha' : B$ *con* $\alpha \notin \mathsf{dom}(\Delta)$.



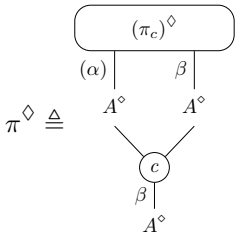
- *si* $\alpha, \alpha' \notin \mathsf{fn}(c)$*, entonces* $\pi_c \rhd_{\mathcal{E}} \Gamma \vdash c \mid \Delta$ *con* $\alpha, \alpha' \notin \mathsf{dom}(\Delta)$.



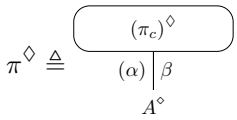
Se agregan nodos de contracción para las conclusiones correspondientes a variables/nombres comunes entre c *y* s*. En los casos en que* $\alpha' \in \mathsf{fn}(s)$ *simplemente se agrega un nodo contracción extra contra la conclusión* α'*.*

- (REN)*:* i.e. $\pi \rhd_{\mathcal{E}} \Gamma \vdash c[\alpha \setminus \beta] \mid \Delta; \beta : A$ *con premisa* π_c*:*

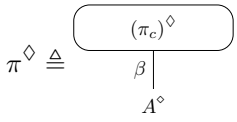
  - *si* $\alpha, \beta \in \mathsf{fn}(c)$*, entonces* $\pi_c \rhd_{\mathcal{E}} \Gamma \vdash c \mid \Delta; \alpha : A; \beta : A$.



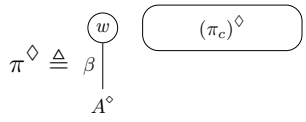
  - *si* $\alpha \in \mathsf{fn}(c)$ *y* $\beta \notin \mathsf{fn}(c)$*, entonces* $\pi_c \rhd_{\mathcal{E}} \Gamma \vdash c \mid \Delta; \alpha : A$ *con* $\alpha' \notin \mathsf{dom}(\Delta)$.



  - *si* $\alpha \notin \mathsf{fn}(c)$ *y* $\beta \in \mathsf{fn}(c)$*, entonces* $\pi_c \rhd_{\mathcal{E}} \Gamma \vdash c \mid \Delta; \beta : A$ *con* $\alpha \notin \mathsf{dom}(\Delta)$.

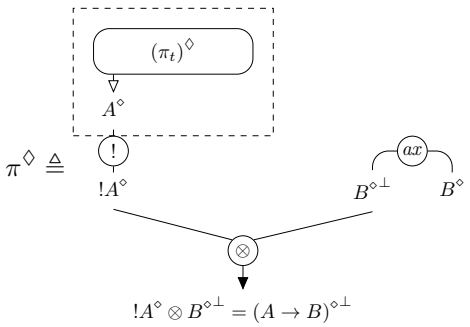


  - *si* $\alpha, \beta \notin \mathsf{fn}(c)$*, entonces* $\pi_c \rhd_{\mathcal{E}} \Gamma \vdash c \mid \Delta$ *con* $\alpha, \alpha' \notin \mathsf{dom}(\Delta)$.

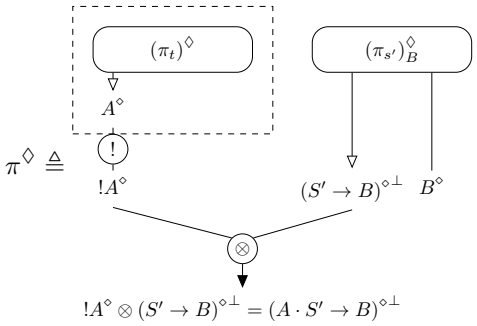


donde la traducción de una derivación de tipo $\pi \rhd_{\mathcal{E}} \Gamma \vdash s : A \mid \Delta$ *en una proof-net polarizada* relativa a un tipo simple B*, notada* $\pi^\Diamond_B$*, se define inductivamente sobre* s *como:*

- $s = t$: i.e. $\pi \rhd_{\mathcal{E}} \Gamma \vdash t : A \mid \Delta$.



- $s = t \cdot s'$: i.e. $\pi \rhd_{\mathcal{E}} \Gamma \cup \Gamma' \vdash t \cdot s' : A \cdot S' \mid \Delta \cup \Delta'$ *con premisas* $\pi_t \rhd_{\mathcal{E}} \Gamma \vdash t : A \mid \Delta$ *y* $\pi_{s'} \rhd_{\mathcal{E}} \Gamma' \vdash s' : S' \mid \Delta$ *por* (STK).



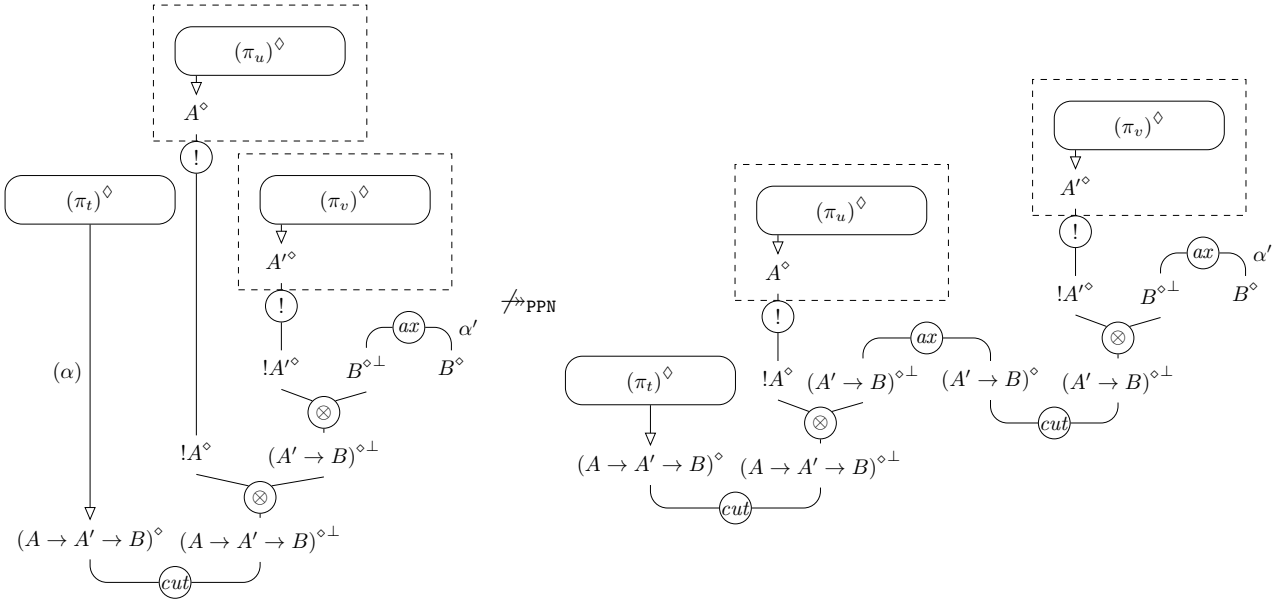
Se agregan nodos de contracción para las conclusiones correspondientes a variables/nombres comunes entre t y s.

Notar que las únicas reglas de tipado que introducen cuts al ser traducidas son (APP), (SUBS) y (REPL). La primera resulta en un cut multiplicativo $\mathtt{C}(\otimes, \parr)$, lo que sustenta la decisión (momentánea) de considerar a las formas canónicas como $\mathcal{NF}_{\mathtt{dBdM}}$. Por su parte, la segunda introduce un cut exponencial, pues intervienen boxes y se aplican necesariamente las reglas de corte de la Fig. 4.8.

Más interesante y sutil es el caso de (REPL) con sujeto $c[\![\alpha \backslash^{\alpha'} s]\!]$, pues la naturaleza del cut resultante depende del número de ocurrencias libres del nombre α en el comando c (*i.e.* $\mathtt{fn}_{\alpha}(c)$). Recordar que la traducción de un tipo funcional a una fórmula polarizada resulta en un par ($\parr$), mientras que su negación es un tensor ($\otimes$). Por construcción, la conclusión distinguida de $(\pi_s)^{\Diamond}_{B}$ siempre proviene de un nodo tensor. Sin embargo, la conclusión α en $(\pi_c)^{\Diamond}$ no necesariamente proviene de un nodo par, pudiendo resultar en un cut multiplicativo ($\mathtt{C}(\otimes, !)$) o exponencial (Fig. 4.9) según el rol de α en el comando c.

A diferencia de la traducción original para el caso del $\lambda\mu$-cálculo (*cf.* Def. 4.1.7), la cual preserva la reducción de PPNs, la extensión presentada aquí requiere resolver algunos cuts multiplicativos para lograr una simulación apropiada. Esto se debe a que los stacks son utilizados en ΛM para acumular argumentos de un comando que, al momento de contraer un $\mathtt{R}$-redex serán aplicados apropiadamente (*cf.* definición de replacement en Sec. 4.2). Sin embargo, dichos stacks son traducidos a $\otimes$-trees, *i.e.* secuencias de nodos $\otimes$ terminadas en un axioma, mientras que las aplicaciones del reducto son interpretadas como un nodo $\otimes$ seguido de un axioma y combinadas mediante nodos cut, resultando así en $n-1$ cuts $\mathtt{C}(ax)$ para un stack de n términos. Por ejemplo, considerar $c = [\alpha]\, t[\![\alpha \backslash^{\alpha'} u \cdot v]\!]$ con

$\alpha \notin \mathsf{fn}(t)$ y α' un nombre fresco. Luego, $c \to_{\mathtt{R}} c' = [\alpha']\, t\, u\, v$ pero $(\pi_c)^{\Diamond} \not\twoheadrightarrow_{\mathtt{PPN}} (\pi_{c'})^{\Diamond}$, como se ilustra en el siguiente diagrama:



Sin embargo, esto puede ser subsanado si se consideran las formas normales multiplicativas de las PPNs en cuestión. Para eso, se introduce una nueva traducción $\_^{\blacklozenge}$ (apelando a las propiedades CR y SN de la reducción de PPNs, *cf.* Sec. 4.1.3) tal que $\pi^{\blacklozenge}$ es la forma normal multiplicativa de $\pi^{\Diamond}$.

Teorema 4.4.3 (Simulación)**.** *Sean $o, p \in \mathbb{O}_{\Lambda M}$ tipados. Si $o \to_{\Lambda M} p$ y π_o, π_p son las derivaciones de tipos relacionadas correspondientes, entonces $(\pi_o)^{\blacklozenge} \twoheadrightarrow_{\mathtt{PPN}} (\pi_p)^{\blacklozenge}$.*

Demostración. Por inducción en $o \to_{\Lambda M} p$ adaptando los Lem. 11.7 y 11.8 de [Lau02]. Detalles en el Ap. C. □

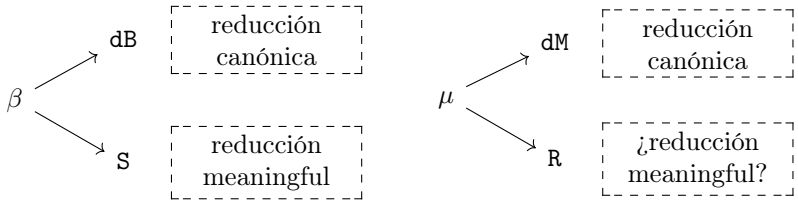
4.5. Reducción meaningful para ΛM-cálculo

Una vez introducido el ΛM-cálculo y tanto su semántica operacional como la interpretación de derivaciones de tipo en proof-nets polarizadas, se cuenta con todas las herramientas necesarias para realizar un análisis acabado del rol de cada regla de reducción a la hora de definir adecuadamente las formas canónicas y la reducción meaningful.

El objetivo principal del presente trabajo es dilucidar una relación de equivalencia estructural sobre el *sub-conjunto* de términos del ΛM-cálculo denominado *formas canónicas*, que sea a su vez una bisimulación fuerte respecto a la noción de *reducción meaningful*. De este modo, se obtendrá para lógica clásica un resultado análogo al presentado en [AK12] para el caso intuicionista. Como se detalló anteriormente, el resultado en [AK12] se vale de una extensión del λ-cálculo con sustituciones explícitas cuya noción de reducción se obtiene separando la regla β en `dB` y `S`. En tal caso, la regla `dB` se corresponde con cuts multiplicativos en la semántica de proof-nets, mientras que `S` captura los cuts exponenciales. Es así que se consideran las $\mathcal{NF}_{\mathtt{dB}}$ el conjunto de formas canónicas y $\to_{\mathtt{S}}$ la reducción meaningful del caso intuicionista.

Tal como se observó en la Sec. 4.4, el hecho de que la regla (APP) introduzca un cut multiplicativo al ser traducida no solo implica que `dB` se corresponde con estos últimos, sino que también `dM` lo hace.

Esto lleva a considerar, en principio, $\mathcal{NF}_{\mathtt{dBdM}}$ como el conjunto de formas canónicas y $\to_{\mathtt{SR}}$ como la reducción meaningful en el caso clásico:



Sin embargo, al analizar la traducción para la regla (REPL) se ve que dB y dM no son los únicas reglas de reducción que resultan multiplicativas respecto a la semántica de proof-nets polarizadas. De hecho, basta analizar el resultado de traducir un comando tipado de la forma $c[\![\alpha \setminus^{\alpha'} s]\!]$ (*i.e.* un R-redex). En particular, el eje α en la traducción puede provenir de un nodo weakening (si $\mathtt{fn}_\alpha(c) = 0$) o de una contracción (si $\mathtt{fn}_\alpha(c) > 1$), resultando ambos casos en cuts exponenciales (*cf.* reglas $\mathtt{C}(\otimes, w)$ y $\mathtt{C}(\otimes, c)$ de la Fig. 4.9 resp.). A su vez, cuando $\mathtt{fn}_\alpha(c) = 1$ se tiene dos posibilidades. Si α se encuentra dentro del argumento de una aplicación o una sustitución explícita, su eje correspondiente al traducir proviene de un box, lo que lleva nuevamente a generar un cut exponencial (en este caso $\mathtt{C}(\otimes, !)$). En el caso contrario, se tiene efectivamente un par ($\parr$) en el nodo origen del eje α, resultado así en un cut multiplicativo $\mathtt{C}(\otimes, \parr)$.

El siguiente análisis busca esclarecer qué instancias de la regla R deben ser consideradas para el cómputo de las formas canónicas, y cuáles otras como parte de la reducción meaningful del caso clásico.

En primer lugar, dado el comando tipado $c[\![\alpha \setminus^{\alpha'} s]\!]$, la linealidad de α es un factor determinante para que la traducción del R-redex resulte en un cut multiplicativo o exponencial. Esto fue remarcado para el $\lambda\mu$-cálculo en [Lau03, Teo. 41], al identificar los pasos de μ-reducción lineales como aquellos que no reflejan cómputo significativo: *i.e.* si o μ-reduce linealmente a o', las formas normales multiplicativas de sus interpretaciones como proof-nets polarizadas son estructuralmente equivalentes. Es por esto que se considera una nueva regla exponencial Rnl para el caso de los replacement *no-lineales*, *i.e.* con $\mathtt{fn}_\alpha(c) \neq 1$:

$$c[\![\alpha \setminus^{\alpha'} s]\!] \quad \mapsto_{\mathtt{Rnl}} \quad \{\!\{\alpha \setminus^{\alpha'} s\}\!\} c \quad \mathtt{fn}_\alpha(c) \neq 1$$

Pero, como se remarcó anteriormente, el número de ocurrencias libres de α en c no es el único factor a considerar. Incluso en el caso $\mathtt{fn}_\alpha(c) = 1$, si α ocurre dentro del parámetro de una aplicación, la traducción del R-redex también resulta en un cut exponencial. Para caracterizar debidamente esta situación, se introduce la noción de *contextos lineales*, generados por la siguiente gramática:

Contextos TT lineales	$\mathtt{LTT}$	$::=$	$\Box \mid \mathtt{LTT}\, t \mid \lambda x.\mathtt{LTT} \mid \mu\alpha.\mathtt{LCT} \mid \mathtt{LTT}[x \setminus t]$
Contextos TC lineales	$\mathtt{LTC}$	$::=$	$\mathtt{LTC}\, t \mid \lambda x.\mathtt{LTC} \mid \mu\alpha.\mathtt{LCC} \mid \mathtt{LTC}[x \setminus t]$
Contextos CC lineales	$\mathtt{LCC}$	$::=$	$\boxdot \mid [\alpha]\,\mathtt{LTC} \mid \mathtt{LCC}[\![\alpha \setminus^{\alpha'} s]\!] \mid \mathtt{LCC}[\alpha \setminus \beta]$
Contextos CT lineales	$\mathtt{LCT}$	$::=$	$[\alpha]\,\mathtt{LTT} \mid \mathtt{LCT}[\![\alpha \setminus^{\alpha'} s]\!] \mid \mathtt{LCT}[\alpha \setminus \beta]$

donde cada categoría LXY representa el contexto lineal que toma un objeto de *sort* Y y retorna otro de *sort* X. Por ejemplo, LTC es un contexto que toma un comando y genera un término. En efecto, la gramática no permite la ocurrencia el agujero del contexto dentro del parámetro de una aplicación o una sustitución explícita.

Analizando el lado izquierdo de la regla R para los casos $\mathtt{fn}_\alpha(c) = 1$ se obtiene las siguientes formas:

$$\mathtt{C}\langle[\alpha]\, t\rangle[\![\alpha \setminus^{\alpha'} s]\!] \qquad \mathtt{C}\langle c'[\![\beta \setminus^{\alpha} s']\!]\rangle[\![\alpha \setminus^{\alpha'} s]\!] \qquad \mathtt{C}\langle c'[\beta \setminus \alpha]\rangle[\![\alpha \setminus^{\alpha'} s]\!]$$

donde α no ocurren en $\mathtt{C}$, t, c' ni s'; lo que deriva en tres posibles reglas de reducción al aplicar la meta-operación de replacement al lado derecho correspondiente:

$$\begin{array}{rcll} \mathtt{C}\langle[\alpha]\,t\rangle[\![\alpha \setminus^{\alpha'} s]\!] & \mapsto_{\mathtt{name}} & \mathtt{C}\langle[\alpha']\,t :: s\rangle & \alpha \notin \mathtt{C}, \alpha \notin t \\ \mathtt{C}\langle c'[\![\beta \setminus^{\alpha} s']\!]\rangle[\![\alpha \setminus^{\alpha'} s]\!] & \mapsto_{\mathtt{comp}} & \mathtt{C}\langle c'[\![\beta \setminus^{\alpha'} s' \cdot s]\!]\rangle & \alpha \notin \mathtt{C}, \alpha \notin c', \alpha \notin s' \\ \mathtt{C}\langle c'[\beta \setminus \alpha]\rangle[\![\alpha \setminus^{\alpha'} s]\!] & \mapsto_{\mathtt{swap}} & \mathtt{C}\langle c'[\![\beta \setminus^{\alpha} s]\!][\alpha \setminus \alpha']\rangle & \alpha \notin \mathtt{C}, \alpha \notin c' \end{array}$$

La primera resuelve el replacement explícito al encontrar el (único) llamado al nombre α; la segunda compone los replacements concatenando sus stacks, notar que el externo actúa sobre el nombre del replacement interno; mientras que la tercera realiza el *swap* entre un replacement y un renaming explícitos, donde el nombre α del lado derecho resulta fresco, como pide la definición de (meta-)replacement, pues no hay otras ocurrencias en el comando (recordar que $\mathtt{fn}_\alpha(c) = 1$). A su vez, el contexto $\mathtt{C}$ puede ser lineal o no-lineal, lo que da lugar a seis posibles reglas (disjuntas) en total. En adelante, se denominan `N` y `Nnl` a las instancias lineal y no-lineal de Name respectivamente; análogamente se tienen `P` y `Pnl` para comP; y por último `W` y `Wnl` para sWap.

Resulta entonces que `R` puede ser particionada en siete reglas de reducción, de las cuales `Rnl`, `Nnl`, `Pnl` y `Wnl` constituyen la porción que introduce cuts exponenciales al traducir; mientras que `N`, `P` y `W` capturan su parte multiplicativa. La *relación de reducción de replacements meaningful* $\to_{\mathtt{R}^\bullet}$ se define como la clausura por contextos de las reglas no-lineales `Rnl`, `Nnl`, `Pnl` y `Wnl`, *i.e.* $\to_{\mathtt{R}^\bullet} \triangleq \mathtt{O_c}\langle \mapsto_{\mathtt{Rnl}} \cup \mapsto_{\mathtt{Nnl}} \cup \mapsto_{\mathtt{Pnl}} \cup \mapsto_{\mathtt{Wnl}}\rangle$. Por su parte, las instancias lineales de `R` permiten manipular replacements explícitos cuya traducción a PPNs introduce cuts multiplicativos, lo que junto con `dB` y `dM` serán consideradas para el cómputo de las formas canónicas.

$$\begin{array}{rcll} \mathtt{LCC}\langle[\alpha]\,t\rangle[\![\alpha \setminus^{\alpha'} s]\!] & \mapsto_{\mathtt{N}} & \mathtt{LCC}\langle[\alpha']\,t :: s\rangle & \alpha \notin \mathtt{LCC}, \alpha \notin t \\ \mathtt{LCC}\langle c'[\![\beta \setminus^{\alpha} s']\!]\rangle[\![\alpha \setminus^{\alpha'} s]\!] & \mapsto_{\mathtt{P}} & \mathtt{LCC}\langle c'[\![\beta \setminus^{\alpha'} s' \cdot s]\!]\rangle & \alpha \notin \mathtt{LCC}, \alpha \notin c', \alpha \notin s' \\ \mathtt{LCC}\langle c'[\beta \setminus \alpha]\rangle[\![\alpha \setminus^{\alpha'} s]\!] & \mapsto_{\mathtt{W}} & \mathtt{LCC}\langle c'[\![\beta \setminus^{\alpha} s]\!][\alpha \setminus \alpha']\rangle & \alpha \notin \mathtt{LCC}, \alpha \notin c' \end{array}$$

La *relación de reducción canónica* $\to_{\mathfrak{C}}$ del ΛM-cálculo se define como la clausura por contextos de las reglas `dB`, `dM`, `N`, `P` y `W`, *i.e.* $\to_{\mathfrak{C}} \triangleq \mathtt{O_t}\langle \mapsto_{\mathtt{dB}} \cup \mapsto_{\mathtt{dM}}\rangle \cup \mathtt{O_c}\langle \mapsto_{\mathtt{N}} \cup \mapsto_{\mathtt{P}} \cup \mapsto_{\mathtt{W}}\rangle$. Luego, el conjunto de *formas canónicas* del ΛM-cálculo queda dado por $\mathcal{NF}_{\mathfrak{C}} = \mathcal{NF}_{\mathtt{dBdMNPW}}$. Más aún, la relación de reducción $\to_{\mathfrak{C}}$ resulta fuertemente normalizante y confluente.

Teorema 4.5.1. *La relación de reducción $\to_{\mathfrak{C}}$ es fuertemente normalizante (SN).*

Demostración. Detalles en el Ap. C. □

Teorema 4.5.2. *La relación de reducción $\to_{\mathfrak{C}}$ es confluente (CR).*

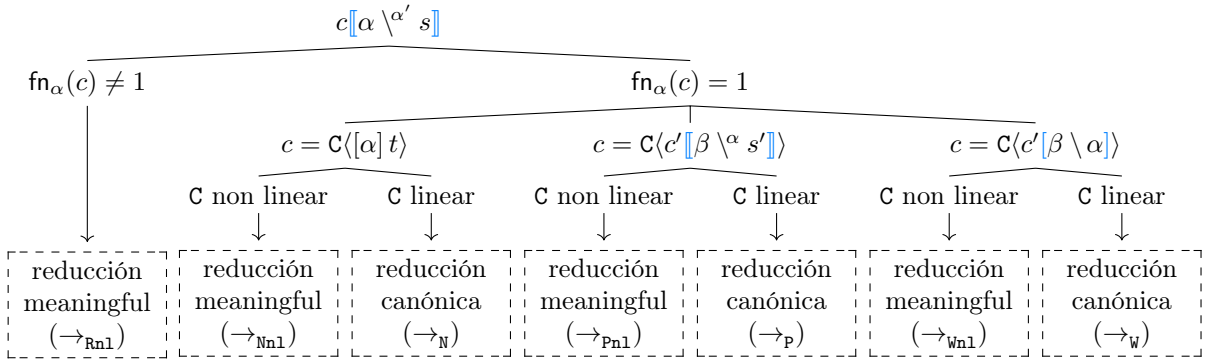
Demostración. Se prueba que relación de reducción $\to_{\mathfrak{C}}$ es débilmente confluente (WCR) y se concluye por Lema de Newman [BN98] apelando al Teo. 4.5.1. Detalles en el Ap. C. □

Luego, es válido referirse a la (única) *forma canónica* de un objeto como $\mathfrak{C}(o) \triangleq \mathtt{nf}_{\mathfrak{C}}(o)$. Finalmente, la *relación de reducción meaningful* $\rightsquigarrow$ para el ΛM-cálculo queda definida sobre formas canónicas en términos de `S` y $\mathtt{R}^\bullet$:

$$o \rightsquigarrow o' \quad \text{sii} \quad t \to_{\mathtt{SR}^\bullet} p \wedge o' = \mathfrak{C}(p)$$

donde $\to_{\mathtt{SR}^\bullet}$ es $\to_{\mathtt{S}} \cup \to_{\mathtt{R}^\bullet}$. Adicionalmente, se usa $\rightsquigarrow_{\mathtt{S}}$ y $\rightsquigarrow_{\mathtt{R}^\bullet}$ para explicitar la regla aplicada en el paso en cuestión.

A modo de resumen, el siguiente diagrama ilustra la descomposición realizada sobre la regla R para terminar obteniendo nociones adecuadas de reducción meaningful y canónica:



Finalmente, se define $\to_{\mathtt{R}\bullet} \triangleq \mathtt{O_c}\langle \mapsto_{\mathtt{Rnl}} \cup \mapsto_{\mathtt{Nnl}} \cup \mapsto_{\mathtt{Pnl}} \cup \mapsto_{\mathtt{Wnl}} \rangle$.

4.6. Equivalencia estructural para ΛM-cálculo

Esta sección introduce la relación de equivalencia estructural $\simeq$ para términos del ΛM-cálculo, separando su presentación en base a las herramientas clave en las que se basa el desarrollo: formas canónicas, contextos lineales y renamings explícitos. Por último se presenta $\simeq$ en sí misma.

Formas canónicas

Como se discutió en las motivaciones del presente capítulo, la intuición inicial para definir una bisimulación fuerte sobre términos de ΛM surge del caso intuicionista: la σ-equivalencia introducida por Regnier no resulta en una bisimulación fuerte, pero al descomponer la regla β y considerar las dB-formas normales a ambos lados de las ecuaciones de la Fig. 4.1 se obtiene la relación de σ^B-equivalencia sobre términos del λ-cálculo con ES que resulta ser efectivamente una bisimulación fuerte respecto a la noción adecuada de reducción meaningful (en tal caso S).

En el caso clásico, una primera aproximación a la solución es considerar las reglas resultantes de tomar la $\mathfrak{C}$-forma normal a cada lado de las ecuaciones de la σ-equivalencia de Laurent (*cf.* Fig. 4.10). La relación $\simeq_{\sigma^B}$ obtenida se ilustra en la Fig. 4.12 pero, desafortunadamente, no satisface la propiedad de bisimulación fuerte buscada, como se explica a continuación.

Renamings explícitos

Como se ilustró en la Sec. 4.1.4, la introducción de la meta-operación de renaming por parte de la regla $\simeq_{\sigma_7^B}$ resulta problemática para la propiedad de bisimulación fuerte buscada (*cf.* ejemplo (4.3)). Es por esto que se introduce en el ΛM-cálculo el operador de renaming explícito y se reemplaza en $\simeq$ la regla $\simeq_{\sigma_7^B}$ por una nueva variante:

$$[\beta]\,\mu\alpha.c \quad \simeq_\rho \quad c[\alpha \setminus \beta]$$

$$
\begin{array}{rcll}
(\lambda y.t)[x \setminus u] & \simeq_{\sigma_1^B} & \lambda y.t[x \setminus u] & y \notin u \\
(t\,v)[x \setminus u] & \simeq_{\sigma_2^B} & t[x \setminus u]\,v & x \notin v \\
(\mu\beta.[\alpha]\,t)[x \setminus u] & \simeq_{\sigma_3^B} & \mu\beta.[\alpha]\,t[x \setminus u] & \beta \notin u \\
[\alpha']\,\mu\alpha''.([\beta']\,\mu\beta''.c[\![\beta \setminus^{\beta''} v]\!])[\![\alpha \setminus^{\alpha''} u]\!] & \simeq_{\sigma_4^B} & [\beta']\,\mu\beta''.([\alpha']\,\mu\alpha''.c[\![\alpha \setminus^{\alpha''} u]\!])[\![\beta \setminus^{\beta''} v]\!] & \alpha \notin v, \alpha'' \notin v, \beta \notin u, \beta'' \notin u, \beta'' \neq \alpha', \alpha'' \neq \beta' \\
[\alpha']\,\mu\alpha''.([\beta']\,\lambda y.\mu\beta.c)[\![\alpha \setminus^{\alpha''} u]\!] & \simeq_{\sigma_5^B} & [\beta']\,\lambda y.\mu\beta.[\alpha']\,\mu\alpha''.c[\![\alpha \setminus^{\alpha''} u]\!] & y \notin u, \beta \notin u, \beta'' \notin u, \beta'' \neq \alpha', \alpha'' \neq \beta' \\
[\alpha']\,\lambda x.\mu\alpha.[\beta']\,\lambda y.\mu\beta.c & \simeq_{\sigma_6^B} & [\beta']\,\lambda y.\mu\beta.[\alpha']\,\lambda x.\mu\alpha.c & \beta \neq \alpha', \alpha \neq \beta' \\
[\beta]\,\mu\alpha.c & \simeq_{\sigma_7^B} & \{\alpha \setminus \beta\}c & \\
\mu\alpha.[\alpha]\,t & \simeq_{\sigma_8^B} & t & \alpha \notin t \\
t[y \setminus v][x \setminus u] & \simeq_{\sigma_9^B} & t[x \setminus u][y \setminus v] & y \notin u, x \notin v
\end{array}
$$

Figura 4.12: σ^B-equivalencia para términos del ΛM-cálculo.

de modo que la situación es ahora resuelta apelando a $\simeq_\rho$ y la reducción canónica. Considerar el ejemplo (4.3) con $\mathsf{fn}_\alpha(c) > 1$ y $\mathsf{fn}_\beta(c) \neq 1$, tomando la $\mathfrak{C}$-forma normal de los objetos iniciales:

$$
\begin{array}{ccc}
\mu\alpha'.([\alpha]\,\mu\beta.c)[\![\alpha \setminus^{\alpha'} u]\!] & \simeq_\rho & \mu\alpha'.c[\beta \setminus \alpha][\![\alpha \setminus^{\alpha'} u]\!] \\
\Big\downarrow{\scriptstyle \mathtt{R}^\bullet} & & \Big\downarrow{\scriptstyle \mathtt{R}^\bullet} \\
\mu\alpha'.[\alpha']\,(\mu\beta.\{\!\{\alpha \setminus^{\alpha'} u\}\!\}c)\,u & & \mu\alpha'.\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} u\}\!\}(c[\beta \setminus \alpha])) \\
\Big\downarrow{\scriptstyle \mathfrak{C}} & & =_\alpha \\
\mu\alpha'.[\alpha']\,\mu\beta'.\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} u\}\!\}c[\![\beta \setminus^{\beta'} u]\!]) & \simeq_\rho & \mu\alpha'.\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} u\}\!\}c)[\![\beta \setminus^{\beta'} u]\!][\beta' \setminus \alpha']
\end{array}
$$

Notar como el comportamiento del (meta-)replacement sobre el renaming explícito juega un rol clave para la α-equivalencia resultante en la parte inferior izquierda del diagrama.

Contextos lineales

Los contextos lineales resultaron de gran utilidad para descomponer la regla $\mathtt{R}$ en la Sec. 4.5. Aquí vuelven a ser utilizados para reducir la cantidad de reglas de la relación, al mismo tiempo que se extiende la misma para soportar conmutaciones de renaming explícitos.

Las ecuaciones $\simeq_{\sigma_1^B}$, $\simeq_{\sigma_2^B}$, $\simeq_{\sigma_3^B}$ y $\simeq_{\sigma_9^B}$ de la Fig. 4.12 pueden ser generalizadas en una única ecuación que refleja el hecho de que una sustitución explícita conmuta con contextos lineales. La misma situación se da con las reglas $\simeq_{\sigma_4^B}$ y $\simeq_{\sigma_5^B}$, entre los contextos lineales y los replacements explícitos. Más aún, los contextos lineales pueden ser atravesados por cualquier operador explícito (sustitución, replacement o renaming) siempre que éste resulte independiente, *i.e.* no haya captura indeseada de variables/nombres libres.

Por lo tanto, se incorporarán a la relación $\simeq$ final tres ecuaciones que reemplazan a $\simeq_{\sigma_1^B}$, $\simeq_{\sigma_2^B}$, $\simeq_{\sigma_3^B}$, $\simeq_{\sigma_4^B}$, $\simeq_{\sigma_5^B}$ y $\simeq_{\sigma_9^B}$, a la vez que extienden su comportamiento a renamings explícitos:

$$
\begin{array}{rcll}
\mathtt{LTT}\langle t\rangle[x \setminus u] & \simeq_{\mathtt{exsubs}} & \mathtt{LTT}\langle t[x \setminus u]\rangle & x \notin \mathtt{LTT}, \mathsf{fc}(u, \mathtt{LTT}) \\
\mathtt{LCC}\langle c\rangle[\![\alpha \setminus^{\alpha'} s]\!] & \simeq_{\mathtt{exrepl}} & \mathtt{LCC}\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle & \alpha \notin \mathtt{LCC}, \mathsf{fc}(\alpha', \mathtt{LCC}), \mathsf{fc}(s, \mathtt{LCC}) \\
\mathtt{LCC}\langle c\rangle[\alpha \setminus \beta] & \simeq_{\mathtt{exren}} & \mathtt{LCC}\langle c[\alpha \setminus \beta]\rangle & \alpha \notin \mathtt{LCC}, \mathsf{fc}(\beta, \mathtt{LCC})
\end{array}
$$

Equivalencia estructural

Finalmente, se está en condiciones de definir la relación de equivalencia estructural $\simeq$ buscada, tomando como base $\simeq_{\sigma^B}$ (*cf.* Fig. 4.12) e incorporando las consideraciones anteriormente mencionadas.

$$
\begin{array}{rcll}
\mathtt{LTT}\langle t\rangle[x \setminus u] & \simeq_{\mathtt{exsubs}} & \mathtt{LTT}\langle t[x \setminus u]\rangle & x \notin \mathtt{LTT}, \mathsf{fc}(u, \mathtt{LTT}) \\
\mathtt{LCC}\langle c\rangle[\![\alpha \setminus^{\alpha'} s]\!] & \simeq_{\mathtt{exrepl}} & \mathtt{LCC}\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle & \alpha \notin \mathtt{LCC}, \mathsf{fc}(\alpha', \mathtt{LCC}), \mathsf{fc}(s, \mathtt{LCC}) \\
\mathtt{LCC}\langle c\rangle[\alpha \setminus \beta] & \simeq_{\mathtt{exren}} & \mathtt{LCC}\langle c[\alpha \setminus \beta]\rangle & \alpha \notin \mathtt{LCC}, \mathsf{fc}(\beta, \mathtt{LCC}) \\
[\alpha']\,\lambda x.\mu\alpha.[\beta']\,\lambda y.\mu\beta.c & \simeq_{\mathtt{pp}} & [\beta']\,\lambda y.\mu\beta.[\alpha']\,\lambda x.\mu\alpha.c & \beta \neq \alpha', \alpha \neq \beta' \\
[\beta]\,\mu\alpha.c & \simeq_{\rho} & c[\alpha \setminus \beta] & \\
\mu\alpha.[\alpha]\,t & \simeq_{\theta} & t & \alpha \notin t
\end{array}
$$

Figura 4.13: Equivalencia estructural $\simeq$ para términos del ΛM-cálculo.

Definición 4.6.1 (Equivalencia estructural para formas canónicas)**.** *La* relación de equivalencia estructural $\simeq$ *es la congruencia sobre formas canónicas del* ΛM*-cálculo generada por las reglas de la Fig. 4.13.*

De este modo, las reglas de conmutación $\simeq_{\sigma_1^B}$, $\simeq_{\sigma_2^B}$, $\simeq_{\sigma_3^B}$ y $\simeq_{\sigma_9^B}$ de la Fig. 4.12 son reemplazadas por $\simeq_{\mathtt{exsubs}}$, mientras que $\simeq_{\mathtt{exrepl}}$ sustituye a $\simeq_{\sigma_4^B}$ y $\simeq_{\sigma_5^B}$, y $\simeq_{\rho}$ hace lo propio con $\simeq_{\sigma_7^B}$. A su vez, se incorpora la nueva regla $\simeq_{\mathtt{exren}}$ discutida anteriormente, y solo se preservan las reglas $\simeq_{\sigma_6^B}$ y $\simeq_{\sigma_8^B}$, aquí llamadas $\simeq_{\mathtt{pp}}$ (por pop/pop) y $\simeq_{\theta}$ respectivamente (ver [Lau03] para el origen de estos nombres). La siguiente tabla resume esta relación entre las distintas reglas:

Regla introducida	**Reglas de σ^B-equivalencia que captura**
$\simeq_{\mathtt{exsubs}}$	$\simeq_{\sigma_1^B}$, $\simeq_{\sigma_2^B}$, $\simeq_{\sigma_3^B}$ y $\simeq_{\sigma_9^B}$
$\simeq_{\mathtt{exrepl}}$	$\simeq_{\sigma_4^B}$ y $\simeq_{\sigma_5^B}$
$\simeq_{\mathtt{exren}}$	
$\simeq_{\mathtt{pp}}$	$\simeq_{\sigma_6^B}$
$\simeq_{\rho}$	$\simeq_{\sigma_7^B}$
$\simeq_{\theta}$	$\simeq_{\sigma_8^B}$

La relación de equivalencia $\simeq$ resultante es de hecho una bisimulación fuerte con respecto a la noción de reducción meaningful, como se demostrará en la Sec. 4.8.

Se presentan a continuación algunas equivalencias admisibles que resultan de interés. En primer lugar los resultados de conmutación entre contextos lineales y contextos de operadores explícitos:

Lema 4.6.2.

1. *Sea* $t \in \mathbb{T}_{\Lambda M}$*. Luego,* $\mathfrak{C}(\mathtt{L}\langle\mathtt{LTT}\langle t\rangle\rangle) \simeq \mathfrak{C}(\mathtt{LTT}\langle\mathtt{L}\langle t\rangle\rangle)$ *si* $\mathsf{bv}(\mathtt{L}) \notin \mathtt{LTT}$ *y* $\mathsf{fc}(\mathtt{L}, \mathtt{LTT})$.
2. *Sea* $c \in \mathbb{C}_{\Lambda M}$*. Luego,* $\mathfrak{C}(\mathtt{R}\langle\mathtt{LCC}\langle c\rangle\rangle) \simeq \mathfrak{C}(\mathtt{LCC}\langle\mathtt{R}\langle c\rangle\rangle)$ *si* $\mathsf{bn}(\mathtt{R}) \notin \mathtt{LCC}$ *y* $\mathsf{fc}(\mathtt{R}, \mathtt{LCC})$.

Una consecuencia de este resultado es la admisibilidad de las reglas reemplazadas de la Fig. 4.12:

- $(\lambda y.t)[x \setminus u] \simeq \lambda y.t[x \setminus u]$ con $y \notin u$, $\mathtt{L} = \Box[x \setminus u]$ y $\mathtt{LTT} = \lambda y.\Box$.
- $(t\,v)[x \setminus u] \simeq t[x \setminus u]\,v$ con $x \notin v$, $\mathtt{L} = \Box[x \setminus u]$ y $\mathtt{LTT} = \Box\,v$.
- $(\mu\beta.[\alpha]\,t)[x \setminus u] \simeq \mu\beta.[\alpha]\,t[x \setminus u]$ con $\alpha \notin u$, $\mathtt{L} = \Box[x \setminus u]$ y $\mathtt{LTT} = \mu\beta.[\alpha]\,\Box$.
- $t[y \setminus v][x \setminus u] \simeq t[x \setminus u][y \setminus v]$ con $y \notin u$, $x \notin v$, $\mathtt{L} = \Box[x \setminus u]$ y $\mathtt{LTT} = \Box[y \setminus v]$.

- $[\alpha']\,\mu\alpha''.([\beta']\,\mu\beta''.c[\![\beta \setminus^{\beta''} v]\!])[\![\alpha \setminus^{\alpha''} u]\!] \simeq [\beta']\,\mu\beta''.([\alpha']\,\mu\alpha''.c[\![\alpha \setminus^{\alpha''} u]\!])[\![\beta \setminus^{\beta''} v]\!]$ con $\alpha \notin v, \alpha'' \notin v, \beta \notin u, \beta'' \notin u, \beta'' \neq \alpha', \alpha'' \neq \beta'$. Este caso es particularmente interesante porque interviene también la regla $\simeq_\rho$. Aplicando precisamente $\simeq_\rho$ dos veces a cada lado, basta ver que
$$(c[\![\beta \setminus^{\beta''} v]\!][\beta'' \setminus \beta'])[\![\alpha \setminus^{\alpha''} u]\!][\alpha'' \setminus \alpha'] \simeq (c[\![\alpha \setminus^{\alpha''} u]\!][\alpha'' \setminus \alpha'])[\![\beta \setminus^{\beta''} v]\!][\beta'' \setminus \beta']$$
lo cual se sigue del Lem. 4.6.2 (2) tomando $\mathtt{R} = \boxdot[\![\alpha \setminus^{\alpha''} u]\!][\alpha'' \setminus \alpha']$ y $\mathtt{LCC} = \boxdot[\![\beta \setminus^{\beta''} v]\!][\beta'' \setminus \beta']$.
- $[\alpha']\,\mu\alpha''.([\beta']\,\lambda y.\mu\beta.c)[\![\alpha \setminus^{\alpha''} u]\!] \simeq [\beta']\,\lambda y.\mu\beta.[\alpha']\,\mu\alpha''.c[\![\alpha \setminus^{\alpha''} u]\!]$ con $y \notin u, \beta \notin u, \beta'' \notin u, \beta'' \neq \alpha', \alpha'' \neq \beta'$. Como en el ejemplo anterior, por $\simeq_\rho$ basta ver que
$$([\beta']\,\lambda y.\mu\beta.c)[\![\alpha \setminus^{\alpha''} u]\!][\alpha'' \setminus \alpha'] \simeq [\beta']\,\lambda y.\mu\beta.c[\![\alpha \setminus^{\alpha''} u]\!][\alpha'' \setminus \alpha']$$
y se concluye por Lem. 4.6.2 (2) tomando $\mathtt{R} = \boxdot[\![\alpha \setminus^{\alpha''} u]\!][\alpha'' \setminus \alpha']$ y $\mathtt{LCC} = [\beta']\,\lambda y.\mu\beta.\boxdot$.

Por último, así como $\simeq_\sigma$ preserva el tipado de objetos del $\lambda\mu$-cálculo (Lem. 4.3.3), la equivalencia estructural aquí definida hace lo propio para objetos de ΛM.

Lema 4.6.3 (Preservación de tipos para $\simeq$)**.** *Sea $o \in \mathbb{O}_{\Lambda M}$. Si $\pi \rhd_{\mathcal{E}} \Gamma \vdash o : T \mid \Delta$ y $o \simeq o'$, entonces existe $\pi' \rhd_{\mathcal{E}} \Gamma \vdash o' : T \mid \Delta'$.*

4.7. Dos resultados de correspondencia

En esta sección se estudia cómo se relaciona la equivalencia $\simeq$ introducida en la Sec. 4.6 con la relación $\simeq_\sigma$ presentada por Laurent y, por lo tanto, con la equivalencia estructural de proof-nets polarizadas $\equiv$. En particular, se busca entender si existen permutaciones capturadas por $\simeq_\sigma$ que sean rechazadas por $\simeq$. Como se vio en la sección Sec. 4.6, las reglas de $\simeq_\sigma$ son capturadas por $\simeq$ una vez adaptadas al marco del ΛM-cálculo (*i.e.* tomando la $\mathfrak{C}$-forma normal de los objetos relacionados), salvo por el caso de la regla $\simeq_{\sigma_7}$:
$$[\beta]\,\mu\alpha.c \quad \simeq_{\sigma_7} \quad \{\alpha \setminus \beta\}c$$
Como se discutió en la Sec. 4.1.4, esta ecuación rompe la propiedad de bisimulación fuerte buscada (*cf.* ejemplo (4.3)), motivando así la introducción del operador de renaming explícito en el ΛM-cálculo y la inclusión de la regla $\simeq_\rho$ en $\simeq$. Se tiene entonces:
$$[\beta]\,\mu\alpha.c \simeq_{\sigma_7} \{\alpha \setminus \beta\}c \quad \text{en } \lambda\mu \qquad \text{vs.} \qquad [\beta]\,\mu\alpha.c \simeq_\rho c[\alpha \setminus \beta] \quad \text{en } \Lambda M$$
lo que da lugar a preguntar si estas permutaciones capturadas por $\simeq_\sigma$ son las únicas que representan un obstáculo a la hora de obtener una bisimulación fuerte. Se prueba en esta sección que ese es precisamente el caso. Esta observación se materializa en la siguiente propiedad (*cf.* Teo. 4.7.8)
$$o \simeq_\sigma p \quad \text{sii} \quad \mathfrak{C}(o) \simeq_{\mathtt{er}} \mathfrak{C}(p)$$
donde $\simeq_{\mathtt{er}}$ es la equivalencia resultante de extender $\simeq$ con una nueva regla que resuelve los renamings explícitos:
$$c[\alpha \setminus \beta] \quad \simeq_{\mathtt{ren}} \quad \{\alpha \setminus \beta\}c$$
Este resultado pone en evidencia la importancia (inesperada) del renaming explícito a la hora de obtener el resultado de bisimulación fuerte.

El siguiente lema representa el resultado de corrección de $\simeq_\sigma$ respecto a $\simeq_{\mathtt{er}}$ (*i.e.* la ida del Teo. 4.7.8), lo cual es relativamente directo de demostrar.

Lema 4.7.1. *Sean $o, p \in \mathbb{O}_{\lambda\mu}$. Si $o \simeq_\sigma p$, entonces $\mathfrak{C}(o) \simeq_{\mathtt{er}} \mathfrak{C}(p)$.*

En adelante se pone el foco en la vuelta del Teo. 4.7.8, *i.e.* el recíproco del lema anterior. El estudio será en dos etapas: por un lado estableciendo formalmente la relación entre $\simeq_{\mathtt{er}}$ y $\equiv$ (Teo. 4.7.7); para luego formalizar la correspondencia entre $\simeq_{\mathtt{er}}$ y $\simeq_\sigma$ (Teo. 4.7.8).

Correspondencia entre $\simeq_{\mathtt{er}}$ y $\equiv$

En una primera etapa se estudia la corrección y completitud de $\simeq_{\mathtt{er}}$ con respecto a la semántica de PPNs. Este resultado se basa en la propiedad de completitud de $\simeq_\sigma$ [Lau03], la cual se describe a continuación. La relación de σ-equivalencia tiene la interesante propiedad de que dos objetos o y p tipables del $\lambda\mu$-cálculo son σ-equivalentes si y solo si sus correspondientes derivaciones de tipo son estructuralmente equivalentes al traducirlas con $\_^\bullet$. En otras palabras, dos objetos σ-equivalentes tiene la misma representación como PPN módulo cuts multiplicativos (*cf.* Teo. 4.1.10).

En la Sec. 4.4 se extendió la traducción $\_^\circ$ de objetos tipados del $\lambda\mu$-cálculo para el caso de ΛM, obteniendo la función $\_^\Diamond$. Del mismo modo, la Sec. 4.3 presenta una extensión del sistema de tipos simples $\mathcal{M}$ del $\lambda\mu$-cálculo para contemplar los operadores explícitos incorporados en ΛM. Luego, dado que $\mathbb{O}_{\Lambda M}$ contiene estrictamente a $\mathbb{O}_{\lambda\mu}$, es evidente que para todo objeto $o \in \mathbb{O}_{\lambda\mu}$ se tiene $(\pi_o)^\Diamond \equiv (\pi_o)^\circ$.

Sea $o \in \mathbb{O}_{\Lambda M}$ un objecto en $\mathfrak{C}$-forma normal, y π_o una derivación de tipos para o. Resulta que $o \in \mathcal{NF}_{\mathfrak{C}}$ no necesariamente implica que $(\pi_o)^\Diamond$ está en forma normal multiplicativa. Considerar por ejemplo el comando $c = [\alpha]\, x[\![\alpha \setminus^{\alpha'} y]\!]$. Si bien c se encuentra en forma canónica, $(\pi_c)^\Diamond$ genera un cut multiplicativo al conectar la PPN $(\pi_{[\alpha]\,x})^\Diamond$ con un box conteniendo $(\pi_y)^\Diamond$ (*cf.* caso (REPL) de la Def. 4.4.2). Es por esto que se apela a la traducción $\_^\blacklozenge$ para objetos de ΛM-cálculo.

Nota 4.7.2. *Para todo $o \in \mathbb{O}_{\lambda\mu}$ tipado se tiene $(\pi_o)^\blacklozenge \equiv (\pi_o)^\bullet$.*

La siguiente tabla resume las distintas traducciones a PPNs mencionadas anteriormente:

Función	Cálculo	Observación
$\_^\circ$	$\lambda\mu$	Incluye cuts multiplicativos
$\_^\bullet$	$\lambda\mu$	Forma normal multiplicativa de $\_^\circ$
$\_^\Diamond$	ΛM	Incluye cuts multiplicativos
$\_^\blacklozenge$	ΛM	Forma normal multiplicativa de $\_^\Diamond$

Un resultado esperado pero igualmente interesante es que las formas normales multiplicativas de las PPNs son preservadas por la reducción canónica.

Lema 4.7.3. *Sean $o, p \in \mathbb{O}_{\Lambda M}$ tipados. Si $o \to_{\mathfrak{C}} p$, entonces $(\pi_o)^\blacklozenge \equiv (\pi_p)^\blacklozenge$, donde π_o y π_p son las correspondientes derivaciones de tipo relacionadas.*

Por otro lado, la equivalencia $\simeq_{\mathtt{er}}$ también preserva las formas normales multiplicativas de las PPNs asociadas. Esto constituye el resultado de corrección de $\simeq_{\mathtt{er}}$ respecto a $\equiv$.

Lema 4.7.4. *Sean $o, p \in \mathbb{O}_{\Lambda M}$ tipados. Si $o \simeq_{\mathtt{er}} p$, entonces $(\pi_o)^\blacklozenge \equiv (\pi_p)^\blacklozenge$, donde π_o y π_p son las correspondientes derivaciones de tipo relacionadas.*

Para obtener el resultado buscado de completitud de $\simeq_{\mathtt{er}}$ respecto a $\equiv$, es necesario relacionar los conjuntos $\mathbb{O}_{\Lambda M}$ y $\mathbb{O}_{\lambda\mu}$ mediante una *función de expansión* $\mathtt{e}(\_)$, que elimina todos los operadores explícitos de un objecto mediante $\mathtt{dB}$, $\mathtt{dM}$ o ρ expansiones:

$$
\begin{array}{rcl}
\mathtt{e}(x) & \triangleq & x \\
\mathtt{e}(t\,u) & \triangleq & \mathtt{e}(t)\,\mathtt{e}(u) \\
\mathtt{e}(\lambda x.t) & \triangleq & \lambda x.\mathtt{e}(t) \\
\mathtt{e}(\mu\alpha.c) & \triangleq & \mu\alpha.\mathtt{e}(c) \\
\mathtt{e}(t[x \setminus u]) & \triangleq & (\lambda x.\mathtt{e}(t))\,\mathtt{e}(u)
\end{array}
\qquad
\begin{array}{rcl}
\mathtt{e}([\alpha]\,t) & \triangleq & [\alpha]\,\mathtt{e}(t) \\
\mathtt{e}(c[\![\alpha \setminus^{\alpha'} s]\!]) & \triangleq & [\alpha']\,(\mu\alpha.\mathtt{e}(c)) :: \mathtt{e}(s) \\
\mathtt{e}(c[\alpha \setminus \beta]) & \triangleq & [\beta]\,\mu\alpha.\mathtt{e}(c) \\
\mathtt{e}(t \cdot s) & \triangleq & \mathtt{e}(t) \cdot \mathtt{e}(s)
\end{array}
$$

Observar que, $\mathsf{e}(\_)$ no es inversa a izquierda de $\mathfrak{C}(\_)$, *i.e.* dado $o \in \mathbb{O}_{\lambda\mu}$, no necesariamente $\mathsf{e}(\mathfrak{C}(o)) = o$. Por ejemplo, tomar $o = (\mu\alpha.[\alpha]\, x)\, y\, z$. Luego, $\mathfrak{C}(o) = \mu\alpha'.([\alpha]\, x)[\![\alpha \setminus^{\alpha'} y \cdot z]\!]$ mientras que $\mathsf{e}(\mathfrak{C}(o)) = \mu\alpha'.[\alpha']\,(\mu\alpha.[\alpha]\, x)\, y\, z$. Sin embargo, se vuelve a un objeto equivalente gracias a la regla $\simeq_{\sigma_8}$.

Dado que los pasos de expansión en el ΛM-cálculo son traducidos a cuts multiplicativos en las proof-nets polarizadas, se tiene:

Lema 4.7.5.

1. *Sea $o \in \mathbb{O}_{\lambda\mu}$. Luego, $(\pi_o)^{\blacklozenge} \equiv (\pi_{\mathfrak{C}(o)})^{\blacklozenge}$, donde π_o y $\pi_{\mathfrak{C}(o)}$ son las correspondientes derivaciones de tipo relacionadas.*
2. *Sea $o \in \mathbb{O}_{\Lambda M}$. Luego, $(\pi_o)^{\blacklozenge} \equiv (\pi_{\mathsf{e}(o)})^{\blacklozenge}$, donde π_o y $\pi_{\mathsf{e}(o)}$ son las correspondientes derivaciones de tipo relacionadas.*

Luego, como corolario inmediato se tiene:

Corolario 4.7.6. *Sean $o \in \mathbb{O}_{\lambda\mu}$ y $o' \in \mathbb{O}_{\Lambda M}$. Luego, $(\pi_o)^{\blacklozenge} \equiv (\pi_{\mathsf{e}(\mathfrak{C}(o))})^{\blacklozenge}$ y $(\pi_{o'})^{\blacklozenge} \equiv (\pi_{\mathfrak{C}(\mathsf{e}(o'))})^{\blacklozenge}$.*

Todo estos resultados auxiliares llevan a la correspondencia buscada entre $\simeq_{\mathtt{er}}$ y la equivalencia estructural de las proof-nets polarizadas.

Teorema 4.7.7. *Sean $o, p \in \mathbb{O}_{\lambda\mu}$. Luego, $\mathfrak{C}(o) \simeq_{\mathtt{er}} \mathfrak{C}(p)$ sii $(\pi_{\mathfrak{C}(o)})^{\blacklozenge} \equiv (\pi_{\mathfrak{C}(p)})^{\blacklozenge}$, donde $\pi_{\mathfrak{C}(o)}$ y $\pi_{\mathfrak{C}(p)}$ son las correspondientes derivaciones de tipo relacionadas.*

Demostración. $\Rightarrow$) Por Lem. 4.7.4, $\mathfrak{C}(o) \simeq_{\mathtt{er}} \mathfrak{C}(p)$ implica $(\pi_{\mathfrak{C}(o)})^{\blacklozenge} \equiv (\pi_{\mathfrak{C}(p)})^{\blacklozenge}$.

$\Leftarrow$) Sean $(\pi_{\mathfrak{C}(o)})^{\blacklozenge} \equiv (\pi_{\mathfrak{C}(p)})^{\blacklozenge}$. Por Lem. 4.7.5 (2), $(\pi_{\mathsf{e}(\mathfrak{C}(o))})^{\blacklozenge} \equiv (\pi_{\mathsf{e}(\mathfrak{C}(p))})^{\blacklozenge}$ y, por Cor. 4.7.6, $(\pi_o)^{\blacklozenge} \equiv (\pi_p)^{\blacklozenge}$. Más aún, por la Nota 4.7.2, $(\pi_o)^{\blacklozenge} \equiv (\pi_o)^{\bullet}$ y $(\pi_p)^{\blacklozenge} \equiv (\pi_p)^{\bullet}$. Luego, por Teo. 4.1.10, $o \simeq_\sigma p$ y se concluye por Lem. 4.7.1, $\mathfrak{C}(o) \simeq_{\mathtt{er}} \mathfrak{C}(p)$. □

Correspondencia entre $\simeq_{\mathtt{er}}$ y $\simeq_\sigma$

Finalmente, se apela a correspondencia entre $\simeq_{\mathtt{er}}$ y la equivalencia estructural de proof-nets polarizadas (Teo. 4.7.7) para terminar relacionando la noción de equivalencia para ΛM-cálculo aquí desarrollada con la usual en la literatura para el $\lambda\mu$-cálculo.

Teorema 4.7.8. *Sean $o, p \in \mathbb{O}_{\lambda\mu}$. Luego, $o \simeq_\sigma p$ sii $\mathfrak{C}(o) \simeq_{\mathtt{er}} \mathfrak{C}(p)$.*

Demostración. $\Rightarrow$) Por Lem. 4.7.1, $o \simeq_\sigma p$ implica $\mathfrak{C}(o) \simeq_{\mathtt{er}} \mathfrak{C}(p)$.

$\Leftarrow$) Sean $\mathfrak{C}(o) \simeq_{\mathtt{er}} \mathfrak{C}(p)$. Por Lem. 4.7.4, $(\pi_{\mathfrak{C}(o)})^{\blacklozenge} \equiv (\pi_{\mathfrak{C}(p)})^{\blacklozenge}$ y, por Lem. 4.7.5 (1), $(\pi_o)^{\blacklozenge} \equiv (\pi_p)^{\blacklozenge}$. Más aún, por la Nota 4.7.2, $(\pi_o)^{\blacklozenge} \equiv (\pi_o)^{\bullet}$ y $(\pi_p)^{\blacklozenge} \equiv (\pi_p)^{\bullet}$. Luego, por Teo. 4.1.10, $(\pi_o)^{\bullet} \equiv (\pi_p)^{\bullet}$ implica $o \simeq_\sigma p$. □

Esto constituye el resultado principal de la presente sección, que puede ser resumido gráficamente por el siguiente diagrama:

$$\begin{array}{ccc} o \simeq_\sigma p & \overset{\text{Teo. 4.7.8}}{\longleftrightarrow} & \mathfrak{C}(o) \simeq_{\mathtt{er}} \mathfrak{C}(p) \\ \Big\updownarrow{\scriptstyle \text{Teo. 4.1.10}} & & \Big\updownarrow{\scriptstyle \text{Teo. 4.7.7}} \\ (\pi_o)^{\bullet} \equiv (\pi_p)^{\bullet} & & (\pi_{\mathfrak{C}(o)})^{\blacklozenge} \equiv (\pi_{\mathfrak{C}(p)})^{\blacklozenge} \end{array}$$

Resulta particulamente interesante resaltar el hecho de que $\simeq_\sigma$ contiene estrictamente a $\simeq$, pero la diferencia entre estas relaciones radica exclusivamente en el manejo que se hace del renaming (implícito vs. explícito). Esto es sucintamente capturado por la regla $\simeq_{\mathtt{ren}}$.

4.8. El resultado de bisimulación fuerte

Tal como establece el Teo. 4.1.10, los objetos del $\lambda\mu$-cálculo que son mapeados a la misma proof-net polarizada (módulo equivalencia estructural) son capturados exactamente por la relación de equivalencia $\simeq_\sigma$ introducida por Laurent, que puede ser vista también como la noción natural de permutación de redexes para el caso de la lógica clásica. Desafortunadamente, como se expuso en anteriormente, $\simeq_\sigma$ no es una bisimulación fuerte respecto a la noción de reducción del $\lambda\mu$-cálculo.

Se propuso en el presente trabajo desarrollar una noción adecuada de equivalencia en donde esta permutación de redexes resulte una bisimulación fuerte, manteniendo al mismo tiempo la semántica estándar de PPNs. La relación $\simeq$ obtenida es efectivamente una bisimulación fuerte sobre formas canónicas respecto a la noción de reducción meaningful definida, *i.e.* términos equivalentes tienen exactamente los mismos redexes. Este resultado se formaliza en la presente sección.

Un aspecto crucial del resultado es la descomposición de $\mapsto_{\mathtt{R}}$ identificando aquellas instancias de replacements *lineales* de las *no-lineales* (*cf.* Sec. 4.5), donde las primeras no involucran cómputo significativo (*i.e.* representan PPNs estructuralmente equivalentes módulo cuts multiplicativos) y son por lo tanto incluidas dentro de la relación de equivalencia $\simeq$. Por su parte, las instancias no-lineales se corresponden con pasos de cut-elimination exponenciales en la semántica de PPNs, siendo consideradas de este modo parte de la reducción meaningful.

En primer lugar se mencionan dos resultados técnicos importantes:

Lema 4.8.1. *Sea $o \in \mathbb{O}_{\Lambda M}$. Si $o \simeq o'$, entonces para todo contexto $\mathtt{O}$ del sort adecuado se tiene $\mathfrak{C}(\mathtt{O}\langle o\rangle) \simeq \mathfrak{C}(\mathtt{O}\langle o'\rangle)$.*

Lema 4.8.2. *Sean $u, s, o \in \mathbb{O}_{\Lambda M}$ en $\mathfrak{C}$-forma normal tales que $o \simeq o'$, $u \simeq u'$ y $s \simeq s'$ con u, u' términos y s, s' stacks. Luego,*

1. $\mathfrak{C}(\{x \setminus u\}o) \simeq \mathfrak{C}(\{x \setminus u\}o')$ *y* $\mathfrak{C}(\{x \setminus u\}o) \simeq \mathfrak{C}(\{x \setminus u'\}o)$.
2. $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o) \simeq \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o')$ *y* $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o) \simeq \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s'\}\!\}o)$.

Por última, se está en condiciones de establecer el resultado principal del presente capítulo. Es decir, el hecho de que $\simeq$ es una bisimulación fuerte respecto a la relación de reducción meaningful.

Teorema 4.8.3. *Sea $o \in \mathbb{O}_{\Lambda M}$. Si $o \simeq p$ y $o \rightsquigarrow o'$, luego existe p' tal que $p \rightsquigarrow p'$ y $o' \simeq p'$.*

Demostración. La prueba es por inducción en $o \simeq p$ utilizando los Lem.4.6.2, 4.8.1 y 4.8.2. Todos los detalles se encuentran en el Ap. C. □

4.9. Conclusión

Este trabajo refina el $\lambda\mu$-cálculo descomponiendo las reglas de reducción en sus fragmentos multiplicativos y exponenciales. Esta nueva presentación de $\lambda\mu$ permite reformular la noción de σ-equivalencia como una bisimulación fuerte $\simeq$ sobre los términos del cálculo extendido ΛM. Adicionalmente, $\simeq$ es conservativa con respecto a la σ-equivalencia original, y términos $\simeq$-equivalentes comparten la misma representación como proof-net polarizada. Más aún, la relación de reducción $\rightarrow_{\Lambda M}$ se demuestra confluente (CR).

Más allá de [Lau03], trabajo que inspiro al presente y fue extensamente discutido, cabe mencionar otros trabajos relacionados. En [Acc13], la MELL polarizada es representada mediante proof-nets sin boxes, usando la información de polaridad para transformar !-boxes explícitos en estructuras más compactas. En [KV17], el $\lambda\mu$-cálculo es reformulado en el $\lambda\mu\mathtt{r}$-cálculo con operadores explícitos, junto con una semántica operacional *small-step* para sustituciones/replacements a distancia. A primera vista

$\lambda\mu\mathtt{r}$ parece ser más atómico que el ΛM-cálculo aunque este último incluye un operador de renaming explícito adicional. Por otro lado, el $\lambda\mu\mathtt{r}$-cálculo fuerza la evaluación de los replacements explícitos de izquierda a derecha, puesto que no cuenta con un mecanismo de composición, y por lo tanto solo se puede aplica el operador sobre *named terms*. Otros refinamientos del $\lambda\mu$-cálculo fueron presentados en [Aud94, Pol04, vBV14]. Otra referencia relacionada es [HL10], donde una correspondencia precisa es establecida entre las PPNs y una versión tipada del π-cálculo asincrónico. Más aún, se muestra que la relación de equivalencia $\simeq_\sigma$ de Laurent se corresponde exactamente con la equivalencia estructural de los procesos del π-cálculo (Prop. 1 en op.cit.). En [LR03], Laurent y Regnier muestran que existe una correspondencia precisa entre las traducciones CPS de los cálculos clásicos (como $\lambda\mu$) en cálculos intuicionistas y las traducciones de PLL en LL.

Además de la confluencia, estudiada en la Sec. 4.2.1, sería interesante analizar otras propiedades de reescritura del ΛM-cálculo, como la preservación de la normalización fuerte respecto a $\lambda\mu$ para las relaciones de reducción $\to_{\Lambda M}$ y $\rightsquigarrow$, o la propia confluencia de $\rightsquigarrow$. Otro tema a estudiar es cómo la noción de bisimulación fuerte propuesta se comporta respecto a otros cálculos clásicos, como por ejemplo $\lambda\mu\widetilde{\mu}$ [CH00]. Más aún, siguiendo la interpretación computacional de *deep inference* provista por el λ-cálculo atómico intuicionista [GHP13], sería interesante investigar una extensión clásica y su correspondiente noción de bisimulación fuerte. También es natural preguntarse cuál sería una sintaxis ideal para la lógica clásica, que sea capaz de capturar la bisimulación fuerte reduciendo los axiomas sintácticos a un conjunto de ecuaciones sencillo.

Se cree también que la relación de reducción $\rightsquigarrow$ es adecuada para derivar una teoría de residuos [Bar85] para el $\lambda\mu$-cálculo. Esto es, tratando $\rightsquigarrow$ como un sistema ortogonal desde un punto de vista diagramático [ABKL14], a pesar de los pares críticos introducidos por las reglas ρ y θ. Eso podría, a su vez, aclarar el panorama en busca de una noción adecuada de call-by-need para el $\lambda\mu$-cálculo vía la noción estándar de reducción necesaria definida por medio de residuos.

Finalmente, la noción de equivalencia $\simeq$ puede facilitar pruebas de corrección entre el $\lambda\mu$-cálculo y máquinas abstractas (como en [ABM14] para el λ-cálculo) y ayudar a establecer si estas últimas son "razonables" [ABM14].

Capítulo 5

Conclusión

En el presente trabajo se exponen tres estudios semánticos realizados sobre el λ-cálculo y algunas de sus extensiones. Este análisis se encuentra enmarcado en el estudio de los fundamentos teóricos subyacentes al desarrollo de lenguajes de programación funcional. Por un lado se estableció una clara relación entre la estrategia de reducción call-by-need del λ-cálculo y la noción semántica de reducción necesaria. En un segundo término se estudió la posibilidad de capturar la noción de path polymorphism por medio de un sistema de tipos para un cálculo con patrones estáticos que garantice la buena semántica operacional del mismo (*i.e.* las propiedades de *subject reduction* y *progress*). Finalmente, un tercer estudio permitió formalizar una noción de equivalencia estructural entre términos de un λ-cálculo enriquecido con operadores de control, garantizando que objetos relacionados poseen exactamente la misma semántica operacional. A continuación de elabora sobre los resultados obtenidos en cada uno de estos estudios y sus posibles líneas de trabajo futuro.

Reducciones necesarias en λ-cálculo. Este estudio es desarrollado en el Cap. 2 de la presente tesis. El principal resultado obtenido es la igualdad observacional entre la estrategia call-by-need y la noción de reducción necesaria. Más precisamente, se muestra que un término del λ-cálculo normaliza al reducir con la estrategia call-by-need si y solo si normaliza para cualquier secuencia de reducción necesaria, *i.e.* que contraiga únicamente redexes needed del término. Para obtener este resultado se hace uso de un sistema de tipos intersección no idempotente –una técnica poderosa capaz de caracterizar diferentes propiedades operacionales– que permite, en particular, caracterizar los redexes (weak-head) needed de un término dado. Dicho sistema se demuestra normalizante: la contracción de redexes tipados lleva siempre a una forma normal (weak-head). A su vez, esto garantiza la normalización de las secuencias de reducciones necesarias, dada la caracterización de redexes (weak-head) needed mencionada. Como resultado indirecto del estudio realizado fue preciso formalizar la noción (no determinista) de reducción para árboles de derivación de tipado.

Los tipos intersección no idempotente proveen una herramienta simple y natural para mostrar la equivalencia observacional obtenida. Sin embargo, existen otras técnicas aplicables a tal fin en la literatura. En particular, en [Bal12] se apela al uso de *sistemas de reducción combinatoria* (CRS) [Klo80, KvOvR93] para demostrar que nociones sintácticas de evaluación lazy resultan optimales [Lév75]. Si bien esta técnica sería aplicable para el fin del presente trabajo, resulta considerablemente más compleja que el uso de tipos intersección, al mismo tiempo que no es adecuada para el caso del λ-cálculo no débil [Gue96].

Esto último representa precisamente una de las posibles líneas de trabajo futuro que se desprenden del estudio realizado. Por un lado, en [Gar94] se introduce originalmente el sistema de tipos intersección no idempotente con el fin de realizar un estudio sobre los redexes needed para la noción de reducción

fuerte del λ-cálculo. Por otro, recientemente se presentó la estrategia de reducción *strong call-by-need* [BBBK17], que describe una estrategia call-by-need determinista a forma normal (full): *i.e.* se permite reducir tanto en el argumento de una aplicación como por debajo de las abstracciones. Esto plantea el interrogante de hasta qué punto el resultado obtenido en el presente trabajo puede ser extendido a la reducción fuerte.

Otra extensión interesante (y a la vez simple) del resultado de caracterización de los redexes (weak-head) needed por parte del sistema de tipos utilizado, sería definir una noción de reducción necesaria adecuada para términos del λ-cálculo con sustituciones explícitas (ES) para así poder establecer una relación más directa entre los redexes needed y aquellos elegidos por la estrategia call-by-need (definida sobre términos con ES). Una herramienta técnica para obtener tal resultado sería el sistema de tipos $\mathcal{A}$ [Kes16], una adaptación directa del sistema $\mathcal{N}$ a la sintaxis del λ-cálculo call-by-need.

Calculus of Applicative Patterns. Esta línea de investigación se desarrolla en el Cap. 3 de la presente tesis, y tiene como motivación la formalización de un prototipo de lenguaje de programación capaz de capturar la novedosa característica de path polymorphism. Para ello se introduce el *Calculus of Applicative Patterns* (CAP) como una variante del fragmento estático de PPC donde la alternativa es una operación nativa (*i.e.* un constructor de términos del cálculo). Se desarrolla un sistema de tipos para el mismo donde términos *path-polimórficos* son tipados estáticamente, garantizando el buen comportamiento dinámico de los mismos: subject reduction y progress. Dicho sistema consta de una combinación no trivial de herramientas conocidas como tipos constantes, aplicativos, unión y recursivos. Las propiedades fundamentales de adecuación recaen crucialmente en dos aspectos del sistema propuesto: por un lado la invertibilidad de la relación de sub-tipado asociada, que se garantiza apelando a la semántica de tipos recursivos sobre árboles infinitos; y por otro una noción novedosa de compatibilidad de patrones que establece una relación entre los tipos asociados a aquellos patrones cuyos conjuntos de argumentos se superponen. Cabe destacar la falta de normalización fuerte en el sistema, la cual se debe principalmente al uso de tipos recursivos con equivalencia fuerte y a la necesidad de capturar la recursión, característica intrínseca del path polymorphism.

En una segunda etapa del estudio se analiza la posibilidad de chequear algorítmicamente la asignación de tipos, lo cual requiere de una reformulación dirigida por sintaxis del sistema. A su vez, es necesario desarrollar algoritmos para el chequeo de equivalencia de tipos y sub-tipado, lo que lleva a introducir formulaciones co-inductivas e invertibles de dichas relaciones. Un primer enfoque es desarrollado derivando en un algoritmo correcto pero ineficiente. En base a este primer estudio se detectan los puntos débiles del algoritmo obtenido y se desarrolla un algoritmo eficiente. Este último se logra principalmente mediante un adecuado cambio de representación, dejando de lado la versión algebraica de la primera etapa e interpretando los μ-tipos contractivos como *term automata*. Finalmente, el algoritmo obtenido resulta tener complejidad polinomial respecto al tamaño de su entrada.

Si bien los resultados obtenidos han sido favorables, existen varias líneas de trabajo futuro con vistas al desarrollo de un prototipo funcional de lenguaje de programación. Por un lado sería deseable incorporar polimorfismo paramétrico, en principio al estilo de *System* $F_{<:}$ [CMMS91, Pie02, CG05], lo que implica adaptar los resultados obtenidos a la presencia de tipos polimórficos teniendo, *a priori*, un impacto directo en la noción de compatibilidad. Por otro lado, es usual utilizar índices de de Bruijn [Bar85, BKvO03] al representar términos del λ-cálculo, para garantizar que no se realizan capturas indeseadas de variables libres producto de un mal manejo de sus nombres. Recientemente se desarrolló una versión *à la* de Bruijn de PPC [MRV20], la cual podría se adaptada a CAP para obtener una representación adecuada de sus términos en el eventual prototipo.

También resulta fundamental definir una estrategia de reducción conveniente para evaluar los términos del cálculo. En esta línea, recientemente se han introducido estrategias de reducción normalizantes para PPC [BKLR12, BKLR17], las cuales deberían ser adaptadas al marco de CAP para el adecuado desarrollo de un prototipo funcional. Cabe destacar que estos dos cálculos comparten la misma opera-

ción de matching. Una extensión más ambiciosa es la de incorporar *pattern polymorphism* al desarrollo. Esto implica incorporar patrones dinámicos a CAP, manteniendo al mismo tiempo las buenas propiedades del sistema, que hoy en día recaen en la noción meramente estática de compatibilidad.

Por último, cabe destacar que el estudio de la normalización fuerte en presencia de tipos recursivos con equivalencia fuerte constituye un problema abierto en área por más de 20 años [BDS13, pág. 515]. Si bien se han logrado avances al respecto en el marco de esta investigación [BV17], no fue posible aún dar con la restricción adecuada que permita garantizar normalización fuerte para un sub-conjunto de los términos del sistema sin descartar por completo el uso de tipos recursivos.

Bisimulación fuerte para operadores de control. Esta última línea de investigación se desarrolla en el Cap. 4 y se basa en un refinamiento del $\lambda\mu$-cálculo, mediante la incorporación de operadores explícitos de sustitución, replacement y renaming, y la descomposición de las reglas de reducción en sus fragmentos multiplicativos y exponenciales. Este nuevo formalismo es denominado ΛM-cálculo, y permite reformular la noción de σ-equivalencia como una bisimulación fuerte $\simeq$ sobre sus objetos (términos y comandos) respecto a la noción de reducción meaningful $\rightsquigarrow$. Adicionalmente, $\simeq$ resulta conservativa con respecto a la σ-equivalencia original de Laurent, al mismo tiempo que objectos $\simeq$-equivalentes comparten la misma representación como proof-nets polarizadas (PPNs). Más aún, la relación de reducción general del $\lambda\mu$-cálculo ($\rightarrow_{\Lambda M}$) se demuestra confluente.

Cabe mencionar distintos trabajos relacionados de los cuales se tomaron elementos para el desarrollo aquí realizado. En primer lugar [Lau03], donde se define la σ-equivalencia original para lógica clásica y se introducen las PPNs. Por otro lado, en [KV17] el $\lambda\mu$-cálculo es reformulado en el $\lambda\mu\mathtt{r}$-cálculo con operadores explícitos, junto con una semántica operacional *small-step* para sustituciones/replacements a distancia. A primera vista $\lambda\mu\mathtt{r}$ parece ser más atómico que el ΛM-cálculo aunque este último incluye un operador de renaming explícito adicional. Por otro lado, el $\lambda\mu\mathtt{r}$-cálculo fuerza la evaluación de los replacements explícitos de izquierda a derecha, puesto que no cuenta con un mecanismo de composición, y por lo tanto solo se puede aplicar el operador sobre *named terms*. La incorporación de operadores explícitos y la descomposición de la relación de reducción en pasos multiplicativos y exponenciales está inspirada en [AK10, ABKL14] donde se aplican estas ideas para formalizar la σ-equivalencia como una bisimulación fuerte para el caso intuicionista.

Entre otros trabajos relacionados que merecen mención se destaca [Acc13], donde la MELL polarizada es representada mediante proof-nets sin boxes, usando la información de polaridad para transformar !-boxes explícitos en estructuras más compactas. Refinamientos adicionales del $\lambda\mu$-cálculo fueron presentados en [Aud94, Pol04, vBV14]. En [HL10] una correspondencia precisa es establecida entre las PPNs y una versión tipada del π-cálculo asincrónico. Más aún, se muestra que la relación de equivalencia $\simeq_\sigma$ de Laurent se corresponde exactamente con la equivalencia estructural de los procesos del π-cálculo (Prop. 1 en op.cit.). Por último, en [LR03] Laurent y Regnier muestran que existe una correspondencia precisa entre las traducciones CPS de los cálculos clásicos (como $\lambda\mu$) en cálculos intuicionistas y las traducciones de PLL en LL.

Respecto a líneas de trabajo futuro, además de la confluencia de ΛM, sería interesante analizar otras propiedades de reescritura del cálculo, como la preservación de la normalización fuerte respecto a $\lambda\mu$ para las relaciones de reducción $\rightarrow_{\Lambda M}$ y $\rightsquigarrow$, o la propia confluencia de $\rightsquigarrow$, la cual no es abordada en el presente trabajo. Otro tema a estudiar es la relación entre la noción de bisimulación fuerte propuesta respecto a otros cálculos clásicos, como $\lambda\mu\widetilde{\mu}$ [CH00]. Más aún, siguiendo la interpretación computacional de *deep inference* provista por el λ-cálculo atómico intuicionista [GHP13], sería interesante investigar una extensión clásica y su correspondiente noción de bisimulación fuerte. También es natural preguntarse si la sintaxis del ΛM-cálculo propuesto es realmente adecuada para la lógica clásica. En otras palabras, sería interesante buscar una sintaxis sencilla para un cálculo que se corresponda (vía Curry–Howard) con la lógica clásica y sea capaz de capturar la bisimulación fuerte reduciendo las ecuaciones sintácticas actuales a un conjunto sencillo de axiomas. Otro punto interesante a investigar

es la posibilidad de derivar una teoría de residuos [Bar85] para el $\lambda\mu$-cálculo basada en la noción de reducción $\rightsquigarrow$. Esto podría ayudar a formalizar una noción adecuada de call-by-need para el $\lambda\mu$-cálculo vía la noción estándar de reducción necesaria definida por medio de residuos. Finalmente, la noción de equivalencia $\simeq$ puede facilitar pruebas de corrección entre el $\lambda\mu$-cálculo y máquinas abstractas (como en [ABM14] para el λ-cálculo) y ayudar a establecer si estas últimas son "razonables" [ABM14].

Apéndice A

Pruebas: Reducciones necesarias en λ-cálculo

Reducciones needed

Residuos

Lema 2.2.2. *Sean* $\mathsf{l}, \mathsf{r}, \mathsf{s} \in \mathsf{rpos}(t)$ *tal que* l *está a la izquierda de* r, $\mathsf{s} \not\leq \mathsf{l}$ *y* $\mathsf{s} : t \to_\beta t'$. *Entonces,* $\mathsf{l} \in \mathsf{rpos}(t')$ *y* l *está a la izquierda de* r' *para toda* $\mathsf{r}' \in \mathsf{r}/\mathsf{s}$.

Demostración. Notar que $\mathsf{s} \not\leq \mathsf{l}$ implica $\mathsf{l}/\mathsf{s} = \{\mathsf{l}\}$, por lo que es inmediato ver que $\mathsf{l} \in \mathsf{rpos}(t')$. Si $\mathsf{s} \not\leq \mathsf{r}$, entonces $\mathsf{r}/\mathsf{s} = \{\mathsf{r}\}$ y el resultado es inmediato. De lo contrario, $\mathsf{s} \leq \mathsf{r}$ implica $\mathsf{s} \leq \mathsf{r}'$ para todo $\mathsf{r}' \in \mathsf{r}/\mathsf{s}$ por definición. Como l está a la izquierda de r, se distinguen dos casos: $\mathsf{r} = \mathsf{lq}$ con $\mathsf{q} \neq \epsilon$; o bien $\mathsf{r} = \mathsf{p1q}$ y $\mathsf{l} = \mathsf{p0q}'$. Ambos casos implican que l y r comparten un prefijo común (l o p respectivamente). Sea p' este prefijo. De $\mathsf{s} \not\leq \mathsf{l}$ y $\mathsf{s} \leq \mathsf{r}$ se tiene que p' es un prefijo propio de s. Por lo tanto, es un prefijo propio de r' también, *i.e.* $\mathsf{r}' = \mathsf{lq}''$ con $\mathsf{q}'' \neq \epsilon$; o $\mathsf{r}' = \mathsf{p1q}''$ para todo $\mathsf{r}' \in \mathsf{r}/\mathsf{s}$. Luego, l está a la izquierda de r' para todo $\mathsf{r}' \in \mathsf{r}/\mathsf{s}$. □

Nociones de reducción needed

Lema 2.2.5. *El leftmost redex de cualquier término no en forma normal (head o weak-head) es needed (head needed o weak-head needed respectivamente).*

Demostración. Dado que la prueba existente en [BKKS87] no se extiende a formas normales weak-head, se da a continuación una prueba alternativa que no solo cubre el caso estándar de reducción needed, sino que además abarca los nuevos de reducción head needed y weak-head needed.

Sea t un término no en forma normal (forma normal head o forma normal weak-head respectivamente). Considerar $\rho : t \twoheadrightarrow_\beta t'$ tal que t' es una forma normal (forma normal head o forma normal weak-head respectivamente). Suponer en pos de una contradicción que el leftmost redex l de t no es usado en ρ. Por el Cor. 2.2.3, l es el leftmost redex de t'. Esto lleva a una contracción con le hecho de que t' es una forma normal (forma normal head o forma normal weak-head respectivamente), en particular porque el leftmost redex de un término no en forma normal head (forma normal weak-head respectivamente) es necesariamente el head redex (weak-head redex respectivamente). □

El sistema de tipos $\mathcal{N}$

Propiedades del sistema

Lema 2.3.5 (Sustitución). *Sean* $\pi_t \rhd_{\mathcal{N}} \Gamma; x : \{\!\{\sigma_i\}\!\}_{i\in I} \vdash t : \tau$ *y* $(\pi^i_u \rhd_{\mathcal{N}} \Delta_i \vdash u : \sigma_i)_{i\in I}$. *Luego, existe una derivación* $\pi_{\{x\setminus u\}t} \rhd_{\mathcal{N}} \Gamma +_{i\in I} \Delta_i \vdash \{x \setminus u\}t : \tau$ *tal que* $\mathsf{sz}(\pi_{\{x\setminus u\}t}) = \mathsf{sz}(\pi_t) +_{i\in I} \mathsf{sz}(\pi^i_u) - |I|$.

Demostración. Por inducción en t.

- $t = x$. Luego, $\pi_t = \mathsf{AX}(x, \tau)$ lo que implica $\Gamma = \emptyset$, $I = \{i_0\}$ y $\sigma_{i_0} = \tau$. Se concluye con $\pi_{\{x\setminus u\}t} = \pi^{i_0}_u$ dado que $\mathsf{sz}(\pi_t) = 1 = |I|$.
- $t = y \neq x$. Luego, $\pi_t = \mathsf{AX}(y, \tau)$ lo que implica $\Gamma = \{y : \{\!\{\tau\}\!\}\}$ e $I = \emptyset$. Se concluye con $\pi_{\{x\setminus u\}t} = \pi_t$ dado que $+_{i\in I}\mathsf{sz}(\pi^{i_0}_u) = 0 = |I|$.
- $t = t'\, s$. Luego, $\pi_t = \mathsf{APP}(\pi_{t'}, s, (\pi^j_s)_{j\in J})$ lo que implica $\Gamma = \Gamma' +_{j\in J} \Gamma_j$, $I = I \uplus (\biguplus_{j\in J} I_j)$ y $\tau = \{\!\{\tau_j\}\!\}_{j\in J} \to \tau'$ tal que

  $$\pi_{t'} \rhd_{\mathcal{N}} \Gamma'; x : \{\!\{\sigma_i\}\!\}_{i\in I'} \vdash t' : \{\!\{\tau_j\}\!\}_{j\in J} \to \tau'$$

  y

  $$(\pi^j_s \rhd_{\mathcal{N}} \Gamma_j; x : \{\!\{\sigma_i\}\!\}_{i\in I_j} \vdash s : \tau_j)_{j\in J}$$

  Por *h.i.* se tienen

  $$\pi_{\{x\setminus u\}t'} \rhd_{\mathcal{N}} \Gamma' +_{i\in I'} \Delta_i \vdash \{x \setminus u\}t' : \{\!\{\tau_j\}\!\}_{j\in J} \to \tau'$$

  y

  $$(\pi^j_{\{x\setminus u\}s} \rhd_{\mathcal{N}} \Gamma_j +_{i\in I_j} \Delta_i \vdash \{x \setminus u\}s : \tau_j)_{j\in J}$$

  con $\mathsf{sz}(\pi_{\{x\setminus u\}t'}) = \mathsf{sz}(\pi_{t'}) +_{i\in I'} \mathsf{sz}(\pi^i_u) - |I'|$ y $\mathsf{sz}(\pi^j_{\{x\setminus u\}s}) = \mathsf{sz}(\pi^j_s) +_{i\in I_j} \mathsf{sz}(\pi^i_u) - |I_j|$ para todo $j \in J$. Se concluye con

  $$\pi_{\{x\setminus u\}t} = \mathsf{APP}(\pi_{\{x\setminus u\}t'}, \{x \setminus u\}s, (\pi^j_{\{x\setminus u\}s})_{j\in J})$$

  dado que

  $$\begin{array}{rcl} \mathsf{sz}(\pi_{\{x\setminus u\}t}) & = & \pi_{\{x\setminus u\}t'} +_{j\in J} \pi^j_{\{x\setminus u\}s} + 1 \\ & = & (\mathsf{sz}(\pi_{t'}) +_{i\in I'} \mathsf{sz}(\pi^i_u) - |I'|) + (+_{j\in J}(\mathsf{sz}(\pi^j_s) +_{i\in I_j} \mathsf{sz}(\pi^i_u) - |I_j|)) + 1 \\ & = & (\mathsf{sz}(\pi_{t'}) +_{j\in J} \mathsf{sz}(\pi^j_s) + 1) +_{i\in I} \mathsf{sz}(\pi^i_u) - |I| \\ & = & \mathsf{sz}(\pi_t) +_{i\in I} \mathsf{sz}(\pi^i_u) - |I| \end{array}$$

- $t = \lambda y.t'$. Luego, $y \neq x$ y se tienen dos casos posibles:
  1. $\pi_t = \mathsf{VAL}(y, t')$, lo que implica $\Gamma = \emptyset$ e $I = \emptyset$. Se concluye con $\pi_{\{x\setminus u\}t} = \mathsf{VAL}(y, \{x \setminus u\}t')$.
  2. $\pi_t = \mathsf{ABS}(y, \pi_{t'})$, lo que implica $\Gamma = \Gamma' \setminus\!\setminus y$ y $\tau = \Gamma'(y) \to \tau'$ con la derivación $\pi_{t'} \rhd_{\mathcal{N}} \Gamma'; x : \{\!\{\sigma_i\}\!\}_{i\in I} \vdash t' : \tau'$. Por *h.i.* se tiene $\pi_{\{x\setminus u\}t'} \rhd_{\mathcal{N}} \Gamma' \vdash \{x \setminus u\}t : \tau'$ con $\mathsf{sz}(\pi_{\{x\setminus u\}t'}) = \mathsf{sz}(\pi_{t'}) +_{i\in I} \mathsf{sz}(\pi^i_u) - |I|$. Se concluye con $\pi_{\{x\setminus u\}t} = \mathsf{ABS}(y, \pi_{\{x\setminus u\}t'})$ dado que

     $$\begin{array}{rcl} \mathsf{sz}(\pi_{\{x\setminus u\}t}) & = & \pi_{\{x\setminus u\}t'} + 1 \\ & = & (\mathsf{sz}(\pi_{t'}) + 1) +_{i\in I} \mathsf{sz}(\pi^i_u) - |I| \\ & = & \mathsf{sz}(\pi_t) +_{i\in I} \mathsf{sz}(\pi^i_u) - |I| \end{array}$$

□

Teorema 2.3.6 (Weighted Subject Reduction). *Sea* $\pi_t \rhd_{\mathcal{N}} \Gamma \vdash t : \tau$. *Si* $\mathsf{r} : t \to_\beta t'$, *entonces existe* $\pi_{t'} \rhd_{\mathcal{N}} \Gamma \vdash t' : \tau$. *Más aún,*

1. *Si* $\mathsf{r} \in \mathsf{tpos}(\pi_t)$, *entonces* $\mathsf{sz}(\pi_t) > \mathsf{sz}(\pi_{t'})$.
2. *Si* $\mathsf{r} \notin \mathsf{tpos}(\pi_t)$, *entonces* $\mathsf{sz}(\pi_t) = \mathsf{sz}(\pi_{t'})$.

Demostración. Por inducción en $\mathsf{r} \in \mathsf{rpos}(t)$.

- $\mathsf{r} = \epsilon$. Luego, $t = (\lambda x.s)\, u$ y $t' = \{x \setminus u\}s$. Más aún, se tiene $\pi_t = \mathsf{APP}(\pi_{\lambda x.s}, u, (\pi_u^i)_{i\in I})$ con $\pi_s = \mathsf{ABS}(x, \pi_s)$. Se concluye pues, por Lem 2.3.5, existe $\pi_{t'}$ tal que $\mathsf{sz}(\pi_{t'}) = \mathsf{sz}(\pi_s) + _{i\in I}\, \mathsf{sz}(\pi_u^i) - |I| < \mathsf{sz}(\pi_s) + _{i\in I}\, \mathsf{sz}(\pi_u^i) + 1 = \mathsf{sz}(\pi_t)$ (notar que $\mathsf{r} \in \mathsf{tpos}(\pi_t)$ por Def. 2.3.1).
- $\mathsf{r} = 0\mathsf{r}'$. Luego, hay dos posibles casos:
  1. Si $t = s\, u$, entonces $t' = s'\, u$ con $\mathsf{r}' : s \to_\beta s'$. Más aún, se tiene $\pi_t = \mathsf{APP}(\pi_s, u, (\pi_u^i)_{i\in I})$. Por *h.i.* existe $\pi_{s'}$ tal que $\mathsf{sz}(\pi_s) > \mathsf{sz}(\pi_{s'})$ si $\mathsf{r}' \in \mathsf{tpos}(\pi_s)$ o $\mathsf{sz}(\pi_s) = \mathsf{sz}(\pi_{s'})$ si no. Se concluye luego con $\pi_{t'} = \mathsf{APP}(\pi_{s'}, u, (\pi_u^i)_{i\in I})$ dado que, por Def. 2.3.1 (3), $\mathsf{r} \in \mathsf{tpos}(\pi_t)$ sii $\mathsf{r}' \in \mathsf{tpos}(\pi_s)$.
  2. Si $t = \lambda x.s$, entonces $t' = \lambda x.s'$ con $\mathsf{r}' : s \to_\beta s'$. Se tienen dos casos para π_t: $\pi_t = \mathsf{VAL}(x, s)$ o $\pi_t = \mathsf{ABS}(x, \pi_s)$. En el primero se concluye con $\pi_{t'} = \mathsf{VAL}(x, s')$ dado que $\mathsf{r} \notin \mathsf{tpos}(\pi_t)$ y $\mathsf{sz}(\pi_t) = \mathsf{sz}(\pi_{t'})$. En el segundo, por *h.i.* existe $\pi_{s'}$ tal que $\mathsf{sz}(\pi_s) > \mathsf{sz}(\pi_{s'})$ si $\mathsf{r}' \in \mathsf{tpos}(\pi_s)$ o $\mathsf{sz}(\pi_s) = \mathsf{sz}(\pi_{s'})$ si no. Se concluye luego con $\pi_{t'} = \mathsf{ABS}(x, \pi_{s'})$ dado que, por Def. 2.3.1 (2), $\mathsf{r} \in \mathsf{tpos}(\pi_t)$ sii $\mathsf{r}' \in \mathsf{tpos}(\pi_s)$.
- $\mathsf{r} = 1\mathsf{r}'$. Luego, $t = s\, u$ y $t' = s\, u'$ con $\mathsf{r}' : u \to_\beta u'$. Más aún, se tiene $\pi_t = \mathsf{APP}(\pi_s, u, (\pi_u^i)_{i\in I})$. Por *h.i.* existen $(\pi_{u'}^i)_{i\in I}$ tal que $\mathsf{sz}(\pi_u^i) > \mathsf{sz}(\pi_{u'}^i)$ si $\mathsf{r}' \in \mathsf{tpos}(\pi_u^i)$ o $\mathsf{sz}(\pi_u^i) = \mathsf{sz}(\pi_{u'}^i)$ si no, para todo $i \in I$. Se concluye luego con $\pi_{t'} = \mathsf{APP}(\pi_s, u', (\pi_{u'}^i)_{i\in I})$ dado que, por Def. 2.3.1 (3), $\mathsf{r} \in \mathsf{tpos}(\pi_t)$ sii existe $k \in I$ tal que $\mathsf{r}' \in \mathsf{tpos}(\pi_u^k)$.

□

Lema 2.3.7 (Anti-sustitución). *Sea* $\pi_{\{x\setminus u\}t} \rhd_{\mathcal{N}} \Gamma \vdash \{x \setminus u\}t : \tau$. *Luego, existen derivaciones* $\pi_t \rhd_{\mathcal{N}} \Gamma'; x : \{\!\{\sigma_i\}\!\}_{i\in I} \vdash t : \tau$ *y* $(\pi_u^i \rhd_{\mathcal{N}} \Delta_i \vdash u : \sigma_i)_{i\in I}$ *tal que* $\Gamma = \Gamma' +_{i\in I} \Delta_i$.

Demostración. Por inducción en t.

- $t = x$. Luego, $\{x \setminus u\}t = u$ y se tiene $I = \{i_0\}$, $\sigma_{i_0} = \tau$. Se concluye con $\pi_t = \mathsf{AX}(x, \tau)$ y $\pi_u^{i_0} = \pi_{\{x\setminus u\}t}$.
- $t = y \neq x$. Luego, $\{x \setminus u\}t = y$ y se concluye con $\pi_t = \pi_{\{x\setminus u\}t}$ e $I = \emptyset$.
- $t = t'\, s$. Luego, $\{x \setminus u\}t = \{x \setminus u\}t'\, \{x \setminus u\}s$. Más aún, $\pi_{\{x\setminus u\}t} = \mathsf{APP}(\pi_{\{x\setminus u\}t'}, \{x \setminus u\}s, (\pi^j_{\{x\setminus u\}s})_{j\in J})$. Por *h.i.* existen $\pi_{t'}$ y $(\pi_s^j)_{j\in J}$ tal que $\mathsf{ctxt}(\pi_{\{x\setminus u\}t'}) = (\mathsf{ctxt}(\pi_{t'}) \setminus\!\setminus x) +_{i\in I'} \mathsf{ctxt}(\pi_u^i)$ y $\mathsf{ctxt}(\pi^j_{\{x\setminus u\}s}) = (\mathsf{ctxt}(\pi_s^j) \setminus\!\setminus x) +_{i\in I_j} \mathsf{ctxt}(\pi_u^i)$ para todo $j \in J$. Se concluye con $\pi_t = \mathsf{APP}(\pi_{t'}, s, (\pi_s^j)_{j\in J})$ e $I = I' \cup (\bigcup_{j\in J} I_j)$, dado que

$$\begin{array}{rcl}
\mathsf{ctxt}(\pi_{\{x\setminus u\}t}) & = & \mathsf{ctxt}(\pi_{\{x\setminus u\}t'}) +_{j\in J} \mathsf{ctxt}(\pi^j_{\{x\setminus u\}s}) \\
& = & ((\mathsf{ctxt}(\pi_{t'}) \setminus\!\setminus x) +_{i\in I'} \mathsf{ctxt}(\pi_u^i)) +_{j\in J} ((\mathsf{ctxt}(\pi_s^j) \setminus\!\setminus x) +_{i\in I_j} \mathsf{ctxt}(\pi_u^i)) \\
& = & ((\mathsf{ctxt}(\pi_{t'}) +_{j\in J} \mathsf{ctxt}(\pi_s^j)) \setminus\!\setminus x) +_{i\in I' \cup (\bigcup_{j\in J} I_j)} \mathsf{ctxt}(\pi_u^i) \\
& = & (\mathsf{ctxt}(\pi_t) \setminus\!\setminus x) +_{i\in I} \mathsf{ctxt}(\pi_u^i)
\end{array}$$

- $t = \lambda y.t'$. Luego, $\{x \setminus u\}t = \lambda y.\{x \setminus u\}t'$ y $\pi_{\{x\setminus u\}t} = \mathsf{ABS}(y, \pi_{\{x\setminus u\}t'})$. Por *h.i.* existe $\pi_{t'}$ tal que $\mathsf{ctxt}(\pi_{\{x\setminus u\}t'}) = (\mathsf{ctxt}(\pi_{t'}) \setminus\!\setminus x) +_{i\in I} \mathsf{ctxt}(\pi^i_u)$. Se concluye con $\pi_t = \mathsf{ABS}(y, \pi_{t'})$ dado que

$$\begin{array}{rcl} \mathsf{ctxt}(\pi_{\{x\setminus u\}t}) & = & \mathsf{ctxt}(\pi_{\{x\setminus u\}t'}) \setminus\!\setminus y \\ & = & ((\mathsf{ctxt}(\pi_{t'}) \setminus\!\setminus x) +_{i\in I} \mathsf{ctxt}(\pi^i_u)) \setminus\!\setminus y \\ & = & ((\mathsf{ctxt}(\pi_{t'}) \setminus\!\setminus y) \setminus\!\setminus x) +_{i\in I} \mathsf{ctxt}(\pi^i_u) \\ & = & (\mathsf{ctxt}(\pi_t) \setminus\!\setminus x) +_{i\in I} \mathsf{ctxt}(\pi^i_u) \end{array}$$

□

Teorema 2.3.8 (Subject Expansion). *Sea $\pi_{t'} \rhd_{\mathcal{N}} \Gamma \vdash t' : \tau$. Si $t \to_\beta t'$, entonces existe $\pi_t \rhd_{\mathcal{N}} \Gamma \vdash t : \tau$.*

Demostración. Por definición $t \to_\beta t'$ implica $t = \mathtt{C}\langle l\rangle$ y $t' = \mathtt{C}\langle r\rangle$ con $l \mapsto_\beta r$. La demostración es por inducción en $\mathtt{C}$.

- $\mathtt{C} = \Box$. Luego, $t = (\lambda x.s)\, u$ y $t' = \{x \setminus u\}s$. Por Lem. 2.3.7, existen π_s y $(\pi^i_u)_{i\in I}$ tal que $\mathsf{ctxt}(\pi_{t'}) = (\mathsf{ctxt}(\pi_s) \setminus\!\setminus x) +_{i\in I} \mathsf{ctxt}(\pi^i_u)$. Se concluye con $\pi_t = \mathsf{APP}(\mathsf{ABS}(x, \pi_s), u, (\pi^i_u)_{i\in I})$ dado que $\mathsf{ctxt}(\pi_t) = \mathsf{ctxt}(\pi_{t'})$ y $\mathsf{type}(\pi_t) = \mathsf{type}(\pi_{t'})$.
- $\mathtt{C} = \mathtt{C}'\, u$. Luego, $t = s\, u$ y $t' = s'\, u$ con $s \to_\beta s'$. Más aún, $\pi_{t'} = \mathsf{APP}(\pi_{s'}, u, (\pi^i_u)_{i\in I})$. Por *h.i.* existe π_s tal que $\mathsf{ctxt}(\pi_s) = \mathsf{ctxt}(\pi_{s'})$ y $\mathsf{type}(\pi_s) = \mathsf{type}(\pi_{s'})$. Se concluye con $\pi_t = \mathsf{APP}(\pi_s, u, (\pi^i_u)_{i\in I})$.
- $\mathtt{C} = s\,\mathtt{C}'$. Luego, $t = s\, u$ y $t' = s\, u'$ con $u \to_\beta u'$. Más aún, $\pi_{t'} = \mathsf{APP}(\pi_s, u', (\pi^i_{u'})_{i\in I})$. Por *h.i.* existen $(\pi^i_u)_{i\in I}$ tal que $\mathsf{ctxt}(\pi^i_u) = \mathsf{ctxt}(\pi^i_{u'})$ y $\mathsf{type}(\pi^i_u) = \mathsf{type}(\pi^i_{u'})$ para todo $i \in I$. Se concluye entonces con $\pi_t = \mathsf{APP}(\pi_s, u, (\pi^i_u)_{i\in I})$.
- $\mathtt{C} = \lambda x.\mathtt{C}'$. Luego, $t = \lambda x.s$ y $t' = \lambda x.s'$ con $s \to_\beta s'$. se tienen dos casos para $\pi_{t'}$: $\pi_{t'} = \mathsf{VAL}(x, s')$ o $\pi_{t'} = \mathsf{ABS}(x, \pi_{s'})$. En el primero se concluye con $\pi_t = \mathsf{VAL}(x, s)$. En el segundo, por *h.i.* existe π_s tal que $\mathsf{ctxt}(\pi_s) = \mathsf{ctxt}(\pi_{s'})$ y $\mathsf{type}(\pi_s) = \mathsf{type}(\pi_{s'})$. Se concluye con $\pi_t = \mathsf{ABS}(x, \pi_s)$.

□

Sustitución y reducción en derivaciones

Lema 2.4.2. *Sean π_t y $(\pi^i_u)_{i\in I}$ derivaciones tal que $\{x \setminus (\pi^i_u)_{i\in I}\}\pi_t$ está definida, y $\mathsf{p} \in \mathsf{pos}(t)$. Entonces,*

1. *$\mathsf{p} \in \mathsf{tpos}(\pi_t)$ sii $\mathsf{p} \in \mathsf{tpos}(\{x \setminus (\pi^i_u)_{i\in I}\}\pi_t)$.*
2. *$\mathsf{q} \in \mathsf{tpos}(\pi^k_u)$ para algún $k \in I$ sii existe $\mathsf{p}' \in \mathsf{tpos}(\pi_t)$ tal que $t|_{\mathsf{p}'} = x$ y $\mathsf{p}'\mathsf{q} \in \mathsf{tpos}(\{x \setminus (\pi^i_u)_{i\in I}\}\pi_t)$.*

Demostración. Se demuestra por separado cada caso.

1. Por inducción en π_t.
   - $\pi_t = \mathsf{AX}(y, \tau)$. Luego, $t = y$ y $\mathsf{pos}(t) = \{\epsilon\}$ por Def. 2.4.1 (1). El resultado es inmediato pues $\epsilon \in \mathsf{tpos}(\pi)$ para toda derivación π.
   - $\pi_t = \mathsf{VAL}(y, t')$. Por Def. 2.4.1 (2), $\{x \setminus (\pi^i_u)_{i\in I}\}\pi_t = \mathsf{VAL}(y, \{x \setminus u\}t')$. Más aún, por Def. 2.3.1 (1), $\mathsf{tpos}(\pi_t) = \{\epsilon\} = \mathsf{tpos}(\{x \setminus (\pi^i_u)_{i\in I}\}\pi_t)$. Luego, el resultado es inmediato.
   - $\pi_t = \mathsf{ABS}(y, \pi_{t'})$. Luego, $t = \lambda y.t'$. Por Def. 2.3.1 (2), se tienen dos casos para p:

   a) $\mathsf{p} = \epsilon$. El resultado es inmediato pues $\epsilon \in \mathsf{tpos}(\pi)$ para toda derivación π.
   b) $\mathsf{p} = \mathsf{0p}'$. Luego, $\mathsf{p}' \in \mathsf{pos}(\pi_{t'})$. Más aún, por Def. 2.4.1 (3), se tiene $\{x \setminus (\pi_u^i)_{i\in I}\}\pi_t = \mathsf{ABS}(y, \{x \setminus (\pi_u^i)_{i\in I}\}\pi_{t'})$. Por *h.i.* se tiene $\mathsf{p}' \in \mathsf{tpos}(\pi_{t'})$ sii $\mathsf{p}' \in \mathsf{tpos}(\{x \setminus (\pi_u^i)_{i\in I}\}\pi_{t'})$. Se concluye por Def. 2.3.1 (2), dado que $\mathsf{subj}(\{x \setminus (\pi_u^i)_{i\in I}\}\pi_t) = \lambda y.\{x \setminus u\}t'$.

- $\pi_t = \mathsf{APP}(\pi_{t'}, s, (\pi_s^j)_{j\in J})$. Luego, $t = t'\, s$. Por Def. 2.3.1 (3), se tienen tres casos para p:
   a) $\mathsf{p} = \epsilon$. El resultado es inmediato pues $\epsilon \in \mathsf{tpos}(\pi)$ para toda derivación π.
   b) $\mathsf{p} = \mathsf{0p}'$. Luego, $\mathsf{p}' \in \mathsf{pos}(\pi_{t'})$. Más aún, por Def. 2.4.1 (4), se tiene $\{x \setminus (\pi_u^i)_{i\in I}\}\pi_t = \mathsf{APP}(\{x \setminus (\pi_u^i)_{i\in I'}\}\pi_{t'}, \{x \setminus u\}s, (\{x \setminus (\pi_u^i)_{i\in I_j}\}\pi_s^j)_{j\in J})$ con $I = I' \uplus (\biguplus_{j\in J} I_j)$. Luego, por *h.i.* $\mathsf{p}' \in \mathsf{tpos}(\pi_{t'})$ sii $\mathsf{p}' \in \mathsf{tpos}(\{x \setminus (\pi_u^i)_{i\in I'}\}\pi_{t'})$. Se concluye por Def. 2.3.1 (3), dado que $\mathsf{subj}(\{x \setminus (\pi_u^i)_{i\in I}\}\pi_t) = \{x \setminus u\}t'\,\{x \setminus u\}s$.
   c) $\mathsf{p} = \mathsf{1p}'$. Luego, $(\mathsf{p}' \in \mathsf{pos}(\pi_s^j))_{j\in J}$. Por Def. 2.4.1 (4), existe una partición tal que $I = I' \uplus (\biguplus_{j\in J} I_j)$ y $\{x \setminus (\pi_u^i)_{i\in I}\}\pi_t = \mathsf{APP}(\{x \setminus (\pi_u^i)_{i\in I'}\}\pi_{t'}, \{x \setminus u\}s, (\{x \setminus (\pi_u^i)_{i\in I_j}\}\pi_s^j)_{j\in J})$. Por *h.i.* se tiene $\mathsf{p}' \in \mathsf{tpos}(\pi_s^j)$ sii $\mathsf{p}' \in \mathsf{tpos}(\{x \setminus (\pi_u^i)_{i\in I_j}\}\pi_s^j)$ para todo $j \in J$. Se concluye por Def. 2.3.1 (3), pues $\mathsf{subj}(\{x \setminus (\pi_u^i)_{i\in I}\}\pi_t) = \{x \setminus u\}t'\,\{x \setminus u\}s$.

2. Por inducción en π_t.

- $\pi_t = \mathsf{AX}(y, \tau)$. Si $y \neq x$, por Def. 2.4.1 (1), $\{x \setminus (\pi_u^i)_{i\in I}\}\pi_t = \mathsf{AX}(y, \tau)$ y el resultado es inmediato pues $I = \emptyset$. Si no (*i.e.* $y = x$), por Def. 2.4.1 (1) se tiene $I = \{i_0\}$ y $\{x \setminus (\pi_u^i)_{i\in I}\}\pi_t = \pi_u^{i_0}$. Luego, se concluye con $\mathsf{p}' = \epsilon$.
- $\pi_t = \mathsf{VAL}(y, t')$. El resultado es inmediato pues, por Def. 2.4.1 (1), se tiene $I = \emptyset$.
- $\pi_t = \mathsf{ABS}(y, \pi_{t'})$. Luego, $t = \lambda y.t'$. Más aún, por Def. 2.4.1 (3), se tiene $\{x \setminus (\pi_u^i)_{i\in I}\}\pi_t = \mathsf{ABS}(y, \{x \setminus (\pi_u^i)_{i\in I}\}\pi_{t'})$. Por *h.i.* $\mathsf{q} \in \mathsf{tpos}(\pi_u^k)$ para algún $k \in I$ sii existe $\mathsf{p}'' \in \mathsf{tpos}(\pi_{t'})$ tal que $t'|_{\mathsf{p}''} = x$ y $\mathsf{p}''\mathsf{q}' \in \mathsf{tpos}(\{x \setminus (\pi_u^i)_{i\in I}\}\pi_{t'})$. Se concluye por Def. 2.3.1 (2), con $\mathsf{p}' = \mathsf{0p}''$.
- $\pi_t = \mathsf{APP}(\pi_s, u, (\pi_u^i)_{i\in I})$. Luego, $t = t'\, s$ y, por Def. 2.4.1 (4), existe una partición tal que $I = I' \uplus (\biguplus_{j\in J} I_j)$ y $\{x \setminus (\pi_u^i)_{i\in I}\}\pi_t = \mathsf{APP}(\{x \setminus (\pi_u^i)_{i\in I'}\}\pi_{t'}, \{x \setminus u\}s, (\{x \setminus (\pi_u^i)_{i\in I_j}\}\pi_s^j)_{j\in J})$. Se tienen dos casos para $k \in I$:
   a) $k \in I'$. Por *h.i.* $\mathsf{q} \in \mathsf{tpos}(\pi_u^k)$ sii existe $\mathsf{p}'' \in \mathsf{tpos}(\pi_{t'})$ tal que $t'|_{\mathsf{p}''} = x$ y $\mathsf{p}''\mathsf{q}' \in \mathsf{tpos}(\{x \setminus (\pi_u^i)_{i\in I'}\}\pi_{t'})$. Se concluye por Def. 2.3.1 (3), con $\mathsf{p}' = \mathsf{0p}''$.
   b) $k \in I_l$ para algún $l \in J$. Por *h.i.* $\mathsf{q} \in \mathsf{tpos}(\pi_u^k)$ sii existe $\mathsf{p}'' \in \mathsf{tpos}(\pi_s^l)$ tal que $s|_{\mathsf{p}''} = x$ y $\mathsf{p}''\mathsf{q}' \in \mathsf{tpos}(\{x \setminus (\pi_u^i)_{i\in I_l}\}\pi_s^l)$. Se concluye por Def. 2.3.1 (3), con $\mathsf{p}' = \mathsf{1p}''$.

□

Redexes tipados vs. weak-head needed

Lema 2.5.1. *Sean* $\mathsf{r} : \pi_t \to_\beta \pi_{t'}$ *y* $\mathsf{p} \in \mathsf{pos}(t)$ *tal que* $\mathsf{p} \neq \mathsf{r}$ *y* $\mathsf{p} \neq \mathsf{0r}$. *Entonces,* $\mathsf{p} \in \mathsf{tpos}(\pi_t)$ *sii existe* $\mathsf{p}' \in \mathsf{p}/\mathsf{r}$ *tal que* $\mathsf{p}' \in \mathsf{tpos}(\pi_{t'})$.

Demostración. Si $\mathsf{p} = \epsilon$ el resultado es inmediato dado que, por Def. 2.2.1, $\epsilon/\mathsf{r} = \{\epsilon\}$ y, por Def. 2.3.1, $\epsilon \in \mathsf{tpos}(\pi)$ para toda posible derivación π. Luego, se asume $\mathsf{p} \neq \epsilon$. La demostración es por inducción en r.

- $\mathsf{r} = \epsilon$. Luego $\mathsf{r} \leq \mathsf{p}$ y $\mathsf{r} \in \mathsf{rpos}(t) \cap \mathsf{tpos}(\pi_t)$. Más aún, $t = t|_{\mathsf{r}} = (\lambda x.s)\, u$ y

$$\pi_t = \frac{\dfrac{\pi_s}{\Gamma' \vdash \lambda x.s : \{\!\{\sigma_i\}\!\}_{i\in I} \to \tau} \quad (\pi_u^i)_{i\in I}}{\Gamma \vdash (\lambda x.s)\, u : \tau} \to_\beta \{x \setminus (\pi_u^i)_{i\in I}\}\pi_s = \pi_{t'}$$

  con $t' = \{x \setminus u\}s$. Entonces, hay dos posibilidades para p.

1. $\mathsf{p} = 00\mathsf{p}'$. Luego, $\mathsf{p}/\epsilon = \{\mathsf{p}'\}$. Por Def. 2.3.1, $\mathsf{p} \in \mathsf{tpos}(\pi_t)$ sii $\mathsf{p}' \in \mathsf{tpos}(\pi_s)$. Se concluye por Lem. 2.4.2 (1), pues $\mathsf{p}' \in \mathsf{tpos}(\pi_s)$ sii $\mathsf{p}' \in \mathsf{tpos}(\pi_{t'})$. Notar que $\mathsf{p}' \in \mathsf{pos}(s)$.
2. $\mathsf{p} = 1\mathsf{p}'$. Luego, $\mathsf{p}/\epsilon = \{\mathsf{q}\mathsf{p}'' \mid s|_{\mathsf{q}} = x\}$. Por Def. 2.3.1, $\mathsf{p} \in \mathsf{tpos}(\pi_t)$ sii $\mathsf{p}'' \in \mathsf{tpos}(\pi_u^k)$ para algún $k \in I$. Se concluye por Lem. 2.4.2 (2), pues $\mathsf{p}'' \in \mathsf{tpos}(\pi_u^k)$ sii existe $q \in \mathsf{tpos}(\pi_s)$ tal que $s|_{\mathsf{q}} = x$ y $\mathsf{p}' = \mathsf{q}\mathsf{p}'' \in \mathsf{tpos}(\pi_{t'})$.

- $\mathsf{r} = 0\mathsf{r}'$. Luego, se analiza la forma de t:
  - $t = s\,u$. Entonces, $t' = s'\,u$ con $\mathsf{r}' : s \to_\beta s'$. Más aún,

$$\pi_t = \frac{\pi_s \quad (\pi_u^i)_{i\in I}}{\Gamma \vdash s\,u : \tau} \quad \to_\beta \quad \frac{\pi_{s'} \quad (\pi_u^i)_{i\in I}}{\Gamma \vdash s'\,u : \tau} = \pi_{t'}$$

    con $\pi_s \to_\beta \pi_{s'}$. Luego, se tiene dos posibilidades para p:
    1. $\mathsf{p} = 0\mathsf{q}$. Por Def. 2.3.1, $\mathsf{p} \in \mathsf{tpos}(\pi_t)$ sii $\mathsf{q} \in \mathsf{tpos}(\pi_s)$. Como $\mathsf{p} \neq \mathsf{r}$ se tiene $\mathsf{q} \neq \mathsf{r}'$. Entonces, por *h.i.* $\mathsf{q} \in \mathsf{tpos}(\pi_s)$ sii existe $\mathsf{q}' \in \mathsf{q}/\mathsf{r}'$ tal que $\mathsf{q}' \in \mathsf{tpos}(\pi_{s'})$. Luego, se concluye por Def. 2.3.1, $\mathsf{p} \in \mathsf{tpos}(\pi_t)$ sii existe $\mathsf{p}' \in \mathsf{p}/\mathsf{r}$ tal que $\mathsf{p}' \in \mathsf{tpos}(\pi_{t'})$ (*i.e.* $\mathsf{p}' = 0\mathsf{q}'$).
    2. $\mathsf{p} = 1\mathsf{q}$. Luego, $\mathsf{r} \not\leq \mathsf{p}$ y $\mathsf{p}/\mathsf{r} = \{\mathsf{p}\}$ (*i.e.* $\mathsf{p}' = \mathsf{p}$). Más aún, por Def. 2.3.1, $\mathsf{p} \in \mathsf{tpos}(\pi_t)$ sii existe $\mathsf{q} \in \mathsf{tpos}(\pi_u^k)$ para algún $k \in I$ sii $\mathsf{p}' \in \mathsf{tpos}(\pi_{t'})$.
  - $t = \lambda x.s$. Luego, $t' = \lambda x.s'$ con $\mathsf{r}' : s \to_\beta s'$. Si $\mathsf{rule}(\pi_t) = (\textsc{val})$, el resultado es inmediato pues $\mathsf{tpos}(\pi_t) = \{\epsilon\} = \mathsf{tpos}(\pi_{t'})$. Si no,

$$\pi_t = \frac{\pi_s}{\Gamma \vdash \lambda x.s : \tau} \quad \to_\beta \quad \frac{\pi_{s'}}{\Gamma \vdash \lambda x.s' : \tau} = \pi_{t'}$$

    con $\pi_s \to_\beta \pi_{s'}$. Entonces, $\mathsf{p} = 0\mathsf{q}$ y, por Def. 2.3.1, $\mathsf{p} \in \mathsf{tpos}(\pi_t)$ sii $\mathsf{q} \in \mathsf{tpos}(\pi_s)$. Por *h.i.* $\mathsf{q} \in \mathsf{tpos}(\pi_s)$ sii existe $\mathsf{q}' \in \mathsf{q}/\mathsf{r}'$ tal que $\mathsf{q}' \in \mathsf{tpos}(\pi_{s'})$. Por lo tanto, se concluye como en el caso anterior.

- $\mathsf{r} = 1\mathsf{r}'$. Este caso es simétrico al presentado anteriormente. Se tienen $t = s\,u$ y $t' = s\,u'$ con $\mathsf{r}' : u \to_\beta u'$ y

$$\pi_t = \frac{\pi_s \quad (\pi_u^i)_{i\in I}}{\Gamma \vdash s\,u : \tau} \quad \to_\beta \quad \frac{\pi_s \quad (\pi_{u'}^i)_{i\in I}}{\Gamma \vdash s\,u' : \tau} = \pi_{t'}$$

  con $(\pi_u^i \to_\beta \pi_{u'}^i)_{i\in I}$. Luego, si $\mathsf{p} = 0\mathsf{q}$ (*i.e.* $\mathsf{r} \not\leq \mathsf{p}$) se concluye por Def. 2.2.1 con $\mathsf{p}/\mathsf{r} = \{\mathsf{p}\}$. Si no, $\mathsf{p} = 1\mathsf{q}$ y, por Def. 2.3.1, $\mathsf{p} \in \mathsf{tpos}(\pi_t)$ sii $\mathsf{q} \in \mathsf{tpos}(\pi_u^k)$ para algún $k \in I$. Finalmente, se concluye por *h.i.* y Def. 2.3.1.

□

Redexes weak-head needed son tipados

Lema 2.5.2. *Sean* $\rho : \pi_t \twoheadrightarrow_\beta \pi_{t'}$ *y* $\mathsf{p} \in \mathsf{pos}(t)$. *Si existe* $\mathsf{p}' \in \mathsf{p}/\rho$ *tal que* $\mathsf{p}' \in \mathsf{tpos}(\pi_{t'})$, *entonces* $\mathsf{p} \in \mathsf{tpos}(\pi_t)$.

Demostración. Por inducción en ρ.

- $\rho = \mathit{nil}$. La propiedad se satisface vacuamente ($\pi_t = \pi_{t'}$).
- $\rho = \mathsf{r}\rho'$. Luego, se tienen $\mathsf{r} : \pi_t \to_\beta \pi_s$ y $\rho' : \pi_s \twoheadrightarrow_\beta \pi_{t'}$. Sea $\mathsf{q} \in \mathsf{p}/\mathsf{r}$ (*i.e.* $\mathsf{q} \in \mathsf{pos}(s)$). Por *h.i.* si existe $\mathsf{p}' \in \mathsf{q}/\rho'$ tal que $\mathsf{p}' \in \mathsf{tpos}(\pi_{t'})$, entonces $\mathsf{q} \in \mathsf{tpos}(\pi_s)$. Se concluye por Lem. 2.5.1, pues $\mathsf{q} \in \mathsf{tpos}(\pi_s)$ implica a su vez $\mathsf{p} \in \mathsf{tpos}(\pi_t)$.

□

Redexes tipados principalmente son weak-head needed

Lema 2.5.5. *Sean π_t una derivación con sujeto t, $\mathsf{r} \in \mathsf{rpos}(t) \cap \mathsf{tpos}(\pi_t)$ y $\rho : \pi_t \twoheadrightarrow_\beta \pi_{t'}$. Si $\pi_{t'}$ es principal normal, entonces ρ usa r.*

Demostración. Por inducción en ρ.

- $\rho = \mathit{nil}$. La propiedad se satisface vacuamente ($\pi_t = \pi_{t'}$).
- $\rho = \mathsf{r}'\rho'$. Si $\mathsf{r}' = \mathsf{r}$ el resultado es inmediato. Si no, por Lem. 2.5.1, existe $\mathsf{r}'' \in \mathsf{r}/\mathsf{r}'$ tal que $\mathsf{r}'' \in \mathsf{tpos}(\pi_s)$. Más aún, $\mathsf{r} \in \mathsf{rpos}(t)$ implica $\mathsf{r}'' \in \mathsf{rpos}(s)$. Luego, por *h.i.* se tiene que ρ' usa r'' y, por lo tanto, ρ usa r.

□

Lema 2.5.6. *Sean t un término weak-head normalizante y π_t un tipado principal. Luego, toda secuencia de reducción tipada leftmost desde π_t es weak-head needed.*

Demostración. Sea $\rho : \pi_t \twoheadrightarrow_\beta \pi_{t'}$ la secuencia de reducción tipada leftmost con $\pi_{t'}$ normal. Se tiene dos casos para ρ:

- $\rho = \mathit{nil}$. La propiedad se satisface vacuamente ($\pi_t = \pi_{t'}$).
- $\rho \neq \mathit{nil}$. Suponer $t \in \mathcal{WHNF}_\beta$. Por Def. 2.5.4, π_t es principal normal, por lo que no tiene redexes tipados. Esto contradice $\rho \neq \mathit{nil}$. Luego, t tiene un weak-head redex r (*i.e.* $t \notin \mathcal{WHNF}_\beta$) que es leftmost por definición. Más aún, r es tanto tipado (Nota 2.3.2) como weak-head needed (Lem. 2.2.5).

□

Normalización weak-head needed

Lema 2.6.1. *Sea $\pi_t \rhd_{\mathcal{N}} \Gamma \vdash t : \tau$. Luego, π_t normal implica $t \in \mathcal{WHNF}_\beta$.*

Demostración. Por inducción en π_t analizando la última regla aplicada.

- (AX). Luego, $t = x \in \mathcal{WHNF}_\beta$.
- (VAL). Luego, $t = \lambda x.t' \in \mathcal{WHNF}_\beta$.
- (ABS). Luego, $t = \lambda x.t' \in \mathcal{WHNF}_\beta$.
- (APP). Luego, $t = t'\,u$ con $\Gamma = \Gamma' +_{i\in I} \Delta_i$, $\pi_{t'} \rhd_{\mathcal{N}} \Gamma' \vdash t' : \{\!\{\sigma_i\}\!\}_{i\in i} \to \tau$ y $(\Delta_i \vdash u : \sigma_i)_{i\in I}$ para algún I. π_t normal implica $\pi_{t'}$ normal también. Por lo tanto, por *h.i.* se tiene $t' \in \mathcal{WHNF}_\beta$. Más aún, $t' \neq \lambda x.s$ pues $\epsilon \in \mathsf{tpos}(\pi_t)$ y π_t es normal. Luego, $t' = x\,t_1 \ldots t_n$ con $n \geq 0$. Se concluye con $t = x\,t_1 \ldots t_n\,u \in \mathcal{WHNF}_\beta$.

□

Apéndice B

Pruebas: Calculus of Applicative Patterns

Sintaxis de CAP

En primer lugar se introducen una serie de definiciones y resultados auxiliares necesarios para probar la confluencia del cálculo.

Lema B.0.1. *Sean* $\{\!\{p \setminus u\}\!\}$ *un match decidido y* σ *una sustitución tal que* p `evita` σ. *Luego,* $\{\!\{p \setminus \sigma u\}\!\}$ *es un match decidido y* $\sigma \circ \{\!\{p \setminus u\}\!\} = \{\!\{p \setminus \sigma u\}\!\} \circ \sigma$. *Más aún,* $\{\!\{p \setminus u\}\!\} = \texttt{fail}$ *sii* $\{\!\{p \setminus \sigma u\}\!\} = \texttt{fail}$.

Demostración. Se analizan por separado los cosos $\{\!\{p \setminus u\}\!\} = \rho$ y $\{\!\{p \setminus u\}\!\} = \texttt{fail}$.

- Si $\{\!\{p \setminus u\}\!\} = \rho$. Se procede por inducción en p.
  - $p = x$. Luego, $\rho = \{x \setminus u\}$ y $\{\!\{p \setminus \sigma u\}\!\} = \{x \setminus \sigma u\}$. Sea $y \in \mathbb{V}$:
    - Si $y \in \mathsf{fv}(p)$ (*i.e.* $y = x$), entonces $\sigma y = y$, dado que p `evita` σ, y $(\sigma \circ \{y \setminus u\})y = \sigma u = \{y \setminus \sigma u\}y = \{y \setminus \sigma u\}\sigma y$.
    - Si $y \notin \mathsf{fv}(p)$ (*i.e.* $y \neq x$), nuevamente dado que p `evita` σ se tiene $(\sigma \circ \{x \setminus u\})y = \sigma y = (\{x \setminus \sigma u\} \circ \sigma)y$.

    En ambos casos se concluye $\sigma \circ \{\!\{p \setminus u\}\!\} = \{\!\{p \setminus \sigma u\}\!\} \circ \sigma$.
  - $p = \mathtt{c}$. Luego, $u = \mathtt{c}$ y $\sigma u = \mathtt{c}$, por lo que el resultado es inmediato.
  - $p = p_0\, p_1$. Luego, $u = u_0\, u_1$ y $\rho = \{\!\{p_0 \setminus u_0\}\!\} \uplus \{\!\{p_1 \setminus u_1\}\!\} = \rho_0 \uplus \rho_1$. Por *h.i.* se tiene $(\{\!\{p_i \setminus \sigma u_i\}\!\} = \rho_i')_{i<2}$ con $(\sigma \circ \rho_i = \rho_i' \circ \sigma)_{i<2}$. Luego, $\{\!\{p \setminus \sigma u\}\!\} = \rho_0' \uplus \rho_1'$ y se concluye dado que $\sigma \circ \{\!\{p \setminus u\}\!\} = (\sigma \circ \rho_0) \uplus (\sigma \circ \rho_1) = (\rho_0' \circ \sigma) \uplus (\rho_1' \circ \sigma) = \{\!\{p \setminus \sigma u\}\!\} \circ \sigma$.
- Si $\{\!\{p \setminus u\}\!\} = \texttt{fail}$. Luego $u \in \mathbb{M}_{\mathsf{CAP}}$. Se procede por inducción en p.
  - $p = x$. Este caso no aplica para un match fallido.
  - $p = \mathtt{c}$. Luego, $u \neq \mathtt{c}$. Es inmediato ver que $\sigma u \in \mathbb{M}_{\mathsf{CAP}}$ y $\sigma u \neq \mathtt{c}$. Por lo tanto, $\{\!\{p \setminus \sigma u\}\!\} = \texttt{fail}$ y se concluye por definición de composición del match.
  - $p = p_0\, p_1$. Se consideran dos casos según la forma de u:
    1. $u = u_0\, u_1$. Luego, el mismatch es interno pues se asumen patrones lineales (*cf.* Sec. 3.2), por lo que la falla no se produce en la unión disjunta. Es decir, se tiene $\{\!\{p \setminus u\}\!\} = \{\!\{p_0 \setminus u_0\}\!\} \uplus \{\!\{p_1 \setminus u_1\}\!\}$ con $\{\!\{p_0 \setminus u_0\}\!\} = \texttt{fail}$ o $\{\!\{p_1 \setminus u_1\}\!\} = \texttt{fail}$. Asumir el primer caso

(el análisis para el segundo es simétrico). Por *h.i.* se tiene $\{\!\{p_0 \setminus \sigma u_0\}\!\} = \texttt{fail}$. Por lo tanto, $\{\!\{p \setminus \sigma u\}\!\} = \{\!\{p_0 \setminus \sigma u_0\}\!\} \uplus \{\!\{p_1 \setminus \sigma u_1\}\!\} = \texttt{fail}$ también. Luego, se concluye por definición de composición del match.

2. $u \neq u_0\, u_1$. Como en el caso de $p = \texttt{c}$, es inmediato ver que $\sigma u \in \mathbb{M}_{\texttt{CAP}}$ y $\sigma u \neq u'_0\, u'_1$. Luego, $\{\!\{p \setminus \sigma p\}\!\} = \texttt{fail}$ y se concluye por definición de composición del match.

□

La relación de reducción $\twoheadrightarrow_{\texttt{CAP}}$ se extiende a sustituciones de modo que $\sigma \twoheadrightarrow_{\texttt{CAP}} \sigma'$ sii $\mathsf{dom}(\sigma) = \mathsf{dom}(\sigma')$ y $\sigma(x) \twoheadrightarrow_{\texttt{CAP}} \sigma'(x)$ para toda $x \in \mathsf{dom}(\sigma)$.

Lema B.0.2. *Sean $t \in \mathbb{T}_{CAP}$ y σ una sustitución. Si $t \twoheadrightarrow_{\texttt{CAP}} t'$ y $\sigma \twoheadrightarrow_{\texttt{CAP}} \sigma'$, entonces $\sigma t \twoheadrightarrow_{\texttt{CAP}} \sigma' t'$.*

Demostración. Por inducción en el número de pasos de reducción m en la secuencia $t \twoheadrightarrow_{\texttt{CAP}} t'$.

- $m = 0$. Luego, $t = t'$ por lo que basta ver que $\sigma t \twoheadrightarrow_{\texttt{CAP}} \sigma t'$. Se procede por inducción en t.
  - $t = x$. Si $x \in \mathsf{dom}(\sigma)$, entonces $\sigma t = \sigma(x) \twoheadrightarrow_{\texttt{CAP}} \sigma'(x) = \sigma' t$ por hipótesis. Si no (*i.e.* $x \notin \mathsf{dom}(\sigma)$), se tiene $\sigma t = x \twoheadrightarrow_{\texttt{CAP}} x = \sigma' t$ por reflexividad.
  - $t = \texttt{c}$. Luego, $\sigma t = \texttt{c} \twoheadrightarrow_{\texttt{CAP}} \texttt{c} = \sigma' t$ por reflexividad.
  - $t = r\, u$. Luego, $\sigma t = \sigma r\, \sigma u$. Por *h.i.* se tienen $\sigma r \twoheadrightarrow_{\texttt{CAP}} \sigma' r$ y $\sigma u \twoheadrightarrow_{\texttt{CAP}} \sigma' u$. Por lo tanto, $\sigma t = \sigma r\, \sigma u \twoheadrightarrow_{\texttt{CAP}} \sigma' r\, \sigma u \twoheadrightarrow_{\texttt{CAP}} \sigma' r\, \sigma' u = \sigma' t$.
  - $t = (p_i \to s_i)_{i<n}$. Sin pérdida de generalidad se asumen $(p_i\, \texttt{evita}\, \sigma)_{i<n}$. Luego, $\sigma t = (p_i \to \sigma s_i)_{i<n}$. Por *h.i.* de tienen $(\sigma s_i \twoheadrightarrow_{\texttt{CAP}} \sigma' s_i)_{i<n}$. Por lo tanto, se concluye como en el caso anterior, $\sigma t = (p_i \to \sigma s_i)_{i<n} \twoheadrightarrow_{\texttt{CAP}} (p_i \to \sigma' s_i)_{i<n} = \sigma' t$.
- $m > 0$. Luego, $t \to_{\texttt{CAP}} t'' \twoheadrightarrow_{\texttt{CAP}} t'$. Por *h.i.* se tiene $\sigma t'' \twoheadrightarrow_{\texttt{CAP}} \sigma' t'$. Basta ver entonces que $\sigma t \to_{\texttt{CAP}} \sigma t''$. Por definición $t = \texttt{C}\langle l \rangle$ y $t'' = \texttt{C}\langle r \rangle$ con $l \mapsto_{\texttt{CAP}} r$. Se procede por inducción en $\texttt{C}$.
  - $\texttt{C} = \Box$. Luego, $t = (p_i \to s_i)_{i<n}\, u$ y $t'' = \rho s_k$ con $(\{\!\{p_i \setminus u\}\!\} = \texttt{fail})_{i<k}$ y $\{\!\{p_k \setminus u\}\!\} = \rho$ para algún $k < n$. Sin pérdida de generalidad se asumen $(p_i\, \texttt{evita}\, \sigma)_{i<n}$. Luego, $\sigma t = (p_i \to \sigma s_i)_{i<n}\, \sigma u$. Por Lem B.0.1, $\{\!\{p_k \setminus \sigma u\}\!\} = \rho'$ con $\sigma \circ \rho = \rho' \circ \sigma$. Más aún, también por Lem. B.0.1, se tienen $(\{\!\{p_i \setminus \sigma u\}\!\} = \texttt{fail})_{i<k}$. Luego, $\sigma t = (p_i \to \sigma s_i)_{i<n}\, \sigma u \mapsto_{\texttt{CAP}} \rho' \sigma s_k = \sigma \rho s_k = \sigma t''$.
  - $\texttt{C} = \texttt{C}'\, u$. Luego, $t = r\, u$ y $t'' = r'\, u$ con $r \to_{\texttt{CAP}} r'$. Por *h.i.* se tiene $\sigma r \to_{\texttt{CAP}} \sigma r'$. Por lo tanto, $\sigma t = \sigma r\, \sigma u \to_{\texttt{CAP}} \sigma r'\, \sigma u = \sigma t''$.
  - $\texttt{C} = r\, \texttt{C}'$. Luego, $t = r\, u$ y $t'' = r\, u'$ con $u \to_{\texttt{CAP}} u'$. Por *h.i.* se tiene $\sigma u \to_{\texttt{CAP}} \sigma u'$. Por lo tanto, $\sigma t = \sigma r\, \sigma u \to_{\texttt{CAP}} \sigma r\, \sigma u' = \sigma t''$.
  - $\texttt{C} = (p_i \to \texttt{C}_k)_{i<n}$. Luego, $t = (p_i \to s_i)_{i<n}$ y $t' = (p_i \to s'_i)_{i<n}$ con $s_k \to_{\texttt{CAP}} s'_k$ y $(s_i = s'_i)_{\substack{i<n \\ i \neq k}}$ para algún $k < n$. Por *h.i.* se tiene $\sigma s_k \to_{\texttt{CAP}} \sigma s'_k$. Sin pérdida de generalidad se asumen $(p_i\, \texttt{evita}\, \sigma)_{i<n}$. Luego, $\sigma t = (p_i \to \sigma s_i)_{i<n} \to_{\texttt{CAP}} (p_i \to \sigma s'_i)_{i<n} = \sigma t''$.

□

Del mismo modo, la relación de reducción $\twoheadrightarrow_{\texttt{CAP}}$ también se extiende al resultado de un match decidido definiendo $\texttt{fail} \twoheadrightarrow_{\texttt{CAP}} \texttt{fail}$.

Lema B.0.3. *Sea $\{\!\{p \setminus u\}\!\}$ un match decidido. Si $u \to_{\texttt{CAP}} u'$, entonces $\{\!\{p \setminus u'\}\!\}$ es un match decidido y $\{\!\{p \setminus u\}\!\} \twoheadrightarrow_{\texttt{CAP}} \{\!\{p \setminus u'\}\!\}$.*

Demostración. Por definición $u = \texttt{C}\langle l \rangle$ y $u' = \texttt{C}\langle r \rangle$ con $l \mapsto_{\texttt{CAP}} r$. Se procede por inducción en $\texttt{C}$.

- $\mathtt{C} = \Box$. Luego, $u = (p_i \to t_i)_{i<n}\, s$. Notar que $u \notin \mathbb{M}_{\mathtt{CAP}}$. Por lo tanto, se tiene necesariamente $p = x$, dado que $\{\!\{p \setminus u\}\!\}$ se encuentra decidido por hipótesis. Luego, $\{\!\{p \setminus u\}\!\} = \{x \setminus u\}$ y $\{\!\{p \setminus u'\}\!\} = \{x \setminus u'\}$. El hecho de que $\{x \setminus u\} \twoheadrightarrow_{\mathtt{CAP}} \{x \setminus u'\}$ es inmediato.
- $\mathtt{C} = \mathtt{C}'\, u_1$. Luego, $u = u_0\, u_1$ y $u' = u'_0\, u_1$ con $u_0 \to_{\mathtt{CAP}} u'_0$.
  - Si $\{\!\{p \setminus u\}\!\} = \sigma$, se distinguen dos casos:
    1. $p = x$. Luego, $\sigma = \{x \setminus u\}$ y se concluye con $\{\!\{p \setminus u'\}\!\} = \{x \setminus u'\}$.
    2. $p = p_0\, p_1$. Luego, $\sigma = \{\!\{p_0 \setminus u_0\}\!\} \uplus \{\!\{p_1 \setminus u_1\}\!\} = \sigma_0 \uplus \sigma_1$. Por *h.i.* se tiene $\{\!\{p_0 \setminus u'_0\}\!\} = \sigma'_0$ con $\sigma_0 \twoheadrightarrow_{\mathtt{CAP}} \sigma'_0$. Más aún, dado que $\mathsf{dom}(\sigma'_0) = \mathsf{dom}(\sigma_0)$, es seguro concluir $\{\!\{p \setminus u'\}\!\} = \sigma'_0 \uplus \sigma_1$.
  - Si $\{\!\{p \setminus u\}\!\} = \mathtt{fail}$, se tiene otros dos casos posibles:
    1. $p = \mathtt{c}$. Luego, se tiene también $\{\!\{\mathtt{c} \setminus u_0\, u_1\}\!\} = \mathtt{fail}$.
    2. $p = p_0\, p_1$. Luego, el mismatch es interno pues se asumen patrones lineales (*cf.* Sec. 3.2), por lo que la falla no se produce en la unión disjunta. Es decir, se tiene $\sigma = \{\!\{p_0 \setminus u_0\}\!\} \uplus \{\!\{p_1 \setminus u_1\}\!\}$ con $\{\!\{p_0 \setminus u_0\}\!\} = \mathtt{fail}$ o $\{\!\{p_1 \setminus u_1\}\!\} = \mathtt{fail}$. Si vale el primer caso, por *h.i.* se tiene $\{\!\{p_0 \setminus u'_0\}\!\} = \mathtt{fail}$. Si no, vale $\{\!\{p_1 \setminus u_1\}\!\} = \mathtt{fail}$. En cualquier caso, $\{\!\{p \setminus u'\}\!\} = \{\!\{p_0 \setminus u'_0\}\!\} \uplus \{\!\{p_1 \setminus u_1\}\!\} = \mathtt{fail}$.
- $\mathtt{C} = u_0\, \mathtt{C}'$. Luego, $u = u_0\, u_1$ y $u' = u_0\, u'_1$ con $u_1 \to_{\mathtt{CAP}} u'_1$. Este caso es análogo al anterior.
- $\mathtt{C} = (p_i \to \mathtt{C}_k)_{i<n}$. Luego, $u = (p_i \to s_i)_{i<n}$ con $s_k \to_{\mathtt{CAP}} s'_k$ y $(s_i = s'_i)_{\substack{i<n \\ i \neq k}}$ para algún $k < n$.
  - Si $\{\!\{p \setminus u\}\!\} = \sigma$, entonces $p = x$. Se concluye con $\{\!\{p \setminus u'\}\!\} = \{x \setminus u'\}$.
  - Si $\{\!\{p \setminus u\}\!\} = \mathtt{fail}$, entonces $p = \mathtt{c}$ o $p = p_0\, p_1$. En ambos casos se tiene $\{\!\{p \setminus u'\}\!\} = \mathtt{fail}$, por lo que se concluye.

□

Corolario B.0.4. *Sea $\{\!\{p \setminus u\}\!\}$ un match decidido. Si $u \twoheadrightarrow_{\mathtt{CAP}} u'$, entonces $\{\!\{p \setminus u'\}\!\}$ es un match decidido y $\{\!\{p \setminus u\}\!\} \twoheadrightarrow_{\mathtt{CAP}} \{\!\{p \setminus u'\}\!\}$.*

La *relación de reducción paralela* $\Rrightarrow_{\mathtt{CAP}}$ para términos de PPC se define inductivamente como:

$$\frac{}{x \Rrightarrow_{\mathtt{CAP}} x}\ (\text{R-VAR}) \qquad \frac{}{\mathtt{c} \Rrightarrow_{\mathtt{CAP}} \mathtt{c}}\ (\text{R-CONST})$$

$$\frac{r \Rrightarrow_{\mathtt{CAP}} r' \quad u \Rrightarrow_{\mathtt{CAP}} u'}{r\, u \Rrightarrow_{\mathtt{CAP}} r'\, u'}\ (\text{R-APP}) \qquad \frac{(s_i \Rrightarrow_{\mathtt{CAP}} s'_i)_{i<n}}{(p_i \to s_i \Rrightarrow_{\mathtt{CAP}} p_i \to s'_i)_{i<n}}\ (\text{R-ABS})$$

$$\frac{(\{\!\{p_i \setminus u\}\!\} = \mathtt{fail})_{i<k} \quad \{\!\{p_k \setminus u\}\!\} = \sigma \quad s_k \Rrightarrow_{\mathtt{CAP}} s'_k \quad k < n \quad u \Rrightarrow_{\mathtt{CAP}} u'}{(p_i \to s_i)_{i<n}\, u \Rrightarrow_{\mathtt{CAP}} \{\!\{p_k \setminus u'\}\!\}\, s'_k}\ (\text{R-BETA})$$

Notar que la relación de reducción paralela resulta reflexiva.

Al igual que $\twoheadrightarrow_{\mathtt{CAP}}$, la noción de reducción paralela $\Rrightarrow_{\mathtt{CAP}}$ se extiende a sustituciones y matchings decididos, de modo que: $\sigma \Rrightarrow_{\mathtt{CAP}} \sigma'$ sii $\mathsf{dom}(\sigma) = \mathsf{dom}(\sigma')$ y $\sigma(x) \Rrightarrow_{\mathtt{CAP}} \sigma'(x)$ para toda $x \in \mathsf{dom}(\sigma)$; y $\mathtt{fail} \Rrightarrow_{\mathtt{CAP}} \mathtt{fail}$.

Lema B.0.5. *Sean $t \in \mathbb{T}_{\mathtt{CAP}}$ y σ una sustitución. Si $t \Rrightarrow_{\mathtt{CAP}} t'$ y $\sigma \Rrightarrow_{\mathtt{CAP}} \sigma'$, entonces $\sigma t \Rrightarrow_{\mathtt{CAP}} \sigma' t'$.*

Demostración. Por inducción en $t \Rrightarrow_{\mathtt{CAP}} t'$.

- (R-VAR). Luego, $t = x = t'$ y se concluye por hipótesis $\sigma x \Rrightarrow_{\mathtt{CAP}} \sigma' x$.
- (R-CONST). Luego, $t = \mathtt{c} = t'$ y $\sigma t = \mathtt{c} = \sigma' t'$. Se concluye por (R-CONST).
- (R-APP). Luego, $t = r\,u$ y $t' = r'\,u'$ con $r \Rrightarrow_{\mathtt{CAP}} r'$ y $u \Rrightarrow_{\mathtt{CAP}} u'$. Por *h.i.* se tienen $\sigma r \Rrightarrow_{\mathtt{CAP}} \sigma' r'$ y $\sigma u \Rrightarrow_{\mathtt{CAP}} \sigma' u'$. Se concluye por (R-APP), $\sigma t = \sigma r\,\sigma u \Rrightarrow_{\mathtt{CAP}} \sigma r'\,\sigma u' = \sigma' t'$.
- (R-ABS). Luego, $t = (p_i \to s_i)_{i<n}$ y $t'(p_i \to s'_i)_{i<n}$ con $(s_i \Rrightarrow_{\mathtt{CAP}} s'_i)_{i<n}$. Sin pérdida de generalidad se asumen $(p_i\ \mathtt{evita}\ \sigma)_{i<n}$ y $(p_i\ \mathtt{evita}\ \sigma')_{i<n}$. Más aún, por *h.i.* se tienen $(\sigma s_i \Rrightarrow_{\mathtt{CAP}} \sigma' s'_i)_{i<n}$. Luego, se concluye por (R-ABS), $\sigma t = (p_i \to \sigma s_i)_{i<n} \Rrightarrow_{\mathtt{CAP}} (p_i \to \sigma s'_i)_{i<n} = \sigma' t'$.
- (R-BETA). Luego, $t = (p_i \to s_i)_{i<n}\,u$ y $t' = \{\!\{p_k \setminus u'\}\!\}\,s'_k$ para algún $k < n$ con $(\{\!\{p_i \setminus u\}\!\} = \mathtt{fail})_{i<k}$, $\{\!\{p_k \setminus u\}\!\} = \rho$, $s_k \Rrightarrow_{\mathtt{CAP}} s'_k$ y $u \Rrightarrow_{\mathtt{CAP}} u'$. Por *h.i.* se tienen $\sigma s_k \Rrightarrow_{\mathtt{CAP}} \sigma' s'_k$ y $\sigma u \Rrightarrow_{\mathtt{CAP}} \sigma' u'$. Sin pérdida de generalidad se asumen $(p_i\ \mathtt{evita}\ \sigma)_{i<n}$. Luego, por Lem B.0.1, $\{\!\{p_k \setminus \sigma u\}\!\} = \rho'$. Más aún, también por Lem. B.0.1, se tienen $(\{\!\{p_i \setminus \sigma u\}\!\} = \mathtt{fail})_{i<k}$. Finalmente, se concluye por (R-BETA), $\sigma t = (p_i \to \sigma s_i)_{i<n}\,u \Rrightarrow_{\mathtt{CAP}} \{\!\{p_k \setminus \sigma' u'\}\!\}\,\sigma' s'_k = \sigma' t'$.

□

Lema B.0.6. *Sea $\{\!\{p \setminus u\}\!\}$ un match decidido. Si $u \Rrightarrow_{\mathtt{CAP}} u'$, entonces $\{\!\{p \setminus u'\}\!\}$ es un match decidido y $\{\!\{p \setminus u\}\!\} \Rrightarrow_{\mathtt{CAP}} \{\!\{p \setminus u'\}\!\}$.*

Demostración. Por inducción en $u \Rrightarrow_{\mathtt{CAP}} u'$.

- (R-VAR). Luego, $u = x = u'$ y el resultado es inmediato.
- (R-CONST). Luego, $u = \mathtt{c} = u'$ y el resultado es inmediato.
- (R-APP). Luego, $u = u_0\,u_1$ y $u' = u'_0\,u'_1$ con $u_0 \Rrightarrow_{\mathtt{CAP}} u'_0$ y $u_1 \Rrightarrow_{\mathtt{CAP}} u'_1$.
  - Si $\{\!\{p \setminus u\}\!\} = \sigma$, se distinguen dos casos:
    1. $p = x$. Luego, $\sigma = \{x \setminus u\}$ y se concluye con $\{\!\{p \setminus u'\}\!\} = \{x \setminus u'\}$.
    2. $p = p_0\,p_1$. Luego, $\sigma = \{\!\{p_0 \setminus u_0\}\!\} \uplus \{\!\{p_1 \setminus u_1\}\!\} = \sigma_0 \uplus \sigma_1$. Por *h.i.* se tiene $(\{\!\{p_i \setminus u'_i\}\!\} = \sigma'_i)_{i<2}$ con $(\sigma_i \Rrightarrow_{\mathtt{CAP}} \sigma'_i)_{i<2}$. Más aún, dado que $(\mathsf{dom}(\sigma'_i) = \mathsf{dom}(\sigma_i))_{i<2}$, es seguro concluir $\{\!\{p \setminus u'\}\!\} = \sigma'_0 \uplus \sigma'_1$.
  - Si $\{\!\{p \setminus u\}\!\} = \mathtt{fail}$, se tiene otros dos casos posibles:
    1. $p = \mathtt{c}$. Luego, se tiene también $\{\!\{\mathtt{c} \setminus u_0\,u_1\}\!\} = \mathtt{fail}$.
    2. $p = p_0\,p_1$. Luego, el mismatch es interno pues se asumen patrones lineales (*cf.* Sec. 3.2), por lo que la falla no se produce en la unión disjunta. Es decir, se tiene $\sigma = \{\!\{p_0 \setminus u_0\}\!\} \uplus \{\!\{p_1 \setminus u_1\}\!\}$ con $\{\!\{p_0 \setminus u_0\}\!\} = \mathtt{fail}$ o $\{\!\{p_1 \setminus u_1\}\!\} = \mathtt{fail}$. Por *h.i.* se tiene $\{\!\{p_0 \setminus u'_0\}\!\} = \mathtt{fail}$ o $\{\!\{p_1 \setminus u'_1\}\!\} = \mathtt{fail}$. Por lo tanto, $\{\!\{p \setminus u'\}\!\} = \{\!\{p_0 \setminus u'_0\}\!\} \uplus \{\!\{p_1 \setminus u'_1\}\!\} = \mathtt{fail}$.
- (R-ABS). Luego, $u = (p_i \to s_i)_{i<n}$ y $u'(p_i \to s'_i)_{i<n}$ con $(s_i \Rrightarrow_{\mathtt{CAP}} s'_i)_{i<n}$.
  - Si $\{\!\{p \setminus u\}\!\} = \sigma$, entonces $p = x$. Se concluye con $\{\!\{p \setminus u'\}\!\} = \{x \setminus u'\}$.
  - Si $\{\!\{p \setminus u\}\!\} = \mathtt{fail}$, entonces $p = \mathtt{c}$ o $p = p_0\,p_1$. En ambos casos se tiene $\{\!\{p \setminus u'\}\!\} = \mathtt{fail}$, por lo que se concluye.
- (R-BETA). Luego, $u = (p_i \to t_i)_{i<n}\,s$. Notar que $u \notin \mathbb{M}_{\mathtt{CAP}}$. Por lo tanto, se tiene necesariamente $p = x$, dado que $\{\!\{p \setminus u\}\!\}$ se encuentra decidido por hipótesis. Luego, $\{\!\{p \setminus u\}\!\} = \{x \setminus u\}$ y $\{\!\{p \setminus u'\}\!\} = \{x \setminus u'\}$. El hecho de que $\{x \setminus u\} \Rrightarrow_{\mathtt{CAP}} \{x \setminus u'\}$ es inmediato.

□

Un *complete development* de un término t, notado t^*, se define inductivamente como:

$$
\begin{array}{rcl}
x^* & \triangleq & x \\
\mathtt{c}^* & \triangleq & \mathtt{c} \\
(t\,u)^* & \triangleq & \begin{cases} \{\!\{p_k \setminus u^*\}\!\}\, s_k{}^* & \text{si } t = (p_i \to s_i)_{i<n},\ \{\!\{p_k \setminus u\}\!\} = \sigma \text{ y} \\ & \quad (\{\!\{p_i \setminus u\}\!\} = \mathtt{fail})_{i<k} \text{ para algún } k < n \\ t^*\, u^* & \text{si no} \end{cases} \\
(p_i \to s_i)_{i<n}{}^* & \triangleq & (p_i \to s_i{}^*)_{i<n}
\end{array}
$$

La noción de complete development es utilizada para mostrar que $\Rrightarrow_{\mathtt{CAP}}$ satisface la propiedad del diamante [Nip90, BKvO03].

Lema B.0.7. *Sea $t \in \mathbb{T}_{CAP}$. Si $t \Rrightarrow_{\mathtt{CAP}} t'$, entonces $t' \Rrightarrow_{\mathtt{CAP}} t^*$.*

Demostración. Por inducción en $t \Rrightarrow_{\mathtt{CAP}} t'$.

- (R-VAR). Luego, $t = x = t' = t^*$. Se concluye por reflexividad de $\Rrightarrow_{\mathtt{CAP}}$.
- (R-CONST). Luego, $t = \mathtt{c} = t' = t^*$. Nuevamente, se concluye por reflexividad de $\Rrightarrow_{\mathtt{CAP}}$.
- (R-APP). Luego, $t = r\,u$ y $t' = r'\,u'$ con $r \Rrightarrow_{\mathtt{CAP}} r'$ y $u \Rrightarrow_{\mathtt{CAP}} u'$. Se distinguen dos casos:
  1. Si t es un redex (R-BETA), entonces $r = (p_i \to s_i)_{i<n}$ y $r' = (p_i \to s'_i)_{i<n}$ con $(s_i \Rrightarrow_{\mathtt{CAP}} s'_i)_{i<n}$, $\{\!\{p_k \setminus u\}\!\} = \sigma$ y $(\{\!\{p_i \setminus u\}\!\} = \mathtt{fail})_{i<k}$ para algún $k < n$. Por *h.i.* se tienen $(s'_i \Rrightarrow_{\mathtt{CAP}} s_i{}^*)_{i<n}$ y $u' \Rrightarrow_{\mathtt{CAP}} u^*$. De $u \Rrightarrow_{\mathtt{CAP}} u'$, por Lem. B.0.6, $\{\!\{p_k \setminus u'\}\!\} = \sigma'$ y $(\{\!\{p_i \setminus u'\}\!\} = \mathtt{fail})_{i<k}$. Luego, por (R-BETA), $t' = (p_i \to s'_i)_{i<n}\, u' \Rrightarrow_{\mathtt{CAP}} \{\!\{p_k \setminus u^*\}\!\}\, s_k{}^* = t^*$.
  2. Si t es un redex (R-BETA), por *h.i.* se tienen $r' \Rrightarrow_{\mathtt{CAP}} r^*$ y $u' \Rrightarrow_{\mathtt{CAP}} u^*$. Luego, se concluye por (R-APP), $t' = r'\,u' \Rrightarrow_{\mathtt{CAP}} r^*\,u^* = t^*$.
- (R-ABS). Luego, $t = (p_i \to s_i)_{i<n}$ y $t'(p_i \to s'_i)_{i<n}$ con $(s_i \Rrightarrow_{\mathtt{CAP}} s'_i)_{i<n}$. Por *h.i.* $(s'_i \Rrightarrow_{\mathtt{CAP}} s_i{}^*)_{i<n}$. Luego, se concluye por (R-ABS), $t' = (p_i \to s'_i)_{i<n} \Rrightarrow_{\mathtt{CAP}} (p_i \to s_i{}^*)_{i<n} = t'^*$.
- (R-BETA). Luego, $t = (p_i \to s_i)_{i<n}\, u$ y $t' = \{\!\{p_k \setminus u'\}\!\}\, s'_k$ para algún $k < n$ con $(\{\!\{p_i \setminus u\}\!\} = \mathtt{fail})_{i<k}$, $\{\!\{p_k \setminus u\}\!\} = \sigma$, $s_k \Rrightarrow_{\mathtt{CAP}} s'_k$ y $u \Rrightarrow_{\mathtt{CAP}} u'$. Por *h.i.* se tienen $s'_k \Rrightarrow_{\mathtt{CAP}} s_k{}^*$ y $u' \Rrightarrow_{\mathtt{CAP}} u^*$. Luego, por Lem B.0.6, $\{\!\{p_k \setminus u\}\!\} \Rrightarrow_{\mathtt{CAP}} \{\!\{p_k \setminus u'\}\!\} \Rrightarrow_{\mathtt{CAP}} \{\!\{p_k \setminus u^*\}\!\}$. Más aún, ambos matching son exitosos (*i.e.* retornan sustituciones). Finalmente, por Lem. B.0.5, $t' = \{\!\{p_k \setminus u'\}\!\}\, s'_k \Rrightarrow_{\mathtt{CAP}} \{\!\{p_k \setminus u^*\}\!\}\, s_k{}^* = t^*$.

□

Corolario B.0.8. *La relación de reducción paralela $\Rrightarrow_{\mathtt{CAP}}$ satisface la propiedad del diamante y, por lo tanto, es confluente (CR).*

Por último, para deducir la confluencia de CAP mediante el método de Tait–Martin-Löf, basta que ver $\Rrightarrow_{\mathtt{CAP}}$ extiende a $\to_{\mathtt{CAP}}$ y al mismo tiempo está incluida en su clausura reflexiva–transitiva $\twoheadrightarrow_{\mathtt{CAP}}$.

Lema B.0.9. $\to_{\mathtt{CAP}} \subseteq \Rrightarrow_{\mathtt{CAP}} \subseteq \twoheadrightarrow_{\mathtt{CAP}}$.

Demostración. Para la primera inclusión ($\to_{\mathtt{CAP}} \subseteq \Rrightarrow_{\mathtt{CAP}}$) considerar $t \to_{\mathtt{CAP}} t'$. Por definición $t = \mathtt{C}\langle l\rangle$ y $t' = \mathtt{C}\langle r\rangle$ con $l \mapsto_{\mathtt{CAP}} r$. Se procede por inducción en $\mathtt{C}$ para mostrar que $t \Rrightarrow_{\mathtt{CAP}} t'$.

- $\mathtt{C} = \Box$. Luego, $t = (p_i \to s_i)_{i<n}\, u$ y $t' = \sigma s_k$ con $(\{\!\{p_i \setminus u\}\!\} = \mathtt{fail})_{i<k}$ y $\{\!\{p_k \setminus u\}\!\} = \sigma$ para algún $k < n$. Por reflexividad de $\Rrightarrow_{\mathtt{CAP}}$ se tienen $s_k \Rrightarrow_{\mathtt{CAP}} s_k$ y $u \Rrightarrow_{\mathtt{CAP}} u$. Luego, por (R-BETA), $t = (p_i \to s_i)_{i<n}\, u \Rrightarrow_{\mathtt{CAP}} \{\!\{p_k \setminus u\}\!\} s_k = t'$.
- $\mathtt{C} = \mathtt{C}'\,u$. Luego, $t = r\,u$ y $t' = r'\,u$ con $r \to_{\mathtt{CAP}} r'$. Por *h.i.* se tiene $r \Rrightarrow_{\mathtt{CAP}} r'$. Más aún, por reflexividad de $\Rrightarrow_{\mathtt{CAP}}$ también se tiene $u \Rrightarrow_{\mathtt{CAP}} u$. Luego, por (R-APP), $t = r\,u \Rrightarrow_{\mathtt{CAP}} r'\,u = t'$.

- $\mathtt{C} = r\,\mathtt{C}'$. Luego, $t = r\,u$ y $t' = r\,u'$ con $u \to_{\mathtt{CAP}} u'$. Por *h.i.* se tiene $u \Rrightarrow_{\mathtt{CAP}} u'$. Más aún, por reflexividad de $\Rrightarrow_{\mathtt{CAP}}$ también se tiene $r \Rrightarrow_{\mathtt{CAP}} r$. Luego, por (R-APP), $t = r\,u \Rrightarrow_{\mathtt{CAP}} r\,u' = t'$.
- $\mathtt{C} = (p_i \to \mathtt{C}_k)_{i<n}$. Luego, $t = (p_i \to s_i)_{i<n}$ y $t' = (p_i \to s'_i)_{i<n}$ con $s_k \to_{\mathtt{CAP}} s'_k$ y $(s_i = s'_i)_{\substack{i<n \\ i\neq k}}$ para algún $k < n$. Por *h.i.* se tiene $s_k \Rrightarrow_{\mathtt{CAP}} s'_k$. Más aún, por reflexividad de $\Rrightarrow_{\mathtt{CAP}}$ también se tienen $(s_i \Rrightarrow_{\mathtt{CAP}} s_i)_{\substack{i<n \\ i\neq k}}$. Luego, por (R-ABS), $t = (p_i \to s_i)_{i<n} \Rrightarrow_{\mathtt{CAP}} (p_i \to s'_i)_{i<n} = t'$.

Para la segunda inclusión ($\Rrightarrow_{\mathtt{CAP}} \subseteq \twoheadrightarrow_{\mathtt{CAP}}$) considerar $t \Rrightarrow_{\mathtt{CAP}} t'$. Se procede por inducción en $t \Rrightarrow_{\mathtt{CAP}} t'$ para mostrar que $t \twoheadrightarrow_{\mathtt{CAP}} t'$.

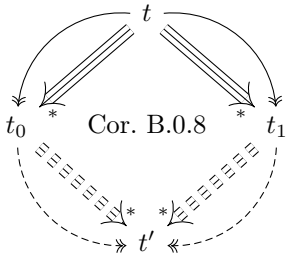
- (R-VAR). Luego, $t = x = t'$. Se concluye por reflexividad de $\twoheadrightarrow_{\mathtt{CAP}}$.
- (R-CONST). Luego, $t = \mathtt{c} = t'$. Se concluye por reflexividad de $\twoheadrightarrow_{\mathtt{CAP}}$.
- (R-APP). Luego, $t = r\,u$ y $t' = r'\,u'$ con $r \Rrightarrow_{\mathtt{CAP}} r'$ y $u \Rrightarrow_{\mathtt{CAP}} u'$. Por *h.i.* se tienen $r \twoheadrightarrow_{\mathtt{CAP}} r'$ y $u \twoheadrightarrow_{\mathtt{CAP}} u'$. Por lo tanto, $t = r\,u \twoheadrightarrow_{\mathtt{CAP}} r'\,u \twoheadrightarrow_{\mathtt{CAP}} r'\,u' = t'$.
- (R-ABS). Luego, $t = (p_i \to s_i)_{i<n}$ y $t'(p_i \to s'_i)_{i<n}$ con $(s_i \Rrightarrow_{\mathtt{CAP}} s'_i)_{i<n}$. Por *h.i.* $(s_i \twoheadrightarrow_{\mathtt{CAP}} s'_i)_{i<n}$. Por lo tanto, se concluye como en el caso anterior, $t = (p_i \to s_i)_{i<n} \twoheadrightarrow_{\mathtt{CAP}} (p_i \to s'_i)_{i<n} = t'$.
- (R-BETA). Luego, $t = (p_i \to s_i)_{i<n}\,u$ y $t' = \{\!\{p_k \setminus u'\}\!\}\,s'_k$ para algún $k < n$ con $(\{\!\{p_i \setminus u\}\!\} = \mathtt{fail})_{i<k}$, $\{\!\{p_k \setminus u\}\!\} = \sigma$, $s_k \Rrightarrow_{\mathtt{CAP}} s'_k$ y $u \Rrightarrow_{\mathtt{CAP}} u'$. Notar que $(p_i \to s_i)_{i<n}\,u \mapsto_{\mathtt{CAP}} \sigma\,s_k$. Más aún, por *h.i.* se tienen $s_k \twoheadrightarrow_{\mathtt{CAP}} s'_k$ y $u \twoheadrightarrow_{\mathtt{CAP}} u'$. Por Cor. B.0.4, $\sigma = \{\!\{p_k \setminus u\}\!\} \twoheadrightarrow_{\mathtt{CAP}} \{\!\{p_k \setminus u'\}\!\} = \sigma'$ y entonces, por Lem. B.0.2, $\sigma\,s_k \twoheadrightarrow_{\mathtt{CAP}} \sigma'\,s'_k$. Es decir, $t = (p_i \to s_i)_{i<n}\,u \twoheadrightarrow_{\mathtt{CAP}} \{\!\{p_k \setminus u'\}\!\}\,s'_k = t'$.

□

La confluencia de $\to_{\mathtt{CAP}}$ es consecuencia inmediata del Cor. B.0.8 y el Lem. B.0.9.

Teorema 3.2.1. *La relación de reducción $\to_{\mathtt{CAP}}$ es confluente (CR).*

Demostración. Dado que las clausuras son, por definición, funciones monótonas e idempotentes, del Lem. B.0.9 se sigue $\Rrightarrow^*_{\mathtt{CAP}} = \twoheadrightarrow_{\mathtt{CAP}}$. Luego, se satisface el siguiente diagrama de confluencia:



□

Relación entre CAP y PPC

La traducción de términos CAP en términos del fragmento estático de PPC se extiende a sustituciones de modo que $\mathsf{dom}(\sigma^{\mathsf{PPC}}) = \mathsf{dom}(\sigma)$ y $\sigma^{\mathsf{PPC}}(x) \triangleq (\sigma(x))^{\mathsf{PPC}}$ para toda $x \in \mathsf{dom}(\sigma)$.

Lema B.0.10. *Sean $t \in \mathbb{T}_{\mathsf{CAP}}$ y σ una sustitución. $(\sigma t)^{\mathsf{PPC}} = \sigma^{\mathsf{PPC}} t^{\mathsf{PPC}}$.*

Demostración. Por inducción en t.

- $t = x$. Luego, ${t}^{\mathsf{PPC}} = x$. Si $x \in \mathsf{dom}(\sigma)$, entonces $(\sigma t)^{\mathsf{PPC}} = (\sigma(x))^{\mathsf{PPC}} = \sigma^{\mathsf{PPC}}(x) = \sigma^{\mathsf{PPC}} t^{\mathsf{PPC}}$. Si no, $(\sigma t)^{\mathsf{PPC}} = x = \sigma^{\mathsf{PPC}} t^{\mathsf{PPC}}$.
- $t = \mathtt{c}$. Luego, ${t}^{\mathsf{PPC}} = \mathtt{c}$ y $(\sigma t)^{\mathsf{PPC}} = \mathtt{c} = \sigma^{\mathsf{PPC}} t^{\mathsf{PPC}}$.
- $t = r\, u$. Luego, $(\sigma t)^{\mathsf{PPC}} = (\sigma r)^{\mathsf{PPC}}\, (\sigma u)^{\mathsf{PPC}}$. Por *h.i.* se tienen $(\sigma r)^{\mathsf{PPC}} = \sigma^{\mathsf{PPC}} r^{\mathsf{PPC}}$ y $(\sigma u)^{\mathsf{PPC}} = \sigma^{\mathsf{PPC}} u^{\mathsf{PPC}}$. Por lo tanto, se concluye $(\sigma t)^{\mathsf{PPC}} = (\sigma^{\mathsf{PPC}} s^{\mathsf{PPC}})\, (\sigma^{\mathsf{PPC}} u^{\mathsf{PPC}}) = \sigma^{\mathsf{PPC}} t^{\mathsf{PPC}}$.
- $t = (p_i \to s_i)_{i<n}$. Sin perdida de generalidad se asumen $(p_i\ \mathtt{evita}\ \sigma)_{i<n}$. Por inducción en n (el número de ramas de la abstracción):
  - $n = 1$. Luego, $(\sigma t)^{\mathsf{PPC}} = (\sigma(p_0 \to s_0))^{\mathsf{PPC}} = p_0 \to (\sigma s_0)^{\mathsf{PPC}}$. Por *h.i.* del lema se tiene $(\sigma s_0)^{\mathsf{PPC}} = \sigma^{\mathsf{PPC}} {s_0}^{\mathsf{PPC}}$. Por lo tanto, se concluye $(\sigma t)^{\mathsf{PPC}} = p_0 \to (\sigma s_0)^{\mathsf{PPC}} = p_0 \to \sigma^{\mathsf{PPC}} {s_0}^{\mathsf{PPC}} = \sigma^{\mathsf{PPC}} t^{\mathsf{PPC}}$.
  - $n > 1$. Luego, apelando a la codificación de la alternativa en PPC (*cf.* Sec. 3.1.1) se tiene

$$(\sigma t)^{\mathsf{PPC}} \quad = \quad x \to (\mathtt{NoMatch}\, y \to y)\, ((p_0 \to z \to \mathtt{NoMatch}\, (\sigma s_0)^{\mathsf{PPC}})\, x\, ((\sigma(p_i \to s_i)_{i\in[1,n)})^{\mathsf{PPC}}\, x))$$

    con x, y y z variables frescas. Notar que la abstracción que se sustituye y traduce internamente contiene una rama menos que la original. Luego, *h.i.* $(\sigma(p_i \to s_i)_{i\in[1,n)})^{\mathsf{PPC}} = \sigma^{\mathsf{PPC}}(p_i \to s_i)_{i\in[1,n)}{}^{\mathsf{PPC}}$. Más aún, por *h.i.* del lema se tiene $(\sigma s_0)^{\mathsf{PPC}} = \sigma^{\mathsf{PPC}} {s_0}^{\mathsf{PPC}}$. Por lo tanto, se concluye

$$\begin{aligned}(\sigma t)^{\mathsf{PPC}} &= x \to (\mathtt{NoMatch}\, y \to y)\, ((p_0 \to z \to \mathtt{NoMatch}\, (\sigma s_0)^{\mathsf{PPC}})\, x\, ((\sigma(p_i \to s_i)_{i\in[1,n)})^{\mathsf{PPC}}\, x)) \\ &= x \to (\mathtt{NoMatch}\, y \to y)\, ((p_0 \to z \to \mathtt{NoMatch}\, \sigma^{\mathsf{PPC}} {s_0}^{\mathsf{PPC}})\, x\, (\sigma^{\mathsf{PPC}}(p_i \to s_i)_{i\in[1,n)}{}^{\mathsf{PPC}}\, x)) \\ &= \sigma^{\mathsf{PPC}} t^{\mathsf{PPC}}\end{aligned}$$

□

Del mismo modo, la traducción se extiende al resultado de un match definiendo $\mathtt{fail}^{\mathsf{PPC}} \triangleq \mathtt{fail}$ y $\mathtt{wait}^{\mathsf{PPC}} \triangleq \mathtt{wait}$. Notar, en particular, que la traducción preserva las matchables forms.

Lema B.0.11. *Sean $p \in \mathbb{P}_{\mathsf{CAP}}$ y $u \in \mathbb{T}_{\mathsf{CAP}}$. $\{\!\{p \setminus u\}\!\}^{\mathsf{PPC}} = \{\!\{p \setminus u^{\mathsf{PPC}}\}\!\}$.*

Demostración. Por inducción en p.

- $p = x$. Luego, $\{\!\{p \setminus u\}\!\}^{\mathsf{PPC}} = \{x \setminus u\}^{\mathsf{PPC}} = \{x \setminus u^{\mathsf{PPC}}\} = \{\!\{p \setminus u^{\mathsf{PPC}}\}\!\}$ por definición.
- $p = \mathtt{c}$. Luego, se analiza la forma de u:
  - $u \in \mathbb{M}_{\mathsf{CAP}}$. Se tienen dos casos posibles:
    1. $u = \mathtt{c}$. Luego, $u^{\mathsf{PPC}} = \mathtt{c}$ y $\{\!\{p \setminus u\}\!\}^{\mathsf{PPC}} = \{\} = \{\!\{p \setminus u^{\mathsf{PPC}}\}\!\}$.
    2. $u \neq \mathtt{c}$. Luego, $u^{\mathsf{PPC}} \neq \mathtt{c}$ y $\{\!\{p \setminus u\}\!\}^{\mathsf{PPC}} = \mathtt{fail} = \{\!\{p \setminus u^{\mathsf{PPC}}\}\!\}$.
  - $u \notin \mathbb{M}_{\mathsf{CAP}}$. Luego, $u^{\mathsf{PPC}} \notin \mathbb{M}_{\mathsf{PPC}}$. Se concluye pues $\{\!\{p \setminus u\}\!\}^{\mathsf{PPC}} = \mathtt{wait} = \{\!\{p \setminus u^{\mathsf{PPC}}\}\!\}$.
- $p = p_0\, p_1$. Luego, se analiza la forma de u:
  - $u \in \mathbb{M}_{\mathsf{CAP}}$. Se tienen dos casos posibles:
    1. $u = u_0\, u_1$. Luego, $u^{\mathsf{PPC}} = u_0\, u_1$ y $\{\!\{p \setminus u\}\!\}^{\mathsf{PPC}} = \{\!\{p_0 \setminus u_0\}\!\}^{\mathsf{PPC}} \uplus \{\!\{p_1 \setminus u_1\}\!\}^{\mathsf{PPC}}$. Por *h.i.* se tienen $(\{\!\{p_i \setminus u_i\}\!\}^{\mathsf{PPC}} = \{\!\{p_i \setminus u_i{}^{\mathsf{PPC}}\}\!\})_{i<2}$, por lo que se concluye, $\{\!\{p \setminus u\}\!\}^{\mathsf{PPC}} = \{\!\{p \setminus u^{\mathsf{PPC}}\}\!\}$.

2. $u \neq u_0\, u_1$. Luego, ${u}^{\mathsf{PPC}} \neq u_0\, u_1$ y $\{\!\{p \setminus u\}\!\}^{\mathsf{PPC}} = \mathtt{fail} = \{\!\{p \setminus {u}^{\mathsf{PPC}}\}\!\}$.

- $u \notin \mathbb{M}_{\mathsf{CAP}}$. Luego, ${u}^{\mathsf{PPC}} \notin \mathbb{M}_{\mathsf{PPC}}$. Se concluye pues $\{\!\{p \setminus u\}\!\}^{\mathsf{PPC}} = \mathtt{wait} = \{\!\{p \setminus {u}^{\mathsf{PPC}}\}\!\}$.

□

Lema 3.2.3. *Sea $t \in \mathbb{T}_{\mathsf{CAP}}$. Si $t \to_{\mathtt{CAP}} t'$, entonces ${t}^{\mathsf{PPC}} \twoheadrightarrow_{\mathtt{PPC}} {t'}^{\mathsf{PPC}}$.*

Demostración. Por definición, $t \to t'$ implica $t = \mathtt{C}\langle l \rangle$ y $t' = \mathtt{C}\langle r \rangle$ con $l \mapsto_\beta r$. La demostración es por inducción en $\mathtt{C}$.

- $\mathtt{C} = \Box$. Luego, $t = (p_i \to s_i)_{i<n}\, u$ y $t' = \sigma s_k$ con $(\{\!\{p_i \setminus u\}\!\} = \mathtt{fail})_{i<k}$ y $\{\!\{p_k \setminus u\}\!\} = \sigma$ para algún $k < n$. Por Def. 3.2.2, ${t}^{\mathsf{PPC}} = (p_i \to {s_i}^{\mathsf{PPC}})_{i<n}$ apelando a la definición de alternativa para PPC (*cf.* Sec. 3.1.1). Se muestra a continuación que ${t}^{\mathsf{PPC}} \twoheadrightarrow_{\mathtt{PPC}} (p_i \to {s_i}^{\mathsf{PPC}})_{i\in[k,n)}\, {u}^{\mathsf{PPC}}$, por inducción en el índice k de la primera rama de matching exitoso:
  - $k = 0$. Luego, ${t}^{\mathsf{PPC}} = (p_i \to {s_i}^{\mathsf{PPC}})_{i\in[k,n)}\, {u}^{\mathsf{PPC}}$ y se concluye por reflexividad de $\twoheadrightarrow_{\mathtt{PPC}}$.
  - $k > 0$. Luego, $\{\!\{p_0 \setminus u\}\!\} = \mathtt{fail}$ y, por Lem. B.0.11, $\{\!\{p_0 \setminus {u}^{\mathsf{PPC}}\}\!\} = \mathtt{fail}$. Entonces,

$$
\begin{array}{lll}
{t}^{\mathsf{PPC}} & = & (p_i \to {s_i}^{\mathsf{PPC}})_{i<n}\, {u}^{\mathsf{PPC}} \\
& = & (p_0 \to {s_0}^{\mathsf{PPC}} \mid (p_i \to {s_i}^{\mathsf{PPC}})_{i\in[1,n)})\, {u}^{\mathsf{PPC}} \\
& = & \underline{(x \to (\mathtt{NoMatch}\, y \to y)\, ((p_0 \to z \to \mathtt{NoMatch}\, {s_0}^{\mathsf{PPC}})\, x\, ((p_i \to {s_i}^{\mathsf{PPC}})_{i\in[1,n)}\, x)))\, {u}^{\mathsf{PPC}}} \\
& \to_{\mathtt{PPC}} & (\mathtt{NoMatch}\, y \to y)\, (\underline{(p_0 \to z \to \mathtt{NoMatch}\, {s_0}^{\mathsf{PPC}})\, {u}^{\mathsf{PPC}}}\, ((p_i \to {s_i}^{\mathsf{PPC}})_{i\in[1,n)}\, {u}^{\mathsf{PPC}})) \\
& \to_{\mathtt{PPC}} & (\mathtt{NoMatch}\, y \to y)\, (\underline{\mathtt{fail}\, (z \to \mathtt{NoMatch}\, {s_0}^{\mathsf{PPC}})}\, ((p_i \to {s_i}^{\mathsf{PPC}})_{i\in[1,n)}\, {u}^{\mathsf{PPC}})) \\
& \to_{\mathtt{PPC}} & \underline{(\mathtt{NoMatch}\, y \to y)\, (\mathtt{NoMatch}\, ((p_i \to {s_i}^{\mathsf{PPC}})_{i\in[1,n)}\, {u}^{\mathsf{PPC}}))} \\
& \to_{\mathtt{PPC}} & (p_i \to {s_i}^{\mathsf{PPC}})_{i\in[1,n)}\, {u}^{\mathsf{PPC}}
\end{array}
$$

    Por *h.i.* se tiene $(p_i \to {s_i}^{\mathsf{PPC}})_{i\in[1,n)}\, {u}^{\mathsf{PPC}} \twoheadrightarrow_{\mathtt{PPC}} (p_i \to {s_i}^{\mathsf{PPC}})_{i\in[k,n)}\, {u}^{\mathsf{PPC}}$, por lo que se concluye.

  Más aún, por Lem. B.0.11, se tiene $\{\!\{p_k \setminus {u}^{\mathsf{PPC}}\}\!\} = {\sigma}^{\mathsf{PPC}}$, por lo que $(p_i \to {s_i}^{\mathsf{PPC}})_{i\in[k,n)}\, {u}^{\mathsf{PPC}} \to_{\mathtt{PPC}} {\sigma}^{\mathsf{PPC}} {s_k}^{\mathsf{PPC}}$. Se concluye por Lem. B.0.10, ${t}^{\mathsf{PPC}} \twoheadrightarrow_{\mathtt{PPC}} {\sigma}^{\mathsf{PPC}} {s_k}^{\mathsf{PPC}} = {t'}^{\mathsf{PPC}}$.

- $\mathtt{C} = \mathtt{C}'\, u$. Luego, $t = r\, u$ y $t' = r'\, u$ con $r \to_{\mathtt{CAP}} r'$. Por *h.i.* se tiene ${r}^{\mathsf{PPC}} \twoheadrightarrow_{\mathtt{PPC}} {r'}^{\mathsf{PPC}}$. Por lo tanto, ${t}^{\mathsf{PPC}} = {r}^{\mathsf{PPC}}\, {u}^{\mathsf{PPC}} \twoheadrightarrow_{\mathtt{PPC}} {r'}^{\mathsf{PPC}}\, {u}^{\mathsf{PPC}} = {t'}^{\mathsf{PPC}}$.

- $\mathtt{C} = s\, \mathtt{C}'$. Luego, $t = r\, u$ y $t' = r\, u'$ con $u \to_{\mathtt{CAP}} u'$. Por *h.i.* se tiene ${u}^{\mathsf{PPC}} \twoheadrightarrow_{\mathtt{PPC}} {u'}^{\mathsf{PPC}}$. Por lo tanto, ${t}^{\mathsf{PPC}} = {r}^{\mathsf{PPC}}\, {u}^{\mathsf{PPC}} \twoheadrightarrow_{\mathtt{PPC}} {r}^{\mathsf{PPC}}\, {u'}^{\mathsf{PPC}} = {t'}^{\mathsf{PPC}}$.

- $\mathtt{C} = (p_i \to \mathtt{C}'_k)_{i<n}$. Luego, $t = (p_i \to s_i)_{i<n}$ y $t' = (p_i \to s'_i)_{i<n}$ con $s_k \to_{\mathtt{CAP}} s'_k$ y $(s_i = s'_i)_{\substack{i<n \\ i\neq k}}$ para algún $k < n$. Por *h.i.* se tiene ${s_k}^{\mathsf{PPC}} \twoheadrightarrow_{\mathtt{PPC}} {s'_k}^{\mathsf{PPC}}$. Por lo tanto, ${t}^{\mathsf{PPC}} = (p_i \to {s_i}^{\mathsf{PPC}})_{i<n} \twoheadrightarrow_{\mathtt{PPC}} (p_i \to {s'_i}^{\mathsf{PPC}})_{i<n} = {t''}^{\mathsf{PPC}}$.

□

Al igual que $\_^{\mathsf{PPC}}$, la traducción de términos del fragmento estático de PPC en términos de CAP se extiende a sustituciones de modo que $\mathsf{dom}({\sigma}^{\mathsf{CAP}}) = \mathsf{dom}(\sigma)$ y ${\sigma}^{\mathsf{CAP}}(x) \triangleq (\sigma(x))^{\mathsf{CAP}}$ para toda $x \in \mathsf{dom}(\sigma)$.

Lema B.0.12. *Sean $t \in \mathbb{T}_{PPC}$ y σ una sustitución.* $(\sigma t)^{CAP} = \sigma^{CAP} t^{CAP}$.

Demostración. Por inducción en t.

- $t = x$. Luego, $t^{\mathsf{CAP}} = x$. Si $x \in \mathsf{dom}(\sigma)$, entonces $(\sigma t)^{\mathsf{CAP}} = (\sigma(x))^{\mathsf{CAP}} = \sigma^{\mathsf{CAP}}(x) = \sigma^{\mathsf{CAP}} t^{\mathsf{CAP}}$. Si no, $(\sigma t)^{\mathsf{CAP}} = x = \sigma^{\mathsf{CAP}} t^{\mathsf{CAP}}$.
- $t = \mathtt{c}$. Luego, $t^{\mathsf{CAP}} = \mathtt{c}$ y $(\sigma t)^{\mathsf{CAP}} = \mathtt{c} = \sigma^{\mathsf{CAP}} t^{\mathsf{CAP}}$.
- $t = r\, u$. Luego, $(\sigma t)^{\mathsf{CAP}} = (\sigma r)^{\mathsf{CAP}}\, (\sigma u)^{\mathsf{CAP}}$. Por *h.i.* se tienen $(\sigma r)^{\mathsf{CAP}} = \sigma^{\mathsf{CAP}} r^{\mathsf{CAP}}$ y $(\sigma u)^{\mathsf{CAP}} = \sigma^{\mathsf{CAP}} u^{\mathsf{CAP}}$. Por lo tanto, se concluye $(\sigma t)^{\mathsf{CAP}} = (\sigma^{\mathsf{CAP}} s^{\mathsf{CAP}})\, (\sigma^{\mathsf{CAP}} u^{\mathsf{CAP}}) = \sigma^{\mathsf{CAP}} t^{\mathsf{CAP}}$.
- $t = p \to s$. Sin perdida de generalidad se asume p `evita` σ. Luego, $(\sigma t)^{\mathsf{CAP}} = (\sigma(p \to s))^{\mathsf{CAP}} = p \to (\sigma s)^{\mathsf{CAP}} \mid x \to \mathtt{NoMatch}$. Por *h.i.* se tiene $(\sigma s)^{\mathsf{CAP}} = \sigma^{\mathsf{CAP}} s^{\mathsf{CAP}}$. Por lo tanto, se concluye $(\sigma t)^{\mathsf{CAP}} = p \to (\sigma s)^{\mathsf{CAP}} \mid x \to \mathtt{NoMatch} = p \to \sigma^{\mathsf{CAP}} s^{\mathsf{CAP}} \mid x \to \mathtt{NoMatch} = \sigma^{\mathsf{CAP}} t^{\mathsf{PPC}}$.

□

Así mismo, la traducción se extiende al resultado de un match definiendo como es esperado: $\mathtt{fail}^{\mathsf{CAP}} \triangleq \mathtt{fail}$ y $\mathtt{wait}^{\mathsf{CAP}} \triangleq \mathtt{wait}$. Al igual que el caso de $\_^{\mathsf{PPC}}$, $\_^{\mathsf{CAP}}$ preserva las matchables forms.

Lema B.0.13. *Sean $p \in \mathbb{P}_{PPC}$ y $u \in \mathbb{T}_{PPC}$.* $\{\!\{p \setminus u\}\!\}^{CAP} = \{\!\{p \setminus u^{CAP}\}\!\}$.

Demostración. Por inducción en p.

- $p = x$. Luego, $\{\!\{p \setminus u\}\!\}^{\mathsf{CAP}} = \{x \setminus u\}^{\mathsf{CAP}} = \{x \setminus u^{\mathsf{CAP}}\} = \{\!\{p \setminus u^{\mathsf{CAP}}\}\!\}$ por definición.
- $p = \mathtt{c}$. Luego, se analiza la forma de u:
  - $u \in \mathbb{M}_{\mathsf{PPC}}$. Se tienen dos casos posibles:
    1. $u = \mathtt{c}$. Luego, $u^{\mathsf{CAP}} = \mathtt{c}$ y $\{\!\{p \setminus u\}\!\}^{\mathsf{CAP}} = \{\} = \{\!\{p \setminus u^{\mathsf{CAP}}\}\!\}$.
    2. $u \neq \mathtt{c}$. Luego, $u^{\mathsf{CAP}} \neq \mathtt{c}$ y $\{\!\{p \setminus u\}\!\}^{\mathsf{CAP}} = \mathtt{fail} = \{\!\{p \setminus u^{\mathsf{CAP}}\}\!\}$.
  - $u \notin \mathbb{M}_{\mathsf{PPC}}$. Luego, $u^{\mathsf{CAP}} \notin \mathbb{M}_{\mathsf{CAP}}$. Se concluye pues $\{\!\{p \setminus u\}\!\}^{\mathsf{CAP}} = \mathtt{wait} = \{\!\{p \setminus u^{\mathsf{CAP}}\}\!\}$.
- $p = p_0\, p_1$. Luego, se analiza la forma de u:
  - $u \in \mathbb{M}_{\mathsf{PPC}}$. Se tienen dos casos posibles:
    1. $u = u_0\, u_1$. Luego, $u^{\mathsf{CAP}} = u_0\, u_1$ y $\{\!\{p \setminus u\}\!\}^{\mathsf{CAP}} = \{\!\{p_0 \setminus u_0\}\!\}^{\mathsf{CAP}} \uplus \{\!\{p_1 \setminus u_1\}\!\}^{\mathsf{CAP}}$. Por *h.i.* se tienen $(\{\!\{p_i \setminus u_i\}\!\}^{\mathsf{CAP}} = \{\!\{p_i \setminus u_i^{\mathsf{CAP}}\}\!\})_{i<2}$, por lo que se concluye, $\{\!\{p \setminus u\}\!\}^{\mathsf{CAP}} = \{\!\{p \setminus u^{\mathsf{CAP}}\}\!\}$.
    2. $u \neq u_0\, u_1$. Luego, $u^{\mathsf{CAP}} \neq u_0\, u_1$ y $\{\!\{p \setminus u\}\!\}^{\mathsf{CAP}} = \mathtt{fail} = \{\!\{p \setminus u^{\mathsf{CAP}}\}\!\}$.
  - $u \notin \mathbb{M}_{\mathsf{PPC}}$. Luego, $u^{\mathsf{CAP}} \notin \mathbb{M}_{\mathsf{CAP}}$. Se concluye pues $\{\!\{p \setminus u\}\!\}^{\mathsf{CAP}} = \mathtt{wait} = \{\!\{p \setminus u^{\mathsf{CAP}}\}\!\}$.

□

Lema 3.2.5. *Sea $t \in \mathbb{T}_{PPC}$. Si $t \to_{\mathtt{PPC}} t'$, entonces $t^{CAP} \to_{\mathtt{CAP}} t'^{CAP}$.*

Demostración. Por definición, $t \to t'$ implica $t = \mathtt{C}\langle l \rangle$ y $t' = \mathtt{C}\langle r \rangle$ con $l \mapsto_{\mathtt{PPC}} r$. La demostración es por inducción en $\mathtt{C}$.

- $\mathtt{C} = \Box$. Luego, $t = (p \to s)\, u$ y $t' = \{\!\{p \setminus u\}\!\}\, s$. Por Def. 3.2.4, $t^{\mathsf{CAP}} = (p \to s^{\mathsf{CAP}} \mid x \to \mathtt{NoMatch})\, u^{\mathsf{CAP}}$. Se tienen dos casos posibles:

1. $\{\!\{p \setminus u\}\!\} = \sigma$. Por Lem. B.0.13, $\{\!\{p \setminus u^{\mathsf{CAP}}\}\!\} = \sigma^{\mathsf{CAP}}$. Luego, $t^{\mathsf{CAP}} \to_{\mathtt{CAP}} \sigma^{\mathsf{CAP}} s^{\mathsf{CAP}} = t'^{\mathsf{CAP}}$. Se concluye por Lem. B.0.12.
2. $\{\!\{p \setminus u\}\!\} = \mathtt{fail}$. Por Lem. B.0.13, $\{\!\{p \setminus u^{\mathsf{CAP}}\}\!\} = \mathtt{fail}$. Luego, $t^{\mathsf{CAP}} \to_{\mathtt{CAP}} \mathtt{NoMatch} = \mathtt{fail}\, s^{\mathsf{CAP}} = t'^{\mathsf{CAP}}$.

- $\mathtt{C} = \mathtt{C}'\, u$. Luego, $t = r\, u$ y $t' = r'\, u$ con $r \to_{\mathtt{PPC}} r'$. Por *h.i.* se tiene $r^{\mathsf{CAP}} \to_{\mathtt{CAP}} r'^{\mathsf{CAP}}$. Por lo tanto, $t^{\mathsf{CAP}} = r^{\mathsf{CAP}}\, u^{\mathsf{CAP}} \to_{\mathtt{CAP}} r'^{\mathsf{CAP}}\, u^{\mathsf{CAP}} = t'^{\mathsf{CAP}}$.
- $\mathtt{C} = s\, \mathtt{C}'$. Luego, $t = r\, u$ y $t' = r\, u'$ con $u \to_{\mathtt{PPC}} u'$. Por *h.i.* se tiene $u^{\mathsf{CAP}} \to_{\mathtt{CAP}} u'^{\mathsf{CAP}}$. Por lo tanto, $t^{\mathsf{CAP}} = r^{\mathsf{CAP}}\, u^{\mathsf{CAP}} \to_{\mathtt{CAP}} r^{\mathsf{CAP}}\, u'^{\mathsf{CAP}} = t'^{\mathsf{CAP}}$.
- $\mathtt{C} = p \to \mathtt{C}'$. Luego, $t = p \to s$ y $t' = p \to s'$ con $s \to_{\mathtt{PPC}} s'$. Por *h.i.* se tiene $s^{\mathsf{CAP}} \to_{\mathtt{CAP}} s'^{\mathsf{CAP}}$. Por lo tanto, $t^{\mathsf{CAP}} = p \to s^{\mathsf{CAP}} \to_{\mathtt{CAP}} p \to s'^{\mathsf{CAP}} = t'^{\mathsf{CAP}}$.

□

Sistema de tipos

Expresiones de tipo

Lema B.0.14. *Sean A y A' dos asociaciones distintas de la expresión $\oplus_{i<n} A_i$. Entonces, $A \simeq_\mu A'$.*

Demostración. Se prueba $\oplus_{i<n} A_i \simeq_\mu (\ldots(A_0 \oplus A_1)\ldots \oplus A_{n-1})$ (*i.e.* toda unión arbitraria es equivalente a una asociada a izquierda), y luego se concluye apelando a las reglas (E-SYMM) y (E-TRANS).

- $n = 1$ o $n = 2$. El resultado es inmediato, pues los tipos A_0 y $A_0 \oplus A_1$ están trivialmente asociados a izquierda.
- $n > 2$. Entonces, $\oplus_{i<n} A_i = (\oplus_{i<k} A_i) \oplus (\oplus_{i\in[k,n)} A_i)$ para algún $k < n-1$. Si $k = n-2$ se tiene $\oplus_{i<n} A_i = \oplus_{i<n-1} A_i \oplus A_{n-1}$ y se concluye directo de la *h.i.* Si no,

$$\begin{array}{lcll} \oplus_{i<n} A_i & = & (\oplus_{i<k} A_i) \oplus (\oplus_{i\in[k,n)} A_i) & \\ & \simeq_\mu & (\oplus_{i<k} A_i) \oplus (\ldots(A_{k+1} \oplus A_{k+2})\ldots \oplus A_{n-1}) & \textit{h.i.} \\ & = & (\oplus_{i<k} A_i) \oplus ((\oplus_{i\in[k,n-1)} A_i) \oplus A_{n-1}) & \\ & \simeq_\mu & (\oplus_{i<n-1} A_i) \oplus A_{n-1} & \text{(E-UNION-ASSOC)} \\ & \simeq_\mu & (\ldots(A_0 \oplus A_1)\ldots \oplus A_{n-2}) \oplus A_{n-1} & \textit{h.i.} \end{array}$$

□

Lema B.0.15. *Sea p una permutación de $[0, n)$. Entonces, $\oplus_{i<n} A_i \simeq_\mu \oplus_{i<n} A_{p(i)}$.*

Demostración. Por inducción en n.

- $n = 1$. Este caso es inmediato con $p = id$.
- $n > 1$. Sea p la función

$$p(i) = \begin{cases} p'(i) & \text{si } i < k \\ n & \text{si } i = k \\ p'(i-1) & \text{si } i > k \end{cases}$$

donde p' es una permutación de $[0, n-1)$ y $k < n$. Es decir, p permuta k con n y se comporta como p' para toda otra posición. Entonces,

$$\begin{array}{lcll} \oplus_{i<n} A_i & \simeq_\mu & (\oplus_{i<n-1} A_i) \oplus A_n & \text{Lem. B.0.14} \\ & \simeq_\mu & (\oplus_{i<n-1} A_{p'(i)}) \oplus A_n & \textit{h.i.} \end{array}$$

Si $k = n$, se concluye por Lem. B.0.14, $(\oplus_{i<n-1} A_{p'(i)}) \oplus A_n \simeq_\mu \oplus_{i<n} A_{p(i)}$. Si no, *i.e.* $k < n-1$,

$$\begin{array}{lll} \oplus_{i<n} A_i & \simeq_\mu & (\oplus_{i<n-1} A_{p'(i)}) \oplus A_n \\ & \simeq_\mu & (\oplus_{i<k} A_{p'(i)}) \oplus ((\oplus_{i\in[k,n-1)} A_{p'(i)}) \oplus A_n) \quad \text{Lem. B.0.14} \\ & \simeq_\mu & (\oplus_{i<k} A_{p'(i)}) \oplus (A_n \oplus (\oplus_{i\in[k,n-1)} A_{p'(i)})) \quad \text{(E-UNION-COMM)} \\ & \simeq_\mu & \oplus_{i<n} A_{p(i)} \quad \text{Lem. B.0.14} \end{array}$$

□

Lema B.0.16. *Sea $J_m = \langle J, m\rangle$ un multi-conjunto[1] finito tal que $J \subseteq [0,n)$, entonces $\oplus_{i<n} A_i \simeq_\mu (\oplus_{i<n} A_i) \oplus (\oplus_{i\in J_m} A_j)$.*

Demostración. Por inducción en $\#(J_m)$.

- $\#(J_m) = 0$. Este caso es inmediato por Lem. B.0.14. Notar que (E-REFL) no es suficiente porque ambos lados de la equivalencia pueden estar asociados de diferentemente.
- $\#(J_m) > 0$. Sea $k \in J_m$, luego

$$\begin{array}{lll} (\oplus_{i<n} A_i) \oplus (\oplus_{i\in J_m} A_j) & \simeq_\mu & ((\oplus_{i<n} A_i) \oplus (\oplus_{\substack{i\in J_m \\ i\neq k}} A_j)) \oplus A_k \quad \text{Lem. B.0.15} \\ & \simeq_\mu & (\oplus_{\substack{i<n \\ i\neq k}} A_i) \oplus A_k \quad \textit{h.i.} \\ & \simeq_\mu & (\oplus_{\substack{i<n \\ i\neq k}} A_i) \oplus (A_k \oplus A_k) \quad \text{Lem. B.0.15} \\ & \simeq_\mu & (\oplus_{i<n} A_i) \oplus A_k \quad \text{(E-UNION-IDEM)} \\ & \simeq_\mu & \oplus_{i<n} A_i \quad \text{Lem. B.0.15} \end{array}$$

□

Lema 3.3.5. *Sean $A = \oplus_{i<n} A_i$, $B = \oplus_{j<m} B_j$ μ-tipos contractivos con funciones $f : [0,n) \to [0,m)$ y $g : [0,m) \to [0,n)$ tal que $(A_i \simeq_\mu B_{f(i)})_{i<n}$ y $(A_{g(j)} \simeq_\mu B_j)_{j<m}$. Entonces, $A \simeq_\mu B$.*

Demostración. Es inmediato ver que para todo multi-conjunto de índices $I \subseteq [0,n)$ vale $\oplus_{i\in I} A_i \simeq_\mu \oplus_{i\in I} B_{f(i)}$, aplicando (E-UNION) tantas veces como sea necesario y apelando eventualmente al Lem. B.0.14. Del mismo modo, $\oplus_{j\in J} B_j \simeq_\mu \oplus_{j\in J} A_{g(j)}$ para todo $J \subseteq [0,m)$. Considerar entonces los multi-conjuntos de índices

$$\begin{array}{rcl} I & \triangleq & \{i \mid i < n, i \in \mathsf{img}(g)\} \\ I' & \triangleq & \{i \mid i < n, i \notin \mathsf{img}(g)\} \\ G & \triangleq & \{g(j) \mid j < m\} \\ F & \triangleq & \{f(i) \mid i < n, i \notin \mathsf{img}(g)\} \end{array}$$

Notar que, por definición, I e I' no contienen elementos repetidos, y

$$G = I \cup G' \tag{B.1}$$

donde $G' \subseteq I$ simplemente contiene los elementos repetidos de G. Por otro lado

$$F \subseteq [0,m) \tag{B.2}$$

[1] Un *multi-conjunto* es un par $\mathcal{M} = \langle \mathcal{X}, m\rangle$ donde $\mathcal{X}$ es el *conjunto subyacente* de $\mathcal{M}$ y $m : \mathcal{X} \to \mathbb{N}$ es su *función de multiplicidad*. Usualmente se denota $\mathcal{M}$ directamente con $\mathcal{X}$ cuando no hay ambigüedad o el significado es claro por contexto.

Finalmente, se concluye apelando a los lemas presentados anteriormente

$$\begin{array}{lcll} A & = & \oplus_{i<n} A_i & \\ & \simeq_\mu & (\oplus_{i\in I} A_i) \oplus (\oplus_{i\in I'} A_i) & \text{Lem. B.0.15} \\ & \simeq_\mu & ((\oplus_{i\in I} A_i) \oplus (\oplus_{i\in G'} A_i)) \oplus (\oplus_{i\in I'} A_i) & \text{Lem. B.0.16 con (B.1)} \\ & = & (\oplus_{i\in G} A_i) \oplus (\oplus_{i\in I'} A_i) & \\ & = & (\oplus_{j<m} A_{g(j)}) \oplus (\oplus_{i\in I'} A_i) & \\ & \simeq_\mu & (\oplus_{j<m} B_j) \oplus (\oplus_{i\in I'} B_{f(i)}) & \\ & = & (\oplus_{j<m} B_j) \oplus (\oplus_{j\in F} B_j) & \\ & \simeq_\mu & \oplus_{j<m} B_j & \text{Lem. B.0.16 con (B.2)} \\ & = & B & \end{array}$$

□

Invertibilidad del sub-tipado

Lema 3.3.9.

1. $\{V \setminus \mathcal{B}\}V = \mathcal{B}$.
2. $\{V \setminus \mathcal{B}\}a = a$ *si* $V \neq a \in \mathcal{V} \cup \mathcal{C}$.
3. $\{V \setminus \mathcal{B}\}(\mathcal{A}_0 \star \mathcal{A}_1) = \{V \setminus \mathcal{B}\}\mathcal{A}_0 \star \{V \setminus \mathcal{B}\}\mathcal{A}_1$ *para* $\star \in \{@, \supset, \oplus\}$.

Demostración. Los tres casos son por análisis de las posiciones definidas en el tipo sustituido.

1. La única posición definida en V es ϵ. Luego, para toda p en $\mathcal{B}$ se tiene
$$(\{V \setminus \mathcal{B}\}V)(\mathsf{p}) = (\{V \setminus \mathcal{B}\}V)(\epsilon\mathsf{p}) = \mathcal{B}(\mathsf{p})$$
2. La única posición definida en $a \neq V$ es ϵ, por lo que se tiene $(\{V \setminus \mathcal{B}\}a)(\epsilon) = a(\epsilon) = a$. Toda otra posición está indefinida.
3. Aquí se tiene $\mathcal{A} = \mathcal{A}_0 \star \mathcal{A}_1$ con $\star \in \{@, \supset, \oplus\}$. Sea p una posición definida en $\mathcal{A}$.
   - $\mathsf{p} = \epsilon$. Luego,
   $$(\{V \setminus \mathcal{B}\}(\mathcal{A}_0 \star \mathcal{A}_1))(\epsilon) = (\mathcal{A}_0 \star \mathcal{A}_1)(\epsilon) = \star = (\{V \setminus \mathcal{B}\}\mathcal{A}_0 \star \{V \setminus \mathcal{B}\}\mathcal{A}_1)(\epsilon)$$
   - $\mathsf{p} = i\mathsf{q}$ con $i \in [0, 1]$. Hay dos casos posibles
     a) $\mathcal{A}(\mathsf{p}) \neq V$. Luego, $\mathcal{A}_i(\mathsf{q}) \neq V$ y se tiene
     $$\begin{array}{lcll} (\{V \setminus \mathcal{B}\}(\mathcal{A}_0 \star \mathcal{A}_1))(\mathsf{p}) & = & (\mathcal{A}_0 \star \mathcal{A}_1)(i\mathsf{q}) & \text{Def. 3.3.8} \\ & = & \mathcal{A}_i(\mathsf{q}) & \\ & = & (\{V \setminus \mathcal{B}\}\mathcal{A}_i)(\mathsf{q}) & \text{Def. 3.3.8} \\ & = & (\{V \setminus \mathcal{B}\}\mathcal{A}_0 \star \{V \setminus \mathcal{B}\}\mathcal{A}_1)(\mathsf{p}) & \end{array}$$
     b) $\mathcal{A}(\mathsf{p}) = V$. Luego, $\mathcal{A}_i(\mathsf{q}) = V$ y, por Def. 3.3.8, para toda posición q' en $\mathcal{B}$
     $$\begin{array}{lcl} (\{V \setminus \mathcal{B}\}(\mathcal{A}_0 \star \mathcal{A}_1))(\mathsf{p}\mathsf{q}') & = & \mathcal{B}(\mathsf{q}') \\ & = & (\{V \setminus \mathcal{B}\}\mathcal{A}_i)(\mathsf{q}\mathsf{q}') \\ & = & (\{V \setminus \mathcal{B}\}\mathcal{A}_0 \star \{V \setminus \mathcal{B}\}\mathcal{A}_1)(\mathsf{p}\mathsf{q}') \end{array}$$

□

Equivalencia de tipos infinitos

Lema 3.3.13. *$\simeq_{\mathfrak{T}}$ es una relación de equivalencia (*i.e. *reflexiva, simétrica y transitiva).*

Demostración. Las tres propiedades se prueban mostrando que los conjuntos que las definen son $\Phi_{\simeq_{\mathfrak{T}}}$-densos (*cf.* Def. 1.2.2 (2)). Luego, se concluye por el principio de co-inducción (*cf.* Cor. 1.2.5 (2)) que éstas valen para $\simeq_{\mathfrak{T}}$.

Reflexividad Sean $\mathit{Refl} \triangleq \{\langle \mathcal{A}, \mathcal{A}\rangle \mid \mathcal{A} \in \mathfrak{T}\}$ y el par $\langle \mathcal{A}, \mathcal{A}\rangle \in \mathit{Refl}$. Se procede analizando la forma de $\mathcal{A}$:

- $\mathcal{A} = a$. Inmediato pues $\langle a, a\rangle \in \Phi_{\simeq_{\mathfrak{T}}}(\mathit{Refl})$ para todo $a \in \mathcal{V} \cup \mathcal{C}$.
- $\mathcal{A} = \mathcal{D} \mathbin{@} \mathcal{A}'$. Por definición $\langle \mathcal{D}, \mathcal{D}\rangle, \langle \mathcal{A}', \mathcal{A}'\rangle \in \mathit{Refl}$. Luego, $\langle \mathcal{A}, \mathcal{A}\rangle \in \Phi_{\simeq_{\mathfrak{T}}}(\mathit{Refl})$.
- $\mathcal{A} = \mathcal{A}' \supset \mathcal{A}''$. Del mismo modo, se tiene $\langle \mathcal{A}', \mathcal{A}'\rangle, \langle \mathcal{A}'', \mathcal{A}''\rangle \in \mathit{Refl}$, por lo tanto $\langle \mathcal{A}, \mathcal{A}\rangle \in \Phi_{\simeq_{\mathfrak{T}}}(\mathit{Refl})$.
- $\mathcal{A} = \oplus_{i<n}\mathcal{A}_i$ with $\mathcal{A}_i \neq \oplus$ para todo $i < n$ con $n > 1$. Luego, como $\langle \mathcal{A}_i, \mathcal{A}_i\rangle \in \mathit{Refl}$ y $n + n > 2$, se concluye $\langle \oplus_{i<n}\mathcal{A}_i, \oplus_{i<n}\mathcal{A}_i\rangle \in \Phi_{\simeq_{\mathfrak{T}}}(\mathit{Refl})$ tomando $f = g = id$ (la función identidad).

Simetría Sea $\mathit{Symm}(\mathcal{X}) \triangleq \{\langle \mathcal{B}, \mathcal{A}\rangle \mid \langle \mathcal{A}, \mathcal{B}\rangle \in \mathcal{X}\}$. Se muestra entonces que $\mathit{Symm}(\simeq_{\mathfrak{T}})$ es $\Phi_{\simeq_{\mathfrak{T}}}$-denso (*i.e.* $\mathit{Symm}(\simeq_{\mathfrak{T}}) \subseteq \Phi_{\simeq_{\mathfrak{T}}}(\mathit{Symm}(\simeq_{\mathfrak{T}}))$) para concluir $\mathit{Symm}(\simeq_{\mathfrak{T}}) \subseteq \simeq_{\mathfrak{T}}$ por Cor. 1.2.5 (2).

Sea $\langle \mathcal{A}, \mathcal{B}\rangle \in \mathit{Symm}(\simeq_{\mathfrak{T}})$, entonces $\langle \mathcal{B}, \mathcal{A}\rangle \in \simeq_{\mathfrak{T}} = \Phi_{\simeq_{\mathfrak{T}}}(\simeq_{\mathfrak{T}})$. Siguiendo lo remarcado en la Nota 3.3.7 basta considerar tipos unión-maximal

$$\begin{array}{rcl} \mathcal{A} = \oplus_{i<n}\mathcal{A}_i & \text{con} & \mathcal{A}_i \neq \oplus, i < n \\ \mathcal{B} = \oplus_{j<m}\mathcal{B}_j & \text{con} & \mathcal{B}_j \neq \oplus, j < m \end{array}$$

por lo que hay solo dos casos a analizar

1. Si $n = m = 1$, entonces ambos tipos $\mathcal{A}$ y $\mathcal{B}$ son no-unión. Luego, se analiza la forma de $\mathcal{B}$
   - $\mathcal{B} = a$. Luego $\mathcal{A} = a$ por definición de $\Phi_{\simeq_{\mathfrak{T}}}$ y el resultado es inmediato, pues $\langle a, a\rangle \in \Phi_{\simeq_{\mathfrak{T}}}(\mathit{Symm}(\simeq_{\mathfrak{T}}))$ para todo $a \in \mathcal{V} \cup \mathcal{C}$.
   - $\mathcal{B} = \mathcal{D}'@\mathcal{B}'$. Nuevamente por definición, se tiene $\mathcal{A} = \mathcal{D}@\mathcal{A}'$ con $\langle \mathcal{D}', \mathcal{D}\rangle, \langle \mathcal{B}', \mathcal{A}'\rangle \in \simeq_{\mathfrak{T}}$. Luego, $\langle \mathcal{D}, \mathcal{D}'\rangle, \langle \mathcal{A}', \mathcal{B}'\rangle \in \mathit{Symm}(\simeq_{\mathfrak{T}})$ y se concluye $\langle \mathcal{A}, \mathcal{B}\rangle \in \Phi_{\simeq_{\mathfrak{T}}}(\mathit{Symm}(\simeq_{\mathfrak{T}}))$.
   - $\mathcal{B} = \mathcal{B}' \supset \mathcal{B}''$. Luego, $\mathcal{A} = \mathcal{A}' \supset \mathcal{A}''$ con $\langle \mathcal{B}', \mathcal{A}'\rangle, \langle \mathcal{B}'', \mathcal{A}''\rangle \in \simeq_{\mathfrak{T}}$. Por lo tanto $\langle \mathcal{A}', \mathcal{B}'\rangle, \langle \mathcal{A}'', \mathcal{B}''\rangle \in \mathit{Symm}(\simeq_{\mathfrak{T}})$ y se concluye $\langle \mathcal{A}, \mathcal{B}\rangle \in \Phi_{\simeq_{\mathfrak{T}}}(\mathit{Symm}(\simeq_{\mathfrak{T}}))$.
2. Si no, se tiene $n + m > 2$ y solo aplica la regla (E-UNION-T). Luego,

   $$\begin{array}{l} \exists g : [0, m) \to [0, n) \text{ t.q. } \langle \mathcal{B}_j, \mathcal{A}_{g(j)}\rangle \in \simeq_{\mathfrak{T}} \text{ para todo } j < m \\ \exists f : [0, n) \to [0, m) \text{ t.q. } \langle \mathcal{B}_{f(i)}, \mathcal{A}_i\rangle \in \simeq_{\mathfrak{T}} \text{ para todo } i < n \end{array}$$

   Aplicando simetría se obtiene $\langle \mathcal{A}_i, \mathcal{B}_{f(i)}\rangle, \langle \mathcal{A}_{g(j)}, \mathcal{B}_j\rangle \in \mathit{Symm}(\simeq_{\mathfrak{T}})$ para todo $i < n, j < m$. Luego, $\langle \mathcal{A}, \mathcal{B}\rangle \in \Phi_{\simeq_{\mathfrak{T}}}(\mathit{Symm}(\simeq_{\mathfrak{T}}))$.

Transitividad Sea $\mathit{Trans}(\mathcal{X}) \triangleq \{\langle \mathcal{A}, \mathcal{B}\rangle \mid \exists \mathcal{C} \in \mathfrak{T}.\langle \mathcal{A}, \mathcal{C}\rangle, \langle \mathcal{C}, \mathcal{B}\rangle \in \mathcal{X}\}$. Como en el caso anterior, se muestra $\mathit{Trans}(\simeq_{\mathfrak{T}}) \subseteq \Phi_{\simeq_{\mathfrak{T}}}(\mathit{Trans}(\simeq_{\mathfrak{T}}))$ y se concluye por Cor. 1.2.5 (2). Sea $\langle \mathcal{A}, \mathcal{B}\rangle \in \mathit{Trans}(\simeq_{\mathfrak{T}})$, luego existe $\mathcal{C} \in \mathfrak{T}$ tal que $\langle \mathcal{A}, \mathcal{C}\rangle, \langle \mathcal{C}, \mathcal{B}\rangle \in \simeq_{\mathfrak{T}} = \Phi_{\simeq_{\mathfrak{T}}}(\simeq_{\mathfrak{T}})$. Nuevamente se apela a la Nota 3.3.7 y se consideran tipos unión-maximal

$$\begin{array}{rcl} \mathcal{A} = \oplus_{i<n}\mathcal{A}_i & \text{con} & \mathcal{A}_i \neq \oplus, i < n \\ \mathcal{B} = \oplus_{j<m}\mathcal{B}_j & \text{con} & \mathcal{B}_j \neq \oplus, j < m \\ \mathcal{C} = \oplus_{k<l}\mathcal{C}_k & \text{con} & \mathcal{C}_k \neq \oplus, k < l \end{array}$$

1. Si $n = m = l = 1$ (*i.e.* los tres tipos son no-unión) se procede analizando la forma de $\mathcal{C}$:
   - $\mathcal{C} = a$. Por definición de $\Phi_{\simeq_\mathfrak{T}}$, $\mathcal{A} = a$ y $\mathcal{B} = a$. Luego, $\langle \mathcal{A}, \mathcal{B} \rangle = \langle a, a \rangle \in \Phi_{\simeq_\mathfrak{T}}(\mathit{Trans}(\simeq_\mathfrak{T}))$ trivialmente.
   - $\mathcal{C} = \mathcal{D}'' \mathbin{@} \mathcal{C}'$. Por definición de $\Phi_{\simeq_\mathfrak{T}}$ nuevamente, $\mathcal{A} = \mathcal{D} \mathbin{@} \mathcal{A}'$ y $\mathcal{B} = \mathcal{D}' \mathbin{@} \mathcal{B}'$ con $\langle \mathcal{D}, \mathcal{D}'' \rangle, \langle \mathcal{A}', \mathcal{C}' \rangle, \langle \mathcal{D}'', \mathcal{D}' \rangle, \langle \mathcal{C}', \mathcal{B}' \rangle \in \simeq_\mathfrak{T}$. Luego, $\langle \mathcal{D}, \mathcal{D}' \rangle, \langle \mathcal{A}', \mathcal{B}' \rangle \in \mathit{Trans}(\simeq_\mathfrak{T})$ y se concluye $\langle \mathcal{A}, \mathcal{B} \rangle \in \Phi_{\simeq_\mathfrak{T}}(\mathit{Trans}(\simeq_\mathfrak{T}))$.
   - $\mathcal{C} = \mathcal{C}' \supset \mathcal{C}''$. Del mismo modo, se tiene $\mathcal{A} = \mathcal{A}' \supset \mathcal{A}''$ y $\mathcal{B} = \mathcal{B}' \supset \mathcal{B}''$ con $\langle \mathcal{A}', \mathcal{C}' \rangle, \langle \mathcal{A}'', \mathcal{C}'' \rangle, \langle \mathcal{C}', \mathcal{B}' \rangle, \langle \mathcal{C}'', \mathcal{B}'' \rangle \in \simeq_\mathfrak{T}$. Luego, $\langle \mathcal{A}', \mathcal{B}' \rangle, \langle \mathcal{A}'', \mathcal{B}'' \rangle \in \mathit{Trans}(\simeq_\mathfrak{T})$, por lo que se concluye $\langle \mathcal{A}, \mathcal{B} \rangle \in \Phi_{\simeq_\mathfrak{T}}(\mathit{Trans}(\simeq_\mathfrak{T}))$.
2. Si no (*i.e.* $n + m + l > 3$), hay tres situaciones a considerar *a)* $n + l > 2$ y $m + l > 2$; *b)* $n > 1$ y $m = l = 1$; o *c)* $m > 1$ y $n = l = 1$. En cuanto a las reglas aplicadas para derivar $\mathcal{A} \simeq_\mathfrak{T} \mathcal{C}$ y $\mathcal{C} \simeq_\mathfrak{T} \mathcal{B}$, en $(2a)$ solo es posible usar (E-UNION-T) en ambos casos, mientras que en $(2b)$ y $(2c)$ se tiene (E-UNION-T) de un lado y cualquiera de las otras tres reglas del otro ((E-REFL-T), (E-COMP-T), (E-FUNC-T)). Notar que estos últimos dos casos son simétricos, por lo que solo se analizan $(2a)$ y $(2b)$ a continuación:

   a) $n + l > 2$ y $m + l > 2$. Por definición de $\Phi_{\simeq_\mathfrak{T}}$

   $$\begin{aligned} \exists f : [0,n) \to [0,l) \text{ t.q. } & \langle \mathcal{A}_i, \mathcal{C}_{f(i)} \rangle \in \simeq_\mathfrak{T} \text{ para todo } i < n \\ \exists g : [0,l) \to [0,n) \text{ t.q. } & \langle \mathcal{A}_{g(k)}, \mathcal{C}_k \rangle \in \simeq_\mathfrak{T} \text{ para todo } k < l \\ \exists f' : [0,l) \to [0,m) \text{ t.q. } & \langle \mathcal{C}_k, \mathcal{B}_{f'(k)} \rangle \in \simeq_\mathfrak{T} \text{ para todo } k < l \\ \exists g' : [0,m) \to [0,l) \text{ t.q. } & \langle \mathcal{C}_{g'(j)}, \mathcal{B}_j \rangle \in \simeq_\mathfrak{T} \text{ para todo } j < m \end{aligned}$$

   Luego, se tiene $\langle \mathcal{A}_i, \mathcal{C}_{f(i)} \rangle, \langle \mathcal{C}_{f(i)}, \mathcal{B}_{f'(f(i))} \rangle \in \simeq_\mathfrak{T}$ para todo $i < n$, y también se tiene $\langle \mathcal{A}_{g(g'(j))}, \mathcal{C}_{g'(j)} \rangle, \langle \mathcal{C}_{g'(j)}, \mathcal{B}_j \rangle \in \simeq_\mathfrak{T}$ para todo $j < m$.
   Nuevamente hay dos posibles situaciones. Si $n = m = 1$ (es decir $l > 1$) es necesariamente el caso en que $\langle \mathcal{A}, \mathcal{C}_{f(1)} \rangle, \langle \mathcal{C}_{f(1)}, \mathcal{B} \rangle \in \simeq_\mathfrak{T}$ con los tres tipos no-unión. Luego, se puede concluir con el mismo análisis hecho para el caso (1), $\langle \mathcal{A}, \mathcal{B} \rangle \in \Phi_{\simeq_\mathfrak{T}}(\mathit{Trans}(\simeq_\mathfrak{T}))$. Si no (*i.e.* $n + m > 2$), se toma $f'' = f' \circ f : [0,n) \to [0,m)$ para obtener $\langle \mathcal{A}_i, \mathcal{B}_{f''(i)} \rangle \in \mathit{Trans}(\simeq_\mathfrak{T})$. Análogamente, $\langle \mathcal{A}_{g''(j)}, \mathcal{B}_j \rangle \in \mathit{Trans}(\simeq_\mathfrak{T})$ para todo $j < m$ con $g'' = g \circ g' : [0,m) \to [0,n)$. Se concluye con la regla (E-UNION-T), $\langle \mathcal{A}, \mathcal{B} \rangle \in \Phi_{\simeq_\mathfrak{T}}(\mathit{Trans}(\simeq_\mathfrak{T}))$.

   b) $n > 1$ y $m = l = 1$. Luego, por definición de $\Phi_{\simeq_\mathfrak{T}}$, $f : [0,n) \to [0,1)$ es una función constante y se tiene $\langle \mathcal{A}_i, \mathcal{C} \rangle \in \simeq_\mathfrak{T}$ para todo $i < n$. Por otro lado se tiene $\langle \mathcal{C}, \mathcal{B} \rangle \in \simeq_\mathfrak{T}$ por hipótesis. Aplicando se obtiene $\langle \mathcal{A}_i, \mathcal{B} \rangle \in \mathit{Trans}(\simeq_\mathfrak{T})$ y se concluye con la misma función constante f, $\langle \mathcal{A}, \mathcal{B} \rangle \in \Phi_{\simeq_\mathfrak{T}}(\mathit{Trans}(\simeq_\mathfrak{T}))$.

□

Lema 3.3.14. *Sean $\mathcal{A}, \mathcal{B} \in \mathfrak{T}$ tipos no-unión tal que $\mathcal{A} \simeq_\mathfrak{T} \mathcal{B}$.*

1. *Si $\mathcal{A} = a$, entonces $\mathcal{B} = a$.*
2. *Si $\mathcal{A} = \mathcal{D} \mathbin{@} \mathcal{A}'$, entonces $\mathcal{B} = \mathcal{D}' \mathbin{@} \mathcal{B}'$ con $\mathcal{D} \simeq_\mathfrak{T} \mathcal{D}'$ y $\mathcal{A}' \simeq_\mathfrak{T} \mathcal{B}'$.*
3. *Si $\mathcal{A} = \mathcal{A}' \supset \mathcal{A}''$, entonces $\mathcal{B} = \mathcal{B}' \supset \mathcal{B}''$ con $\mathcal{A}' \simeq_\mathfrak{T} \mathcal{B}'$ y $\mathcal{A}'' \simeq_\mathfrak{T} \mathcal{B}''$.*

Demostración. Inmediato de la Def. 3.3.12. Notar que hay solo una regla aplicable en cada caso. □

Lema 3.3.15. *Sean $\mathcal{A} \simeq_\mathfrak{T} \mathcal{A}'$ y $\mathcal{B} \simeq_\mathfrak{T} \mathcal{B}'$. Luego, $\{V \setminus \mathcal{B}\}\mathcal{A} \simeq_\mathfrak{T} \{V \setminus \mathcal{B}'\}\mathcal{A}'$.*

Demostración. Sea $\mathcal{S} \triangleq \{\langle \{V \setminus \mathcal{B}\}\mathcal{A}, \{V \setminus \mathcal{B}'\}\mathcal{A}'\rangle \mid \mathcal{A} \simeq_{\mathfrak{T}} \mathcal{A}', \mathcal{B} \simeq_{\mathfrak{T}} \mathcal{B}'\}$. Se muestra que $\mathcal{S} \cup \simeq_{\mathfrak{T}}$ es $\Phi_{\simeq_{\mathfrak{T}}}$-denso.

Sea $\langle \mathcal{C}, \mathcal{C}'\rangle \in \mathcal{S} \cup \simeq_{\mathfrak{T}}$. Si $\langle \mathcal{C}, \mathcal{C}'\rangle \in \simeq_{\mathfrak{T}}$ el resultado es inmediato por monotonía de $\Phi_{\simeq_{\mathfrak{T}}}$, pues $\simeq_{\mathfrak{T}} = \Phi_{\simeq_{\mathfrak{T}}}(\simeq_{\mathfrak{T}}) \subseteq \Phi_{\simeq_{\mathfrak{T}}}(\mathcal{S} \cup \simeq_{\mathfrak{T}})$. Basta analizar entonces el caso $\langle \mathcal{C}, \mathcal{C}'\rangle \in \mathcal{S}$, *i.e.* $\mathcal{C} = \{V \setminus \mathcal{B}\}\mathcal{A}$ y $\mathcal{C}' = \{V \setminus \mathcal{B}'\}\mathcal{A}'$ con $\mathcal{A} \simeq_{\mathfrak{T}} \mathcal{A}'$ y $\mathcal{B} \simeq_{\mathfrak{T}} \mathcal{B}'$. Sean (por Nota 3.3.7)

$$\begin{array}{rcl} \mathcal{A} = \bigoplus_{i<n}\mathcal{A}_i & \text{con} & \mathcal{A}_i \neq \oplus, i < n \\ \mathcal{A}' = \bigoplus_{j<m}\mathcal{A}'_j & \text{con} & \mathcal{A}'_j \neq \oplus, j < m \end{array}$$

1. Si $n = m = 1$ (*i.e.* $\mathcal{A}, \mathcal{A}' \neq \oplus$), se analizar la forma de $\mathcal{A}$

   - $\mathcal{A} = a$. Por Lem. 3.3.14 (1), $\mathcal{A}' = a$ y se tiene dos posibles casos. Si $a \neq V$, por Lem. 3.3.9 (2), $\mathcal{C} = a = \mathcal{C}'$. Si no, por Lem. 3.3.9 (1), $\mathcal{C} = \mathcal{B} \simeq_{\mathfrak{T}} \mathcal{B}' = \mathcal{C}'$. Ambos casos son inmediatos por definición de $\simeq_{\mathfrak{T}} \subseteq \Phi_{\simeq_{\mathfrak{T}}}(\mathcal{S} \cup \simeq_{\mathfrak{T}})$.
   - $\mathcal{A} = \mathcal{D} \mathbin{@} \mathcal{A}_1$. Por Lem. 3.3.14 (2), $\mathcal{A}' = \mathcal{D}' \mathbin{@} \mathcal{A}'_1$ con $\mathcal{D} \simeq_{\mathfrak{T}} \mathcal{D}'$ y $\mathcal{A}_1 \simeq_{\mathfrak{T}} \mathcal{A}'_1$. Luego, por definición de $\mathcal{S}$, $\langle \{V \setminus \mathcal{B}\}\mathcal{D}, \{V \setminus \mathcal{B}'\}\mathcal{D}'\rangle$ y $\langle \{V \setminus \mathcal{B}\}\mathcal{A}_1, \{V \setminus \mathcal{B}'\}\mathcal{A}'_1\rangle \in \mathcal{S} \cup \simeq_{\mathfrak{T}}$. Finalmente, $\langle \mathcal{C}, \mathcal{C}'\rangle \in \Phi_{\simeq_{\mathfrak{T}}}(\mathcal{S} \cup \simeq_{\mathfrak{T}})$ pues, por Lem. 3.3.9 (3),

   $$\begin{array}{rcl} \mathcal{C} & = & \{V \setminus \mathcal{B}\}(\mathcal{D} \mathbin{@} \mathcal{A}_1) \;=\; \{V \setminus \mathcal{B}\}\mathcal{D} \mathbin{@} \{V \setminus \mathcal{B}\}\mathcal{A}_1 \\ \mathcal{C}' & = & \{V \setminus \mathcal{B}'\}(\mathcal{D}' \mathbin{@} \mathcal{A}'_1) = \{V \setminus \mathcal{B}'\}\mathcal{D}' \mathbin{@} \{V \setminus \mathcal{B}'\}\mathcal{A}'_1 \end{array}$$

   - $\mathcal{A} = \mathcal{A}_0 \supset \mathcal{A}_1$. Al igual que el caso anterior, por Lem. 3.3.14 (3), $\mathcal{A}' = \mathcal{A}'_0 \supset \mathcal{A}'_1$ con $\mathcal{A}_0 \simeq_{\mathfrak{T}} \mathcal{A}'_0$ y $\mathcal{A}_1 \simeq_{\mathfrak{T}} \mathcal{A}'_1$. Por definición se $\mathcal{S}$ se tiene $\langle \{V \setminus \mathcal{B}\}\mathcal{A}_0, \{V \setminus \mathcal{B}'\}\mathcal{A}'_0\rangle$ y $\langle \{V \setminus \mathcal{B}\}\mathcal{A}_1, \{V \setminus \mathcal{B}'\}\mathcal{A}'_1\rangle \in \mathcal{S} \cup \simeq_{\mathfrak{T}}$. Se concluye por Lem. 3.3.9 (3), $\langle \mathcal{C}, \mathcal{C}'\rangle \in \Phi_{\simeq_{\mathfrak{T}}}(\mathcal{S} \cup \simeq_{\mathfrak{T}})$.

2. Si $n + m > 2$, por (E-UNION-T) se tiene

   $$\begin{array}{l} \exists f : [0, n) \to [0, m) \text{ t.q. } \mathcal{A}_i \simeq_{\mathfrak{T}} \mathcal{A}'_{f(i)} \text{ para todo } i < n \\ \exists g : [0, m) \to [0, n) \text{ t.q. } \mathcal{A}_{g(j)} \simeq_{\mathfrak{T}} \mathcal{A}'_j \text{ para todo } j < m \end{array}$$

   Luego, $\langle \{V \setminus \mathcal{B}\}\mathcal{A}_i, \{V \setminus \mathcal{B}'\}\mathcal{A}'_{f(i)}\rangle$ y $\langle \{V \setminus \mathcal{B}\}\mathcal{A}_{g(j)}, \{V \setminus \mathcal{B}'\}\mathcal{A}'_j\rangle \in \mathcal{S} \cup \simeq_{\mathfrak{T}}$ para todo $i < n, j < m$. Nuevamente, se concluye por definición de $\Phi_{\simeq_{\mathfrak{T}}}$ apelando al Lem. 3.3.9 (3), $\langle \mathcal{C}, \mathcal{C}'\rangle \in \Phi_{\simeq_{\mathfrak{T}}}(\mathcal{S} \cup \simeq_{\mathfrak{T}})$.

□

Lema 3.3.16. $\mathcal{A} \simeq_{\mathfrak{T}} \mathcal{B}$ *sii* $\forall k \in \mathbb{N}. \mathcal{A}\rfloor_k \simeq_{\mathfrak{T}} \mathcal{B}\rfloor_k$.

Demostración. $\Rightarrow$) Se muestra que $\mathcal{S} \triangleq \{\langle \mathcal{A}\rfloor_k, \mathcal{B}\rfloor_k\rangle \mid \mathcal{A} \simeq_{\mathfrak{T}} \mathcal{B}, k \in \mathbb{N}\}$ es $\Phi_{\simeq_{\mathfrak{T}}}$-denso. Sea $\langle \mathcal{A}\rfloor_k, \mathcal{B}\rfloor_k\rangle \in \mathcal{S}$. Si $k = 0$, por Def. 3.3.10, $\mathcal{A}\rfloor_k = \varepsilon = \mathcal{B}\rfloor_k$ y trivialmente se tiene $\langle \varepsilon, \varepsilon\rangle \in \Phi_{\simeq_{\mathfrak{T}}}(\mathcal{S})$. Se analizan a continuación los casos donde $k > 0$, teniendo $\mathcal{A} \simeq_{\mathfrak{T}} \mathcal{B}$ por definición de $\mathcal{S}$. Apelando a la Nota 3.3.7 se consideran tipos unión-maximal

$$\begin{array}{rcl} \mathcal{A} = \bigoplus_{i<n}\mathcal{A}_i & \text{con} & \mathcal{A}_i \neq \oplus, i < n \\ \mathcal{B} = \bigoplus_{j<m}\mathcal{B}_j & \text{con} & \mathcal{B}_j \neq \oplus, j < m \end{array}$$

y se analizan por separado los casos donde tanto $\mathcal{A}$ como $\mathcal{B}$ son tipos no-unión.

- Si $n = m = 0$ se analiza la forma de $\mathcal{A}$:

  - $\mathcal{A} = a$. Por Lem. 3.3.14 (1), $\mathcal{B} = a$ y $a\rfloor_k = a$ para todo $k > 0$. Luego, se concluye por definición de $\Phi_{\simeq_{\mathfrak{T}}}$, $\langle a, a\rangle \in \Phi_{\simeq_{\mathfrak{T}}}(\mathcal{S})$.

- $\mathcal{A} = \mathcal{D} \mathbin{@} \mathcal{A}'$. Por Lem. 3.3.14 (2), $\mathcal{B} = \mathcal{D}' \mathbin{@} \mathcal{B}'$ con $\mathcal{D} \simeq_{\mathfrak{T}} \mathcal{D}'$ y $\mathcal{A}' \simeq_{\mathfrak{T}} \mathcal{B}'$. Por definición de $\mathcal{S}$, se tiene $\langle \mathcal{D}\rfloor_{k-1}, \mathcal{D}'\rfloor_{k-1}\rangle, \langle \mathcal{A}'\rfloor_{k-1}, \mathcal{B}'\rfloor_{k-1}\rangle \in \mathcal{S}$. Luego, se concluye con $\langle \mathcal{A}\rfloor_k, \mathcal{B}\rfloor_k\rangle = \langle \mathcal{D}\rfloor_{k-1} \mathbin{@} \mathcal{A}'\rfloor_{k-1}, \mathcal{D}'\rfloor_{k-1} \mathbin{@} \mathcal{B}'\rfloor_{k-1}\rangle \in \Phi_{\simeq_{\mathfrak{T}}}(\mathcal{S})$.
- $\mathcal{A} = \mathcal{A}' \supset \mathcal{A}''$. Por Lem. 3.3.14 (3), $\mathcal{B} = \mathcal{B}' \supset \mathcal{B}''$ con $\mathcal{A}' \simeq_{\mathfrak{T}} \mathcal{B}'$ y $\mathcal{A}'' \simeq_{\mathfrak{T}} \mathcal{B}''$. Por definición, se tienen $\langle \mathcal{A}'\rfloor_{k-1}, \mathcal{B}'\rfloor_{k-1}\rangle, \langle \mathcal{A}''\rfloor_{k-1}, \mathcal{B}''\rfloor_{k-1}\rangle \in \mathcal{S}$. Luego, se concluye con $\langle \mathcal{A}\rfloor_k, \mathcal{B}\rfloor_k\rangle = \langle \mathcal{A}'\rfloor_{k-1} \supset \mathcal{A}''\rfloor_{k-1}, \mathcal{B}'\rfloor_{k-1} \supset \mathcal{B}''\rfloor_{k-1}\rangle \in \Phi_{\simeq_{\mathfrak{T}}}(\mathcal{S})$.

- Si $n + m > 2$, por (E-UNION-T) se tiene

$$\exists f : [0, n) \to [0, m) \text{ t.q. } \mathcal{A}_i \simeq_{\mathfrak{T}} \mathcal{B}_{f(i)} \text{ para todo } i < n$$
$$\exists g : [0, m) \to [0, n) \text{ t.q. } \mathcal{A}_{g(j)} \simeq_{\mathfrak{T}} \mathcal{B}_j \text{ para todo } j < m$$

  Luego, por definición de $\mathcal{S}$, $\langle \mathcal{A}_i\rfloor_k, \mathcal{B}_{f(i)}\rfloor_k\rangle, \langle \mathcal{A}_{g(j)}\rfloor_k, \mathcal{B}_j\rfloor_k\rangle \in \mathcal{S}$ para todo $i < n$, $j < m$ y $k > 0$. Entonces, se concluye definición de $\Phi_{\simeq_{\mathfrak{T}}}$ apelando a la Nota 3.3.11, $\langle \mathcal{A}\rfloor_k, \mathcal{B}\rfloor_k\rangle \in \Phi_{\simeq_{\mathfrak{T}}}(\mathcal{S})$.

$\Leftarrow$) Para esta parte de la prueba se muestra que la relación $\mathcal{S} \triangleq \{\langle \mathcal{A}, \mathcal{B}\rangle \mid \forall k \in \mathbb{N}. \mathcal{A}\rfloor_k \simeq_{\mathfrak{T}} \mathcal{B}\rfloor_k\}$ es $\Phi_{\simeq_{\mathfrak{T}}}$-densa. Sea $\langle \mathcal{A}, \mathcal{B}\rangle \in \mathcal{S}$. Entonces, para todo $k \in \mathbb{N}$ se tiene $\mathcal{A}\rfloor_k \simeq_{\mathfrak{T}} \mathcal{B}\rfloor_k$. Nuevamente se consideran tipos unión-maximal, apelando a la Nota 3.3.7

$$\mathcal{A} = \oplus_{i<n}\mathcal{A}_i \quad \text{con} \quad \mathcal{A}_i \neq \oplus, i < n$$
$$\mathcal{B} = \oplus_{j<m}\mathcal{B}_j \quad \text{con} \quad \mathcal{B}_j \neq \oplus, j < m$$

- Si $n = m = 1$ (*i.e.* $\mathcal{A}, \mathcal{B} \neq \oplus$), se analiza la forma de $\mathcal{A}$:
  - $\mathcal{A} = a$. Entonces, $\mathcal{A}\rfloor_k = a$ para todo $k > 0$ y, por Lem. 3.3.14 (1), $\mathcal{B}\rfloor_k = a$. Por lo tanto, $\mathcal{B} = a$ y se concluye por definición de $\Phi_{\simeq_{\mathfrak{T}}}$, $\langle a, a\rangle \in \Phi_{\simeq_{\mathfrak{T}}}(\mathcal{S})$.
  - $\mathcal{A} = \mathcal{D} \mathbin{@} \mathcal{A}'$. Del mismo modo, se tiene $\mathcal{A}\rfloor_k = \mathcal{D}\rfloor_{k-1} \mathbin{@} \mathcal{A}'\rfloor_{k-1}$ para todo $k > 0$. Por Lem. 3.3.14 (2), $\mathcal{B}\rfloor_k = \mathcal{D}'_k \mathbin{@} \mathcal{B}'_k$ con $\mathcal{D}\rfloor_{k-1} \simeq_{\mathfrak{T}} \mathcal{D}'_k$ y $\mathcal{A}'\rfloor_{k-1} \simeq_{\mathfrak{T}} \mathcal{B}'_k$. Notar que para cada k se tienen sub-árboles $\mathcal{D}'_k$ y $\mathcal{B}'_k$ diferentes pero, dado que el Lem. 3.3.14 apela a la igualdad de árboles (no equivalencia) al determinar la forma de $\mathcal{B}$, es inmediato ver de la Def. 3.3.10 que $\mathcal{B} = \mathcal{D}' \mathbin{@} \mathcal{B}'$ con $\mathcal{D}'_k = \mathcal{D}'\rfloor_{k-1}$ y $\mathcal{B}'_k = \mathcal{B}'\rfloor_{k-1}$ para todo $k > 0$. Por lo tanto, $\mathcal{D}\rfloor_{k-1} \simeq_{\mathfrak{T}} \mathcal{D}'\rfloor_{k-1}$ y $\mathcal{A}'\rfloor_{k-1} \simeq_{\mathfrak{T}} \mathcal{B}'\rfloor_{k-1}$ para todo $k > 0$. Luego, por definición se tienen $\langle \mathcal{D}, \mathcal{D}'\rangle, \langle \mathcal{A}', \mathcal{B}'\rangle \in \mathcal{S}$ y $\langle \mathcal{D} \mathbin{@} \mathcal{A}', \mathcal{D}' \mathbin{@} \mathcal{B}'\rangle \in \Phi_{\simeq_{\mathfrak{T}}}(\mathcal{S})$.
  - $\mathcal{A} = \mathcal{A}' \supset \mathcal{A}''$. El análisis para este caso es similar al anterior. De $\mathcal{A}\rfloor_k = \mathcal{A}'\rfloor_{k-1} \supset \mathcal{A}''\rfloor_{k-1}$ se tiene $\mathcal{B} = \mathcal{B}' \supset \mathcal{B}''$ con $\mathcal{A}'\rfloor_{k-1} \simeq_{\mathfrak{T}} \mathcal{B}'\rfloor_{k-1}$ y $\mathcal{A}''\rfloor_{k-1} \simeq_{\mathfrak{T}} \mathcal{B}''\rfloor_{k-1}$ para todo $k > 0$, por Lem. 3.3.14 (3) y Def. 3.3.10. Luego, $\langle \mathcal{A}', \mathcal{B}'\rangle, \langle \mathcal{A}'', \mathcal{B}''\rangle \in \mathcal{S}$ y $\langle \mathcal{A}' \supset \mathcal{B}', \mathcal{A}'' \supset \mathcal{B}''\rangle \in \Phi_{\simeq_{\mathfrak{T}}}(\mathcal{S})$.
- Si $n+m > 2$, se tiene $\mathcal{A}\rfloor_k = \oplus_{i<n}(\mathcal{A}_i\rfloor_k)$ y $\mathcal{B}\rfloor_k = \oplus_{j<m}(\mathcal{B}_j\rfloor_k)$ para todo $k > 0$. De $\mathcal{A}\rfloor_k \simeq_{\mathfrak{T}} \mathcal{B}\rfloor_k$, por (E-UNION-T), se tiene

$$\exists f : [0, n) \to [0, m) \text{ t.q. } \mathcal{A}_i\rfloor_k \simeq_{\mathfrak{T}} \mathcal{B}_{f(i)}\rfloor_k \text{ para todo } i < n$$
$$\exists g : [0, m) \to [0, n) \text{ t.q. } \mathcal{A}_{g(j)}\rfloor_k \simeq_{\mathfrak{T}} \mathcal{B}_j\rfloor_k \text{ para todo } j < m$$

  Más aún, como $\mathcal{C}\rfloor_0 = \varepsilon$ pata todo $\mathcal{C} \in \mathfrak{T}$, set tiene $\mathcal{A}_i\rfloor_0 \simeq_{\mathfrak{T}} \mathcal{B}_{f(i)}\rfloor_0$ y $\mathcal{A}_{g(j)}\rfloor_0 \simeq_{\mathfrak{T}} \mathcal{B}_j\rfloor_0$ por reflexividad. Luego, $\mathcal{A}_i\rfloor_k \simeq_{\mathfrak{T}} \mathcal{B}_{f(i)}\rfloor_k$ y $\mathcal{A}_{g(j)}\rfloor_k \simeq_{\mathfrak{T}} \mathcal{B}_j\rfloor_k$ para todo $k \in \mathbb{N}$. Entonces, $\langle \mathcal{A}_i, \mathcal{B}_{f(i)}\rangle, \langle \mathcal{A}_{g(j)}, \mathcal{B}_j\rangle \in \mathcal{S}$ para todo $i < n$, $j < m$, y $\langle \mathcal{A}, \mathcal{B}\rangle \in \Phi_{\simeq_{\mathfrak{T}}}(\mathcal{S})$ por definición.

□

Sub-tipado de tipos infinitos

Lema 3.3.18. *$\preceq_{\mathfrak{T}}$ es un pre-orden (*i.e. *reflexivo y transitivo).*

Demostración. Similar a la presentada para el Lem. 3.3.13. Se muestra que los conjuntos que definen son las propiedades son $\Phi_{\preceq_{\mathfrak{T}}}$-densos (*cf.* Def. 1.2.2 (2)), para concluir por el principio de co-inducción (*cf.* Cor. 1.2.5 (2)).

Reflexividad Sean $\mathit{Refl} \triangleq \{\langle \mathcal{A}, \mathcal{A}\rangle \mid \mathcal{A} \in \mathfrak{T}\}$ y el par $\langle \mathcal{A}, \mathcal{A}\rangle \in \mathit{Refl}$. Se procede analizando la forma de $\mathcal{A}$:

- $\mathcal{A} = a$. Inmediato pues $\langle a, a\rangle \in \Phi_{\preceq_{\mathfrak{T}}}(\mathit{Refl})$ para todo $a \in \mathcal{V} \cup \mathcal{C}$.
- $\mathcal{A} = \mathcal{D} \mathbin{@} \mathcal{A}'$. Por definición $\langle \mathcal{D}, \mathcal{D}\rangle, \langle \mathcal{A}', \mathcal{A}'\rangle \in \mathit{Refl}$. Luego, $\langle \mathcal{A}, \mathcal{A}\rangle \in \Phi_{\preceq_{\mathfrak{T}}}(\mathit{Refl})$.
- $\mathcal{A} = \mathcal{A}' \supset \mathcal{A}''$. Del mismo modo, se tiene $\langle \mathcal{A}', \mathcal{A}'\rangle, \langle \mathcal{A}'', \mathcal{A}''\rangle \in \mathit{Refl}$, por lo tanto $\langle \mathcal{A}, \mathcal{A}\rangle \in \Phi_{\preceq_{\mathfrak{T}}}(\mathit{Refl})$.
- $\mathcal{A} = \oplus_{i<n}\mathcal{A}_i$ with $\mathcal{A}_i \neq \oplus$ para todo $i < n$ con $n > 1$. Luego, como $\langle \mathcal{A}_i, \mathcal{A}_i\rangle \in \mathit{Refl}$ y $n + n > 2$, se concluye $\langle \oplus_{i<n}\mathcal{A}_i, \oplus_{i<n}\mathcal{A}_i\rangle \in \Phi_{\preceq_{\mathfrak{T}}}(\mathit{Refl})$ tomando $f = id$ (la función identidad).

Transitividad Sea $\mathit{Trans}(\mathcal{X}) \triangleq \{\langle \mathcal{A}, \mathcal{B}\rangle \mid \exists \mathcal{C} \in \mathfrak{T}.\langle \mathcal{A}, \mathcal{C}\rangle, \langle \mathcal{C}, \mathcal{B}\rangle \in \mathcal{X}\}$. Como en el caso anterior, se muestra $\mathit{Trans}(\preceq_{\mathfrak{T}}) \subseteq \Phi_{\preceq_{\mathfrak{T}}}(\mathit{Trans}(\preceq_{\mathfrak{T}}))$ y se concluye por Cor. 1.2.5 (2). Sea $\langle \mathcal{A}, \mathcal{B}\rangle \in \mathit{Trans}(\preceq_{\mathfrak{T}})$, luego existe $\mathcal{C} \in \mathfrak{T}$ tal que $\langle \mathcal{A}, \mathcal{C}\rangle, \langle \mathcal{C}, \mathcal{B}\rangle \in \preceq_{\mathfrak{T}} = \Phi_{\preceq_{\mathfrak{T}}}(\preceq_{\mathfrak{T}})$. Nuevamente se apela a la Nota 3.3.7 y se consideran tipos unión-maximal

$$\begin{array}{rcl} \mathcal{A} = \oplus_{i<n}\mathcal{A}_i & \text{con} & \mathcal{A}_i \neq \oplus, i < n \\ \mathcal{B} = \oplus_{j<m}\mathcal{B}_j & \text{con} & \mathcal{B}_j \neq \oplus, j < m \\ \mathcal{C} = \oplus_{k<l}\mathcal{C}_k & \text{con} & \mathcal{C}_k \neq \oplus, k < l \end{array}$$

1. Si $n = m = l = 1$ (*i.e.* los tres tipos son no-unión) se procede analizando la forma de $\mathcal{C}$:
   - $\mathcal{C} = a$. Por definición de $\Phi_{\preceq_{\mathfrak{T}}}$, $\mathcal{A} = a$ y $\mathcal{B} = a$. Luego, $\langle \mathcal{A}, \mathcal{B}\rangle = \langle a, a\rangle \in \Phi_{\preceq_{\mathfrak{T}}}(\mathit{Trans}(\preceq_{\mathfrak{T}}))$ trivialmente.
   - $\mathcal{C} = \mathcal{D}'' \mathbin{@} \mathcal{C}'$. Por definición de $\Phi_{\preceq_{\mathfrak{T}}}$ nuevamente, $\mathcal{A} = \mathcal{D} \mathbin{@} \mathcal{A}'$ y $\mathcal{B} = \mathcal{D}' \mathbin{@} \mathcal{B}'$ con $\langle \mathcal{D}, \mathcal{D}''\rangle, \langle \mathcal{A}', \mathcal{C}'\rangle, \langle \mathcal{D}'', \mathcal{D}'\rangle, \langle \mathcal{C}', \mathcal{B}'\rangle \in \preceq_{\mathfrak{T}}$. Luego, $\langle \mathcal{D}, \mathcal{D}'\rangle, \langle \mathcal{A}', \mathcal{B}'\rangle \in \mathit{Trans}(\preceq_{\mathfrak{T}})$ y se concluye $\langle \mathcal{A}, \mathcal{B}\rangle \in \Phi_{\preceq_{\mathfrak{T}}}(\mathit{Trans}(\preceq_{\mathfrak{T}}))$.
   - $\mathcal{C} = \mathcal{C}' \supset \mathcal{C}''$. Del mismo modo, se tiene $\mathcal{A} = \mathcal{A}' \supset \mathcal{A}''$ y $\mathcal{B} = \mathcal{B}' \supset \mathcal{B}''$ con $\langle \mathcal{C}', \mathcal{A}'\rangle, \langle \mathcal{A}'', \mathcal{C}''\rangle, \langle \mathcal{B}', \mathcal{C}'\rangle, \langle \mathcal{C}'', \mathcal{B}''\rangle \in \preceq_{\mathfrak{T}}$. Luego, $\langle \mathcal{B}', \mathcal{A}'\rangle, \langle \mathcal{A}'', \mathcal{B}''\rangle \in \mathit{Trans}(\preceq_{\mathfrak{T}})$, por lo que se concluye $\langle \mathcal{A}, \mathcal{B}\rangle \in \Phi_{\preceq_{\mathfrak{T}}}(\mathit{Trans}(\preceq_{\mathfrak{T}}))$.
2. Si no (*i.e.* $n + m + l > 3$), hay tres situaciones a considerar *a)* $n + l > 2$ y $m + l > 2$; *b)* $n > 1$ y $m = l = 1$; o *c)* $m > 1$ y $n = l = 1$. En cuanto a las reglas aplicadas para derivar $\mathcal{A} \preceq_{\mathfrak{T}} \mathcal{C}$ y $\mathcal{C} \preceq_{\mathfrak{T}} \mathcal{B}$, en $(2a)$ solo es posible usar (S-UNION-T) en ambos casos, mientras que en $(2b)$ y $(2c)$ se tiene (S-UNION-T) de un lado y cualquiera de las otras tres reglas del otro ((S-REFL-T), (S-COMP-T), (S-FUNC-T)). Notar que estos últimos dos casos son simétricos, por lo que solo se analizan $(2a)$ y $(2b)$ a continuación:

   a) $n + l > 2$ y $m + l > 2$. Por definición de $\Phi_{\preceq_{\mathfrak{T}}}$

   $$\begin{array}{r} \exists f : [0, n) \to [0, l) \text{ t.q. } \langle \mathcal{A}_i, \mathcal{C}_{f(i)}\rangle \in \preceq_{\mathfrak{T}} \text{ para todo } i < n \\ \exists f' : [0, l) \to [0, m) \text{ t.q. } \langle \mathcal{C}_k, \mathcal{B}_{f'(k)}\rangle \in \preceq_{\mathfrak{T}} \text{ para todo } k < l \end{array}$$

   Luego, se tiene $\langle \mathcal{A}_i, \mathcal{C}_{f(i)}\rangle, \langle \mathcal{C}_{f(i)}, \mathcal{B}_{f'(f(i))}\rangle \in \preceq_{\mathfrak{T}}$ para todo $i < n$.

Nuevamente hay dos posibles situaciones. Si $n = m = 1$ (es decir $l > 1$) es necesariamente el caso en que $\langle \mathcal{A}, \mathcal{C}_{f(0)} \rangle, \langle \mathcal{C}_{f(0)}, \mathcal{B} \rangle \in \preceq_{\mathfrak{T}}$ con los tres tipos no-unión. Luego, se puede concluir con el mismo análisis hecho para el caso (1), $\langle \mathcal{A}, \mathcal{B} \rangle \in \Phi_{\preceq_{\mathfrak{T}}}(\mathit{Trans}(\preceq_{\mathfrak{T}}))$. Si no (*i.e.* $n + m > 2$), se toma $f'' = f' \circ f : [0, n) \to [0, m)$ para obtener $\langle \mathcal{A}_i, \mathcal{B}_{f''(i)} \rangle \in \mathit{Trans}(\preceq_{\mathfrak{T}})$. Finalmente, se concluye con la regla (S-UNION-T), $\langle \mathcal{A}, \mathcal{B} \rangle \in \Phi_{\preceq_{\mathfrak{T}}}(\mathit{Trans}(\preceq_{\mathfrak{T}}))$.

b) $n > 1$ y $m = l = 1$. Luego, por definición de $\Phi_{\preceq_{\mathfrak{T}}}$, $f : [0, n) \to [0, 1)$ es una función constante y se tiene $\langle \mathcal{A}_i, \mathcal{C} \rangle \in \preceq_{\mathfrak{T}}$ para todo $i < n$. Por otro lado se tiene $\langle \mathcal{C}, \mathcal{B} \rangle \in \preceq_{\mathfrak{T}}$ por hipótesis. Aplicando se obtiene $\langle \mathcal{A}_i, \mathcal{B} \rangle \in \mathit{Trans}(\preceq_{\mathfrak{T}})$ y se concluye con la misma función constante f, $\langle \mathcal{A}, \mathcal{B} \rangle \in \Phi_{\preceq_{\mathfrak{T}}}(\mathit{Trans}(\preceq_{\mathfrak{T}}))$.

□

Lema 3.3.19. *Sean $\mathcal{A}, \mathcal{B} \in \mathfrak{T}$ tipos no-unión tal que $\mathcal{A} \preceq_{\mathfrak{T}} \mathcal{B}$.*

1. *$\mathcal{A} = a$ sii $\mathcal{B} = a$.*
2. *$\mathcal{A} = \mathcal{D} \mathbin{@} \mathcal{A}'$ sii $\mathcal{B} = \mathcal{D}' \mathbin{@} \mathcal{B}'$, con $\mathcal{D} \preceq_{\mathfrak{T}} \mathcal{D}'$ y $\mathcal{A}' \preceq_{\mathfrak{T}} \mathcal{B}'$.*
3. *$\mathcal{A} = \mathcal{A}' \supset \mathcal{A}''$ sii $\mathcal{B} = \mathcal{B}' \supset \mathcal{B}''$, con $\mathcal{B}' \preceq_{\mathfrak{T}} \mathcal{A}'$ y $\mathcal{A}'' \preceq_{\mathfrak{T}} \mathcal{B}''$.*

Demostración. Inmediato de la Def. 3.3.17. Notar que hay solo una regla aplicable en cada caso. □

Lema 3.3.20. *Si $\mathcal{A} \simeq_{\mathfrak{T}} \mathcal{B}$, entonces $\mathcal{A} \preceq_{\mathfrak{T}} \mathcal{B}$.*

Demostración. Se muestra que $\simeq_{\mathfrak{T}} = \Phi_{\simeq_{\mathfrak{T}}}(\simeq_{\mathfrak{T}})$ es $\Phi_{\preceq_{\mathfrak{T}}}$-denso. Sea $\mathcal{A} \simeq_{\mathfrak{T}} \mathcal{B}$. Apelando a la Nota 3.3.7 se consideran tipos unión-maximal

$$\begin{array}{rcl} \mathcal{A} = \oplus_{i<n} \mathcal{A}_i & \text{con} & \mathcal{A}_i \neq \oplus, i < n \\ \mathcal{B} = \oplus_{j<m} \mathcal{B}_j & \text{con} & \mathcal{B}_j \neq \oplus, j < m \end{array}$$

y se tiene dos posibles casos:

1. Si $n = m = 1$, entonces tanto $\mathcal{A}$ como $\mathcal{B}$ son tipos no-unión. Basta analizar la forma de $\mathcal{A}$:
   - $\mathcal{A} = a$. Por Lem. 3.3.14 (1), $\mathcal{B} = a$ y el resultado es inmediato, pues $\langle a, a \rangle \in \Phi_{\preceq_{\mathfrak{T}}}(\simeq_{\mathfrak{T}})$ para todo $a \in \mathcal{V} \cup \mathcal{C}$.
   - $\mathcal{A} = \mathcal{D} \mathbin{@} \mathcal{A}'$. Por Lem. 3.3.14 (2), $\mathcal{B} = \mathcal{D}' \mathbin{@} \mathcal{B}'$ con $\mathcal{D} \simeq_{\mathfrak{T}} \mathcal{D}'$ y $\mathcal{A}' \simeq_{\mathfrak{T}} \mathcal{B}'$. Luego, por definición de $\Phi_{\preceq_{\mathfrak{T}}}$, $\langle \mathcal{D} \mathbin{@} \mathcal{A}', \mathcal{D}' \mathbin{@} \mathcal{B}' \rangle \in \Phi_{\preceq_{\mathfrak{T}}}(\simeq_{\mathfrak{T}})$.
   - $\mathcal{A} = \mathcal{A}' \supset \mathcal{A}''$. Del mismo modo, por Lem. 3.3.14 (3), $\mathcal{B} = \mathcal{B}' \supset \mathcal{B}''$ con $\mathcal{A}' \simeq_{\mathfrak{T}} \mathcal{B}'$ y $\mathcal{A}'' \simeq_{\mathfrak{T}} \mathcal{B}''$. Más aún, por simetría $\mathcal{B}' \simeq_{\mathfrak{T}} \mathcal{A}'$. Luego, por definición de $\Phi_{\preceq_{\mathfrak{T}}}$, $\langle \mathcal{A}' \supset \mathcal{A}'', \mathcal{A}'' \supset \mathcal{B}'' \rangle \in \Phi_{\preceq_{\mathfrak{T}}}(\simeq_{\mathfrak{T}})$.
2. Si no (*i.e.* $n + m > 2$), entonces aplica la regla (E-UNION-T) y

$$\begin{array}{l} \exists f : [0, n) \to [0, m) \text{ t.q. } \langle \mathcal{A}_i, \mathcal{B}_{f(i)} \rangle \in \simeq_{\mathfrak{T}} \text{ para todo } i < n \\ \exists g : [0, m) \to [0, n) \text{ t.q. } \langle \mathcal{A}_{g(j)}, \mathcal{B}_j \rangle \in \simeq_{\mathfrak{T}} \text{ para todo } j < m \end{array}$$

   Luego, se concluye con la misma función f, $\langle \mathcal{A}, \mathcal{B} \rangle \in \Phi_{\preceq_{\mathfrak{T}}}(\simeq_{\mathfrak{T}})$.

□

Lema 3.3.21. *$\mathcal{A} \preceq_{\mathfrak{T}} \mathcal{B}$ sii $\forall k \in \mathbb{N}. \mathcal{A}\rfloor_k \preceq_{\mathfrak{T}} \mathcal{B}\rfloor_k$.*

Demostración. $\Rightarrow$) Se muestra que $\mathcal{S} \triangleq \{\langle \mathcal{A}\rfloor_k, \mathcal{B}\rfloor_k\rangle \mid \mathcal{A} \preceq_{\mathfrak{T}} \mathcal{B}, k \in \mathbb{N}\}$ es $\Phi_{\preceq_{\mathfrak{T}}}$-denso. Sea $\langle \mathcal{A}\rfloor_k, \mathcal{B}\rfloor_k\rangle \in \mathcal{S}$. El caso $k = 0$ es inmediato, por lo que solo se analiza el $k > 0$. Apelando a la Nota 3.3.7 se consideran tipos unión-maximal

$$\begin{array}{rcl} \mathcal{A} = \oplus_{i<n}\mathcal{A}_i & \text{con} & \mathcal{A}_i \neq \oplus, i < n \\ \mathcal{B} = \oplus_{j<m}\mathcal{B}_j & \text{con} & \mathcal{B}_j \neq \oplus, j < m \end{array}$$

- Si $n = m = 1$ (*i.e.* $\mathcal{A}, \mathcal{B} \neq \oplus$), se analiza la forma de $\mathcal{A}$:
  - $\mathcal{A} = a$. Por Lem. 3.3.19 (1), $\mathcal{B} = a$ y el resultado es inmediato.
  - $\mathcal{A} = \mathcal{D} \mathbin{@} \mathcal{A}'$. Por Lem. 3.3.19 (2), $\mathcal{B} = \mathcal{D}' \mathbin{@} \mathcal{B}'$ con $\mathcal{D} \preceq_{\mathfrak{T}} \mathcal{D}'$ y $\mathcal{A}' \preceq_{\mathfrak{T}} \mathcal{B}'$. Luego, se tienen $\langle \mathcal{D}\rfloor_{k-1}, \mathcal{D}'\rfloor_{k-1}\rangle, \langle \mathcal{A}'\rfloor_{k-1}, \mathcal{B}'\rfloor_{k-1}\rangle \in \mathcal{S}$ para todo $k > 0$, y se concluye por definición de $\Phi_{\preceq_{\mathfrak{T}}}$, $\langle \mathcal{A}\rfloor_k, \mathcal{B}\rfloor_k\rangle \in \Phi_{\preceq_{\mathfrak{T}}}(\mathcal{S})$.
  - $\mathcal{A} = \mathcal{A}' \supset \mathcal{A}''$. Por Lem. 3.3.19 (3), $\mathcal{B} = \mathcal{B}' \supset \mathcal{B}''$ con $\mathcal{B}' \preceq_{\mathfrak{T}} \mathcal{A}'$ y $\mathcal{A}'' \preceq_{\mathfrak{T}} \mathcal{B}''$. Como en el caso anterior, se concluye por definición de $\mathcal{S}$ y de la función generadora $\Phi_{\preceq_{\mathfrak{T}}}$, obteniendo $\langle \mathcal{A}\rfloor_k, \mathcal{B}\rfloor_k\rangle = \langle \mathcal{A}'\rfloor_{k-1} \supset \mathcal{A}''\rfloor_{k-1}, \mathcal{B}'\rfloor_{k-1} \supset \mathcal{B}''\rfloor_{k-1}\rangle \in \Phi_{\preceq_{\mathfrak{T}}}(\mathcal{S})$.
- Si $n+m > 2$, por (S-UNION-T) existe función $f : [0, n) \to [0, m)$ t.q. $\mathcal{A}_i \preceq_{\mathfrak{T}} \mathcal{B}_{f(i)}$ para todo $i < n$. Luego, $\langle \mathcal{A}_i\rfloor_k, \mathcal{B}_{f(i)}\rfloor_k\rangle \in \mathcal{S}$ para todo $i < n$ por definición, y se concluye $\langle \mathcal{A}\rfloor_k, \mathcal{B}\rfloor_k\rangle \in \Phi_{\preceq_{\mathfrak{T}}}(\mathcal{S})$.

$\Leftarrow$) Al igual que en el caso anterior, se define $\mathcal{S} \triangleq \{\langle \mathcal{A}, \mathcal{B}\rangle \mid \forall k \in \mathbb{N}.\mathcal{A}\rfloor_k \preceq_{\mathfrak{T}} \mathcal{B}\rfloor_k\}$ y se muestra que es $\Phi_{\preceq_{\mathfrak{T}}}$-denso. Sea $\langle \mathcal{A}, \mathcal{B}\rangle \in \mathcal{S}$. Entonces, para todo $k \in \mathbb{N}$ se tiene $\mathcal{A}\rfloor_k \preceq_{\mathfrak{T}} \mathcal{B}\rfloor_k$. Sin pérdida de generalidad se consideran tipos unión-maximal (Nota 3.3.7)

$$\begin{array}{rcl} \mathcal{A} = \oplus_{i<n}\mathcal{A}_i & \text{con} & \mathcal{A}_i \neq \oplus, i < n \\ \mathcal{B} = \oplus_{j<m}\mathcal{B}_j & \text{con} & \mathcal{B}_j \neq \oplus, j < m \end{array}$$

- Si $n = m = 1$ (*i.e.* $\mathcal{A}, \mathcal{B} \neq \oplus$), se analiza la forma de $\mathcal{A}$:
  - $\mathcal{A} = a$. Entonces, $\mathcal{A}\rfloor_k = a$ para todo $k > 0$ y, por Lem. 3.3.19 (1), $\mathcal{B}\rfloor_k = a$. Por lo tanto, $\mathcal{B} = a$ y se concluye por definición de $\Phi_{\preceq_{\mathfrak{T}}}$, $\langle a, a\rangle \in \Phi_{\preceq_{\mathfrak{T}}}(\mathcal{S})$.
  - $\mathcal{A} = \mathcal{D} \mathbin{@} \mathcal{A}'$. Del mismo modo, se tiene $\mathcal{A}\rfloor_k = \mathcal{D}\rfloor_{k-1} \mathbin{@} \mathcal{A}'\rfloor_{k-1}$ para todo $k > 0$. Por Lem. 3.3.19 (2), $\mathcal{B}\rfloor_k = \mathcal{D}'_k \mathbin{@} \mathcal{B}'_k$ con $\mathcal{D}\rfloor_{k-1} \preceq_{\mathfrak{T}} \mathcal{D}'_k$ y $\mathcal{A}'\rfloor_{k-1} \preceq \mathcal{B}'_k$. Notar que para cada k se tienen sub-árboles $\mathcal{D}'_k$ y $\mathcal{B}'_k$ diferentes pero, dado que el Lem. 3.3.19 apela a la igualdad de árboles al determinar la forma de $\mathcal{B}$, es inmediato ver de la Def. 3.3.10 que $\mathcal{B} = \mathcal{D}' \mathbin{@} \mathcal{B}'$ con $\mathcal{D}'_k = \mathcal{D}'\rfloor_{k-1}$ y $\mathcal{B}'_k = \mathcal{B}'\rfloor_{k-1}$ para todo $k > 0$. Por lo tanto, $\mathcal{D}\rfloor_{k-1} \preceq_{\mathfrak{T}} \mathcal{D}'\rfloor_{k-1}$ y $\mathcal{A}'\rfloor_{k-1} \preceq_{\mathfrak{T}} \mathcal{B}'\rfloor_{k-1}$ para todo $k > 0$. Luego, por definición se tienen $\langle \mathcal{D}, \mathcal{D}'\rangle, \langle \mathcal{A}', \mathcal{B}'\rangle \in \mathcal{S}$ y se concluye $\langle \mathcal{D} \mathbin{@} \mathcal{A}', \mathcal{D}' \mathbin{@} \mathcal{B}'\rangle \in \Phi_{\preceq_{\mathfrak{T}}}(\mathcal{S})$.
  - $\mathcal{A} = \mathcal{A}' \supset \mathcal{A}''$. El análisis para este caso es similar al anterior. De $\mathcal{A}\rfloor_k = \mathcal{A}'\rfloor_{k-1} \supset \mathcal{A}''\rfloor_{k-1}$ se tiene $\mathcal{B} = \mathcal{B}' \supset \mathcal{B}''$ con $\mathcal{B}'\rfloor_{k-1} \preceq_{\mathfrak{T}} \mathcal{A}'\rfloor_{k-1}$ y $\mathcal{A}''\rfloor_{k-1} \preceq_{\mathfrak{T}} \mathcal{B}''\rfloor_{k-1}$ para todo $k > 0$, por Lem. 3.3.19 (3) y Def. 3.3.10. Luego, $\langle \mathcal{B}', \mathcal{A}'\rangle, \langle \mathcal{A}'', \mathcal{B}''\rangle \in \mathcal{S}$ y $\langle \mathcal{A}' \supset \mathcal{B}', \mathcal{A}'' \supset \mathcal{B}''\rangle \in \Phi_{\preceq_{\mathfrak{T}}}(\mathcal{S})$.
- Si $n+m > 2$, se tiene $\mathcal{A}\rfloor_k = \oplus_{i<n}(\mathcal{A}_i\rfloor_k)$ y $\mathcal{B}\rfloor_k = \oplus_{j<m}(\mathcal{B}_j\rfloor_k)$ para todo $k > 0$. De $\mathcal{A}\rfloor_k \preceq_{\mathfrak{T}} \mathcal{B}\rfloor_k$, por (S-UNION-T), se tiene

  $$\exists f : [0, n) \to [0, m) \text{ t.q. } \mathcal{A}_i\rfloor_k \preceq_{\mathfrak{T}} \mathcal{B}_{f(i)}\rfloor_k \text{ para todo } i < n$$

  Más aún, como $\mathcal{C}\rfloor_0 = \varepsilon$ pata todo $\mathcal{C} \in \mathfrak{T}$, set tiene $\mathcal{A}_i\rfloor_0 \preceq_{\mathfrak{T}} \mathcal{B}_{f(i)}\rfloor_0$ por reflexividad. Luego, $\mathcal{A}_i\rfloor_k \preceq_{\mathfrak{T}} \mathcal{B}_{f(i)}\rfloor_k$ $k \in \mathbb{N}$. Entonces, $\langle \mathcal{A}_i, \mathcal{B}_{f(i)}\rangle \in \mathcal{S}$ para todo $i < n$ y $\langle \mathcal{A}, \mathcal{B}\rangle \in \Phi_{\preceq_{\mathfrak{T}}}(\mathcal{S})$ por definición.

□

Correspondencia con μ-tipos contractivos

Lema 3.3.23. $[\![\{V \setminus B\}A]\!]^{\mathfrak{T}} = \{V \setminus [\![B]\!]^{\mathfrak{T}}\}[\![A]\!]^{\mathfrak{T}}$.

Demostración. Con un argumento similar al empleado en la demostración del Lem. 3.3.16 puede verse que, dados $\mathcal{A}, \mathcal{B} \in \mathfrak{T}$, $\mathcal{A} = \mathcal{B}$ sii $\forall k \in \mathbb{N}.\mathcal{A}\rfloor_k = \mathcal{B}\rfloor_k$. Luego, basta probar la siguiente propiedad para concluir: $\forall k \in \mathbb{N}.[\![\{V \setminus B\}A]\!]^{\mathfrak{T}}\rfloor_k = \{V \setminus [\![B]\!]^{\mathfrak{T}}\}[\![A]\!]^{\mathfrak{T}}\rfloor_k$.

La prueba es por inducción en el orden lexicográfico $\langle h([\![\{V \setminus B\}A]\!]^{\mathfrak{T}}\rfloor_k), \#_{\mu\oplus}(A)\rangle$, con $h : \mathfrak{T}^{fin} \to \mathbb{N}$ la función de altura para árboles finitos, y $\#_{\mu\oplus}(A)$ el número de ocurrencias de μ y $\oplus$ a la cabeza del μ-tipo contractivo A.

Se analiza la forma de A asumiendo $k > 0$ (el caso $k = 0$ es inmediato).

- $A = V$. Por Lem. 3.3.9 (1), $[\![\{V \setminus B\}V]\!]^{\mathfrak{T}}\rfloor_k = [\![B]\!]^{\mathfrak{T}}\rfloor_k = (\{V \setminus [\![B]\!]^{\mathfrak{T}}\}V)\rfloor_k$.
- $A = a \neq V$. Luego, $[\![\{V \setminus B\}a]\!]^{\mathfrak{T}}\rfloor_k = [\![a]\!]^{\mathfrak{T}}\rfloor_k = a = (\{V \setminus [\![B]\!]^{\mathfrak{T}}\}a)\rfloor_k$, por Def. 3.3.22 y Lem. 3.3.9 (2).
- $A = D \mathbin{@} A'$. Luego,

$$\begin{array}{rcll}
[\![\{V \setminus B\}A]\!]^{\mathfrak{T}}\rfloor_k & = & [\![\{V \setminus B\}D \mathbin{@} \{V \setminus B\}A']\!]^{\mathfrak{T}}\rfloor_k & \\
& = & ([\![\{V \setminus B\}D]\!]^{\mathfrak{T}} \mathbin{@} [\![\{V \setminus B\}A']\!]^{\mathfrak{T}})\rfloor_k & \text{Def. 3.3.22} \\
& = & [\![\{V \setminus B\}D]\!]^{\mathfrak{T}}\rfloor_{k-1} \mathbin{@} [\![\{V \setminus B\}A']\!]^{\mathfrak{T}}\rfloor_{k-1} & \text{Def. 3.3.10} \\
& = & (\{V \setminus [\![B]\!]^{\mathfrak{T}}\}[\![D]\!]^{\mathfrak{T}})\rfloor_{k-1} \mathbin{@} (\{V \setminus [\![B]\!]^{\mathfrak{T}}\}[\![A']\!]^{\mathfrak{T}})\rfloor_{k-1} & \textit{h.i.} \\
& = & (\{V \setminus [\![B]\!]^{\mathfrak{T}}\}[\![D]\!]^{\mathfrak{T}} \mathbin{@} \{V \setminus [\![B]\!]^{\mathfrak{T}}\}[\![A']\!]^{\mathfrak{T}})\rfloor_k & \text{Def. 3.3.10} \\
& = & (\{V \setminus [\![B]\!]^{\mathfrak{T}}\}([\![D]\!]^{\mathfrak{T}} \mathbin{@} [\![A']\!]^{\mathfrak{T}}))\rfloor_k & \text{Lem. 3.3.9 (3)} \\
& = & (\{V \setminus [\![B]\!]^{\mathfrak{T}}\}[\![D \mathbin{@} A']\!]^{\mathfrak{T}})\rfloor_k & \text{Def. 3.3.22}
\end{array}$$

- $A = A' \supset A''$. Similar al caso anterior.
- $A = A_0 \oplus A_1$. El análisis para este caso es similar al anterior, pero cabe destacar que se tiene el mismo k al aplicar la *h.i.* (en lugar de $k - 1$ como antes), por Def. 3.3.10. Sin embargo, la hipótesis es aplicable de todos modos, pues se tiene

$$h(\{V \setminus [\![B]\!]^{\mathfrak{T}}\}[\![A]\!]^{\mathfrak{T}}) \geq h(\{V \setminus [\![B]\!]^{\mathfrak{T}}\}[\![A_i]\!]^{\mathfrak{T}}) \quad \text{pero} \quad \#_{\mu\oplus}(A) > \#_{\mu\oplus}(A_i)$$

  Por lo tanto, es seguro concluir $[\![\{V \setminus B\}A]\!]^{\mathfrak{T}}\rfloor_k = \{V \setminus [\![B]\!]^{\mathfrak{T}}\}[\![A]\!]^{\mathfrak{T}}\rfloor_k$.
- $A = \mu W.A'$. Sin pérdida de generalidad se asume W `evita` $\{V \setminus B\}$. Luego,

$$\begin{array}{rcll}
[\![\{V \setminus B\}A]\!]^{\mathfrak{T}}\rfloor_k & = & [\![\mu W.\{V \setminus B\}A']\!]^{\mathfrak{T}}\rfloor_k & \\
& = & [\![\{W \setminus \mu W.\{V \setminus B\}A'\}\{V \setminus B\}A']\!]^{\mathfrak{T}}\rfloor_k & \text{Def. 3.3.22} \\
& = & [\![\{V \setminus B\}\{W \setminus A\}A']\!]^{\mathfrak{T}}\rfloor_k & \\
& = & (\{V \setminus [\![B]\!]^{\mathfrak{T}}\}[\![\{W \setminus A\}A']\!]^{\mathfrak{T}})\rfloor_k & \textit{h.i.} \\
& = & (\{V \setminus [\![B]\!]^{\mathfrak{T}}\}[\![A]\!]^{\mathfrak{T}})\rfloor_k & \text{Def. 3.3.22}
\end{array}$$

□

Lema 3.3.25. *Sean $A = \mu V.A'$ y B μ-tipos contractivos, y σ sustitución. Luego, $\forall k \in \mathbb{N}.[\![A^k_{B,\sigma}]\!]^{\mathfrak{T}}\rfloor_k \simeq_{\mathfrak{T}} [\![\sigma A]\!]^{\mathfrak{T}}\rfloor_k$.*

Demostración. Por inducción en k. Sin pérdida de generalidad, se asume σ `evita` V.

- $k = 0$. Luego, $[\![A^0_{B,\sigma}]\!]^{\mathfrak{T}}\rfloor_0 = \varepsilon = [\![\sigma A]\!]^{\mathfrak{T}}\rfloor_0$ por Def. 3.3.10.

- $k > 0$. Por *h.i.* se tiene $\lfloor [\![A^{k-1}_{B,\sigma}]\!]^{\mathfrak{T}} \rfloor_{k-1} \simeq_{\mathfrak{T}} \lfloor [\![\sigma A]\!]^{\mathfrak{T}} \rfloor_{k-1}$. Más aún, por ser A contractivo, la primer ocurrencia de V en A' se a profundidad $n > 1$. Luego, $k \leq k - 1 + n$ y se concluye apelando a los Lem. 3.3.15 y 3.3.23

$$\begin{array}{rcll}
\lfloor [\![A^{k}_{B,\sigma}]\!]^{\mathfrak{T}} \rfloor_k & = & \lfloor [\![(\sigma \uplus \{V \setminus A^{k-1}_{B,\sigma}\})A']\!]^{\mathfrak{T}} \rfloor_k & \\
& = & \lfloor (\{V \setminus [\![A^{k-1}_{B,\sigma}]\!]^{\mathfrak{T}}\}[\![\sigma A']\!]^{\mathfrak{T}}) \rfloor_k & \text{Lem. 3.3.23} \\
& = & \lfloor (\{V \setminus \lfloor [\![A^{k-1}_{B,\sigma}]\!]^{\mathfrak{T}} \rfloor_{k-1}\}[\![\sigma A']\!]^{\mathfrak{T}}) \rfloor_k & k \leq k - 1 + n \\
& \simeq_{\mathfrak{T}} & \lfloor (\{V \setminus \lfloor [\![\sigma A]\!]^{\mathfrak{T}} \rfloor_{k-1}\}[\![\sigma A']\!]^{\mathfrak{T}}) \rfloor_k & \textit{h.i.} \text{ y Lem. 3.3.15} \\
& = & \lfloor (\{V \setminus [\![\sigma A]\!]^{\mathfrak{T}}\}[\![\sigma A']\!]^{\mathfrak{T}}) \rfloor_k & k \leq k - 1 + n \\
& = & \lfloor [\![(\sigma \uplus \{V \setminus \sigma A\})A']\!]^{\mathfrak{T}} \rfloor_k & \text{Lem. 3.3.23} \\
& = & \lfloor [\![\sigma A]\!]^{\mathfrak{T}} \rfloor_k &
\end{array}$$

□

Lema 3.3.27. *$A \simeq_\mu B$ sii $\forall k \in \mathbb{N}. \lfloor [\![A]\!]^{\mathfrak{T}} \rfloor_k \simeq_{\mathfrak{T}} \lfloor [\![B]\!]^{\mathfrak{T}} \rfloor_k$.*

Demostración. $\Rightarrow$) Esta parte de la demostración es por inducción en $A \simeq_\mu B$, analizando la última regla aplicada. Notar que el caso $k = 0$ es trivial por Def. 3.3.10. Por lo tanto, se asume $k >$ para el resto de la prueba.

- (E-REFL). Luego, $B = A$ y se concluye por reflexividad de $\simeq_{\mathfrak{T}}$, $\lfloor [\![A]\!]^{\mathfrak{T}} \rfloor_k \simeq_{\mathfrak{T}} \lfloor [\![B]\!]^{\mathfrak{T}} \rfloor_k$ para todo $k \in \mathbb{N}$.

- (E-TRANS). Luego, $A \simeq_\mu C$ y $C \simeq_\mu B$. Por *h.i.* $\lfloor [\![A]\!]^{\mathfrak{T}} \rfloor_k \simeq_{\mathfrak{T}} \lfloor [\![C]\!]^{\mathfrak{T}} \rfloor_k$ y $\lfloor [\![C]\!]^{\mathfrak{T}} \rfloor_k \simeq_{\mathfrak{T}} \lfloor [\![B]\!]^{\mathfrak{T}} \rfloor_k$ para todo $k \in \mathbb{N}$. Se concluye por transitividad de $\simeq_{\mathfrak{T}}$.

- (E-SYMM). Luego, $B \simeq_\mu A$, Por *h.i.* $\lfloor [\![B]\!]^{\mathfrak{T}} \rfloor_k \simeq_{\mathfrak{T}} \lfloor [\![C]\!]^{\mathfrak{T}} \rfloor_k$ para todo $k \in \mathbb{N}$. Se concluye por simetría de $\simeq_{\mathfrak{T}}$.

- (E-COMP). Luego, $A = D@A'$ y $B = D'@B'$ con $D \simeq_\mu D'$ y $A' \simeq_\mu B'$. Por *h.i.* $\lfloor [\![D]\!]^{\mathfrak{T}} \rfloor_k \simeq_{\mathfrak{T}} \lfloor [\![D']\!]^{\mathfrak{T}} \rfloor_k$ y $\lfloor [\![A']\!]^{\mathfrak{T}} \rfloor_k \simeq_{\mathfrak{T}} \lfloor [\![B']\!]^{\mathfrak{T}} \rfloor_k$ para todo $k \in \mathbb{N}$ (en particular para $k > 0$). Finalmente,

$$\lfloor [\![A]\!]^{\mathfrak{T}} \rfloor_k = \lfloor [\![D]\!]^{\mathfrak{T}} \rfloor_{k-1} @ \lfloor [\![A']\!]^{\mathfrak{T}} \rfloor_{k-1} \simeq_{\mathfrak{T}} \lfloor [\![D']\!]^{\mathfrak{T}} \rfloor_{k-1} @ \lfloor [\![B']\!]^{\mathfrak{T}} \rfloor_{k-1} = \lfloor [\![B]\!]^{\mathfrak{T}} \rfloor_k$$

- (E-FUNC). Luego, $A = A' \supset A''$ y $B = B' \supset B''$ con $A' \simeq_\mu B'$ y $A'' \simeq_\mu B''$. Como en el caso anterior, se concluye directo por la *h.i.*

- (E-UNION-IDEM). Luego, $A = B \oplus B$. Notar que B puede ser, a su vez, un tipo unión. Sean $\lfloor [\![A]\!]^{\mathfrak{T}} \rfloor_k = \oplus_{i<n} \mathcal{A}_i$ y $\lfloor [\![B]\!]^{\mathfrak{T}} \rfloor_k = \oplus_{j<m} \mathcal{B}_i$ con $(\mathcal{A}_i \neq \oplus)_{i<n}$ y $(\mathcal{B}_j \neq \oplus)_{j<m}$ (*cf.* Nota 3.3.7). Es inmediato ver que $A = B \oplus B$ implica $n = 2 * m$ y $\mathcal{A}_j = \mathcal{A}_{2*j} = \mathcal{B}_j$ para todo $j < m$. Luego, se concluye por (E-UNION-T),

$$\begin{array}{rcl}
\lfloor [\![A]\!]^{\mathfrak{T}} \rfloor_k & = & \oplus_{i<n} \mathcal{A}_i \\
& = & (\oplus_{j<m} \mathcal{B}_i) \oplus (\oplus_{j<m} \mathcal{B}_i) \\
& \simeq_{\mathfrak{T}} & \oplus_{j<m} \mathcal{B}_i \\
& = & \lfloor [\![B]\!]^{\mathfrak{T}} \rfloor_k
\end{array}$$

- (E-UNION-COMM). Luego, $A = C_0 \oplus C_1$ y $C_1 \oplus C_0$. Como en el caso anterior, considerar $\lfloor [\![A]\!]^{\mathfrak{T}} \rfloor_k = \oplus_{i<n} \mathcal{A}_i$ y $\lfloor [\![B]\!]^{\mathfrak{T}} \rfloor_k = \oplus_{j<m} \mathcal{B}_i$ con $\mathcal{A}_i, \mathcal{B}_j \neq \oplus$ para todo $i < n$, $j < m$. Se tiene $n = m > 1$ y, por lo tanto $n + m > 2$. Más aún, sea $\mathcal{A}_l$ el primer componente de $\lfloor [\![C_1]\!]^{\mathfrak{T}} \rfloor_k$: se tiene $\mathcal{A}_i = \mathcal{B}_{i+n-l}$ para todo $i < l$, y $\mathcal{A}_i = \mathcal{B}_{i-l}$ para todo $i \geq l$. Luego, se concluye por (E-UNION-T).

- (E-UNION-ASSOC). Luego, $A = C_0 \oplus (C_1 \oplus C_2)$ y $B = (C_0 \oplus C_1) \oplus C_2$. Considerando tipos unión-maximal se tiene $[\![A]\!]^{\mathfrak{T}}\rfloor_k = \oplus_{i<n}\mathcal{A}_i$ y $[\![B]\!]^{\mathfrak{T}}\rfloor_k = \oplus_{j<m}\mathcal{B}_i$ con $\mathcal{A}_i, \mathcal{B}_j \neq \oplus$ para todo $i < n$, $j < m$. Más aún, $n = m > 2$ y $\mathcal{A}_i = \mathcal{B}_i$ para todo $i < n$. Luego, se concluye por (E-UNION-T) utilizando la función identidad id.

- (E-UNION). Luego, $A = A' \oplus A''$ y $B = B' \oplus B''$ con $A' \simeq_\mu B'$ y $A'' \simeq_\mu B''$. Por *h.i.* $[\![A']\!]^{\mathfrak{T}}\rfloor_k \simeq_{\mathfrak{T}} [\![B']\!]^{\mathfrak{T}}\rfloor_k$ y $[\![A'']\!]^{\mathfrak{T}}\rfloor_k \simeq_{\mathfrak{T}} [\![B'']\!]^{\mathfrak{T}}\rfloor_k$ para todo $k \in \mathbb{N}$. Considerar

$$\begin{array}{lll} [\![A']\!]^{\mathfrak{T}}\rfloor_k = \oplus_{i<n}\mathcal{A}'_i & \text{con} & \mathcal{A}'_i \neq \oplus, i < n \\ [\![B']\!]^{\mathfrak{T}}\rfloor_k = \oplus_{j<m}\mathcal{B}'_j & \text{con} & \mathcal{B}'_j \neq \oplus, j < m \end{array}$$

  Si $n + m > 2$, por (E-UNION-T), existen funciones $f : [0, n) \to [0, m)$ y $g : [0, m) \to [0, n)$ tal que $(\mathcal{A}'_i \simeq_{\mathfrak{T}} \mathcal{B}'_{f(i)})_{i<n}$ y $(\mathcal{A}'_{g(j)} \simeq_{\mathfrak{T}} \mathcal{B}'_j)_{j<m}$. Si no (*i.e.* $n = m = 1$), simplemente tomar $f = g = id$.

  Del mismo modo, con A'' y B'' se tiene

$$\begin{array}{lll} [\![A'']\!]^{\mathfrak{T}}\rfloor_k = \oplus_{i<n'}\mathcal{A}''_i & \text{con} & \mathcal{A}''_i \neq \oplus, i < n' \\ [\![B'']\!]^{\mathfrak{T}}\rfloor_k = \oplus_{j<m'}\mathcal{B}''_j & \text{con} & \mathcal{B}''_j \neq \oplus, j < m' \end{array}$$

  con $f' : [0, n') \to [0, m')$ y $g' : [0, m') \to [0, n')$ tal que $(\mathcal{A}''_i \simeq_{\mathfrak{T}} \mathcal{B}''_{f'(i)})_{i<n'}$ y $(\mathcal{A}''_{g'(j)} \simeq_{\mathfrak{T}} \mathcal{B}''_j)_{j<m'}$.

  Finalmente, como $n + n' + m + m' > 2$, se concluye por (E-UNION-T),

$$\begin{array}{rcl} [\![A]\!]^{\mathfrak{T}}\rfloor_k & = & [\![A']\!]^{\mathfrak{T}}\rfloor_k \oplus [\![A'']\!]^{\mathfrak{T}}\rfloor_k \\ & = & (\oplus_{i<n}\mathcal{A}'_i) \oplus (\oplus_{i<n'}\mathcal{A}''_i) \\ & \simeq_{\mathfrak{T}} & (\oplus_{j<m}\mathcal{B}'_j) \oplus (\oplus_{j<m'}\mathcal{B}''_j) \\ & = & [\![B']\!]^{\mathfrak{T}}\rfloor_k \oplus [\![B']\!]^{\mathfrak{T}}\rfloor_k \\ & = & [\![B]\!]^{\mathfrak{T}}\rfloor_k \end{array}$$

- (E-REC). Luego, $A = \mu V.A'$ y $\mu V.B'$ con $A' \simeq_{\mathfrak{T}} B'$. Por *h.i.* $[\![A']\!]^{\mathfrak{T}}\rfloor_k \simeq_{\mathfrak{T}} [\![B']\!]^{\mathfrak{T}}\rfloor_k$ para todo $k \in \mathbb{N}$ y, por Lem. 3.3.16, $[\![A']\!]^{\mathfrak{T}} \simeq_{\mathfrak{T}} [\![B']\!]^{\mathfrak{T}}$. Apelando a los desdoblamientos finitos (Def. 3.3.24) de A y B respectivamente, se muestra a continuación que $[\![A^n_{A',id}]\!]^{\mathfrak{T}} \simeq_{\mathfrak{T}} [\![B^n_{B',id}]\!]^{\mathfrak{T}}$ para todo $n \in \mathbb{N}$. La prueba es por inducción en n.

  - $n = 0$. Luego, se tiene $[\![A^0_{A',id}]\!]^{\mathfrak{T}} = [\![A']\!]^{\mathfrak{T}} \simeq_{\mathfrak{T}} [\![B']\!]^{\mathfrak{T}} = [\![B^0_{B',id}]\!]^{\mathfrak{T}}$.
  - $n > 0$. Se sabe $[\![A']\!]^{\mathfrak{T}} \simeq_{\mathfrak{T}} [\![B']\!]^{\mathfrak{T}}$ y por *h.i.* $[\![A^{n-1}_{A',id}]\!]^{\mathfrak{T}} \simeq_{\mathfrak{T}} [\![B^{n-1}_{B',id}]\!]^{\mathfrak{T}}$. Luego, se concluye por Lem. 3.3.15 y 3.3.23

$$[\![A^n_{A',id}]\!]^{\mathfrak{T}} = \{V \setminus [\![A^{n-1}_{A',id}]\!]^{\mathfrak{T}}\}[\![A']\!]^{\mathfrak{T}} \simeq_{\mathfrak{T}} \{V \setminus [\![B^{n-1}_{B',id}]\!]^{\mathfrak{T}}\}[\![B']\!]^{\mathfrak{T}} = [\![B^n_{B',id}]\!]^{\mathfrak{T}}$$

  Finalmente, por Lem. 3.3.16, $[\![A^n_{A',id}]\!]^{\mathfrak{T}}\rfloor_k \simeq_{\mathfrak{T}} [\![B^n_{B',id}]\!]^{\mathfrak{T}}\rfloor_k$ para todo $k, n \in \mathbb{N}$. Luego, se concluye por Lem. 3.3.25

$$[\![A]\!]^{\mathfrak{T}}\rfloor_k \simeq_{\mathfrak{T}} [\![A^n_{A',id}]\!]^{\mathfrak{T}}\rfloor_k \simeq_{\mathfrak{T}} [\![B^n_{B',id}]\!]^{\mathfrak{T}}\rfloor_k \simeq_{\mathfrak{T}} [\![B]\!]^{\mathfrak{T}}\rfloor_k$$

- (E-FOLD). Luego, $A = \mu V.A'$ y $B = \{V \setminus A\}A'$. El resultado es inmediato de la Def. 3.3.22, $[\![A]\!]^{\mathfrak{T}} = [\![\mu V.A']\!]^{\mathfrak{T}} = [\![\{V \setminus A\}A']\!]^{\mathfrak{T}} = [\![B]\!]^{\mathfrak{T}}$ por reflexividad de $\simeq_{\mathfrak{T}}$.

- (E-CONTR). Luego, $B = \mu V.B'$ es contractivo y $A \simeq_\mu \{V \setminus A\}B'$. Por *h.i.* y Lem. 3.3.16, $[\![A]\!]^{\mathfrak{T}} \simeq_{\mathfrak{T}} [\![\{V \setminus A\}B']\!]^{\mathfrak{T}}$. Como en el caso anterior, se apela al desdoblamiento finito de B (Def. 3.3.24) y se muestra $[\![A]\!]^{\mathfrak{T}} \simeq_{\mathfrak{T}} [\![B^n_{A,id}]\!]^{\mathfrak{T}}$ para todo $n \in \mathbb{N}$.

  - $n = 0$. Inmediato de la Def. 3.3.24 por reflexividad, pues $B^0_{A,id} = A$.

- $n > 0$. Por Def. 3.3.22 y Lem. 3.3.23, $[\![B^n_{A,id}]\!]^{\mathfrak{T}} = \{V \setminus [\![B^{n-1}_{A,id}]\!]^{\mathfrak{T}}\}[\![B']\!]^{\mathfrak{T}}$. Por *h.i.* se tiene $[\![A]\!]^{\mathfrak{T}} \simeq_{\mathfrak{T}} [\![B^{n-1}_{A,id}]\!]^{\mathfrak{T}}$ y, por Lem. 3.3.15, $[\![B^n_{A,id}]\!]^{\mathfrak{T}} = \{V \setminus [\![A]\!]^{\mathfrak{T}}\}[\![B']\!]^{\mathfrak{T}}$. Finalmente, se concluye por Lem. 3.3.23 y transitividad de $\simeq_{\mathfrak{T}}$ con la hipótesis $[\![A]\!]^{\mathfrak{T}} \simeq_{\mathfrak{T}} [\![\{V \setminus A\}B']\!]^{\mathfrak{T}}$

$$[\![B^n_{A,id}]\!]^{\mathfrak{T}} \simeq_{\mathfrak{T}} [\![\{V \setminus A\}B']\!]^{\mathfrak{T}} \simeq_{\mathfrak{T}} [\![A]\!]^{\mathfrak{T}}$$

Luego, por Lem. 3.3.16, $[\![A]\!]^{\mathfrak{T}}\rfloor_k \simeq_{\mathfrak{T}} [\![B^n_{A,id}]\!]^{\mathfrak{T}}\rfloor_k$ para todo $k, n \in \mathbb{N}$. Además, por Lem. 3.3.25 se tiene $[\![B^k_{A,id}]\!]^{\mathfrak{T}}\rfloor_k \simeq_{\mathfrak{T}} [\![B]\!]^{\mathfrak{T}}\rfloor_k$ para todo $k \in \mathbb{N}$. Luego, se concluye

$$[\![A]\!]^{\mathfrak{T}}\rfloor_k \simeq_{\mathfrak{T}} [\![B^k_{A,id}]\!]^{\mathfrak{T}}\rfloor_k \simeq_{\mathfrak{T}} [\![B]\!]^{\mathfrak{T}}\rfloor_k$$

$\Leftarrow$) Dado un tipo $A = \oplus_{i<n} A_i$ con $i \geq 0$, es inmediato ver que si $A_k = \mu V.A'_k$ para algún $k < n$, entonces $B = \oplus_{i<n} B_i$ con $B_i = A_i$ para todo $i < n$ tal $i \neq k$ y $B_k = \{V \setminus \mu V.A'_k\}A'_k$ satisface $B \simeq_\mu A$ por (E-FOLD), donde $B_k \neq \mu$ por contractividad (Nota 3.3.4). Más aún, por Def. 3.3.22, $[\![A]\!]^{\mathfrak{T}} = [\![B]\!]^{\mathfrak{T}}$. Luego, basta considerar tipos unión-maximal $A = \oplus_{i<n} A_i$ con $(A_i \neq \mu)_{i<n}$, y luego concluir para todo μ-tipo contractivo apelando al argumento anterior.

Sea $[\![A]\!]^{\mathfrak{T}}\rfloor_k \simeq_{\mathfrak{T}} [\![B]\!]^{\mathfrak{T}}\rfloor_k$ para todo $k \in \mathbb{N}$. Se asume, $A = \oplus_{i<n} A_i$ y $B = \oplus_{j<m} B_j$ con $(A_i \neq \mu, \oplus)_{i<n}$ y $(B_j \neq \mu, \oplus)_{j<m}$ por el argumento anterior. Luego, por Def. 3.3.10 y 3.3.22, se tiene

$$\begin{array}{rcl} [\![A]\!]^{\mathfrak{T}}\rfloor_k = \oplus_{i<n}[\![A_i]\!]^{\mathfrak{T}} & \text{con} & \mathcal{A}_i \neq \oplus, i < n \\ [\![B]\!]^{\mathfrak{T}}\rfloor_k = \oplus_{j<m}[\![B_j]\!]^{\mathfrak{T}} & \text{con} & \mathcal{B}_j \neq \oplus, j < m \end{array}$$

La prueba es por inducción en $h([\![A]\!]^{\mathfrak{T}}\rfloor_k) + h([\![B]\!]^{\mathfrak{T}}\rfloor_k)$, donde $h : \mathfrak{T}^{fin} \to \mathbb{N}$ es la función de altura para árboles finitos.

Se analizan entonces las posibilidades para $n + m$.

- Si $n = m = 0$, se analiza la forma de A:
  - $A = a$. Luego, $[\![A]\!]^{\mathfrak{T}}\rfloor_k = a$ para todo $k > 0$. Por Lem. 3.3.14 (1), $[\![B]\!]^{\mathfrak{T}}\rfloor_k = a$ para todo $k > 0$. Entonces, por Def. 3.3.22 y 3.3.10 se tiene $B = a$ y se concluye por (E-REFL).
  - $A = D \mathbin{@} A'$. Luego, $[\![A]\!]^{\mathfrak{T}}\rfloor_k = [\![D]\!]^{\mathfrak{T}}\rfloor_{k-1} \mathbin{@} [\![A']\!]^{\mathfrak{T}}\rfloor_{k-1}$ para todo $k > 0$. Por Lem. 3.3.14 (2), $[\![B]\!]^{\mathfrak{T}}\rfloor_k = \mathcal{D}'_k \mathbin{@} \mathcal{B}'_k$ con $[\![D]\!]^{\mathfrak{T}}\rfloor_{k-1} \simeq_{\mathfrak{T}} \mathcal{D}'_k$ y $[\![A']\!]^{\mathfrak{T}}\rfloor_{k-1} \simeq_{\mathfrak{T}} \mathcal{B}'_k$ para todo $k > 0$. Notar que para cada k se tienen sub-árboles $\mathcal{D}'_k$ y $\mathcal{B}'_k$ diferentes pero, dado que el Lem. 3.3.14 apela a la igualdad de árboles (no equivalencia) al determinar la forma de $[\![B]\!]^{\mathfrak{T}}\rfloor_k$, es inmediato ver de la Def. 3.3.10 que $[\![B]\!]^{\mathfrak{T}} = \mathcal{D}' \mathbin{@} \mathcal{B}'$ con $\mathcal{D}'_k = \mathcal{D}'\rfloor_{k-1}$ y $\mathcal{B}'_k = \mathcal{B}'\rfloor_{k-1}$ para todo $k > 0$. Más aún, $m = 1$ implica $B \neq \mu$ por hipótesis. Luego, por Def. 3.3.22, $B = D' \mathbin{@} B'$ con $[\![D']\!]^{\mathfrak{T}} = \mathcal{D}'$ y $[\![B']\!]^{\mathfrak{T}} = \mathcal{B}'$. Es decir, $[\![B]\!]^{\mathfrak{T}}\rfloor_k = [\![D']\!]^{\mathfrak{T}}\rfloor_{k-1} \mathbin{@} [\![B']\!]^{\mathfrak{T}}\rfloor_{k-1}$ con $[\![D]\!]^{\mathfrak{T}}\rfloor_{k-1} \simeq_{\mathfrak{T}} [\![D']\!]^{\mathfrak{T}}\rfloor_{k-1}$ y $[\![A']\!]^{\mathfrak{T}}\rfloor_{k-1} \simeq_{\mathfrak{T}} [\![B']\!]^{\mathfrak{T}}\rfloor_{k-1}$ para todo $k > 0$. Finalmente, de la *h.i.* se obtienen $D \simeq_\mu D'$ y $A' \simeq_\mu B'$, y se concluye por (E-COMP).
  - $A = A' \supset A''$. El análisis para este caso es similar al anterior. De $[\![A]\!]^{\mathfrak{T}}\rfloor_k = [\![A']\!]^{\mathfrak{T}}\rfloor_{k-1} \supset [\![A'']\!]^{\mathfrak{T}}\rfloor_{k-1}$ se obtienen $B = B' \supset B''$ y $[\![B]\!]^{\mathfrak{T}}\rfloor_k = [\![B']\!]^{\mathfrak{T}}\rfloor_{k-1} \supset [\![B'']\!]^{\mathfrak{T}}\rfloor_{k-1}$ para todo $k > 0$, por Lem. 3.3.14 (3), Def. 3.3.10 y 3.3.22 con la hipótesis $B \neq \mu$. Luego, por *h.i.* se tienen $A' \simeq_\mu B'$ y $A'' \simeq_\mu B''$, y se concluye por (E-FUNC).
  - $A = \mu V.A'$. Este caso no aplica por hipótesis.
- Si $n + m > 2$, la última regla aplicada para derivar $[\![A]\!]^{\mathfrak{T}}\rfloor_k \simeq_{\mathfrak{T}} [\![B]\!]^{\mathfrak{T}}\rfloor_k$ es necesariamente (E-UNION-T). Luego, existen funciones $f : [0, n) \to [0, m)$ y $g : [0, m) \to [0, n)$ tal que se tiene $([\![A_i]\!]^{\mathfrak{T}}\rfloor_k \simeq_{\mathfrak{T}} [\![B_{f(i)}]\!]^{\mathfrak{T}}\rfloor_k)_{i<n}$ y $([\![A_{g(j)}]\!]^{\mathfrak{T}}\rfloor_k \simeq_{\mathfrak{T}} [\![B_j]\!]^{\mathfrak{T}}\rfloor_k)_{j<m}$ para todo $k > 0$. Por *h.i.* se tiene $(A_i \simeq_\mu B_{f(i)})_{i<n}$ y $(A_{g(j)} \simeq_\mu B_j)_{j<m}$. Finalmente, se concluye por Lem. 3.3.5, $\oplus_{i<n} A_i \simeq_\mu \oplus_{j<m} B_j$.

□

Lema 3.3.29. *Sean* $\Sigma = \{V_i \preceq_\mu W_i\}_{i<n}$ *un contexto de sub-tipado y* σ *una sustitución tal que* $\mathsf{dom}(\sigma) = \{V_i, W_i\}_{i<n}$, $\sigma(V_i) = A_i$ *y* $\sigma(W_i) = B_i$ *con* $\mathsf{dom}(\sigma) \cap \mathsf{fv}(\{A_i, B_i\}_{i<n}) = \emptyset$, $[\![A_i]\!]^{\mathfrak{T}} \preceq_{\mathfrak{T}} [\![B_i]\!]^{\mathfrak{T}}$ *y* $A_i, B_i \in \mathcal{T}$ *para todo* $i < n$. *Si* $\Sigma \vdash A \simeq_\mu B$, *entonces* $[\![\sigma A]\!]^{\mathfrak{T}} \simeq_{\mathfrak{T}} [\![\sigma B]\!]^{\mathfrak{T}}$.

Demostración. Por inducción en $\vdash \Sigma A \preceq_\mu B$ analizando la última regla aplicada.

- (S-HYP). Luego, $A = V$ y $B = W$ con $\Sigma = \Sigma'; V \preceq_\mu W$. El resultado es inmediato por hipótesis, pues $\sigma A = A_n$, $\sigma B = B_n$ y $[\![A_n]\!]^{\mathfrak{T}} \preceq_{\mathfrak{T}} [\![B_n]\!]^{\mathfrak{T}}$.
- (S-EQ). Luego $\vdash A \simeq_\mu B$. Más aún, se tiene $\vdash \sigma A \simeq_\mu \sigma B$ por ser $\simeq_\mu$ es una congruencia. Por $[\![\sigma A]\!]^{\mathfrak{T}} \simeq_{\mathfrak{T}} [\![\sigma B]\!]^{\mathfrak{T}}$ y se concluye por Lem. 3.3.20.
- (S-REFL). Luego, $A = B$ y el resultado es inmediato por reflexividad de $\preceq_{\mathfrak{T}}$.
- (S-TRANS). Luego, $\vdash \Sigma A \preceq_\mu C$ y $\vdash \Sigma C \preceq_\mu B$ para algún $C \in \mathcal{T}$. Por *h.i.* se tienen $[\![\sigma A]\!]^{\mathfrak{T}} \preceq_{\mathfrak{T}} [\![\sigma C]\!]^{\mathfrak{T}}$ y $[\![\sigma C]\!]^{\mathfrak{T}} \preceq_{\mathfrak{T}} [\![\sigma B]\!]^{\mathfrak{T}}$. Finalmente, se concluye por transitividad de $\preceq_{\mathfrak{T}}$.
- (S-COMP). Luego, $A = D \mathbin{@} A'$ y $B = D' \mathbin{@} B'$ con $\vdash \Sigma D \preceq_\mu D'$ y $\vdash \Sigma A' \preceq_\mu B'$. Por *h.i.* se tienen $[\![\sigma D]\!]^{\mathfrak{T}} \preceq_{\mathfrak{T}} [\![\sigma D']\!]^{\mathfrak{T}}$ y $[\![\sigma A']\!]^{\mathfrak{T}} \preceq_{\mathfrak{T}} [\![\sigma B']\!]^{\mathfrak{T}}$. Luego, por (S-COMP-T),

$$[\![\sigma A]\!]^{\mathfrak{T}} = [\![\sigma D]\!]^{\mathfrak{T}} \mathbin{@} [\![\sigma A']\!]^{\mathfrak{T}} \preceq_\mu [\![\sigma D']\!]^{\mathfrak{T}} \mathbin{@} [\![\sigma B']\!]^{\mathfrak{T}} = [\![\sigma B]\!]^{\mathfrak{T}}$$

- (S-FUNC). Luego, $A = A' \supset A''$ y $B = B' \mathbin{@} B''$ con $\vdash \Sigma B' \preceq_\mu A'$ y $\vdash \Sigma A'' \preceq_\mu B''$. Como en el caso anterior, se concluye por *h.i.* y (S-FUNC-T).
- (S-UNION-L). Luego, $A = A' \oplus A''$ con $\vdash \Sigma A' \preceq_\mu B$ y $\vdash \Sigma A'' \preceq_\mu B$. Por *h.i.* se tienen $[\![\sigma A']\!]^{\mathfrak{T}} \preceq_{\mathfrak{T}} [\![\sigma B]\!]^{\mathfrak{T}}$ y $[\![\sigma A'']\!]^{\mathfrak{T}} \preceq_{\mathfrak{T}} [\![\sigma B]\!]^{\mathfrak{T}}$. Considerar (Nota 3.3.7)

$$\begin{array}{rcl} [\![A']\!]^{\mathfrak{T}} = \oplus_{i<l} \mathcal{A}'_i & \text{con} & \mathcal{A}'_i \neq \oplus, i < l \\ [\![A'']\!]^{\mathfrak{T}} = \oplus_{i<l'} \mathcal{A}'_i & \text{con} & \mathcal{A}'_i \neq \oplus, i < l' \\ [\![B]\!]^{\mathfrak{T}} = \oplus_{j<m} \mathcal{B}_j & \text{con} & \mathcal{B}_j \neq \oplus, j < m \end{array}$$

  Se tiene luego tres posibles casos:

  1. $l = l' = m = 1$. Luego, se concluye directo de la *h.i.* por (S-UNION-T),

$$[\![\sigma A]\!]^{\mathfrak{T}} = [\![\sigma A']\!]^{\mathfrak{T}} \oplus [\![\sigma A'']\!]^{\mathfrak{T}} = \mathcal{A}'_1 \oplus \mathcal{A}''_1 \preceq_\mu \mathcal{B}_1 = [\![\sigma B]\!]^{\mathfrak{T}}$$

  2. $l + m > 2$. Luego, existe una función $f : [0, l) \to [0, m)$ tal que $\mathcal{A}'_i \preceq_{\mathfrak{T}} \mathcal{B}_{f(i)}$ para todo $i < l$. Se tienen dos casos posibles:
     a) $l' = m = 1$. Luego, $\mathcal{A}'_i \preceq_{\mathfrak{T}} \mathcal{B}_1$ para todo $i < l$ (*i.e.* f es una función constante) y $\mathcal{A}''_1 \preceq_{\mathfrak{T}} \mathcal{B}_1$. Se concluye por (S-UNION-T),

$$[\![\sigma A]\!]^{\mathfrak{T}} = [\![\sigma A']\!]^{\mathfrak{T}} \oplus [\![\sigma A'']\!]^{\mathfrak{T}} = (\textstyle\bigoplus_{i<l} \mathcal{A}'_i) \oplus \mathcal{A}''_1 \preceq_\mu \mathcal{B}_1 = [\![\sigma B]\!]^{\mathfrak{T}}$$

     b) $l' + m > 2$. Luego, existe una función $g : [0, l') \to [0, m)$ tal que $\mathcal{A}''_i \preceq_{\mathfrak{T}} \mathcal{B}_{g(i)}$ para todo $i < l'$. Nuevamente se concluye por (S-UNION-T) combinando convenientemente f y g,

$$[\![\sigma A]\!]^{\mathfrak{T}} = (\textstyle\bigoplus_{i<l} \mathcal{A}'_i) \oplus (\bigoplus_{i<l'} \mathcal{A}''_i) \preceq_\mu \bigoplus_{j<m} \mathcal{B}_j = [\![\sigma B]\!]^{\mathfrak{T}}$$

  3. $l' + m > 2$ con $l = m = 1$. Análogo al caso $2a$) anterior.

Se concluye entonces, $[\![\sigma A]\!]^{\mathfrak{T}} = [\![\sigma A']\!]^{\mathfrak{T}} \oplus [\![\sigma A'']\!]^{\mathfrak{T}} \preceq_{\mathfrak{T}} [\![\sigma B]\!]^{\mathfrak{T}}$.

- (S-UNION-R1). Luego, $B = B' \oplus B''$ con $\vdash \Sigma A \preceq_\mu B'$. Por *h.i.* se tiene $[\![\sigma A]\!]^{\mathfrak{T}} \preceq_{\mathfrak{T}} [\![\sigma B']\!]^{\mathfrak{T}}$. Considerar (Nota 3.3.7)

$$\begin{array}{rcl} [\![A]\!]^{\mathfrak{T}} = \oplus_{i<l}\mathcal{A}_i & \text{con} & \mathcal{A}_i \neq \oplus, i < l \\ [\![B']\!]^{\mathfrak{T}} = \oplus_{j<m}\mathcal{B}'_j & \text{con} & \mathcal{B}'_j \neq \oplus, j < m \\ [\![B'']\!]^{\mathfrak{T}} = \oplus_{j<m'}\mathcal{B}''_j & \text{con} & \mathcal{B}''_j \neq \oplus, j < m' \end{array}$$

  Se tiene luego dos posibles casos:

  - $l = m = 1$. Luego, $\mathcal{A}_1 \preceq_{\mathfrak{T}} \mathcal{B}'_1$ y se concluye por (S-UNION-T),

$$[\![\sigma A]\!]^{\mathfrak{T}} = \mathcal{A}_1 \preceq_{\mathfrak{T}} \mathcal{B}'_1 \oplus (\textstyle\bigoplus_{j<m'}\mathcal{B}''_j) = [\![\sigma B']\!]^{\mathfrak{T}} \oplus [\![\sigma B'']\!]^{\mathfrak{T}} = [\![\sigma B]\!]^{\mathfrak{T}}$$

  - $l + m > 2$. Luego, existe una función $f : [0, l) \to [0, m)$ tal que $\mathcal{A}_i \preceq_{\mathfrak{T}} \mathcal{B}'_{f(i)}$ para todo $i < l$. Nuevamente se concluye por (S-UNION-T),

$$[\![\sigma A]\!]^{\mathfrak{T}} = \textstyle\bigoplus_{i<l}\mathcal{A}_i \preceq_\mu (\bigoplus_{j<m}\mathcal{B}'_j) \oplus (\bigoplus_{j<m'}\mathcal{B}''_j) = [\![\sigma B]\!]^{\mathfrak{T}}$$

  Se concluye entonces, $[\![\sigma A]\!]^{\mathfrak{T}} \preceq_{\mathfrak{T}} [\![\sigma B']\!]^{\mathfrak{T}} \oplus [\![\sigma B'']\!]^{\mathfrak{T}} = [\![\sigma B]\!]^{\mathfrak{T}}$.

- (S-UNION-R2). Luego, $B = B' \oplus B''$ con $\vdash \Sigma A \preceq_\mu B''$. Este caso es similar al anterior.

- (S-REC). Luego, $A = \mu V.A'$ y $B = \mu W.B'$ con $\vdash \Sigma; V \preceq_\mu W A' \preceq_\mu B'$, $W \notin \mathsf{fv}(A')$ y $V \notin \mathsf{fv}(B')$. Apelando a los desdoblamientos finitos (Def. 3.3.24) de A y B respectivamente, se muestra a continuación que $[\![A^m_{\varepsilon,\sigma}]\!]^{\mathfrak{T}} \preceq_{\mathfrak{T}} [\![B^m_{\varepsilon,\sigma}]\!]^{\mathfrak{T}}$ para todo $m \in \mathbb{N}$. La prueba es por inducción en m.

  - $m = 0$. Inmediato de la Def. 3.3.24 y 3.3.22 por reflexividad de $\preceq_{\mathfrak{T}}$, pues $A^0_{\varepsilon,\sigma} = \varepsilon = B^0_{\varepsilon,\sigma}$.
  - $m > 0$. Luego, se tienen $A^m_{\varepsilon,\sigma} = (\sigma \uplus \{V \setminus A^{m-1}_{\varepsilon,\sigma}\})A'$ y $B^m_{\varepsilon,\sigma} = (\sigma \uplus \{W \setminus B^{m-1}_{\varepsilon,\sigma}\})B'$ por Def. 3.3.24. Más aún, por *h.i.* se tiene $[\![A^{m-1}_{\varepsilon,\sigma}]\!]^{\mathfrak{T}} \preceq_{\mathfrak{T}} [\![B^{m-1}_{\varepsilon,\sigma}]\!]^{\mathfrak{T}}$.
    Considerar $\rho = \sigma \uplus \{V \setminus A^{m-1}_{\varepsilon,\sigma}\} \uplus \{W \setminus B^{m-1}_{\varepsilon,\sigma}\}$. Es inmediato ver que $\rho A' = A^m_{\varepsilon,\sigma}$ y $\rho B' = B^m_{\varepsilon,\sigma}$, pues $W \notin \mathsf{fv}(A')$ y $V \notin \mathsf{fv}(B')$. Más aún, puede verse que ρ satisface las hipótesis del lema para el contexto de sub-tipado $\Sigma; V \preceq_{\mathfrak{T}} W$. En particular, las variables en $\mathsf{dom}(\sigma)$ fueron sustituidas recursivamente en $A^{m-1}_{\varepsilon,\sigma}$ y $B^{m-1}_{\varepsilon,\sigma}$, por lo que se satisface la condición $(\{V_i, W_i\}_{i<n} \cup \{V, W\}) \cap \mathsf{fv}(\{A_i, B_i\}_{i<n} \cup \{A^{m-1}_{\varepsilon,\sigma}, B^{m-1}_{\varepsilon,\sigma}\}) = \emptyset$. Luego, por *h.i.* del lema, $[\![A^m_{\varepsilon,\sigma}]\!]^{\mathfrak{T}} = [\![\rho A']\!]^{\mathfrak{T}} \preceq_{\mathfrak{T}} [\![\rho B']\!]^{\mathfrak{T}} = [\![B^m_{\varepsilon,\sigma}]\!]^{\mathfrak{T}}$.

  Luego, $[\![A^m_{\varepsilon,\sigma}]\!]^{\mathfrak{T}} \preceq_{\mathfrak{T}} [\![B^m_{\varepsilon,\sigma}]\!]^{\mathfrak{T}}$ para todo $m \in \mathbb{N}$ y, por Lem. 3.3.16, se tiene $[\![A^m_{\varepsilon,\sigma}]\!]^{\mathfrak{T}}\rfloor_k \preceq_{\mathfrak{T}} [\![B^m_{\varepsilon,\sigma}]\!]^{\mathfrak{T}}\rfloor_k$ para todo $k, m \in \mathbb{N}$. Más aún, por Lem. 3.3.25, $[\![A^k_{\varepsilon,\sigma}]\!]^{\mathfrak{T}}\rfloor_k \simeq_{\mathfrak{T}} [\![\sigma A]\!]^{\mathfrak{T}}\rfloor_k$ y $[\![B^k_{\varepsilon,\sigma}]\!]^{\mathfrak{T}}\rfloor_k \simeq_{\mathfrak{T}} [\![\sigma B]\!]^{\mathfrak{T}}\rfloor_k$ para todo $k \in \mathbb{N}$. Luego, por simetría de $\simeq_{\mathfrak{T}}$ y transitividad de $\preceq_{\mathfrak{T}}$ (apelando al Lem. 3.3.20) se tiene $[\![\sigma A]\!]^{\mathfrak{T}}\rfloor_k \preceq_{\mathfrak{T}} [\![\sigma B]\!]^{\mathfrak{T}}\rfloor_k$ para todo $k \in \mathbb{N}$. Se concluye por Lem. 3.3.21.

□

Teorema 3.3.30. *$A \preceq_\mu B$ sii $[\![A]\!]^{\mathfrak{T}} \preceq_{\mathfrak{T}} [\![B]\!]^{\mathfrak{T}}$.*

Demostración. $\Rightarrow$) Esta implicación se sigue inmediatamente del Lem. 3.3.29, teniendo Σ vacío y, por lo tanto, σ resulta en la sustitución identidad. Luego, de $A \preceq_\mu B$ se obtiene $[\![A]\!]^{\mathfrak{T}} \preceq_{\mathfrak{T}} [\![B]\!]^{\mathfrak{T}}$.

$\Leftarrow$) Al igual que para la prueba del Lem. 3.3.27, basta considerar tipos unión-maximal de la forma $A = \oplus_{i<n}A_i$ con $A_i \neq \mu$ para todo $i < n$ (Nota 3.3.4), y luego concluir para todo μ-tipo contractivo apelando a (E-FOLD) y (S-EQ).

Para esta parte de la demostración, se prueba el resultado equivalente: $\forall k \in \mathbb{N}. \llbracket A \rrbracket^{\mathfrak{T}} \rfloor_k \preceq_{\mathfrak{T}} \llbracket B \rrbracket^{\mathfrak{T}} \rfloor_k$ implica $A \preceq_\mu B$. Luego, se concluye por Lem. 3.3.21. Sea $\llbracket A \rrbracket^{\mathfrak{T}} \rfloor_k \preceq_{\mathfrak{T}} \llbracket B \rrbracket^{\mathfrak{T}} \rfloor_k$ para todo $k \in \mathbb{N}$. Se asume, $A = \oplus_{i<n} A_i$ y $B = \oplus_{j<m} B_j$ con $(A_i \neq \mu, \oplus)_{i<n}$ y $(B_j \neq \mu, \oplus)_{j<m}$ por el argumento anterior. Luego, por Def. 3.3.10 y 3.3.22, se tiene

$$\begin{array}{rcl} \llbracket A \rrbracket^{\mathfrak{T}} \rfloor_k = \oplus_{i<n} \llbracket A_i \rrbracket^{\mathfrak{T}} & \text{con} & \mathcal{A}_i \neq \oplus, i < n \\ \llbracket B \rrbracket^{\mathfrak{T}} \rfloor_k = \oplus_{j<m} \llbracket B_j \rrbracket^{\mathfrak{T}} & \text{con} & \mathcal{B}_j \neq \oplus, j < m \end{array}$$

La prueba es por inducción en $h(\llbracket A \rrbracket^{\mathfrak{T}} \rfloor_k) + h(\llbracket B \rrbracket^{\mathfrak{T}} \rfloor_k)$, donde $h : \mathfrak{T}^{fin} \to \mathbb{N}$ es la función de altura para árboles finitos.

Se analizan entonces las posibilidades para $n + m$.

- Si $n = m = 0$, se analiza la forma de A:
  - $A = a$. Luego, $\llbracket A \rrbracket^{\mathfrak{T}} \rfloor_k = a$ para todo $k > 0$. Por Lem. 3.3.19 (1), $\llbracket B \rrbracket^{\mathfrak{T}} \rfloor_k = a$ para todo $k > 0$. Entonces, por Def. 3.3.22 y 3.3.10 se tiene $B = a$ y se concluye por (S-REFL).
  - $A = D \mathbin{@} A'$. Luego, $\llbracket A \rrbracket^{\mathfrak{T}} \rfloor_k = \llbracket D \rrbracket^{\mathfrak{T}} \rfloor_{k-1} \mathbin{@} \llbracket A' \rrbracket^{\mathfrak{T}} \rfloor_{k-1}$ para todo $k > 0$. Por Lem. 3.3.19 (2), $\llbracket B \rrbracket^{\mathfrak{T}} \rfloor_k = \mathcal{D}'_k \mathbin{@} \mathcal{B}'_k$ con $\llbracket D \rrbracket^{\mathfrak{T}} \rfloor_{k-1} \preceq_{\mathfrak{T}} \mathcal{D}'_k$ y $\llbracket A' \rrbracket^{\mathfrak{T}} \rfloor_{k-1} \preceq_{\mathfrak{T}} \mathcal{B}'_k$ para todo $k > 0$. Notar que para cada k se tienen sub-árboles $\mathcal{D}'_k$ y $\mathcal{B}'_k$ diferentes pero, dado que el Lem. 3.3.19 apela a la igualdad de árboles al determinar la forma de $\llbracket B \rrbracket^{\mathfrak{T}} \rfloor_k$, es inmediato ver de la Def. 3.3.10 que $\llbracket B \rrbracket^{\mathfrak{T}} = \mathcal{D}' \mathbin{@} \mathcal{B}'$ con $\mathcal{D}'_k = \mathcal{D}' \rfloor_{k-1}$ y $\mathcal{B}'_k = \mathcal{B}' \rfloor_{k-1}$ para todo $k > 0$. Más aún, $m = 1$ implica $B \neq \mu$ por hipótesis. Luego, por Def. 3.3.22, $B = D' \mathbin{@} B'$ con $\llbracket D' \rrbracket^{\mathfrak{T}} = \mathcal{D}'$ y $\llbracket B' \rrbracket^{\mathfrak{T}} = \mathcal{B}'$. Es decir, $\llbracket B \rrbracket^{\mathfrak{T}} \rfloor_k = \llbracket D' \rrbracket^{\mathfrak{T}} \rfloor_{k-1} \mathbin{@} \llbracket B' \rrbracket^{\mathfrak{T}} \rfloor_{k-1}$ con $\llbracket D \rrbracket^{\mathfrak{T}} \rfloor_{k-1} \preceq_{\mathfrak{T}} \llbracket D' \rrbracket^{\mathfrak{T}} \rfloor_{k-1}$ y $\llbracket A' \rrbracket^{\mathfrak{T}} \rfloor_{k-1} \preceq_{\mathfrak{T}} \llbracket B' \rrbracket^{\mathfrak{T}} \rfloor_{k-1}$ para todo $k > 0$. Finalmente, de la *h.i.* se obtienen $D \preceq_\mu D'$ y $A' \preceq_\mu B'$, y se concluye por (S-COMP).
  - $A = A' \supset A''$. El análisis para este caso es similar al anterior. De $\llbracket A \rrbracket^{\mathfrak{T}} \rfloor_k = \llbracket A' \rrbracket^{\mathfrak{T}} \rfloor_{k-1} \supset \llbracket A'' \rrbracket^{\mathfrak{T}} \rfloor_{k-1}$ se obtienen $B = B' \supset B''$ y $\llbracket B \rrbracket^{\mathfrak{T}} \rfloor_k = \llbracket B' \rrbracket^{\mathfrak{T}} \rfloor_{k-1} \supset \llbracket B'' \rrbracket^{\mathfrak{T}} \rfloor_{k-1}$ para todo $k > 0$, por Lem. 3.3.19 (3), Def. 3.3.10 y 3.3.22 con la hipótesis $B \neq \mu$. Luego, por *h.i.* se tienen $B' \preceq_\mu A'$ y $A'' \preceq_\mu B''$, y se concluye por (S-FUNC).
  - $A = \mu V.A'$. Este caso no aplica por hipótesis.
- Si $n + m > 2$, la última regla aplicada para derivar $\llbracket A \rrbracket^{\mathfrak{T}} \rfloor_k \preceq_{\mathfrak{T}} \llbracket B \rrbracket^{\mathfrak{T}} \rfloor_k$ es necesariamente (S-UNION-T). Luego, existe una función $f : [0, n) \to [0, m)$ tal que $(\llbracket A_i \rrbracket^{\mathfrak{T}} \rfloor_k \preceq_{\mathfrak{T}} \llbracket B_{f(i)} \rrbracket^{\mathfrak{T}} \rfloor_k)_{i<n}$ para todo $k > 0$. Por *h.i.* se tiene $(A_i \preceq_\mu B_{f(i)})_{i<n}$. Finalmente, se concluye por con múltiples aplicaciones de (S-UNION-L), (S-UNION-R1) y/o (S-UNION-R2), $\oplus_{i<n} A_i \preceq_\mu \oplus_{j<m} B_j$.

□

Propiedades adicionales de los μ-tipos contractivos

Lema 3.3.32. *Sean $(A_i)_{i<n}, (B_j)_{j<m} \in \mathcal{T}$ tipos no-unión. Luego, $\oplus_{i<n} A_i \simeq_\mu \oplus_{j<m} B_j$ sii existen funciones $f : [0, n) \to [0, m)$ y $g : [0, m) \to [0, n)$ tales que $(A_i \simeq_\mu B_{f(i)})_{i<n}$ y $(A_{g(j)} \simeq_\mu B_j)_{j<m}$.*

Demostración. Por análisis de casos sobre $n + m$. Si $n = m = 1$, el resultado es inmediato con $f = g = id$ (la función identidad). Sino (*i.e.* $n + m > 2$), se concluye por Teo. 3.3.28 y Def. 3.3.22, pues la única regla aplicable es (E-UNION-T). Notar que el hecho de que $(A_i)_{i<n}$ y $(B_j)_{j<m}$ sean tipos no-unión implica $(\llbracket A_i \rrbracket^{\mathfrak{T}} \neq \oplus)_{i<n}$ y $(\llbracket B_j \rrbracket^{\mathfrak{T}} \neq \oplus)_{j<m}$ respectivamente. □

Lema 3.3.33. *Sean $(A_i)_{i<n} \in \mathcal{T}$ tipos no-unión. Luego, $\oplus_{i<n} A_i \preceq_\mu \oplus_{j<m} B_j$ sii existe una función $f : [0, n) \to [0, m)$ tal que $(A_i \preceq_\mu B_{f(i)})_{i<n}$.*

Demostración. Por análisis de casos sobre $n+m$. Si $n=m=1$, el resultado es inmediato con $f=id$ (la función identidad). Sino (*i.e.* $n+m>2$), por Def. 3.3.22, se tiene $[\![\oplus_{i<n}A_i]\!]^{\mathfrak{T}} = \oplus_{i<n}[\![A_i]\!]^{\mathfrak{T}}$ con $([\![A_i]\!]^{\mathfrak{T}} \neq \oplus)_{i<n}$, y $[\![\oplus_{j<m}B_j]\!]^{\mathfrak{T}} = \oplus_{l<m'}\mathcal{B}_l$ tal que $(\mathcal{B}_l \neq \oplus)_{l<m'}$ con $m' \geq m$, $[0,m') = \bigcup_{j<m} I_j$ y $([\![B_j]\!]^{\mathfrak{T}} = \oplus_{l\in I_j}\mathcal{B}_l)_{j<m}$. Por Teo. 3.3.30, $\oplus_{i<n}A_i \preceq_\mu \oplus_{j<m}B_j$ sii $[\![\oplus_{i<n}A_i]\!]^{\mathfrak{T}} \preceq_{\mathfrak{T}} [\![\oplus_{j<m}B_j]\!]^{\mathfrak{T}}$ y, por (S-UNION-T), esto vale sii existe $g:[0,n)\to[0,m')$ tal que $([\![A_i]\!]^{\mathfrak{T}} \preceq_{\mathfrak{T}} \mathcal{B}_{g(i)})_{i<n}$. Más aún, como $[0,m') = \bigcup_{j<m} I_j$, para todo $i<n$ se tiene $g(i)\in I_j$ para algún $j<m$. Es decir, $([\![A_i]\!]^{\mathfrak{T}} \preceq_{\mathfrak{T}} \mathcal{B}_{g(i)})_{i<n}$ sii $([\![A_i]\!]^{\mathfrak{T}} \preceq_{\mathfrak{T}} [\![B_{f(i)}]\!]^{\mathfrak{T}})_{i<n}$ con $f:[0,n)\to[0,m)$ asocia cada $i<n$ con un $j<m$ tal que $g(i)\in I_j$. □

Compatibilidad

Lema 3.3.36. *Sea $\pi_p \rhd_{\mathcal{P}} \theta \vdash_{\mathsf{p}} p : A$. Luego,* $\mathsf{pos}(p) \subseteq \mathsf{ipos}(A)$.

Demostración. Por inducción en π_p analizando la última regla aplicada.

- (P-MATCH). Luego, $p=x$ y $\mathsf{pos}(p)=\{\epsilon\}$. El resultado es inmediato pues $\epsilon\in\mathsf{ipos}(A)$ para todo A.
- (P-CONST). Luego, $p=\mathsf{c}$ y $\mathsf{pos}(p)=\{\epsilon\}$. El resultado es inmediato.
- (P-COMP). Luego, $p=p_0\,p_1$ con $\theta=\theta_0;\theta_1$ y $A=D\mathbin{@}A'$ tal que $\pi_{p_0}\rhd_{\mathcal{P}}\theta_0\vdash_{\mathsf{p}} p_0 : D$ y $\pi_{p_1}\rhd_{\mathcal{P}}\theta_1\vdash_{\mathsf{p}} p_1 : A'$. Sea $\mathsf{p}\in\mathsf{pos}(A)$. Si $\mathsf{p}=\epsilon$ el resultado es inmediato. Sino, $\mathsf{p}=i\mathsf{q}$ con $\mathsf{q}\in\mathsf{pos}(p_i)$ para algún $i\in[0,1]$. Suponer $i=0$ (el análisis para el caso $i=1$ es simétrico), es decir $\mathsf{q}\in\mathsf{pos}(p_0)$. Por *h.i.* se tienen $\mathsf{q}\in\mathsf{ipos}(D)=\mathsf{ipos}([\![D]\!]^{\mathfrak{T}})$ y, por Def. 3.3.35, $\mathsf{p}=0\mathsf{q}\in\mathsf{ipos}([\![D\mathbin{@}A']\!]^{\mathfrak{T}}) = \mathsf{ipos}(D\mathbin{@}A')$.

□

Lema 3.3.37. *Sean $\mathcal{A}\simeq_{\mathfrak{T}}\mathcal{B}$. Luego,* $\mathsf{ipos}(\mathcal{A})=\mathsf{ipos}(\mathcal{B})$.

Demostración. $\Rightarrow$) Por inducción en $\mathsf{p}\in\mathsf{ipos}(\mathcal{A})$.

- $\mathsf{p}=\epsilon$. El resultado es inmediato pues $\epsilon\in\mathsf{ipos}(\mathcal{B})$ para todo $\mathcal{B}$.
- $\mathsf{p}=i\mathsf{q}$ con $i\in[0,1]$. Suponer $i=0$ (el análisis para el caso $i=1$ es simétrico). Considerar los tipos unión-maximal (Nota 3.3.7)

$$\begin{array}{rcl} \mathcal{A} = \oplus_{i<n}\mathcal{A}_i & \text{con} & \mathcal{A}_i\neq\oplus, i<n \\ \mathcal{B} = \oplus_{j<m}\mathcal{B}_j & \text{con} & \mathcal{B}_j\neq\oplus, j<m \end{array}$$

  Por Def. 3.3.35, existe $i_0<n$ tal que $\mathsf{p}\in\mathsf{ipos}(\mathcal{A}_{i_0})$. Más aún, necesariamente es el caso por $\mathcal{A}_{i_0}=\mathcal{D}\mathbin{@}\mathcal{A}'$. Luego, $\mathsf{q}\in\mathsf{ipos}(\mathcal{D})$. Por (E-UNION-T), existe una función $f:[0,n)\to[0,m)$ tal que $\mathcal{A}_{i_0}\simeq_{\mathfrak{T}}\mathcal{B}_{f(i_0)}$. Por Lem. 3.3.14 (2), $\mathcal{B}_{f(i)}=\mathcal{D}'\mathbin{@}\mathcal{B}'$ con $\mathcal{D}\simeq_{\mathfrak{T}}\mathcal{D}'$. Por *h.i.* se tiene entonces $\mathsf{q}\in\mathsf{ipos}(\mathcal{D}')$. Se concluye por Def. 3.3.35, $\mathsf{p}\in\mathsf{ipos}(\mathcal{B})$.

$\Leftarrow$) Este caso es análogo al anterior, tomando $\mathsf{p}\in\mathsf{ipos}(\mathcal{B})$ para concluir $\mathsf{p}\in\mathsf{ipos}(\mathcal{A})$. Notar que por (E-UNION-T) también existe una función $g:[0,m)\to[0,n)$ que relaciona cada sub-árbol no-unión en $\mathcal{B}$ con su contra-parte en $\mathcal{A}$. □

Lema 3.3.39. *Sean $\mathcal{A}\preceq_{\mathfrak{T}}\mathcal{B}$. Luego,* $\mathsf{ipos}(\mathcal{A})\subseteq\mathsf{ipos}(\mathcal{B})$.

Demostración. Por inducción en $\mathsf{p}\in\mathsf{ipos}(\mathcal{A})$.

- $\mathsf{p}=\epsilon$. El resultado es inmediato pues $\epsilon\in\mathsf{ipos}(\mathcal{B})$ para todo $\mathcal{B}$.

- $\mathsf{p} = i\mathsf{q}$ con $i \in [0,1]$. Suponer $i = 0$ (el análisis para el caso $i = 1$ es simétrico). Considerar los tipos unión-maximal (Nota 3.3.7)

$$\begin{array}{rcl} \mathcal{A} = \oplus_{i<n}\mathcal{A}_i & \text{con} & \mathcal{A}_i \neq \oplus, i < n \\ \mathcal{B} = \oplus_{j<m}\mathcal{B}_j & \text{con} & \mathcal{B}_j \neq \oplus, j < m \end{array}$$

  Por Def. 3.3.35, existe $i_0 < n$ tal que $\mathsf{p} \in \mathsf{ipos}(\mathcal{A}_{i_0})$. Más aún, necesariamente es el caso por $\mathcal{A}_{i_0} = \mathcal{D} \mathbin{@} \mathcal{A}'$. Luego, $\mathsf{q} \in \mathsf{ipos}(\mathcal{D})$. Por (S-UNION-T), existe una función $f : [0,n) \to [0,m)$ tal que $\mathcal{A}_{i_0} \preceq_{\mathfrak{T}} \mathcal{B}_{f(i_0)}$. Por Lem. 3.3.19 (2), $\mathcal{B}_{f(i)} = \mathcal{D}' \mathbin{@} \mathcal{B}'$ con $\mathcal{D} \preceq_{\mathfrak{T}} \mathcal{D}'$. Por *h.i.* se tiene entonces $\mathsf{q} \in \mathsf{ipos}(\mathcal{D}')$. Se concluye por Def. 3.3.35, $\mathsf{p} \in \mathsf{ipos}(\mathcal{B})$.

□

Lema 3.3.42. *Sean $\mathcal{A} \simeq_{\mathfrak{T}} \mathcal{B}$ y $\mathsf{p} \in \mathsf{ipos}(\mathcal{A})$. Luego, $\mathcal{A}\downarrow_{\mathsf{p}} = \mathcal{B}\downarrow_{\mathsf{p}}$.*

Demostración. Por inducción en p. Notar que $\mathsf{ipos}(\mathcal{A}) = \mathsf{ipos}(\mathcal{B})$ por Lem. 3.3.37. Considerar los tipos unión-maximal (Nota 3.3.7)

$$\begin{array}{rcl} \mathcal{A} = \oplus_{i<n}\mathcal{A}_i & \text{con} & \mathcal{A}_i \neq \oplus, i < n \\ \mathcal{B} = \oplus_{j<m}\mathcal{B}_j & \text{con} & \mathcal{B}_j \neq \oplus, j < m \end{array}$$

Por (E-UNION-T), existen funciones $f : [0,n) \to [0,m)$ y $g : [0,m) \to [0,n)$ tal que $\mathcal{A}_i \simeq_{\mathfrak{T}} \mathcal{B}_{f(i)}$ y $\mathcal{A}_{g(j)} \simeq_{\mathfrak{T}} \mathcal{B}_j$ para todo $i < n$, $j < m$.

- $\mathsf{p} = \epsilon$. Por Def. 3.3.41, $\mathcal{A}\downarrow_\epsilon = \{\mathcal{A}_i(\epsilon) \mid i < n\}$ y $\mathcal{B}\downarrow_\epsilon = \{\mathcal{B}_j(\epsilon) \mid j < m\}$. Más aún, por Lem. 3.3.14, $\mathcal{A}_i(\epsilon) = \mathcal{B}_{f(i)}(\epsilon)$ y $\mathcal{A}_{g(j)}(\epsilon) = \mathcal{B}_j(\epsilon)$ para todo $i < n$, $j < m$. Luego, $\mathcal{A}\downarrow_\epsilon = \mathcal{B}\downarrow_\epsilon$.
- $\mathsf{p} = l\mathsf{q}$ con $l \in [0,1]$. Por Def. 3.3.35, $\mathcal{A}_i(\mathsf{p}) = @$ para algunos $i < n$. Del mismo modo, $\mathcal{B}_j(\mathsf{p}) = @$ para algunos $j < m$. Para todos estos i, j se tiene $\mathcal{A}_i\downarrow_{\mathsf{q}} = \mathcal{B}_{f(i)}\downarrow_{\mathsf{q}}$ y $\mathcal{A}_{g(j)}\downarrow_{\mathsf{q}} = \mathcal{B}_j\downarrow_{\mathsf{q}}$ por *h.i.* Para aquellos $i < n$, $j < m$ tal que $\mathcal{A}_i(\mathsf{p}), \mathcal{B}_j(\mathsf{p}) \neq @$, se tiene $\mathcal{A}_i\downarrow_{\mathsf{q}} = \mathcal{B}_j\downarrow_{\mathsf{q}} = \emptyset$. Luego, se concluye $\mathcal{A}\downarrow_{\mathsf{p}} = \bigcup_{i<n} \mathcal{A}_i\downarrow_{\mathsf{q}} = \bigcup_{j<m} \mathcal{B}_j\downarrow_{\mathsf{q}} = \mathcal{B}\downarrow_{\mathsf{p}}$.

□

Lema 3.3.44. *Sean $\mathcal{A} \preceq_{\mathfrak{T}} \mathcal{B}$ y $\mathsf{p} \in \mathsf{ipos}(\mathcal{A})$. Luego, $\mathcal{A}\downarrow_{\mathsf{p}} \subseteq \mathcal{B}\downarrow_{\mathsf{p}}$.*

Demostración. Por inducción en p. Notar que $\mathsf{ipos}(\mathcal{A}) \subseteq \mathsf{ipos}(\mathcal{B})$ por Lem. 3.3.39. Considerar los tipos unión-maximal (Nota 3.3.7)

$$\begin{array}{rcl} \mathcal{A} = \oplus_{i<n}\mathcal{A}_i & \text{con} & \mathcal{A}_i \neq \oplus, i < n \\ \mathcal{B} = \oplus_{j<m}\mathcal{B}_j & \text{con} & \mathcal{B}_j \neq \oplus, j < m \end{array}$$

Por (S-UNION-T), existe una función $f : [0,n) \to [0,m)$ tal que $\mathcal{A}_i \simeq_{\mathfrak{T}} \mathcal{B}_{f(i)}$ para todo $i < n$.

- $\mathsf{p} = \epsilon$. Por Def. 3.3.41, $\mathcal{A}\downarrow_\epsilon = \{\mathcal{A}_i(\epsilon) \mid i < n\}$ y $\mathcal{B}\downarrow_\epsilon = \{\mathcal{B}_j(\epsilon) \mid j < m\}$. Más aún, por Lem. 3.3.19, $\mathcal{A}_i(\epsilon) = \mathcal{B}_{f(i)}(\epsilon)$ para todo $i < n$. Luego, $\mathcal{A}\downarrow_\epsilon \subseteq \mathcal{B}\downarrow_\epsilon$.
- $\mathsf{p} = l\mathsf{q}$ con $l \in [0,1]$. Por Def. 3.3.35, $\mathcal{A}_i(\mathsf{p}) = @$ para algunos $i < n$. Para todos estos i se tiene $\mathcal{A}_i\downarrow_{\mathsf{q}} \subseteq \mathcal{B}_{f(i)}\downarrow_{\mathsf{q}}$ por *h.i.* Para aquellos $i < n$ tal que $\mathcal{A}_i(\mathsf{p}) \neq @$, se tiene $\mathcal{A}_i\downarrow_{\mathsf{q}} = \emptyset$. Luego, se concluye $\mathcal{A}\downarrow_{\mathsf{p}} = \bigcup_{i<n} \mathcal{A}_i\downarrow_{\mathsf{q}} \subseteq \bigcup_{j<m} \mathcal{B}_j\downarrow_{\mathsf{q}} = \mathcal{B}\downarrow_{\mathsf{p}}$.

□

Propiedades del sistema

Lema 3.3.48 (Generación de patrones)**.** *Sea $\pi_p \rhd_{\mathcal{P}} \theta \vdash_{\mathsf{p}} p : A$. Luego,*

1. *Si $p = x$, entonces $\theta(x) = A$.*
2. *Si $p = \mathsf{c}$, entonces $A = \mathbb{c}$.*
3. *Si $p = p_0\, p_1$, entonces $\theta = \theta_0; \theta_1$ y $A = D \mathbin{@} A'$ tal que $\pi_{p_0} \rhd_{\mathcal{P}} \theta_0 \vdash_{\mathsf{p}} p_0 : D$ y $\pi_{p_1} \rhd_{\mathcal{P}} \theta_1 \vdash_{\mathsf{p}} p_1 : A'$.*

Demostración. Por un simple análisis de la última regla aplicada al derivar π_p. Notar que hay solo una regla aplicable en cada caso. □

Lema 3.3.49 (Generación)**.** *Sea $\pi_t \rhd_{\mathcal{P}} \Gamma \vdash t : A$. Luego,*

1. *Si $t = x$, entonces $\Gamma(x) = A'$ y $A' \preceq_\mu A$.*
2. *Si $t = \mathsf{c}$, entonces $\mathbb{c} \preceq_\mu A$.*
3. *Si $t = r\, u$, entonces:*
   a) *$\exists D, A' \in \mathcal{T}$ tal que $D \mathbin{@} A' \preceq_\mu A$, $\pi_r \rhd_{\mathcal{P}} \Gamma \vdash r : D$ y $\pi_u \rhd_{\mathcal{P}} \Gamma \vdash u : A'$; o bien*
   b) *$\exists (A_i)_{i<n}, A' \in \mathcal{T}$ tal que $A' \preceq_\mu A$, $\pi_r \rhd_{\mathcal{P}} \Gamma \vdash r : \bigoplus_{i<n} A_i \supset A'$ y $\pi_u \rhd_{\mathcal{P}} \Gamma \vdash u : A_k$ para algún $k < n$.*
4. *Si $t = (p_i \to s_i)_{i<n}$, entonces $\exists (A_i)_{i<n}, B \in \mathcal{T}$ tal que $\bigoplus_{i<n} A_i \supset B \preceq_\mu A$, $(\pi_{p_i} \rhd_{\mathcal{P}} \theta_i \vdash_{\mathsf{p}} p_i : A_i)_{i<n}$, $\mathsf{cmp}([p_i : A_i]_{i<n})$ y $(\pi_{s_i} \rhd_{\mathcal{P}} \Gamma; \theta_i \vdash s_i : B)_{i<n}$.*

Demostración. Por inducción en π_t, analizando la última regla aplicada.

- (T-VAR). Luego, $t = x$ con $\Gamma(x) = A$. Se concluye (1) por (S-REFL).
- (T-CONST). Luego, $t = \mathsf{c}$ y $A = \mathbb{c}$. Se concluye (2) por (S-REFL).
- (T-COMP). Luego, $t = r\, s$ y $A = D \mathbin{@} A'$ con premisas $\pi_r \rhd_{\mathcal{P}} \Gamma \vdash r : D$ y $\pi_u \rhd_{\mathcal{P}} \Gamma \vdash u : A'$. Se concluye (3a) por (S-REFL).
- (T-ABS). Luego, $t = (p_i \to_{\theta_i} s_i)_{i<n}$ y $A = \bigoplus_{i<n} A_i \supset B$ con premisas $(\pi_{p_i} \rhd_{\mathcal{P}} \theta_i \vdash_{\mathsf{p}} p_i : A_i)_{i<n}$, $\mathsf{cmp}([p_i : A_i]_{i<n})$, $(\pi_{s_i} \rhd_{\mathcal{P}} \Gamma; \theta_i \vdash s_i : B)_{i<n}$. Se concluye (4) por (S-REFL).
- (T-APP). Luego, $t = r\, u$ con premisas $\pi_r \rhd_{\mathcal{P}} \Gamma \vdash r : \bigoplus_{i<n} A_i \supset A$ y $\pi_r \rhd_{\mathcal{P}} \Gamma \vdash u : A_k$ para algún $k < n$. Se concluye (3b) por (S-REFL).
- (T-SUBS). Luego, se tiene la premisa $\pi'_t \rhd_{\mathcal{P}} \Gamma \vdash t : A'$ con $A' \preceq_\mu A$. Por *h.i.* se obtiene $A'' \preceq_\mu A'$ con las condiciones correspondientes según el caso. Finalmente, se concluye por (S-TRANS).

□

Lema 3.3.50. *Sea $\pi \rhd_{\mathcal{P}} \Gamma \vdash t : A$. Luego,*

1. *Sea Δ un contexto de tipado tal que $\Gamma \subseteq \Delta$, entonces $\pi' \rhd_{\mathcal{P}} \Delta \vdash t : A$.*
2. *$\mathsf{fv}(t) \subseteq \mathsf{dom}(\Gamma)$.*
3. *$\pi' \rhd_{\mathcal{P}} \Gamma|_{\mathsf{fv}(t)} \vdash t : A$.*

Demostración. Se demuestra cada ítem por separado.

1. Por inducción en t. Sea $\Delta \supseteq \Gamma$ un contexto de tipado.

   - $t = x$. Por Lem. 3.3.49 (1), $\Gamma(x) = A'$ y $A' \preceq_\mu A$. Luego, $\Delta(x) = A'$ también. Se concluye por (T-VAR) y (T-SUBS).
   - $t = \mathsf{c}$. Por Lem. 3.3.49 (2), $\mathfrak{c} \preceq_\mu A$. Se concluye por (T-CONST) y (T-SUBS).
   - $t = r\,u$. Por Lem. 3.3.49 (3) se tienen dos casos:
     a) $\exists D, A' \in \mathcal{T}$ tal que $D \mathbin{@} A' \preceq_\mu A$, $\pi_r \rhd_\mathcal{P} \Gamma \vdash r : D$ y $\pi_u \rhd_\mathcal{P} \Gamma \vdash u : A'$. Por *h.i.* se tiene $\pi'_r \rhd_\mathcal{P} \Delta \vdash r : D$ y $\pi'_u \rhd_\mathcal{P} \Delta \vdash u : A'$. Luego, se concluye por (T-COMP) y (T-SUBS).
     b) $\exists (A_i)_{i<n}, A' \in \mathcal{T}$ tal que $A' \preceq_\mu A$, $\pi_r \rhd_\mathcal{P} \Gamma \vdash r : \bigoplus_{i<n} A_i \supset A'$ y $\pi_u \rhd_\mathcal{P} \Gamma \vdash u : A_k$ para algún $k < n$. Por *h.i.* se tiene $\pi'_r \rhd_\mathcal{P} \Delta \vdash r : \bigoplus_{i<n} A_i \supset A'$ y $\pi'_u \rhd_\mathcal{P} \Delta \vdash u : A_k$. Luego, se concluye por (T-APP) y (T-SUBS).
   - $t = (p_i \to_{\theta_i} s_i)_{i<n}$. Por Lem. 3.3.49 (4), $\exists (A_i)_{i<n}, B \in \mathcal{T}$ tal que $\bigoplus_{i<n} A_i \supset B \preceq_\mu A$, $(\pi_{p_i} \rhd_\mathcal{P} \theta_i \vdash_\mathsf{p} p_i : A_i)_{i<n}$, $\mathsf{cmp}([p_i : A_i]_{i<n})$ y $(\pi_{s_i} \rhd_\mathcal{P} \Gamma; \theta_i \vdash s_i : B)_{i<n}$. Sin pérdida de generalidad se asume $\mathsf{dom}(\Delta) \cap \mathsf{dom}(\theta_i) = \emptyset$ para todo $i < n$. Luego, $\Delta; \theta_i \supseteq \Gamma; \theta_i$ es un contexto de tipado válido y por *h.i.* se tiene $\pi'_{s_i} \rhd_\mathcal{P} \Delta; \theta_i \vdash s_i : B$ para todo $i < n$. Finalmente, se concluye por (T-ABS) y (T-SUBS).

2. Por inducción en t.

   - $t = x$. Por Lem. 3.3.49 (1), $\Gamma(x) = A'$. Luego, $\mathsf{fv}(t) = \{x\} \subseteq \mathsf{dom}(\Gamma)$.
   - $t = \mathsf{c}$. Trivialmente, $\mathsf{fv}(t) = \emptyset \subseteq \mathsf{dom}(\Gamma)$.
   - $t = r\,u$. Por Lem. 3.3.49 (3), $\pi_r \rhd_\mathcal{P} \Gamma \vdash r : B$ y $\pi_u \rhd_\mathcal{P} \Gamma \vdash u : C$ para tipos B y C adecuados. Luego, por *h.i.* se tienen $\mathsf{fv}(r) \subseteq \mathsf{dom}(\Gamma)$ y $\mathsf{fv}(u) \subseteq \mathsf{dom}(\Gamma)$, por lo que $\mathsf{fv}(r\,u) = \mathsf{fv}(u) \cup \mathsf{fv}(u) \subseteq \mathsf{dom}(\Gamma)$ también.
   - $t = (p_i \to_{\theta_i} s_i)_{i<n}$. Por Lem. 3.3.49 (4), $\exists (A_i)_{i<n}, B \in \mathcal{T}$ tal que $(\pi_{p_i} \rhd_\mathcal{P} \theta_i \vdash_\mathsf{p} p_i : A_i)_{i<n}$ y $(\pi_{s_i} \rhd_\mathcal{P} \Gamma; \theta_i \vdash s_i : B)_{i<n}$. En particular, $\mathsf{dom}(\theta_i) = \mathsf{fv}(p_i)$ para todo $i < n$. Por *h.i.* se tiene $\mathsf{fv}(s_i) \subseteq \mathsf{dom}(\Gamma; \theta_i) = \mathsf{dom}(\Gamma) \uplus \mathsf{fv}(p_i)$ (*i.e.* $\mathsf{fv}(s_i) \setminus \mathsf{fv}(p_i) \subseteq \mathsf{dom}(\Gamma)$) para todo $i < n$. Luego, se concluye pues $\mathsf{fv}(s) = \bigcup_{i<n} (\mathsf{fv}(s_i) \setminus \mathsf{fv}(p_i)) \subseteq \mathsf{dom}(\Gamma)$.

3. Por inducción en t.

   - $t = x$. Por Lem. 3.3.49 (1), $\Gamma(x) = A'$ y $A' \preceq_\mu A$. Más aún, por definición $\Gamma|_{\mathsf{fv}(t)}(x) = A'$ también. Luego, se concluye por (T-VAR) y (T-SUBS).
   - $t = \mathsf{c}$. Por Lem. 3.3.49 (2), $\mathfrak{c} \preceq_\mu A$. Se concluye por (T-CONST) y (T-SUBS).
   - $t = r\,u$. Por Lem. 3.3.49 (3) se tienen dos casos:
     a) $\exists D, A' \in \mathcal{T}$ tal que $D \mathbin{@} A' \preceq_\mu A$, $\pi_r \rhd_\mathcal{P} \Gamma \vdash r : D$ y $\pi_u \rhd_\mathcal{P} \Gamma \vdash u : A'$. Por *h.i.* se tiene $\pi'_r \rhd_\mathcal{P} \Gamma|_{\mathsf{fv}(r)} \vdash r : D$ y $\pi'_u \rhd_\mathcal{P} \Gamma|_{\mathsf{fv}(u)} \vdash u : A'$. Como Γ es un contexto de tipado válido, $\Gamma|_{\mathsf{fv}(t)} = \Gamma|_{\mathsf{fv}(r)} \cup \Gamma|_{\mathsf{fv}(u)} \subseteq \Gamma$ también lo es. Luego, por Lem. 3.3.50 (1), se tienen $\pi'_r \rhd_\mathcal{P} \Gamma|_{\mathsf{fv}(t)} \vdash r : D$ y $\pi'_u \rhd_\mathcal{P} \Gamma|_{\mathsf{fv}(t)} \vdash u : A'$. Finalmente, se concluye por (T-COMP) y (T-SUBS).
     b) $\exists (A_i)_{i<n}, A' \in \mathcal{T}$ tal que $A' \preceq_\mu A$, $\pi_r \rhd_\mathcal{P} \Gamma \vdash r : \bigoplus_{i<n} A_i \supset A'$ y $\pi_u \rhd_\mathcal{P} \Gamma \vdash u : A_k$ para algún $k < n$. Por *h.i.* se tiene $\pi'_r \rhd_\mathcal{P} \Gamma|_{\mathsf{fv}(r)} \vdash r : \bigoplus_{i<n} A_i \supset A'$ y $\pi'_u \rhd_\mathcal{P} \Gamma|_{\mathsf{fv}(u)} \vdash u : A_k$. Nuevamente se tiene $\Gamma|_{\mathsf{fv}(t)} = \Gamma|_{\mathsf{fv}(r)} \cup \Gamma|_{\mathsf{fv}(u)} \subseteq \Gamma$ un contexto de tipado válido. Luego, por Lem. 3.3.50 (1), $\pi'_r \rhd_\mathcal{P} \Gamma|_{\mathsf{fv}(t)} \vdash r : \bigoplus_{i<n} A_i \supset A'$ y $\pi'_u \rhd_\mathcal{P} \Gamma|_{\mathsf{fv}(t)} \vdash u : A_k$. Se concluye por (T-APP) y (T-SUBS).

- $t = (p_i \to_{\theta_i} s_i)_{i<n}$. Por Lem. 3.3.49 (4), $\exists (A_i)_{i<n}, B \in \mathcal{T}$ tal que $\oplus_{i<n} A_i \supset B \preceq_\mu A$, $(\pi_{p_i} \rhd_\mathcal{P} \theta_i \vdash_\mathsf{p} p_i : A_i)_{i<n}$, $\mathsf{cmp}([p_i : A_i]_{i<n})$ y $(\pi_{s_i} \rhd_\mathcal{P} \Gamma; \theta_i \vdash s_i : B)_{i<n}$. Es inmediato ver que $(\Gamma;\theta_i)|_{\mathsf{fv}(s_i)} \subseteq \Gamma|_{\mathsf{fv}(s_i)};\theta_i$ para todo $i < n$. Más aún, como $\mathsf{dom}(\theta_i) = \mathsf{fv}(p_i)$, se tiene $\Gamma|_{\mathsf{fv}(s_i)} \subseteq \Gamma|_{(\mathsf{fv}(s_i)\setminus\mathsf{fv}(p_i))} \subseteq \Gamma|_{\bigcup_{i<n}(\mathsf{fv}(s_i)\setminus\mathsf{fv}(p_i))}$. Luego, $\Gamma|_{\mathsf{fv}(s)};\theta_i \supseteq \Gamma|_{\mathsf{fv}(s_i)};\theta_i \supseteq (\Gamma;\theta_i)|_{\mathsf{fv}(s_i)}$ es un contexto de tipado válido. Entonces, por Lem. 3.3.50 (1), $\pi_{s_i} \rhd_\mathcal{P} \Gamma|_{\mathsf{fv}(s)};\theta_i \vdash s_i : B$ para todo $i < n$. Finalmente, se concluye por (T-ABS) y (T-SUBS).

□

Lema 3.3.51 (Sustitución)**.** *Sean $\pi_s \rhd_\mathcal{P} \Gamma;\theta \vdash s : A$ y $\pi_\sigma \rhd_\mathcal{P} \Gamma \vdash \sigma : \theta$. Luego, $\pi_{\sigma s} \rhd_\mathcal{P} \Gamma \vdash \sigma s : A$.*

Demostración. Por inducción en s.

- $s = x$. Por Lem. 3.3.49 (1), $(\Gamma;\theta)(x) = A'$ y $A' \preceq_\mu A$. Si $x \in \mathsf{dom}(\sigma)$, como $\mathsf{dom}(\sigma) = \mathsf{dom}(\theta)$, se tiene $\theta(x) = A'$ y, por hipótesis, $\rhd_\mathcal{P} \Gamma \vdash \sigma(x) : \theta(x)$. Luego, se concluye por (T-SUBS). Si no (*i.e.* $x \notin \mathsf{dom}(\theta)$), se tiene $\Gamma(x) = A'$ y $\sigma x = x$, en cuyo caso se concluye por (T-VAR) y (T-SUBS).
- $s = \mathsf{c}$. Luego, $\sigma s = \mathsf{c}$ y, por Lem. 3.3.49 (2), $\mathbb{c} \preceq_\mu A$. Se concluye por (T-CONST) y (T-SUBS).
- $t = r\,u$. Luego, $\sigma s = \sigma r\,\sigma u$ y, por Lem. 3.3.49 (3) se tienen dos casos:
  a) $\exists D, A' \in \mathcal{T}$ tal que $D \mathbin{@} A' \preceq_\mu A$, $\pi_r \rhd_\mathcal{P} \Gamma \vdash r : D$ y $\pi_u \rhd_\mathcal{P} \Gamma \vdash u : A'$. Por *h.i.* se tiene $\pi_{\sigma r} \rhd_\mathcal{P} \Gamma \vdash \sigma r : D$ y $\pi_{\sigma u} \rhd_\mathcal{P} \Gamma \vdash \sigma u : A'$. Luego, se concluye por (T-COMP) y (T-SUBS).
  b) $\exists (A_i)_{i<n}, A' \in \mathcal{T}$ tal que $A' \preceq_\mu A$, $\pi_r \rhd_\mathcal{P} \Gamma \vdash r : \oplus_{i<n} A_i \supset A'$ y $\pi_u \rhd_\mathcal{P} \Gamma \vdash u : A_k$ para algún $k < n$. Por *h.i.* se tiene $\pi_{\sigma r} \rhd_\mathcal{P} \Gamma \vdash \sigma r : \oplus_{i<n} A_i \supset A'$ y $\pi_{\sigma u} \rhd_\mathcal{P} \Gamma \vdash \sigma u : A_k$. Luego, se concluye por (T-APP) y (T-SUBS).
- $s = (p_i \to_{\theta_i} s_i)_{i<n}$. Sin perdida de generalidad se asume σ $\mathtt{evita}$ θ_i para todo $i < n$. Luego, $\sigma s = (p_i \to_{\theta_i} \sigma s_i)_{i<n}$. Por Lem. 3.3.49 (4), $\exists (A_i)_{i<n}, B \in \mathcal{T}$ tal que $\oplus_{i<n} A_i \supset B \preceq_\mu A$, $(\pi_{p_i} \rhd_\mathcal{P} \theta_i \vdash_\mathsf{p} p_i : A_i)_{i<n}$, $\mathsf{cmp}([p_i : A_i]_{i<n})$ y $(\pi_{s_i} \rhd_\mathcal{P} \Gamma;\theta_i \vdash s_i : B)_{i<n}$. Más aún, apelando al Lem. 3.3.50 (1) sobre la definición de π_σ, se tiene $\rhd_\mathcal{P} \Gamma;\theta_i \vdash \sigma : \theta$ para todo $i < n$. Luego, por *h.i.* $(\pi_{\sigma s_i} \rhd_\mathcal{P} \Gamma;\theta_i \vdash \sigma s_i : B)_{i<n}$. Finalmente, se concluye por (T-ABS) y (T-SUBS).

□

Lema 3.3.52. *Sea $\pi_d \rhd_\mathcal{P} \Gamma \vdash d : A$ con $d \in \mathbb{D}_{\mathsf{CAP}}$. Luego, existe $D \in \mathcal{D}$ no-unión tal que $D \preceq_\mu A$ y $\pi'_d \rhd_\mathcal{P} \Gamma \vdash d : D$. Más aún,*

1. *Si $d = \mathsf{c}$, entonces $D = \mathbb{c}$.*
2. *Si $d = d'\,t$, entonces $D = D' \mathbin{@} A'$ tal que $\pi'_{d'} \rhd_\mathcal{P} \Gamma \vdash d' : D'$ y $\pi'_t \rhd_\mathcal{P} \Gamma \vdash t : A'$.*

Demostración. Por inducción en d.

- $d = \mathsf{c}$. Por Lem. 3.3.49 (2), $\mathbb{c} \preceq_\mu A$. Se concluye por (T-CONST) con $D = \mathbb{c}$.
- $d = d'\,t$. Por Lem. 3.3.49 (3) se tienen dos casos:
  a) $\exists D', A' \in \mathcal{T}$ tal que $D' \mathbin{@} A' \preceq_\mu A$, $\pi_{d'} \rhd_\mathcal{P} \Gamma \vdash d : D'$ y $\pi_t \rhd_\mathcal{P} \Gamma \vdash t : A'$. Se concluye por (T-COMP) con $D = D' \mathbin{@} A'$.
  b) $\exists (A_i)_{i<n}, A' \in \mathcal{T}$ tal que $A' \preceq_\mu A$, $\pi_{d'} \rhd_\mathcal{P} \Gamma \vdash d' : \oplus_{i<n} A_i \supset A'$. Por *h.i.* existe $D \in \mathcal{D}$ no-unión tal que $D \preceq_\mu \oplus_{i<n} A_i \supset A'$. Más aún, por Teo. 3.3.31 (3), $D = D' \supset D''$, lo que lleva a una contradicción con $D \in \mathcal{D}$. La contradicción viene se suponer que el Lem. 3.3.49 (3b) aplica para $d \in \mathbb{D}_{\mathsf{CAP}}$. Luego, este caso no es posible.

□

Lema 3.3.53. *Sean* $\pi_u \rhd_{\mathcal{P}} \Gamma \vdash u : B$, $\pi_p \rhd_{\mathcal{P}} \theta \vdash_{\mathsf{p}} p : A$ *y* $\{\!\{p \setminus u\}\!\} = \texttt{fail}$. *Luego,* $B \not\preceq_\mu A$.

Demostración. Por inducción en p. Notar que $\{\!\{p \setminus u\}\!\} = \texttt{fail}$ implica $u \in \mathbb{M}_{\mathsf{CAP}}$.

- $p = x$. Este caso no aplica dado que $\{\!\{p \setminus u\}\!\} = \texttt{fail}$ por hipótesis.
- $p = \texttt{c}$. Por Lem. 3.3.48 (2), $\mathbb{c} = A$. Se analiza la forma de $u \in \mathbb{M}_{\mathsf{CAP}}$.
  - $u = \texttt{d} \neq \texttt{c}$. Por Lem. 3.3.52 (1), $\mathbb{d} \preceq_\mu B$. Suponer $B \preceq_\mu A$. Luego, por (S-TRANS), se tiene $\mathbb{d} \preceq_\mu \mathbb{c}$, lo que lleva a una contradicción con la Teo. 3.3.31 (1). La contradicción viene de suponer $B \preceq_\mu A$, por lo que se concluye $B \not\preceq_\mu A$.
  - $u = d\,u'$. Por Lem. 3.3.52 (2), $\exists D', B' \in \mathcal{T}$ tal que $D' \mathbin{@} B' \preceq_\mu B$. Como en el caso anterior, si $B \preceq_\mu A$, por (S-TRANS), $D' \mathbin{@} B' \preceq_\mu \mathbb{c}$ y se tiene una contradicción con la Teo. 3.3.31. Luego, $B \not\preceq_\mu A$.
  - $u = (q_j \to_{\theta_j} u_j)_{j<m}$. Por Lem. 3.3.49 (4), $\exists (B_j)_{j<m}, B' \in \mathcal{T}$ tal que $\oplus_{j<m} B_j \supset B' \preceq_\mu B$. Nuevamente, si $B \preceq_\mu A$ se tiene $\oplus_{j<m} B_j \supset B' \preceq_\mu \mathbb{c}$, lo que contradice la Teo. 3.3.31.
- $p = p_0\,p_1$. Por Lem. 3.3.48 (3), $\theta = \theta_0; \theta_1$ y $A = D \mathbin{@} A'$ tal que $\pi_{p_0} \rhd_{\mathcal{P}} \theta_0 \vdash_{\mathsf{p}} p_0 : D$ y $\pi_{p_1} \rhd_{\mathcal{P}} \theta_1 \vdash_{\mathsf{p}} p_1 : A'$. Se analiza la forma de $u \in \mathbb{M}_{\mathsf{CAP}}$.
  - $u = \texttt{d}$. Por Lem. 3.3.52 (1), $\mathbb{d} \preceq_\mu B$. Nuevamente, si $B \preceq_\mu A$, por (S-TRANS), $\mathbb{d} \preceq_\mu D \mathbin{@} A'$ y se tiene una contradicción con la Teo. 3.3.31. Luego, $B \not\preceq_\mu A$.
  - $u = d\,u'$. Luego, el mismatch es interno pues se asumen patrones lineales (*cf.* Sec. 3.2), por lo que la falla no se produce en la unión disjunta. Es decir, se tiene $\{\!\{p_0 \setminus d\}\!\} = \texttt{fail}$ o $\{\!\{p_1 \setminus u'\}\!\} = \texttt{fail}$. Por Lem. 3.3.52 (2), $\exists D', B' \in \mathcal{T}$ tal que $D' \mathbin{@} B' \preceq_\mu B$, $\pi_d \rhd_{\mathcal{P}} \Gamma \vdash d : D'$ y $\pi_{u'} \rhd_{\mathcal{P}} \Gamma \vdash u' : B'$. Luego, por *h.i.* se tiene $D' \not\preceq_\mu D$ o $B' \not\preceq_\mu A'$. Por otro lado, suponer $B \preceq_\mu A$. Por (S-TRANS) se tiene $D' \mathbin{@} B' \preceq_\mu D \mathbin{@} A$ y, por Teo. 3.3.31 (2), $D' \preceq_\mu D$ y $B' \preceq_\mu A'$, lo que lleva a una contradicción. Luego, $B \not\preceq_\mu A$.
  - $u = (q_j \to_{\theta_j} u_j)_{j<m}$. Por Lem. 3.3.49 (4), $\exists (B_j)_{j<m}, B' \in \mathcal{T}$ tal que $\oplus_{j<m} B_j \supset B' \preceq_\mu B$. Nuevamente, si $B \preceq_\mu A$ se tiene $\oplus_{j<m} B_j \supset B' \preceq_\mu D \mathbin{@} A$, lo que contradice la Teo. 3.3.31.

□

Lema 3.3.54. *Sean* $\pi_u \rhd_{\mathcal{P}} \Gamma \vdash u : A$, $\pi_p \rhd_{\mathcal{P}} \theta \vdash_{\mathsf{p}} p : A$ *y* $\{\!\{p \setminus u\}\!\} = \sigma$. *Luego,* $\pi_\sigma \rhd_{\mathcal{P}} \Gamma \vdash \sigma : \theta$.

Demostración. Por inducción en p.

- $p = x$. Luego, $\sigma = \{x \setminus u\}$ y, por Lem. 3.3.48 (1), $\theta(x) = A$. Más aún, $\mathsf{dom}(\theta) = \mathsf{fv}(p) = \{x\} = \mathsf{dom}(\sigma)$. El resultado es inmediato por hipótesis.
- $p = \texttt{c}$. La propiedad se satisface vacuamente pues $\mathsf{dom}(\theta) = \emptyset = \mathsf{dom}(\sigma)$.
- $p = p_0\,p_1$. Luego, $u = d\,u'$ y $\sigma = \sigma_0 \uplus \sigma_1$ con $\sigma_0 = \{\!\{p_0 \setminus d\}\!\}$ y $\sigma_1 = \{\!\{p_1 \setminus u'\}\!\}$. Por Lem. 3.3.48 (3), $\theta = \theta_0; \theta_1$ y $A = D \mathbin{@} A'$ tal que $\pi_{p_0} \rhd_{\mathcal{P}} \theta_0 \vdash_{\mathsf{p}} p_0 : D$ y $\pi_{p_1} \rhd_{\mathcal{P}} \theta_1 \vdash_{\mathsf{p}} p_1 : A'$. Por otro lado, por Lem. 3.3.52 (2), $\exists D', A'' \in \mathcal{T}$ tal que $D' \mathbin{@} A'' \preceq_\mu A$, $\pi_d \rhd_{\mathcal{P}} \Gamma \vdash d : D'$ y $\pi_{u'} \rhd_{\mathcal{P}} \Gamma \vdash u' : A''$. Más aún, de $D' \mathbin{@} A'' \preceq_\mu A = D \mathbin{@} A'$, por Teo. 3.3.31 (2), se tiene $D' \preceq_\mu D$ y $A'' \preceq_\mu A'$. Luego, por (T-SUBS), se derivan $\rhd_{\mathcal{P}} \Gamma \vdash d : D$ y $\rhd_{\mathcal{P}} \Gamma \vdash u' : A'$. Finalmente, por *h.i.* se tiene $\pi_{\sigma_0} \rhd_{\mathcal{P}} \Gamma \vdash \sigma_0 : \theta_0$ y $\pi_{\sigma_1} \rhd_{\mathcal{P}} \Gamma \vdash \sigma_0 : \theta_1$. Es seguro concluir dado que $\mathsf{dom}(\theta_0) \cap \mathsf{dom}(\theta_1) = \emptyset$ y $\mathsf{dom}(\sigma_0) \cap \mathsf{dom}(\sigma_1) = \emptyset$.

□

Adecuación del sistema

Lema 3.4.2. *Sean* $\pi_v \rhd_{\mathcal{P}} \vdash v : A$ *y* $\pi_p \rhd_{\mathcal{P}} \theta \vdash_{\mathsf{p}} p : A$. *Luego,* $\{\!\{p \setminus v\}\!\} = \sigma$.

Demostración. Por inducción en p. Notar que v es un término cerrado por estar tipado con contexto vacío (*cf.* Lem. 3.3.50). Luego, v es necesariamente un data structure o una abstracción.

- $p = x$. El resultado es inmediato con $\sigma = \{x \setminus v\}$.
- $p = \mathsf{c}$. Por Lem. 3.3.48 (2), $A = \mathbb{c}$. Suponer $v = (q_j \to_{\theta_j} s_j)_{j<m}$. Por Lem. 3.3.49 (4), $\exists B, B' \in \mathcal{T}$ tal que $B \supset B' \preceq_\mu \mathbb{c}$, lo que lleva a una contradicción con Teo. 3.3.31. Luego, v es necesariamente un data structure. Por Lem. 3.3.52, $\exists D \in \mathcal{D}$ no-unión tal que $D \preceq_\mu \mathbb{c}$ y $\rhd_{\mathcal{P}} \vdash v : D$. Más aún, suponer que vale Lem. 3.3.52 (2). Entonces, $D = D' \mathbin{@} A' \preceq_\mu \mathbb{c}$, lo que contradice la Teo. 3.3.31. Luego, es necesariamente el caso Lem. 3.3.52 (1) con $u = \mathsf{c}$. Se concluye pues $\{\!\{p \setminus u\}\!\} = \{\}$.
- $p = p_0\, p_1$. Por Lem. 3.3.48 (3), $\theta = \theta_0; \theta_1$ y $A = D \mathbin{@} A'$ tal que $\pi_{p_0} \rhd_{\mathcal{P}} \theta_0 \vdash_{\mathsf{p}} p_0 : D$ y $\pi_{p_1} \rhd_{\mathcal{P}} \theta_1 \vdash_{\mathsf{p}} p_1 : A'$. Como antes, si $v = (q_j \to_{\theta_j} s_j)_{j<m}$, por Lem. 3.3.49 (4), $\exists B, B' \in \mathcal{T}$ tal que $B \supset B' \preceq_\mu D \mathbin{@} A'$, lo que lleva a una contradicción con Teo. 3.3.31. Luego, v es necesariamente un data structure. Por Lem. 3.3.52 (2), $\exists D \in \mathcal{D}$ no-unión tal que $D \preceq_\mu D \mathbin{@} A'$ y $\rhd_{\mathcal{P}} \vdash v : D$. Nuevamente por Teo. 3.3.31, se puede ver que es necesariamente el caso (2) del Lem. 3.3.52. Luego, $v = v_0\, v_1$ y $D = D' \mathbin{@} A''$ tal que $\rhd_{\mathcal{P}} \vdash v_0 : D'$ y $\rhd_{\mathcal{P}} \vdash v_1 : A''$. Más aún, Teo. 3.3.31 (2), $D' \preceq_\mu D$ y $A'' \preceq_\mu A'$. Luego, por (T-SUBS), $\rhd_{\mathcal{P}} \vdash v_0 : D$ y $\rhd_{\mathcal{P}} \vdash v_1 : A'$. Por *h.i.* se tienen $\{\!\{p_0 \setminus v_0\}\!\} = \sigma_0$ y $\{\!\{p_1 \setminus v_1\}\!\} = \sigma_1$. Finalmente, como $\theta = \theta_0; \theta_1$ (*i.e.* la unión es disjunta) y $(\mathsf{dom}(\sigma_i) = \theta_i)_{i<2}$, se concluye con $\{\!\{p \setminus v\}\!\} = \sigma_0 \uplus \sigma_1$.

□

Chequeo de equivalencia y sub-tipado

Chequeo de equivalencia

Lema 3.6.6. *Sean* $A = \mathcal{A}\langle \vec{A} \rangle$ *y* $\mathsf{p}_0, \ldots, \mathsf{p}_{n-1} \in \mathsf{ppos}(A)$ *las posiciones de* $\Box$ *en* $\mathcal{A}$ *(enumeradas de izquierda a derecha). Luego,* $[\![A]\!]^{\mathfrak{T}} = \mathcal{A}^{\setminus\mu}\langle [\![A{\downarrow_{\mathsf{p}_0}}]\!]^{\mathfrak{T}}, \ldots, [\![A{\downarrow_{\mathsf{p}_{n-1}}}]\!]^{\mathfrak{T}} \rangle$.

Demostración. Por inducción en el contexto $\mathcal{A}$.

- $\mathcal{A} = \Box$. Luego, $n = 1$ con $A_0 = A$ y $\mathsf{p}_0 = \epsilon$ por lo que el resultado es inmediato de las Def. 3.6.4 y 3.6.5.
- $\mathcal{A} = \mu V.\mathcal{A}'$. Luego, $(\mathsf{p}_i = \mathsf{0q}_i)_{i<n}$. Por Def. 3.3.22, $[\![A]\!]^{\mathfrak{T}} = [\![\mathcal{A}\langle \vec{A} \rangle]\!]^{\mathfrak{T}} = [\![\{V \setminus A\}\mathcal{A}'\langle \vec{A} \rangle]\!]^{\mathfrak{T}}$. Por *h.i.* se tiene
$$[\![A]\!]^{\mathfrak{T}} = \mathcal{A}'^{\setminus\mu}\langle [\![\{V \setminus A\}\mathcal{A}'\langle \vec{A} \rangle{\downarrow_{\mathsf{q}_0}}]\!]^{\mathfrak{T}}, \ldots, [\![\{V \setminus A\}\mathcal{A}'\langle \vec{A} \rangle{\downarrow_{\mathsf{q}_{n-1}}}]\!]^{\mathfrak{T}} \rangle$$
Luego, por Def. 3.6.4, $[\![A]\!]^{\mathfrak{T}} = \mathcal{A}'^{\setminus\mu}\langle [\![A{\downarrow_{\mathsf{p}_0}}]\!]^{\mathfrak{T}}, \ldots, [\![A{\downarrow_{\mathsf{p}_{n-1}}}]\!]^{\mathfrak{T}} \rangle$. Finalmente, se concluye por Def. 3.6.5, dado que $\mathcal{A}^{\setminus\mu} = \mathcal{A}'^{\setminus\mu}$.
- $\mathcal{A} = \mathcal{A}_0 \oplus \mathcal{A}_1$. Luego, existe $k < n$ tal que $(\mathsf{p}_i = \mathsf{0q}_i)_{i<k}$, $(\mathsf{p}_i = \mathsf{1q}_i)_{k<i<n}$ y $\vec{A} = \vec{A'}, \vec{A''}$ con $\vec{A'} = A_0, \ldots, A_{k-1}$ y $\vec{A''} = A_k, \ldots, A_{n-1}$. Por Def. 3.3.22, $[\![A]\!]^{\mathfrak{T}} = [\![\mathcal{A}_0\langle \vec{A'} \rangle \oplus \mathcal{A}_1\langle \vec{A''} \rangle]\!]^{\mathfrak{T}} = [\![\mathcal{A}_0\langle \vec{A'} \rangle]\!]^{\mathfrak{T}} \oplus [\![\mathcal{A}_1\langle \vec{A''} \rangle]\!]^{\mathfrak{T}}$. Por *h.i.* se tiene
$$[\![A]\!]^{\mathfrak{T}} = \mathcal{A}_0{}^{\setminus\mu}\langle [\![\mathcal{A}_0\langle \vec{A'} \rangle{\downarrow_{\mathsf{q}_0}}]\!]^{\mathfrak{T}}, \ldots, [\![\mathcal{A}_0\langle \vec{A'} \rangle{\downarrow_{\mathsf{q}_{k-1}}}]\!]^{\mathfrak{T}} \rangle \oplus \mathcal{A}_1{}^{\setminus\mu}\langle [\![\mathcal{A}_1\langle \vec{A''} \rangle{\downarrow_{\mathsf{q}_k}}]\!]^{\mathfrak{T}} \rangle, \ldots, [\![\mathcal{A}_1\langle \vec{A''} \rangle{\downarrow_{\mathsf{q}_{n-1}}}]\!]^{\mathfrak{T}}$$
Luego, por Def. 3.6.4,
$$[\![A]\!]^{\mathfrak{T}} = \mathcal{A}_0{}^{\setminus\mu}\langle [\![A{\downarrow_{\mathsf{p}_0}}]\!]^{\mathfrak{T}}, \ldots, [\![A{\downarrow_{\mathsf{p}_{k-1}}}]\!]^{\mathfrak{T}} \rangle \oplus \mathcal{A}_1{}^{\setminus\mu}\langle [\![A{\downarrow_{\mathsf{p}_k}}]\!]^{\mathfrak{T}}, \ldots, [\![A{\downarrow_{\mathsf{p}_{n-1}}}]\!]^{\mathfrak{T}} \rangle$$
Finalmente, se concluye por Def. 3.6.5, dado que $\mathcal{A}^{\setminus\mu} = \mathcal{A}_0{}^{\setminus\mu} \oplus \mathcal{A}_1{}^{\setminus\mu}$.

□

Lema 3.6.7. *Sea A un μ-tipo contractivo tal que $[\![A]\!]^{\mathfrak{T}} = \oplus_{i<n}\mathcal{A}_i$ con $(\mathcal{A}_i \neq \oplus)_{i<n}$. Luego, existen $n' \leq n$, $\oplus$-contextos maximales $\mathcal{A}$ y $(\mathcal{A}_l)_{l<n'}$, μ-tipos contractivos $(A_l \neq \oplus)_{l<n'}$ y funciones $b, e : [0, n') \to [0, n)$ tales que $A = \mathcal{A}\langle\vec{A}\rangle$ y $([\![A_l]\!]^{\mathfrak{T}} = \mathcal{A}_l\langle\mathcal{A}_{b(l)}, \ldots, \mathcal{A}_{e(l)}\rangle)_{l<n'}$.*

Demostración. Por Nota 3.6.2, A puede ser unívocamente expresado por un $\mu\oplus$-contexto maximal $\mathcal{B}$ y μ-tipos contractivos $B_1, \ldots, B_m$ tal que $(B_j \neq \mu, \oplus)_{j<m}$, *i.e.* $A = \mathcal{B}\langle\vec{B}\rangle$. Más aún, sean $\mathsf{q}_1, \ldots, \mathsf{q}_m \in \mathsf{ppos}(A)$ las posiciones de $\Box$ en $\mathcal{B}$ (enumeradas de izquierda a derecha), con una simple inducción en A puede verificarse que $m = n$ y $([\![\mathcal{B}\langle\vec{B}\rangle{\downarrow_{\mathsf{q}_i}}]\!]^{\mathfrak{T}} = \mathcal{A}_i)_{i<n}$ por Def. 3.3.22 y 3.6.4. Luego, es posible descomponer $\mathcal{B}$ en dos partes: por un lado un $\oplus$-contexto maximal $\mathcal{A}$ con $n' \leq n$ agujeros, y por otro múltiples $\mu\oplus$-contextos $\mathcal{B}_1, \ldots, \mathcal{B}_{n'}$ tal que $\mathcal{B} = \mathcal{A}\langle\vec{\mathcal{B}}\rangle$, donde $(\mathcal{B}_l = \Box, \mu)_{l<n'}$. Entonces, existen μ-tipos contractivos $A_1, \ldots, A_{n'}$ y funciones $b, e : [0, n') \to [0, n)$ tal que $(A_l = \mathcal{B}_l\langle B_{b(l)}, \ldots, B_{e(l)}\rangle)_{l<n'}$ y, por lo tanto, $A = \mathcal{A}\langle\vec{A}\rangle$. Notar que $(A_l \neq \oplus)_{l<n'}$.

Sean $\mathsf{q}'_{b(l)}, \ldots, \mathsf{q}'_{e(l)} \in \mathsf{ppos}(A_l)$ las posiciones de $\Box$ en $\mathcal{B}_l$. Luego, por Lem. 3.6.6, se tiene $[\![A_l]\!]^{\mathfrak{T}} = \mathcal{B}_l{}^{\setminus\mu}\langle[\![A_l{\downarrow_{\mathsf{q}'_{b(l)}}}]\!]^{\mathfrak{T}}, \ldots, [\![A_l{\downarrow_{\mathsf{q}'_{e(l)}}}]\!]^{\mathfrak{T}}\rangle$. Más aún, como $\mathcal{B} = \mathcal{A}\langle\vec{\mathcal{B}}\rangle$ y $\mathcal{A}$ es un $\oplus$-contexto, se tiene que q'_i sufijo de q_i para todo $i \in [b(l), e(l)]$. Entonces, por Def. 3.6.4

$$[\![A_l]\!]^{\mathfrak{T}} = \mathcal{B}_l{}^{\setminus\mu}\langle[\![\mathcal{B}\langle\vec{B}\rangle{\downarrow_{\mathsf{q}_{b(l)}}}]\!]^{\mathfrak{T}}, \ldots, [\![\mathcal{B}\langle\vec{B}\rangle{\downarrow_{\mathsf{q}_{e(l)}}}]\!]^{\mathfrak{T}}\rangle$$

y de $([\![\mathcal{B}\langle\vec{B}\rangle{\downarrow_{\mathsf{q}_i}}]\!]^{\mathfrak{T}} = \mathcal{A}_i)_{i<n}$ se obtiene $[\![A_l]\!]^{\mathfrak{T}} = \mathcal{B}_l{}^{\setminus\mu}\langle\mathcal{A}_{b(l)}, \ldots, \mathcal{A}_{e(l)}\rangle$. Por último, se toma $(\mathcal{A}_l = \mathcal{B}_l{}^{\setminus\mu})_{l<n'}$ para concluir. □

Lema 3.6.8. *$A \simeq_{\vec{\mu}} B$ sii $\forall k \in \mathbb{N}.\lfloor[\![A]\!]^{\mathfrak{T}}\rfloor_k \simeq_{\mathfrak{T}} \lfloor[\![B]\!]^{\mathfrak{T}}\rfloor_k$.*

Demostración. Para esta prueba se denota con $\#_\mu(A)$ el número de ocurrencias del constructor μ en el $\mu\oplus$-contexto maximal $\mathcal{A}$ que caracteriza $A = \mathcal{A}\langle\vec{A}\rangle$ (*cf.* Nota 3.6.2).

$\Rightarrow$) Dado $A \simeq_{\vec{\mu}} B$, existe $\mathcal{S} \in \wp(\mathcal{T} \times \mathcal{T})$ tal que $\mathcal{S} \subseteq \Phi_{\simeq_{\vec{\mu}}}(\mathcal{S})$ y $\langle A, B\rangle \in \mathcal{S}$. Se define entonces $\mathcal{R}(\mathcal{S}) \triangleq \{\langle\lfloor[\![A']\!]^{\mathfrak{T}}\rfloor_k, \lfloor[\![B']\!]^{\mathfrak{T}}\rfloor_k\rangle \mid \langle A', B'\rangle \in S, k \in \mathbb{N}\}$ y se muestra que $\mathcal{R}(\mathcal{S})$ es $\Phi_{\simeq_{\mathfrak{T}}}$-denso.

Sea $\mathcal{S}_c \triangleq \{\langle A', B'\rangle \in \mathcal{S} \mid c = \#_\mu(A') + \#_\mu(B')\}$. Luego, dado $k_0 \in \mathbb{N}$, se tiene $\langle\lfloor[\![A']\!]^{\mathfrak{T}}\rfloor_{k_0}, \lfloor[\![B']\!]^{\mathfrak{T}}\rfloor_{k_0}\rangle \in \mathcal{R}(\mathcal{S}) \implies \langle A', B'\rangle \in \mathcal{S} \implies \exists c \in \mathbb{N}.\langle A', B'\rangle \in \mathcal{S}_c$. Basta ver entonces que para cada $c \in \mathbb{N}$ vale

$$\langle A', B'\rangle \in \mathcal{S}_c \implies \forall k > 0.\langle\lfloor[\![A']\!]^{\mathfrak{T}}\rfloor_k, \lfloor[\![B']\!]^{\mathfrak{T}}\rfloor_k\rangle \in \Phi_{\simeq_{\mathfrak{T}}}(\mathcal{R}(\mathcal{S}))$$

Notar que el caso faltante ($k = 0$) es inmediato por (E-REFL-T). La demostración es por inducción en c.

- $c = 0$. Como $\langle A', B'\rangle \in \mathcal{S}_0$ implica $\langle A', B'\rangle \in \mathcal{S} \subseteq \Phi_{\simeq_{\vec{\mu}}}(\mathcal{S})$, se tienen cuatro casos posibles:

  1. $\langle A', B'\rangle = \langle a, a\rangle$. Luego, $\langle\lfloor[\![A']\!]^{\mathfrak{T}}\rfloor_k, \lfloor[\![B']\!]^{\mathfrak{T}}\rfloor_k\rangle = \langle a, a\rangle$ para todo $k > 0$ y se concluye por (E-REFL-T).
  2. $\langle A', B'\rangle = \langle D \mathbin{@} A'', D' \mathbin{@} B''\rangle$ con $\langle D, D'\rangle, \langle A'', B''\rangle \in \mathcal{S}$. Por definición de $\mathcal{R}$, se tienen $\langle\lfloor[\![D]\!]^{\mathfrak{T}}\rfloor_{k-1}, \lfloor[\![D']\!]^{\mathfrak{T}}\rfloor_{k-1}\rangle, \langle\lfloor[\![A'']\!]^{\mathfrak{T}}\rfloor_{k-1}, \lfloor[\![B'']\!]^{\mathfrak{T}}\rfloor_{k-1}\rangle \in \mathcal{R}(\mathcal{S})$ para todo $k > 0$. Finalmente, por (E-COMP-T), Def. 3.3.10 y 3.3.22

     $$\langle\lfloor[\![D \mathbin{@} A'']\!]^{\mathfrak{T}}\rfloor_k, \lfloor[\![D' \mathbin{@} B'']\!]^{\mathfrak{T}}\rfloor_k\rangle \in \Phi_{\simeq_{\mathfrak{T}}}(\mathcal{R}(\mathcal{S}))$$

     para todo $k > 0$.

3. $\langle A', B'\rangle = \langle A'_0 \supset A'_1, B'_0 \mathbin{@} B'_1\rangle$ con $\langle A'_0, B'_0\rangle, \langle A'_1, B'_1\rangle \in \mathcal{S}$. Por definición de $\mathcal{R}$, se tienen $\langle \llbracket A'_0 \rrbracket^{\mathfrak{T}}\rfloor_{k-1}, \llbracket B'_0 \rrbracket^{\mathfrak{T}}\rfloor_{k-1}\rangle, \langle \llbracket A'_1 \rrbracket^{\mathfrak{T}}\rfloor_{k-1}, \llbracket B'_1 \rrbracket^{\mathfrak{T}}\rfloor_{k-1}\rangle \in \mathcal{R}(\mathcal{S})$ para todo $k > 0$. Finalmente, se concluye por (E-FUNC-T), Def. 3.3.10 y 3.3.22

$$\langle \llbracket A'_0 \supset A'_1 \rrbracket^{\mathfrak{T}}\rfloor_k, \llbracket A'_1 \mathbin{@} B'_1 \rrbracket^{\mathfrak{T}}\rfloor_k\rangle \in \Phi_{\simeq_{\mathfrak{T}}}(\mathcal{R}(\mathcal{S}))$$

para todo $k > 0$.

4. $\langle A', B'\rangle = \langle \bigoplus_{i<n} A'_i, \bigoplus_{j<m} B'_j\rangle$ con $n + m > 2$, $(A'_i \neq \mu, \oplus)_{i<n}$, $(B'_j \neq \mu, \oplus)_{j<m}$ y funciones $f : [0, n) \to [0, m)$, $g : [0, m) \to [0, n)$ tal que $(\langle A'_i, B'_{f(i)}\rangle)_{i<n}, (\langle A'_{g(j)}, B'_j\rangle)_{j<m} \in \mathcal{S}$. Por definición de $\mathcal{R}$ y Def. 3.3.22, se tienen $(\langle \llbracket A'_i \rrbracket^{\mathfrak{T}}\rfloor_k, \llbracket B'_{f(i)} \rrbracket^{\mathfrak{T}}\rfloor_k\rangle)_{i<n}, (\langle \llbracket A'_{g(j)} \rrbracket^{\mathfrak{T}}\rfloor_k, \llbracket B'_j \rrbracket^{\mathfrak{T}}\rfloor_k\rangle)_{j<m} \in \mathcal{R}(\mathcal{S})$ para todo $k > 0$. Más aún, como $(A'_i \neq \mu, \oplus)_{i<n}$ y $(B'_j \neq \mu, \oplus)_{j<m}$, se tienen tanto $(\llbracket A'_i \rrbracket^{\mathfrak{T}}\rfloor_k \neq \oplus)_{i<n}$ como $(\llbracket B'_j \rrbracket^{\mathfrak{T}}\rfloor_k \neq \oplus)_{j<m}$. Entonces, puede aplicarse (E-UNION-T) con las funciones f y g para concluir por Def. 3.3.10 y 3.3.22

$$\langle \llbracket \bigoplus_{i<n} A'_i \rrbracket^{\mathfrak{T}}\rfloor_k, \llbracket \bigoplus_{j<m} B'_j \rrbracket^{\mathfrak{T}}\rfloor_k\rangle \in \Phi_{\simeq_{\mathfrak{T}}}(\mathcal{R}(\mathcal{S}))$$

para todo $k > 0$.

- $c > 0$. Luego, hay dos casos posibles:

1. $\langle A', B'\rangle = \langle \mathscr{C}\langle \mu V.A''\rangle, B'\rangle$ con $\langle \mathscr{C}\langle \{V \setminus A'\} A''\rangle, B'\rangle \in \mathcal{S}$. Por contractividad, se garantiza $\#_\mu(\mathscr{C}\langle \{V \setminus A'\} A''\rangle) < \#_\mu(\mathscr{C}\langle \mu V.A''\rangle)$, lo que implica $\langle \mathscr{C}\langle \{V \setminus A'\} A''\rangle, B'\rangle \in \mathcal{S}_{c'}$ con $c' < c$. Por *h.i.* se tiene
$$\langle \llbracket \mathscr{C}\langle \{V \setminus A'\} A''\rangle \rrbracket^{\mathfrak{T}}\rfloor_k, \llbracket B' \rrbracket^{\mathfrak{T}}\rfloor_k\rangle \in \Phi_{\simeq_{\mathfrak{T}}}(\mathcal{R}(\mathcal{S}))$$
para todo $k > 0$. Se concluye por Def. 3.3.22, dado que $\llbracket \mathscr{C}\langle \mu V.A''\rangle \rrbracket^{\mathfrak{T}} = \llbracket \mathscr{C}\langle \{V \setminus A'\} A''\rangle \rrbracket^{\mathfrak{T}}$.
2. $\langle A', B'\rangle = \langle A', \mathscr{D}\langle \mu W.B''\rangle\rangle$ con $\langle A', \mathscr{D}\langle \{W \setminus B'\} B''\rangle\rangle \in \mathcal{S}$. Como en el caso anterior, por *h.i.* apelando a la contractividad y se concluye por Def. 3.3.22, $\langle \llbracket A' \rrbracket^{\mathfrak{T}}\rfloor_k, \llbracket \mathscr{D}\langle \mu W.B''\rangle \rrbracket^{\mathfrak{T}}\rfloor_k\rangle \in \Phi_{\simeq_{\mathfrak{T}}}(\mathcal{R}(\mathcal{S}))$ para todo $k > 0$.

$\Leftarrow$) Se muestra que $\mathcal{R} \triangleq \{\langle A', B'\rangle \mid A', B' \in \mathcal{T}, \forall k \in \mathbb{N}. \llbracket A' \rrbracket^{\mathfrak{T}}\rfloor_k \simeq_{\mathfrak{T}} \llbracket B' \rrbracket^{\mathfrak{T}}\rfloor_k\}$ es $\Phi_{\simeq_{\vec{\mu}}}$-denso. Sea $\langle A, B\rangle \in \mathcal{R}$. La demostración es por análisis de casos en $c = \#_\mu(A) + \#_\mu(B)$. Se consideran únicamente los casos $k > 0$ para evitar el caso borde trivial $\langle \llbracket A \rrbracket^{\mathfrak{T}}\rfloor_0, \llbracket B \rrbracket^{\mathfrak{T}}\rfloor_0\rangle = \langle \varepsilon, \varepsilon\rangle$.

- $c = 0$. Como $\langle \llbracket A \rrbracket^{\mathfrak{T}}\rfloor_k, \llbracket B \rrbracket^{\mathfrak{T}}\rfloor_k\rangle \in \simeq_{\mathfrak{T}}$ y $\simeq_{\mathfrak{T}} = \Phi_{\simeq_{\mathfrak{T}}}(\simeq_{\mathfrak{T}})$, se tienen cuatro casos posibles:

1. $\langle \llbracket A \rrbracket^{\mathfrak{T}}\rfloor_k, \llbracket B \rrbracket^{\mathfrak{T}}\rfloor_k\rangle = \langle a, a\rangle$. Por Def. 3.3.10 y 3.3.22 se tiene necesariamente $A = a = B$, pues $c = 0$. Luego, por (E-REFL-AL), $\langle A, B\rangle \in \Phi_{\simeq_{\vec{\mu}}}(\mathcal{R})$.
2. $\langle \llbracket A \rrbracket^{\mathfrak{T}}\rfloor_k, \llbracket B \rrbracket^{\mathfrak{T}}\rfloor_k\rangle = \langle \mathcal{D}\rfloor_{k-1} \mathbin{@} \mathcal{A}'\rfloor_{k-1}, \mathcal{D}'\rfloor_{k-1} \mathbin{@} \mathcal{B}'\rfloor_{k-1}\rangle$ con $\mathcal{D}\rfloor_{k-1} \simeq_{\mathfrak{T}} \mathcal{D}'\rfloor_{k-1}$ y $\mathcal{A}'\rfloor_{k-1} \simeq_{\mathfrak{T}} \mathcal{B}'\rfloor_{k-1}$. Por Def. 3.3.22 y $c = 0$, existen $D, D' \in \mathcal{D}$ y $A', B' \in \mathcal{T}$ tal que $A = D \mathbin{@} A'$ y $B = D' \mathbin{@} B'$ con $\llbracket D \rrbracket^{\mathfrak{T}} = \mathcal{D}$, $\llbracket D \rrbracket^{\mathfrak{T}} = \mathcal{D}'$, $\llbracket A' \rrbracket^{\mathfrak{T}} = \mathcal{A}'$ y $\llbracket B' \rrbracket^{\mathfrak{T}} = \mathcal{B}'$. Más aún, por definición de $\mathcal{R}$, $\langle D, D'\rangle, \langle A', B'\rangle \in \mathcal{R}$. Luego, se concluye por (E-COMP-AL), $\langle A, B\rangle \in \Phi_{\simeq_{\vec{\mu}}}(\mathcal{R})$.
3. $\langle \llbracket A \rrbracket^{\mathfrak{T}}\rfloor_k, \llbracket B \rrbracket^{\mathfrak{T}}\rfloor_k\rangle = \langle \mathcal{A}'\rfloor_{k-1} \supset \mathcal{A}''\rfloor_{k-1}, \mathcal{B}'\rfloor_{k-1} \supset \mathcal{B}''\rfloor_{k-1}\rangle$ con las equivalencias $\mathcal{A}'\rfloor_{k-1} \simeq_{\mathfrak{T}} \mathcal{B}'\rfloor_{k-1}$ y $\mathcal{A}''\rfloor_{k-1} \simeq_{\mathfrak{T}} \mathcal{B}''\rfloor_{k-1}$. Por Def. 3.3.22 y $c = 0$, existen $A', A'', B', B'' \in \mathcal{T}$ tal que $A = A' \supset A''$ y $B = B' \supset B''$ con $\llbracket A' \rrbracket^{\mathfrak{T}} = \mathcal{A}'$, $\llbracket A'' \rrbracket^{\mathfrak{T}} = \mathcal{A}''$, $\llbracket B' \rrbracket^{\mathfrak{T}} = \mathcal{B}'$ y $\llbracket B'' \rrbracket^{\mathfrak{T}} = \mathcal{B}''$. Más aún, por definición de $\mathcal{R}$, $\langle A', B'\rangle, \langle A'', B''\rangle \in \mathcal{R}$. Luego, se concluye por (E-FUNC-AL), $\langle A, B\rangle \in \Phi_{\simeq_{\vec{\mu}}}(\mathcal{R})$.
4. $\langle \llbracket A \rrbracket^{\mathfrak{T}}\rfloor_k, \llbracket B \rrbracket^{\mathfrak{T}}\rfloor_k\rangle = \langle \bigoplus_{i<n} \mathcal{A}_i\rfloor_k, \bigoplus_{j<m} \mathcal{B}_j\rfloor_k\rangle$ con $n+m > 2$, $(\mathcal{A}_i\rfloor_k)_{i<n}$, $(\mathcal{B}_j\rfloor_k)_{j<m}$ y funciones $f : [0, n) \to [0, m)$, $g : [0, m) \to [0, n)$ tal que valen las equivalencias $(\mathcal{A}_i\rfloor_k \simeq_{\mathfrak{T}} \mathcal{B}_{f(i)}\rfloor_k)_{i<n}$ y $(\mathcal{A}_{g(j)}\rfloor_k \simeq_{\mathfrak{T}} \mathcal{B}_j\rfloor_k)_{j<m}$. Por Lem. 3.6.7, existen $n' \leq n$, $\oplus$-contextos maximales $\mathcal{A}$ y

$(\mathcal{A}_l)_{l<n'}$, μ-tipos contractivos $(A_l \neq \oplus)_{l<n'}$ y funciones $b, e : [0, n') \to [0, n)$ tal que $A = \mathcal{A}\langle\vec{A}\rangle$ y $([\![A_l]\!]^{\mathfrak{T}} = \mathcal{A}_l\langle\mathcal{A}_{b(l)}, \ldots, \mathcal{A}_{e(l)}\rangle)_{l<n'}$. Más aún, como $\#_\mu(A) = 0$ y $\mathcal{A}$ es maximal, es necesariamente el caso $n' = n$ y $(\mathcal{A}_l = \Box)_{l<n'}$. Por lo tanto, $b = e = id$, $([\![A_i]\!]^{\mathfrak{T}} = \mathcal{A}_i)_{i<n}$ y $(A_i \neq \mu, \oplus)_{i<n}$. Análogamente, se tiene $B = \mathcal{B}\langle\vec{B}\rangle$ donde $\mathcal{B}$ es un $\oplus$-contexto maximal, $([\![B_j]\!]^{\mathfrak{T}} = \mathcal{B}_j)_{j<m}$ y $(B_j \neq \mu, \oplus)_{j<m}$. Luego, se tienen $([\![A_i]\!]^{\mathfrak{T}}\rfloor_k \simeq_{\mathfrak{T}} [\![B_{f(i)}]\!]^{\mathfrak{T}}\rfloor_k)_{i<n}$ y $([\![A_{g(j)}]\!]^{\mathfrak{T}}\rfloor_k \simeq_{\mathfrak{T}} [\![B_j]\!]^{\mathfrak{T}}\rfloor_k)_{j<m}$. Finalmente, se concluye por (E-UNION-AL), $\langle A, B\rangle \in \Phi_{\simeq_{\vec{\mu}}}(\mathcal{R})$.

- $c > 0$. Luego, se distinguen dos casos:

  1. $\#_\mu(A) = 0$. Luego, $\#_\mu(B) > 0$. Por lo tanto, $B = \mathscr{D}\langle\mu W.B'\rangle$ y, por Def. 3.3.22, se tiene $[\![B]\!]^{\mathfrak{T}} = [\![\mathscr{D}\langle\{W \setminus B\}B'\rangle]\!]^{\mathfrak{T}}$. Luego, $\langle A, \mathscr{D}\langle\{W \setminus B\}B'\rangle\rangle \in \mathcal{R}$. Notar que $\#_\mu(A) = 0$ implica $A \neq \mathscr{C}\langle\mu V.A'\rangle$. Se concluye por (E-REC-R-AL), $\langle A, B\rangle \in \Phi_{\simeq_{\vec{\mu}}}(\mathcal{R})$.
  2. $\#_\mu(A) > 0$. Luego, $A = \mathscr{C}\langle\mu V.A'\rangle$ y, por Def. 3.3.22, $[\![A]\!]^{\mathfrak{T}} = [\![\mathscr{C}\langle\{V \setminus A\}A'\rangle]\!]^{\mathfrak{T}}$. Luego, se tiene $\langle\mathscr{C}\langle\{V \setminus A\}A'\rangle, B\rangle \in \mathcal{R}$ y se concluye por (E-REC-L-AL), $\langle A, B\rangle \in \Phi_{\simeq_{\vec{\mu}}}(\mathcal{R})$.

□

Chequeo de sub-tipado

Lema 3.6.11. *Sean $\langle A, B\rangle \in \mathcal{S}$ con $\mathcal{S} \subseteq \Phi_{\preceq_{\vec{\mu}}}(\mathcal{S})$, $(A_i)_{i<n}, (B_j)_{j<m}$ μ-tipos contractivos y $\mathcal{A}, \mathcal{B}$ $\mu\oplus$-contextos maximales tal que $A = \mathcal{A}\langle\vec{A}\rangle$ y $B = \mathcal{B}\langle\vec{B}\rangle$. Luego, existe una función $f : [0, n) \to [0, m)$ tal que $(\langle A{\downarrow_{\mathsf{p}_i}}, B{\downarrow_{\mathsf{q}_{f(i)}}}\rangle \in \mathcal{S})_{i<n}$, donde $\mathsf{p}_0, \ldots, \mathsf{p}_{n-1} \in \mathsf{ppos}(A)$ y $\mathsf{q}_0, \ldots, \mathsf{q}_{m-1} \in \mathsf{ppos}(B)$ son las posiciones de $\Box$ en $\mathcal{A}$ y $\mathcal{B}$ respectivamente.*

Demostración. Por inducción en $c = \#_{\mu\oplus}(A) + \#_{\mu\oplus}(B)$ (*i.e.* la suma del tamaño de los $\mu\oplus$-contextos maximales $\mathcal{A}$ y $\mathcal{B}$).

- $c = 0$. Luego, $\mathcal{A} = \Box = \mathcal{B}$. El resultado es inmediato pues $\mathsf{ppos}(A) = \{\epsilon\} = \mathsf{ppos}(B)$ y $\langle A{\downarrow_\epsilon}, B{\downarrow_\epsilon}\rangle = \langle A, B\rangle \in \mathcal{S}$ por hipótesis.
- $c > 0$. Luego, se analiza la forma de $\mathcal{A}$.
  - $\mathcal{A} = \Box$. Luego, $\mathsf{ppos}(A) = \{\epsilon\}$ y $A{\downarrow_\epsilon} = A$ por Def. 3.6.4 (*i.e.* $n = 1$). Más aún, $A \neq \mu, \oplus$ por hipótesis. Se tienen entonces dos posibles casos para $\mathcal{B}$.
    1. $\mathcal{B} = \mu W.\mathcal{B}'$. Luego, $\langle A, B\rangle \in \mathcal{S} \subseteq \Phi_{\preceq_{\vec{\mu}}}(\mathcal{S})$ es necesariamente derivado por (S-REC-R-AL), *i.e.* $\langle A, \{W \setminus B\}\mathcal{B}'\langle\vec{B}\rangle\rangle \in \mathcal{S}$. Más aún, por contractividad $(B_j \neq W)_{j<m}$. Entonces, $\mathcal{B}'$ es un $\mu\oplus$-contexto maximal para $\mathcal{B}'\langle\{W \setminus B\}\vec{B}\rangle = \{W \setminus B\}\mathcal{B}'\langle\vec{B}\rangle$. Por *h.i.* existe una función $f : [0, 1) \to [0, m)$ tal que $\langle A{\downarrow_\epsilon}, \{W \setminus B\}\mathcal{B}'\langle\vec{B}\rangle{\downarrow_{\mathsf{q}'_{f(0)}}}\rangle \in \mathcal{S}$, donde $\mathsf{q}'_1, \ldots, \mathsf{q}'_m \in \mathsf{ppos}(\mathcal{B}'\langle\vec{B}\rangle)$ son las posiciones de $\Box$ en $\mathcal{B}'$. Notar que $(\mathsf{q}_j = 0\mathsf{q}'_j \in \mathsf{ppos}(B))_{j<m}$. Luego, por Def. 3.6.4, $\{W \setminus B\}\mathcal{B}'\langle\vec{B}\rangle{\downarrow_{\mathsf{q}'_{f(0)}}} = (\mu W.\mathcal{B}'\langle\vec{B}\rangle){\downarrow_{0\mathsf{q}'_{f(0)}}} = B{\downarrow_{\mathsf{q}_{f(0)}}}$, por lo que se concluye $\langle A{\downarrow_\epsilon}, B{\downarrow_{\mathsf{q}_{f(0)}}}\rangle \in \mathcal{S}$.
    2. $\mathcal{B} = \oplus_{l<m'}\mathcal{B}_l$ con $(\mathcal{B}_l \neq \oplus)_{l<m'}$ y $1 < m' \leq m$. Puede particionarse $[0, m)$ con funciones $b, e : [0, m') \to [0, m)$ tal que $[0, m) = +_{l<m'}[b(l), e(l)]$ y $(B'_l = \mathcal{B}_l\langle B_{b(l)}, \ldots, B_{e(l)}\rangle)_{l<m'}$. Más aún, de $\langle A, B\rangle \in \mathcal{S} \subseteq \Phi_{\preceq_{\vec{\mu}}}(\mathcal{S})$ se tiene $\langle A, B'_k\rangle \in \mathcal{S}$ para algún $k < m'$ por (S-UNION-R-AL). Por *h.i.* existe $f : [0, 1) \to [b(k), e(k)]$ tal que $\langle A{\downarrow_\epsilon}, B'_k{\downarrow_{\mathsf{q}''_{f(0)}}}\rangle \in \mathcal{S}$, donde $\mathsf{q}''_{b(k)}, \ldots, \mathsf{q}''_{e(k)} \in \mathsf{ppos}(B'_k)$ son las posiciones de $\Box$ en $\mathcal{B}_k$. Más aún, sea q'_k la posición de $\mathcal{B}_k$ en $\mathcal{B}$, se tiene $(\mathsf{q}_j = \mathsf{q}'_k\mathsf{q}''_j \in \mathsf{ppos}(B))_{j\in[b(k),e(k)]}$. Con una simple inducción en q'_k, utilizando la Def. 3.6.4, se verifica $B'_k{\downarrow_{\mathsf{q}''_{f(0)}}} = B{\downarrow_{\mathsf{q}_{f(0)}}}$. Luego, se concluye $\langle A{\downarrow_\epsilon}, B{\downarrow_{\mathsf{q}_{f(0)}}}\rangle \in \mathcal{S}$.

- $\mathcal{A} = \mu V.\mathcal{A}'$. Luego, por (S-REC-L-AL), $\langle A, B\rangle \in \mathcal{S} \subseteq \Phi_{\preceq_{\vec{\mu}}}(\mathcal{S})$ se deriva a partir del par $\langle \{V \setminus A\}\mathcal{A}'\langle\vec{A}\rangle, B\rangle \in \mathcal{S}$. Por contractividad $(A_i \neq V)_{i<n}$. Entonces, $\mathcal{A}'$ es un $\mu\oplus$-contexto maximal para $\mathcal{A}'\langle\{V \setminus A\}\vec{A}\rangle = \{V \setminus A\}\mathcal{A}'\langle\vec{A}\rangle$. Por *h.i.* existe una función $f : [0, n) \to [0, m)$ tal que $(\langle \mathcal{A}'\langle\{V \setminus A\}\vec{A}\rangle{\downarrow_{\mathsf{p}'_i}}, B{\downarrow_{\mathsf{q}_{f(i)}}}\rangle \in \mathcal{S})_{i<n}$, donde $\mathsf{p}'_1, \ldots, \mathsf{p}'_n \in \mathsf{ppos}(\mathcal{A}'\langle\vec{A}\rangle)$ y $\mathsf{q}_1, \ldots, \mathsf{q}_m \in \mathsf{ppos}(B)$ son las posiciones de $\Box$ en $\mathcal{A}'$ y $\mathcal{B}$ respectivamente. Notar que $(\mathsf{p}_i = \mathsf{0}\mathsf{p}'_i \in \mathsf{ppos}(A))_{i<n}$. Luego, por Def. 3.6.4, $(\{V \setminus A\}\mathcal{A}'\langle\vec{A}\rangle{\downarrow_{\mathsf{p}'_i}} = (\mu V.\mathcal{A}'\langle\vec{A}\rangle){\downarrow_{\mathsf{0}\mathsf{p}'_i}} = A{\downarrow_{\mathsf{p}_i}})_{i<n}$, por lo que se concluye $(\langle A{\downarrow_{\mathsf{p}_i}}, B{\downarrow_{\mathsf{q}_{f(i)}}}\rangle \in \mathcal{S})_{i<n}$.
- $\mathcal{A} = \oplus_{l<n'}\mathcal{A}_l$ con $(\mathcal{A}_l \neq \oplus)_{l<n'}$ y $1 < n' \leq n$. Luego, hay dos posibles casos para B.
  1. $B = \mu$ (*i.e.* $\mathcal{B} = \mu W.\mathcal{B}'\langle\vec{B}\rangle$). Notar que, en particular, $A \neq \mu$. Luego, de $\langle A, B\rangle \in \mathcal{S} \subseteq \Phi_{\preceq_{\vec{\mu}}}(\mathcal{S})$ se tiene $\langle A, \{W \setminus B\}\mathcal{B}'\langle\vec{B}\rangle\rangle \in \mathcal{S}$ por (S-REC-R-AL). Más aún, por contractividad se tiene $(B_j \neq W)_{j<m}$. entonces, $\mathcal{B}'$ es un $\mu\oplus$-contexto maximal para $\mathcal{B}'\langle\{W \setminus B\}\vec{B}\rangle = \{W \setminus B\}\mathcal{B}'\langle\vec{B}\rangle$. Luego, por *h.i.* existe una función $f : [0, n) \to [0, m)$ tal que se tienen $(\langle A{\downarrow_{\mathsf{p}_i}}, \mathcal{B}'\langle\{W \setminus B\}\vec{B}\rangle{\downarrow_{\mathsf{q}'_{f(i)}}}\rangle \in \mathcal{S})_{i<n}$, donde $\mathsf{p}_1, \ldots, \mathsf{p}_n \in \mathsf{ppos}(A)$ y $\mathsf{q}'_1, \ldots, \mathsf{q}'_m \in \mathsf{ppos}(\mathcal{B}'\langle\vec{B}\rangle)$ son las posiciones de $\Box$ en $\mathcal{A}$ y $\mathcal{B}'$ respectivamente. Notar que, en particular, $(\mathsf{q}_j = \mathsf{0}\mathsf{q}'_j \in \mathsf{ppos}(B))_{j<m}$. Luego, por Def. 3.6.4, se tienen las proyecciones $(\{W \setminus B\}\mathcal{B}'\langle\vec{B}\rangle{\downarrow_{\mathsf{q}'_j}} = (\mu W.\mathcal{B}'\langle\vec{B}\rangle){\downarrow_{\mathsf{0}\mathsf{q}'_j}} = B{\downarrow_{\mathsf{q}_j}})_{j<m}$, por lo que finalmente se concluye $(\langle A{\downarrow_{\mathsf{p}_i}}, B{\downarrow_{\mathsf{q}_{f(i)}}}\rangle \in \mathcal{S})_{i<n}$.
  2. $B \neq \mu$. Notar que puede particionarse $[0, n)$ mediante funciones $b, e : [0, n') \to [0, n)$ tal que $[0, n) = +_{l<n'}[b(l), e(l)]$ y $(A'_l = \mathcal{A}_l\langle A_{b(l)}, \ldots, A_{e(l)}\rangle)_{l<n'}$. Más aún, de $\langle A, B\rangle \in \mathcal{S} \subseteq \Phi_{\preceq_{\vec{\mu}}}(\mathcal{S})$ se tiene $(\langle A'_l, B\rangle \in \mathcal{S})_{l<n'}$ por (S-UNION-L-AL). Por *h.i.* existen funciones $(f_l : [b(l), e(l)] \to [0, m))_{l<n'}$ tal que se tienen $(\langle A'_l{\downarrow_{\mathsf{p}''_i}}, B{\downarrow_{\mathsf{q}_{f_l(i)}}}\rangle \in \mathcal{S})_{i\in[b(l),e(l)]}$, donde $\mathsf{p}''_{b(l)}, \ldots, \mathsf{p}''_{e(l)} \in \mathsf{ppos}(A'_l)$ son las posiciones de $\Box$ en $\mathcal{A}_l$ para cada $l < n'$. Más aún, sea p'_l la posición de $\mathcal{A}_l$ en $\mathcal{A}$, se tiene $(\mathsf{p}_i = \mathsf{p}'_l\mathsf{p}''_i \in \mathsf{ppos}(A))_{i\in[b(l),e(l)]}$ para cada $l < n'$. Con una simple inducción en p'_l, utilizando la Def. 3.6.4, se verifica $(A'_l{\downarrow_{\mathsf{p}''_i}} = A{\downarrow_{\mathsf{p}_i}})_{i\in[b(l),e(l)]}$ para cada $l < n'$. Más aún, la función $f = f_1 \circ \ldots \circ f_{n'}$ resulta bien definida dado que $+_{l<n'}[b(l), e(l)]$ constituye una partición de $[0, n)$. Finalmente, se concluye con f, $(\langle A{\downarrow_{\mathsf{p}_i}}, B{\downarrow_{\mathsf{q}_{f(i)}}}\rangle \in \mathcal{S})_{i<n}$.

□

Lema 3.6.12. *$A \preceq_{\vec{\mu}} B$ sii $\forall k \in \mathbb{N}.\lfloor[\![A]\!]^{\mathfrak{T}}\rfloor_k \preceq_{\mathfrak{T}} \lfloor[\![B]\!]^{\mathfrak{T}}\rfloor_k$.*

Demostración. Para esta prueba se denota con $\#_\mu(A)$ el número de ocurrencias del constructor μ a la cabeza A: *i.e.* $\#_\mu(A) \triangleq 0$ si $A \neq \mu$; $\#_\mu(\mu V.A') \triangleq 1 + \#_\mu(A')$.

$\Rightarrow$) Dado $A \preceq_{\vec{\mu}} B$, existe $\mathcal{S} \in \wp(\mathcal{T} \times \mathcal{T})$ tal que $\mathcal{S} \subseteq \Phi_{\preceq_{\vec{\mu}}}(\mathcal{S})$ y $\langle A, B\rangle \in \mathcal{S}$. Se define entonces $\mathcal{R}(\mathcal{S}) \triangleq \{\langle\lfloor[\![A']\!]^{\mathfrak{T}}\rfloor_k, \lfloor[\![B']\!]^{\mathfrak{T}}\rfloor_k\rangle \mid \langle A', B'\rangle \in S, k \in \mathbb{N}\}$ y se muestra que $\mathcal{R}(\mathcal{S})$ es $\Phi_{\preceq_{\mathfrak{T}}}$-denso.

Sea $\mathcal{S}_c \triangleq \{\langle A', B'\rangle \in \mathcal{S} \mid c = \#_\mu(A') + \#_\mu(B')\}$. Luego, dado $k_0 \in \mathbb{N}$, se tiene $\langle\lfloor[\![A']\!]^{\mathfrak{T}}\rfloor_{k_0}, \lfloor[\![B']\!]^{\mathfrak{T}}\rfloor_{k_0}\rangle \in \mathcal{R}(\mathcal{S}) \implies \langle A', B'\rangle \in \mathcal{S} \implies \exists c \in \mathbb{N}.\langle A', B'\rangle \in \mathcal{S}_c$. Basta ver entonces que para cada $c \in \mathbb{N}$ vale

$$\langle A', B'\rangle \in \mathcal{S}_c \implies \forall k > 0.\langle\lfloor[\![A']\!]^{\mathfrak{T}}\rfloor_k, \lfloor[\![B']\!]^{\mathfrak{T}}\rfloor_k\rangle \in \Phi_{\preceq_{\mathfrak{T}}}(\mathcal{R}(\mathcal{S}))$$

Notar que el caso faltante ($k = 0$) es inmediato por (S-REFL-T). La demostración es por inducción en c.

- $c = 0$. Como $\langle A', B'\rangle \in \mathcal{S}_0$ implica $\langle A', B'\rangle \in \mathcal{S} \subseteq \Phi_{\preceq_{\vec{\mu}}}(\mathcal{S})$, se tienen cinco casos posibles:
  1. $\langle A', B'\rangle = \langle a, a\rangle$. Luego, $\langle\lfloor[\![A']\!]^{\mathfrak{T}}\rfloor_k, \lfloor[\![B']\!]^{\mathfrak{T}}\rfloor_k\rangle = \langle a, a\rangle$ para todo $k > 0$ y se concluye por (S-REFL-T).

2. $\langle A', B'\rangle = \langle D \mathbin{@} A'', D' \mathbin{@} B''\rangle$ con $\langle D, D'\rangle, \langle A'', B''\rangle \in \mathcal{S}$. Por definición de $\mathcal{R}$, se tienen $\langle [\![D]\!]^{\mathfrak{T}}\rfloor_{k-1}, [\![D']\!]^{\mathfrak{T}}\rfloor_{k-1}\rangle, \langle [\![A'']\!]^{\mathfrak{T}}\rfloor_{k-1}, [\![B'']\!]^{\mathfrak{T}}\rfloor_{k-1}\rangle \in \mathcal{R}(\mathcal{S})$ para todo $k > 0$. Finalmente, por (S-COMP-T), Def. 3.3.10 y 3.3.22

$$\langle [\![D \mathbin{@} A'']\!]^{\mathfrak{T}}\rfloor_k, [\![D' \mathbin{@} B'']\!]^{\mathfrak{T}}\rfloor_k\rangle \in \Phi_{\preceq_{\mathfrak{T}}}(\mathcal{R}(\mathcal{S}))$$

para todo $k > 0$.

3. $\langle A', B'\rangle = \langle A'_0 \supset A'_1, B'_0 \mathbin{@} B'_1\rangle$ con $\langle B'_0, A'_0\rangle, \langle A'_1, B'_1\rangle \in \mathcal{S}$. Por definición de $\mathcal{R}$, se tienen $\langle [\![B'_0]\!]^{\mathfrak{T}}\rfloor_{k-1}, [\![A'_0]\!]^{\mathfrak{T}}\rfloor_{k-1}\rangle, \langle [\![A'_1]\!]^{\mathfrak{T}}\rfloor_{k-1}, [\![B'_1]\!]^{\mathfrak{T}}\rfloor_{k-1}\rangle \in \mathcal{R}(\mathcal{S})$ para todo $k > 0$. Finalmente, se concluye por (S-FUNC-T), Def. 3.3.10 y 3.3.22

$$\langle [\![A'_0 \supset A'_1]\!]^{\mathfrak{T}}\rfloor_k, [\![A'_1 \mathbin{@} B'_1]\!]^{\mathfrak{T}}\rfloor_k\rangle \in \Phi_{\preceq_{\mathfrak{T}}}(\mathcal{R}(\mathcal{S}))$$

para todo $k > 0$.

4. $\langle A', B'\rangle = \langle \bigoplus_{i<n} A'_i, B'\rangle$ con $n > 1$, $B' \neq \mu$, $(A'_i \neq \oplus)_{i<n}$ y $(\langle A'_i, B'\rangle \in \mathcal{S})_{i<n}$. Alternativamente, puede denotarse $A = \mathcal{A}'\langle \vec{A'}\rangle$ con $\mathcal{A}'$ un $\oplus$-contexto maximal. Más aún, sea $\mathcal{A}''$ un $\mu\oplus$-contexto maximal tal que $A' = \mathcal{A}''\langle \vec{A''}\rangle$ con $(A''_l \neq \mu, \oplus)_{l<n'}$ para algún $n' \geq n$, luego existen $\mu\oplus$-contextos $(\mathcal{A}'_i)_{i<n}$ tal que $\mathcal{A}'' = \mathcal{A}'\langle \vec{\mathcal{A}'}\rangle$. Del mismo modo, sea $\mathcal{B}'$ un $\mu\oplus$-contexto maximal tal que $B' = \mathcal{B}'\langle \vec{B'}\rangle$ con $(B'_j \neq \mu, \oplus)_{j<m}$ para algún $m > 0$. Notar que puede particionarse $[0, n')$ mediante funciones $b, e : [0, n) \to [0, n')$ tal que $[0, n') = +_{i<n}[b(i), e(i)]$ y $(A'_i = \mathcal{A}'_i\langle A''_{b(i)}, \ldots, A''_{e(i)}\rangle)_{i<n}$. Por Lem. 3.6.11 con $\langle A', B'\rangle \in \mathcal{S}$, existe una función $f : [0, n') \to [0, m)$ tal que $(\langle A'{\downarrow_{\mathsf{p}_l}}, B'{\downarrow_{\mathsf{q}_{f(l)}}}\rangle \in \mathcal{S})_{l<n'}$, donde $\mathsf{p}_1, \ldots, \mathsf{p}_{n'} \in \mathsf{ppos}(A')$ y $\mathsf{q}_1, \ldots, \mathsf{q}_m \in \mathsf{ppos}(B')$ son las posiciones de $\Box$ en $\mathcal{A}''$ y $\mathcal{B}'$ respectivamente. Luego, $(\langle [\![A'{\downarrow_{\mathsf{p}_l}}]\!]^{\mathfrak{T}}\rfloor_k, [\![B'{\downarrow_{\mathsf{q}_{f(l)}}}]\!]^{\mathfrak{T}}\rfloor_k\rangle \in \mathcal{R}(\mathcal{S}))_{l<n'}$ para todo $k > 0$, por definición de $\mathcal{R}$. Dado que $\mathcal{A}''$ es un $\mu\oplus$-contexto maximal, por contractividad y Def. 3.6.4 se tiene $(A'{\downarrow_{\mathsf{p}_l}} \neq \mu, \oplus)_{l<n'}$. Más aún, por Def. 3.3.10 y 3.3.22, $([\![A'{\downarrow_{\mathsf{p}_l}}]\!]^{\mathfrak{T}}\rfloor_k \neq \oplus)_{l<n'}$ para todo $k > 0$. Del mismo modo, $([\![B'{\downarrow_{\mathsf{q}_j}}]\!]^{\mathfrak{T}}\rfloor_k \neq \oplus)_{j<m}$ para todo $k > 0$. Luego, por (S-UNION-T) (notar que $1 < n \leq n'$ y $0 < m$ implica $n' + m > 2$),

$$\langle \mathcal{A}''^{\setminus\mu}\langle [\![A'{\downarrow_{\mathsf{p}_1}}]\!]^{\mathfrak{T}}\rfloor_k, \ldots, [\![A'{\downarrow_{\mathsf{p}_{n'}}}]\!]^{\mathfrak{T}}\rfloor_k\rangle, \mathcal{B}'^{\setminus\mu}\langle [\![B'{\downarrow_{\mathsf{q}_1}}]\!]^{\mathfrak{T}}\rfloor_k, \ldots, [\![B'{\downarrow_{\mathsf{q}_m}}]\!]^{\mathfrak{T}}\rfloor_k\rangle\rangle \in \Phi_{\preceq_{\mathfrak{T}}}(\mathcal{R}(\mathcal{S}))$$

Se concluye por Lem. 3.6.6 y Def. 3.3.10, $\langle [\![A']\!]^{\mathfrak{T}}\rfloor_k, [\![B']\!]^{\mathfrak{T}}\rfloor_k\rangle \in \Phi_{\preceq_{\mathfrak{T}}}(\mathcal{R}(\mathcal{S}))$.

5. $\langle A', B'\rangle = \langle A', \bigoplus_{j<m} B'_j\rangle$ con $m > 1$, $A' \neq \mu, \oplus$, $(B'_j \neq \oplus)_{j<m}$ y $\langle A', B'_{j_0}\rangle \in \mathcal{S}$ para algún $j_0 < m$. Alternativamente, puede denotarse $B = \mathcal{B}'\langle \vec{B'}\rangle$ con $\mathcal{B}'$ un $\oplus$-contexto maximal. Más aún, sea $\mathcal{B}''$ un $\mu\oplus$-contexto maximal tal que $B' = \mathcal{B}''\langle \vec{B''}\rangle$ con $(B''_l \neq \mu, \oplus)_{l<m'}$ para algún $m' \geq m$, luego existen $\mu\oplus$-contextos $(\mathcal{B}'_j)_{j<m}$ tal que $\mathcal{B}'' = \mathcal{B}'\langle \vec{\mathcal{B}'}\rangle$. Por otro lado, $A' \neq \mu, \oplus$ implica que el $\mu\oplus$-contexto maximal que lo caracteriza es $\Box$, por lo que $\mathsf{ppos}(A') = \{\epsilon\}$. Notar que puede particionarse $[0, m')$ mediante funciones $b, e : [0, m) \to [0, m')$ tal que $[0, m') = +_{j<m}[b(j), e(j)]$ y $(B'_j = \mathcal{B}'_j\langle B''_{b(j)}, \ldots, B''_{e(j)}\rangle)_{j<m}$. Por Lem. 3.6.11 con $\langle A', B'\rangle \in \mathcal{S}$, existe una función $f : [0] \to [0, m')$ tal que $\langle A'{\downarrow_\epsilon}, B'{\downarrow_{\mathsf{q}_{f(0)}}}\rangle \in \mathcal{S}$, donde $\mathsf{q}_1, \ldots, \mathsf{q}_{m'} \in \mathsf{ppos}(B')$ son las posiciones de $\Box$ en $\mathcal{B}''$. Luego, $\langle [\![A'{\downarrow_\epsilon}]\!]^{\mathfrak{T}}\rfloor_k, [\![B'{\downarrow_{\mathsf{q}_{f(0)}}}]\!]^{\mathfrak{T}}\rfloor_k\rangle \in \mathcal{R}(\mathcal{S})$ para todo $k > 0$, por definición de $\mathcal{R}$. Dado que $\mathcal{B}''$ es un $\mu\oplus$-contexto maximal, por contractividad y Def. 3.6.4 se tiene $(B'{\downarrow_{\mathsf{q}_l}} \neq \mu, \oplus)_{l<m'}$. Más aún, por Def. 3.3.10 y 3.3.22, $([\![B'{\downarrow_{\mathsf{q}_l}}]\!]^{\mathfrak{T}}\rfloor_k \neq \oplus)_{l<m'}$ para todo $k > 0$. Del mismo modo, $A \neq \mu, \oplus$ implica $[\![A'{\downarrow_\epsilon}]\!]^{\mathfrak{T}}\rfloor_k \neq \oplus$. Luego, por (S-UNION-T) (notar que $1 < m \leq m'$ implica $1 + m' > 2$), $\langle [\![A'{\downarrow_\epsilon}]\!]^{\mathfrak{T}}\rfloor_k, \mathcal{B}''^{\setminus\mu}\langle [\![B'{\downarrow_{\mathsf{q}_1}}]\!]^{\mathfrak{T}}\rfloor_k, \ldots, [\![B'{\downarrow_{\mathsf{q}_{m'}}}]\!]^{\mathfrak{T}}\rfloor_k\rangle\rangle \in \Phi_{\preceq_{\mathfrak{T}}}(\mathcal{R}(\mathcal{S}))$. Se concluye por Lem. 3.6.6 y Def. 3.3.10, $\langle [\![A']\!]^{\mathfrak{T}}\rfloor_k, [\![B']\!]^{\mathfrak{T}}\rfloor_k\rangle \in \Phi_{\preceq_{\mathfrak{T}}}(\mathcal{R}(\mathcal{S}))$.

- $c > 0$. Luego, hay dos casos posibles:

1. $\langle A', B'\rangle = \langle \mu V.A'', B'\rangle$ con $\langle \{V \setminus A'\}A'', B'\rangle \in \mathcal{S}$. Por contractividad, $\#_\mu(\{V \setminus A'\}A'') < \#_\mu(\mu V.A'')$. Luego, $\langle \{V \setminus A'\}A'', B'\rangle \in \mathcal{S}_{c'}$ con $c' < c$ y por *h.i.* se tiene
$$\langle [\![\{V \setminus A'\}A'']\!]^{\mathfrak{T}}\rfloor_k, [\![B']\!]^{\mathfrak{T}}\rfloor_k\rangle \in \Phi_{\preceq_{\mathfrak{T}}}(\mathcal{R}(\mathcal{S}))$$
para todo $k > 0$. Se concluye por Def. 3.3.22, dado que $[\![\mu V.A'']\!]^{\mathfrak{T}} = [\![\{V \setminus A'\}A'']\!]^{\mathfrak{T}}$.
2. $\langle A', B'\rangle = \langle A', \mu W.B''\rangle$ con $\langle A', \{W \setminus B'\}B''\rangle \in \mathcal{S}$. Como en el caso anterior, se aplica la *h.i.* apelando a la contractividad y se concluye por Def. 3.3.22, $\langle [\![A']\!]^{\mathfrak{T}}\rfloor_k, [\![\mu W.B'']\!]^{\mathfrak{T}}\rfloor_k\rangle \in \Phi_{\preceq_{\mathfrak{T}}}(\mathcal{R}(\mathcal{S}))$ para todo $k > 0$.

$\Leftarrow$) Se muestra que $\mathcal{R} \triangleq \{\langle A', B'\rangle \mid A', B' \in \mathcal{T}, \forall k \in \mathbb{N}. [\![A']\!]^{\mathfrak{T}}\rfloor_k \preceq_{\mathfrak{T}} [\![B']\!]^{\mathfrak{T}}\rfloor_k\}$ es $\Phi_{\preceq_{\vec{\mu}}}$-denso. Sea $\langle A, B\rangle \in \mathcal{R}$. La demostración es por análisis de casos en $c = \#_\mu(A) + \#_\mu(B)$. Se consideran únicamente los casos $k > 0$ para evitar el caso borde trivial $\langle [\![A]\!]^{\mathfrak{T}}\rfloor_0, [\![B]\!]^{\mathfrak{T}}\rfloor_0\rangle = \langle \varepsilon, \varepsilon\rangle$.

- $c = 0$. Como $\langle [\![A]\!]^{\mathfrak{T}}\rfloor_k, [\![B]\!]^{\mathfrak{T}}\rfloor_k\rangle \in \preceq_{\mathfrak{T}}$ y $\preceq_{\mathfrak{T}} = \Phi_{\preceq_{\mathfrak{T}}}(\preceq_{\mathfrak{T}})$, se tienen cuatro casos posibles:
  1. $\langle [\![A]\!]^{\mathfrak{T}}\rfloor_k, [\![B]\!]^{\mathfrak{T}}\rfloor_k\rangle = \langle a, a\rangle$. Por Def. 3.3.10 y 3.3.22 se tiene necesariamente $A = a = B$, pues $c = 0$. Luego, por (S-REFL-AL), $\langle A, B\rangle \in \Phi_{\preceq_{\vec{\mu}}}(\mathcal{R})$.
  2. $\langle [\![A]\!]^{\mathfrak{T}}\rfloor_k, [\![B]\!]^{\mathfrak{T}}\rfloor_k\rangle = \langle \mathcal{D}\rfloor_{k-1} @ \mathcal{A}'\rfloor_{k-1}, \mathcal{D}'\rfloor_{k-1} @ \mathcal{B}'\rfloor_{k-1}\rangle$ con $\mathcal{D}\rfloor_{k-1} \preceq_{\mathfrak{T}} \mathcal{D}'\rfloor_{k-1}$ y $\mathcal{A}'\rfloor_{k-1} \preceq_{\mathfrak{T}} \mathcal{B}'\rfloor_{k-1}$. Por Def. 3.3.22 y $c = 0$, existen $D, D' \in \mathcal{D}$ y $A', B' \in \mathcal{T}$ tal que $A = D @ A'$ y $B = D' @ B'$ con $[\![D]\!]^{\mathfrak{T}} = \mathcal{D}$, $[\![D]\!]^{\mathfrak{T}} = \mathcal{D}'$, $[\![A']\!]^{\mathfrak{T}} = \mathcal{A}'$ y $[\![B']\!]^{\mathfrak{T}} = \mathcal{B}'$. Más aún, por definición de $\mathcal{R}$, $\langle D, D'\rangle, \langle A', B'\rangle \in \mathcal{R}$. Luego, se concluye por (S-COMP-AL), $\langle A, B\rangle \in \Phi_{\preceq_{\vec{\mu}}}(\mathcal{R})$.
  3. $\langle [\![A]\!]^{\mathfrak{T}}\rfloor_k, [\![B]\!]^{\mathfrak{T}}\rfloor_k\rangle = \langle \mathcal{A}'\rfloor_{k-1} \supset \mathcal{A}''\rfloor_{k-1}, \mathcal{B}'\rfloor_{k-1} \supset \mathcal{B}''\rfloor_{k-1}\rangle$ con las relaciones $\mathcal{B}'\rfloor_{k-1} \preceq_{\mathfrak{T}} \mathcal{A}'\rfloor_{k-1}$ y $\mathcal{A}''\rfloor_{k-1} \preceq_{\mathfrak{T}} \mathcal{B}''\rfloor_{k-1}$. Por Def. 3.3.22 y $c = 0$, existen $A', A'', B', B'' \in \mathcal{T}$ tal que $A = A' \supset A''$ y $B = B' \supset B''$ con $[\![A']\!]^{\mathfrak{T}} = \mathcal{A}'$, $[\![A'']\!]^{\mathfrak{T}} = \mathcal{A}''$, $[\![B']\!]^{\mathfrak{T}} = \mathcal{B}'$ y $[\![B'']\!]^{\mathfrak{T}} = \mathcal{B}''$. Más aún, por definición de $\mathcal{R}$, $\langle B', A'\rangle, \langle A'', B''\rangle \in \mathcal{R}$. Luego, se concluye por (S-FUNC-AL), $\langle A, B\rangle \in \Phi_{\preceq_{\vec{\mu}}}(\mathcal{R})$.
  4. $\langle [\![A]\!]^{\mathfrak{T}}\rfloor_k, [\![B]\!]^{\mathfrak{T}}\rfloor_k\rangle = \langle \oplus_{i<n}\mathcal{A}_i\rfloor_k, \oplus_{j<m}\mathcal{B}_j\rfloor_k\rangle$ con $n + m > 2$, $(\mathcal{A}_i\rfloor_k)_{i<n}$, $(\mathcal{B}_j\rfloor_k)_{j<m}$ y una función $f : [0, n) \to [0, m)$ tal que $(\mathcal{A}_i\rfloor_k \preceq_{\mathfrak{T}} \mathcal{B}_{f(i)}\rfloor_k)_{i<n}$ y $(\mathcal{A}_{g(j)}\rfloor_k \preceq_{\mathfrak{T}} \mathcal{B}_j\rfloor_k)_{j<m}$. Se distinguen dos posibles casos:
     a) $n = 1$. Luego, $[\![A]\!]^{\mathfrak{T}}\rfloor_k = \mathcal{A}_0\rfloor_k \neq \mu, \oplus$ y $m > 1$. Más aún, por Lem. 3.6.7 sobre B, existen $m' \leq m$, $\oplus$-contextos maximales $\mathcal{B}$ y $(\mathcal{B}_l)_{l<m'}$, μ-tipos contractivos $(B_l \neq \oplus)_{l<m'}$ y funciones $b, e : [0, m') \to [0, m)$ tal que $B = \mathcal{B}\langle\vec{B}\rangle$ y $([\![B_l]\!]^{\mathfrak{T}} = \mathcal{B}_l\langle\mathcal{B}_{b(l)}, \ldots, \mathcal{B}_{e(l)}\rangle)_{l<m'}$. Sea $l_0 \in [0, m')$ tal que $f(0) \in [b(l_0), e(l_0)]$. Por (S-UNION-T) se tiene $[\![A]\!]^{\mathfrak{T}}\rfloor_k = \mathcal{A}_0\rfloor_k \preceq_{\mathfrak{T}} \oplus_{j\in[b(l_0),e(l_0)]}\mathcal{B}_j\rfloor_k = [\![B_{l_0}]\!]^{\mathfrak{T}}$ y, por lo tanto, $\langle A, B_{l_0}\rangle \in \mathcal{R}$. Más aún, $c = 0$ implica $m' > 1$. Luego, se concluye por (S-UNION-R-AL) con $f' : [0, n) \to [0, m')$ tal que $f'(0) = l_0$, $\langle A, B\rangle \in \Phi_{\preceq_{\vec{\mu}}}(\mathcal{R})$.
     b) $n > 1$. Por Lem. 3.6.7, existen $n' \leq n$, $\oplus$-contextos maximales $\mathcal{A}$ y $(\mathcal{A}_l)_{l<n'}$, μ-tipos contractivos $(A_l \neq \oplus)_{l<n'}$ y funciones $b, e : [0, n') \to [0, n)$ tal que $A = \mathcal{A}\langle\vec{A}\rangle$ y $([\![A_l]\!]^{\mathfrak{T}} = \mathcal{A}_l\langle\mathcal{A}_{b(l)}, \ldots, \mathcal{A}_{e(l)}\rangle)_{l<n'}$. Más aún, por (S-UNION-T), se obtienen las relaciones $([\![A_l]\!]^{\mathfrak{T}}\rfloor_k = \oplus_{i\in[b(l),e(l)]}\mathcal{A}_i \preceq_{\mathfrak{T}} [\![B]\!]^{\mathfrak{T}}\rfloor_k)_{l<n'}$ y, por lo tanto, $(\langle A_l, B\rangle \in \mathcal{R})_{l<n'}$. Además, $c = 0$ implica $m' > 1$ y $B \neq \oplus$. Luego, se concluye por (S-UNION-L-AL), $\langle A, B\rangle \in \Phi_{\preceq_{\vec{\mu}}}(\mathcal{R})$.
- $c > 0$. Luego, se distinguen dos casos:
  1. $\#_\mu(A) = 0$. Luego, $\#_\mu(B) > 0$. Por lo tanto, $B = \mu W.B'$ y, por Def. 3.3.22, se tiene $[\![B]\!]^{\mathfrak{T}} = [\![\{W \setminus B\}B']\!]^{\mathfrak{T}}$. Luego, $\langle A, \{W \setminus B\}B'\rangle \in \mathcal{R}$. Notar que $\#_\mu(A) = 0$ implica $A \neq \mu V.A'$. Se concluye por (S-REC-R-AL), $\langle A, B\rangle \in \Phi_{\preceq_{\vec{\mu}}}(\mathcal{R})$.
  2. $\#_\mu(A) > 0$. Luego, $A = \mu V.A'$ y, por Def. 3.3.22, $[\![A]\!]^{\mathfrak{T}} = [\![\{V \setminus A\}A']\!]^{\mathfrak{T}}$. Luego, se tiene $\langle \{V \setminus A\}A', B\rangle \in \mathcal{R}$ y se concluye por (S-REC-L-AL), $\langle A, B\rangle \in \Phi_{\preceq_{\vec{\mu}}}(\mathcal{R})$.

□

Hacia un chequeo de tipos eficiente

Chequeo de sub-tipado

Lema 3.6.18. *$\mathcal{A} \preceq_{\mathfrak{T}} \mathcal{B}$ sii $\mathcal{A} \preceq^{\emptyset}_{\mathfrak{n}} \mathcal{B}$.*

Demostración. $\Rightarrow$) Se muestra que $\mathcal{S} \triangleq \{\langle [\![\mathcal{A}]\!]^{\mathfrak{n}}, [\![\mathcal{B}]\!]^{\mathfrak{n}}\rangle \mid \mathcal{A} \preceq_{\mathfrak{T}} \mathcal{B}\}$ es $\Phi_{\preceq^{\emptyset}_{\mathfrak{n}}}$-denso. Se analiza la forma de los posibles pares en $\mathcal{S}$.

- $\langle [\![\mathcal{A}]\!]^{\mathfrak{n}}, [\![\mathcal{B}]\!]^{\mathfrak{n}}\rangle = \langle a, a\rangle$. El resultado es inmediato por definición de $\Phi_{\preceq^{\emptyset}_{\mathfrak{n}}}$.
- $\langle [\![\mathcal{A}]\!]^{\mathfrak{n}}, [\![\mathcal{B}]\!]^{\mathfrak{n}}\rangle = \langle \mathcal{T}' \mathbin{@} \mathcal{T}'', \mathcal{S}' \mathbin{@} \mathcal{S}''\rangle$. Por Def. 3.6.15, $[\![\mathcal{A}]\!]^{\mathfrak{n}}(\epsilon) = @$ sii $\mathcal{A}(\epsilon) = @$. Luego, $\mathcal{A} = \mathcal{D} \mathbin{@} \mathcal{A}'$ con $[\![\mathcal{D}]\!]^{\mathfrak{n}} = \mathcal{T}'$ y $[\![\mathcal{A}']\!]^{\mathfrak{n}} = \mathcal{T}''$. Análogamente, $\mathcal{B} = \mathcal{D}' \mathbin{@} \mathcal{B}'$ con $[\![\mathcal{D}']\!]^{\mathfrak{n}} = \mathcal{S}'$ y $[\![\mathcal{B}']\!]^{\mathfrak{n}} = \mathcal{S}''$. Más aún, por (S-COMP-T) se tienen $\mathcal{D} \preceq_{\mathfrak{T}} \mathcal{D}'$ y $\mathcal{A}' \preceq_{\mathfrak{T}} \mathcal{B}'$. Luego, $\langle [\![\mathcal{D}]\!]^{\mathfrak{n}}, [\![\mathcal{D}']\!]^{\mathfrak{n}}\rangle, \langle [\![\mathcal{A}']\!]^{\mathfrak{n}}, [\![\mathcal{B}']\!]^{\mathfrak{n}}\rangle \in \mathcal{S}$. Se concluye por (S-COMP-UP) y Def. 3.6.15, $\langle [\![\mathcal{D} \mathbin{@} \mathcal{A}']\!]^{\mathfrak{n}}, [\![\mathcal{D}' \mathbin{@} \mathcal{B}']\!]^{\mathfrak{n}}\rangle \in \Phi_{\preceq^{\emptyset}_{\mathfrak{n}}}(\mathcal{S})$.
- $\langle [\![\mathcal{A}]\!]^{\mathfrak{n}}, [\![\mathcal{B}]\!]^{\mathfrak{n}}\rangle = \langle \mathcal{T}' \supset \mathcal{T}'', \mathcal{S}' \supset \mathcal{S}''\rangle$. Por Def. 3.6.15, $[\![\mathcal{A}]\!]^{\mathfrak{n}}(\epsilon) = \supset$ sii $\mathcal{A}(\epsilon) = \supset$. Luego, $\mathcal{A} = \mathcal{A}' \supset \mathcal{A}''$ con $[\![\mathcal{A}']\!]^{\mathfrak{n}} = \mathcal{T}'$ y $[\![\mathcal{A}'']\!]^{\mathfrak{n}} = \mathcal{T}''$. Análogamente, $\mathcal{B} = \mathcal{B}' \supset \mathcal{B}''$ con $[\![\mathcal{B}']\!]^{\mathfrak{n}} = \mathcal{S}'$ y $[\![\mathcal{B}'']\!]^{\mathfrak{n}} = \mathcal{S}''$. Más aún, por (S-FUNC-T) se tienen $\mathcal{B}' \preceq_{\mathfrak{T}} \mathcal{A}'$ y $\mathcal{A}'' \preceq_{\mathfrak{T}} \mathcal{B}''$. Luego, $\langle [\![\mathcal{B}']\!]^{\mathfrak{n}}, [\![\mathcal{A}']\!]^{\mathfrak{n}}\rangle, \langle [\![\mathcal{A}'']\!]^{\mathfrak{n}}, [\![\mathcal{B}'']\!]^{\mathfrak{n}}\rangle \in \mathcal{S}$. Se concluye por (S-FUNC-UP) y Def. 3.6.15, $\langle [\![\mathcal{A}' \supset \mathcal{A}'']\!]^{\mathfrak{n}}, [\![\mathcal{B}' \supset \mathcal{B}'']\!]^{\mathfrak{n}}\rangle \in \Phi_{\preceq^{\emptyset}_{\mathfrak{n}}}(\mathcal{S})$.
- $\langle [\![\mathcal{A}]\!]^{\mathfrak{n}}, [\![\mathcal{B}]\!]^{\mathfrak{n}}\rangle = \langle \oplus^n_i \mathcal{T}_i, \mathcal{S}\rangle$ con $(\mathcal{T}_i \neq \oplus)_{i<n}$ y $\mathcal{S} \neq \oplus$. Por Def. 3.6.15, $[\![\mathcal{A}]\!]^{\mathfrak{n}}(\epsilon) = \oplus^n$ sii $\mathcal{A} = \oplus_{i<n}\mathcal{A}_i$ con $([\![\mathcal{A}_i]\!]^{\mathfrak{n}} = \mathcal{T}_i)_{i<n}$. Análogamente, $\mathcal{S} \neq \oplus$ implica $\mathcal{B} \neq \oplus$. Luego, por (S-UNION-T) se tienen $(\mathcal{A}_i \preceq_{\mathfrak{T}} \mathcal{B})_{i<n}$. Por lo tanto, $(\langle [\![\mathcal{A}_i]\!]^{\mathfrak{n}}, [\![\mathcal{B}]\!]^{\mathfrak{n}}\rangle \in \mathcal{S})_{i<n}$. Se concluye por (S-UNION-L-UP) y Def. 3.6.15, $\langle [\![\oplus^n_i \mathcal{A}_i]\!]^{\mathfrak{n}}, [\![\mathcal{B}]\!]^{\mathfrak{n}}\rangle \in \Phi_{\preceq^{\emptyset}_{\mathfrak{n}}}(\mathcal{S})$.
- $\langle [\![\mathcal{A}]\!]^{\mathfrak{n}}, [\![\mathcal{B}]\!]^{\mathfrak{n}}\rangle = \langle \mathcal{T}, \oplus^m_j \mathcal{S}_j\rangle$ con $\mathcal{T} \neq \oplus$ y $(\mathcal{S}_j \neq \oplus)_{j<m}$. Por Def. 3.6.15, $[\![\mathcal{B}]\!]^{\mathfrak{n}}(\epsilon) = \oplus^m$ sii $\mathcal{B} = \oplus_{j<m}\mathcal{B}_j$ con $([\![\mathcal{B}_j]\!]^{\mathfrak{n}} = \mathcal{S}_j)_{j<m}$. Análogamente, $\mathcal{T} \neq \oplus$ implica $\mathcal{A} \neq \oplus$. Luego, por (S-UNION-T) se tiene $\mathcal{A} \preceq_{\mathfrak{T}} \mathcal{B}_k$ para algún $k \in [0, m)$. Por lo tanto $\langle [\![\mathcal{A}]\!]^{\mathfrak{n}}, [\![\mathcal{B}_k]\!]^{\mathfrak{n}}\rangle \in \mathcal{S}$. Se concluye por (S-UNION-R-UP) y Def. 3.6.15, $\langle [\![\mathcal{A}]\!]^{\mathfrak{n}}, [\![\oplus^m_j \mathcal{B}_j]\!]^{\mathfrak{n}}\rangle \in \Phi_{\preceq^{\emptyset}_{\mathfrak{n}}}(\mathcal{S})$.
- $\langle [\![\mathcal{A}]\!]^{\mathfrak{n}}, [\![\mathcal{B}]\!]^{\mathfrak{n}}\rangle = \langle \oplus^n_i \mathcal{T}_i, \oplus^m_j \mathcal{S}_j\rangle$ con $(\mathcal{T}_i \neq \oplus)_{i<n}$ y $(\mathcal{S}_j \neq \oplus)_{j<m}$. Por Def. 3.6.15, $[\![\mathcal{A}]\!]^{\mathfrak{n}}(\epsilon) = \oplus^n$ sii $\mathcal{A} = \oplus_{i<n}\mathcal{A}_i$ con $([\![\mathcal{A}_i]\!]^{\mathfrak{n}} = \mathcal{T}_i)_{i<n}$. Análogamente, $\mathcal{B} = \oplus_{j<m}\mathcal{B}_j$ con $([\![\mathcal{B}_j]\!]^{\mathfrak{n}} = \mathcal{S}_j)_{j<n}$. Más aún, por (S-UNION-T) existe una función $f : [0, n) \to [0, m)$ tal que $(\mathcal{A}_i \preceq_{\mathfrak{T}} \mathcal{B}_{f(i)})_{i<n}$. Luego, $(\langle [\![\mathcal{A}_i]\!]^{\mathfrak{n}}, [\![\mathcal{B}_{f(i)}]\!]^{\mathfrak{n}}\rangle \in \mathcal{S})_{i<n}$. Se concluye por (S-UNION-UP) y Def. 3.6.15, $\langle [\![\oplus^n_i \mathcal{A}_i]\!]^{\mathfrak{n}}, [\![\oplus^m_j \mathcal{B}_j]\!]^{\mathfrak{n}}\rangle \in \Phi_{\preceq^{\emptyset}_{\mathfrak{n}}}(\mathcal{S})$.

$\Leftarrow$) Se muestra que $\mathcal{S} \triangleq \{\langle \mathcal{A}, \mathcal{B}\rangle \mid [\![\mathcal{A}]\!]^{\mathfrak{n}} \preceq^{\emptyset}_{\mathfrak{n}} [\![\mathcal{B}]\!]^{\mathfrak{n}}\}$ es $\Phi_{\preceq_{\mathfrak{T}}}$-denso. Se analiza la forma de los posibles pares en $\mathcal{S}$.

- $\langle \mathcal{A}, \mathcal{B}\rangle = \langle a, a\rangle$. El resultado es inmediato por definición de $\Phi_{\preceq_{\mathfrak{T}}}$.
- $\langle \mathcal{A}, \mathcal{B}\rangle = \langle \mathcal{D} \mathbin{@} \mathcal{A}', \mathcal{D}' \mathbin{@} \mathcal{B}'\rangle$. Por Def. 3.6.15, $[\![\mathcal{A}]\!]^{\mathfrak{n}} = [\![\mathcal{D}]\!]^{\mathfrak{n}} \mathbin{@} [\![\mathcal{A}']\!]^{\mathfrak{n}}$ y $[\![\mathcal{B}]\!]^{\mathfrak{n}} = [\![\mathcal{D}']\!]^{\mathfrak{n}} \mathbin{@} [\![\mathcal{B}']\!]^{\mathfrak{n}}$. Más aún, por (S-COMP-UP), $[\![\mathcal{D}]\!]^{\mathfrak{n}} \preceq^{\emptyset}_{\mathfrak{n}} [\![\mathcal{D}']\!]^{\mathfrak{n}}$ y $[\![\mathcal{A}']\!]^{\mathfrak{n}} \preceq^{\emptyset}_{\mathfrak{n}} [\![\mathcal{B}']\!]^{\mathfrak{n}}$. Luego, $\langle \mathcal{D}, \mathcal{D}'\rangle, \langle \mathcal{A}', \mathcal{B}'\rangle \in \mathcal{S}$. Se concluye por (S-COMP-T), $\langle \mathcal{D} \mathbin{@} \mathcal{A}', \mathcal{D}' \mathbin{@} \mathcal{B}'\rangle \in \Phi_{\preceq^{\emptyset}_{\mathfrak{n}}}(\mathcal{S})$.
- $\langle \mathcal{A}, \mathcal{B}\rangle = \langle \mathcal{A}' \supset \mathcal{A}'', \mathcal{B}' \supset \mathcal{B}''\rangle$. Por Def. 3.6.15, $[\![\mathcal{A}]\!]^{\mathfrak{n}} = [\![\mathcal{A}']\!]^{\mathfrak{n}} \supset [\![\mathcal{A}'']\!]^{\mathfrak{n}}$ y $[\![\mathcal{B}]\!]^{\mathfrak{n}} = [\![\mathcal{B}']\!]^{\mathfrak{n}} \supset [\![\mathcal{B}'']\!]^{\mathfrak{n}}$. Más aún, por (S-FUNC-UP), $[\![\mathcal{B}']\!]^{\mathfrak{n}} \preceq^{\emptyset}_{\mathfrak{n}} [\![\mathcal{A}']\!]^{\mathfrak{n}}$ y $[\![\mathcal{A}'']\!]^{\mathfrak{n}} \preceq^{\emptyset}_{\mathfrak{n}} [\![\mathcal{B}'']\!]^{\mathfrak{n}}$. Luego, $\langle \mathcal{B}', \mathcal{A}'\rangle, \langle \mathcal{A}'', \mathcal{B}''\rangle \in \mathcal{S}$. Se concluye por (S-FUNC-T), $\langle \mathcal{A}' \supset \mathcal{A}'', \mathcal{B}' \supset \mathcal{B}''\rangle \in \Phi_{\preceq^{\emptyset}_{\mathfrak{n}}}(\mathcal{S})$.
- $\langle \mathcal{A}, \mathcal{B}\rangle = \langle \oplus_{i<n}\mathcal{A}_i, \oplus_{j<m}\mathcal{B}_j\rangle$ con $(\mathcal{A}_i \neq \oplus)_{i<n}$, $(\mathcal{B}_j \neq \oplus)_{j<m}$ y $n + m > 2$. Luego, se distinguen tres casos:

  1. $n = 1$. Luego, $m > 1$ y $[\![\mathcal{B}]\!]^{\mathfrak{n}} = \oplus^m_j [\![\mathcal{B}_j]\!]^{\mathfrak{n}}$. Más aún, por (S-UNION-R-UP), $[\![\mathcal{A}]\!]^{\mathfrak{n}} \preceq^{\emptyset}_{\mathfrak{n}} [\![\mathcal{B}_k]\!]^{\mathfrak{n}}$ para algún $k \in [0, m)$. Por lo tanto, $\langle \mathcal{A}, \mathcal{B}_k\rangle \in \mathcal{S}$. Se concluye por (S-UNION-T), $\langle \mathcal{A}, \mathcal{B}\rangle \in \Phi_{\preceq^{\emptyset}_{\mathfrak{n}}}(\mathcal{S})$.

2. $m = 1$. Luego, $n > 1$ y $[\![\mathcal{A}]\!]^{\mathfrak{n}} = \oplus_i^n [\![\mathcal{A}_i]\!]^{\mathfrak{n}}$. Más aún, por (S-UNION-L-UP), $([\![\mathcal{A}_i]\!]^{\mathfrak{n}} \preceq_{\mathfrak{n}}^{\emptyset} [\![\mathcal{B}]\!]^{\mathfrak{n}})_{i<n}$. Por lo tanto, $(\langle \mathcal{A}_i, \mathcal{B}\rangle \in \mathcal{S})_{i<n}$. Se concluye por (S-UNION-T), $\langle \mathcal{A}, \mathcal{B}\rangle \in \Phi_{\preceq_{\mathfrak{n}}^{\emptyset}}(\mathcal{S})$.
3. $n > 1$ y $m > 1$. Luego, $[\![\mathcal{A}]\!]^{\mathfrak{n}} = \oplus_i^n [\![\mathcal{A}_i]\!]^{\mathfrak{n}}$ y $[\![\mathcal{B}]\!]^{\mathfrak{n}} = \oplus_j^m [\![\mathcal{B}_j]\!]^{\mathfrak{n}}$. Más aún, por (S-UNION-UP), existe una función $f : [0, n) \to [0, m)$ tal que $([\![\mathcal{A}_i]\!]^{\mathfrak{n}} \preceq_{\mathfrak{n}}^{\emptyset} [\![\mathcal{B}_{f(i)}]\!]^{\mathfrak{n}})_{i<n}$. Por lo tanto, se tienen $(\langle \mathcal{A}_i, \mathcal{B}_{f(i)}\rangle \in \mathcal{S})_{i<n}$. Se concluye por (S-UNION-T), $\langle \mathcal{A}, \mathcal{B}\rangle \in \Phi_{\preceq_{\mathfrak{n}}^{\emptyset}}(\mathcal{S})$.

□

Chequeo de equivalencia

Lema 3.6.23. *$\mathcal{A} \simeq_{\mathfrak{T}} \mathcal{B}$ sii $\mathcal{A} \simeq_{\mathfrak{n}}^{\emptyset} \mathcal{B}$.*

Demostración. $\Rightarrow$) Se muestra que $\mathcal{S} \triangleq \{\langle [\![\mathcal{A}]\!]^{\mathfrak{n}}, [\![\mathcal{B}]\!]^{\mathfrak{n}}\rangle \mid \mathcal{A} \simeq_{\mathfrak{T}} \mathcal{B}\}$ es $\Phi_{\simeq_{\mathfrak{n}}^{\emptyset}}$-denso. Se analiza la forma de los posibles pares en $\mathcal{S}$.

- $\langle [\![\mathcal{A}]\!]^{\mathfrak{n}}, [\![\mathcal{B}]\!]^{\mathfrak{n}}\rangle = \langle a, a\rangle$. El resultado es inmediato por definición de $\Phi_{\simeq_{\mathfrak{n}}^{\emptyset}}$.
- $\langle [\![\mathcal{A}]\!]^{\mathfrak{n}}, [\![\mathcal{B}]\!]^{\mathfrak{n}}\rangle = \langle \mathcal{T}' @ \mathcal{T}'', \mathcal{S}' @ \mathcal{S}''\rangle$. Por Def. 3.6.15, $[\![\mathcal{A}]\!]^{\mathfrak{n}}(\epsilon) = @$ sii $\mathcal{A}(\epsilon) = @$. Luego, $\mathcal{A} = \mathcal{D} @ \mathcal{A}'$ con $[\![\mathcal{D}]\!]^{\mathfrak{n}} = \mathcal{T}'$ y $[\![\mathcal{A}']\!]^{\mathfrak{n}} = \mathcal{T}''$. Análogamente, $\mathcal{B} = \mathcal{D}' @ \mathcal{B}'$ con $[\![\mathcal{D}']\!]^{\mathfrak{n}} = \mathcal{S}'$ y $[\![\mathcal{B}']\!]^{\mathfrak{n}} = \mathcal{S}''$. Más aún, por (E-COMP-T) se tienen $\mathcal{D} \simeq_{\mathfrak{T}} \mathcal{D}'$ y $\mathcal{A}' \simeq_{\mathfrak{T}} \mathcal{B}'$. Luego, $\langle [\![\mathcal{D}]\!]^{\mathfrak{n}}, [\![\mathcal{D}']\!]^{\mathfrak{n}}\rangle, \langle [\![\mathcal{A}']\!]^{\mathfrak{n}}, [\![\mathcal{B}']\!]^{\mathfrak{n}}\rangle \in \mathcal{S}$. Se concluye por (E-COMP-UP) y Def. 3.6.15, $\langle [\![\mathcal{D} @ \mathcal{A}']\!]^{\mathfrak{n}}, [\![\mathcal{D}' @ \mathcal{B}']\!]^{\mathfrak{n}}\rangle \in \Phi_{\simeq_{\mathfrak{n}}^{\emptyset}}(\mathcal{S})$.
- $\langle [\![\mathcal{A}]\!]^{\mathfrak{n}}, [\![\mathcal{B}]\!]^{\mathfrak{n}}\rangle = \langle \mathcal{T}' \supset \mathcal{T}'', \mathcal{S}' \supset \mathcal{S}''\rangle$. Por Def. 3.6.15, $[\![\mathcal{A}]\!]^{\mathfrak{n}}(\epsilon) = \supset$ sii $\mathcal{A}(\epsilon) = \supset$. Luego, $\mathcal{A} = \mathcal{A}' \supset \mathcal{A}''$ con $[\![\mathcal{A}']\!]^{\mathfrak{n}} = \mathcal{T}'$ y $[\![\mathcal{A}'']\!]^{\mathfrak{n}} = \mathcal{T}''$. Análogamente, $\mathcal{B} = \mathcal{B}' \supset \mathcal{B}''$ con $[\![\mathcal{B}']\!]^{\mathfrak{n}} = \mathcal{S}'$ y $[\![\mathcal{B}'']\!]^{\mathfrak{n}} = \mathcal{S}''$. Más aún, por (E-FUNC-T) se tienen $\mathcal{B}' \simeq_{\mathfrak{T}} \mathcal{A}'$ y $\mathcal{A}'' \simeq_{\mathfrak{T}} \mathcal{B}''$. Luego, $\langle [\![\mathcal{B}']\!]^{\mathfrak{n}}, [\![\mathcal{A}']\!]^{\mathfrak{n}}\rangle, \langle [\![\mathcal{A}'']\!]^{\mathfrak{n}}, [\![\mathcal{B}'']\!]^{\mathfrak{n}}\rangle \in \mathcal{S}$. Se concluye por (E-FUNC-UP) y Def. 3.6.15, $\langle [\![\mathcal{A}' \supset \mathcal{A}'']\!]^{\mathfrak{n}}, [\![\mathcal{B}' \supset \mathcal{B}'']\!]^{\mathfrak{n}}\rangle \in \Phi_{\simeq_{\mathfrak{n}}^{\emptyset}}(\mathcal{S})$.
- $\langle [\![\mathcal{A}]\!]^{\mathfrak{n}}, [\![\mathcal{B}]\!]^{\mathfrak{n}}\rangle = \langle \oplus_i^n \mathcal{T}_i, \mathcal{S}\rangle$ con $(\mathcal{T}_i \neq \oplus)_{i<n}$ y $\mathcal{S} \neq \oplus$. Por Def. 3.6.15, $[\![\mathcal{A}]\!]^{\mathfrak{n}}(\epsilon) = \oplus^n$ sii $\mathcal{A} = \bigoplus_{i<n} \mathcal{A}_i$ con $([\![\mathcal{A}_i]\!]^{\mathfrak{n}} = \mathcal{T}_i)_{i<n}$. Análogamente, $\mathcal{S} \neq \oplus$ implica $\mathcal{B} \neq \oplus$. Luego, por (E-UNION-T) se tienen $(\mathcal{A}_i \simeq_{\mathfrak{T}} \mathcal{B})_{i<n}$. Por lo tanto, $(\langle [\![\mathcal{A}_i]\!]^{\mathfrak{n}}, [\![\mathcal{B}]\!]^{\mathfrak{n}}\rangle \in \mathcal{S})_{i<n}$. Se concluye por (E-UNION-L-UP) y Def. 3.6.15, $\langle [\![\oplus_i^n \mathcal{A}_i]\!]^{\mathfrak{n}}, [\![\mathcal{B}]\!]^{\mathfrak{n}}\rangle \in \Phi_{\simeq_{\mathfrak{n}}^{\emptyset}}(\mathcal{S})$.
- $\langle [\![\mathcal{A}]\!]^{\mathfrak{n}}, [\![\mathcal{B}]\!]^{\mathfrak{n}}\rangle = \langle \mathcal{T}, \oplus_j^m \mathcal{S}_j\rangle$ con $\mathcal{T} \neq \oplus$ y $(\mathcal{S}_j \neq \oplus)_{j<m}$. Por Def. 3.6.15, $[\![\mathcal{B}]\!]^{\mathfrak{n}}(\epsilon) = \oplus^m$ sii $\mathcal{B} = \bigoplus_{j<m} \mathcal{B}_j$ con $([\![\mathcal{B}_j]\!]^{\mathfrak{n}} = \mathcal{S}_j)_{j<m}$. Análogamente, $\mathcal{T} \neq \oplus$ implica $\mathcal{A} \neq \oplus$. Luego, por (E-UNION-T) se tiene $(\mathcal{A} \simeq_{\mathfrak{T}} \mathcal{B}_j)_{j<m}$. Por lo tanto $(\langle [\![\mathcal{A}]\!]^{\mathfrak{n}}, [\![\mathcal{B}_j]\!]^{\mathfrak{n}}\rangle \in \mathcal{S})_{j<m}$. Se concluye por (E-UNION-R-UP) y Def. 3.6.15, $\langle [\![\mathcal{A}]\!]^{\mathfrak{n}}, [\![\oplus_j^m \mathcal{B}_j]\!]^{\mathfrak{n}}\rangle \in \Phi_{\simeq_{\mathfrak{n}}^{\emptyset}}(\mathcal{S})$.
- $\langle [\![\mathcal{A}]\!]^{\mathfrak{n}}, [\![\mathcal{B}]\!]^{\mathfrak{n}}\rangle = \langle \oplus_i^n \mathcal{T}_i, \oplus_j^m \mathcal{S}_j\rangle$ con $(\mathcal{T}_i \neq \oplus)_{i<n}$ y $(\mathcal{S}_j \neq \oplus)_{j<m}$. Por Def. 3.6.15, $[\![\mathcal{A}]\!]^{\mathfrak{n}}(\epsilon) = \oplus^n$ sii $\mathcal{A} = \bigoplus_{i<n} \mathcal{A}_i$ con $([\![\mathcal{A}_i]\!]^{\mathfrak{n}} = \mathcal{T}_i)_{i<n}$. Análogamente, $\mathcal{B} = \bigoplus_{j<m} \mathcal{B}_j$ con $([\![\mathcal{B}_j]\!]^{\mathfrak{n}} = \mathcal{S}_j)_{j<n}$. Más aún, por (E-UNION-T) existen funciones $f : [0, n) \to [0, m)$ y $g : [0, m) \to [0, n)$ tal que $(\mathcal{A}_i \simeq_{\mathfrak{T}} \mathcal{B}_{f(i)})_{i<n}$ y $(\mathcal{A}_g(j) \simeq_{\mathfrak{T}} \mathcal{B}_j)_{j<m}$. Luego, $(\langle [\![\mathcal{A}_i]\!]^{\mathfrak{n}}, [\![\mathcal{B}_{f(i)}]\!]^{\mathfrak{n}}\rangle \in \mathcal{S})_{i<n}$ y $(\langle [\![\mathcal{A}_{g(j)}]\!]^{\mathfrak{n}}, [\![\mathcal{B}_j]\!]^{\mathfrak{n}}\rangle \in \mathcal{S})_{j<m}$. Se concluye por (E-UNION-UP) y Def. 3.6.15, $\langle [\![\oplus_i^n \mathcal{A}_i]\!]^{\mathfrak{n}}, [\![\oplus_j^m \mathcal{B}_j]\!]^{\mathfrak{n}}\rangle \in \Phi_{\simeq_{\mathfrak{n}}^{\emptyset}}(\mathcal{S})$.

$\Leftarrow$) Se muestra que $\mathcal{S} \triangleq \{\langle \mathcal{A}, \mathcal{B}\rangle \mid [\![\mathcal{A}]\!]^{\mathfrak{n}} \simeq_{\mathfrak{n}}^{\emptyset} [\![\mathcal{B}]\!]^{\mathfrak{n}}\}$ es $\Phi_{\simeq_{\mathfrak{T}}}$-denso. Se analiza la forma de los posibles pares en $\mathcal{S}$.

- $\langle \mathcal{A}, \mathcal{B}\rangle = \langle a, a\rangle$. El resultado es inmediato por definición de $\Phi_{\simeq_{\mathfrak{T}}}$.
- $\langle \mathcal{A}, \mathcal{B}\rangle = \langle \mathcal{D} @ \mathcal{A}', \mathcal{D}' @ \mathcal{B}'\rangle$. Por Def. 3.6.15, $[\![\mathcal{A}]\!]^{\mathfrak{n}} = [\![\mathcal{D}]\!]^{\mathfrak{n}} @ [\![\mathcal{A}']\!]^{\mathfrak{n}}$ y $[\![\mathcal{B}]\!]^{\mathfrak{n}} = [\![\mathcal{D}']\!]^{\mathfrak{n}} @ [\![\mathcal{B}']\!]^{\mathfrak{n}}$. Más aún, por (E-COMP-UP), $[\![\mathcal{D}]\!]^{\mathfrak{n}} \simeq_{\mathfrak{n}}^{\emptyset} [\![\mathcal{D}']\!]^{\mathfrak{n}}$ y $[\![\mathcal{A}']\!]^{\mathfrak{n}} \simeq_{\mathfrak{n}}^{\emptyset} [\![\mathcal{B}']\!]^{\mathfrak{n}}$. Luego, $\langle \mathcal{D}, \mathcal{D}'\rangle, \langle \mathcal{A}', \mathcal{B}'\rangle \in \mathcal{S}$. Se concluye por (E-COMP-T), $\langle \mathcal{D} @ \mathcal{A}', \mathcal{D}' @ \mathcal{B}'\rangle \in \Phi_{\simeq_{\mathfrak{n}}^{\emptyset}}(\mathcal{S})$.

- $\langle \mathcal{A}, \mathcal{B} \rangle = \langle \mathcal{A}' \supset \mathcal{A}'', \mathcal{B}' \supset \mathcal{B}'' \rangle$. Por Def. 3.6.15, $[\![\mathcal{A}]\!]^{\mathfrak{n}} = [\![\mathcal{A}']\!]^{\mathfrak{n}} \supset [\![\mathcal{A}'']\!]^{\mathfrak{n}}$ y $[\![\mathcal{B}]\!]^{\mathfrak{n}} = [\![\mathcal{B}']\!]^{\mathfrak{n}} \supset [\![\mathcal{B}'']\!]^{\mathfrak{n}}$. Más aún, por (E-FUNC-UP), $[\![\mathcal{B}']\!]^{\mathfrak{n}} \simeq^{\emptyset}_{\mathfrak{n}} [\![\mathcal{A}']\!]^{\mathfrak{n}}$ y $[\![\mathcal{A}'']\!]^{\mathfrak{n}} \simeq^{\emptyset}_{\mathfrak{n}} [\![\mathcal{B}'']\!]^{\mathfrak{n}}$. Luego, $\langle \mathcal{B}', \mathcal{A}' \rangle, \langle \mathcal{A}'', \mathcal{B}'' \rangle \in \mathcal{S}$. Se concluye por (E-FUNC-T), $\langle \mathcal{A}' \supset \mathcal{A}'', \mathcal{B}' \supset \mathcal{B}'' \rangle \in \Phi_{\simeq^{\emptyset}_{\mathfrak{n}}}(\mathcal{S})$.
- $\langle \mathcal{A}, \mathcal{B} \rangle = \langle \oplus_{i<n} \mathcal{A}_i, \oplus_{j<m} \mathcal{B}_j \rangle$ con $(\mathcal{A}_i \neq \oplus)_{i<n}$, $(\mathcal{B}_j \neq \oplus)_{j<m}$ y $n + m > 2$. Luego, se distinguen tres casos:
  1. $n = 1$. Luego, $m > 1$ y $[\![\mathcal{B}]\!]^{\mathfrak{n}} = \oplus_j^m [\![\mathcal{B}_j]\!]^{\mathfrak{n}}$. Más aún, por (E-UNION-R-UP), $([\![\mathcal{A}]\!]^{\mathfrak{n}} \simeq^{\emptyset}_{\mathfrak{n}} [\![\mathcal{B}_j]\!]^{\mathfrak{n}})_{j<m}$. Por lo tanto, $(\langle \mathcal{A}, \mathcal{B}_j \rangle \in \mathcal{S})_{j<m}$. Se concluye por (E-UNION-T), $\langle \mathcal{A}, \mathcal{B} \rangle \in \Phi_{\simeq^{\emptyset}_{\mathfrak{n}}}(\mathcal{S})$.
  2. $m = 1$. Luego, $n > 1$ y $[\![\mathcal{A}]\!]^{\mathfrak{n}} = \oplus_i^n [\![\mathcal{A}_i]\!]^{\mathfrak{n}}$. Más aún, por (E-UNION-L-UP), $([\![\mathcal{A}_i]\!]^{\mathfrak{n}} \simeq^{\emptyset}_{\mathfrak{n}} [\![\mathcal{B}]\!]^{\mathfrak{n}})_{i<n}$. Por lo tanto, $(\langle \mathcal{A}_i, \mathcal{B} \rangle \in \mathcal{S})_{i<n}$. Se concluye por (E-UNION-T), $\langle \mathcal{A}, \mathcal{B} \rangle \in \Phi_{\simeq^{\emptyset}_{\mathfrak{n}}}(\mathcal{S})$.
  3. $n > 1$ y $m > 1$. Luego, $[\![\mathcal{A}]\!]^{\mathfrak{n}} = \oplus_i^n [\![\mathcal{A}_i]\!]^{\mathfrak{n}}$ y $[\![\mathcal{B}]\!]^{\mathfrak{n}} = \oplus_j^m [\![\mathcal{B}_j]\!]^{\mathfrak{n}}$. Más aún, por (E-UNION-UP), existen funciones $f : [0, n) \to [0, m)$ y $g : [0, m) \to [0, n)$ tal que $([\![\mathcal{A}_i]\!]^{\mathfrak{n}} \simeq^{\emptyset}_{\mathfrak{n}} [\![\mathcal{B}_{f(i)}]\!]^{\mathfrak{n}})_{i<n}$ y $([\![\mathcal{A}_{g(j)}]\!]^{\mathfrak{n}} \simeq^{\emptyset}_{\mathfrak{n}} [\![\mathcal{B}_j]\!]^{\mathfrak{n}})_{j<m}$. Por lo tanto, $(\langle \mathcal{A}_i, \mathcal{B}_{f(i)} \rangle \in \mathcal{S})_{i<n}$ y $(\langle \mathcal{A}_{g(j)}, \mathcal{B}_j \rangle \in \mathcal{S})_{j<m}$. Se concluye por (E-UNION-T), $\langle \mathcal{A}, \mathcal{B} \rangle \in \Phi_{\simeq^{\emptyset}_{\mathfrak{n}}}(\mathcal{S})$.

□

Apéndice C

Pruebas: Bisimulación fuerte para operadores de control

El ΛM-cálculo

Semántica operacional

Lema 4.2.1. *Sea $o \in \mathbb{O}_{\lambda\mu}$. Si $o \to_{\lambda\mu} o'$, entonces $o \twoheadrightarrow_{\Lambda M} o'$.*

Demostración. Por definición, $o \to_{\lambda\mu} o'$ implica $o = \mathtt{O}\langle l\rangle$ y $o' = \mathtt{O}\langle r\rangle$ con $l \mapsto_* r$, $* \in \{\beta, \mu\}$. La demostración es por inducción en $\mathtt{O}$.

- $\mathtt{O} = \Box$. Se tiene dos casos posibles según la regla de reescritura aplicada.
  1. β. Luego, $o = (\lambda x.t)\, u$ y $o' = \{x \setminus u\}t$. Se concluye pues $o \to_{\mathtt{dB}} t[x \setminus u] \to_{\mathtt{S}} o'$.
  2. μ. Luego, $o = (\mu\alpha.c)\, u$ y $o' = \mu\alpha'.\{\!\{\alpha \setminus^{\alpha'} u\}\!\}c$. Se concluye pues $o \to_{\mathtt{dM}} \mu\alpha'.c[\![\alpha \setminus^{\alpha'} u]\!] \to_{\mathtt{R}} o'$.
- $\mathtt{O} = \mathtt{T}\, u$. Luego, $o = t\, u$ y $o' = t'\, u$ con $t \to_{\lambda\mu} t'$. Por *h.i.* se tiene $t \twoheadrightarrow_{\Lambda M} t'$, por lo que se concluye $o \twoheadrightarrow_{\Lambda M} o'$.
- $\mathtt{O} = t\,\mathtt{T}$. Luego, $o = t\, u$ y $o' = t\, u'$ con $u \to_{\lambda\mu} u'$. Por *h.i.* se tiene $u \twoheadrightarrow_{\Lambda M} u'$, por lo que se concluye $o \twoheadrightarrow_{\Lambda M} o'$.
- $\mathtt{O} = \lambda x.\mathtt{T}$. Luego, $o = \lambda x.t$ y $o' = \lambda x.t'$ con $t \to_{\lambda\mu} t'$. Por *h.i.* se tiene $t \twoheadrightarrow_{\Lambda M} t'$, por lo que se concluye $o \twoheadrightarrow_{\Lambda M} o'$.
- $\mathtt{O} = \mu\alpha.\mathtt{C}$. Luego, $o = \mu\alpha.c$ y $o' = \mu\alpha.c'$ con $c \to_{\lambda\mu} c'$. Por *h.i.* se tiene $c \twoheadrightarrow_{\Lambda M} c'$, por lo que se concluye $o \twoheadrightarrow_{\Lambda M} o'$.
- $\mathtt{O} = [\alpha]\,\mathtt{T}$. Luego, $o = [\alpha]\, t$ y $o' = [\alpha]\, t'$ con $t \to_{\lambda\mu} t'$. Por *h.i.* se tiene $t \twoheadrightarrow_{\Lambda M} t'$, por lo que se concluye $o \twoheadrightarrow_{\Lambda M} o'$.

□

Confluencia

Para probar la confluencia del ΛM-cálculo se apela al método de interpretación de Hardin [CHL96], donde ΛM es proyectado sobre el $\lambda\mu\rho$-cálculo introducido a continuación.

Dado conjuntos infinito numerables de variables $\mathbb{V}$ $(x, y, \ldots)$ y nombres $\mathbb{N}$ $(\alpha, \beta, \ldots)$, los conjuntos de *objetos* $\mathbb{O}_{\lambda\mu\rho}$, *términos* $\mathbb{T}_{\lambda\mu\rho}$, *comandos* $\mathbb{C}_{\lambda\mu\rho}$, *stacks* y *contextos* del $\lambda\mu\rho$-cálculo se definen mediante la siguiente gramática:

$$
\begin{array}{rlll}
\textbf{Objetos} & o & ::= & t \mid c \mid s \\
\textbf{Términos} & t & ::= & x \in \mathbb{V} \mid t\,t \mid \lambda x.t \mid \mu\alpha.c \\
\textbf{Comandos} & c & ::= & [\alpha]\,t \mid c[\alpha \setminus \beta] \\
\textbf{Stacks} & s & ::= & t \mid t \cdot s \\
\textbf{Contextos} & \mathtt{O} & ::= & \mathtt{T} \mid \mathtt{C} \mid \mathtt{S} \\
\textbf{Contextos de término} & \mathtt{T} & ::= & \Box \mid \mathtt{T}\,t \mid t\,\mathtt{T} \mid \lambda x.\mathtt{T} \mid \mu\alpha.\mathtt{C} \\
\textbf{Contextos de comando} & \mathtt{C} & ::= & \boxdot \mid [\alpha]\,\mathtt{T} \mid \mathtt{C}[\alpha \setminus \beta] \\
\textbf{Contextos de stack} & \mathtt{S} & ::= & \boxdot \mid \mathtt{T} \cdot s \mid t \cdot \mathtt{S}
\end{array}
$$

Los comandos del $\lambda\mu$-cálculo son enriquecidos un operador de *renaming explícito* $[\alpha \setminus \beta]$ (donde α es el nombre reemplazado y β el introducido en su lugar). Al igual que para ΛM-cálculo, los stacks son esencialmente pilas no vacías de términos y son, en este caso, utilizados únicamente para la operación de (meta-)replacement. Notar sin embargo, que $\mathbb{O}_{\lambda\mu\rho} \subset \mathbb{O}_{\Lambda M}$ (análogamente $\mathbb{T}_{\lambda\mu\rho} \subset \mathbb{T}_{\Lambda M}$ y $\mathbb{C}_{\lambda\mu\rho} \subset \mathbb{C}_{\Lambda M}$).

Se denota con $\vec{\alpha}$ una secuencia no vacía de nombres distintos $\alpha_1, \ldots, \alpha_n$ y con $\mathsf{sz}(\vec{\alpha})$ el número de nombres en la secuencia. La *operación de replacement simultáneo* se define inductivamente como:

$$
\begin{array}{rcll}
\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle x & \triangleq & x & \\
\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle (t\,u) & \triangleq & \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle t\, \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle u & \\
\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle (\lambda x.t) & \triangleq & \lambda x.\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle t & x \notin s \\
\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle (\mu\gamma.c) & \triangleq & \mu\gamma.\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle c & \gamma \notin s, \gamma \notin \vec{\alpha}, \gamma \notin \vec{\beta} \\
\langle\!\langle \vec{\alpha_1}\alpha\vec{\alpha_2} \setminus^{\vec{\beta_1}\beta\vec{\beta_2}} s \rangle\!\rangle ([\alpha]\,t) & \triangleq & [\beta]\,(\langle\!\langle \vec{\alpha_1}\alpha\vec{\alpha_2} \setminus^{\vec{\beta_1}\beta\vec{\beta_2}} s \rangle\!\rangle t) :: s & \\
\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle ([\gamma]\,t) & \triangleq & [\gamma]\,\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle t & \gamma \notin \vec{\alpha} \\
\langle\!\langle \vec{\alpha_1}\alpha\vec{\alpha_2} \setminus^{\vec{\beta_1}\beta\vec{\beta_2}} s \rangle\!\rangle (c[\gamma \setminus \alpha]) & \triangleq & \langle\!\langle \gamma\vec{\alpha_1}\alpha\vec{\alpha_2} \setminus^{\delta\vec{\beta_1}\beta\vec{\beta_2}} s \rangle\!\rangle c[\delta \setminus \beta] & \delta \text{ fresca} \\
\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle (c[\gamma \setminus \beta]) & \triangleq & (\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle c)[\gamma \setminus \beta] & \gamma \notin \vec{\alpha}, \gamma \notin \vec{\beta} \\
\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle (t \cdot s') & \triangleq & \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle t \cdot \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle s' &
\end{array}
$$

La necesidad de realizar replacements simultáneos se manifiesta en la primera cláusula de definición para renaming explícitos, donde se acumulan los nombres γ y δ (este último fresco) al continuar aplicando el replacement inductivamente sobre el comando c.

En tanto, la *aplicación de una sustitución* $\{x \setminus u\}$ a un objeto o de $\lambda\mu\rho$ se extiende a objetos del $\lambda\mu$-cálculo de manera natural, quedando definida módulo α-conversión para evitar capturas de variables/nombres libres, al igual que en la Sec. 4.1.1.

La *relación de reducción en un paso* $\to_{\lambda\mu\rho}$ se define como la clausura por contextos $\mathtt{O_t}$ de las reglas de reescritura β y μ^s:

$$
\begin{array}{rcl}
(\lambda x.t)\,u & \mapsto_{\beta} & \{x \setminus u\}t \\
(\mu\alpha.c) :: s & \mapsto_{\mu^s} & \mu\alpha'.\langle\!\langle \alpha \setminus^{\alpha'} s \rangle\!\rangle c
\end{array}
$$

donde $\mapsto_{\mu^s}$ requiere que α' sea un nombre fresco (*i.e.* $\alpha' \neq \alpha$ y $\alpha' \notin c$).

Se introducen a continuación una serie de resultados auxiliares útiles sobre la operación de replacement simultáneo.

Lema C.0.1. *Sean $o \in \mathbb{O}_{\lambda\mu\rho}$ y s un stack. Luego, $\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle o = \langle\!\langle p(\vec{\alpha}) \setminus^{p(\vec{\beta})} s \rangle\!\rangle o$ para toda permutación p del intervalo $[1, \mathsf{sz}(\vec{\alpha})]$.*

Demostración. Por inducción en o.

- $o = x$. Luego, el resultado es inmediato pues $\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle o = x = \langle p(\vec{\alpha}) \setminus^{p(\vec{\beta})} s\rangle o$.
- $o = t\,u$. Luego, $\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle o = \langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle t\,\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle u$. Por *h.i.* se tienen $\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle t = \langle p(\vec{\alpha}) \setminus^{p(\vec{\beta})} s\rangle t$ y $\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle u = \langle p(\vec{\alpha}) \setminus^{p(\vec{\beta})} s\rangle u$, por lo que se concluye $\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle o = \langle p(\vec{\alpha}) \setminus^{p(\vec{\beta})} s\rangle o$.
- $o = \lambda x.t$ con $x \notin s$. Luego, $\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle o = \lambda x.\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle t$. Por *h.i.* se tiene $\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle t = \langle p(\vec{\alpha}) \setminus^{p(\vec{\beta})} s\rangle t$, por lo que se concluye $\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle o = \langle p(\vec{\alpha}) \setminus^{p(\vec{\beta})} s\rangle o$.
- $o = \mu\gamma.c$ con $\gamma \notin \vec{\alpha}$, $\gamma \notin \vec{\beta}$ y $\gamma \notin s$. Luego, $\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle o = \mu\gamma.\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle c$. Por *h.i.* se tiene $\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle c = \langle p(\vec{\alpha}) \setminus^{p(\vec{\beta})} s\rangle c$, por lo que se concluye $\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle o = \langle p(\vec{\alpha}) \setminus^{p(\vec{\beta})} s\rangle o$.
- $o = [\delta]\,t$. Luego, se tienen dos casos posibles:
  - $\delta \in \vec{\alpha}$. Luego, $\delta = \alpha_k = \alpha_{p(k)}$ para algún $k \in [1, \mathsf{sz}(\vec{\alpha})]$. Por *h.i.* se tiene $\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle t = \langle p(\vec{\alpha}) \setminus^{p(\vec{\beta})} s\rangle t$. Finalmente, $\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle o = [\beta_k]\,\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle t :: s = [\beta_{p(k)}]\,\langle p(\vec{\alpha}) \setminus^{p(\vec{\beta})} s\rangle t :: s = \langle p(\vec{\alpha}) \setminus^{p(\vec{\beta})} s\rangle o$, por lo que se concluye.
  - $\delta \notin \vec{\alpha}$. Luego, $\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle o = [\delta]\,\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle t$. Por *h.i.* se tiene $\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle t = \langle p(\vec{\alpha}) \setminus^{p(\vec{\beta})} s\rangle t$, por lo que se concluye $\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle o = \langle p(\vec{\alpha}) \setminus^{p(\vec{\beta})} s\rangle o$.
- $o = c[\gamma \setminus \delta]$ con $\gamma \notin \vec{\alpha}$, $\gamma \notin \vec{\beta}$ y $\gamma \notin s$. Luego, se tienen dos casos posibles:
  - $\delta \in \vec{\alpha}$. Luego, $\delta = \alpha_k = \alpha_{p(k)}$ para algún $k \in [1, \mathsf{sz}(\vec{\alpha})]$. Más aún, sea γ' un nombre fresco. Por *h.i.* se tiene $\langle\gamma\vec{\alpha} \setminus^{\gamma'\vec{\beta}} s\rangle c = \langle\gamma p(\vec{\alpha}) \setminus^{\gamma' p(\vec{\beta})} s\rangle c$. Finalmente, $\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle o = (\langle\gamma\vec{\alpha} \setminus^{\gamma'\vec{\beta}} s\rangle c)[\gamma' \setminus \beta_k] = (\langle\gamma p(\vec{\alpha}) \setminus^{\gamma' p(\vec{\beta})} s\rangle c)[\gamma' \setminus \beta_{p(k)}] = \langle p(\vec{\alpha}) \setminus^{p(\vec{\beta})} s\rangle o$, por lo que se concluye.
  - $\delta \notin \vec{\alpha}$. Luego, $\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle o = (\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle c)[\gamma \setminus \delta]$. Por *h.i.* se tiene $\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle c = \langle p(\vec{\alpha}) \setminus^{p(\vec{\beta})} s\rangle c$, por lo que se concluye $\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle o = \langle p(\vec{\alpha}) \setminus^{p(\vec{\beta})} s\rangle o$.
- $o = t \cdot s'$. Luego, $\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle o = \langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle t \cdot \langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle s'$. Por *h.i.* se tienen $\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle t = \langle p(\vec{\alpha}) \setminus^{p(\vec{\beta})} s\rangle t$ y $\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle s' = \langle p(\vec{\alpha}) \setminus^{p(\vec{\beta})} s\rangle s'$, por lo que se concluye $\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle o = \langle p(\vec{\alpha}) \setminus^{p(\vec{\beta})} s\rangle o$.

□

Lema C.0.2. *Sean $o \in \mathbb{O}_{\lambda\mu\rho}$ y s un stack. Si $\alpha \notin \mathsf{fn}(o)$, entonces $\langle\vec{\alpha_1}\alpha\vec{\alpha_2} \setminus^{\vec{\beta_1}\beta\vec{\beta_2}} s\rangle o = \langle\vec{\alpha_1}\vec{\alpha_2} \setminus^{\vec{\beta_1}\vec{\beta_2}} s\rangle o$.*

Demostración. Por inducción en o.

- $o = x$. Luego, el resultado es inmediato pues $\langle\vec{\alpha_1}\alpha\vec{\alpha_2} \setminus^{\vec{\beta_1}\beta\vec{\beta_2}} s\rangle o = x = \langle\vec{\alpha_1}\vec{\alpha_2} \setminus^{\vec{\beta_1}\vec{\beta_2}} s\rangle o$.
- $o = t\,u$. Luego, $\langle\vec{\alpha_1}\alpha\vec{\alpha_2} \setminus^{\vec{\beta_1}\beta\vec{\beta_2}} s\rangle o = \langle\vec{\alpha_1}\alpha\vec{\alpha_2} \setminus^{\vec{\beta_1}\beta\vec{\beta_2}} s\rangle t\,\langle\vec{\alpha_1}\alpha\vec{\alpha_2} \setminus^{\vec{\beta_1}\beta\vec{\beta_2}} s\rangle u$. Más aún, $\alpha \notin \mathsf{fn}(o)$ implica $\alpha \notin \mathsf{fn}(t)$ y $\alpha \notin \mathsf{fn}(u)$. Entonces, por *h.i.* se tienen $\langle\vec{\alpha_1}\alpha\vec{\alpha_2} \setminus^{\vec{\beta_1}\beta\vec{\beta_2}} s\rangle t = \langle\vec{\alpha_1}\vec{\alpha_2} \setminus^{\vec{\beta_1}\vec{\beta_2}} s\rangle t$ y $\langle\vec{\alpha_1}\alpha\vec{\alpha_2} \setminus^{\vec{\beta_1}\beta\vec{\beta_2}} s\rangle u = \langle\vec{\alpha_1}\vec{\alpha_2} \setminus^{\vec{\beta_1}\vec{\beta_2}} s\rangle u$. Luego, se concluye $\langle\vec{\alpha_1}\alpha\vec{\alpha_2} \setminus^{\vec{\beta_1}\beta\vec{\beta_2}} s\rangle o = \langle\vec{\alpha_1}\vec{\alpha_2} \setminus^{\vec{\beta_1}\vec{\beta_2}} s\rangle o$.
- $o = \lambda x.t$ con $x \notin s$. Luego, $\langle\vec{\alpha_1}\alpha\vec{\alpha_2} \setminus^{\vec{\beta_1}\beta\vec{\beta_2}} s\rangle o = \lambda x.\langle\vec{\alpha_1}\alpha\vec{\alpha_2} \setminus^{\vec{\beta_1}\beta\vec{\beta_2}} s\rangle t$. Más aún, $\alpha \notin \mathsf{fn}(o)$ implica $\alpha \notin \mathsf{fn}(t)$. Entonces, por *h.i.* se tiene $\langle\vec{\alpha_1}\alpha\vec{\alpha_2} \setminus^{\vec{\beta_1}\beta\vec{\beta_2}} s\rangle t = \langle\vec{\alpha_1}\vec{\alpha_2} \setminus^{\vec{\beta_1}\vec{\beta_2}} s\rangle t$, por lo que se concluye $\langle\vec{\alpha_1}\alpha\vec{\alpha_2} \setminus^{\vec{\beta_1}\beta\vec{\beta_2}} s\rangle o = \langle\vec{\alpha_1}\vec{\alpha_2} \setminus^{\vec{\beta_1}\vec{\beta_2}} s\rangle o$.
- $o = \mu\gamma.c$ con $\gamma \notin \vec{\alpha_1}\alpha\vec{\alpha_2}$, $\gamma \notin \vec{\beta_1}\beta\vec{\beta_2}$ y $\gamma \notin s$. Luego, $\langle\vec{\alpha_1}\alpha\vec{\alpha_2} \setminus^{\vec{\beta_1}\beta\vec{\beta_2}} s\rangle o = \mu\gamma.\langle\vec{\alpha_1}\alpha\vec{\alpha_2} \setminus^{\vec{\beta_1}\beta\vec{\beta_2}} s\rangle c$. Más aún, $\alpha \notin \mathsf{fn}(o)$ y $\gamma \neq \alpha$ implican $\alpha \notin \mathsf{fn}(c)$. Entonces, por *h.i.* se tiene $\langle\vec{\alpha_1}\alpha\vec{\alpha_2} \setminus^{\vec{\beta_1}\beta\vec{\beta_2}} s\rangle c = \langle\vec{\alpha_1}\vec{\alpha_2} \setminus^{\vec{\beta_1}\vec{\beta_2}} s\rangle c$, por lo que se concluye $\langle\vec{\alpha_1}\alpha\vec{\alpha_2} \setminus^{\vec{\beta_1}\beta\vec{\beta_2}} s\rangle o = \langle\vec{\alpha_1}\vec{\alpha_2} \setminus^{\vec{\beta_1}\vec{\beta_2}} s\rangle o$.

- $o = [\delta]\, t$. Luego, $\alpha \notin \mathsf{fn}(o)$ implica $\delta \neq \alpha$. Se tienen dos casos posibles:
  - $\delta \in \vec{\alpha_1}\vec{\alpha_2}$. Luego, $\alpha \notin \mathsf{fn}(o)$ implica $\alpha \notin \mathsf{fn}(t)$ y por *h.i.* se tiene $\langle\!\langle \vec{\alpha_1}\alpha\vec{\alpha_2} \setminus^{\vec{\beta_1}\beta\vec{\beta_2}} s \rangle\!\rangle t = \langle\!\langle \vec{\alpha_1}\vec{\alpha_2} \setminus^{\vec{\beta_1}\vec{\beta_2}} s \rangle\!\rangle t$. Sea $\delta' \in \vec{\beta_1}\vec{\beta_2}$ en la misma posición que δ en $\vec{\alpha_1}\vec{\alpha_2}$. Entonces, se tiene $\langle\!\langle \vec{\alpha_1}\alpha\vec{\alpha_2} \setminus^{\vec{\beta_1}\beta\vec{\beta_2}} s \rangle\!\rangle o = [\delta']\, \langle\!\langle \vec{\alpha_1}\alpha\vec{\alpha_2} \setminus^{\vec{\beta_1}\beta\vec{\beta_2}} s \rangle\!\rangle t :: s = [\delta']\, \langle\!\langle \vec{\alpha_1}\vec{\alpha_2} \setminus^{\vec{\beta_1}\vec{\beta_2}} s \rangle\!\rangle t :: s = \langle\!\langle \vec{\alpha_1}\vec{\alpha_2} \setminus^{\vec{\beta_1}\vec{\beta_2}} s \rangle\!\rangle o$, por lo que se concluye.
  - $\delta \notin \vec{\alpha_1}\vec{\alpha_2}$. Luego, $\langle\!\langle \vec{\alpha_1}\alpha\vec{\alpha_2} \setminus^{\vec{\beta_1}\beta\vec{\beta_2}} s \rangle\!\rangle o = [\delta]\, \langle\!\langle \vec{\alpha_1}\alpha\vec{\alpha_2} \setminus^{\vec{\beta_1}\beta\vec{\beta_2}} s \rangle\!\rangle t$. Más aún, $\alpha \notin \mathsf{fn}(o)$ implica $\alpha \notin \mathsf{fn}(t)$. Entonces, por *h.i.* se tiene $\langle\!\langle \vec{\alpha_1}\alpha\vec{\alpha_2} \setminus^{\vec{\beta_1}\beta\vec{\beta_2}} s \rangle\!\rangle t = \langle\!\langle \vec{\alpha_1}\vec{\alpha_2} \setminus^{\vec{\beta_1}\vec{\beta_2}} s \rangle\!\rangle t$, por lo que se concluye $\langle\!\langle \vec{\alpha_1}\alpha\vec{\alpha_2} \setminus^{\vec{\beta_1}\beta\vec{\beta_2}} s \rangle\!\rangle o = \langle\!\langle \vec{\alpha_1}\vec{\alpha_2} \setminus^{\vec{\beta_1}\vec{\beta_2}} s \rangle\!\rangle o$.
- $o = c[\gamma \setminus \delta]$ con $\gamma \notin \vec{\alpha_1}\alpha\vec{\alpha_2}$, $\gamma \notin \vec{\beta_1}\beta\vec{\beta_2}$ y $\gamma \notin s$. Luego, $\alpha \notin \mathsf{fn}(o)$ implica $\delta \neq \alpha$. Se tienen dos casos posibles:
  - $\delta \in \vec{\alpha_1}\vec{\alpha_2}$. Luego, $\alpha \notin \mathsf{fn}(o)$ implica $\alpha \notin \mathsf{fn}(c)$. Más aún, sea γ' un nombre fresco. Por *h.i.* se tiene $\langle\!\langle \gamma\vec{\alpha_1}\alpha\vec{\alpha_2} \setminus^{\gamma'\vec{\beta_1}\beta\vec{\beta_2}} s \rangle\!\rangle c = \langle\!\langle \gamma\vec{\alpha_1}\vec{\alpha_2} \setminus^{\gamma'\vec{\beta_1}\beta\vec{\beta_2}} s \rangle\!\rangle c$. Sea $\delta' \in \vec{\beta_1}\vec{\beta_2}$ en la misma posición que δ en $\vec{\alpha_1}\vec{\alpha_2}$. Entonces, $\langle\!\langle \vec{\alpha_1}\alpha\vec{\alpha_2} \setminus^{\vec{\beta_1}\beta\vec{\beta_2}} s \rangle\!\rangle o = (\langle\!\langle \gamma\vec{\alpha_1}\alpha\vec{\alpha_2} \setminus^{\gamma'\vec{\beta_1}\beta\vec{\beta_2}} s \rangle\!\rangle c)[\gamma' \setminus \delta'] = (\langle\!\langle \gamma\vec{\alpha_1}\vec{\alpha_2} \setminus^{\gamma'\vec{\beta_1}\vec{\beta_2}} s \rangle\!\rangle c)[\gamma' \setminus \delta'] = \langle\!\langle \vec{\alpha_1}\vec{\alpha_2} \setminus^{\vec{\beta_1}\vec{\beta_2}} s \rangle\!\rangle o$, por lo que se concluye.
  - $\delta \notin \vec{\alpha_1}\vec{\alpha_2}$. Luego, $\langle\!\langle \vec{\alpha_1}\alpha\vec{\alpha_2} \setminus^{\vec{\beta_1}\beta\vec{\beta_2}} s \rangle\!\rangle o = (\langle\!\langle \vec{\alpha_1}\alpha\vec{\alpha_2} \setminus^{\vec{\beta_1}\beta\vec{\beta_2}} s \rangle\!\rangle c)[\gamma \setminus \delta]$. Más aún, $\alpha \notin \mathsf{fn}(o)$ y $\gamma \neq \alpha$ implican $\alpha \notin \mathsf{fn}(c)$. Entonces, por *h.i.* se tiene $\langle\!\langle \vec{\alpha_1}\alpha\vec{\alpha_2} \setminus^{\vec{\beta_1}\beta\vec{\beta_2}} s \rangle\!\rangle c = \langle\!\langle \vec{\alpha_1}\vec{\alpha_2} \setminus^{\vec{\beta_1}\vec{\beta_2}} s \rangle\!\rangle c$, por lo que se concluye $\langle\!\langle \vec{\alpha_1}\alpha\vec{\alpha_2} \setminus^{\vec{\beta_1}\beta\vec{\beta_2}} s \rangle\!\rangle o = \langle\!\langle \vec{\alpha_1}\vec{\alpha_2} \setminus^{\vec{\beta_1}\vec{\beta_2}} s \rangle\!\rangle o$.
- $o = t \cdot s'$. Luego, $\langle\!\langle \vec{\alpha_1}\alpha\vec{\alpha_2} \setminus^{\vec{\beta_1}\beta\vec{\beta_2}} s \rangle\!\rangle o = \langle\!\langle \vec{\alpha_1}\alpha\vec{\alpha_2} \setminus^{\vec{\beta_1}\beta\vec{\beta_2}} s \rangle\!\rangle t \cdot \langle\!\langle \vec{\alpha_1}\alpha\vec{\alpha_2} \setminus^{\vec{\beta_1}\beta\vec{\beta_2}} s \rangle\!\rangle s'$. Más aún, $\alpha \notin \mathsf{fn}(o)$ implica $\alpha \notin \mathsf{fn}(t)$ y $\alpha \notin \mathsf{fn}(s')$. Entonces, por *h.i.* se tienen $\langle\!\langle \vec{\alpha_1}\alpha\vec{\alpha_2} \setminus^{\vec{\beta_1}\beta\vec{\beta_2}} s \rangle\!\rangle t = \langle\!\langle \vec{\alpha_1}\vec{\alpha_2} \setminus^{\vec{\beta_1}\vec{\beta_2}} s \rangle\!\rangle t$ y $\langle\!\langle \vec{\alpha_1}\alpha\vec{\alpha_2} \setminus^{\vec{\beta_1}\beta\vec{\beta_2}} s \rangle\!\rangle s' = \langle\!\langle \vec{\alpha_1}\vec{\alpha_2} \setminus^{\vec{\beta_1}\vec{\beta_2}} s \rangle\!\rangle s'$. Luego, se concluye $\langle\!\langle \vec{\alpha_1}\alpha\vec{\alpha_2} \setminus^{\vec{\beta_1}\beta\vec{\beta_2}} s \rangle\!\rangle o = \langle\!\langle \vec{\alpha_1}\vec{\alpha_2} \setminus^{\vec{\beta_1}\vec{\beta_2}} s \rangle\!\rangle o$.

□

Corolario C.0.3. *Sean $o \in \mathbb{O}_{\lambda\mu\rho}$ y s un stack. Si $\vec{\alpha} \cap \mathsf{fn}(o) = \emptyset$, entonces $\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle o = o$.*

Lema C.0.4. *Sean $o \in \mathbb{O}_{\lambda\mu\rho}$, $u \in \mathbb{T}_{\lambda\mu\rho}$ y s un stack.*

1. *Si $x \notin \mathsf{fv}(s)$, entonces $\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle\{x \setminus u\}o = \{x \setminus \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle u\}\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle o$.*
2. *Si $\vec{\alpha} \cap \mathsf{fn}(u) = \emptyset$, entonces $\{x \setminus u\}\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle o = \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} \{x \setminus u\}s \rangle\!\rangle\{x \setminus u\}o$.*

Demostración. 1. Por inducción en o.

- $o = x$. Luego, $\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle\{x \setminus u\}o = \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle u = \{x \setminus \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle u\}\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle o$, por lo que se concluye.
- $o = y \neq x$. El resultado es inmediato pues $\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle\{x \setminus u\}o = y = \{x \setminus \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle u\}\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle o$.
- $o = t\, v$. Luego, se tiene $\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle\{x \setminus u\}o = \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle\{x \setminus u\}t\, \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle\{x \setminus u\}v$. Por *h.i.* se tiene $\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle\{x \setminus u\}t = \{x \setminus \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle u\}\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle t$, mientras que también $\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle\{x \setminus u\}u = \{x \setminus \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle u\}\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle u$, por lo que se concluye $\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle\{x \setminus u\}o = \{x \setminus \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle u\}\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle o$.
- $o = \lambda y.t$ con $y \notin u$ e $y \notin s$. Luego, se tiene $\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle\{x \setminus u\}o = \lambda y.\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle\{x \setminus u\}t$. Por *h.i.* se tiene $\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle\{x \setminus u\}t = \{x \setminus \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle u\}\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle t$, por lo que se concluye $\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle\{x \setminus u\}o = \{x \setminus \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle u\}\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle o$.

- $o = \mu\gamma.c$ con $\gamma \notin u$ y $\gamma \notin s$. Luego, se tiene $\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle \{x \setminus u\} o = \mu\gamma.\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle \{x \setminus u\} c$. Por *h.i.* se tiene $\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle \{x \setminus u\} c = \{x \setminus \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle u\} \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle c$, por lo que se concluye $\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle \{x \setminus u\} o = \{x \setminus \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle u\} \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle o$.
- $o = [\delta]\, t$. Luego, se tienen dos casos posibles:
  - $\delta \in \vec{\alpha}$. Luego, $\delta = \alpha_k$ para algún $k \in [1, \mathsf{sz}(\vec{\alpha})]$. Por *h.i.* se tiene $\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle \{x \setminus u\} t = \{x \setminus \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle u\} \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle t$. Finalmente, se tiene $\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle \{x \setminus u\} o = [\beta_k]\, \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle \{x \setminus u\} t :: s = [\beta_k]\, \{x \setminus \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle u\} \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle t :: s = \{x \setminus \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle u\} \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle o$, por lo que se concluye.
  - $\delta \notin \vec{\alpha}$. Luego, $\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle \{x \setminus u\} o = [\delta]\, \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle \{x \setminus u\} t$. Por *h.i.* se tiene $\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle \{x \setminus u\} t = \{x \setminus \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle u\} \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle t$, por lo que se concluye $\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle \{x \setminus u\} o = \{x \setminus \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle u\} \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle o$.
- $o = c[\gamma \setminus \delta]$ con $\gamma \notin \vec{\alpha}$, $\gamma \notin \vec{\beta}$ y $\gamma \notin s$. Luego, se tienen dos casos posibles:
  - $\delta \in \vec{\alpha}$. Luego, $\delta = \alpha_k$ para algún $k \in [1, \mathsf{sz}(\vec{\alpha})]$. Más aún, sea γ' un nombre fresco. Por *h.i.* se tiene $\langle\!\langle \gamma\vec{\alpha} \setminus^{\gamma'\vec{\beta}} s \rangle\!\rangle \{x \setminus u\} c = \{x \setminus \langle\!\langle \gamma\vec{\alpha} \setminus^{\gamma'\vec{\beta}} s \rangle\!\rangle u\} \langle\!\langle \gamma\vec{\alpha} \setminus^{\gamma'\vec{\beta}} s \rangle\!\rangle c$. Finalmente, se tiene $\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle \{x \setminus u\} o = (\langle\!\langle \gamma\vec{\alpha} \setminus^{\gamma'\vec{\beta}} s \rangle\!\rangle \{x \setminus u\} c)[\gamma' \setminus \beta_k] = (\{x \setminus \langle\!\langle \gamma\vec{\alpha} \setminus^{\gamma'\vec{\beta}} s \rangle\!\rangle u\} \langle\!\langle \gamma\vec{\alpha} \setminus^{\gamma'\vec{\beta}} s \rangle\!\rangle c)[\gamma' \setminus \beta_k] = \{x \setminus \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle u\} \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle o$, por lo que se concluye.
  - $\delta \notin \vec{\alpha}$. Luego, $\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle \{x \setminus u\} o = (\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle \{x \setminus u\} c)[\gamma \setminus \delta]$. Por *h.i.* $\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle \{x \setminus u\} c = \{x \setminus \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle u\} \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle c$, por lo que se concluye $\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle \{x \setminus u\} o = \{x \setminus \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle u\} \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle o$.
- $o = t \cdot s'$. Luego, se tiene $\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle \{x \setminus u\} o = \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle \{x \setminus u\} t \cdot \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle \{x \setminus u\} s'$. Por *h.i.* se tiene $\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle \{x \setminus u\} t = \{x \setminus \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle u\} \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle t$, mientras que también $\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle \{x \setminus u\} u = \{x \setminus \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle u\} \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle u$, por lo que se concluye $\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle \{x \setminus u\} o = \{x \setminus \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle u\} \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle o$.

2. Por inducción en o.

- $o = x$. Luego, $\{x \setminus u\} \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle o = \{x \setminus u\} o = u$. Más aún, por Cor. C.0.3 con $\vec{\alpha} \cap \mathsf{fn}(u) = \emptyset$, se tiene $u = \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} \{x \setminus u\} s \rangle\!\rangle u$ por lo que se concluye.
- $o = y \neq x$. El resultado es inmediato pues $\{x \setminus u\} \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle o = y = \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} \{x \setminus u\} s \rangle\!\rangle \{x \setminus u\} o$.
- $o = t\, v$. Luego, se tiene $\{x \setminus u\} \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle o = \{x \setminus u\} \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle t\, \{x \setminus u\} \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle v$. Por *h.i.* se tiene $\{x \setminus u\} \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle t = \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} \{x \setminus u\} s \rangle\!\rangle \{x \setminus u\} t$, mientras que también $\{x \setminus u\} \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle v = \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} \{x \setminus u\} s \rangle\!\rangle \{x \setminus u\} v$, por lo que se concluye $\{x \setminus u\} \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle o = \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} \{x \setminus u\} s \rangle\!\rangle \{x \setminus u\} o$.
- $o = \lambda y.t$ con $y \notin u$ e $y \notin s$. Luego, se tiene $\{x \setminus u\} \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle o = \lambda y.\{x \setminus u\} \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle t$. Por *h.i.* se tiene $\{x \setminus u\} \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle t = \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} \{x \setminus u\} s \rangle\!\rangle \{x \setminus u\} t$, por lo que se concluye $\{x \setminus u\} \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle o = \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} \{x \setminus u\} s \rangle\!\rangle \{x \setminus u\} o$.
- $o = \mu\gamma.c$ con $\gamma \notin u$ y $\gamma \notin s$. Luego, se tiene $\{x \setminus u\} \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle o = \mu\gamma.\{x \setminus u\} \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle c$. Por *h.i.* se tiene $\{x \setminus u\} \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle c = \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} \{x \setminus u\} s \rangle\!\rangle \{x \setminus u\} c$, por lo que se concluye $\{x \setminus u\} \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle o = \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} \{x \setminus u\} s \rangle\!\rangle \{x \setminus u\} o$.
- $o = [\delta]\, t$. Luego, se tienen dos casos posibles:
  - $\delta \in \vec{\alpha}$. Luego, $\delta = \alpha_k$ para algún $k \in [1, \mathsf{sz}(\vec{\alpha})]$. Por *h.i.* se tiene $\{x \setminus u\} \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle t = \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} \{x \setminus u\} s \rangle\!\rangle \{x \setminus u\} t$. Finalmente, $\{x \setminus u\} \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle o = [\beta_k]\, \{x \setminus u\} \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle t :: \{x \setminus u\} s = [\beta_k]\, \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} \{x \setminus u\} s \rangle\!\rangle \{x \setminus u\} t :: \{x \setminus u\} s = \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} \{x \setminus u\} s \rangle\!\rangle \{x \setminus u\} o$, por lo que se concluye.
  - $\delta \notin \vec{\alpha}$. Luego, $\{x \setminus u\} \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle o = [\delta]\, \{x \setminus u\} \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle t$. Por *h.i.* se tiene $\{x \setminus u\} \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle t = \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} \{x \setminus u\} s \rangle\!\rangle \{x \setminus u\} t$, por lo que se concluye $\{x \setminus u\} \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle o = \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} \{x \setminus u\} s \rangle\!\rangle \{x \setminus u\} o$.
- $o = c[\gamma \setminus \delta]$ con $\gamma \notin \vec{\alpha}$, $\gamma \notin \vec{\beta}$ y $\gamma \notin s$. Luego, se tienen dos casos posibles:

- $\delta \in \vec{\alpha}$. Luego, $\delta = \alpha_k$ para algún $k \in [1, \mathsf{sz}(\vec{\alpha})]$. Más aún, sea γ' un nombre fresco. Por *h.i.* se tiene $\{x \setminus u\}\langle\!\langle \gamma\vec{\alpha} \setminus^{\gamma'\vec{\beta}} s\rangle\!\rangle c = \langle\!\langle \gamma\vec{\alpha} \setminus^{\gamma'\vec{\beta}} \{x \setminus u\}s\rangle\!\rangle \{x \setminus u\}c$. Finalmente, se tiene $\{x \setminus u\}\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s\rangle\!\rangle o = (\{x \setminus u\}\langle\!\langle \gamma\vec{\alpha} \setminus^{\gamma'\vec{\beta}} s\rangle\!\rangle c)[\gamma' \setminus \beta_k] = (\langle\!\langle \gamma\vec{\alpha} \setminus^{\gamma'\vec{\beta}} \{x \setminus u\}s\rangle\!\rangle \{x \setminus u\}c)[\gamma' \setminus \beta_k] = \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} \{x \setminus u\}s\rangle\!\rangle \{x \setminus u\}o$, por lo que se concluye.
- $\delta \notin \vec{\alpha}$. Luego, $\{x \setminus u\}\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s\rangle\!\rangle o = (\{x \setminus u\}\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s\rangle\!\rangle c)[\gamma \setminus \delta]$. Por *h.i.* $\{x \setminus u\}\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s\rangle\!\rangle c = \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} \{x \setminus u\}s\rangle\!\rangle \{x \setminus u\}c$, por lo que se concluye $\{x \setminus u\}\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s\rangle\!\rangle o = \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} \{x \setminus u\}s\rangle\!\rangle \{x \setminus u\}o$.

- $o = t \cdot s'$. Luego, se tiene $\{x \setminus u\}\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s\rangle\!\rangle o = \{x \setminus u\}\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s\rangle\!\rangle t \cdot \{x \setminus u\}\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s\rangle\!\rangle s'$. Por *h.i.* se tiene $\{x \setminus u\}\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s\rangle\!\rangle t = \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} \{x \setminus u\}s\rangle\!\rangle \{x \setminus u\}t$, mientras que también $\{x \setminus u\}\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s\rangle\!\rangle s' = \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} \{x \setminus u\}s\rangle\!\rangle \{x \setminus u\}s'$, por lo que se concluye $\{x \setminus u\}\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s\rangle\!\rangle o = \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} \{x \setminus u\}s\rangle\!\rangle \{x \setminus u\}o$.

□

Lema C.0.5. *Sean $o \in \mathbb{O}_{\lambda\mu\rho}$ y s, s' stacks. Si se tienen secuencias de nombres $\vec{\alpha} = \vec{\alpha_1}\vec{\alpha_2}$, $\vec{\beta} = \vec{\beta_1}\vec{\beta_2}$, $\vec{\gamma} = \vec{\gamma_1}\vec{\gamma_2}$ y $\vec{\delta} = \vec{\delta_1}\vec{\delta_2}$ tales que $\vec{\beta_1} = \vec{\gamma_1}$, $\vec{\beta_2} \cap \vec{\gamma_2} = \emptyset$, $\chi \notin s'$ para todo $\chi \in \vec{\alpha}$, y $\vec{\beta}\vec{\delta}$ son nombres frescos, entonces $\langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s'\rangle\!\rangle\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s\rangle\!\rangle o = \langle\!\langle \vec{\alpha_2} \setminus^{\vec{\beta_2}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s'\rangle\!\rangle s\rangle\!\rangle\langle\!\langle \vec{\alpha_1} \setminus^{\vec{\delta_1}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s'\rangle\!\rangle s \cdot s'\rangle\!\rangle\langle\!\langle \vec{\gamma_2} \setminus^{\vec{\delta_2}} s'\rangle\!\rangle o$.*

Demostración. Por inducción en o.

- $o = x$. Luego, el resultado es inmediato pues, por definición, se tiene $\langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s'\rangle\!\rangle\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s\rangle\!\rangle o = x = \langle\!\langle \vec{\alpha_2} \setminus^{\vec{\beta_2}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s'\rangle\!\rangle s\rangle\!\rangle\langle\!\langle \vec{\alpha_1} \setminus^{\vec{\delta_1}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s'\rangle\!\rangle s \cdot s'\rangle\!\rangle\langle\!\langle \vec{\gamma_2} \setminus^{\vec{\delta_2}} s'\rangle\!\rangle o$.

- $o = t\,u$. Luego, $\langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s'\rangle\!\rangle\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s\rangle\!\rangle o = \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s'\rangle\!\rangle\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s\rangle\!\rangle t\,\langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s'\rangle\!\rangle\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s\rangle\!\rangle u$. Por *h.i.* se tiene $\langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s'\rangle\!\rangle\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s\rangle\!\rangle t = \langle\!\langle \vec{\alpha_2} \setminus^{\vec{\beta_2}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s'\rangle\!\rangle s\rangle\!\rangle\langle\!\langle \vec{\alpha_1} \setminus^{\vec{\delta_1}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s'\rangle\!\rangle s \cdot s'\rangle\!\rangle\langle\!\langle \vec{\gamma_2} \setminus^{\vec{\delta_2}} s'\rangle\!\rangle t$, mientras que también $\langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s'\rangle\!\rangle\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s\rangle\!\rangle u = \langle\!\langle \vec{\alpha_2} \setminus^{\vec{\beta_2}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s'\rangle\!\rangle s\rangle\!\rangle\langle\!\langle \vec{\alpha_1} \setminus^{\vec{\delta_1}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s'\rangle\!\rangle s \cdot s'\rangle\!\rangle\langle\!\langle \vec{\gamma_2} \setminus^{\vec{\delta_2}} s'\rangle\!\rangle u$, por lo que se concluye $\langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s'\rangle\!\rangle\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s\rangle\!\rangle o = \langle\!\langle \vec{\alpha_2} \setminus^{\vec{\beta_2}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s'\rangle\!\rangle s\rangle\!\rangle\langle\!\langle \vec{\alpha_1} \setminus^{\vec{\delta_1}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s'\rangle\!\rangle s \cdot s'\rangle\!\rangle\langle\!\langle \vec{\gamma_2} \setminus^{\vec{\delta_2}} s'\rangle\!\rangle o$.

- $o = \lambda x.t$ con $x \notin s, s'$. Luego, $\langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s'\rangle\!\rangle\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s\rangle\!\rangle o = \lambda x.\langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s'\rangle\!\rangle\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s\rangle\!\rangle t$. Por *h.i.* se tiene $\langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s'\rangle\!\rangle\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s\rangle\!\rangle t = \langle\!\langle \vec{\alpha_2} \setminus^{\vec{\beta_2}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s'\rangle\!\rangle s\rangle\!\rangle\langle\!\langle \vec{\alpha_1} \setminus^{\vec{\delta_1}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s'\rangle\!\rangle s \cdot s'\rangle\!\rangle\langle\!\langle \vec{\gamma_2} \setminus^{\vec{\delta_2}} s'\rangle\!\rangle t$, por lo que se concluye $\langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s'\rangle\!\rangle\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s\rangle\!\rangle o = \langle\!\langle \vec{\alpha_2} \setminus^{\vec{\beta_2}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s'\rangle\!\rangle s\rangle\!\rangle\langle\!\langle \vec{\alpha_1} \setminus^{\vec{\delta_1}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s'\rangle\!\rangle s \cdot s'\rangle\!\rangle\langle\!\langle \vec{\gamma_2} \setminus^{\vec{\delta_2}} s'\rangle\!\rangle o$.

- $o = \mu\xi.c$ con $\xi \notin \vec{\alpha}\vec{\beta}\vec{\gamma}\vec{\delta}$ y $\xi \notin s, s'$. Luego, $\langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s'\rangle\!\rangle\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s\rangle\!\rangle o = \mu\xi.\langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s'\rangle\!\rangle\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s\rangle\!\rangle c$. Por *h.i.* se tiene $\langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s'\rangle\!\rangle\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s\rangle\!\rangle c = \langle\!\langle \vec{\alpha_2} \setminus^{\vec{\beta_2}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s'\rangle\!\rangle s\rangle\!\rangle\langle\!\langle \vec{\alpha_1} \setminus^{\vec{\delta_1}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s'\rangle\!\rangle s \cdot s'\rangle\!\rangle\langle\!\langle \vec{\gamma_2} \setminus^{\vec{\delta_2}} s'\rangle\!\rangle c$, por lo que se concluye $\langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s'\rangle\!\rangle\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s\rangle\!\rangle o = \langle\!\langle \vec{\alpha_2} \setminus^{\vec{\beta_2}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s'\rangle\!\rangle s\rangle\!\rangle\langle\!\langle \vec{\alpha_1} \setminus^{\vec{\delta_1}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s'\rangle\!\rangle s \cdot s'\rangle\!\rangle\langle\!\langle \vec{\gamma_2} \setminus^{\vec{\delta_2}} s'\rangle\!\rangle o$.

- $o = [\eta]\,t$. Se tienen cuatro casos posibles:

  1. $\eta \in \vec{\alpha_1}$. Luego, $\eta = \alpha_k$ para algún $k \in [1, \mathsf{sz}(\vec{\alpha_1})]$. Más aún, por hipótesis, $\beta_k = \gamma_k$. Además, por *h.i.* $\langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s'\rangle\!\rangle\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s\rangle\!\rangle t = \langle\!\langle \vec{\alpha_2} \setminus^{\vec{\beta_2}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s'\rangle\!\rangle s\rangle\!\rangle\langle\!\langle \vec{\alpha_1} \setminus^{\vec{\delta_1}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s'\rangle\!\rangle s \cdot s'\rangle\!\rangle\langle\!\langle \vec{\gamma_2} \setminus^{\vec{\delta_2}} s'\rangle\!\rangle t$. Entonces, se concluye aplicando la definición de replacement simultaneo:

$$
\begin{aligned}
\langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s'\rangle\!\rangle\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s\rangle\!\rangle o &= \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s'\rangle\!\rangle\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s\rangle\!\rangle [\alpha_k]\,t \\
&= \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s'\rangle\!\rangle([\beta_k]\,\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s\rangle\!\rangle t :: s) \\
&= [\delta_k]\,\langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s'\rangle\!\rangle\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s\rangle\!\rangle t :: \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s'\rangle\!\rangle s :: s' \\
&= [\delta_k]\,\langle\!\langle \vec{\alpha_2} \setminus^{\vec{\beta_2}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s'\rangle\!\rangle s\rangle\!\rangle\langle\!\langle \vec{\alpha_1} \setminus^{\vec{\delta_1}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s'\rangle\!\rangle s \cdot s'\rangle\!\rangle\langle\!\langle \vec{\gamma_2} \setminus^{\vec{\delta_2}} s'\rangle\!\rangle t :: \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s'\rangle\!\rangle s :: s' \\
&= \langle\!\langle \vec{\alpha_2} \setminus^{\vec{\beta_2}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s'\rangle\!\rangle s\rangle\!\rangle([\delta_k]\,\langle\!\langle \vec{\alpha_1} \setminus^{\vec{\delta_1}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s'\rangle\!\rangle s \cdot s'\rangle\!\rangle\langle\!\langle \vec{\gamma_2} \setminus^{\vec{\delta_2}} s'\rangle\!\rangle t :: \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s'\rangle\!\rangle s :: s') \\
&= \langle\!\langle \vec{\alpha_2} \setminus^{\vec{\beta_2}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s'\rangle\!\rangle s\rangle\!\rangle([\delta_k]\,\langle\!\langle \vec{\alpha_1} \setminus^{\vec{\delta_1}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s'\rangle\!\rangle s \cdot s'\rangle\!\rangle\langle\!\langle \vec{\gamma_2} \setminus^{\vec{\delta_2}} s'\rangle\!\rangle t :: (\langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s'\rangle\!\rangle s \cdot s')) \\
&= \langle\!\langle \vec{\alpha_2} \setminus^{\vec{\beta_2}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s'\rangle\!\rangle s\rangle\!\rangle\langle\!\langle \vec{\alpha_1} \setminus^{\vec{\delta_1}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s'\rangle\!\rangle s \cdot s'\rangle\!\rangle([\alpha_k]\,\langle\!\langle \vec{\gamma_2} \setminus^{\vec{\delta_2}} s'\rangle\!\rangle t) \\
&= \langle\!\langle \vec{\alpha_2} \setminus^{\vec{\beta_2}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s'\rangle\!\rangle s\rangle\!\rangle\langle\!\langle \vec{\alpha_1} \setminus^{\vec{\delta_1}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s'\rangle\!\rangle s \cdot s'\rangle\!\rangle\langle\!\langle \vec{\gamma_2} \setminus^{\vec{\delta_2}} s'\rangle\!\rangle([\alpha_k]\,t) \\
&= \langle\!\langle \vec{\alpha_2} \setminus^{\vec{\beta_2}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s'\rangle\!\rangle s\rangle\!\rangle\langle\!\langle \vec{\alpha_1} \setminus^{\vec{\delta_1}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s'\rangle\!\rangle s \cdot s'\rangle\!\rangle\langle\!\langle \vec{\gamma_2} \setminus^{\vec{\delta_2}} s'\rangle\!\rangle o
\end{aligned}
$$

2. $\eta \in \vec{\alpha_2}$. Luego, $\eta = \alpha_k$ para algún $k \in [1, \mathsf{sz}(\vec{\alpha_2})]$. Además, por *h.i.* $\langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle t = \langle\!\langle \vec{\alpha_2} \setminus^{\vec{\beta_2}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle s \rangle\!\rangle \langle\!\langle \vec{\alpha_1} \setminus^{\vec{\delta_1}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle s \cdot s' \rangle\!\rangle \langle\!\langle \vec{\gamma_2} \setminus^{\vec{\delta_2}} s' \rangle\!\rangle t$. Entonces, se concluye aplicando la definición de replacement simultaneo:

$$\begin{array}{rcl}
\langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle o & = & \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle [\alpha_k]\, t \\
& = & \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle ([\beta_k] \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle t :: s) \\
& = & [\beta_k] \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle t :: \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle s \\
& = & [\beta_k] \langle\!\langle \vec{\alpha_2} \setminus^{\vec{\beta_2}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle s \rangle\!\rangle \langle\!\langle \vec{\alpha_1} \setminus^{\vec{\delta_1}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle s \cdot s' \rangle\!\rangle \langle\!\langle \vec{\gamma_2} \setminus^{\vec{\delta_2}} s' \rangle\!\rangle t :: \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle s \\
& = & \langle\!\langle \vec{\alpha_2} \setminus^{\vec{\beta_2}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle s \rangle\!\rangle ([\alpha_k] \langle\!\langle \vec{\alpha_1} \setminus^{\vec{\delta_1}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle s \cdot s' \rangle\!\rangle \langle\!\langle \vec{\gamma_2} \setminus^{\vec{\delta_2}} s' \rangle\!\rangle t :: \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle s) \\
& = & \langle\!\langle \vec{\alpha_2} \setminus^{\vec{\beta_2}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle s \rangle\!\rangle \langle\!\langle \vec{\alpha_1} \setminus^{\vec{\delta_1}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle s \cdot s' \rangle\!\rangle ([\alpha_k] \langle\!\langle \vec{\gamma_2} \setminus^{\vec{\delta_2}} s' \rangle\!\rangle t) \\
& = & \langle\!\langle \vec{\alpha_2} \setminus^{\vec{\beta_2}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle s \rangle\!\rangle \langle\!\langle \vec{\alpha_1} \setminus^{\vec{\delta_1}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle s \cdot s' \rangle\!\rangle \langle\!\langle \vec{\gamma_2} \setminus^{\vec{\delta_2}} s' \rangle\!\rangle ([\alpha_k]\, t) \\
& = & \langle\!\langle \vec{\alpha_2} \setminus^{\vec{\beta_2}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle s \rangle\!\rangle \langle\!\langle \vec{\alpha_1} \setminus^{\vec{\delta_1}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle s \cdot s' \rangle\!\rangle \langle\!\langle \vec{\gamma_2} \setminus^{\vec{\delta_2}} s' \rangle\!\rangle o
\end{array}$$

3. $\eta \in \vec{\gamma}$. Luego, necesariamente $\eta \in \vec{\gamma_2}$, pues $\vec{\gamma_1} = \vec{\beta_1}$ y $\vec{\beta}$ contiene nombre frescos por hipótesis. Entonces, $\eta = \gamma_k$ para algún $k \in [1, \mathsf{sz}(\vec{\gamma_2})]$. Por *h.i.* se tiene $\langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle t = \langle\!\langle \vec{\alpha_2} \setminus^{\vec{\beta_2}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle s \rangle\!\rangle \langle\!\langle \vec{\alpha_1} \setminus^{\vec{\delta_1}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle s \cdot s' \rangle\!\rangle \langle\!\langle \vec{\gamma_2} \setminus^{\vec{\delta_2}} s' \rangle\!\rangle t$. Entonces, se concluye aplicando la definición de replacement simultaneo:

$$\begin{array}{rcl}
\langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle o & = & \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle [\gamma_k]\, t \\
& = & \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle ([\gamma_k] \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle t) \\
& = & [\delta_k] \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle t :: s' \\
& = & [\delta_k] \langle\!\langle \vec{\alpha_2} \setminus^{\vec{\beta_2}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle s \rangle\!\rangle \langle\!\langle \vec{\alpha_1} \setminus^{\vec{\delta_1}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle s \cdot s' \rangle\!\rangle \langle\!\langle \vec{\gamma_2} \setminus^{\vec{\delta_2}} s' \rangle\!\rangle t :: s' \\
& = & \langle\!\langle \vec{\alpha_2} \setminus^{\vec{\beta_2}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle s \rangle\!\rangle ([\delta_k] \langle\!\langle \vec{\alpha_1} \setminus^{\vec{\delta_1}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle s \cdot s' \rangle\!\rangle \langle\!\langle \vec{\gamma_2} \setminus^{\vec{\delta_2}} s' \rangle\!\rangle t :: s') \\
& = & \langle\!\langle \vec{\alpha_2} \setminus^{\vec{\beta_2}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle s \rangle\!\rangle \langle\!\langle \vec{\alpha_1} \setminus^{\vec{\delta_1}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle s \cdot s' \rangle\!\rangle ([\delta_k] \langle\!\langle \vec{\gamma_2} \setminus^{\vec{\delta_2}} s' \rangle\!\rangle t :: s') \\
& = & \langle\!\langle \vec{\alpha_2} \setminus^{\vec{\beta_2}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle s \rangle\!\rangle \langle\!\langle \vec{\alpha_1} \setminus^{\vec{\delta_1}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle s \cdot s' \rangle\!\rangle \langle\!\langle \vec{\gamma_2} \setminus^{\vec{\delta_2}} s' \rangle\!\rangle ([\gamma_k]\, t) \\
& = & \langle\!\langle \vec{\alpha_2} \setminus^{\vec{\beta_2}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle s \rangle\!\rangle \langle\!\langle \vec{\alpha_1} \setminus^{\vec{\delta_1}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle s \cdot s' \rangle\!\rangle \langle\!\langle \vec{\gamma_2} \setminus^{\vec{\delta_2}} s' \rangle\!\rangle o
\end{array}$$

4. $\eta \notin \vec{\alpha}\vec{\gamma}$. Luego, por definición se tiene $\langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle o = [\eta] \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle t$. Por *h.i.* se tiene $\langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle t = \langle\!\langle \vec{\alpha_2} \setminus^{\vec{\beta_2}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle s \rangle\!\rangle \langle\!\langle \vec{\alpha_1} \setminus^{\vec{\delta_1}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle s \cdot s' \rangle\!\rangle \langle\!\langle \vec{\gamma_2} \setminus^{\vec{\delta_2}} s' \rangle\!\rangle t$, por lo que se concluye $\langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle o = \langle\!\langle \vec{\alpha_2} \setminus^{\vec{\beta_2}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle s \rangle\!\rangle \langle\!\langle \vec{\alpha_1} \setminus^{\vec{\delta_1}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle s \cdot s' \rangle\!\rangle \langle\!\langle \vec{\gamma_2} \setminus^{\vec{\delta_2}} s' \rangle\!\rangle o$.

- $o = c[\xi \setminus \eta]$. con $\xi \notin \vec{\alpha}\vec{\beta}\vec{\gamma}\vec{\delta}$ y $\xi \notin s, s'$. Se tienen cuatro casos posibles:

1. $\eta \in \vec{\alpha_1}$. Luego, $\eta = \alpha_k$ para algún $k \in [1, \mathsf{sz}(\vec{\alpha_1})]$. Más aún, por hipótesis, $\beta_k = \gamma_k$. Por *h.i.* $\langle\!\langle \xi'\vec{\gamma} \setminus^{\xi''\vec{\delta}} s' \rangle\!\rangle \langle\!\langle \xi\vec{\alpha} \setminus^{\xi'\vec{\beta}} s \rangle\!\rangle c = \langle\!\langle \vec{\alpha_2} \setminus^{\vec{\beta_2}} \langle\!\langle \xi'\vec{\gamma} \setminus^{\xi''\vec{\delta}} s' \rangle\!\rangle s \rangle\!\rangle \langle\!\langle \xi\vec{\alpha_1} \setminus^{\xi''\vec{\delta_1}} \langle\!\langle \xi'\vec{\gamma} \setminus^{\xi''\vec{\delta}} s' \rangle\!\rangle s \cdot s' \rangle\!\rangle \langle\!\langle \vec{\gamma_2} \setminus^{\vec{\delta_2}} s' \rangle\!\rangle c$ donde ξ' y ξ'' son nombres frescos. Entonces, se concluye apelando al Lem. C.0.2 para eliminar nombres espurios del replacement simultaneo:

$$\begin{array}{rcl}
\langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle o & = & \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle (c[\xi \setminus \alpha_k]) \\
& = & \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle ((\langle\!\langle \xi\vec{\alpha} \setminus^{\xi'\vec{\beta}} s \rangle\!\rangle c)[\xi' \setminus \beta_k]) \\
& = & (\langle\!\langle \xi'\vec{\gamma} \setminus^{\xi''\vec{\delta}} s' \rangle\!\rangle \langle\!\langle \xi\vec{\alpha} \setminus^{\xi'\vec{\beta}} s \rangle\!\rangle c)[\xi'' \setminus \beta_k] \\
& = & (\langle\!\langle \vec{\alpha_2} \setminus^{\vec{\beta_2}} \langle\!\langle \xi'\vec{\gamma} \setminus^{\xi''\vec{\delta}} s' \rangle\!\rangle s \rangle\!\rangle \langle\!\langle \xi\vec{\alpha_1} \setminus^{\xi''\vec{\delta_1}} \langle\!\langle \xi'\vec{\gamma} \setminus^{\xi''\vec{\delta}} s' \rangle\!\rangle s \cdot s' \rangle\!\rangle \langle\!\langle \vec{\gamma_2} \setminus^{\vec{\delta_2}} s' \rangle\!\rangle c)[\xi'' \setminus \beta_k] \\
& = & (\langle\!\langle \vec{\alpha_2} \setminus^{\vec{\beta_2}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle s \rangle\!\rangle \langle\!\langle \xi\vec{\alpha_1} \setminus^{\xi''\vec{\delta_1}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle s \cdot s' \rangle\!\rangle \langle\!\langle \vec{\gamma_2} \setminus^{\vec{\delta_2}} s' \rangle\!\rangle c)[\xi'' \setminus \beta_k] \\
& = & \langle\!\langle \vec{\alpha_2} \setminus^{\vec{\beta_2}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle s \rangle\!\rangle ((\langle\!\langle \xi\vec{\alpha_1} \setminus^{\xi''\vec{\delta_1}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle s \cdot s' \rangle\!\rangle \langle\!\langle \vec{\gamma_2} \setminus^{\vec{\delta_2}} s' \rangle\!\rangle c)[\xi'' \setminus \beta_k]) \\
& = & \langle\!\langle \vec{\alpha_2} \setminus^{\vec{\beta_2}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle s \rangle\!\rangle \langle\!\langle \vec{\alpha_1} \setminus^{\vec{\delta_1}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle s \cdot s' \rangle\!\rangle ((\langle\!\langle \vec{\gamma_2} \setminus^{\vec{\delta_2}} s' \rangle\!\rangle c)[\xi \setminus \alpha_k]) \\
& = & \langle\!\langle \vec{\alpha_2} \setminus^{\vec{\beta_2}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle s \rangle\!\rangle \langle\!\langle \vec{\alpha_1} \setminus^{\vec{\delta_1}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle s \cdot s' \rangle\!\rangle \langle\!\langle \vec{\gamma_2} \setminus^{\vec{\delta_2}} s' \rangle\!\rangle (c[\xi \setminus \alpha_k]) \\
& = & \langle\!\langle \vec{\alpha_2} \setminus^{\vec{\beta_2}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle s \rangle\!\rangle \langle\!\langle \vec{\alpha_1} \setminus^{\vec{\delta_1}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle s \cdot s' \rangle\!\rangle \langle\!\langle \vec{\gamma_2} \setminus^{\vec{\delta_2}} s' \rangle\!\rangle o
\end{array}$$

2. $\eta \in \vec{\alpha_2}$. Luego, $\eta = \alpha_k$ para algún $k \in [1, \mathsf{sz}(\vec{\alpha_1})]$. Además, por *h.i.* $\langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle \langle\!\langle \vec{\alpha}\xi \setminus^{\vec{\beta}\xi'} s \rangle\!\rangle c = \langle\!\langle \vec{\alpha_2}\xi \setminus^{\vec{\beta_2}\xi'} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle s \rangle\!\rangle \langle\!\langle \vec{\alpha_1} \setminus^{\vec{\delta_1}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle s \cdot s' \rangle\!\rangle \langle\!\langle \vec{\gamma_2} \setminus^{\vec{\delta_2}} s' \rangle\!\rangle c$ donde ξ' es un nombre fresco. Entonces, se concluye apelando al Lem. C.0.1 para permutar consistentemente nombres en el replacement simultaneo:

$$\begin{aligned}
\langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle o &= \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle (c[\xi \setminus \alpha_k]) \\
&= \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle ((\langle\!\langle \xi\vec{\alpha} \setminus^{\xi'\vec{\beta}} s \rangle\!\rangle c)[\xi' \setminus \beta_k]) \\
&= \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle ((\langle\!\langle \vec{\alpha}\xi \setminus^{\vec{\beta}\xi'} s \rangle\!\rangle c)[\xi' \setminus \beta_k]) \\
&= (\langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle \langle\!\langle \vec{\alpha}\xi \setminus^{\vec{\beta}\xi'} s \rangle\!\rangle c)[\xi' \setminus \beta_k] \\
&= (\langle\!\langle \vec{\alpha_2}\xi \setminus^{\vec{\beta_2}\xi'} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle s \rangle\!\rangle \langle\!\langle \vec{\alpha_1} \setminus^{\vec{\delta_1}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle s \cdot s' \rangle\!\rangle \langle\!\langle \vec{\gamma_2} \setminus^{\vec{\delta_2}} s' \rangle\!\rangle c)[\xi' \setminus \beta_k] \\
&= (\langle\!\langle \xi\vec{\alpha_2} \setminus^{\xi'\vec{\beta_2}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle s \rangle\!\rangle \langle\!\langle \vec{\alpha_1} \setminus^{\vec{\delta_1}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle s \cdot s' \rangle\!\rangle \langle\!\langle \vec{\gamma_2} \setminus^{\vec{\delta_2}} s' \rangle\!\rangle c)[\xi' \setminus \beta_k] \\
&= \langle\!\langle \vec{\alpha_2} \setminus^{\vec{\beta_2}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle s \rangle\!\rangle ((\langle\!\langle \vec{\alpha_1} \setminus^{\vec{\delta_1}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle s \cdot s' \rangle\!\rangle \langle\!\langle \vec{\gamma_2} \setminus^{\vec{\delta_2}} s' \rangle\!\rangle c)[\xi \setminus \alpha_k]) \\
&= \langle\!\langle \vec{\alpha_2} \setminus^{\vec{\beta_2}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle s \rangle\!\rangle \langle\!\langle \vec{\alpha_1} \setminus^{\vec{\delta_1}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle s \cdot s' \rangle\!\rangle ((\langle\!\langle \vec{\gamma_2} \setminus^{\vec{\delta_2}} s' \rangle\!\rangle c)[\xi \setminus \alpha_k]) \\
&= \langle\!\langle \vec{\alpha_2} \setminus^{\vec{\beta_2}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle s \rangle\!\rangle \langle\!\langle \vec{\alpha_1} \setminus^{\vec{\delta_1}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle s \cdot s' \rangle\!\rangle \langle\!\langle \vec{\gamma_2} \setminus^{\vec{\delta_2}} s' \rangle\!\rangle (c[\xi \setminus \alpha_k]) \\
&= \langle\!\langle \vec{\alpha_2} \setminus^{\vec{\beta_2}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle s \rangle\!\rangle \langle\!\langle \vec{\alpha_1} \setminus^{\vec{\delta_1}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle s \cdot s' \rangle\!\rangle \langle\!\langle \vec{\gamma_2} \setminus^{\vec{\delta_2}} s' \rangle\!\rangle o
\end{aligned}$$

3. $\eta \in \vec{\gamma}$. Luego, necesariamente $\eta \in \vec{\gamma_2}$, pues $\vec{\gamma_1} = \vec{\beta_1}$ y $\vec{\beta}$ contiene nombre frescos por hipótesis. Entonces, $\eta = \gamma_k$ para algún $k \in [1, \mathsf{sz}(\vec{\gamma_2})]$. Por *h.i.* se tiene $\langle\!\langle \vec{\gamma}\xi \setminus^{\vec{\delta}\xi'} s' \rangle\!\rangle \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle c = \langle\!\langle \vec{\alpha_2} \setminus^{\vec{\beta_2}} \langle\!\langle \vec{\gamma}\xi \setminus^{\vec{\delta}\xi'} s' \rangle\!\rangle s \rangle\!\rangle \langle\!\langle \vec{\alpha_1} \setminus^{\vec{\delta_1}} \langle\!\langle \vec{\gamma}\xi \setminus^{\vec{\delta}\xi'} s' \rangle\!\rangle s \cdot s' \rangle\!\rangle \langle\!\langle \vec{\gamma_2}\xi \setminus^{\vec{\delta_2}\xi'} s' \rangle\!\rangle c$ donde ξ' es un nombre fresco. Entonces, se concluye apelando al Lem. C.0.1 para permutar consistentemente nombres en el replacement simultaneo, y al Lem. C.0.2 para eliminar nombres espurios para la operación:

$$\begin{aligned}
\langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle o &= \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle (c[\xi \setminus \gamma_k]) \\
&= \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle ((\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle c)[\xi \setminus \gamma_k]) \\
&= (\langle\!\langle \xi\vec{\gamma} \setminus^{\xi'\vec{\delta}} s' \rangle\!\rangle \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle c)[\xi' \setminus \delta_k] \\
&= (\langle\!\langle \vec{\gamma}\xi \setminus^{\vec{\delta}\xi'} s' \rangle\!\rangle \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle c)[\xi' \setminus \delta_k] \\
&= (\langle\!\langle \vec{\alpha_2} \setminus^{\vec{\beta_2}} \langle\!\langle \vec{\gamma}\xi \setminus^{\vec{\delta}\xi'} s' \rangle\!\rangle s \rangle\!\rangle \langle\!\langle \vec{\alpha_1} \setminus^{\vec{\delta_1}} \langle\!\langle \vec{\gamma}\xi \setminus^{\vec{\delta}\xi'} s' \rangle\!\rangle s \cdot s' \rangle\!\rangle \langle\!\langle \vec{\gamma_2}\xi \setminus^{\vec{\delta_2}\xi'} s' \rangle\!\rangle c)[\xi' \setminus \delta_k] \\
&= (\langle\!\langle \vec{\alpha_2} \setminus^{\vec{\beta_2}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle s \rangle\!\rangle \langle\!\langle \vec{\alpha_1} \setminus^{\vec{\delta_1}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle s \cdot s' \rangle\!\rangle \langle\!\langle \vec{\gamma_2}\xi \setminus^{\vec{\delta_2}\xi'} s' \rangle\!\rangle c)[\xi' \setminus \delta_k] \\
&= \langle\!\langle \vec{\alpha_2} \setminus^{\vec{\beta_2}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle s \rangle\!\rangle ((\langle\!\langle \vec{\alpha_1} \setminus^{\vec{\delta_1}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle s \cdot s' \rangle\!\rangle \langle\!\langle \vec{\gamma_2}\xi \setminus^{\vec{\delta_2}\xi'} s' \rangle\!\rangle c)[\xi' \setminus \delta_k]) \\
&= \langle\!\langle \vec{\alpha_2} \setminus^{\vec{\beta_2}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle s \rangle\!\rangle \langle\!\langle \vec{\alpha_1} \setminus^{\vec{\delta_1}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle s \cdot s' \rangle\!\rangle ((\langle\!\langle \vec{\gamma_2}\xi \setminus^{\vec{\delta_2}\xi'} s' \rangle\!\rangle c)[\xi' \setminus \delta_k]) \\
&= \langle\!\langle \vec{\alpha_2} \setminus^{\vec{\beta_2}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle s \rangle\!\rangle \langle\!\langle \vec{\alpha_1} \setminus^{\vec{\delta_1}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle s \cdot s' \rangle\!\rangle ((\langle\!\langle \xi\vec{\gamma_2} \setminus^{\xi'\vec{\delta_2}} s' \rangle\!\rangle c)[\xi' \setminus \delta_k]) \\
&= \langle\!\langle \vec{\alpha_2} \setminus^{\vec{\beta_2}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle s \rangle\!\rangle \langle\!\langle \vec{\alpha_1} \setminus^{\vec{\delta_1}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle s \cdot s' \rangle\!\rangle \langle\!\langle \vec{\gamma_2} \setminus^{\vec{\delta_2}} s' \rangle\!\rangle (c[\xi \setminus \gamma_k]) \\
&= \langle\!\langle \vec{\alpha_2} \setminus^{\vec{\beta_2}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle s \rangle\!\rangle \langle\!\langle \vec{\alpha_1} \setminus^{\vec{\delta_1}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle s \cdot s' \rangle\!\rangle \langle\!\langle \vec{\gamma_2} \setminus^{\vec{\delta_2}} s' \rangle\!\rangle o
\end{aligned}$$

4. $\eta \notin \vec{\alpha}\vec{\gamma}$. Luego, por definición se tiene $\langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle o = (\langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle c)[\xi \setminus \eta]$. Por *h.i.* se tiene $\langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle c = \langle\!\langle \vec{\alpha_2} \setminus^{\vec{\beta_2}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle s \rangle\!\rangle \langle\!\langle \vec{\alpha_1} \setminus^{\vec{\delta_1}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle s \cdot s' \rangle\!\rangle \langle\!\langle \vec{\gamma_2} \setminus^{\vec{\delta_2}} s' \rangle\!\rangle c$, por lo que se concluye $\langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle o = \langle\!\langle \vec{\alpha_2} \setminus^{\vec{\beta_2}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle s \rangle\!\rangle \langle\!\langle \vec{\alpha_1} \setminus^{\vec{\delta_1}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle s \cdot s' \rangle\!\rangle \langle\!\langle \vec{\gamma_2} \setminus^{\vec{\delta_2}} s' \rangle\!\rangle o$.

- $o = t \cdot s''$. Luego, $\langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle o = \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle t \cdot \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle s''$. Por *h.i.* se tiene $\langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle t = \langle\!\langle \vec{\alpha_2} \setminus^{\vec{\beta_2}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle s \rangle\!\rangle \langle\!\langle \vec{\alpha_1} \setminus^{\vec{\delta_1}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle s \cdot s' \rangle\!\rangle \langle\!\langle \vec{\gamma_2} \setminus^{\vec{\delta_2}} s' \rangle\!\rangle t$, mientras que también $\langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle s'' = \langle\!\langle \vec{\alpha_2} \setminus^{\vec{\beta_2}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle s \rangle\!\rangle \langle\!\langle \vec{\alpha_1} \setminus^{\vec{\delta_1}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle s \cdot s' \rangle\!\rangle \langle\!\langle \vec{\gamma_2} \setminus^{\vec{\delta_2}} s' \rangle\!\rangle s''$, por lo que se concluye $\langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle o = \langle\!\langle \vec{\alpha_2} \setminus^{\vec{\beta_2}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle s \rangle\!\rangle \langle\!\langle \vec{\alpha_1} \setminus^{\vec{\delta_1}} \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s' \rangle\!\rangle s \cdot s' \rangle\!\rangle \langle\!\langle \vec{\gamma_2} \setminus^{\vec{\delta_2}} s' \rangle\!\rangle o$.

□

Lema C.0.6. *Sean $o \in \mathbb{O}_{\lambda\mu\rho}$ y s stacks. Si se tienen secuencias de nombres $\vec{\alpha}$, $\vec{\beta}$, $\vec{\gamma}$ y $\vec{\delta}$ tales que $\vec{\beta} \cap \vec{\gamma} = \emptyset$ y $\chi \notin s$ para todo $\chi \in \vec{\gamma}$, entonces $\langle\vec{\gamma} \setminus^{\vec{\delta}} s\rangle\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle o = \langle\vec{\gamma}\vec{\alpha} \setminus^{\vec{\delta}\vec{\beta}} s\rangle o$.*

Demostración. Por inducción en o.

- $o = x$. Luego, el resultado es inmediato pues $\langle\vec{\gamma} \setminus^{\vec{\delta}} s\rangle\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle o = x = \langle\vec{\gamma}\vec{\alpha} \setminus^{\vec{\delta}\vec{\beta}} s\rangle o$.
- $o = t\,u$. Luego, se tiene $\langle\vec{\gamma} \setminus^{\vec{\delta}} s\rangle\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle o = \langle\vec{\gamma} \setminus^{\vec{\delta}} s\rangle\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle t\,\langle\vec{\gamma} \setminus^{\vec{\delta}} s\rangle\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle u$. Por *h.i.* se tienen $\langle\vec{\gamma} \setminus^{\vec{\delta}} s\rangle\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle t = \langle\vec{\gamma}\vec{\alpha} \setminus^{\vec{\delta}\vec{\beta}} s\rangle t$ y $\langle\vec{\gamma} \setminus^{\vec{\delta}} s\rangle\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle u = \langle\vec{\gamma}\vec{\alpha} \setminus^{\vec{\delta}\vec{\beta}} s\rangle u$, por lo que se concluye $\langle\vec{\gamma} \setminus^{\vec{\delta}} s\rangle\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle o = \langle\vec{\gamma}\vec{\alpha} \setminus^{\vec{\delta}\vec{\beta}} s\rangle t\,\langle\vec{\gamma}\vec{\alpha} \setminus^{\vec{\delta}\vec{\beta}} s\rangle u = \langle\vec{\gamma}\vec{\alpha} \setminus^{\vec{\delta}\vec{\beta}} s\rangle o$.
- $o = \lambda x.t$ con $x \notin s$. Luego, se tiene $\langle\vec{\gamma} \setminus^{\vec{\delta}} s\rangle\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle o = \lambda x.\langle\vec{\gamma} \setminus^{\vec{\delta}} s\rangle\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle t$. Por *h.i.* se tiene $\langle\vec{\gamma} \setminus^{\vec{\delta}} s\rangle\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle t = \langle\vec{\gamma}\vec{\alpha} \setminus^{\vec{\delta}\vec{\beta}} s\rangle t$, por lo que se concluye $\langle\vec{\gamma} \setminus^{\vec{\delta}} s\rangle\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle o = \lambda x.\langle\vec{\gamma}\vec{\alpha} \setminus^{\vec{\delta}\vec{\beta}} s\rangle t = \langle\vec{\gamma}\vec{\alpha} \setminus^{\vec{\delta}\vec{\beta}} s\rangle o$.
- $o = \mu\xi.c$ con $\xi \notin \vec{\alpha}\vec{\beta}\vec{\gamma}\vec{\delta}$ y $\xi \notin s$. Luego, $\langle\vec{\gamma} \setminus^{\vec{\delta}} s\rangle\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle o = \mu\xi.\langle\vec{\gamma} \setminus^{\vec{\delta}} s\rangle\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle c$. Por *h.i.* se tiene $\langle\vec{\gamma} \setminus^{\vec{\delta}} s\rangle\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle c = \langle\vec{\gamma}\vec{\alpha} \setminus^{\vec{\delta}\vec{\beta}} s\rangle c$, por lo que se concluye $\langle\vec{\gamma} \setminus^{\vec{\delta}} s\rangle\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle o = \mu\xi.\langle\vec{\gamma}\vec{\alpha} \setminus^{\vec{\delta}\vec{\beta}} s\rangle c = \langle\vec{\gamma}\vec{\alpha} \setminus^{\vec{\delta}\vec{\beta}} s\rangle o$.
- $o = [\eta]\,t$. Se tienen tres casos posibles:

  1. $\eta \in \vec{\alpha}$. Luego, $\eta = \alpha_k$ para algún $k \in [1, \mathsf{sz}(\vec{\alpha})]$. Por *h.i.* se tiene $\langle\vec{\gamma} \setminus^{\vec{\delta}} s\rangle\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle t = \langle\vec{\gamma}\vec{\alpha} \setminus^{\vec{\delta}\vec{\beta}} s\rangle t$. Más aún, por hipótesis, $\beta_k \notin \vec{\gamma}$ y $\chi \notin s$ para todo $\chi \in \vec{\gamma}$. Entonces, se concluye apelando al Cor. C.0.3:

  $$\begin{aligned}
  \langle\vec{\gamma} \setminus^{\vec{\delta}} s\rangle\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle o &= \langle\vec{\gamma} \setminus^{\vec{\delta}} s\rangle\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle[\alpha_k]\,t \\
  &= \langle\vec{\gamma} \setminus^{\vec{\delta}} s\rangle[\beta_k]\,\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle t :: s \\
  &= [\beta_k]\,\langle\vec{\gamma} \setminus^{\vec{\delta}} s\rangle\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle t :: \langle\vec{\gamma} \setminus^{\vec{\delta}} s\rangle s \\
  &= [\beta_k]\,\langle\vec{\gamma} \setminus^{\vec{\delta}} s\rangle\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle t :: s \\
  &= [\beta_k]\,\langle\vec{\gamma}\vec{\alpha} \setminus^{\vec{\delta}\vec{\beta}} s\rangle t :: s \\
  &= \langle\vec{\gamma}\vec{\alpha} \setminus^{\vec{\delta}\vec{\beta}} s\rangle[\alpha_k]\,t \\
  &= \langle\vec{\gamma}\vec{\alpha} \setminus^{\vec{\delta}\vec{\beta}} s\rangle o
  \end{aligned}$$

  2. $\eta \in \vec{\gamma}$. Luego, $\eta = \gamma_k$ para algún $k \in [1, \mathsf{sz}(\vec{\gamma})]$. Por *h.i.* se tiene $\langle\vec{\gamma} \setminus^{\vec{\delta}} s\rangle\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle t = \langle\vec{\gamma}\vec{\alpha} \setminus^{\vec{\delta}\vec{\beta}} s\rangle t$. Se concluye aplicando la definición de replacement simultaneo:

  $$\begin{aligned}
  \langle\vec{\gamma} \setminus^{\vec{\delta}} s\rangle\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle o &= \langle\vec{\gamma} \setminus^{\vec{\delta}} s\rangle\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle[\gamma_k]\,t \\
  &= \langle\vec{\gamma} \setminus^{\vec{\delta}} s\rangle[\gamma_k]\,\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle t \\
  &= [\delta_k]\,\langle\vec{\gamma} \setminus^{\vec{\delta}} s\rangle\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle t :: s \\
  &= [\delta_k]\,\langle\vec{\gamma}\vec{\alpha} \setminus^{\vec{\delta}\vec{\beta}} s\rangle t :: s \\
  &= \langle\vec{\gamma}\vec{\alpha} \setminus^{\vec{\delta}\vec{\beta}} s\rangle[\gamma_k]\,t \\
  &= \langle\vec{\gamma}\vec{\alpha} \setminus^{\vec{\delta}\vec{\beta}} s\rangle o
  \end{aligned}$$

  3. $\eta \notin \vec{\alpha}\vec{\gamma}$. Luego, por definición $\langle\vec{\gamma} \setminus^{\vec{\delta}} s\rangle\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle o = [\eta]\,\langle\vec{\gamma} \setminus^{\vec{\delta}} s\rangle\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle t$. Por *h.i.* se tiene $\langle\vec{\gamma} \setminus^{\vec{\delta}} s\rangle\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle t = \langle\vec{\gamma}\vec{\alpha} \setminus^{\vec{\delta}\vec{\beta}} s\rangle t$, por lo que se concluye $\langle\vec{\gamma} \setminus^{\vec{\delta}} s\rangle\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle o = [\eta]\,\langle\vec{\gamma}\vec{\alpha} \setminus^{\vec{\delta}\vec{\beta}} s\rangle t = \langle\vec{\gamma}\vec{\alpha} \setminus^{\vec{\delta}\vec{\beta}} s\rangle o$.

- $o = c[\xi \setminus \eta]$. con $\xi \notin \vec{\alpha}\vec{\beta}\vec{\gamma}\vec{\delta}$ y $\xi \notin s$. Se tienen tres casos posibles:

1. $\eta \in \vec{\alpha}$. Luego, $\eta = \alpha_k$ para algún $k \in [1, \mathsf{sz}(\vec{\alpha})]$. Por *h.i.* se tiene $\langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s \rangle\!\rangle \langle\!\langle \xi\vec{\alpha} \setminus^{\xi'\vec{\beta}} s \rangle\!\rangle c = \langle\!\langle \vec{\gamma}\xi\vec{\alpha} \setminus^{\vec{\delta}\xi'\vec{\beta}} s \rangle\!\rangle c$ donde ξ' es un nombre fresco. Más aún, por hipótesis, $\beta_k \notin \vec{\gamma}$. Se concluye apelando al Lem. C.0.1 para permutar consistentemente nombres en el replacement simultaneo:

$$\begin{array}{rcl}
\langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s \rangle\!\rangle \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle o & = & \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s \rangle\!\rangle \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle (c[\xi \setminus \alpha_k]) \\
& = & \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s \rangle\!\rangle ((\langle\!\langle \xi\vec{\alpha} \setminus^{\xi'\vec{\beta}} s \rangle\!\rangle c)[\xi' \setminus \beta_k]) \\
& = & (\langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s \rangle\!\rangle \langle\!\langle \xi\vec{\alpha} \setminus^{\xi'\vec{\beta}} s \rangle\!\rangle c)[\xi' \setminus \beta_k] \\
& = & (\langle\!\langle \vec{\gamma}\xi\vec{\alpha} \setminus^{\vec{\delta}\xi'\vec{\beta}} s \rangle\!\rangle c)[\xi' \setminus \beta_k] \\
& = & (\langle\!\langle \xi\vec{\gamma}\vec{\alpha} \setminus^{\xi'\vec{\delta}\vec{\beta}} s \rangle\!\rangle c)[\xi' \setminus \beta_k] \\
& = & \langle\!\langle \vec{\gamma}\vec{\alpha} \setminus^{\vec{\delta}\vec{\beta}} s \rangle\!\rangle (c[\xi \setminus \alpha_k]) \\
& = & \langle\!\langle \vec{\gamma}\vec{\alpha} \setminus^{\vec{\delta}\vec{\beta}} s \rangle\!\rangle o
\end{array}$$

2. $\eta \in \vec{\gamma}$. Luego, $\eta = \gamma_k$ para algún $k \in [1, \mathsf{sz}(\vec{\gamma})]$. Por *h.i.* se tiene $\langle\!\langle \xi\vec{\gamma} \setminus^{\xi'\vec{\delta}} s \rangle\!\rangle \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle c = \langle\!\langle \xi\vec{\gamma}\vec{\alpha} \setminus^{\xi'\vec{\delta}\vec{\beta}} s \rangle\!\rangle c$ donde ξ' es un nombre fresco. Se concluye aplicando la definición de replacement simultaneo:

$$\begin{array}{rcl}
\langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s \rangle\!\rangle \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle o & = & \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s \rangle\!\rangle \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle (c[\xi \setminus \gamma_k]) \\
& = & \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s \rangle\!\rangle ((\langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle c)[\xi \setminus \gamma_k]) \\
& = & (\langle\!\langle \xi\vec{\gamma} \setminus^{\xi'\vec{\delta}} s \rangle\!\rangle \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle c)[\xi' \setminus \delta_k] \\
& = & (\langle\!\langle \xi\vec{\gamma}\vec{\alpha} \setminus^{\xi'\vec{\delta}\vec{\beta}} s \rangle\!\rangle c)[\xi' \setminus \delta_k] \\
& = & \langle\!\langle \vec{\gamma}\vec{\alpha} \setminus^{\vec{\delta}\vec{\beta}} s \rangle\!\rangle (c[\xi \setminus \gamma_k]) \\
& = & \langle\!\langle \vec{\gamma}\vec{\alpha} \setminus^{\vec{\delta}\vec{\beta}} s \rangle\!\rangle o
\end{array}$$

3. $\eta \notin \vec{\alpha}\vec{\gamma}$. Luego, por definición se tiene $\langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s \rangle\!\rangle \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle o = (\langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s \rangle\!\rangle \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle c)[\xi \setminus \eta]$. Por *h.i.* se tiene $\langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s \rangle\!\rangle \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle c = \langle\!\langle \vec{\gamma}\vec{\alpha} \setminus^{\vec{\delta}\vec{\beta}} s \rangle\!\rangle c$, por lo que se concluye $\langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s \rangle\!\rangle \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle o = (\langle\!\langle \vec{\gamma}\vec{\alpha} \setminus^{\vec{\delta}\vec{\beta}} s \rangle\!\rangle c)[\xi \setminus \eta] = \langle\!\langle \vec{\gamma}\vec{\alpha} \setminus^{\vec{\delta}\vec{\beta}} s \rangle\!\rangle o$.

- $o = t \cdot s'$. Luego, se tiene $\langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s \rangle\!\rangle \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle o = \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s \rangle\!\rangle \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle t \cdot \langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s \rangle\!\rangle \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle s'$. Por *h.i.* se tienen $\langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s \rangle\!\rangle \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle t = \langle\!\langle \vec{\gamma}\vec{\alpha} \setminus^{\vec{\delta}\vec{\beta}} s \rangle\!\rangle t$ y $\langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s \rangle\!\rangle \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle s' = \langle\!\langle \vec{\gamma}\vec{\alpha} \setminus^{\vec{\delta}\vec{\beta}} s \rangle\!\rangle s'$, por lo que se concluye $\langle\!\langle \vec{\gamma} \setminus^{\vec{\delta}} s \rangle\!\rangle \langle\!\langle \vec{\alpha} \setminus^{\vec{\beta}} s \rangle\!\rangle o = \langle\!\langle \vec{\gamma}\vec{\alpha} \setminus^{\vec{\delta}\vec{\beta}} s \rangle\!\rangle t \cdot \langle\!\langle \vec{\gamma}\vec{\alpha} \setminus^{\vec{\delta}\vec{\beta}} s \rangle\!\rangle s' = \langle\!\langle \vec{\gamma}\vec{\alpha} \setminus^{\vec{\delta}\vec{\beta}} s \rangle\!\rangle o$.

□

Confluencia del $\lambda\mu\rho$-cálculo

La confluencia de $\lambda\mu\rho$ de prueba mediante el método de Tait–Martin-Löf [Tak95], para lo cual se introduce una noción de reducción en paralelo sobre objetos del $\lambda\mu\rho$-cálculo.

La *relación de reducción paralela* $\Rightarrow_{\lambda\mu\rho}$ para objetos de $\lambda\mu\rho$ se define inductivamente como:

$$\frac{}{x \Rightarrow_{\lambda\mu\rho} x}\ (\text{R-VAR}) \qquad \frac{t \Rightarrow_{\lambda\mu\rho} t' \quad u \Rightarrow_{\lambda\mu\rho} u'}{t\,u \Rightarrow_{\lambda\mu\rho} t'\,u'}\ (\text{R-APP}) \qquad \frac{t \Rightarrow_{\lambda\mu\rho} t'}{\lambda x.t \Rightarrow_{\lambda\mu\rho} \lambda x.t'}\ (\text{R-ABS})$$

$$\frac{c \Rightarrow_{\lambda\mu\rho} c'}{\mu\alpha.c \Rightarrow_{\lambda\mu\rho} \mu\alpha.c'}\ (\text{R-CONT}) \qquad \frac{t \Rightarrow_{\lambda\mu\rho} t'}{[\alpha]\,t \Rightarrow_{\lambda\mu\rho} [\alpha]\,t'}\ (\text{R-NAME})$$

$$\frac{c \Rightarrow_{\lambda\mu\rho} c'}{c[\alpha \setminus \beta] \Rightarrow_{\lambda\mu\rho} c'[\alpha \setminus \beta]}\ (\text{R-REN}) \qquad \frac{t \Rightarrow_{\lambda\mu\rho} t' \quad s \Rightarrow_{\lambda\mu\rho} s'}{t \cdot s \Rightarrow_{\lambda\mu\rho} t' \cdot s'}\ (\text{R-PUSH})$$

$$\frac{t \Rightarrow_{\lambda\mu\rho} t' \quad u \Rightarrow_{\lambda\mu\rho} u'}{(\lambda x.t)\,u \Rightarrow_{\lambda\mu\rho} \{x \setminus t'\}s'}\ (\text{R-BETA}) \qquad \frac{c \Rightarrow_{\lambda\mu\rho} c' \quad s \Rightarrow_{\lambda\mu\rho} s'}{(\mu\alpha.c) :: s \Rightarrow_{\lambda\mu\rho} \mu\alpha'.\langle\!\langle \alpha \setminus^{\alpha'} s' \rangle\!\rangle c'}\ (\text{R-MU})$$

Notar que la relación de reducción paralela resulta reflexiva.

A continuación se presentan los lemas de sustitución y replacement simultáneo para la relación de reducción paralela $\Rightarrow_{\lambda\mu\rho}$.

Lema C.0.7 (Sustitución)**.** *Sean* $o, o' \in \mathbb{O}_{\lambda\mu\rho}$ *y* $u, u' \in \mathbb{T}_{\lambda\mu\rho}$ *tal que* $o \Rightarrow_{\lambda\mu\rho} o'$ *y* $u \Rightarrow_{\lambda\mu\rho} u'$*. Luego,* $\{x \setminus u\}o \Rightarrow_{\lambda\mu\rho} \{x \setminus u'\}o'$*.*

Demostración. Por inducción en $o \Rightarrow_{\lambda\mu\rho} o'$.

- (R-VAR). Luego, $o = y = o'$. Se tienen dos casos posible:
  1. $y = x$. Luego, $\{x \setminus u\}o = u \Rightarrow_{\lambda\mu\rho} u' = \{x \setminus u'\}o'$ por hipótesis.
  2. $y \neq x$. Luego, $\{x \setminus u\}o = y \Rightarrow_{\lambda\mu\rho} y = \{x \setminus u'\}o'$ por (R-VAR).
- (R-APP). Luego, $o = t\,v$ y $o' = t'\,v'$ con $t \Rightarrow_{\lambda\mu\rho} t'$ y $v \Rightarrow_{\lambda\mu\rho} v'$. Por *h.i.* se tienen $\{x \setminus u\}t \Rightarrow_{\lambda\mu\rho} \{x \setminus u'\}t'$ y $\{x \setminus u\}v \Rightarrow_{\lambda\mu\rho} \{x \setminus u'\}v'$. Se concluye entonces por (R-APP), $\{x \setminus u\}o \Rightarrow_{\lambda\mu\rho} \{x \setminus u'\}o'$.
- (R-ABS). Luego, $o = \lambda y.t$ y $o' = \lambda y.t'$ con $y \neq x$, $y \notin \mathsf{fv}(u)$ y $t \Rightarrow_{\lambda\mu\rho} t'$. Por *h.i.* se tiene $\{x \setminus u\}t \Rightarrow_{\lambda\mu\rho} \{x \setminus u'\}t'$. Más aún, $y \notin \mathsf{fv}(u)$ implica $y \notin \mathsf{fv}(u')$. Se concluye entonces por (R-ABS), $\{x \setminus u\}o \Rightarrow_{\lambda\mu\rho} \{x \setminus u'\}o'$.
- (R-CONT). Luego, $o = \mu\alpha.c$ y $o' = \mu\alpha.c'$ con $\alpha \notin \mathsf{fn}(u)$ y $c \Rightarrow_{\lambda\mu\rho} c'$. Por *h.i.* se tiene $\{x \setminus u\}c \Rightarrow_{\lambda\mu\rho} \{x \setminus u'\}c'$. Más aún, $\alpha \notin \mathsf{fv}(u)$ implica $\alpha \notin \mathsf{fv}(u')$. Se concluye entonces por (R-CONT), $\{x \setminus u\}o \Rightarrow_{\lambda\mu\rho} \{x \setminus u'\}o'$.
- (R-NAME). Luego, $o = [\alpha]\,t$ y $o' = [\alpha]\,t'$ con $t \Rightarrow_{\lambda\mu\rho} t'$. Por *h.i.* se tiene $\{x \setminus u\}t \Rightarrow_{\lambda\mu\rho} \{x \setminus u'\}t'$. Se concluye entonces por (R-NAME), $\{x \setminus u\}o \Rightarrow_{\lambda\mu\rho} \{x \setminus u'\}o'$.
- (R-REN). Luego, $o = c[\alpha \setminus \beta]$ y $o' = c'[\alpha \setminus \beta]$ con $\alpha \notin \mathsf{fn}(u)$ y $c \Rightarrow_{\lambda\mu\rho} c'$. Por *h.i.* se tiene $\{x \setminus u\}c \Rightarrow_{\lambda\mu\rho} \{x \setminus u'\}c'$. Más aún, $\alpha \notin \mathsf{fv}(u)$ implica $\alpha \notin \mathsf{fv}(u')$. Se concluye entonces por (R-REN), $\{x \setminus u\}o \Rightarrow_{\lambda\mu\rho} \{x \setminus u'\}o'$.
- (R-PUSH). Luego, $o = t \cdot s$ y $o' = t'\,s'$ con $t \Rightarrow_{\lambda\mu\rho} t'$ y $s \Rightarrow_{\lambda\mu\rho} s'$. Por *h.i.* se tienen $\{x \setminus u\}t \Rightarrow_{\lambda\mu\rho} \{x \setminus u'\}t'$ y $\{x \setminus u\}s \Rightarrow_{\lambda\mu\rho} \{x \setminus u'\}s'$. Se concluye entonces por (R-PUSH), $\{x \setminus u\}o \Rightarrow_{\lambda\mu\rho} \{x \setminus u'\}o'$.
- (R-BETA). Luego, $o = (\lambda y.t)\,v$ y $o' = \{y \setminus v'\}t'$ con $y \neq x$, $y \notin \mathsf{fv}(u)$, $t \Rightarrow_{\lambda\mu\rho} t'$ y $v \Rightarrow_{\lambda\mu\rho} v'$. Por *h.i.* se tienen $\{x \setminus u\}t \Rightarrow_{\lambda\mu\rho} \{x \setminus u'\}t'$ y $\{x \setminus u\}v \Rightarrow_{\lambda\mu\rho} \{x \setminus u'\}v'$. Se concluye entonces por (R-BETA), $\{x \setminus u\}o = (\lambda y.\{x \setminus u'\}t)\,\{x \setminus u'\}v \Rightarrow_{\lambda\mu\rho} \{y \setminus \{x \setminus u'\}v'\}\{x \setminus u'\}t' = \{x \setminus u'\}\{y \setminus v'\}t' = \{x \setminus u'\}o'$.

- (R-MU). Luego, $o = (\mu\alpha.c) :: s$ y $o' = \mu\alpha'.\langle\alpha \setminus^{\alpha'} s'\rangle c'$ con $\alpha \notin \mathsf{fn}(u)$, α' fresco, $c \Rrightarrow_{\lambda\mu\rho} c'$ y $s \Rrightarrow_{\lambda\mu\rho} s'$. Por *h.i.* se tienen $\{x \setminus u\}c \Rrightarrow_{\lambda\mu\rho} \{x \setminus u'\}c'$ y $\{x \setminus u\}s \Rrightarrow_{\lambda\mu\rho} \{x \setminus u'\}s'$. Se concluye entonces por (R-MU), $\{x \setminus u\}o = (\mu\alpha.\{x \setminus u\}c) :: \{x \setminus u\}s \Rrightarrow_{\lambda\mu\rho} \mu\alpha'.\langle\alpha \setminus^{\alpha'} \{x \setminus u'\}s'\rangle\{x \setminus u'\}c' = \{x \setminus u'\}\mu\alpha'.\langle\alpha \setminus^{\alpha'} s'\rangle c' = \{x \setminus u'\}o'$.

□

Lema C.0.8 (Replacement simutáneo)**.** *Sean $o, o' \in \mathbb{O}_{\lambda\mu\rho}$ y s, s' stacks tal que $o \Rrightarrow_{\lambda\mu\rho} o'$ y $s \Rrightarrow_{\lambda\mu\rho} s'$. Luego, $\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle o \Rrightarrow_{\lambda\mu\rho} \langle\vec{\alpha} \setminus^{\vec{\beta}} s'\rangle o'$.*

Demostración. Por inducción en $o \Rrightarrow_{\lambda\mu\rho} o'$.

- (R-VAR). Luego, $o = x = o'$. Se concluye entonces por (R-VAR), $\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle o = x \Rrightarrow_{\lambda\mu\rho} x = \langle\vec{\alpha} \setminus^{\vec{\beta}} s'\rangle o'$.

- (R-APP). Luego, $o = t\,u$ y $o' = t'\,u'$ con $t \Rrightarrow_{\lambda\mu\rho} t'$ y $u \Rrightarrow_{\lambda\mu\rho} u'$. Por *h.i.* se tienen $\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle t \Rrightarrow_{\lambda\mu\rho} \langle\vec{\alpha} \setminus^{\vec{\beta}} s'\rangle t'$ y $\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle u \Rrightarrow_{\lambda\mu\rho} \langle\vec{\alpha} \setminus^{\vec{\beta}} s'\rangle u'$. Se concluye entonces por (R-APP), $\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle o \Rrightarrow_{\lambda\mu\rho} \langle\vec{\alpha} \setminus^{\vec{\beta}} s'\rangle o'$.

- (R-ABS). Luego, $o = \lambda x.t$ y $o' = \lambda x.t'$ con $x \notin \mathsf{fv}(s)$ y $t \Rrightarrow_{\lambda\mu\rho} t'$. Por *h.i.* se tiene $\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle t \Rrightarrow_{\lambda\mu\rho} \langle\vec{\alpha} \setminus^{\vec{\beta}} s'\rangle t'$. Más aún, $x \notin \mathsf{fv}(s)$ implica $x \notin \mathsf{fv}(s')$. Se concluye entonces por (R-ABS), $\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle o \Rrightarrow_{\lambda\mu\rho} \langle\vec{\alpha} \setminus^{\vec{\beta}} s'\rangle o'$.

- (R-CONT). Luego, $o = \mu\gamma.c$ y $o' = \mu\gamma.c'$ con $\gamma \notin \mathsf{fn}(s)$ y $c \Rrightarrow_{\lambda\mu\rho} c'$. Por *h.i.* se tiene $\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle c \Rrightarrow_{\lambda\mu\rho} \langle\vec{\alpha} \setminus^{\vec{\beta}} s'\rangle c'$. Más aún, $\alpha \notin \mathsf{fv}(s)$ implica $\alpha \notin \mathsf{fv}(s')$. Se concluye entonces por (R-CONT), $\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle o \Rrightarrow_{\lambda\mu\rho} \langle\vec{\alpha} \setminus^{\vec{\beta}} s'\rangle o'$.

- (R-NAME). Luego, $o = [\delta]\,t$ y $o' = [\delta]\,t'$ con $t \Rrightarrow_{\lambda\mu\rho} t'$. Por *h.i.* se tiene $\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle t \Rrightarrow_{\lambda\mu\rho} \langle\vec{\alpha} \setminus^{\vec{\beta}} s'\rangle t'$. Se concluye entonces por (R-NAME), $\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle o \Rrightarrow_{\lambda\mu\rho} \langle\vec{\alpha} \setminus^{\vec{\beta}} s'\rangle o'$.

- (R-REN). Luego, $o = c[\gamma \setminus \delta]$ y $o' = c'[\gamma \setminus \delta]$ con $\gamma \notin \mathsf{fn}(s)$ y $c \Rrightarrow_{\lambda\mu\rho} c'$. Por *h.i.* se tiene $\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle c \Rrightarrow_{\lambda\mu\rho} \langle\vec{\alpha} \setminus^{\vec{\beta}} s'\rangle c'$. Más aún, $\gamma \notin \mathsf{fv}(s)$ implica $\gamma \notin \mathsf{fv}(s')$. Se concluye entonces por (R-REN), $\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle o \Rrightarrow_{\lambda\mu\rho} \langle\vec{\alpha} \setminus^{\vec{\beta}} s'\rangle o'$.

- (R-PUSH). Luego, $o = t \cdot s_0$ y $o' = t'\,s_0'$ con $t \Rrightarrow_{\lambda\mu\rho} t'$ y $s_0 \Rrightarrow_{\lambda\mu\rho} s_0'$. Por *h.i.* se tienen $\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle t \Rrightarrow_{\lambda\mu\rho} \langle\vec{\alpha} \setminus^{\vec{\beta}} s'\rangle t'$ y $\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle s_0 \Rrightarrow_{\lambda\mu\rho} \langle\vec{\alpha} \setminus^{\vec{\beta}} s'\rangle s_0'$. Se concluye entonces por (R-PUSH), $\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle o \Rrightarrow_{\lambda\mu\rho} \langle\vec{\alpha} \setminus^{\vec{\beta}} s'\rangle o'$.

- (R-BETA). Luego, $o = (\lambda x.t)\,u$ y $o' = \{x \setminus u'\}t'$ con $x \notin \mathsf{fv}(s)$, $t \Rrightarrow_{\lambda\mu\rho} t'$ y $u \Rrightarrow_{\lambda\mu\rho} u'$. Por *h.i.* se tienen $\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle t \Rrightarrow_{\lambda\mu\rho} \langle\vec{\alpha} \setminus^{\vec{\beta}} s'\rangle t'$ y $\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle u \Rrightarrow_{\lambda\mu\rho} \langle\vec{\alpha} \setminus^{\vec{\beta}} s'\rangle u'$. Se concluye por (R-BETA) apelando al Lem. C.0.4 (1), $\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle o = (\lambda x.\langle\vec{\alpha} \setminus^{\vec{\beta}} s'\rangle t)\,\langle\vec{\alpha} \setminus^{\vec{\beta}} s'\rangle u \Rrightarrow_{\lambda\mu\rho} \{x \setminus \langle\vec{\alpha} \setminus^{\vec{\beta}} s'\rangle u'\}\langle\vec{\alpha} \setminus^{\vec{\beta}} s'\rangle t' = \langle\vec{\alpha} \setminus^{\vec{\beta}} s'\rangle\{x \setminus u'\}t' = \langle\vec{\alpha} \setminus^{\vec{\beta}} s'\rangle o'$.

- (R-MU). Luego, $o = (\mu\gamma.c) :: s_0$ y $o' = \mu\delta.\langle\gamma \setminus^{\delta} s_0'\rangle c'$ con $\gamma \notin \mathsf{fn}(s)$, α' fresco, $c \Rrightarrow_{\lambda\mu\rho} c'$ y $s_0 \Rrightarrow_{\lambda\mu\rho} s_0'$. Por *h.i.* se tienen $\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle c \Rrightarrow_{\lambda\mu\rho} \langle\vec{\alpha} \setminus^{\vec{\beta}} s'\rangle c'$ y $\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle s_0 \Rrightarrow_{\lambda\mu\rho} \langle\vec{\alpha} \setminus^{\vec{\beta}} s'\rangle s_0'$. Se concluye entonces por (R-MU) apelando al Lem. C.0.5, $\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle o = (\mu\gamma.\langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle c) :: \langle\vec{\alpha} \setminus^{\vec{\beta}} s\rangle s_0 \Rrightarrow_{\lambda\mu\rho} \mu\delta.\langle\gamma \setminus^{\delta} \langle\vec{\alpha} \setminus^{\vec{\beta}} s'\rangle s_0'\rangle\langle\vec{\alpha} \setminus^{\vec{\beta}} s'\rangle c' = \langle\vec{\alpha} \setminus^{\vec{\beta}} s'\rangle\mu\delta.\langle\gamma \setminus^{\delta} s'\rangle c' = \langle\vec{\alpha} \setminus^{\vec{\beta}} s'\rangle o'$.

□

Un último resultado auxilar muestra que la reducción paralela a patir de un término de la forma $(\mu\alpha.c) :: s$ puede implicar que la μ-abstracción consuma todos, algunos o ningún argumento en s.

Lema C.0.9. *Sean $t \in \mathbb{T}_{\lambda\mu\rho}$, $c \in \mathbb{C}_{\lambda\mu\rho}$ y s un stack tal que $(\mu\alpha.c) :: s \Rrightarrow_{\lambda\mu\rho} t$. Luego,*

1. $t = (\mu\alpha.c') :: s'$ *con* $c \Rrightarrow_{\lambda\mu\rho} c'$ *y* $s \Rrightarrow_{\lambda\mu\rho} s'$; *o*
2. $t = \mu\alpha'.\langle\!\langle \alpha \setminus^{\alpha'} s' \rangle\!\rangle c'$ *con* $c \Rrightarrow_{\lambda\mu\rho} c'$ *y* $s \Rrightarrow_{\lambda\mu\rho} s'$; *o*
3. $t = (\mu\alpha'.\langle\!\langle \alpha \setminus^{\alpha'} s'_0 \rangle\!\rangle c') :: s'_1$ *con* $s = s_0 \cdot s_1$, $c \Rrightarrow_{\lambda\mu\rho} c'$ *y* $s \Rrightarrow_{\lambda\mu\rho} s'_0 \cdot s'_1$.

Demostración. Por inducción en $(\mu\alpha.c) :: s \Rrightarrow_{\lambda\mu\rho} t$. Notar que solo hay dos reglas aplicables dada la forma del redex.

- (R-APP). Luego, $(\mu\alpha.c) :: s = t_0\, u_0$ con $t_0 \Rrightarrow_{\lambda\mu\rho} t'_0$ y $u_0 \Rrightarrow_{\lambda\mu\rho} u'_0$. Se tienen dos casos posibles.
  1. $s = u_0$. Luego, $t_0 = \mu\alpha.c$ y, por lo tanto, $t'_0 = \mu\alpha.c'$ con $c \Rrightarrow_{\lambda\mu\rho} c'$. Luego, se satisface el ítem 1.
  2. $s = s_2 \cdot u_0$. Luego, $t_0 = (\mu\alpha.c) :: s_2$ y, por lo tanto, $(\mu\alpha.c) :: s_2 \Rrightarrow_{\lambda\mu\rho} t'_0$. Por *h.i.* se tienen tres casos:
     1 Luego, $t'_0 = (\mu\alpha.c'_0) :: s'_2$ con $c \Rrightarrow_{\lambda\mu\rho} c'_0$ y $s_2 \Rrightarrow_{\lambda\mu\rho} s'_2$. Se satisface el ítem 1 tomando $c' = c'_0$ y $s' = s'_2 \cdot u'_0$.
     2 Luego, $t'_0 = \mu\alpha'.c'_0[\![\alpha \setminus^{\alpha'} s'_2]\!]$ con $c \Rrightarrow_{\lambda\mu\rho} c'_0$ y $s_2 \Rrightarrow_{\lambda\mu\rho} s'_2$. Se satisface el ítem 3 tomando $s_0 = s_2$, $s_1 = u_0$, $c' = c'_0$, $s'_0 = s'_2$ y $s'_1 = u'_0$.
     3 Luego, $t'_0 = (\mu\alpha'.c'_0[\![\alpha \setminus^{\alpha'} s'_3]\!]) :: s'_4$ con $s_2 = s_3 \cdot s_4$, $c \Rrightarrow_{\lambda\mu\rho} c'_0$ y $s_2 \Rrightarrow_{\lambda\mu\rho} s'_3 \cdot s'_4$. Se satisface el ítem 3 tomando $s_0 = s_3$, $s_1 = s_4 \cdot u_0$, $c' = c'_0$, $s'_0 = s'_3$ y $s'_1 = s'_4 \cdot u'_0$.
- (R-MU). Luego, $(\mu\alpha.c) :: s \Rrightarrow_{\lambda\mu\rho} \mu\alpha'.c'[\![\alpha \setminus^{\alpha'} s']\!]$ con $c \Rrightarrow_{\lambda\mu\rho} c'$ y $s \Rrightarrow_{\lambda\mu\rho} s'$ satisfaciendo por lo tanto el ítem 2.

□

La confluencia de la relación de reducción paralela de prueba mostrando que $\Rrightarrow_{\lambda\mu\rho}$ satisface la propiedad del diamante [Nip90, BKvO03]. Para ello se apela a los resultados anteriormente demostrados.

Lema C.0.10. *La relación de reducción paralela $\Rrightarrow_{\lambda\mu\rho}$ satisface la propiedad del diamante y, por lo tanto, es confluente (CR).*

Demostración. Sean $o \Rrightarrow_{\lambda\mu\rho} o_0$ y $o \Rrightarrow_{\lambda\mu\rho} o_1$, se prueba que existe o' tal que $o_0 \Rrightarrow_{\lambda\mu\rho} o'$ y $o_1 \Rrightarrow_{\lambda\mu\rho} o'$. La prueba es por inducción en o.

- $o = x$. Luego, se tiene necesariamente $o_0 = x = o_1$ y el resultado es inmediato por (R-VAR), tomando $o' = x$.
- $o = t\, u$. Luego, se tienen siete posibles combinaciones de reglas para $o_0 \Rrightarrow_{\lambda\mu\rho} o'$ y $o_1 \Rrightarrow_{\lambda\mu\rho} o'$.
  1. (R-BETA)/(R-APP). Luego, por (R-BETA) se tiene $t = \lambda x.v$, por lo que $o = (\lambda x.v)\, u$ y $o_0 = \{x \setminus u_0\}v_0$ con $v \Rrightarrow_{\lambda\mu\rho} v_0$ y $u \Rrightarrow_{\lambda\mu\rho} u_0$. Además, por (R-APP) se tienen $o_1 = t_1\, u_1$ con $t \Rrightarrow_{\lambda\mu\rho} t_1$ y $u \Rrightarrow_{\lambda\mu\rho} u_1$. Más aún, $t \Rrightarrow_{\lambda\mu\rho} t_1$ implica $t_1 = \lambda x.v_1$ con $v \Rrightarrow_{\lambda\mu\rho} v_1$ por (R-ABS). Por *h.i.* existen v' y u' tales que $v_0 \Rrightarrow_{\lambda\mu\rho} v'$, $v_1 \Rrightarrow_{\lambda\mu\rho} v'$, $u_0 \Rrightarrow_{\lambda\mu\rho} u'$ y $u_1 \Rrightarrow_{\lambda\mu\rho} u'$. Se concluye entonces tomando $o' = \{x \setminus u'\}v'$ con $o_0 \Rrightarrow_{\lambda\mu\rho} o'$ por Lem. C.0.7 y $o_1 \Rrightarrow_{\lambda\mu\rho} o'$ por (R-BETA).
  2. (R-APP)/(R-BETA). Este caso es análogo al anterior.

3. (R-BETA)/(R-BETA). Luego, se tiene $t = \lambda x.v$, por lo que $o = (\lambda x.v)\,u$, $o_0 = \{x \setminus u_0\}v_0$ y $o_1 = \{x \setminus u_1\}v_1$ con $v \Rightarrow_{\lambda\mu\rho} v_0$, $u \Rightarrow_{\lambda\mu\rho} u_0$, $v \Rightarrow_{\lambda\mu\rho} v_1$ y $u \Rightarrow_{\lambda\mu\rho} u_1$. Por *h.i.* existen v' y u' tales que $v_0 \Rightarrow_{\lambda\mu\rho} v'$, $v_1 \Rightarrow_{\lambda\mu\rho} v'$, $u_0 \Rightarrow_{\lambda\mu\rho} u'$ y $u_1 \Rightarrow_{\lambda\mu\rho} u'$. Se concluye entonces tomando $o' = \{x \setminus u'\}v'$ con $o_0 \Rightarrow_{\lambda\mu\rho} o'$ y $o_1 \Rightarrow_{\lambda\mu\rho} o'$ por Lem. C.0.7.
4. (R-MU)/(R-APP). Luego, se tienen dos casos posibles.
   a) $t = \mu\alpha.c$ con $o = (\mu\alpha.c)\,u$. Luego, $o_0 = \mu\alpha_0.\langle\alpha \setminus^{\alpha_0} u_0\rangle c_0$ y $o_1 = t_1\,u_1$ con $c \Rightarrow_{\lambda\mu\rho} c_0$, $u \Rightarrow_{\lambda\mu\rho} u_0$, $t \Rightarrow_{\lambda\mu\rho} t_1$ y $u \Rightarrow_{\lambda\mu\rho} u_1$. Más aún, $t \Rightarrow_{\lambda\mu\rho} t_1$ implica $t_1 = \mu\alpha.c_1$ con $c \Rightarrow_{\lambda\mu\rho} c_1$. Por *h.i.* existen c' y u' tales que $c_0 \Rightarrow_{\lambda\mu\rho} c'$, $c_1 \Rightarrow_{\lambda\mu\rho} c'$, $u_0 \Rightarrow_{\lambda\mu\rho} u'$ y $u_1 \Rightarrow_{\lambda\mu\rho} u'$. Se concluye entonces apelando a la α-conversión, tomando $o' = \mu\alpha'.\langle\alpha \setminus^{\alpha'} u'\rangle c'$ con $o_0 \Rightarrow_{\lambda\mu\rho} o'$ por Lem. C.0.8 y $o_1 \Rightarrow_{\lambda\mu\rho} o'$ por (R-MU).
   b) $t = (\mu\alpha.c) :: s$ con $o = (\mu\alpha.c) :: (s \cdot u)$. Luego, $o_0 = \mu\alpha_0.\langle\alpha \setminus^{\alpha_0} s_0\rangle c_0$ y $o_1 = t_1\,u_1$ con $c \Rightarrow_{\lambda\mu\rho} c_0$, $s \cdot u \Rightarrow_{\lambda\mu\rho} s_0$, $t \Rightarrow_{\lambda\mu\rho} t_1$ y $u \Rightarrow_{\lambda\mu\rho} u_1$. Por Lem. C.0.9 con $t \Rightarrow_{\lambda\mu\rho} t_1$ se tienen tres posibles casos para t_1.
      1) $t_1 = (\mu\alpha_1.c_1) :: s_1$ con $c \Rightarrow_{\lambda\mu\rho} c_1$ y $s \Rightarrow_{\lambda\mu\rho} s_1$. Más aún, $s \Rightarrow_{\lambda\mu\rho} s_1$ y $u \Rightarrow_{\lambda\mu\rho} u_1$ implican $s \cdot u \Rightarrow_{\lambda\mu\rho} s_1 \cdot u_1$. Luego, por *h.i.* existen c' y s' tales que $c_0 \Rightarrow_{\lambda\mu\rho} c'$, $c_1 \Rightarrow_{\lambda\mu\rho} c'$, $s_0 \Rightarrow_{\lambda\mu\rho} s'$, $s_1 \cdot u_1 \Rightarrow_{\lambda\mu\rho} s'$. Se concluye entonces apelando a la α-conversión, tomando $o' = \mu\alpha'.\langle\alpha \setminus^{\alpha'} s'\rangle c'$ con $o_0 \Rightarrow_{\lambda\mu\rho} o'$ por Lem. C.0.8 y $o_1 \Rightarrow_{\lambda\mu\rho} o'$ por (R-MU).
      2) $t_1 = \mu\alpha_1.\langle\alpha \setminus^{\alpha_1} s_1\rangle c_1$ con $c \Rightarrow_{\lambda\mu\rho} c_1$ y $s \Rightarrow_{\lambda\mu\rho} s_1$. Más aún, $s \cdot u \Rightarrow_{\lambda\mu\rho} s_0$ implica $s_0 = s_0' \cdot u_0$ con $s \Rightarrow_{\lambda\mu\rho} s_0'$ y $u \Rightarrow_{\lambda\mu\rho} u_0$. Por *h.i.* existen c', s' y u' tales que $c_0 \Rightarrow_{\lambda\mu\rho} c'$, $c_1 \Rightarrow_{\lambda\mu\rho} c'$, $s_0' \Rightarrow_{\lambda\mu\rho} s'$, $s_1 \Rightarrow_{\lambda\mu\rho} s'$, $u_0 \Rightarrow_{\lambda\mu\rho} u'$ y $u_1 \Rightarrow_{\lambda\mu\rho} u'$. Luego, en particular, $s_0 \Rightarrow_{\lambda\mu\rho} s' \cdot u'$ por (R-PUSH). Apelando a la α-conversión, por Lem. C.0.8 se tiene $o_0 \Rightarrow_{\lambda\mu\rho} \mu\alpha'.\langle\alpha \setminus^{\alpha'} s' \cdot u'\rangle c'$ y, por (R-MU), $o_1 \Rightarrow_{\lambda\mu\rho} \mu\alpha'.\langle\alpha_1 \setminus^{\alpha'} u'\rangle\langle\alpha \setminus^{\alpha_1} s'\rangle c'$. Se concluye por Lem. C.0.5, con $o' = \mu\alpha'.\langle\alpha \setminus^{\alpha'} s' \cdot u'\rangle c' = \mu\alpha'.\langle\alpha_1 \setminus^{\alpha'} u'\rangle\langle\alpha \setminus^{\alpha_1} s'\rangle c'$.
      3) $t_1 = (\mu\alpha_1.\langle\alpha \setminus^{\alpha_1} s_1\rangle c_1) :: s_1'$ con $s = s_2 \cdot s_2'$, $c \Rightarrow_{\lambda\mu\rho} c_1$ y $s \Rightarrow_{\lambda\mu\rho} s_1 \cdot s_1'$. Más aún, $s \Rightarrow_{\lambda\mu\rho} s_1 \cdot s_1'$ y $u \Rightarrow_{\lambda\mu\rho} u_1$ implican $s \cdot u \Rightarrow_{\lambda\mu\rho} s_1 \cdot s_1' \cdot u_1$ por (R-PUSH). Luego, por *h.i.* existen c' y s' tales que $c_0 \Rightarrow_{\lambda\mu\rho} c'$, $c_1 \Rightarrow_{\lambda\mu\rho} c'$, $s_0 \Rightarrow_{\lambda\mu\rho} s'$ y $s_1 \cdot s_1' \cdot u_1 \Rightarrow_{\lambda\mu\rho} s'$, lo que implica $s' = s_3 \cdot s_3' \cdot u_3$ con $s_1 \Rightarrow_{\lambda\mu\rho} s_3$, $s_1' \cdot u_1 \Rightarrow_{\lambda\mu\rho} s_3' \cdot u_3$ por (R-PUSH). Apelando a la α-conversión, por Lem. C.0.8 se tiene $o_0 \Rightarrow_{\lambda\mu\rho} \mu\alpha'.\langle\alpha \setminus^{\alpha'} s'\rangle c'$ y, por (R-MU), $o_1 \Rightarrow_{\lambda\mu\rho} \mu\alpha'.\langle\alpha_1 \setminus^{\alpha'} s_3' \cdot u_3\rangle\langle\alpha \setminus^{\alpha_1} s_3\rangle c'$. Se concluye por Lem. C.0.5, con $o' = \mu\alpha'.\langle\alpha \setminus^{\alpha'} s'\rangle c' = \mu\alpha'.\langle\alpha_1 \setminus^{\alpha'} s_3' \cdot u_3\rangle\langle\alpha \setminus^{\alpha_1} s_3\rangle c'$.
5. (R-APP)/(R-MU). Este caso es análogo al anterior.
6. (R-MU)/(R-MU). Luego, se tienen dos casos posibles.
   a) $t = \mu\alpha.c$ con $o = (\mu\alpha.c)\,u$. Luego, se tienen $o_0 = \mu\alpha_0.\langle\alpha \setminus^{\alpha_0} u_0\rangle c_0$ y $o_1 = \mu\alpha_1.\langle\alpha \setminus^{\alpha_1} u_1\rangle c_1$ con $c \Rightarrow_{\lambda\mu\rho} c_0$, $u \Rightarrow_{\lambda\mu\rho} u_0$, $c \Rightarrow_{\lambda\mu\rho} c_1$ y $u \Rightarrow_{\lambda\mu\rho} u_1$. Por *h.i.* existen c' y u' tales que $c_0 \Rightarrow_{\lambda\mu\rho} c'$, $c_1 \Rightarrow_{\lambda\mu\rho} c'$, $u_0 \Rightarrow_{\lambda\mu\rho} u'$ y $u_1 \Rightarrow_{\lambda\mu\rho} u'$. Se concluye entonces apelando a la α-conversión, tomando $o' = \mu\alpha'.\langle\alpha \setminus^{\alpha'} u'\rangle c'$ con $o_0 \Rightarrow_{\lambda\mu\rho} o'$ y $o_1 \Rightarrow_{\lambda\mu\rho} o'$ por Lem. C.0.8.
   b) $t = (\mu\alpha.c) :: s$ con $o = (\mu\alpha.c) :: (s \cdot u)$. Luego, se tienen $o_0 = \mu\alpha_0.\langle\alpha \setminus^{\alpha_0} s_0\rangle c_0$ y $o_1 = \mu\alpha_1.\langle\alpha \setminus^{\alpha_1} s_1\rangle c_1$ con $c \Rightarrow_{\lambda\mu\rho} c_0$, $s \cdot u \Rightarrow_{\lambda\mu\rho} s_0$, $c \Rightarrow_{\lambda\mu\rho} c_1$ y $s \cdot u \Rightarrow_{\lambda\mu\rho} s_1$. Por *h.i.* existen c' y s' tales que $c_0 \Rightarrow_{\lambda\mu\rho} c'$, $c_1 \Rightarrow_{\lambda\mu\rho} c'$, $s_0 \Rightarrow_{\lambda\mu\rho} s'$ y $s_1 \Rightarrow_{\lambda\mu\rho} s'$. Se concluye entonces apelando a la α-conversión, tomando $o' = \mu\alpha'.\langle\alpha \setminus^{\alpha'} s'\rangle c'$ con $o_0 \Rightarrow_{\lambda\mu\rho} o'$ y $o_1 \Rightarrow_{\lambda\mu\rho} o'$ por Lem. C.0.8.
7. (R-APP)/(R-APP). Luego, se tienen $o_0 = t_0\,u_0$ y $o_1 = t_1\,u_1$ con $t \Rightarrow_{\lambda\mu\rho} t_0$, $t \Rightarrow_{\lambda\mu\rho} t_1$, $u \Rightarrow_{\lambda\mu\rho} u_0$ y $u \Rightarrow_{\lambda\mu\rho} u_1$. Por *h.i.* existen t' y u' tales que $t_0 \Rightarrow_{\lambda\mu\rho} t'$, $t_1 \Rightarrow_{\lambda\mu\rho} t'$, $u_0 \Rightarrow_{\lambda\mu\rho} u'$ y $u_1 \Rightarrow_{\lambda\mu\rho} u'$. Se concluye entonces tomando $o' = t'\,u'$ con $o_0 \Rightarrow_{\lambda\mu\rho} o'$ y $o_1 \Rightarrow_{\lambda\mu\rho} o'$ por (R-APP).

- $o = \lambda x.t$. Luego, por (R-ABS) se tienen $o_0 = \lambda x.t_0$ y $o_1 = \lambda x.t_1$ con $t \Rrightarrow_{\lambda\mu\rho} t_0$ y $t \Rrightarrow_{\lambda\mu\rho} t_1$. Por *h.i.* existe t' tal que $t_0 \Rrightarrow_{\lambda\mu\rho} t'$ y $t_1 \Rrightarrow_{\lambda\mu\rho} t'$. Se concluye entonces tomando $o' = \lambda x.t'$ con $o_0 \Rrightarrow_{\lambda\mu\rho} o'$ y $o_1 \Rrightarrow_{\lambda\mu\rho} o'$ por (R-ABS).

- $o = \mu\alpha.c$. Luego, por (R-CONT) se tienen $o_0 = \mu\alpha.c_0$ y $o_1 = \mu\alpha.c_1$ con $c \Rrightarrow_{\lambda\mu\rho} c_0$ y $c \Rrightarrow_{\lambda\mu\rho} c_1$. Por *h.i.* existe c' tal que $c_0 \Rrightarrow_{\lambda\mu\rho} c'$ y $c_1 \Rrightarrow_{\lambda\mu\rho} c'$. Se concluye entonces tomando $c' = \mu\alpha.c'$ con $o_0 \Rrightarrow_{\lambda\mu\rho} o'$ y $o_1 \Rrightarrow_{\lambda\mu\rho} o'$ por (R-CONT).

- $o = [\alpha]\, t$. Luego, por (R-NAME) se tienen $o_0 = [\alpha]\, t_0$ y $o_1 = [\alpha]\, t_1$ con $t \Rrightarrow_{\lambda\mu\rho} t_0$ y $t \Rrightarrow_{\lambda\mu\rho} t_1$. Por *h.i.* existe t' tal que $t_0 \Rrightarrow_{\lambda\mu\rho} t'$ y $t_1 \Rrightarrow_{\lambda\mu\rho} t'$. Se concluye entonces tomando $o' = [\alpha]\, t'$ con $o_0 \Rrightarrow_{\lambda\mu\rho} o'$ y $o_1 \Rrightarrow_{\lambda\mu\rho} o'$ por (R-NAME).

- $o = c[\alpha \backslash \beta]$. Luego, por (R-REN) se tienen $o_0 = c_0[\alpha \backslash \beta]$ y $o_1 = c_1[\alpha \backslash \beta]$ con $c \Rrightarrow_{\lambda\mu\rho} c_0$ y $c \Rrightarrow_{\lambda\mu\rho} c_1$. Por *h.i.* existe c' tal que $c_0 \Rrightarrow_{\lambda\mu\rho} c'$ y $c_1 \Rrightarrow_{\lambda\mu\rho} c'$. Se concluye entonces tomando $c' = c'[\alpha \setminus \beta]$ con $o_0 \Rrightarrow_{\lambda\mu\rho} o'$ y $o_1 \Rrightarrow_{\lambda\mu\rho} o'$ por (R-REN).

- $o = t \cdot s$. Luego, por (R-APP) se tienen $o_0 = t_0 \cdot s_0$ y $o_1 = t_1 \cdot s_1$ con $t \Rrightarrow_{\lambda\mu\rho} t_0$, $t \Rrightarrow_{\lambda\mu\rho} t_1$, $s \Rrightarrow_{\lambda\mu\rho} s_0$ y $s \Rrightarrow_{\lambda\mu\rho} s_1$. Por *h.i.* existen t' y s' tales que $t_0 \Rrightarrow_{\lambda\mu\rho} t'$, $t_1 \Rrightarrow_{\lambda\mu\rho} t'$, $s_0 \Rrightarrow_{\lambda\mu\rho} s'$ y $s_1 \Rrightarrow_{\lambda\mu\rho} s'$. Se concluye entonces tomando $o' = t' \cdot s'$ con $o_0 \Rrightarrow_{\lambda\mu\rho} o'$ y $o_1 \Rrightarrow_{\lambda\mu\rho} o'$ por (R-PUSH).

□

Por último, para deducir la confluencia de $\lambda\mu\rho$ mediante el método de Tait–Martin-Löf, basta que ver $\Rrightarrow_{\lambda\mu\rho}$ extiende a $\to_{\lambda\mu\rho}$ y al mismo tiempo está incluida en su clausura reflexiva–transitiva $\twoheadrightarrow_{\lambda\mu\rho}$.

Lema C.0.11. $\to_{\lambda\mu\rho} \subseteq \Rrightarrow_{\lambda\mu\rho} \subseteq \twoheadrightarrow_{\lambda\mu\rho}$.

Demostración. Para la primera inclusión ($\to_{\lambda\mu\rho} \subseteq \Rrightarrow_{\lambda\mu\rho}$) $o \to_{\Lambda M} o'$. Por definición $o = \mathtt{O}\langle l \rangle$ y $o' = \mathtt{O}\langle r \rangle$ con $l \mapsto_* r$, $* \in \{\beta, \mu^s\}$. Se procede por inducción en $\mathtt{O}$ para mostrar que $o \Rrightarrow_{\lambda\mu\rho} o'$.

- $\mathtt{O} = \Box$. Luego, se tienen dos casos posibles según la regla de reescritura aplicada al reducir.

  1. β. Luego, $o = (\lambda x.t)\, u$ y $o' = \{x \setminus u\}t$. Por reflexividad de $\Rrightarrow_{\lambda\mu\rho}$ se tienen $t \Rrightarrow_{\lambda\mu\rho} t$ y $u \Rrightarrow_{\lambda\mu\rho} u$. Luego, por (R-BETA), $(\lambda x.t)\, u \Rrightarrow_{\lambda\mu\rho} \{x \setminus u\}t$.
  2. μ^s. Luego, $o = (\mu\alpha.c) :: s$ y $o' = \mu\alpha'.\langle\!\langle \alpha \setminus^{\alpha'} s \rangle\!\rangle c$. Por reflexividad de $\Rrightarrow_{\lambda\mu\rho}$ se tienen $c \Rrightarrow_{\lambda\mu\rho} c$ y $s \Rrightarrow_{\lambda\mu\rho} s$. Luego, por (R-MU), $(\mu\alpha.c) :: s \Rrightarrow_{\lambda\mu\rho} \mu\alpha'.\langle\!\langle \alpha \setminus^{\alpha'} s \rangle\!\rangle c$.

- $\mathtt{O} = \boxdot$. Este caso no aplica dado que no hay reglas de reescritura sobre comandos.

- $\mathtt{O} = \mathtt{T}\, u$. Luego, $o = t\, u$ y $o' = t'\, u$ con $t \to_{\lambda\mu\rho} t'$. Por *h.i.* se tiene $t \Rrightarrow_{\lambda\mu\rho} t'$. Más aún, por reflexividad de $\Rrightarrow_{\lambda\mu\rho}$ se tiene $u \Rrightarrow_{\lambda\mu\rho} u$. Se concluye por (R-APP), $t\, u \Rrightarrow_{\lambda\mu\rho} t'\, u$.

- $\mathtt{O} = \mathtt{T}\, u$. Luego, $o = t\, u$ y $o' = t\, u'$ con $u \to_{\lambda\mu\rho} u'$. Por *h.i.* se tiene $u \Rrightarrow_{\lambda\mu\rho} u'$. Más aún, por reflexividad de $\Rrightarrow_{\lambda\mu\rho}$ se tiene $t \Rrightarrow_{\lambda\mu\rho} t$. Se concluye por (R-APP), $t\, u \Rrightarrow_{\lambda\mu\rho} t\, u'$.

- $\mathtt{O} = \lambda x.\mathtt{T}$. Luego, $o = \lambda x.t$ y $o' = \lambda x.t'$ con $t \to_{\lambda\mu\rho} t'$. Por *h.i.* se tiene $t \Rrightarrow_{\lambda\mu\rho} t'$. Se concluye por (R-ABS), $\lambda x.t \Rrightarrow_{\lambda\mu\rho} \lambda x.t'$.

- $\mathtt{O} = \mu\alpha.\mathtt{C}$. Luego, $o = \mu\alpha.c$ y $o' = \mu\alpha.c'$ con $c \to_{\lambda\mu\rho} c'$. Por *h.i.* se tiene $c \Rrightarrow_{\lambda\mu\rho} c'$. Se concluye por (R-CONT), $\mu\alpha.c \Rrightarrow_{\lambda\mu\rho} \mu\alpha.c'$.

- $\mathtt{O} = [\alpha]\, \mathtt{T}$. Luego, $o = [\alpha]\, t$ y $o' = [\alpha]\, t'$ con $t \to_{\lambda\mu\rho} t'$. Por *h.i.* se tiene $t \Rrightarrow_{\lambda\mu\rho} t'$. Se concluye por (R-NAME), $[\alpha]\, t \Rrightarrow_{\lambda\mu\rho} [\alpha]\, t'$.

- $\mathtt{O} = \mathtt{C}[\alpha \setminus \beta]$. Luego, $o = c[\alpha \setminus \beta]$ y $o' = c'[\alpha \setminus \beta]$ con $c \to_{\lambda\mu\rho} c'$. Por *h.i.* se tiene $c \Rrightarrow_{\lambda\mu\rho} c'$. Se concluye por (R-REN), $c[\alpha \setminus \beta] \Rrightarrow_{\lambda\mu\rho} c'[\alpha \setminus \beta]$.
- $\mathtt{O} = \mathtt{T} \cdot s$. Luego, $o = t \cdot s$ y $o' = t' \cdot s$ con $t \to_{\lambda\mu\rho} t'$. Por *h.i.* se tiene $t \Rrightarrow_{\lambda\mu\rho} t'$. Más aún, por reflexividad de $\Rrightarrow_{\lambda\mu\rho}$ se tiene $s \Rrightarrow_{\lambda\mu\rho} s$. Se concluye por (R-APP), $t \cdot s \Rrightarrow_{\lambda\mu\rho} t' \cdot s$.
- $\mathtt{O} = t \cdot \mathtt{S}$. Luego, $o = t \cdot s$ y $o' = t \cdot s'$ con $s \to_{\lambda\mu\rho} s'$. Por *h.i.* se tiene $s \Rrightarrow_{\lambda\mu\rho} s'$. Más aún, por reflexividad de $\Rrightarrow_{\lambda\mu\rho}$ se tiene $t \Rrightarrow_{\lambda\mu\rho} t$. Se concluye por (R-APP), $t \cdot s \Rrightarrow_{\lambda\mu\rho} t \cdot s'$.

Para la segunda inclusión ($\Rrightarrow_{\lambda\mu\rho} \subseteq \twoheadrightarrow_{\lambda\mu\rho}$) considerar $o \Rrightarrow_{\lambda\mu\rho} o'$. Se procede por inducción en $o \Rrightarrow_{\lambda\mu\rho} o'$ para mostrar que $o \twoheadrightarrow_{\lambda\mu\rho} o'$.

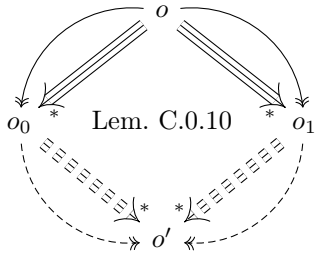
- (R-VAR). Luego, $o = x = o'$ y se concluye por reflexividad de $\twoheadrightarrow_{\lambda\mu\rho}$.
- (R-APP). Luego, $o = t\,u$ y $o' = t'\,u'$ con $t \Rrightarrow_{\lambda\mu\rho} t'$ y $u \Rrightarrow_{\lambda\mu\rho} u'$. Por *h.i.* se tienen $t \twoheadrightarrow_{\lambda\mu\rho} t'$ y $u \twoheadrightarrow_{\lambda\mu\rho} u'$. Luego, $t\,u \twoheadrightarrow_{\lambda\mu\rho} t'\,u \twoheadrightarrow_{\lambda\mu\rho} t'\,u'$. Se concluye por transitividad de $\twoheadrightarrow_{\lambda\mu\rho}$.
- (R-ABS). Luego, $o = \lambda x.t$ y $o' = \lambda x.t'$ con $t \Rrightarrow_{\lambda\mu\rho} t'$. Por *h.i.* se tiene $t \twoheadrightarrow_{\lambda\mu\rho} t'$. Se concluye entonces $\lambda x.t \twoheadrightarrow_{\lambda\mu\rho} \lambda x.t'$.
- (R-CONT). Luego, $o = \mu\alpha.c$ y $o' = \mu\alpha.c'$ con $c \Rrightarrow_{\lambda\mu\rho} c'$. Por *h.i.* se tiene $c \twoheadrightarrow_{\lambda\mu\rho} c'$. Se concluye entonces $\mu\alpha.c \twoheadrightarrow_{\lambda\mu\rho} \mu\alpha.c'$.
- (R-NAME). Luego, $o = [\alpha]\,t$ y $o' = [\alpha]\,t'$ con $t \Rrightarrow_{\lambda\mu\rho} t'$. Por *h.i.* se tiene $t \twoheadrightarrow_{\lambda\mu\rho} t'$. Se concluye entonces $[\alpha]\,t \twoheadrightarrow_{\lambda\mu\rho} [\alpha]\,t'$.
- (R-REN). Luego, $o = c[\alpha \setminus \beta]$ y $o' = c'[\alpha \setminus \beta]$ con $c \Rrightarrow_{\lambda\mu\rho} c'$. Por *h.i.* se tiene $c \twoheadrightarrow_{\lambda\mu\rho} c'$. Se concluye entonces $c[\alpha \setminus \beta] \twoheadrightarrow_{\lambda\mu\rho} c'[\alpha \setminus \beta]$.
- (R-PUSH). Luego, $o = t \cdot s$ y $o' = t' \cdot s'$ con $t \Rrightarrow_{\lambda\mu\rho} t'$ y $s \Rrightarrow_{\lambda\mu\rho} s'$. Por *h.i.* se tienen $t \twoheadrightarrow_{\lambda\mu\rho} t'$ y $s \twoheadrightarrow_{\lambda\mu\rho} s'$. Luego, $t \cdot s \twoheadrightarrow_{\lambda\mu\rho} t' \cdot s \twoheadrightarrow_{\lambda\mu\rho} t' \cdot s'$. Se concluye por transitividad de $\twoheadrightarrow_{\lambda\mu\rho}$.
- (R-BETA). Luego, $o = (\lambda x.t)\,u$ y $o' = \{x \setminus u'\}t'$ con $t \Rrightarrow_{\lambda\mu\rho} t'$ y $u \Rrightarrow_{\lambda\mu\rho} u'$. Por *h.i.* se tienen $t \twoheadrightarrow_{\lambda\mu\rho} t'$ y $u \twoheadrightarrow_{\lambda\mu\rho} u'$. Luego, $(\lambda x.t)\,u \twoheadrightarrow_{\lambda\mu\rho} (\lambda x.t')\,u \twoheadrightarrow_{\lambda\mu\rho} (\lambda x.t')\,u' \to_{\lambda\mu\rho} \{x \setminus u'\}t'$. Se concluye por transitividad de $\twoheadrightarrow_{\lambda\mu\rho}$.
- (R-MU). Luego, $o = (\mu\alpha.c) :: s$ y $o' = \mu\alpha'.\langle\!\langle \alpha \setminus^{\alpha'} s' \rangle\!\rangle c'$ con $c \Rrightarrow_{\lambda\mu\rho} c'$ y $s \Rrightarrow_{\lambda\mu\rho} s'$. Por *h.i.* se tienen $c \twoheadrightarrow_{\lambda\mu\rho} c'$ y $s \twoheadrightarrow_{\lambda\mu\rho} s'$. Luego, por transitividad de $\twoheadrightarrow_{\lambda\mu\rho}$ se tiene $(\mu\alpha.c) :: s \twoheadrightarrow_{\lambda\mu\rho} (\mu\alpha.c') :: s' \to_{\lambda\mu\rho} \mu\alpha'.\langle\!\langle \alpha \setminus^{\alpha'} s' \rangle\!\rangle c'$. Se concluye por transitividad de $\twoheadrightarrow_{\lambda\mu\rho}$.

□

La confluencia de $\to_{\lambda\mu\rho}$ es consecuencia inmediata del Lem. C.0.10 y el Lem. C.0.11.

Teorema C.0.12. *La relación de reducción $\to_{\lambda\mu\rho}$ es confluente (CR).*

Demostración. Dado que las clausuras son, por definición, funciones monótonas e idempotentes, del Lem. C.0.11 se sigue $\Rrightarrow^*_{\lambda\mu\rho} = \twoheadrightarrow_{\lambda\mu\rho}$. Luego, se satisface el siguiente diagrama de confluencia:



□

Confluencia del ΛM-cálculo

La confluencia del ΛM-cálculo se prueba mediante el método de interpretación de Hardin [CHL96], proyectando los objetos de ΛM en objetos del $\lambda\mu\rho$-cálculo.

Definición C.0.13. *La* proyección $\_^{\downarrow}$ *de objetos del ΛM-cálculo en objetos del $\lambda\mu\rho$-cálculo se define como*

$$\begin{array}{rcl} x^{\downarrow} & \triangleq & x \\ (t\,u)^{\downarrow} & \triangleq & t^{\downarrow}\,u^{\downarrow} \\ (\lambda x.t)^{\downarrow} & \triangleq & \lambda x.t^{\downarrow} \\ (\mu\alpha.c)^{\downarrow} & \triangleq & \mu\alpha.c^{\downarrow} \\ (t[x \setminus u])^{\downarrow} & \triangleq & \{x \setminus u^{\downarrow}\}t^{\downarrow} \end{array} \qquad \begin{array}{rcl} ([\alpha]\,t)^{\downarrow} & \triangleq & [\alpha]\,t^{\downarrow} \\ (c[\![\alpha \setminus^{\alpha'} s]\!])^{\downarrow} & \triangleq & \langle\!\langle \alpha \setminus^{\alpha'} s^{\downarrow} \rangle\!\rangle c^{\downarrow} \\ (c[\alpha \setminus \beta])^{\downarrow} & \triangleq & c^{\downarrow}[\alpha \setminus \beta] \\ (s \cdot t)^{\downarrow} & \triangleq & t^{\downarrow} \cdot s^{\downarrow} \end{array}$$

Lema C.0.14. *Sean $o \in \mathbb{O}_{\Lambda M}$ y $u \in \mathbb{T}_{\Lambda M}$. Luego, $(\{x \setminus u\}o)^{\downarrow} = \{x \setminus u^{\downarrow}\}o^{\downarrow}$.*

Demostración. Por inducción en o.

- $o = x$. Luego, se tiene $(\{x \setminus u\}o)^{\downarrow} = u^{\downarrow} = \{x \setminus u^{\downarrow}\}x = \{x \setminus u^{\downarrow}\}o^{\downarrow}$, por lo que se concluye.
- $o = y \neq x$. Luego, se tiene $(\{x \setminus u\}o)^{\downarrow} = y = \{x \setminus u^{\downarrow}\}y = \{x \setminus u^{\downarrow}\}o^{\downarrow}$, por lo que se concluye.
- $o = t\,v$. Luego, se tiene $(\{x \setminus u\}o)^{\downarrow} = (\{x \setminus u\}t)^{\downarrow}\,(\{x \setminus u\}v)^{\downarrow}$. Por *h.i.* se tienen $(\{x \setminus u\}t)^{\downarrow} = \{x \setminus u^{\downarrow}\}t^{\downarrow}$ y $(\{x \setminus u\}v)^{\downarrow} = \{x \setminus u^{\downarrow}\}v^{\downarrow}$. Finalmente, se tiene $(\{x \setminus u\}o)^{\downarrow} = (\{x \setminus u\}t)^{\downarrow}\,(\{x \setminus u\}v)^{\downarrow} = \{x \setminus u^{\downarrow}\}t^{\downarrow}\,\{x \setminus u^{\downarrow}\}v^{\downarrow} = \{x \setminus u^{\downarrow}\}o^{\downarrow}$, por lo que se concluye.
- $o = \lambda y.t$ con $y \notin u$. Luego, se tiene $(\{x \setminus u\}o)^{\downarrow} = \lambda y.(\{x \setminus u\}t)^{\downarrow}$. Por *h.i.* se tiene $(\{x \setminus u\}t)^{\downarrow} = \{x \setminus u^{\downarrow}\}t^{\downarrow}$. Se concluye entonces, $(\{x \setminus u\}o)^{\downarrow} = \lambda y.(\{x \setminus u\}t)^{\downarrow} = \lambda y.\{x \setminus u^{\downarrow}\}t^{\downarrow} = \{x \setminus u^{\downarrow}\}o^{\downarrow}$.
- $o = \mu\alpha.c$ con $\alpha \notin u$. Luego, se tiene $(\{x \setminus u\}o)^{\downarrow} = \mu\alpha.(\{x \setminus u\}c)^{\downarrow}$. Por *h.i.* se tiene $(\{x \setminus u\}c)^{\downarrow} = \{x \setminus u^{\downarrow}\}c^{\downarrow}$. Se concluye entonces, $(\{x \setminus u\}o)^{\downarrow} = \mu\alpha.(\{x \setminus u\}c)^{\downarrow} = \mu\alpha.\{x \setminus u^{\downarrow}\}c^{\downarrow} = \{x \setminus u^{\downarrow}\}o^{\downarrow}$.
- $o = t[y \setminus v]$ con $y \notin u$. Luego, por definición se tiene $(\{x \setminus u\}o)^{\downarrow} = ((\{x \setminus u\}t)[y \setminus \{x \setminus u\}v])^{\downarrow} = \{y \setminus (\{x \setminus u\}v)^{\downarrow}\}(\{x \setminus u\}t)^{\downarrow}$. Por *h.i.* se tienen $(\{x \setminus u\}t)^{\downarrow} = \{x \setminus u^{\downarrow}\}t^{\downarrow}$ y $(\{x \setminus u\}v)^{\downarrow} = \{x \setminus u^{\downarrow}\}v^{\downarrow}$. Finalmente, se tiene $(\{x \setminus u\}o)^{\downarrow} = \{y \setminus (\{x \setminus u\}v)^{\downarrow}\}(\{x \setminus u\}t)^{\downarrow} = \{y \setminus \{x \setminus u^{\downarrow}\}v^{\downarrow}\}\{x \setminus u^{\downarrow}\}t^{\downarrow} = \{x \setminus u^{\downarrow}\}\{y \setminus v^{\downarrow}\}t^{\downarrow} = \{x \setminus u^{\downarrow}\}(t^{\downarrow}[y \setminus v^{\downarrow}]) = \{x \setminus u^{\downarrow}\}o^{\downarrow}$, por lo que se concluye.
- $o = [\alpha]\,t$. Luego, se tiene $(\{x \setminus u\}o)^{\downarrow} = [\alpha]\,(\{x \setminus u\}t)^{\downarrow}$. Por *h.i.* se tiene $(\{x \setminus u\}t)^{\downarrow} = \{x \setminus u^{\downarrow}\}t^{\downarrow}$. Se concluye entonces dado que $(\{x \setminus u\}o)^{\downarrow} = [\alpha]\,(\{x \setminus u\}t)^{\downarrow} = [\alpha]\,\{x \setminus u^{\downarrow}\}t^{\downarrow} = \{x \setminus u^{\downarrow}\}o^{\downarrow}$.
- $o = c[\![\alpha \setminus^{\alpha'} s]\!]$ con $\alpha \notin u$. Luego, por definición se tiene $(\{x \setminus u\}o)^{\downarrow} = ((\{x \setminus u\}c)[\![\alpha \setminus^{\alpha'} \{x \setminus u\}s]\!])^{\downarrow} = \langle\!\langle \alpha \setminus^{\alpha'} (\{x \setminus u\}s)^{\downarrow} \rangle\!\rangle (\{x \setminus u\}c)^{\downarrow}$. Por *h.i.* $(\{x \setminus u\}c)^{\downarrow} = \{x \setminus u^{\downarrow}\}c^{\downarrow}$ y $(\{x \setminus u\}s)^{\downarrow} = \{x \setminus u^{\downarrow}\}s^{\downarrow}$. Más aún, por Lem. C.0.4 (2) con $\alpha \notin u$, se tiene $(\{x \setminus u\}o)^{\downarrow} = \langle\!\langle \alpha \setminus^{\alpha'} (\{x \setminus u\}s)^{\downarrow} \rangle\!\rangle (\{x \setminus u\}c)^{\downarrow} = \langle\!\langle \alpha \setminus^{\alpha'} \{x \setminus u^{\downarrow}\}s^{\downarrow} \rangle\!\rangle \{x \setminus u^{\downarrow}\}c^{\downarrow} = \{x \setminus u^{\downarrow}\}\langle\!\langle \alpha \setminus^{\alpha'} s^{\downarrow} \rangle\!\rangle c^{\downarrow}$. Finalmente, se concluye pues $(\{x \setminus u\}o)^{\downarrow} = \{x \setminus u^{\downarrow}\}\langle\!\langle \alpha \setminus^{\alpha'} s^{\downarrow} \rangle\!\rangle c^{\downarrow} = \{x \setminus u^{\downarrow}\}(c[\![\alpha \setminus^{\alpha'} s]\!])^{\downarrow} = \{x \setminus u^{\downarrow}\}o^{\downarrow}$.
- $o = c[\alpha \setminus \beta]$ con $\alpha \notin u$. Luego, se tiene $(\{x \setminus u\}o)^{\downarrow} = (\{x \setminus u\}c)^{\downarrow}[\alpha \setminus \beta]$. Por *h.i.* se tiene $(\{x \setminus u\}c)^{\downarrow} = \{x \setminus u^{\downarrow}\}c^{\downarrow}$. Se concluye entonces dado que $(\{x \setminus u\}o)^{\downarrow} = (\{x \setminus u\}c)^{\downarrow}[\alpha \setminus \beta] = (\{x \setminus u^{\downarrow}\}c^{\downarrow})[\alpha \setminus \beta] = \{x \setminus u^{\downarrow}\}o^{\downarrow}$.

- $o = t \cdot s$. Luego, se tiene $(\{x \setminus u\}o)^{\downarrow} = (\{x \setminus u\}t)^{\downarrow} \cdot (\{x \setminus u\}s)^{\downarrow}$. Por *h.i.* se tienen $(\{x \setminus u\}t)^{\downarrow} = \{x \setminus u^{\downarrow}\}t^{\downarrow}$ y $(\{x \setminus u\}s)^{\downarrow} = \{x \setminus u^{\downarrow}\}s^{\downarrow}$. Se concluye entonces dado que $(\{x \setminus u\}o)^{\downarrow} = (\{x \setminus u\}t)^{\downarrow} \cdot (\{x \setminus u\}s)^{\downarrow} = \{x \setminus u^{\downarrow}\}t^{\downarrow} \cdot \{x \setminus u^{\downarrow}\}s^{\downarrow} = \{x \setminus u^{\downarrow}\}o^{\downarrow}$.

□

Lema C.0.15. *Sean $o, o' \in \mathbb{O}_{\lambda\mu\rho}$ y $u, u' \in \mathbb{T}_{\lambda\mu\rho}$ tal que $o \to_{\lambda\mu\rho} o'$ y $u \to_{\lambda\mu\rho} u'$. Luego,*

1. $\{x \setminus u\}o \to_{\lambda\mu\rho} \{x \setminus u\}o'$.
2. $\{x \setminus u\}o \twoheadrightarrow_{\lambda\mu\rho} \{x \setminus u'\}o$.

Demostración. 1. Por definición $o \to_{\lambda\mu\rho} o'$ implica $o = \mathtt{O}\langle l\rangle$ y $o' = \mathtt{O}\langle r\rangle$ con $l \mapsto_* r$, $* \in \{\beta, \mu^s\}$. Se procede por inducción en $\mathtt{O}$.

- $\mathtt{O} = \Box$. Luego, se tienen dos casos posibles según la regla de reescritura aplicada al reducir.
  - *a)* β. Luego, $o = (\lambda y.t)\,v$ y $o' = \{y \setminus v\}t$. Por α-conversión se asume $y \notin u$. Entonces, $\{x \setminus u\}o = (\lambda y.\{x \setminus u\}t)\,\{x \setminus u\}v \to_{\lambda\mu\rho} \{y \setminus \{x \setminus u\}v\}\{x \setminus u\}t = \{x \setminus u\}\{y \setminus v\}t = \{x \setminus u\}o'$, por lo que se concluye.
  - *b)* μ^s. Luego, $o = (\mu\alpha.c) :: s$ y $o' = \mu\alpha'.\langle\!\langle \alpha \setminus^{\alpha'} s\rangle\!\rangle c$ con α' un nombre fresco. Por α-conversión se asume $\alpha \notin u$. Entonces, $\{x \setminus u\}o = (\mu\alpha.\{x \setminus u\}c) :: \{x \setminus u\}s \to_{\lambda\mu\rho} \mu\alpha'.\langle\!\langle \alpha \setminus^{\alpha'} \{x \setminus u\}s\rangle\!\rangle\{x \setminus u\}c$. Más aún, por Lem. C.0.4 (2), $\langle\!\langle \alpha \setminus^{\alpha'} \{x \setminus u\}s\rangle\!\rangle\{x \setminus u\}c = \{x \setminus u\}\langle\!\langle \alpha \setminus^{\alpha'} s\rangle\!\rangle c$. Finalmente, se tiene $\{x \setminus u\}o \to_{\lambda\mu\rho} \mu\alpha'.\langle\!\langle \alpha \setminus^{\alpha'} \{x \setminus u\}s\rangle\!\rangle\{x \setminus u\}c = \mu\alpha'.\{x \setminus u\}\langle\!\langle \alpha \setminus^{\alpha'} s\rangle\!\rangle c = \{x \setminus u\}o'$, por lo que se concluye.
- $\mathtt{O} = \boxdot$. Este caso no aplica dado que no hay reglas de reescritura sobre comandos.
- $\mathtt{O} = \mathtt{T}\,v$. Luego, $o = t\,v$ y $o' = t'\,v$ con $t \to_{\lambda\mu\rho} t'$. Por *h.i.* se tiene $\{x \setminus u\}t \to_{\lambda\mu\rho} \{x \setminus u\}t'$. Se concluye dado que $\{x \setminus u\}o = \{x \setminus u\}t\,\{x \setminus u\}v \to_{\lambda\mu\rho} \{x \setminus u\}t'\,\{x \setminus u\}v = \{x \setminus u\}o'$.
- $\mathtt{O} = t\,\mathtt{T}$. Luego, $o = t\,v$ y $o' = t\,v'$ con $v \to_{\lambda\mu\rho} v'$. Por *h.i.* se tiene $\{x \setminus u\}v \to_{\lambda\mu\rho} \{x \setminus u\}v'$. Se concluye dado que $\{x \setminus u\}o = \{x \setminus u\}t\,\{x \setminus u\}v \to_{\lambda\mu\rho} \{x \setminus u\}t\,\{x \setminus u\}v' = \{x \setminus u\}o'$.
- $\mathtt{O} = \lambda y.\mathtt{T}$. Luego, $o = \lambda y.t$ y $o' = \lambda y.t'$ con $t \to_{\lambda\mu\rho} t'$. Por α-conversión se asume $y \notin u$. Por *h.i.* se tiene $\{x \setminus u\}t \to_{\lambda\mu\rho} \{x \setminus u\}t'$. Se concluye dado que $\{x \setminus u\}o = \lambda y.\{x \setminus u\}t \to_{\lambda\mu\rho} \lambda y.\{x \setminus u\}t' = \{x \setminus u\}o'$.
- $\mathtt{O} = \mu\alpha.\mathtt{C}$. Luego, $o = \mu\alpha.c$ y $o' = \mu\alpha.c'$ con $c \to_{\lambda\mu\rho} c'$. Por α-conversión se asume $\alpha \notin u$. Por *h.i.* se tiene $\{x \setminus u\}c \to_{\lambda\mu\rho} \{x \setminus u\}c'$. Se concluye dado que $\{x \setminus u\}o = \mu\alpha.\{x \setminus u\}c \to_{\lambda\mu\rho} \mu\alpha.\{x \setminus u\}c' = \{x \setminus u\}o'$.
- $\mathtt{O} = [\alpha]\,\mathtt{T}$. Luego, $o = [\alpha]\,t$ y $o' = [\alpha]\,t'$ con $t \to_{\lambda\mu\rho} t'$. Por *h.i.* se tiene $\{x \setminus u\}t \to_{\lambda\mu\rho} \{x \setminus u\}t'$. Se concluye dado que $\{x \setminus u\}o = [\alpha]\,\{x \setminus u\}t \to_{\lambda\mu\rho} [\alpha]\,\{x \setminus u\}t' = \{x \setminus u\}o'$.
- $\mathtt{O} = \mathtt{C}[\alpha \setminus \beta]$. Luego, $o = c[\alpha \setminus \beta]$ y $o' = c'[\alpha \setminus \beta]$ con $c \to_{\lambda\mu\rho} c'$. Por α-conversión se asume $\alpha \notin u$. Por *h.i.* se tiene $\{x \setminus u\}c \to_{\lambda\mu\rho} \{x \setminus u\}c'$. Se concluye dado que $\{x \setminus u\}o = \{x \setminus u\}c[\alpha \setminus \beta] \to_{\lambda\mu\rho} \{x \setminus u\}c'[\alpha \setminus \beta] = \{x \setminus u\}o'$.
- $\mathtt{O} = \mathtt{T} \cdot s$. Luego, $o = t \cdot s$ y $o' = t' \cdot s$ con $t \to_{\lambda\mu\rho} t'$. Por *h.i.* se tiene $\{x \setminus u\}t \to_{\lambda\mu\rho} \{x \setminus u\}t'$. Se concluye dado que $\{x \setminus u\}o = \{x \setminus u\}t \cdot \{x \setminus u\}s \to_{\lambda\mu\rho} \{x \setminus u\}t' \cdot \{x \setminus u\}s = \{x \setminus u\}o'$.
- $\mathtt{O} = t \cdot \mathtt{S}$. Luego, $o = t \cdot s$ y $o' = t \cdot s'$ con $s \to_{\lambda\mu\rho} s'$. Por *h.i.* se tiene $\{x \setminus u\}s \to_{\lambda\mu\rho} \{x \setminus u\}s'$. Se concluye dado que $\{x \setminus u\}o = \{x \setminus u\}t \cdot \{x \setminus u\}s \to_{\lambda\mu\rho} \{x \setminus u\}t \cdot \{x \setminus u\}s' = \{x \setminus u\}o'$.

2. Por inducción en o.

- $o = x$. Luego, $\{x \setminus u\}o = u \to_{\lambda\mu\rho} u' = \{x \setminus u'\}o$, por lo que se concluye.

- $o = y \neq x$. Luego, $\{x \setminus u\}o = y = \{x \setminus u'\}o$. Se concluye por reflexividad de $\twoheadrightarrow_{\lambda\mu\rho}$.
- $o = t\,v$. Luego, $\{x \setminus u\}o = \{x \setminus u\}t\,\{x \setminus u\}v$. Por *h.i.* se tienen $\{x \setminus u\}t \twoheadrightarrow_{\lambda\mu\rho} \{x \setminus u'\}t$ y $\{x \setminus u\}v \twoheadrightarrow_{\lambda\mu\rho} \{x \setminus u'\}v$. Finalmente, se concluye por transitividad de $\twoheadrightarrow_{\lambda\mu\rho}$, dado que $\{x \setminus u\}o \twoheadrightarrow_{\lambda\mu\rho} \{x \setminus u'\}t\,\{x \setminus u\}v \twoheadrightarrow_{\lambda\mu\rho} \{x \setminus u'\}t\,\{x \setminus u'\}v = \{x \setminus u'\}o$.
- $o = \lambda y.t$ con $y \notin u$. Luego, $\{x \setminus u\}o = \lambda y.\{x \setminus u\}t$. Por *h.i.* se tiene $\{x \setminus u\}t \twoheadrightarrow_{\lambda\mu\rho} \{x \setminus u'\}t$. Finalmente, se concluye dado que $\{x \setminus u\}o = \lambda y.\{x \setminus u\}t \twoheadrightarrow_{\lambda\mu\rho} \lambda y.\{x \setminus u'\}t = \{x \setminus u'\}o$.
- $o = \mu\alpha.c$ con $\alpha \notin u$. Luego, $\{x \setminus u\}o = \mu\alpha.\{x \setminus u\}c$. Por *h.i.* se tiene $\{x \setminus u\}c \twoheadrightarrow_{\lambda\mu\rho} \{x \setminus u'\}c$. Finalmente, se concluye dado que $\{x \setminus u\}o = \mu\alpha.\{x \setminus u\}c \twoheadrightarrow_{\lambda\mu\rho} \mu\alpha.\{x \setminus u'\}c = \{x \setminus u'\}o$.
- $o = [\alpha]\,t$. Luego, $\{x \setminus u\}o = [\alpha]\,\{x \setminus u\}t$. Por *h.i.* se tiene $\{x \setminus u\}t \twoheadrightarrow_{\lambda\mu\rho} \{x \setminus u'\}t$. Finalmente, se concluye dado que $\{x \setminus u\}o = [\alpha]\,\{x \setminus u\}t \twoheadrightarrow_{\lambda\mu\rho} [\alpha]\,\{x \setminus u'\}t = \{x \setminus u'\}o$.
- $o = c[\alpha \setminus \beta]$ con $\alpha \notin u$. Luego, $\{x \setminus u\}o = \{x \setminus u\}c[\alpha \setminus \beta]$. Por *h.i.* se tiene $\{x \setminus u\}c \twoheadrightarrow_{\lambda\mu\rho} \{x \setminus u'\}c$. Finalmente, se concluye dado que $\{x \setminus u\}o = \{x \setminus u\}c[\alpha\setminus\beta] \twoheadrightarrow_{\lambda\mu\rho} \{x \setminus u'\}c[\alpha\setminus\beta] = \{x \setminus u'\}o$.
- $o = t \cdot s$. Luego, $\{x \setminus u\}o = \{x \setminus u\}t \cdot \{x \setminus u\}s$. Por *h.i.* se tienen $\{x \setminus u\}t \twoheadrightarrow_{\lambda\mu\rho} \{x \setminus u'\}t$ y $\{x \setminus u\}s \twoheadrightarrow_{\lambda\mu\rho} \{x \setminus u'\}s$. Finalmente, se concluye por transitividad de $\twoheadrightarrow_{\lambda\mu\rho}$, dado que $\{x \setminus u\}o \twoheadrightarrow_{\lambda\mu\rho} \{x \setminus u'\}t \cdot \{x \setminus u\}s \twoheadrightarrow_{\lambda\mu\rho} \{x \setminus u'\}t \cdot \{x \setminus u'\}s = \{x \setminus u'\}o$.

□

Lema C.0.16. *Sean $o \in \mathbb{O}_{\Lambda M}$ y s un stack. Luego, $(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o)^{\downarrow} = \langle\!\langle \alpha \setminus^{\alpha'} s^{\downarrow} \rangle\!\rangle o^{\downarrow}$.*

Demostración. Por inducción en o.

- $o = x$. Luego, se tiene $(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o)^{\downarrow} = x = \langle\!\langle \alpha \setminus^{\alpha'} s^{\downarrow} \rangle\!\rangle x = \langle\!\langle \alpha \setminus^{\alpha'} s^{\downarrow} \rangle\!\rangle o^{\downarrow}$, por lo que se concluye.
- $o = t\,u$. Luego, por definición se tiene $(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o)^{\downarrow} = (\{\!\{\alpha \setminus^{\alpha'} s\}\!\}t)^{\downarrow}\,(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}u)^{\downarrow}$. Por *h.i.* se tienen $(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}t)^{\downarrow} = \langle\!\langle \alpha \setminus^{\alpha'} s^{\downarrow} \rangle\!\rangle t^{\downarrow}$ y $(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}u)^{\downarrow} = \langle\!\langle \alpha \setminus^{\alpha'} s^{\downarrow} \rangle\!\rangle u^{\downarrow}$. Finalmente, se concluye dado que $(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o)^{\downarrow} = (\{\!\{\alpha \setminus^{\alpha'} s\}\!\}t)^{\downarrow}\,(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}u)^{\downarrow} = \langle\!\langle \alpha \setminus^{\alpha'} s^{\downarrow} \rangle\!\rangle t^{\downarrow}\,\langle\!\langle \alpha \setminus^{\alpha'} s^{\downarrow} \rangle\!\rangle u^{\downarrow} = \langle\!\langle \alpha \setminus^{\alpha'} s^{\downarrow} \rangle\!\rangle o^{\downarrow}$.
- $o = \lambda x.t$ con $x \notin s$. Luego, por definición se tiene $(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o)^{\downarrow} = \lambda x.(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}t)^{\downarrow}$. Por *h.i.* se tiene $(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}t)^{\downarrow} = \langle\!\langle \alpha \setminus^{\alpha'} s^{\downarrow} \rangle\!\rangle t^{\downarrow}$. Se concluye entonces, $(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o)^{\downarrow} = \lambda x.(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}t)^{\downarrow} = \lambda x.\langle\!\langle \alpha \setminus^{\alpha'} s^{\downarrow} \rangle\!\rangle t^{\downarrow} = \langle\!\langle \alpha \setminus^{\alpha'} s^{\downarrow} \rangle\!\rangle o^{\downarrow}$.
- $o = \mu\gamma.c$ con $\gamma \notin s$ y $\gamma \neq \alpha'$. Luego, se tiene $(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o)^{\downarrow} = \mu\gamma.(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}c)^{\downarrow}$. Por *h.i.* se tiene $(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}c)^{\downarrow} = \langle\!\langle \alpha \setminus^{\alpha'} s^{\downarrow} \rangle\!\rangle c^{\downarrow}$. Se concluye entonces, $(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o)^{\downarrow} = \mu\gamma.(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}c)^{\downarrow} = \mu\gamma.\langle\!\langle \alpha \setminus^{\alpha'} s^{\downarrow} \rangle\!\rangle c^{\downarrow} = \langle\!\langle \alpha \setminus^{\alpha'} s^{\downarrow} \rangle\!\rangle o^{\downarrow}$.
- $o = t[x \setminus u]$ con $x \notin s$. Luego, por definición $(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o)^{\downarrow} = ((\{\!\{\alpha \setminus^{\alpha'} s\}\!\}t)[x \setminus \{\!\{\alpha \setminus^{\alpha'} s\}\!\}u])^{\downarrow} = \{x \setminus (\{\!\{\alpha \setminus^{\alpha'} s\}\!\}u)^{\downarrow}\}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}t)^{\downarrow}$. Por *h.i.* se tienen $(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}t)^{\downarrow} = \langle\!\langle \alpha \setminus^{\alpha'} s^{\downarrow} \rangle\!\rangle t^{\downarrow}$ y $(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}u)^{\downarrow} = \langle\!\langle \alpha \setminus^{\alpha'} s^{\downarrow} \rangle\!\rangle u^{\downarrow}$. Entonces, por Lem. C.0.4 (1), $(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o)^{\downarrow} = \{x \setminus (\{\!\{\alpha \setminus^{\alpha'} s\}\!\}u)^{\downarrow}\}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}t)^{\downarrow} = \{x \setminus \langle\!\langle \alpha \setminus^{\alpha'} s^{\downarrow} \rangle\!\rangle u^{\downarrow}\}\langle\!\langle \alpha \setminus^{\alpha'} s^{\downarrow} \rangle\!\rangle t^{\downarrow} = \langle\!\langle \alpha \setminus^{\alpha'} s^{\downarrow} \rangle\!\rangle \{x \setminus u^{\downarrow}\}t^{\downarrow}$. Finalmente, se concluye $(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o)^{\downarrow} = \langle\!\langle \alpha \setminus^{\alpha'} s^{\downarrow} \rangle\!\rangle \{x \setminus u^{\downarrow}\}t^{\downarrow} = \langle\!\langle \alpha \setminus^{\alpha'} s^{\downarrow} \rangle\!\rangle (t^{\downarrow}[x \setminus u^{\downarrow}]) = \langle\!\langle \alpha \setminus^{\alpha'} s^{\downarrow} \rangle\!\rangle o^{\downarrow}$.
- $o = [\alpha]\,t$. Luego, se tiene $(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o)^{\downarrow} = ([\alpha']\,\{\!\{\alpha \setminus^{\alpha'} s\}\!\}t :: s)^{\downarrow} = [\alpha']\,(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}t)^{\downarrow} :: s^{\downarrow}$. Por *h.i.* se tiene $(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}t)^{\downarrow} = \langle\!\langle \alpha \setminus^{\alpha'} s^{\downarrow} \rangle\!\rangle t^{\downarrow}$. Finalmente, se concluye dado que $(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o)^{\downarrow} = [\alpha']\,(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}t)^{\downarrow} :: s^{\downarrow} = [\alpha]\,\langle\!\langle \alpha \setminus^{\alpha'} s^{\downarrow} \rangle\!\rangle t^{\downarrow} :: s^{\downarrow} = \langle\!\langle \alpha \setminus^{\alpha'} s^{\downarrow} \rangle\!\rangle o^{\downarrow}$.

- $o = [\delta]\, t$ con $\delta \neq \alpha$. Luego, por definición $(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o)^{\downarrow} = [\delta]\,(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} t)^{\downarrow}$. Por *h.i.* se tiene $(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} t)^{\downarrow} = \langle\!\langle \alpha \setminus^{\alpha'} s^{\downarrow} \rangle\!\rangle t^{\downarrow}$. Se concluye entonces dado que $(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o)^{\downarrow} = [\delta]\,(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} t)^{\downarrow} = [\delta]\,\langle\!\langle \alpha \setminus^{\alpha'} s^{\downarrow} \rangle\!\rangle t^{\downarrow} = \langle\!\langle \alpha \setminus^{\alpha'} s^{\downarrow} \rangle\!\rangle o^{\downarrow}$.

- $o = c[\![\gamma \setminus^{\alpha} s']\!]$ con $\gamma \notin s$ y $\gamma \neq \alpha'$. Luego, $(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o)^{\downarrow} = ((\{\!\{\alpha \setminus^{\alpha'} s\}\!\} c)[\![\gamma \setminus^{\alpha'} \{\!\{\alpha \setminus^{\alpha'} s\}\!\} s' \cdot s]\!])^{\downarrow} = \langle\!\langle \gamma \setminus^{\alpha'} (\{\!\{\alpha \setminus^{\alpha'} s\}\!\} s')^{\downarrow} \cdot s^{\downarrow} \rangle\!\rangle (\{\!\{\alpha \setminus^{\alpha'} s\}\!\} c)^{\downarrow}$. Por *h.i.* $(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} c)^{\downarrow} = \langle\!\langle \alpha \setminus^{\alpha'} s^{\downarrow} \rangle\!\rangle c^{\downarrow}$ y $(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} s')^{\downarrow} = \langle\!\langle \alpha \setminus^{\alpha'} s^{\downarrow} \rangle\!\rangle s'^{\downarrow}$. Por Lem. C.0.5, se tiene $(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o)^{\downarrow} = \langle\!\langle \gamma \setminus^{\alpha'} (\{\!\{\alpha \setminus^{\alpha'} s\}\!\} s')^{\downarrow} \cdot s^{\downarrow} \rangle\!\rangle (\{\!\{\alpha \setminus^{\alpha'} s\}\!\} c)^{\downarrow} = \langle\!\langle \gamma \setminus^{\alpha'} \langle\!\langle \alpha \setminus^{\alpha'} s^{\downarrow} \rangle\!\rangle s'^{\downarrow} \cdot s^{\downarrow} \rangle\!\rangle \langle\!\langle \alpha \setminus^{\alpha'} s^{\downarrow} \rangle\!\rangle c^{\downarrow} = \langle\!\langle \alpha \setminus^{\alpha'} s^{\downarrow} \rangle\!\rangle \langle\!\langle \gamma \setminus^{\alpha} s'^{\downarrow} \rangle\!\rangle c^{\downarrow}$. Finalmente, se concluye dado que $(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o)^{\downarrow} = \langle\!\langle \alpha \setminus^{\alpha'} s^{\downarrow} \rangle\!\rangle \langle\!\langle \gamma \setminus^{\alpha} s'^{\downarrow} \rangle\!\rangle c^{\downarrow} = \langle\!\langle \alpha \setminus^{\alpha'} s^{\downarrow} \rangle\!\rangle (c[\![\gamma \setminus^{\alpha} s]\!])^{\downarrow} = \langle\!\langle \alpha \setminus^{\alpha'} s^{\downarrow} \rangle\!\rangle o^{\downarrow}$.

- $o = c[\![\gamma \setminus^{\delta} s']\!]$ con $\gamma \notin s$, $\gamma \neq \alpha'$ y $\delta \neq \alpha$. Luego, por definición se tiene $(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o)^{\downarrow} = ((\{\!\{\alpha \setminus^{\alpha'} s\}\!\} c)[\![\gamma \setminus^{\delta} \{\!\{\alpha \setminus^{\alpha'} s\}\!\} s']\!])^{\downarrow} = \langle\!\langle \gamma \setminus^{\delta} (\{\!\{\alpha \setminus^{\alpha'} s\}\!\} s')^{\downarrow} \rangle\!\rangle (\{\!\{\alpha \setminus^{\alpha'} s\}\!\} c)^{\downarrow}$. Por *h.i.* $(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} c)^{\downarrow} = \langle\!\langle \alpha \setminus^{\alpha'} s^{\downarrow} \rangle\!\rangle c^{\downarrow}$ y $(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} s')^{\downarrow} = \langle\!\langle \alpha \setminus^{\alpha'} s^{\downarrow} \rangle\!\rangle s'^{\downarrow}$. Más aún, por Lem. C.0.5 se tiene $(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o)^{\downarrow} = \langle\!\langle \gamma \setminus^{\delta} (\{\!\{\alpha \setminus^{\alpha'} s\}\!\} s')^{\downarrow} \rangle\!\rangle (\{\!\{\alpha \setminus^{\alpha'} s\}\!\} c)^{\downarrow} = \langle\!\langle \gamma \setminus^{\delta} \langle\!\langle \alpha \setminus^{\alpha'} s^{\downarrow} \rangle\!\rangle s'^{\downarrow} \rangle\!\rangle \langle\!\langle \alpha \setminus^{\alpha'} s^{\downarrow} \rangle\!\rangle c^{\downarrow} = \langle\!\langle \alpha \setminus^{\alpha'} s^{\downarrow} \rangle\!\rangle \langle\!\langle \gamma \setminus^{\delta} s'^{\downarrow} \rangle\!\rangle c^{\downarrow}$. Finalmente, se concluye dado que $(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o)^{\downarrow} = \langle\!\langle \alpha \setminus^{\alpha'} s^{\downarrow} \rangle\!\rangle \langle\!\langle \gamma \setminus^{\delta} s'^{\downarrow} \rangle\!\rangle c^{\downarrow} = \langle\!\langle \alpha \setminus^{\alpha'} s^{\downarrow} \rangle\!\rangle (c[\![\gamma \setminus^{\delta} s']\!])^{\downarrow} = \langle\!\langle \alpha \setminus^{\alpha'} s^{\downarrow} \rangle\!\rangle o^{\downarrow}$.

- $o = c[\gamma \setminus \alpha]$ con $\gamma \notin s$ y $\gamma \neq \alpha'$. Luego, se tiene $(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o)^{\downarrow} = ((\{\!\{\alpha \setminus^{\alpha'} s\}\!\} c)[\![\gamma \setminus^{\delta} s]\!][\delta \setminus \alpha'])^{\downarrow} = (\langle\!\langle \gamma \setminus^{\delta} s^{\downarrow} \rangle\!\rangle (\{\!\{\alpha \setminus^{\alpha'} s\}\!\} c)^{\downarrow})[\delta \setminus \alpha']$ con δ un nombre fresco. Por *h.i.* $(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} c)^{\downarrow} = \langle\!\langle \alpha \setminus^{\alpha'} s^{\downarrow} \rangle\!\rangle c^{\downarrow}$. Más aún, por Lem. C.0.6, se tiene entonces $(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o)^{\downarrow} = (\langle\!\langle \gamma \setminus^{\delta} s^{\downarrow} \rangle\!\rangle (\{\!\{\alpha \setminus^{\alpha'} s\}\!\} c)^{\downarrow})[\delta \setminus \alpha'] = (\langle\!\langle \gamma \setminus^{\delta} s^{\downarrow} \rangle\!\rangle \langle\!\langle \alpha \setminus^{\alpha'} s^{\downarrow} \rangle\!\rangle c^{\downarrow})[\delta \setminus \alpha'] = (\langle\!\langle \gamma\alpha \setminus^{\delta\alpha'} s^{\downarrow} \rangle\!\rangle c^{\downarrow})[\delta \setminus \alpha']$. Finalmente, se concluye $(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o)^{\downarrow} = (\langle\!\langle \gamma\alpha \setminus^{\delta\alpha'} s^{\downarrow} \rangle\!\rangle c^{\downarrow})[\delta \setminus \alpha'] = \langle\!\langle \alpha \setminus^{\alpha'} s^{\downarrow} \rangle\!\rangle (c[\delta \setminus \alpha']^{\downarrow}) = \langle\!\langle \alpha \setminus^{\alpha'} s^{\downarrow} \rangle\!\rangle o^{\downarrow}$.

- $o = c[\gamma \setminus \delta]$ con $\gamma \notin s$, $\gamma \neq \alpha'$ y $\delta \neq \alpha$. Luego, se tiene $(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o)^{\downarrow} = (\{\!\{\alpha \setminus^{\alpha'} s\}\!\} c)^{\downarrow}[\gamma \setminus \delta]$. Por *h.i.* se tiene $(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} c)^{\downarrow} = \langle\!\langle \alpha \setminus^{\alpha'} s^{\downarrow} \rangle\!\rangle c^{\downarrow}$. Se concluye entonces dado que $(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o)^{\downarrow} = (\{\!\{\alpha \setminus^{\alpha'} s\}\!\} c)^{\downarrow}[\gamma \setminus \delta] = (\langle\!\langle \alpha \setminus^{\alpha'} s^{\downarrow} \rangle\!\rangle c^{\downarrow})[\gamma \setminus \delta] = \langle\!\langle \alpha \setminus^{\alpha'} s^{\downarrow} \rangle\!\rangle o^{\downarrow}$.

- $o = t \cdot s$. Luego, se tiene $(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o)^{\downarrow} = (\{\!\{\alpha \setminus^{\alpha'} s\}\!\} t)^{\downarrow} \cdot (\{\!\{\alpha \setminus^{\alpha'} s\}\!\} s)^{\downarrow}$. Por *h.i.* se tienen $(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} t)^{\downarrow} = \langle\!\langle \alpha \setminus^{\alpha'} s^{\downarrow} \rangle\!\rangle t^{\downarrow}$ y $(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} s)^{\downarrow} = \langle\!\langle \alpha \setminus^{\alpha'} s^{\downarrow} \rangle\!\rangle s^{\downarrow}$. Se concluye entonces dado que $(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o)^{\downarrow} = (\{\!\{\alpha \setminus^{\alpha'} s\}\!\} t)^{\downarrow} \cdot (\{\!\{\alpha \setminus^{\alpha'} s\}\!\} s)^{\downarrow} = \langle\!\langle \alpha \setminus^{\alpha'} s^{\downarrow} \rangle\!\rangle t^{\downarrow} \cdot \langle\!\langle \alpha \setminus^{\alpha'} s^{\downarrow} \rangle\!\rangle s^{\downarrow} = \langle\!\langle \alpha \setminus^{\alpha'} s^{\downarrow} \rangle\!\rangle o^{\downarrow}$.

□

Lema C.0.17. *Sean $o, o' \in \mathbb{O}_{\lambda\mu\rho}$ y s, s' stacks tal que $o \to_{\lambda\mu\rho} o'$ y $s \to_{\lambda\mu\rho} s'$. Luego,*

1. $\langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle o \to_{\lambda\mu\rho} \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle o'$.
2. $\langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle o \twoheadrightarrow_{\lambda\mu\rho} \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s' \rangle\!\rangle o$.

Demostración. 1. Por definición $o \to_{\lambda\mu\rho} o'$ implica $o = \mathtt{O}\langle l \rangle$ y $o' = \mathtt{O}\langle r \rangle$ con $l \mapsto_* r$, $* \in \{\beta, \mu^s\}$. Se procede por inducción en $\mathtt{O}$.

- $\mathtt{O} = \Box$. Luego, se tienen dos casos posibles según la regla de reescritura aplicada al reducir.

   a) β. Luego, $o = (\lambda x.t)\,u$ y $o' = \{x \setminus u\}t$. Por α-conversión se asume $x \notin s$. Entoces, $\langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle o = (\lambda x.\langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle t)\,\langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle u \to_{\lambda\mu\rho} \{x \setminus \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle u\}\langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle t$. Más aún, por Lem. C.0.4 (1), se tiene $\{x \setminus \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle u\}\langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle t = \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle\{x \setminus u\}t$. Finalmente, se tiene $\langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle o \to_{\lambda\mu\rho} \{x \setminus \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle u\}\langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle t = \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle\{x \setminus u\}t = \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle o'$, por lo que se concluye.

   b) μ^s. Luego, $o = (\mu\gamma.c) :: s''$ y $o' = \mu\delta.\langle\!\langle \gamma \setminus^{\delta} s'' \rangle\!\rangle c$ con δ un nombre fresco. Por α-conversión se asume $\gamma \notin s$. Entonces, por definición $\langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle o = (\mu\gamma.\langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle c) :: \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle s'' \to_{\lambda\mu\rho} \mu\delta.\langle\!\langle \gamma \setminus^{\delta} \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle s'' \rangle\!\rangle\langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle c$. Más aún, por Lem. C.0.5, se tiene $\langle\!\langle \gamma \setminus^{\delta} \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle s'' \rangle\!\rangle\langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle c = \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle\langle\!\langle \gamma \setminus^{\delta} s'' \rangle\!\rangle c$. Finalmente, se tiene $\langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle o \to_{\lambda\mu\rho} \mu\delta.\langle\!\langle \gamma \setminus^{\delta} \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle s'' \rangle\!\rangle\langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle c = \mu\delta.\langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle\langle\!\langle \gamma \setminus^{\delta} s'' \rangle\!\rangle c = \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle o'$, por lo que se concluye.

- $\mathtt{O} = \boxdot$. Este caso no aplica dado que no hay reglas de reescritura sobre comandos.
- $\mathtt{O} = \mathtt{T}\,u$. Luego, $o = t\,u$ y $o' = t'\,u$ con $t \to_{\lambda\mu\rho} t'$. Por *h.i.* se tiene $\langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle t \to_{\lambda\mu\rho} \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle t'$. Finalmente, $\langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle o = \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle t\,\langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle u \to_{\lambda\mu\rho} \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle t'\,\langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle u = \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle o'$, por lo que se concluye.
- $\mathtt{O} = t\,\mathtt{T}$. Luego, $o = t\,u$ y $o' = t\,u'$ con $u \to_{\lambda\mu\rho} u'$. Por *h.i.* se tiene $\langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle u \to_{\lambda\mu\rho} \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle u'$. Finalmente, $\langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle o = \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle t\,\langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle u \to_{\lambda\mu\rho} \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle t\,\langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle u' = \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle o'$, por lo que se concluye.
- $\mathtt{O} = \lambda x.\mathtt{T}$. Luego, $o = \lambda x.t$ y $o' = \lambda x.t'$ con $t \to_{\lambda\mu\rho} t'$. Por α-conversión se asume $x \notin s$. Por *h.i.* se tiene $\langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle t \to_{\lambda\mu\rho} \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle t'$. Se concluye dado que $\langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle o = \lambda x.\langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle t \to_{\lambda\mu\rho} \lambda x.\langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle t' = \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle o'$.
- $\mathtt{O} = \mu\gamma.\mathtt{C}$. Luego, $o = \mu\gamma.c$ y $o' = \mu\gamma.c'$ con $c \to_{\lambda\mu\rho} c'$. Por α-conversión se asumen $\gamma \notin s$ y $\gamma \notin \vec{\alpha'}$. Por *h.i.* se tiene $\langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle c \to_{\lambda\mu\rho} \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle c'$. Se concluye dado que $\langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle o = \mu\gamma.\langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle c \to_{\lambda\mu\rho} \mu\gamma.\langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle c' = \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle o'$.
- $\mathtt{O} = [\delta]\,\mathtt{T}$. Luego, $o = [\delta]\,t$ y $o' = [\delta]\,t'$ con $t \to_{\lambda\mu\rho} t'$. Se tienen dos casos posibles:

   a) $\delta \in \vec{\alpha}$. Luego, $\delta = \alpha_k$ para algún $k \in [1, \mathsf{sz}(\vec{\alpha})]$. Por *h.i.* se tiene $\langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle t \to_{\lambda\mu\rho} \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle t'$. Entonces, se tiene $\langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle o = [\beta_k]\,\langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle t :: s \to_{\lambda\mu\rho} [\beta_k]\,\langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle t' :: s = \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle o'$, por lo que se concluye.

   b) $\delta \notin \vec{\alpha}$. Luego, por *h.i.* se tiene $\langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle t \to_{\lambda\mu\rho} \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle t'$ y se concluye dado que $\langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle o = [\delta]\,\langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle t \to_{\lambda\mu\rho} [\delta]\,\langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle t' = \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle o'$.
- $\mathtt{O} = \mathtt{C}[\gamma \setminus \delta]$. Luego, $o = c[\gamma \setminus \delta]$ y $o' = c'[\gamma \setminus \delta]$ con $c \to_{\lambda\mu\rho} c'$. Por α-conversión se asumen $\gamma \notin s$ y $\gamma \notin \vec{\alpha'}$. Se tienen dos casos posibles:

   a) $\delta \in \vec{\alpha}$. Luego, $\delta = \alpha_k$ para algún $k \in [1, \mathsf{sz}(\vec{\alpha})]$. Por *h.i.* se tiene $\langle\!\langle \gamma\vec{\alpha} \setminus^{\gamma'\vec{\alpha'}} s \rangle\!\rangle c \to_{\lambda\mu\rho} \langle\!\langle \gamma\vec{\alpha} \setminus^{\gamma'\vec{\alpha'}} s \rangle\!\rangle c'$ con γ' un nombre fresco. Finalmente, $\langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle o = (\langle\!\langle \gamma\vec{\alpha} \setminus^{\gamma'\vec{\alpha'}} s \rangle\!\rangle c)[\gamma' \setminus \beta_k] \to_{\lambda\mu\rho} (\langle\!\langle \gamma\vec{\alpha} \setminus^{\gamma'\vec{\alpha'}} s \rangle\!\rangle c')[\gamma' \setminus \beta_k] = \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle o'$, por lo que se concluye.

   b) $\delta \notin \vec{\alpha}$. Luego, por *h.i.* se tiene $\langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle c \to_{\lambda\mu\rho} \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle c'$ y se concluye dado que $\langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle o = (\langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle c)[\gamma \setminus \delta] \to_{\lambda\mu\rho} (\langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle c')[\gamma \setminus \delta] = \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle o'$.
- $\mathtt{O} = \mathtt{T} \cdot s''$. Luego, $o = t \cdot s''$ y $o' = t \cdot s''$ con $t \to_{\lambda\mu\rho} t'$. Por *h.i.* se tiene $\langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle t \to_{\lambda\mu\rho} \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle t'$. Finalmente, $\langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle o = \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle t \cdot \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle s'' \to_{\lambda\mu\rho} \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle t' \cdot \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle s'' = \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle o'$, por lo que se concluye.

- $\mathtt{0} = t \cdot \mathtt{S}$. Luego, $o = t \cdot s''$ y $o' = t \cdot s'''$ con $s'' \to_{\lambda\mu\rho} s'''$. Por *h.i.* se tiene $\langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle s'' \to_{\lambda\mu\rho} \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle s'''$. Finalmente, $\langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle o = \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle t \cdot \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle s'' \to_{\lambda\mu\rho} \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle t \cdot \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle s''' = \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle o'$, por lo que se concluye.

2. Por inducción en o.

- $o = x$. Luego, $\langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle o = x = \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s' \rangle\!\rangle o$. Se concluye por reflexividad de $\twoheadrightarrow_{\lambda\mu\rho}$.
- $o = t\,u$. Luego, $\langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle o = \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle t \, \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle u$. Por *h.i.* se tienen $\langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle t \twoheadrightarrow_{\lambda\mu\rho} \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s' \rangle\!\rangle t$ y $\langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle u \twoheadrightarrow_{\lambda\mu\rho} \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s' \rangle\!\rangle u$. Finalmente, se concluye por transitividad de $\twoheadrightarrow_{\lambda\mu\rho}$, dado que $\langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle o \twoheadrightarrow_{\lambda\mu\rho} \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s' \rangle\!\rangle t \, \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle u \twoheadrightarrow_{\lambda\mu\rho} \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s' \rangle\!\rangle t \, \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s' \rangle\!\rangle u = \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s' \rangle\!\rangle o$.
- $o = \lambda x.t$ con $x \notin s$. Luego, $\langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle o = \lambda x. \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle t$. Por *h.i.* se tiene $\langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle t \twoheadrightarrow_{\lambda\mu\rho} \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s' \rangle\!\rangle t$. Entonces, se tiene $\langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle o = \lambda x. \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle t \twoheadrightarrow_{\lambda\mu\rho} \lambda x. \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s' \rangle\!\rangle t = \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s' \rangle\!\rangle o$, por lo que se concluye.
- $o = \mu\gamma.c$ con $\gamma \notin s$ y $\gamma \notin \vec{\alpha'}$. Luego, $\langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle o = \mu\alpha. \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle c$. Por *h.i.* se tiene $\langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle c \twoheadrightarrow_{\lambda\mu\rho} \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s' \rangle\!\rangle c$. Entonces, se tiene $\langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle o = \mu\alpha. \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle c \twoheadrightarrow_{\lambda\mu\rho} \mu\alpha. \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s' \rangle\!\rangle c = \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s' \rangle\!\rangle o$, por lo que se concluye.
- $o = [\delta]\,t$. Luego, se tienen dos casos posibles:
  a) $\delta \in \vec{\alpha}$. Luego, $\delta = \alpha_k$ para algún $k \in [1, \mathsf{sz}(\vec{\alpha})]$. Más aún, por definición $\langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle o = [\beta_k] \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle t :: s$. Por *h.i.* se tiene $\langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle t \twoheadrightarrow_{\lambda\mu\rho} \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s' \rangle\!\rangle t$. Finalmente, se concluye por transitividad de $\twoheadrightarrow_{\lambda\mu\rho}$, dado que $\langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle o \twoheadrightarrow_{\lambda\mu\rho} [\beta_k] \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s' \rangle\!\rangle t :: s \to_{\lambda\mu\rho} [\beta_k] \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s' \rangle\!\rangle t :: s' = \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s' \rangle\!\rangle o$.
  b) $\delta \notin \vec{\alpha}$. Luego, $\langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle o = [\delta] \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle t$. Por *h.i.* se tiene $\langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle t \twoheadrightarrow_{\lambda\mu\rho} \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s' \rangle\!\rangle t$. Finalmente, se concluye dado que $\langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle o = [\delta] \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle t \twoheadrightarrow_{\lambda\mu\rho} [\delta] \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s' \rangle\!\rangle t = \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s' \rangle\!\rangle o$.
- $o = c[\gamma \setminus \delta]$ con $\gamma \notin s$ y $\gamma \notin \vec{\alpha'}$. Luego, se tienen dos casos posibles:
  a) $\delta \in \vec{\alpha}$. Luego, $\delta = \alpha_k$ para algún $k \in [1, \mathsf{sz}(\vec{\alpha})]$. Más aún, por definición $\langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle o = (\langle\!\langle \gamma\vec{\alpha} \setminus^{\gamma'\vec{\alpha'}} s \rangle\!\rangle c)[\gamma' \setminus \beta_k]$ con γ' un nombre fresco. Por *h.i.* se tiene $\langle\!\langle \gamma\vec{\alpha} \setminus^{\gamma'\vec{\alpha'}} s \rangle\!\rangle c \twoheadrightarrow_{\lambda\mu\rho} \langle\!\langle \gamma\vec{\alpha} \setminus^{\gamma'\vec{\alpha'}} s' \rangle\!\rangle c$. Finalmente, se concluye dado que $\langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle o = (\langle\!\langle \gamma\vec{\alpha} \setminus^{\gamma'\vec{\alpha'}} s \rangle\!\rangle c)[\gamma' \setminus \beta_k] \to_{\lambda\mu\rho} (\langle\!\langle \gamma\vec{\alpha} \setminus^{\gamma'\vec{\alpha'}} s' \rangle\!\rangle c)[\gamma' \setminus \beta_k] = \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s' \rangle\!\rangle o$.
  b) $\delta \notin \vec{\alpha}$. Luego, $\langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle o = (\langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle c)[\gamma \setminus \delta]$. Por *h.i.* se tiene $\langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle c \twoheadrightarrow_{\lambda\mu\rho} \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s' \rangle\!\rangle c$. Finalmente, se concluye dado que $\langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle o = (\langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle c)[\gamma \setminus \delta] \twoheadrightarrow_{\lambda\mu\rho} (\langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s' \rangle\!\rangle c)[\gamma \setminus \delta] = \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s' \rangle\!\rangle o$.
- $o = t \cdot s''$. Luego, $\langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle o = \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle t \cdot \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle s''$. Por *h.i.* se tienen $\langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle t \twoheadrightarrow_{\lambda\mu\rho} \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s' \rangle\!\rangle t$ y $\langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle s'' \twoheadrightarrow_{\lambda\mu\rho} \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s' \rangle\!\rangle s''$. Finalmente, se concluye por transitividad de $\twoheadrightarrow_{\lambda\mu\rho}$, dado que $\langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle o \twoheadrightarrow_{\lambda\mu\rho} \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s' \rangle\!\rangle t \cdot \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle s'' \twoheadrightarrow_{\lambda\mu\rho} \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s' \rangle\!\rangle t \cdot \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s' \rangle\!\rangle s'' = \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s' \rangle\!\rangle o$.

□

Dado un stack s de objetos del $\lambda\mu\rho$-cálculo y secuencias de nombres diferentes $\vec{\alpha}$ y $\vec{\alpha'}$ tales que $\mathsf{sz}(\alpha) = \mathsf{sz}(\alpha') = n$, se denota con $(\{\!\{\alpha_i \setminus^{\alpha'_i} s\}\!\})_{i \leq n}$ a la secuencia de replacements individuales sobre cada nombre de la secuencia $\vec{\alpha}$ (en el mismo orden), $\{\!\{\alpha_1 \setminus^{\alpha'_1} s\}\!\} \ldots \{\!\{\alpha_n \setminus^{\alpha'_n} s\}\!\}$.

Lema C.0.18. *Sean $o \in \mathbb{O}_{\lambda\mu\rho}$, s un stack, $\vec{\alpha}$ y $\vec{\alpha'}$ secuencias de nombres diferentes tales que $\vec{\alpha} \cap \mathsf{fn}(s) = \emptyset$. Luego, $(\{\!\{\alpha_i \setminus^{\alpha'_i} s\}\!\})_{i \leq \mathsf{sz}(\vec{\alpha})} o \twoheadrightarrow_{\Lambda M} \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle o$.*

Demostración. Por inducción en o.

- $o = x$. Luego, $(\{\!\{\alpha_i \setminus^{\alpha'_i} s\}\!\})_{i \leq \mathsf{sz}(\vec{\alpha})} o = x = \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle o$. Se concluye por reflexividad de $\twoheadrightarrow_{\Lambda M}$.
- $o = t\,u$. Luego, $(\{\!\{\alpha_i \setminus^{\alpha'_i} s\}\!\})_{i \leq \mathsf{sz}(\vec{\alpha})} o = (\{\!\{\alpha_i \setminus^{\alpha'_i} s\}\!\})_{i \leq \mathsf{sz}(\vec{\alpha})} t\,(\{\!\{\alpha_i \setminus^{\alpha'_i} s\}\!\})_{i \leq \mathsf{sz}(\vec{\alpha})} u$. Por *h.i.* se tienen $(\{\!\{\alpha_i \setminus^{\alpha'_i} s\}\!\})_{i \leq \mathsf{sz}(\vec{\alpha})} t \twoheadrightarrow_{\Lambda M} \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle t$ y $(\{\!\{\alpha_i \setminus^{\alpha'_i} s\}\!\})_{i \leq \mathsf{sz}(\vec{\alpha})} u \twoheadrightarrow_{\Lambda M} \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle u$. Finalmente, se tiene $(\{\!\{\alpha_i \setminus^{\alpha'_i} s\}\!\})_{i \leq \mathsf{sz}(\vec{\alpha})} o \twoheadrightarrow_{\Lambda M} \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle t\,(\{\!\{\alpha_i \setminus^{\alpha'_i} s\}\!\})_{i \leq \mathsf{sz}(\vec{\alpha})} u \twoheadrightarrow_{\Lambda M} \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle t \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle u = \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle o$. Se concluye por transitividad de $\twoheadrightarrow_{\Lambda M}$.
- $o = \lambda x.t$ con $x \notin s$. Luego, $(\{\!\{\alpha_i \setminus^{\alpha'_i} s\}\!\})_{i \leq \mathsf{sz}(\vec{\alpha})} o = \lambda x.(\{\!\{\alpha_i \setminus^{\alpha'_i} s\}\!\})_{i \leq \mathsf{sz}(\vec{\alpha})} t$. Por *h.i.* se tiene $(\{\!\{\alpha_i \setminus^{\alpha'_i} s\}\!\})_{i \leq \mathsf{sz}(\vec{\alpha})} t \twoheadrightarrow_{\Lambda M} \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle t$. Finalmente, se concluye dado que $(\{\!\{\alpha_i \setminus^{\alpha'_i} s\}\!\})_{i \leq \mathsf{sz}(\vec{\alpha})} o = \lambda x.(\{\!\{\alpha_i \setminus^{\alpha'_i} s\}\!\})_{i \leq \mathsf{sz}(\vec{\alpha})} t \twoheadrightarrow_{\Lambda M} \lambda x.\langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle t = \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle o$.
- $o = \mu\gamma.c$ con $\gamma \notin s$ y $\gamma \notin \vec{\alpha'}$. Luego, $(\{\!\{\alpha_i \setminus^{\alpha'_i} s\}\!\})_{i \leq \mathsf{sz}(\vec{\alpha})} o = \mu\alpha.(\{\!\{\alpha_i \setminus^{\alpha'_i} s\}\!\})_{i \leq \mathsf{sz}(\vec{\alpha})} c$. Por *h.i.* se tiene $(\{\!\{\alpha_i \setminus^{\alpha'_i} s\}\!\})_{i \leq \mathsf{sz}(\vec{\alpha})} c \twoheadrightarrow_{\Lambda M} \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle c$. Entonces, se concluye dado que $(\{\!\{\alpha_i \setminus^{\alpha'_i} s\}\!\})_{i \leq \mathsf{sz}(\vec{\alpha})} o = \mu\alpha.(\{\!\{\alpha_i \setminus^{\alpha'_i} s\}\!\})_{i \leq \mathsf{sz}(\vec{\alpha})} c \twoheadrightarrow_{\Lambda M} \mu\alpha.\langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle c = \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle o$.
- $o = [\delta]\,t$. Luego, se tienen dos casos posibles:
  1. $\delta \in \vec{\alpha}$. Luego, $\delta = \alpha_k$ para algún $k \in [1, \mathsf{sz}(\vec{\alpha})]$. Por definición se tiene $(\{\!\{\alpha_i \setminus^{\alpha'_i} s\}\!\})_{i \leq \mathsf{sz}(\vec{\alpha})} o = [\beta_k]\,(\{\!\{\alpha_i \setminus^{\alpha'_i} s\}\!\})_{i \leq \mathsf{sz}(\vec{\alpha})} t :: (\{\!\{\alpha_i \setminus^{\alpha'_i} s\}\!\})_{k < i \leq \mathsf{sz}(\vec{\alpha})} s$. Más aún, por hipótesis $\vec{\alpha} \cap \mathsf{fn}(s) = \emptyset$, lo que implica $(\{\!\{\alpha_i \setminus^{\alpha'_i} s\}\!\})_{k < i \leq \mathsf{sz}(\vec{\alpha})} s = s$. Por *h.i.* se tiene $(\{\!\{\alpha_i \setminus^{\alpha'_i} s\}\!\})_{i \leq \mathsf{sz}(\vec{\alpha})} t \twoheadrightarrow_{\Lambda M} \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle t$. Finalmente, se concluye dado que $(\{\!\{\alpha_i \setminus^{\alpha'_i} s\}\!\})_{i \leq \mathsf{sz}(\vec{\alpha})} o = [\beta_k]\,(\{\!\{\alpha_i \setminus^{\alpha'_i} s\}\!\})_{i \leq \mathsf{sz}(\vec{\alpha})} t :: s \to_{\Lambda M} [\beta_k]\,\langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle t :: s = \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle o$.
  2. $\delta \notin \vec{\alpha}$. Luego, $(\{\!\{\alpha_i \setminus^{\alpha'_i} s\}\!\})_{i \leq \mathsf{sz}(\vec{\alpha})} o = [\delta]\,(\{\!\{\alpha_i \setminus^{\alpha'_i} s\}\!\})_{i \leq \mathsf{sz}(\vec{\alpha})} t$. Por *h.i.* se tiene entonces $(\{\!\{\alpha_i \setminus^{\alpha'_i} s\}\!\})_{i \leq \mathsf{sz}(\vec{\alpha})} t \twoheadrightarrow_{\Lambda M} \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle t$. Finalmente, se concluye pues $(\{\!\{\alpha_i \setminus^{\alpha'_i} s\}\!\})_{i \leq \mathsf{sz}(\vec{\alpha})} o = [\delta]\,(\{\!\{\alpha_i \setminus^{\alpha'_i} s\}\!\})_{i \leq \mathsf{sz}(\vec{\alpha})} t \twoheadrightarrow_{\Lambda M} [\delta]\,\langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle t = \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle o$.
- $o = c[\gamma \setminus \delta]$ con $\gamma \notin s$ y $\gamma \notin \vec{\alpha'}$. Luego, se tienen dos casos posibles:
  1. $\delta \in \vec{\alpha}$. Luego, $\delta = \alpha_k$ para algún $k \in [1, \mathsf{sz}(\vec{\alpha})]$. Por definición se tiene $(\{\!\{\alpha_i \setminus^{\alpha'_i} s\}\!\})_{i \leq \mathsf{sz}(\vec{\alpha})} o = ((\{\!\{\alpha_i \setminus^{\alpha'_i} s\}\!\})_{i \leq \mathsf{sz}(\vec{\alpha})} c)[\![\gamma \setminus^{\gamma'} (\{\!\{\gamma \setminus^{\gamma'} s\}\!\})_{k < i \leq \mathsf{sz}(\vec{\alpha})} s]\!][\gamma' \setminus \beta_k]$. Más aún, por hipótesis $\vec{\alpha} \cap \mathsf{fn}(s) = \emptyset$, lo que implica $(\{\!\{\alpha_i \setminus^{\alpha'_i} s\}\!\})_{k < i \leq \mathsf{sz}(\vec{\alpha})} s = s$. Por lo tanto, se tiene $(\{\!\{\alpha_i \setminus^{\alpha'_i} s\}\!\})_{i \leq \mathsf{sz}(\vec{\alpha})} o = ((\{\!\{\alpha_i \setminus^{\alpha'_i} s\}\!\})_{i \leq \mathsf{sz}(\vec{\alpha})} c)[\![\gamma \setminus^{\gamma'} s]\!][\gamma' \setminus \beta_k] \to_{\Lambda M} (\{\!\{\gamma \setminus^{\gamma'} s\}\!\}(\{\!\{\alpha_i \setminus^{\alpha'_i} s\}\!\})_{i \leq \mathsf{sz}(\vec{\alpha})} c)[\gamma' \setminus \beta_k]$. Por α-conversión se asume $\gamma \neq \vec{\alpha}$. Luego, por *h.i.* y Lem. C.0.1, $\{\!\{\gamma \setminus^{\gamma'} s\}\!\}(\{\!\{\alpha_i \setminus^{\alpha'_i} s\}\!\})_{i \leq \mathsf{sz}(\vec{\alpha})} c \twoheadrightarrow_{\Lambda M} \langle\!\langle \vec{\alpha}\gamma \setminus^{\vec{\alpha'}\gamma'} s \rangle\!\rangle c = \langle\!\langle \gamma\vec{\alpha} \setminus^{\gamma'\vec{\alpha'}} s \rangle\!\rangle c$. Finalmente, se obtiene la secuencia $(\{\!\{\alpha_i \setminus^{\alpha'_i} s\}\!\})_{i \leq \mathsf{sz}(\vec{\alpha})} o \to_{\Lambda M} (\{\!\{\gamma \setminus^{\gamma'} s\}\!\}(\{\!\{\alpha_i \setminus^{\alpha'_i} s\}\!\})_{i \leq \mathsf{sz}(\vec{\alpha})} c)[\gamma' \setminus \beta_k] \twoheadrightarrow_{\Lambda M} (\langle\!\langle \gamma\vec{\alpha} \setminus^{\gamma'\vec{\alpha'}} s \rangle\!\rangle c)[\gamma' \setminus \beta_k] = \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle o$. Se concluye por transitividad de $\twoheadrightarrow_{\Lambda M}$.
  2. $\delta \notin \vec{\alpha}$. Luego, $(\{\!\{\alpha_i \setminus^{\alpha'_i} s\}\!\})_{i \leq \mathsf{sz}(\vec{\alpha})} o = ((\{\!\{\alpha_i \setminus^{\alpha'_i} s\}\!\})_{i \leq \mathsf{sz}(\vec{\alpha})} c)[\gamma \setminus \delta]$. Por *h.i.* se tiene entonces $(\{\!\{\alpha_i \setminus^{\alpha'_i} s\}\!\})_{i \leq \mathsf{sz}(\vec{\alpha})} c \twoheadrightarrow_{\Lambda M} \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle c$. Finalmente, se concluye pues $(\{\!\{\alpha_i \setminus^{\alpha'_i} s\}\!\})_{i \leq \mathsf{sz}(\vec{\alpha})} o = ((\{\!\{\alpha_i \setminus^{\alpha'_i} s\}\!\})_{i \leq \mathsf{sz}(\vec{\alpha})} c)[\gamma \setminus \delta] \twoheadrightarrow_{\Lambda M} (\langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle c)[\gamma \setminus \delta] = \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle o$.
- $o = t \cdot s'$. Luego, $(\{\!\{\alpha_i \setminus^{\alpha'_i} s\}\!\})_{i \leq \mathsf{sz}(\vec{\alpha})} o = (\{\!\{\alpha_i \setminus^{\alpha'_i} s\}\!\})_{i \leq \mathsf{sz}(\vec{\alpha})} t \cdot (\{\!\{\alpha_i \setminus^{\alpha'_i} s\}\!\})_{i \leq \mathsf{sz}(\vec{\alpha})} s'$. Por *h.i.* se tienen $(\{\!\{\alpha_i \setminus^{\alpha'_i} s\}\!\})_{i \leq \mathsf{sz}(\vec{\alpha})} t \twoheadrightarrow_{\Lambda M} \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle t$ y $(\{\!\{\alpha_i \setminus^{\alpha'_i} s\}\!\})_{i \leq \mathsf{sz}(\vec{\alpha})} s' \twoheadrightarrow_{\Lambda M} \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle s'$. Finalmente,

se tiene $(\{\!\{\alpha_i \setminus^{\alpha'_i} s\}\!\})_{i\leq \mathsf{sz}(\vec{\alpha})} o \twoheadrightarrow_{\Lambda M} \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle t \cdot (\{\!\{\alpha_i \setminus^{\alpha'_i} s\}\!\})_{i\leq \mathsf{sz}(\vec{\alpha})} s' \twoheadrightarrow_{\Lambda M} \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle t \cdot \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle s' = \langle\!\langle \vec{\alpha} \setminus^{\vec{\alpha'}} s \rangle\!\rangle o$. Se concluye por transitividad de $\twoheadrightarrow_{\Lambda M}$.

□

Por último, para aplicar el método de interpretación de Handin es preciso relacionar las relaciones de reducción $\to_{\Lambda M}$ y $\to_{\lambda\mu\rho}$ mediante la función de proyección $\_^{\downarrow}$.

Lema C.0.19.

1. *Sea* $o \in \mathbb{O}_{\Lambda M}$. *Luego,* $o \twoheadrightarrow_{\Lambda M} o^{\downarrow}$.
2. *Sean* $o, o' \in \mathbb{O}_{\lambda\mu\rho}$. *Si* $o \to_{\lambda\mu\rho} o'$, *entonces* $o \twoheadrightarrow_{\Lambda M} o'$.
3. *Sean* $o, o' \in \mathbb{O}_{\Lambda M}$. *Si* $o \to_{\Lambda M} o'$, *entonces* $o^{\downarrow} \twoheadrightarrow_{\lambda\mu\rho} o'^{\downarrow}$.

Demostración. ▪ Por inducción en o.

1. $o = x$. Luego, $o^{\downarrow} = x$. Se concluye por reflexividad de $\twoheadrightarrow_{\Lambda M}$.
2. $o = t\,u$. Luego, $o^{\downarrow} = t^{\downarrow}\,u^{\downarrow}$. Por *h.i.* se tienen $t \twoheadrightarrow_{\Lambda M} t^{\downarrow}$ y $u \twoheadrightarrow_{\Lambda M} u^{\downarrow}$. Finalmente, se concluye por transitividad de $\twoheadrightarrow_{\Lambda M}$, dado que $o \twoheadrightarrow_{\Lambda M} t^{\downarrow}\,u \twoheadrightarrow_{\Lambda M} t^{\downarrow}\,u^{\downarrow} = o^{\downarrow}$.
3. $o = \lambda x.t$. Luego, $o^{\downarrow} = \lambda x.t^{\downarrow}$. Por *h.i.* se tiene $t \twoheadrightarrow_{\Lambda M} t^{\downarrow}$. Finalmente, se concluye dado que $o = \lambda x.t \twoheadrightarrow_{\Lambda M} \lambda x.t^{\downarrow} = o^{\downarrow}$.
4. $o = \mu\alpha.c$. Luego, $o^{\downarrow} = \mu\alpha.c^{\downarrow}$. Por *h.i.* se tiene $c \twoheadrightarrow_{\Lambda M} c^{\downarrow}$. Finalmente, se concluye dado que $o = \mu\alpha.c \twoheadrightarrow_{\Lambda M} \mu\alpha.c^{\downarrow} = o^{\downarrow}$.
5. $o = t[x \setminus u]$. Luego, $o^{\downarrow} = \{x \setminus u^{\downarrow}\}t^{\downarrow}$. Por *h.i.* se tienen $t \twoheadrightarrow_{\Lambda M} t^{\downarrow}$ y $u \twoheadrightarrow_{\Lambda M} u^{\downarrow}$. Finalmente, se tiene $o \twoheadrightarrow_{\Lambda M} t^{\downarrow}[x \setminus u] \twoheadrightarrow_{\Lambda M} t^{\downarrow}[x \setminus u^{\downarrow}] \to_{\Lambda M} \{x \setminus u^{\downarrow}\}t^{\downarrow} = o^{\downarrow}$. Se concluye por transitividad de $\twoheadrightarrow_{\Lambda M}$.
6. $o = [\alpha]\,t$. Luego, $o^{\downarrow} = [\alpha]\,t^{\downarrow}$. Por *h.i.* se tiene $t \twoheadrightarrow_{\Lambda M} t^{\downarrow}$. Finalmente, se concluye dado que $o = [\alpha]\,t \twoheadrightarrow_{\Lambda M} [\alpha]\,t^{\downarrow} = o^{\downarrow}$.
7. $o = c[\![\alpha \setminus^{\alpha'} s]\!]$. Luego, $o^{\downarrow} = \langle\!\langle \alpha \setminus^{\alpha'} s^{\downarrow} \rangle\!\rangle c^{\downarrow}$. Por *h.i.* se tienen $c \twoheadrightarrow_{\Lambda M} c^{\downarrow}$ y $s \twoheadrightarrow_{\Lambda M} s^{\downarrow}$. Entonces, se tiene $o \twoheadrightarrow_{\Lambda M} c^{\downarrow}[\![\alpha \setminus^{\alpha'} s]\!] \twoheadrightarrow_{\Lambda M} c^{\downarrow}[\![\alpha \setminus^{\alpha'} s^{\downarrow}]\!] \to_{\Lambda M} \langle\!\langle \alpha \setminus^{\alpha'} s^{\downarrow} \rangle\!\rangle c^{\downarrow}$. Más aún, por Lem. C.0.18, $\{\!\{\alpha \setminus^{\alpha'} s^{\downarrow}\}\!\} c^{\downarrow} \twoheadrightarrow_{\Lambda M} \langle\!\langle \alpha \setminus^{\alpha'} s^{\downarrow} \rangle\!\rangle c^{\downarrow} = o^{\downarrow}$. Se concluye por transitividad de $\twoheadrightarrow_{\Lambda M}$.
8. $o = c[\alpha \setminus \beta]$. Luego, $o^{\downarrow} = c^{\downarrow}[\alpha \setminus \beta]$. Por *h.i.* se tiene $c \twoheadrightarrow_{\Lambda M} c^{\downarrow}$. Finalmente, se concluye dado que $o = c[\alpha \setminus \beta] \twoheadrightarrow_{\Lambda M} c^{\downarrow}[\alpha \setminus \beta] = o^{\downarrow}$.
9. $o = t \cdot s$. Luego, $o^{\downarrow} = t^{\downarrow} \cdot s^{\downarrow}$. Por *h.i.* se tienen $t \twoheadrightarrow_{\Lambda M} t^{\downarrow}$ y $s \twoheadrightarrow_{\Lambda M} s^{\downarrow}$. Finalmente, se concluye por transitividad de $\twoheadrightarrow_{\Lambda M}$, dado que $o \twoheadrightarrow_{\Lambda M} t^{\downarrow} \cdot s \twoheadrightarrow_{\Lambda M} t^{\downarrow} \cdot s^{\downarrow} = o^{\downarrow}$.

▪ Por definición $o \to_{\lambda\mu\rho} o'$ implica $o = \mathtt{O}\langle l \rangle$ y $o' = \mathtt{O}\langle r \rangle$ con $l \mapsto_* r$, $* \in \{\beta, \mu^s\}$. Se procede por inducción en $\mathtt{O}$.

- $\mathtt{O} = \Box$. Luego, se tienen dos casos posibles según la regla de reescritura aplicada al reducir.
  1. β. Luego, $o = (\lambda x.t)\,u$ y $o' = \{x \setminus u\}t$. Entonces, $o \to_{\mathtt{dB}} t[x \setminus u] \to_{\mathtt{S}} o'$ por lo que se concluye.
  2. μ^s. Luego, $o = (\mu\alpha.c) :: s$ y $o' = \mu\alpha'.\langle\!\langle \alpha \setminus^{\alpha'} s \rangle\!\rangle c$ con α' un nombre fresco. Se procede por inducción en s.

- $s = u$. Luego, $o = (\mu\alpha.c)\, u$ y $o' = \mu\alpha'.\langle\!\langle \alpha \setminus^{\alpha'} u \rangle\!\rangle c$. Entonces, $o \to_{\mathtt{dM}} \mu\alpha'.c[\![\alpha \setminus^{\alpha'} u]\!] \to_{\mathtt{R}} \mu\alpha'.\{\!\{\alpha \setminus^{\alpha'} u\}\!\}c$. Más aún, por Lem. C.0.18, $\mu\alpha'.\{\!\{\alpha \setminus^{\alpha'} u\}\!\}c \twoheadrightarrow_{\Lambda M} o'$. Se concluye por transitividad de $\twoheadrightarrow_{\Lambda M}$.
- $s = u \cdot s'$. Luego, $o = ((\mu\alpha.c)\, u) :: s'$ y $o' = \mu\alpha'.[\alpha']\alpha u \cdot s' c$. Entonces, $o \to_{\mathtt{dM}} (\mu\beta.c[\![\alpha \setminus^{\beta} u]\!]) :: s' \to_{\mathtt{R}} (\mu\beta.\{\!\{\alpha \setminus^{\beta} u\}\!\}c) :: s'$ con β un nombre fresco. Por *h.i.* se tiene $(\mu\beta.\{\!\{\alpha \setminus^{\beta} u\}\!\}c) :: s' \twoheadrightarrow_{\Lambda M} \mu\alpha'.\langle\!\langle \beta \setminus^{\alpha'} s' \rangle\!\rangle \{\!\{\alpha \setminus^{\beta} u\}\!\}c$. Más aún, por Lem. C.0.18, $\mu\alpha'.\langle\!\langle \beta \setminus^{\alpha'} s' \rangle\!\rangle \{\!\{\alpha \setminus^{\beta} u\}\!\}c \twoheadrightarrow_{\Lambda M} \mu\alpha'.\langle\!\langle \beta \setminus^{\alpha'} s' \rangle\!\rangle \langle\!\langle \alpha \setminus^{\beta} u \rangle\!\rangle c$. Luego, por Lem. C.0.5, se tiene $\mu\alpha'.\langle\!\langle \beta \setminus^{\alpha'} s' \rangle\!\rangle \langle\!\langle \alpha \setminus^{\beta} u \rangle\!\rangle c = \mu\alpha'.\langle\!\langle \alpha \setminus^{\beta} \langle\!\langle \beta \setminus^{\alpha'} s' \rangle\!\rangle u \cdot s' \rangle\!\rangle c$ y, por Cor. C.0.3 con $\beta \notin \mathtt{fn}(u)$, $\mu\alpha'.\langle\!\langle \alpha \setminus^{\beta} \langle\!\langle \beta \setminus^{\alpha'} s' \rangle\!\rangle u \cdot s' \rangle\!\rangle c = \mu\alpha'.\langle\!\langle \alpha \setminus^{\beta} u \cdot s' \rangle\!\rangle c$. Finalmente, por transitividad de $\twoheadrightarrow_{\Lambda M}$, se tiene $o \twoheadrightarrow_{\Lambda M} \mu\alpha'.\langle\!\langle \alpha \setminus^{\beta} u \cdot s' \rangle\!\rangle c = o'$, por lo que se concluye.

- $\mathtt{O} = \boxdot$. Este caso no aplica dado que no hay reglas de reescritura sobre comandos.
- $\mathtt{O} = \mathtt{T}\, u$. Luego, $o = t\, u$ y $o' = t'\, u$ con $t \to_{\lambda\mu\rho} t'$. Por *h.i.* se tiene $t \to_{\Lambda M} t'$. Finalmente, $o = t\, u \to_{\Lambda M} t'\, u = o'$, por lo que se concluye.
- $\mathtt{O} = t\, \mathtt{T}$. Luego, $o = t\, u$ y $o' = t\, u'$ con $u \to_{\lambda\mu\rho} u'$. Por *h.i.* se tiene $u \to_{\Lambda M} u'$. Finalmente, $o = t\, u \to_{\Lambda M} t\, u' = o'$, por lo que se concluye.
- $\mathtt{O} = \lambda x.\mathtt{T}$. Luego, $o = \lambda x.t$ y $o' = \lambda x.t'$ con $t \to_{\lambda\mu\rho} t'$. Por *h.i.* se tiene $t \to_{\Lambda M} t'$. Finalmente, $o = \lambda x.t \to_{\Lambda M} \lambda x.t' = o'$, por lo que se concluye.
- $\mathtt{O} = \mu\alpha.\mathtt{C}$. Luego, $o = \mu\alpha.c$ y $o' = \mu\alpha.c'$ con $c \to_{\lambda\mu\rho} c'$. Por *h.i.* se tiene $c \to_{\Lambda M} c'$. Finalmente, $o = \mu\alpha.c \to_{\Lambda M} \mu\alpha.c' = o'$, por lo que se concluye.
- $\mathtt{O} = [\alpha]\, \mathtt{T}$. Luego, $o = [\alpha]\, t$ y $o' = [\alpha]\, t'$ con $t \to_{\lambda\mu\rho} t'$. Por *h.i.* se tiene $t \to_{\Lambda M} t'$. Finalmente, $o = [\alpha]\, t \to_{\Lambda M} [\alpha]\, t' = o'$, por lo que se concluye.
- $\mathtt{O} = \mathtt{C}[\alpha \setminus \beta]$. Luego, $o = c[\alpha \setminus \beta]$ y $o' = c'[\alpha \setminus \beta]$ con $c \to_{\lambda\mu\rho} c'$. Por *h.i.* se tiene $c \to_{\Lambda M} c'$. Finalmente, $o = c[\alpha \setminus \beta] \to_{\Lambda M} c'[\alpha \setminus \beta] = o'$, por lo que se concluye.
- $\mathtt{O} = \mathtt{T} \cdot s$. Luego, $o = t \cdot s$ y $o' = t' \cdot s$ con $t \to_{\lambda\mu\rho} t'$. Por *h.i.* se tiene $t \to_{\Lambda M} t'$. Finalmente, $o = t \cdot s \to_{\Lambda M} t' \cdot s = o'$, por lo que se concluye.
- $\mathtt{O} = t \cdot \mathtt{S}$. Luego, $o = t \cdot s$ y $o' = t \cdot s'$ con $s \to_{\lambda\mu\rho} s'$. Por *h.i.* se tiene $s \to_{\Lambda M} s'$. Finalmente, $o = t \cdot s \to_{\Lambda M} t \cdot s' = o'$, por lo que se concluye.

- Por definición $o \to_{\Lambda M} o'$ implica $o = \mathtt{O}\langle l \rangle$ y $o' = \mathtt{O}\langle r \rangle$ con $l \mapsto_* r$, $* \in \{\mathtt{dB}, \mathtt{S}, \mathtt{dM}, \mathtt{R}\}$. Se procede por inducción en $\mathtt{O}$.

  - $\mathtt{O} = \Box$. Luego, se tienen tres casos posibles según la regla de reescritura aplicada al reducir.
    1. $\mathtt{dB}$. Luego, $o = \mathtt{L}\langle \lambda x.t \rangle\, u$ y $o' = \mathtt{L}\langle t[x \setminus u] \rangle$ con $\mathtt{fc}(u, \mathtt{L})$. Se procede por inducción en $\mathtt{L}$.
       - $\mathtt{L} = \Box$. Luego, $o^{\downarrow} = (\lambda x.t^{\downarrow})\, u^{\downarrow} \to_{\beta} \{x \setminus u^{\downarrow}\} t^{\downarrow} = o'^{\downarrow}$, por lo que se concluye.
       - $\mathtt{L} = \mathtt{L}'[y \setminus v]$. Luego, $o^{\downarrow} = \{y \setminus v^{\downarrow}\} \mathtt{L}'\langle \lambda x.t \rangle^{\downarrow}\, u^{\downarrow} = \{y \setminus v^{\downarrow}\} (\mathtt{L}'\langle \lambda x.t \rangle\, u)^{\downarrow}$, dado que $y \notin u$ por hipótesis. Por *h.i.* se tiene entonces $(\mathtt{L}'\langle \lambda x.t \rangle\, u)^{\downarrow} \to_{\lambda\mu\rho} (\mathtt{L}'\langle t[x \setminus u] \rangle)^{\downarrow}$. Más aún, por Lem. C.0.15 (1), $o^{\downarrow} = \{y \setminus v^{\downarrow}\} (\mathtt{L}'\langle \lambda x.t \rangle\, u)^{\downarrow} \to_{\lambda\mu\rho} \{y \setminus v^{\downarrow}\} (\mathtt{L}'\langle t[x \setminus u] \rangle)^{\downarrow} = o'^{\downarrow}$, por lo que se concluye.
    2. $\mathtt{S}$. Luego, $o = t[x \setminus u]$ y $o' = \{x \setminus u\} t$. Se concluye por Lem. C.0.14, dado que $o^{\downarrow} = \{x \setminus u^{\downarrow}\} t^{\downarrow} = (\{x \setminus u\} t)^{\downarrow} = o'^{\downarrow}$.
    3. $\mathtt{dM}$. Luego, $o = \mathtt{L}\langle \mu\alpha.c \rangle\, u$ y $o' = \mathtt{L}\langle \mu\alpha'.c[\![\alpha \setminus^{\alpha'} u]\!] \rangle$ con $\mathtt{fc}(u, \mathtt{L})$ y α' un nombre fresco. Se procede por inducción en $\mathtt{L}$.
       - $\mathtt{L} = \Box$. Luego, $o^{\downarrow} = (\mu\alpha.c^{\downarrow})\, u^{\downarrow} \to_{\mu^s} \mu\alpha'.\langle\!\langle \alpha \setminus^{\alpha'} u^{\downarrow} \rangle\!\rangle c^{\downarrow} = \mu\alpha'.(c[\![\alpha \setminus^{\alpha'} u]\!])^{\downarrow} = o'^{\downarrow}$, por lo que se concluye.

  - $\mathtt{L} = \mathtt{L}'[y \setminus v]$. Luego, $o^{\downarrow} = \{y \setminus v^{\downarrow}\}\mathtt{L}'\langle\mu\alpha.t\rangle^{\downarrow}\, u^{\downarrow} = \{y \setminus v^{\downarrow}\}(\mathtt{L}'\langle\mu\alpha.t\rangle\, u)^{\downarrow}$, dado que $y \notin u$ por hipótesis. Por *h.i.* se tiene $(\mathtt{L}'\langle\mu\alpha.t\rangle\, u)^{\downarrow} \to_{\lambda\mu\rho} (\mathtt{L}'\langle\mu\alpha'.c[\![\alpha \setminus^{\alpha'} u]\!]\rangle)^{\downarrow}$. Más aún, por Lem. C.0.17 (1), $o^{\downarrow} = \{y \setminus v^{\downarrow}\}(\mathtt{L}'\langle\mu\alpha.t\rangle\, u)^{\downarrow} \to_{\lambda\mu\rho} \{y \setminus v^{\downarrow}\}(\mathtt{L}'\langle\mu\alpha'.c[\![\alpha \setminus^{\alpha'} u]\!]\rangle)^{\downarrow} = o'^{\downarrow}$, por lo que se concluye.

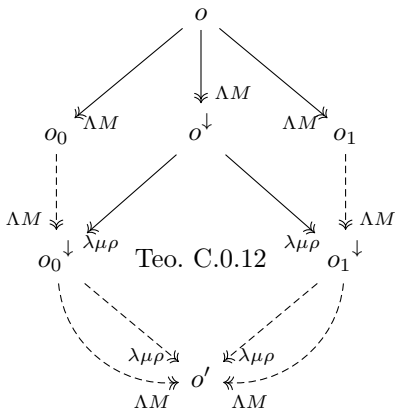
- $\mathtt{O} = \boxdot$. Luego, la regla de reescritura aplicada es necesariamente $\mathtt{R}$, *i.e.* $o = c[\![\alpha \setminus^{\alpha'} s]\!]$ y $o' = \{\!\{\alpha \setminus^{\alpha'} s\}\!\}c$. Se concluye por Lem. C.0.16, dado que $o^{\downarrow} = \langle\!\langle\alpha \setminus^{\alpha'} s^{\downarrow}\rangle\!\rangle c^{\downarrow} = (\{\!\{\alpha \setminus^{\alpha'} s\}\!\}c)^{\downarrow} = o'^{\downarrow}$.
- $\mathtt{O} = \mathtt{T}\, u$. Luego, $o = t\, u$ y $o' = t'\, u$ con $t \to_{\Lambda M} t'$. Por *h.i.* se tiene $t^{\downarrow} \twoheadrightarrow_{\lambda\mu\rho} t'^{\downarrow}$. Finalmente, $o^{\downarrow} = t^{\downarrow}\, u^{\downarrow} \twoheadrightarrow_{\lambda\mu\rho} t'^{\downarrow}\, u^{\downarrow} = o'^{\downarrow}$, por lo que se concluye.
- $\mathtt{O} = t\, \mathtt{T}$. Luego, $o = t\, u$ y $o' = t\, u'$ con $u \to_{\Lambda M} u'$. Por *h.i.* se tiene $u^{\downarrow} \twoheadrightarrow_{\lambda\mu\rho} u'$. Finalmente, $o^{\downarrow} = t^{\downarrow}\, u^{\downarrow} \twoheadrightarrow_{\lambda\mu\rho} t^{\downarrow}\, u'^{\downarrow} = o'^{\downarrow}$, por lo que se concluye.
- $\mathtt{O} = \lambda x.\mathtt{T}$. Luego, $o = \lambda x.t$ y $o' = \lambda x.t'$ con $t \to_{\Lambda M} t'$. Por *h.i.* se tiene $t^{\downarrow} \twoheadrightarrow_{\lambda\mu\rho} t'^{\downarrow}$. Finalmente, $o^{\downarrow} = \lambda x.t^{\downarrow} \twoheadrightarrow_{\lambda\mu\rho} \lambda x.t'^{\downarrow} = o'^{\downarrow}$, por lo que se concluye.
- $\mathtt{O} = \mu\alpha.\mathtt{C}$. Luego, $o = \mu\alpha.c$ y $o' = \mu\alpha.c'$ con $c \to_{\Lambda M} c'$. Por *h.i.* se tiene $c^{\downarrow} \twoheadrightarrow_{\lambda\mu\rho} c'^{\downarrow}$. Finalmente, $o^{\downarrow} = \mu\alpha.c^{\downarrow} \twoheadrightarrow_{\lambda\mu\rho} \mu\alpha.c'^{\downarrow} = o'^{\downarrow}$, por lo que se concluye.
- $\mathtt{O} = \mathtt{T}[x \setminus u]$. Luego, $o = t[x \setminus u]$ y $o' = t'[x \setminus u]$ con $t \to_{\Lambda M} t'$. Por *h.i.* se tiene $t^{\downarrow} \twoheadrightarrow_{\lambda\mu\rho} t'^{\downarrow}$. Se concluye por Lem. C.0.15 (1), dado que $o^{\downarrow} = \{x \setminus u^{\downarrow}\}t^{\downarrow} \twoheadrightarrow_{\lambda\mu\rho} \{x \setminus u^{\downarrow}\}t'^{\downarrow} = o'^{\downarrow}$.
- $\mathtt{O} = t[x \setminus \mathtt{T}]$. Luego, $o = t[x \setminus u]$ y $o' = t[x \setminus u']$ con $u \to_{\Lambda M} u'$. Por *h.i.* se tiene $u^{\downarrow} \twoheadrightarrow_{\lambda\mu\rho} u'^{\downarrow}$. Se concluye por Lem. C.0.15 (2), dado que $o^{\downarrow} = \{x \setminus u^{\downarrow}\}t^{\downarrow} \twoheadrightarrow_{\lambda\mu\rho} \{x \setminus u'^{\downarrow}\}t^{\downarrow} = o'^{\downarrow}$.
- $\mathtt{O} = [\alpha]\, \mathtt{T}$. Luego, $o = [\alpha]\, t$ y $o' = [\alpha]\, t'$ con $t \to_{\Lambda M} t'$. Por *h.i.* se tiene $t^{\downarrow} \twoheadrightarrow_{\lambda\mu\rho} t'^{\downarrow}$. Finalmente, $o^{\downarrow} = [\alpha]\, t^{\downarrow} \twoheadrightarrow_{\lambda\mu\rho} [\alpha]\, t'^{\downarrow} = o'^{\downarrow}$, por lo que se concluye.
- $\mathtt{O} = \mathtt{C}[\![\alpha \setminus^{\alpha'} s]\!]$. Luego, $o = c[\![\alpha \setminus^{\alpha'} s]\!]$ y $o' = c'[\![\alpha \setminus^{\alpha'} s]\!]$ con $c \to_{\Lambda M} c'$. Por *h.i.* se tiene $c^{\downarrow} \twoheadrightarrow_{\lambda\mu\rho} c'^{\downarrow}$. Se concluye por Lem. C.0.17 (1), dado que $o^{\downarrow} = \langle\!\langle\alpha \setminus^{\alpha'} s^{\downarrow}\rangle\!\rangle c^{\downarrow} \twoheadrightarrow_{\lambda\mu\rho} \langle\!\langle\alpha \setminus^{\alpha'} s^{\downarrow}\rangle\!\rangle c'^{\downarrow} = o'^{\downarrow}$.
- $\mathtt{O} = c[\![\alpha \setminus^{\alpha'} \mathtt{S}]\!]$. Luego, $o = c[\![\alpha \setminus^{\alpha'} s]\!]$ y $o' = c[\![\alpha \setminus^{\alpha'} s']\!]$ con $s \to_{\Lambda M} s'$. Por *h.i.* se tiene $s^{\downarrow} \twoheadrightarrow_{\lambda\mu\rho} s'^{\downarrow}$. Se concluye por Lem. C.0.17 (2), dado que $o^{\downarrow} = \langle\!\langle\alpha \setminus^{\alpha'} s^{\downarrow}\rangle\!\rangle c^{\downarrow} \twoheadrightarrow_{\lambda\mu\rho} \langle\!\langle\alpha \setminus^{\alpha'} s'^{\downarrow}\rangle\!\rangle c^{\downarrow} = o'^{\downarrow}$.
- $\mathtt{O} = \mathtt{C}[\alpha \setminus \beta]$. Luego, $o = c[\alpha \setminus \beta]$ y $o' = c'[\alpha \setminus \beta]$ con $c \to_{\Lambda M} c'$. Por *h.i.* se tiene $c^{\downarrow} \twoheadrightarrow_{\lambda\mu\rho} c'^{\downarrow}$. Finalmente, $o^{\downarrow} = c^{\downarrow}[\alpha \setminus \beta] \twoheadrightarrow_{\lambda\mu\rho} c'^{\downarrow}[\alpha \setminus \beta] = o'^{\downarrow}$, por lo que se concluye.
- $\mathtt{O} = \mathtt{T} \cdot s$. Luego, $o = t \cdot s$ y $o' = t' \cdot s$ con $t \to_{\Lambda M} t'$. Por *h.i.* se tiene $t^{\downarrow} \twoheadrightarrow_{\lambda\mu\rho} t'^{\downarrow}$. Finalmente, $o^{\downarrow} = t^{\downarrow} \cdot s^{\downarrow} \twoheadrightarrow_{\lambda\mu\rho} t'^{\downarrow} \cdot s^{\downarrow} = o'^{\downarrow}$, por lo que se concluye.
- $\mathtt{O} = t \cdot \mathtt{S}$. Luego, $o = t \cdot s$ y $o' = t \cdot s'$ con $s \to_{\Lambda M} s'$. Por *h.i.* se tiene $s^{\downarrow} \twoheadrightarrow_{\lambda\mu\rho} s'^{\downarrow}$. Finalmente, $o^{\downarrow} = t^{\downarrow} \cdot s^{\downarrow} \twoheadrightarrow_{\lambda\mu\rho} t^{\downarrow} \cdot s'^{\downarrow} = o'^{\downarrow}$, por lo que se concluye.

□

La confluencia de $\to_{\Lambda M}$ es consecuencia del Lem. C.0.19 y la confluencia de $\to_{\lambda\mu\rho}$ (Teo. C.0.12).

Teorema 4.2.2. *La relación de reducción $\to_{\Lambda M}$ es confluente (CR).*

Demostración. Por método de intepretación, apelando a la confluencia de $\to_{\lambda\mu\rho}$ y el Lem. C.0.19:



donde $o \twoheadrightarrow_{\Lambda M} o_0$ y $o \twoheadrightarrow_{\Lambda M} o_1$ son las hipótesis del teorema. Las tres reducciones verticales se justifican por Lem. C.0.19 (1), dado que $p \twoheadrightarrow_{\Lambda M} p^\downarrow$ para todo $o \in \mathbb{O}_{\Lambda M}$. Las reducciones $o^\downarrow \twoheadrightarrow_{\lambda\mu\rho} {o_0}^\downarrow$ y $o^\downarrow \twoheadrightarrow_{\lambda\mu\rho} {o_1}^\downarrow$ se siguen de las hipótesis por Lem. C.0.19 (3). El diagrama se cierra por Teo. C.0.12 obteniendo luego ${o_0}^\downarrow \twoheadrightarrow_{\Lambda M} o'$ y ${o_1}^\downarrow \twoheadrightarrow_{\Lambda M} o'$ por Lem. C.0.19 (2). □

Sistema de tipos simples para ΛM-cálculo

Lema C.0.20 (Stacks)**.**

1. *Sean* $\pi_s \rhd_{\mathcal{E}} \Gamma \vdash s : S \mid \Delta$ *y* $\pi_{s'} \rhd_{\mathcal{E}} \Gamma' \vdash s' : S' \mid \Delta'$*. Luego, existe* $\pi_{s\cdot s'} \rhd_{\mathcal{E}} \Gamma \cup \Gamma' \vdash s \cdot s' : S \cdot S' \mid \Delta \cup \Delta'$*.*
2. *Sea* $\pi_{s\cdot s'} \rhd_{\mathcal{E}} \Gamma^* \vdash s \cdot s' : S^* \mid \Delta^*$*. Luego, existen* $\pi_s \rhd_{\mathcal{E}} \Gamma \vdash s : S \mid \Delta$ *y* $\pi_{s'} \rhd_{\mathcal{E}} \Gamma' \vdash s' : S' \mid \Delta'$ *tal que* $S^* = S \cdot S'$*,* $\Gamma^* = \Gamma \cup \Gamma'$ *y* $\Delta^* = \Delta \cup \Delta'$*.*
3. *Sean* $\pi_t \rhd_{\mathcal{E}} \Gamma \vdash t : S \to A \mid \Delta$ *y* $\pi_s \rhd_{\mathcal{E}} \Gamma' \vdash s : S \mid \Delta'$*. Luego, existe* $\pi_{t::s} \rhd_{\mathcal{E}} \Gamma \cup \Gamma' \vdash t :: s : A \mid \Delta \cup \Delta'$*.*
4. *Sea* $\pi_{t::s} \rhd_{\mathcal{E}} \Gamma^* \vdash t :: s : A \mid \Delta^*$*. Luego, existen* $\pi_t \rhd_{\mathcal{E}} \Gamma \vdash t : S \to A \mid \Delta$ *y* $\pi_s \rhd_{\mathcal{E}} \Gamma' \vdash s : S \mid \Delta'$ *tal que* $\Gamma^* = \Gamma \cup \Gamma'$ *y* $\Delta^* = \Delta \cup \Delta'$*.*

Demostración.

1. Por inducción en s.
   - $s = t$. Luego, $S = A$ un tipo simple y se concluye por (STK) con $\pi_{s'}$, obteniendo la derivación $\pi_{s\cdot s'} \rhd_{\mathcal{E}} \Gamma \cup \Gamma' \vdash t \cdot s' : A \cdot S' \mid \Delta \cup \Delta'$.
   - $s = t \cdot s''$. Luego, por (STK) se tiene $S = A \cdot S''$, $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta = \Delta_0 \cup \Delta_1$ con $\pi_t \rhd_{\mathcal{E}} \Gamma_0 \vdash t : A \mid \Delta_0$ y $\pi_{s''} \rhd_{\mathcal{E}} \Gamma_1 \vdash s'' : S'' \mid \Delta_1$. Por *h.i.* existe una derivación $\pi_{s''\cdot s'} \rhd_{\mathcal{E}} \Gamma \cup \Gamma_1 \vdash s'' \cdot s' : S'' \cdot S' \mid \Delta \cup \Delta_1$. Finalmente, se concluye por (STK), obteniendo la derivación $\pi_{s\cdot s'} \rhd_{\mathcal{E}} \Gamma \cup \Gamma' \vdash t \cdot s'' \cdot s' : A \cdot S'' \cdot S' \mid \Delta \cup \Delta'$.
2. Por inducción en s.
   - $s = t$. Por (STK) se tienen $S^* = A \cdot S'$, $\Gamma^* = \Gamma \cup \Gamma'$ y $\Delta^* = \Delta \cup \Delta'$ con $\pi_t \rhd_{\mathcal{E}} \Gamma \vdash t : A \mid \Delta$ y $\pi_{s'} \rhd_{\mathcal{E}} \Gamma' \vdash s' : S' \mid \Delta'$.
   - $s = t \cdot s''$. Luego, por (STK) se tienen $S^* = A \cdot S^{**}$, $\Gamma^* = \Gamma_0 \cup \Gamma^{**}$ y $\Delta^* = \Delta_0 \cup \Delta^{**}$ con $\pi_t \rhd_{\mathcal{E}} \Gamma_0 \vdash t : A \mid \Delta_0$ y $\pi_{s''\cdot s'} \rhd_{\mathcal{E}} \Gamma^{**} \vdash s'' \cdot s' : S^{**} \mid \Delta^{**}$. Por *h.i.* existen $\pi_{s''} \rhd_{\mathcal{E}} \Gamma_1 \vdash s'' : S'' \mid \Delta_1$ y $\pi_{s'} \rhd_{\mathcal{E}} \Gamma' \vdash s' : S' \mid \Delta'$ tal que $S^{**} = S'' \cdot S'$, $\Gamma^{**} = \Gamma_1 \cup \Gamma'$ y $\Delta^{**} = \Delta_1 \cup \Delta'$. Finalmente, se concluye por (STK) entre π_t y $\pi_{s''}$, $\pi_s \rhd_{\mathcal{E}} \Gamma \vdash s : S \mid \Delta$ donde $S = A \cdot S''$, $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta = \Delta_0 \cup \Delta_1$.

3. Por inducción en s.

   - $s = u$. Luego, $S = B$ un tipo simple y se concluye por (APP) con π_t puesto que $t :: u = t\,u$, $\pi_{t\,u} \rhd_{\mathcal{E}} \Gamma \cup \Gamma' \vdash t\,u : A \mid \Delta \cup \Delta'$.
   - $s = s' \cdot u$. Por Lem. C.0.20 (2), $S = S' \cdot B$, $\Gamma' = \Gamma'_0 \cup \Gamma'_1$ y $\Delta' = \Delta'_0 \cup \Delta'_1$ con $\pi_{s'} \rhd_{\mathcal{E}} \Gamma'_0 \vdash s' : S' \mid \Delta'_0$ y $\pi_u \rhd_{\mathcal{E}} \Gamma'_1 \vdash u : B \mid \Delta'_1$. Notar que $S \to A = S' \cdot B \to A = S' \to B \to A$. Luego, por *h.i.* con π_t y $\pi_{s'}$ se tiene $\pi_{t::s'} \rhd_{\mathcal{E}} \Gamma \cup \Gamma'_0 \vdash t :: s' : B \to A \mid \Delta \cup \Delta'_0$. Finalmente, se concluye por (APP) con π_u, $\pi_{t::s} \rhd_{\mathcal{E}} \Gamma \cup \Gamma' \vdash t :: (s' \cdot u) : A \mid \Delta \cup \Delta'$.

4. Por inducción en s.

   - $s = u$. Luego, $t :: u = t\,u$ y por (APP) se tienen $\Gamma^* = \Gamma \cup \Gamma'$ y $\Delta^* = \Delta \cup \Delta'$ con $\pi_t \rhd_{\mathcal{E}} \Gamma \vdash t : B \to A \mid \Delta$ y $\pi_s \rhd_{\mathcal{E}} \Gamma' \vdash u : B \mid \Delta'$. Se concluye con $S = B$.
   - $s = s' \cdot u$. Luego, $t :: s = (t :: s')\,u$ y por (APP) se tienen $\Gamma^* = \Gamma^{**} \cup \Gamma'_1$ y $\Delta^* = \Delta^{**} \cup \Delta'_1$ con $\pi_{t::s'} \rhd_{\mathcal{E}} \Gamma^{**} \vdash t :: s' : B \to A \mid \Delta^{**}$ y $\pi_u \rhd_{\mathcal{E}} \Gamma'_1 \vdash u : B \mid \Delta'_1$. Por *h.i.* existen $\pi_t \rhd_{\mathcal{E}} \Gamma \vdash t : S' \to B \to A \mid \Delta$ y $\pi_{s'} \rhd_{\mathcal{E}} \Gamma'_0 \vdash s : S' \mid \Delta'_0$ tal que $\Gamma^{**} = \Gamma \cup \Gamma'_0$ y $\Delta^{**} = \Delta \cup \Delta'_0$. Se concluye pues, por Lem. C.0.20 (1) con $\pi_{s'}$ y π_u, existe $\pi_s \rhd_{\mathcal{E}} \Gamma' \vdash s' \cdot u : S' \cdot B \mid \Delta'$ donde $\Gamma' = \Gamma'_0 \cup \Gamma'_1$ y $\Delta' = \Delta'_0 \cup \Delta'_1$. Notar que $S' \cdot B \to A = S' \to B \to A$. Luego, se toma $S = S' \cdot B$.

□

Lema 4.3.1 (Relevance)**.** *Sea* $o \in \mathbb{O}_{\Lambda M}$*. Si* $\pi \rhd_{\mathcal{E}} \Gamma \vdash o : T \mid \Delta$*, entonces* $\mathsf{dom}(\Gamma) = \mathsf{fv}(o)$ *y* $\mathsf{dom}(\Delta) = \mathsf{fn}(o)$.

Demostración. Por inducción en π analizando la última regla aplicada.

- (VAR). Luego, $o = x$, $\Gamma = \{x : T\}$ y $\Delta = \emptyset$. El resultado es inmediato pues $\mathsf{fv}(o) = \{x\}$ y $\mathsf{fn}(o) = \emptyset$.
- (APP). Luego, $o = t\,u$, $\Gamma = \Gamma' \cup \Gamma''$ y $\Delta = \Delta' \cup \Delta''$ con $\pi_t \rhd_{\mathcal{E}} \Gamma' \vdash t : A \to T \mid \Delta'$ y $\pi_u \rhd_{\mathcal{E}} \Gamma'' \vdash u : A \mid \Delta''$. Por *h.i.* se tienen $\mathsf{dom}(\Gamma') \subseteq \mathsf{fv}(t)$, $\mathsf{dom}(\Delta') \subseteq \mathsf{fn}(t)$, $\mathsf{dom}(\Gamma'') \subseteq \mathsf{fv}(u)$ y $\mathsf{dom}(\Delta') \subseteq \mathsf{fn}(u)$. Se concluye pues $\mathsf{dom}(\Gamma) = \mathsf{dom}(\Gamma') \cup \mathsf{dom}(\Gamma'') \subseteq \mathsf{fv}(t) \cup \mathsf{fv}(u) = \mathsf{fv}(o)$ y $\mathsf{dom}(\Delta) = \mathsf{dom}(\Delta') \cup \mathsf{dom}(\Delta'') \subseteq \mathsf{fn}(t) \cup \mathsf{fn}(u) = \mathsf{fn}(o)$.
- (ABS). Luego, $o = \lambda x.t$ y $T = A \to B$ con $\pi_t \rhd_{\mathcal{E}} \Gamma; (x : A)^{\leq 1} \vdash t : B \mid \Delta$. Por *h.i.* se tienen $\mathsf{dom}(\Gamma; (x : A)^{\leq 1}) \subseteq \mathsf{fv}(t)$ y $\mathsf{dom}(\Delta) \subseteq \mathsf{fn}(t)$. Se concluye pues $\mathsf{dom}(\Gamma) = \mathsf{dom}(\Gamma; (x : A)^{\leq 1}) \setminus \{x\} \subseteq \mathsf{fv}(t) \setminus \{x\} = \mathsf{fv}(o)$ y $\mathsf{dom}(\Delta) \subseteq \mathsf{fn}(t) = \mathsf{fn}(o)$.
- (CONT). Luego, $o = \mu\alpha.c$ con $\pi_c \rhd_{\mathcal{E}} \Gamma \vdash c \mid \Delta; (\alpha : A)^{\leq 1}$. Por *h.i.* se tienen $\mathsf{dom}(\Gamma) \subseteq \mathsf{fv}(c)$ y $\mathsf{dom}(\Delta) \subseteq \mathsf{fn}(c)$. Se concluye pues $\mathsf{dom}(\Gamma) \subseteq \mathsf{fv}(c) = \mathsf{fv}(o)$ y $\mathsf{dom}(\Delta) = \mathsf{dom}(\Delta; (\alpha : A)^{\leq 1}) \setminus \{\alpha\} \subseteq \mathsf{fn}(c) \setminus \{\alpha\} = \mathsf{fn}(o)$.
- (SUBS). Luego, $o = t[x \setminus u]$, $\Gamma = \Gamma' \cup \Gamma''$ y $\Delta = \Delta' \cup \Delta''$ con $\pi_t \rhd_{\mathcal{E}} \Gamma'; (x : B)^{\leq 1} \vdash t : T \mid \Delta'$ y $\pi_u \rhd_{\mathcal{E}} \Gamma'' \vdash u : B \mid \Delta''$. Por *h.i.* se tienen $\mathsf{dom}(\Gamma') \subseteq \mathsf{fv}(t)$, $\mathsf{dom}(\Delta') \subseteq \mathsf{fv}(t)$, $\mathsf{dom}(\Gamma'') \subseteq \mathsf{fv}(u)$ y $\mathsf{dom}(\Delta'') \subseteq \mathsf{fn}(u)$. Luego, se concluye pues $\mathsf{dom}(\Gamma) = (\mathsf{dom}(\Gamma') \setminus \{x\}) \cup \mathsf{dom}(\Gamma'') \subseteq (\mathsf{fv}(t) \setminus \{x\}) \cup \mathsf{fv}(u) = \mathsf{fv}(o)$ y $\mathsf{dom}(\Delta) = \mathsf{dom}(\Delta') \cup \mathsf{dom}(\Delta'') \subseteq \mathsf{fn}(t) \cup \mathsf{fn}(u) = \mathsf{fn}(o)$.
- (NAME). Luego, $o = [\alpha]\,t$ y $\Delta = \Delta'; \alpha : A$ con $\pi_t \rhd_{\mathcal{E}} \Gamma \vdash t : A \mid \Delta'; (\alpha : A)^{\leq 1}$. Por *h.i.* se tienen $\mathsf{dom}(\Gamma) \subseteq \mathsf{fv}(t)$ y $\mathsf{dom}(\Delta'; (\alpha : A)^{\leq 1}) \subseteq \mathsf{fn}(t)$. Luego, se concluye pues $\mathsf{dom}(\Gamma) \subseteq \mathsf{fv}(t) = \mathsf{fv}(o)$ y $\mathsf{dom}(\Delta) = \mathsf{dom}(\Delta') \cup \{\alpha\} = \mathsf{dom}(\Delta'; (\alpha : A)^{\leq 1}) \cup \{\alpha\} \subseteq \mathsf{fn}(t) \cup \{\alpha\} = \mathsf{fn}(o)$.

- (REPL). Luego, se tiene $o = c[\![\alpha \setminus^{\alpha'} s]\!]$, $\Gamma = \Gamma' \cup \Gamma''$ y $\Delta = \Delta' \cup \Delta''; \alpha' : B$ con las derivaciones $\pi_c \rhd_{\mathcal{E}} \Gamma' \vdash c \mid \Delta'; (\alpha : S \to B)^{\leq 1}; (\alpha' : B)^{\leq 1}$ y $\pi_s \rhd_{\mathcal{E}} \Gamma'' \vdash s : S \mid \Delta''; (\alpha' : B)^{\leq 1}$. Por *h.i.* se tiene por un lado $\mathsf{dom}(\Gamma') \subseteq \mathsf{fv}(c)$ y $\mathsf{dom}(\Delta'; (\alpha : S \to B)^{\leq 1}; (\alpha' : B)^{\leq 1}) \subseteq \mathsf{fn}(c)$, y por otro $\mathsf{dom}(\Gamma'') \subseteq \mathsf{fv}(s)$ y $\mathsf{dom}(\Delta''; (\alpha' : B)^{\leq 1}) \subseteq \mathsf{fn}(s)$. Se concluye pues $\mathsf{dom}(\Gamma) = \mathsf{dom}(\Gamma') \cup \mathsf{dom}(\Gamma'') \subseteq \mathsf{fv}(c) \cup \mathsf{fv}(s) = \mathsf{fv}(o)$ y $\mathsf{dom}(\Delta) = \mathsf{dom}(\Delta') \cup \mathsf{dom}(\Delta'') \cup \{\alpha'\} = (\mathsf{dom}(\Delta'; (\alpha : S \to B)^{\leq 1}; (\alpha' : B)^{\leq 1}) \setminus \{\alpha\}) \cup \mathsf{dom}(\Delta''; (\alpha' : B)^{\leq 1}) \cup \{\alpha'\} \subseteq (\mathsf{fn}(c) \setminus \{\alpha\}) \cup \mathsf{fn}(s) \cup \{\alpha'\} = \mathsf{fn}(o)$.

- (REN). Luego, $o = c[\alpha \setminus \beta]$, $\Delta = \Delta'; \beta : A$ con $\pi_c \rhd_{\mathcal{E}} \Gamma \vdash c \mid \Delta'; (\alpha : A)^{\leq 1}; (\beta : A)^{\leq 1}$. Por *h.i.* se tienen $\mathsf{dom}(\Gamma) \subseteq \mathsf{fv}(c)$ y $\mathsf{dom}(\Delta'; (\alpha : A)^{\leq 1}; (\beta : A)^{\leq 1}) \subseteq \mathsf{fn}(c)$. Se concluye pues $\mathsf{dom}(\Gamma) \subseteq \mathsf{fv}(c) = \mathsf{fv}(o)$ y $\mathsf{dom}(\Delta) = \mathsf{dom}(\Delta') \cup \{\beta\} = (\mathsf{dom}(\Delta'; (\alpha : A)^{\leq 1}; (\beta : A)^{\leq 1}) \setminus \{\alpha\}) \cup \{\beta\} \subseteq (\mathsf{fn}(c) \setminus \{\alpha\}) \cup \{\beta\} = \mathsf{fn}(o)$.

- (STK). Luego, $o = t \cdot s$, $T = A \cdot S$, $\Gamma = \Gamma' \cup \Gamma''$ y $\Delta = \Delta' \cup \Delta''$ con $\pi_t \rhd_{\mathcal{E}} \Gamma' \vdash t : A \mid \Delta'$ y $\pi_s \rhd_{\mathcal{E}} \Gamma'' \vdash s : S \mid \Delta''$. Por *h.i.* se tienen $\mathsf{dom}(\Gamma') \subseteq \mathsf{fv}(t)$, $\mathsf{dom}(\Delta') \subseteq \mathsf{fn}(t)$, $\mathsf{dom}(\Gamma'') \subseteq \mathsf{fv}(s)$ y $\mathsf{dom}(\Delta'') \subseteq \mathsf{fn}(s)$. Se concluye pues $\mathsf{dom}(\Gamma) = \mathsf{dom}(\Gamma') \cup \mathsf{dom}(\Gamma'') \subseteq \mathsf{fv}(t) \cup \mathsf{fv}(s) = \mathsf{fv}(o)$ y $\mathsf{dom}(\Delta) = \mathsf{dom}(\Delta') \cup \mathsf{dom}(\Delta'') \subseteq \mathsf{fn}(t) \cup \mathsf{fn}(s) = \mathsf{fn}(o)$.

□

Lema C.0.21 (Sustitución). *Sean $\pi_o \rhd_{\mathcal{E}} \Gamma; (x : B)^{\leq 1} \vdash o : T \mid \Delta$ y $\pi_u \rhd_{\mathcal{E}} \Gamma' \vdash u : B \mid \Delta'$. Luego, existe $\pi_{\{x \setminus u\}o} \rhd_{\mathcal{E}} \Gamma^* \vdash \{x \setminus u\}o : T \mid \Delta^*$ tal que $\Gamma^* \subseteq \Gamma \cup \Gamma'$ y $\Delta^* \subseteq \Delta \cup \Delta'$.*

Demostración. Por inducción en o.

- $o = x$. Luego, $\{x \setminus u\}o = u$. Más aún, por (VAR), $\Gamma = \emptyset$ y $\Delta = \emptyset$. Se concluye con $\pi_{\{x \setminus u\}o} = \pi_u$, $\Gamma^* = \Gamma'$ y $\Delta^* = \Delta'$.

- $o = y \neq x$. Luego, $\{x \setminus u\}o = y$. Más aún, por (VAR), $\Gamma = \{y : A\}$ para algún A, $(x : B)^{\leq 1}$ es vacío y $\Delta = \emptyset$. Se concluye con $\pi_{\{x \setminus u\}o} = \pi_o$, $\Gamma^* = \Gamma$ y $\Delta^* = \Delta$.

- $o = t\,v$. Luego, $\{x \setminus u\}o = \{x \setminus u\}t\,\{x \setminus u\}v$. Más aún, por (APP), se tienen $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta = \Delta_0 \cup \Delta_1$ con $\pi_t \rhd_{\mathcal{E}} \Gamma_0 \vdash t : A \to T \mid \Delta_0$ y $\pi_v \rhd_{\mathcal{E}} \Gamma_1 \vdash v : A \mid \Delta_1$. Por *h.i.* existen $\pi_{\{x \setminus u\}t} \rhd_{\mathcal{E}} \Gamma_0^* \vdash \{x \setminus u\}t : A \to T \mid \Delta_0^*$ y $\pi_{\{x \setminus u\}v} \rhd_{\mathcal{E}} \Gamma_1^* \vdash \{x \setminus u\}v : A \mid \Delta_1^*$ tal que $\Gamma_0^* \subseteq \Gamma_0 \cup \Gamma'$, $\Delta_0^* \subseteq \Delta_0 \cup \Delta'$, $\Gamma_1^* \subseteq \Gamma_1 \cup \Gamma'$ y $\Delta_1^* \subseteq \Delta_1 \cup \Delta'$. Finalmente, concluye por (APP) con $\pi_{\{x \setminus u\}o} \rhd_{\mathcal{E}} \Gamma_0^* \cup \Gamma_1^* \vdash \{x \setminus u\}o : T \mid \Delta_0^* \cup \Delta_1^*$. Notar que $\Gamma^* = \Gamma_0^* \cup \Gamma_1^* \subseteq \Gamma_0 \cup \Gamma_1 \cup \Gamma' = \Gamma \cup \Gamma'$ y $\Delta^* = \Delta_0^* \cup \Delta_1^* \subseteq \Delta_0 \cup \Delta_1 \cup \Delta' = \Delta \cup \Delta'$.

- $o = \lambda y.t$ con $x \neq y$ e $y \notin u$. Luego, $\{x \setminus u\}o = \lambda y.\{x \setminus u\}t$. Más aún, por (ABS), se tiene $T = A' \to A$ con $\pi_t \rhd_{\mathcal{E}} \Gamma; (y : A')^{\leq 1} \vdash t : A \mid \Delta$. Por *h.i.* existe una derivación $\pi_{\{x \setminus u\}t} \rhd_{\mathcal{E}} \Gamma^*; (y : A')^{\leq 1} \vdash \{x \setminus u\}t : A \mid \Delta^*$ tal que $\Gamma^*; (y : A')^{\leq 1} \subseteq (\Gamma; (y : A')^{\leq 1}) \cup \Gamma'$ y $\Delta^* \subseteq \Delta \cup \Delta'$. Por último, se concluye por (ABS) nuevamente, $\pi_{\{x \setminus u\}o} \rhd_{\mathcal{E}} \Gamma^* \vdash \{x \setminus u\}o : T \mid \Delta^*$.

- $o = \mu\alpha.c$ con $\alpha \notin u$. Luego, $\{x \setminus u\}o = \mu\alpha.\{x \setminus u\}c$. Más aún, por (CONT), $\pi_c \rhd_{\mathcal{E}} \Gamma \vdash c \mid \Delta; (\alpha : T)^{\leq 1}$. Por *h.i.* existe $\pi_{\{x \setminus u\}c} \rhd_{\mathcal{E}} \Gamma^* \vdash \{x \setminus u\}c \mid \Delta^*; (\alpha : T)^{\leq 1}$ tal que $\Gamma^* \subseteq \Gamma \cup \Gamma'$ y $\Delta^*; (\alpha : T)^{\leq 1} \subseteq (\Delta; (\alpha : T)^{\leq 1}) \cup \Delta'$. Finalmente, se concluye por (CONT), $\pi_{\{x \setminus u\}o} \rhd_{\mathcal{E}} \Gamma^* \vdash \{x \setminus u\}o \mid \Delta^*$.

- $o = t[y \setminus v]$ con $x \neq y$ e $y \notin u$. Luego, $\{x \setminus u\}o = \{x \setminus u\}t[y \setminus \{x \setminus u\}v]$. Más aún, por (SUBS), $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta = \Delta_0 \cup \Delta_1$ con $\pi_t \rhd_{\mathcal{E}} \Gamma_0; (y : A)^{\leq 1} \vdash t : T \mid \Delta_0$ y $\pi_v \rhd_{\mathcal{E}} \Gamma_1 \vdash v : A \mid \Delta_1$. Por *h.i.* existen $\pi_{\{x \setminus u\}t} \rhd_{\mathcal{E}} \Gamma_0^*; (y : A)^{\leq 1} \vdash \{x \setminus u\}t : T \mid \Delta_0^*$ y $\pi_{\{x \setminus u\}v} \rhd_{\mathcal{E}} \Gamma_1^* \vdash \{x \setminus u\}v : A \mid \Delta_1^*$ tal que $\Gamma_0^*; (y : A')^{\leq 1} \subseteq (\Gamma_0; (y : A')^{\leq 1}) \cup \Gamma'$, $\Delta_0^* \subseteq \Delta_0 \cup \Delta'$, $\Gamma_1^* \subseteq \Gamma_1 \cup \Gamma'$ y $\Delta_1^* \subseteq \Delta_1 \cup \Delta'$. Finalmente, se concluye por (SUBS) con $\pi_{\{x \setminus u\}o} \rhd_{\mathcal{E}} \Gamma_0^* \cup \Gamma_1^* \vdash \{x \setminus u\}o : T \mid \Delta_0^* \cup \Delta_1^*$. Notar que $\Gamma^* = \Gamma_0^* \cup \Gamma_1^* \subseteq \Gamma_0 \cup \Gamma_1 \cup \Gamma' = \Gamma \cup \Gamma'$ y $\Delta^* = \Delta_0^* \cup \Delta_1^* \subseteq \Delta_0 \cup \Delta_1 \cup \Delta' = \Delta \cup \Delta'$.

- $o = [\alpha]\, t$. Luego, $\{x \setminus u\}o = [\alpha]\, \{x \setminus u\}t$. Más aún, por (NAME), se tiene $\Delta = \Delta_0; \alpha : T$ con $\pi_t \rhd_{\mathcal{E}} \Gamma \vdash t : T \mid \Delta_0; (\alpha : T)^{\leq 1}$. Por *h.i.* existe $\pi_{\{x\setminus u\}t} \rhd_{\mathcal{E}} \Gamma^* \vdash \{x \setminus u\}t : T \mid \Delta_0^*; (\alpha : T)^{\leq 1}$ tal que $\Gamma^* \subseteq \Gamma \cup \Gamma'$ y $\Delta_0^*; (\alpha : T)^{\leq 1} \subseteq (\Delta_0; (\alpha : T)^{\leq 1}) \cup \Delta'$. Por último, se concluye por (NAME) nuevamente, $\pi_{\{x\setminus u\}o} \rhd_{\mathcal{E}} \Gamma^* \vdash \{x \setminus u\}o \mid \Delta^*$ con $\Gamma^* \subseteq \Gamma \cup \Gamma'$ y $\Delta^* = \Delta_0^*; \alpha : T \subseteq (\Delta_0; \alpha : T) \cup \Delta' = \Delta \cup \Delta'$.

- $o = c[\![\alpha \setminus^{\alpha'} s]\!]$ con $\alpha \notin u$. Luego, $\{x \setminus u\}o = \{x \setminus u\}c[\![\alpha \setminus^{\alpha'} \{x \setminus u\}s]\!]$. Más aún, por (REPL), se tienen $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta = \Delta_0 \cup \Delta_1; \alpha' : A$ con derivaciones $\pi_c \rhd_{\mathcal{E}} \Gamma_0 \vdash c \mid \Delta_0; (\alpha : S \to A)^{\leq 1}; (\alpha' : A)^{\leq 1}$ y $\pi_s \rhd_{\mathcal{E}} \Gamma_1 \vdash s : S \mid \Delta_1; (\alpha' : A)^{\leq 1}$. Entonces, por *h.i.* con π_c y π_s, existen derivaciones $\pi_{\{x\setminus u\}c} \rhd_{\mathcal{E}} \Gamma_0^* \vdash \{x \setminus u\}c \mid \Delta_0^*; (\alpha : S \to A)^{\leq 1}; (\alpha' : A)^{\leq 1}$ y $\pi_{\{x\setminus u\}s} \rhd_{\mathcal{E}} \Gamma_1^* \vdash \{x \setminus u\}s : S \mid \Delta_1^*; (\alpha' : A)^{\leq 1}$ con contextos $\Gamma_0^* \subseteq \Gamma_0 \cup \Gamma'$, $\Delta_0^*; (\alpha : S \to A)^{\leq 1}; (\alpha' : A)^{\leq 1} \subseteq (\Delta_0; (\alpha : S \to A)^{\leq 1}; (\alpha' : A)^{\leq 1}) \cup \Delta'$, $\Gamma_1^* \subseteq \Gamma_1 \cup \Gamma'$ y $\Delta_1^*; (\alpha' : A)^{\leq 1} \subseteq (\Delta_1; (\alpha' : A)^{\leq 1}) \cup \Delta'$. Finalmente, se concluye por (REPL), $\pi_{\{x\setminus u\}o} \rhd_{\mathcal{E}} \Gamma^* \vdash \{x \setminus u\}o \mid \Delta^*$ con $\Gamma^* = \Gamma_0^* \cup \Gamma_1^* \subseteq \Gamma_0 \cup \Gamma_1 \cup \Gamma' = \Gamma \cup \Gamma'$ y $\Delta^* = \Delta_0^* \cup \Delta_1^*; \alpha' : A \subseteq (\Delta_0 \cup \Delta_1; \alpha' : A) \cup \Delta' = \Delta \cup \Delta'$.

- $o = c[\alpha \setminus \beta]$ con $\alpha \notin u$. Luego, $\{x \setminus u\}o = \{x \setminus u\}c[\alpha \setminus \beta]$. Más aún, por (REN), se tiene $\Delta = \Delta_0; \beta : A$ con $\pi_c \rhd_{\mathcal{E}} \Gamma \vdash c \mid \Delta_0; (\alpha : A)^{\leq 1}; (\beta : A)^{\leq 1}$. Entonces, por *h.i.* existe una derivación $\pi_{\{x\setminus u\}c} \rhd_{\mathcal{E}} \Gamma^* \vdash \{x \setminus u\}c \mid \Delta_0^*; (\alpha : A)^{\leq 1}; (\beta : A)^{\leq 1}$ tal que $\Gamma^* \subseteq \Gamma \cup \Gamma'$ y $\Delta_0^*; (\alpha : A)^{\leq 1}; (\beta : A)^{\leq 1} \subseteq (\Delta_0; (\alpha : A)^{\leq 1}; (\beta : A)^{\leq 1}) \cup \Delta'$. Finalmente, se concluye por (REN), $\pi_{\{x\setminus u\}o} \rhd_{\mathcal{E}} \Gamma^* \vdash \{x \setminus u\}o \mid \Delta^*$ con $\Gamma^* \subseteq \Gamma \cup \Gamma'$ y $\Delta^* = \Delta_0^*; \beta : A \subseteq (\Delta_0; \beta : A) \cup \Delta' = \Delta \cup \Delta'$.

- $o = t \cdot s$. Luego, $\{x \setminus u\}o = \{x \setminus u\}t \cdot \{x \setminus u\}s$. Más aún, por (STK), se tienen $T = A \cdot S$, $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta = \Delta_0 \cup \Delta_1$ con $\pi_t \rhd_{\mathcal{E}} \Gamma_0 \vdash t : A \mid \Delta_0$ y $\pi_s \rhd_{\mathcal{E}} \Gamma_1 \vdash s : S \mid \Delta_1$. Por *h.i.* existen $\pi_{\{x\setminus u\}t} \rhd_{\mathcal{E}} \Gamma_0^* \vdash \{x \setminus u\}t : A \mid \Delta_0^*$ y $\pi_{\{x\setminus u\}s} \rhd_{\mathcal{E}} \Gamma_1^* \vdash \{x \setminus u\}s : S \mid \Delta_1^*$ tal que $\Gamma_0^* \subseteq \Gamma_0 \cup \Gamma'$, $\Delta_0^* \subseteq \Delta_0 \cup \Delta'$, $\Gamma_1^* \subseteq \Gamma_1 \cup \Gamma'$ y $\Delta_1^* \subseteq \Delta_1 \cup \Delta'$. Finalmente, concluye por (STK) con $\pi_{\{x\setminus u\}o} \rhd_{\mathcal{E}} \Gamma_0^* \cup \Gamma_1^* \vdash \{x \setminus u\}o : T \mid \Delta_0^* \cup \Delta_1^*$. Notar que $\Gamma^* = \Gamma_0^* \cup \Gamma_1^* \subseteq \Gamma_0 \cup \Gamma_1 \cup \Gamma' = \Gamma \cup \Gamma'$ y $\Delta^* = \Delta_0^* \cup \Delta_1^* \subseteq \Delta_0 \cup \Delta_1 \cup \Delta' = \Delta \cup \Delta'$.

□

Lema C.0.22 (Replacement)**.** *Sean* $\pi_o \rhd_{\mathcal{E}} \Gamma \vdash o : T \mid \Delta; (\alpha : S \to B)^{\leq 1}$ *y* $\pi_s \rhd_{\mathcal{E}} \Gamma' \vdash s : S \mid \Delta'$. *Luego, existe* $\pi_{\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o} \rhd_{\mathcal{E}} \Gamma^* \vdash \{\!\{\alpha \setminus^{\alpha'} s\}\!\}o : T \mid \Delta^*$ *tal que* $\Gamma^* \subseteq \Gamma \cup \Gamma'$ *y* $\Delta^* \subseteq \Delta \cup \Delta' \cup \alpha' : B$.

Demostración. Por inducción en o.

- $o = x$. Luego, $\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o = x$. Más aún, por (VAR), se tienen $\Gamma = \{x : T\}$ y $\Delta = \emptyset$. Se concluye con $\pi_{\{\!\{\alpha\setminus^{\alpha'} s\}\!\}o} = \pi_o$, $\Gamma^* = \Gamma$ y $\Delta^* = \Delta$.

- $o = t\, u$. Luego, $\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o = \{\!\{\alpha \setminus^{\alpha'} s\}\!\}t\, \{\!\{\alpha \setminus^{\alpha'} s\}\!\}u$. Más aún, por (APP), $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta = \Delta_0 \cup \Delta_1$ con $\pi_t \rhd_{\mathcal{E}} \Gamma_0 \vdash t : A \to T \mid \Delta_0; (\alpha : S \to B)^{\leq 1}$ y $\pi_u \rhd_{\mathcal{E}} \Gamma_1 \vdash u : A \mid \Delta_1; (\alpha : S \to B)^{\leq 1}$. Por *h.i.* existen $\pi_{\{\!\{\alpha\setminus^{\alpha'} s\}\!\}t} \rhd_{\mathcal{E}} \Gamma_0^* \vdash \{\!\{\alpha \setminus^{\alpha'} s\}\!\}t : A \to T \mid \Delta_0^*$ y $\pi_{\{\!\{\alpha\setminus^{\alpha'} s\}\!\}u} \rhd_{\mathcal{E}} \Gamma_1^* \vdash \{\!\{\alpha \setminus^{\alpha'} s\}\!\}u : A \mid \Delta_1^*$ tal que $\Gamma_0^* \subseteq \Gamma_0 \cup \Gamma'$, $\Delta_0^* \subseteq \Delta_0 \cup \Delta' \cup \alpha' : B$, $\Gamma_1^* \subseteq \Gamma_1 \cup \Gamma'$ y $\Delta_1^* \subseteq \Delta_1 \cup \Delta' \cup \alpha' : B$. Se concluye por (APP), $\pi_{\{\!\{\alpha\setminus^{\alpha'} s\}\!\}o} \rhd_{\mathcal{E}} \Gamma_0^* \cup \Gamma_1^* \vdash \{\!\{\alpha \setminus^{\alpha'} s\}\!\}o : T \mid \Delta_0^* \cup \Delta_1^*$. Notar que $\Gamma^* = \Gamma_0^* \cup \Gamma_1^* \subseteq \Gamma_0 \cup \Gamma_1 \cup \Gamma' = \Gamma \cup \Gamma'$ y $\Delta^* = \Delta_0^* \cup \Delta_1^* \subseteq \Delta_0 \cup \Delta_1 \cup \Delta' \cup \alpha' : B = \Delta \cup \Delta' \cup \alpha' : B$.

- $o = \lambda x.t$ con $x \notin s$. Luego, $\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o = \lambda x.\{\!\{\alpha \setminus^{\alpha'} s\}\!\}t$. Más aún, por (ABS), se tienen $T = A' \to A$ y $\pi_t \rhd_{\mathcal{E}} \Gamma; (x : A')^{\leq 1} \vdash t : A \mid \Delta; (\alpha : S \to B)^{\leq 1}$. Por *h.i.* existe una derivación $\pi_{\{\!\{\alpha\setminus^{\alpha'} s\}\!\}t} \rhd_{\mathcal{E}} \Gamma^*; (x : A')^{\leq 1} \vdash \{\!\{\alpha \setminus^{\alpha'} s\}\!\}t : A \mid \Delta^*$ tal que $\Gamma^*; (x : A')^{\leq 1} \subseteq (\Gamma; (x : A')^{\leq 1}) \cup \Gamma'$ y $\Delta^* \subseteq \Delta \cup \Delta' \cup \alpha' : B$. Finalmente, se concluye por (ABS), $\pi_{\{\!\{\alpha\setminus^{\alpha'} s\}\!\}o} \rhd_{\mathcal{E}} \Gamma^* \vdash \{\!\{\alpha \setminus^{\alpha'} s\}\!\}o : T \mid \Delta^*$.

- $o = \mu\beta.c$ con $\alpha \neq \beta$, $\alpha' \neq \beta$ y $\beta \notin s$. Luego, $\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o = \mu\beta.\{\!\{\alpha \setminus^{\alpha'} s\}\!\}c$. Más aún, por (CONT), se tiene la derivación $\pi_c \rhd_{\mathcal{E}} \Gamma \vdash c \mid \Delta; (\alpha : S \to B)^{\leq 1}; (\beta : T)^{\leq 1}$. Entonces, por *h.i.* existe otra derivación $\pi_{\{\!\{\alpha\setminus^{\alpha'} s\}\!\}c} \rhd_{\mathcal{E}} \Gamma^* \vdash \{\!\{\alpha \setminus^{\alpha'} s\}\!\}c \mid \Delta^*; (\beta : T)^{\leq 1}$ tal que $\Gamma^* \subseteq \Gamma \cup \Gamma'$ y $\Delta^*; (\beta : T)^{\leq 1} \subseteq (\Delta; (\beta : T)^{\leq 1}) \cup \Delta' \cup \alpha' : B$. Luego, se concluye por (CONT), $\pi_{\{\!\{\alpha\setminus^{\alpha'} s\}\!\}o} \rhd_{\mathcal{E}} \Gamma^* \vdash \{\!\{\alpha \setminus^{\alpha'} s\}\!\}o \mid \Delta^*$.

- $o = t[x \setminus u]$ con $x \notin s$. Luego, $\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o = \{\!\{\alpha \setminus^{\alpha'} s\}\!\} t[x \setminus \{\!\{\alpha \setminus^{\alpha'} s\}\!\} u]$. Más aún, por (SUBS), se tienen $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta = \Delta_0 \cup \Delta_1$ con derivaciones $\pi_t \rhd_{\mathcal{E}} \Gamma_0; (x : A)^{\leq 1} \vdash t : T \mid \Delta_0; (\alpha : S \to B)^{\leq 1}$ y $\pi_u \rhd_{\mathcal{E}} \Gamma_1 \vdash u : A \mid \Delta_1; (\alpha : S \to B)^{\leq 1}$. Entonces, por *h.i.* con π_y y π_u, existen dos derivaciones $\pi_{\{\!\{\alpha \setminus^{\alpha'} s\}\!\} t} \rhd_{\mathcal{E}} \Gamma_0^*; (x : A)^{\leq 1} \vdash \{\!\{\alpha \setminus^{\alpha'} s\}\!\} t : T \mid \Delta_0^*$ y $\pi_{\{\!\{\alpha \setminus^{\alpha'} s\}\!\} u} \rhd_{\mathcal{E}} \Gamma_1^* \vdash \{\!\{\alpha \setminus^{\alpha'} s\}\!\} u : A \mid \Delta_1^*$ tal que $\Gamma_0^*; (x : A')^{\leq 1} \subseteq (\Gamma_0; (x : A')^{\leq 1}) \cup \Gamma'$, $\Delta_0^* \subseteq \Delta_0 \cup \Delta' \cup \alpha' : B$, $\Gamma_1^* \subseteq \Gamma_1 \cup \Gamma'$ y $\Delta_1^* \subseteq \Delta_1 \cup \Delta' \cup \alpha' : B$. Finalmente, se concluye por (SUBS), $\pi_{\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o} \rhd_{\mathcal{E}} \Gamma_0^* \cup \Gamma_1^* \vdash \{\!\{\alpha \setminus^{\alpha'} s\}\!\} o : T \mid \Delta_0^* \cup \Delta_1^*$. Notar que $\Gamma^* = \Gamma_0^* \cup \Gamma_1^* \subseteq \Gamma_0 \cup \Gamma_1 \cup \Gamma' = \Gamma \cup \Gamma'$ y $\Delta^* = \Delta_0^* \cup \Delta_1^* \subseteq \Delta_0 \cup \Delta_1 \cup \Delta' \cup \alpha' : B = \Delta \cup \Delta' \cup \alpha' : B$.

- $o = [\alpha]\, t$. Luego, $\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o = [\alpha']\, \{\!\{\alpha \setminus^{\alpha'} s\}\!\} t :: s$. Más aún, por (NAME), se tiene $(\alpha : S \to B)^{\leq 1}$ no vacío en π_o y la derivación $\pi_t \rhd_{\mathcal{E}} \Gamma \vdash t : S \to B \mid \Delta; (\alpha : S \to B)^{\leq 1}$. Por *h.i.* existe una derivación $\pi_{\{\!\{\alpha \setminus^{\alpha'} s\}\!\} t} \rhd_{\mathcal{E}} \Gamma^{**} \vdash \{\!\{\alpha \setminus^{\alpha'} s\}\!\} t : S \to B \mid \Delta^{**}$ tal que $\Gamma^{**} \subseteq \Gamma \cup \Gamma'$ y $\Delta^{**} \subseteq \Delta \cup \Delta' \cup \alpha' : B$. Por Lem. C.0.20 (3), $\pi_{\{\!\{\alpha \setminus^{\alpha'} s\}\!\} t :: s} \rhd_{\mathcal{E}} \Gamma^{**} \cup \Gamma' \vdash \{\!\{\alpha \setminus^{\alpha'} s\}\!\} t :: s : B \mid \Delta^{**} \cup \Delta'$. Finalmente, se concluye por (NAME), $\pi_{\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o} \rhd_{\mathcal{E}} \Gamma^* \vdash \{\!\{\alpha \setminus^{\alpha'} s\}\!\} o \mid \Delta^*$ tomando $\Gamma^* = \Gamma^{**} \cup \Gamma' \subseteq \Gamma \cup \Gamma'$ y $\Delta^* = \Delta^{**} \cup \Delta'; \alpha' : B \subseteq \Delta \cup \Delta' \cup \alpha' : B$.

- $o = [\beta]\, t$ con $\beta \neq \alpha$. Luego, $\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o = [\beta]\, \{\!\{\alpha \setminus^{\alpha'} s\}\!\} t$. Más aún, por (NAME), se tiene $\Delta = \Delta_0; \beta : T$ con la derivación $\pi_t \rhd_{\mathcal{E}} \Gamma \vdash t : T \mid \Delta_0; (\beta : T)^{\leq 1}; (\alpha : S \to B)^{\leq 1}$. Por *h.i.* existe una derivación $\pi_{\{\!\{\alpha \setminus^{\alpha'} s\}\!\} t} \rhd_{\mathcal{E}} \Gamma^* \vdash \{\!\{\alpha \setminus^{\alpha'} s\}\!\} t : T \mid \Delta_0^*; (\beta : T)^{\leq 1}$ tal que $\Gamma^* \subseteq \Gamma \cup \Gamma'$ y $\Delta_0^*; (\beta : T)^{\leq 1} \subseteq (\Delta_0; (\beta : T)^{\leq 1}) \cup \Delta' \cup \alpha' : B$. Por último, se concluye por (NAME) nuevamente, $\pi_{\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o} \rhd_{\mathcal{E}} \Gamma^* \vdash \{\!\{\alpha \setminus^{\alpha'} s\}\!\} o \mid \Delta^*$ con $\Gamma^* \subseteq \Gamma \cup \Gamma'$ y $\Delta^* = \Delta_0^*; \beta : T \subseteq (\Delta_0; \beta : T) \cup \Delta' \cup \alpha' : B = \Delta \cup \Delta' \cup \alpha' : B$.

- $o = c[\![\beta \setminus^{\alpha} s']\!]$ con $\alpha \neq \beta$, $\alpha' \neq \beta$ y $\beta \notin s$. Luego, $\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o = \{\!\{\alpha \setminus^{\alpha'} s\}\!\} c[\![\beta \setminus^{\alpha'} \{\!\{\alpha \setminus^{\alpha'} s\}\!\} s' \cdot s]\!]$. Más aún, por (REPL), se tiene $(\alpha : S \to B)^{\leq 1}$ no vacío en π_o, $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta = \Delta_0 \cup \Delta_1$ con $\pi_c \rhd_{\mathcal{E}} \Gamma_0 \vdash c \mid \Delta_0; (\beta : S' \to S \to B)^{\leq 1}; (\alpha : S \to B)^{\leq 1}$ y $\pi_{s'} \rhd_{\mathcal{E}} \Gamma_1 \vdash s' : S' \mid \Delta_1; (\alpha : S \to B)^{\leq 1}$. Notar que $S' \to S \to B = S' \cdot S \to B$. Luego, por *h.i.* con π_c y $\pi_{s'}$, existen dos derivaciones $\pi_{\{\!\{\alpha \setminus^{\alpha'} s\}\!\} c} \rhd_{\mathcal{E}} \Gamma_0^* \vdash \{\!\{\alpha \setminus^{\alpha'} s\}\!\} c \mid \Delta_0^*; (\beta : S' \cdot S \to B)^{\leq 1}$ y $\pi_{\{\!\{\alpha \setminus^{\alpha'} s\}\!\} s'} \rhd_{\mathcal{E}} \Gamma_1^* \vdash \{\!\{\alpha \setminus^{\alpha'} s\}\!\} s' : S' \mid \Delta_1^*$ tal que se tienen $\Gamma_0^* \subseteq \Gamma_0 \cup \Gamma'$, $\Delta_0^*; (\beta : S' \cdot S \to B)^{\leq 1} \subseteq (\Delta_0; (\beta : S' \cdot S \to B)^{\leq 1}) \cup \Delta' \cup \alpha' : B$, $\Gamma_1^* \subseteq \Gamma_1 \cup \Gamma'$ y $\Delta_1^* \subseteq \Delta_1 \cup \Delta' \cup \alpha' : B$. Por Lem. C.0.20 (1) con $\pi_{\{\!\{\alpha \setminus^{\alpha'} s\}\!\} s'}$ y π_s, se tiene la derivación $\pi_{\{\!\{\alpha \setminus^{\alpha'} s\}\!\} s' \cdot s} \rhd_{\mathcal{E}} \Gamma_1^* \cup \Gamma' \vdash \{\!\{\alpha \setminus^{\alpha'} s\}\!\} s' \cdot s : S' \cdot S \mid \Delta_1^* \cup \Delta'$. Finalmente, aplicando (REPL) con $\pi_{\{\!\{\alpha \setminus^{\alpha'} s\}\!\} c}$, se concluye $\pi_{\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o} \rhd_{\mathcal{E}} \Gamma^* \vdash \{\!\{\alpha \setminus^{\alpha'} s\}\!\} o \mid \Delta^*$ tomando $\Gamma^* = \Gamma_0^* \cup \Gamma_1^* \cup \Gamma' \subseteq \Gamma_0 \cup \Gamma_1 \cup \Gamma' = \Gamma \cup \Gamma'$ y $\Delta^* = \Delta_0^* \cup \Delta_1^* \cup \Delta'; \alpha' : B \subseteq \Delta_0 \cup \Delta_1 \cup \Delta' \cup \alpha' : B = \Delta \cup \Delta' \cup \alpha' : B$.

- $o = c[\![\beta \setminus^{\beta'} s']\!]$ con $\alpha \neq \beta$, $\alpha \neq \beta'$, $\alpha' \neq \beta$ y $\beta \notin s$. Luego, se tiene $\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o = \{\!\{\alpha \setminus^{\alpha'} s\}\!\} c[\![\beta \setminus^{\beta'} \{\!\{\alpha \setminus^{\alpha'} s\}\!\} s']\!]$. Más aún, por (REPL), se tienen $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta = \Delta_0 \cup \Delta_1; \beta' : A$ con las derivaciones $\pi_c \rhd_{\mathcal{E}} \Gamma_0 \vdash c \mid \Delta_0; (\beta : S' \to A)^{\leq 1}; (\beta' : A)^{\leq 1}$ y $\pi_{s'} \rhd_{\mathcal{E}} \Gamma_1 \vdash s' : S' \mid \Delta_1; (\beta' : A)^{\leq 1}$. Por *h.i.* con π_c, existe una derivación $\pi_{\{\!\{\alpha \setminus^{\alpha'} s\}\!\} c} \rhd_{\mathcal{E}} \Gamma_0^* \vdash \{\!\{\alpha \setminus^{\alpha'} s\}\!\} c \mid \Delta_0^*; (\beta : S' \to A)^{\leq 1}; (\beta' : A)^{\leq 1}$ con $\Gamma_0^* \subseteq \Gamma_0 \cup \Gamma'$ y $\Delta_0^*; (\beta : S' \to A)^{\leq 1}; (\beta' : A)^{\leq 1} \subseteq (\Delta_0; (\beta : S' \to A)^{\leq 1}; (\beta' : A)^{\leq 1}) \cup \Delta' \cup \alpha' : B$. Por *h.i.* sobre $\pi_{s'}$ existe $\pi_{\{\!\{\alpha \setminus^{\alpha'} s\}\!\} s'} \rhd_{\mathcal{E}} \Gamma_1^* \vdash \{\!\{\alpha \setminus^{\alpha'} s\}\!\} s' : S' \mid \Delta_1^*; (\beta' : A)^{\leq 1}$ tal que $\Gamma_1^* \subseteq \Gamma_1 \cup \Gamma'$ y $\Delta_1^*; (\beta' : A)^{\leq 1} \subseteq (\Delta_1; (\beta' : A)^{\leq 1}) \cup \Delta' \cup \alpha' : B$. Finalmente, se concluye por (REPL), $\pi_{\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o} \rhd_{\mathcal{E}} \Gamma^* \vdash \{\!\{\alpha \setminus^{\alpha'} s\}\!\} o \mid \Delta^*$ con $\Gamma^* = \Gamma_0^* \cup \Gamma_1^* \subseteq \Gamma_0 \cup \Gamma_1 \cup \Gamma' = \Gamma \cup \Gamma'$ y $\Delta^* = \Delta_0^* \cup \Delta_1^*; \beta' : A \subseteq (\Delta_0 \cup \Delta_1; \beta' : A) \cup \Delta' \cup \alpha' : B = \Delta \cup \Delta' \cup \alpha' : B$.

- $o = c[\beta \setminus \alpha]$ con $\alpha \neq \beta$, $\alpha' \neq \beta$ y $\beta \notin s$. Luego, $\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o = \{\!\{\alpha \setminus^{\alpha'} s\}\!\} c[\![\beta \setminus^{\gamma} s]\!][\gamma \setminus \alpha']$ con γ un nombre fresco. Más aún, por (REN), se tiene $(\alpha : S \to B)^{\leq 1}$ no vacío en π_o y la derivación $\pi_c \rhd_{\mathcal{E}} \Gamma \vdash c \mid \Delta; (\beta : S \to B)^{\leq 1}; (\alpha : S \to B)^{\leq 1}$. Entonces, por *h.i.* existe una derivación $\pi_{\{\!\{\alpha \setminus^{\alpha'} s\}\!\} c} \rhd_{\mathcal{E}} \Gamma^{**} \vdash \{\!\{\alpha \setminus^{\alpha'} s\}\!\} c \mid \Delta^{**}; (\beta : S \to B)^{\leq 1}$ tal que $\Gamma^{**} \subseteq \Gamma \cup \Gamma'$ y $\Delta^{**}; (\beta : S \to B)^{\leq 1} \subseteq (\Delta; (\beta : S \to B)^{\leq 1}) \cup \Delta' \cup \alpha' : B$. Por (REPL) con π_s se obtiene la derivación $\pi_{\{\!\{\alpha \setminus^{\alpha'} s\}\!\} c[\![\beta \setminus \gamma\, s]\!]} \rhd_{\mathcal{E}}$

$\Gamma^{**} \cup \Gamma' \vdash \{\!\{\alpha \setminus^{\alpha'} s\}\!\} c[\![\beta \setminus^{\gamma} s]\!] \mid \Delta^{**} \cup \Delta'; \gamma : B$. Finalmente, se concluye por (REN), $\pi_{\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o} \rhd_{\mathcal{E}} \Gamma^* \vdash \{\!\{\alpha \setminus^{\alpha'} s\}\!\} o \mid \Delta^*$ con $\Gamma^* = \Gamma^{**} \cup \Gamma' \subseteq \Gamma \cup \Gamma'$ y $\Delta^* = \Delta^{**} \cup \Delta'; \alpha' : B \subseteq \Delta \cup \Delta' \cup \alpha' : B$.

- $o = c[\beta \setminus \beta']$ con $\alpha \neq \beta$, $\alpha \neq \beta'$, $\alpha' \neq \beta$ y $\beta \notin s$. Luego, $\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o = \{\!\{\alpha \setminus^{\alpha'} s\}\!\} c[\beta \setminus \beta']$. Más aún, por (REN), se tiene $\Delta = \Delta_0; \beta' : A$ y la derivación $\pi_c \rhd_{\mathcal{E}} \Gamma \vdash c \mid \Delta_0; (\beta : A)^{\leq 1}; (\beta' : A)^{\leq 1}$. Por *h.i.* existe una derivación $\pi_{\{\!\{\alpha \setminus^{\alpha'} s\}\!\} c} \rhd_{\mathcal{E}} \Gamma^* \vdash \{\!\{\alpha \setminus^{\alpha'} s\}\!\} c \mid \Delta_0^*; (\beta : A)^{\leq 1}; (\beta' : A)^{\leq 1}$ tal que $\Gamma^* \subseteq \Gamma \cup \Gamma'$ y $\Delta_0^*; (\beta : A)^{\leq 1}; (\beta' : A)^{\leq 1} \subseteq (\Delta_0; (\beta : A)^{\leq 1}; (\beta' : A)^{\leq 1}) \cup \Delta' \cup \alpha' : B$. Finalmente, se concluye por (REN), $\pi_{\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o} \rhd_{\mathcal{E}} \Gamma^* \vdash \{\!\{\alpha \setminus^{\alpha'} s\}\!\} o \mid \Delta^*$ con $\Gamma^* \subseteq \Gamma \cup \Gamma'$ y $\Delta^* = \Delta_0^*; \beta' : A \subseteq (\Delta_0; \beta' : A) \cup \Delta' \cup \alpha' : B = \Delta \cup \Delta' \cup \alpha' : B$.

- $o = t \cdot s'$. Luego, $\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o = \{\!\{\alpha \setminus^{\alpha'} s\}\!\} t \cdot \{\!\{\alpha \setminus^{\alpha'} s\}\!\} s'$. Más aún, por (STK), se tienen $T = A \cdot S'$, $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta = \Delta_0 \cup \Delta_1$ con $\pi_t \rhd_{\mathcal{E}} \Gamma_0 \vdash t : A \mid \Delta_0$ y $\pi_{s'} \rhd_{\mathcal{E}} \Gamma_1 \vdash s' : S' \mid \Delta_1$. Por *h.i.* existen $\pi_{\{\!\{\alpha \setminus^{\alpha'} s\}\!\} t} \rhd_{\mathcal{E}} \Gamma_0^* \vdash \{\!\{\alpha \setminus^{\alpha'} s\}\!\} t : A \mid \Delta_0^*$ y $\pi_{\{\!\{\alpha \setminus^{\alpha'} s\}\!\} s'} \rhd_{\mathcal{E}} \Gamma_1^* \vdash \{\!\{\alpha \setminus^{\alpha'} s\}\!\} s' : S' \mid \Delta_1^*$ tal que $\Gamma_0^* \subseteq \Gamma_0 \cup \Gamma'$, $\Delta_0^* \subseteq \Delta_0 \cup \Delta' \cup \alpha' : B$, $\Gamma_1^* \subseteq \Gamma_1 \cup \Gamma'$ y $\Delta_1^* \subseteq \Delta_1 \cup \Delta' \cup \alpha' : B$. Finalmente, concluye por (STK) con $\pi_{\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o} \rhd_{\mathcal{E}} \Gamma_0^* \cup \Gamma_1^* \vdash \{\!\{\alpha \setminus^{\alpha'} s\}\!\} o : T \mid \Delta_0^* \cup \Delta_1^*$. Notar que $\Gamma^* = \Gamma_0^* \cup \Gamma_1^* \subseteq \Gamma_0 \cup \Gamma_1 \cup \Gamma' = \Gamma \cup \Gamma'$ y $\Delta^* = \Delta_0^* \cup \Delta_1^* \subseteq \Delta_0 \cup \Delta_1 \cup \Delta' \cup \alpha' : B = \Delta \cup \Delta' \cup \alpha' : B$.

□

Teorema 4.3.2 (Subject Reduction)**.** *Sea $o \in \mathbb{O}_{\Lambda M}$. Si $\pi \rhd_{\mathcal{E}} \Gamma \vdash o : T \mid \Delta$ y $o \to_{\Lambda M} o'$, entonces existe $\pi' \rhd_{\mathcal{E}} \Gamma' \vdash o' : T \mid \Delta'$ tal que $\Gamma' \subseteq \Gamma$ y $\Delta' \subseteq \Delta$.*

Demostración. Por definición, $o \to_{\Lambda M} o'$ implica $o = \mathtt{O}\langle l \rangle$ y $o' = \mathtt{O}\langle r \rangle$ con $l \mapsto_* r$, $* \in \{\mathtt{dB}, \mathtt{dM}, \mathtt{S}, \mathtt{R}\}$. La demostración es por inducción en $\mathtt{O}$.

- $\mathtt{O} = \Box$. Luego, se tienen tres casos posibles según la regla de reescritura aplicada al reducir.

  1. $\mathtt{dB}$. Luego, $o = \mathtt{L}\langle \lambda x.t \rangle\, u$ y $o' = \mathtt{L}\langle t[x \setminus u] \rangle$. Por (APP) se tienen $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta = \Delta_0 \cup \Delta_1$ con $\pi_{\mathtt{L}\langle \lambda x.t \rangle} \rhd_{\mathcal{E}} \Gamma_0 \vdash \mathtt{L}\langle \lambda x.t \rangle : A \to T \mid \Delta_0$ y $\pi_u \rhd_{\mathcal{E}} \Gamma_1 \vdash u : A \mid \Delta_1$. Se procede por inducción en $\mathtt{L}$.

     - $\mathtt{L} = \Box$. Luego, de $\pi_{\mathtt{L}\langle \lambda x.t \rangle}$ se tiene $\pi_t \rhd_{\mathcal{E}} \Gamma_0; (x : A)^{\leq 1} \vdash t : T \mid \Delta_0$ por (ABS). Se concluye por (SUBS) con π_t y π_u, obteniendo $\pi' \rhd_{\mathcal{E}} \Gamma \vdash t[x \setminus u] : T \mid \Delta$.
     - $\mathtt{L} = \mathtt{L}'[y \setminus v]$ con $y \notin u$. Luego, por Lem. 4.3.1, $y \notin \mathsf{dom}(\Gamma_1)$. De $\pi_{\mathtt{L}\langle \lambda x.t \rangle}$ por (SUBS) se tienen $\Gamma_0 = \Gamma_0' \cup \Gamma_0''$ y $\Delta_0 = \Delta_0' \cup \Delta_0''$ con $\pi_{\mathtt{L}'\langle \lambda x.t \rangle} \rhd_{\mathcal{E}} \Gamma_0'; (y : B)^{\leq 1} \vdash \mathtt{L}'\langle \lambda x.t \rangle : A \to T \mid \Delta_0'$ y $\pi_v \rhd_{\mathcal{E}} \Gamma_0'' \vdash v : B \mid \Delta_0''$. Luego, por (APP) con $\pi_{\mathtt{L}'\langle \lambda x.t \rangle}$ y π_u se obtiene $\pi_{\mathtt{L}'\langle \lambda x.t \rangle\, u} \rhd_{\mathcal{E}} \Gamma_0' \cup \Gamma_1; (y : B)^{\leq 1} \vdash \mathtt{L}'\langle \lambda x.t \rangle\, u : T \mid \Delta_0' \cup \Delta_1$. Por *h.i.* existe una derivación $\pi_{\mathtt{L}'\langle t[x \setminus u] \rangle} \rhd_{\mathcal{E}} \Gamma^*; (y : B)^{\leq 1} \vdash \mathtt{L}'\langle t[x \setminus u] \rangle : T \mid \Delta^*$ tal que $\Gamma^*; (y : B)^{\leq 1} \subseteq \Gamma_0' \cup \Gamma_1; (y : B)^{\leq 1}$ y $\Delta^* \subseteq \Delta_0' \cup \Delta_1$. Finalmente, se concluye por (SUBS) con π_v, $\pi' \rhd_{\mathcal{E}} \Gamma' \vdash \mathtt{L}\langle t[x \setminus u] \rangle : T \mid \Delta'$ con $\Gamma' = \Gamma^* \cup \Gamma_0'' \subseteq \Gamma_0' \cup \Gamma_0'' \cup \Gamma_1 = \Gamma$ y $\Delta' = \Delta^* \cup \Delta_0'' \subseteq \Delta_0' \cup \Delta_0'' \cup \Delta_1 = \Delta$.

  2. $\mathtt{S}$. Luego, $o = t[x \setminus u]$ y $o' = \{x \setminus u\} t$. Por (SUBS) se tienen $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta = \Delta_0 \cup \Delta_1$ con $\pi_t \rhd_{\mathcal{E}} \Gamma_0; (x : A)^{\leq 1} \vdash t : T \mid \Delta_0$ y $\pi_t \rhd_{\mathcal{E}} \Gamma_1 \vdash u : A \mid \Delta_1$. Se concluye pues, por Lem. C.0.21, existe $\pi_{\{x \setminus u\} t} \rhd_{\mathcal{E}} \Gamma' \vdash \{x \setminus u\} t : T \mid \Delta'$ tal que $\Gamma' \subseteq \Gamma_0 \cup \Gamma_1 = \Gamma$ y $\Delta' \subseteq \Delta_0 \cup \Delta_1 = \Delta$.

  3. $\mathtt{dM}$. Luego, $o = \mathtt{L}\langle \mu\alpha.c \rangle\, u$ y $o' = \mathtt{L}\langle \mu\alpha'.c[\![\alpha \setminus^{\alpha'} u]\!] \rangle$ con α' un nombre fresco. Por (APP) se tienen $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta = \Delta_0 \cup \Delta_1$ con $\pi_{\mathtt{L}\langle \mu\alpha.c \rangle} \rhd_{\mathcal{E}} \Gamma_0 \vdash \mathtt{L}\langle \mu\alpha.c \rangle : A \to T \mid \Delta_0$ y $\pi_u \rhd_{\mathcal{E}} \Gamma_1 \vdash u : A \mid \Delta_1$. Se procede por inducción en $\mathtt{L}$.

     - $\mathtt{L} = \Box$. Luego, de $\pi_{\mathtt{L}\langle \mu\alpha.c \rangle}$ se tiene $\pi_c \rhd_{\mathcal{E}} \Gamma_1 \vdash c \mid \Delta_1; (\alpha : A \to T)^{\leq 1}$ por (CONT). Por (REPL) con π_c y π_u se obtiene $\pi_{c[\![\alpha \setminus^{\alpha'} u]\!]} \rhd_{\mathcal{E}} \Gamma \vdash c[\![\alpha \setminus^{\alpha'} u]\!] \mid \Delta; \alpha' : T$. Luego, se concluye por (CONT), $\pi' \rhd_{\mathcal{E}} \Gamma \vdash \mu\alpha'.c[\![\alpha \setminus^{\alpha'} u]\!] : T \mid \Delta$.

  - $\mathtt{L} = \mathtt{L}'[y \backslash v]$ con $y \notin u$. Luego, por Lem. 4.3.1, $y \notin \mathsf{dom}(\Gamma_1)$. De $\pi_{\mathtt{L}\langle \mu\alpha.c\rangle}$ por (SUBS) se tienen $\Gamma_0 = \Gamma_0' \cup \Gamma_0''$ y $\Delta_0 = \Delta_0' \cup \Delta_0''$ con $\pi_{\mathtt{L}'\langle \mu\alpha.c\rangle} \rhd_{\mathcal{E}} \Gamma_0'; (y : B)^{\leq 1} \vdash \mathtt{L}'\langle \mu\alpha.c\rangle : A \to T \mid \Delta_0'$ y $\pi_v \rhd_{\mathcal{E}} \Gamma_0'' \vdash v : B \mid \Delta_0''$. Más aún, por (APP) con π_u se obtiene la derivación $\pi_{\mathtt{L}'\langle \mu\alpha.c\rangle\, u} \rhd_{\mathcal{E}} \Gamma_0' \cup \Gamma_1; (y : B)^{\leq 1} \vdash \mathtt{L}'\langle \mu\alpha.c\rangle\, u : T \mid \Delta_0' \cup \Delta_1$. Luego, por *h.i.* existe $\pi_{\mathtt{L}'\langle \mu\alpha'.c[\![\alpha \backslash^{\alpha'} u]\!]\rangle} \rhd_{\mathcal{E}} \Gamma^*; (y : B)^{\leq 1} \vdash \mathtt{L}'\langle \mu\alpha'.c[\![\alpha \backslash^{\alpha'} u]\!]\rangle : T \mid \Delta^*$ con $\Gamma^*; (y : B)^{\leq 1} \subseteq \Gamma_0' \cup \Gamma_1; (y : B)^{\leq 1}$ y $\Delta^* \subseteq \Gamma_0' \cup \Gamma_1$. Finalmente, se concluye por (SUBS) con π_v, $\pi' \rhd_{\mathcal{E}} \Gamma' \vdash \mathtt{L}\langle \mu\alpha'.c[\![\alpha \backslash^{\alpha'} u]\!]\rangle : T \mid \Delta'$ con $\Gamma' = \Gamma^* \cup \Gamma_0'' \subseteq \Gamma_0' \cup \Gamma_0'' \cup \Gamma_1 = \Gamma$ y $\Delta' = \Delta^* \cup \Delta_0'' \subseteq \Delta_0' \cup \Delta_0'' \cup \Delta_1 = \Delta$.

- $\mathtt{O} = \boxdot$. Luego, la regla de reescritura aplicada es necesariamente R, *i.e.* $o = c[\![\alpha \backslash^{\alpha'} s]\!]$ y $o' = \{\!\{\alpha \backslash^{\alpha'} s\}\!\}c$. Por (REPL) se tienen $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta = \Delta_0 \cup \Delta_1; \alpha' : A$ con derivaciones $\pi_c \rhd_{\mathcal{E}} \Gamma_0 \vdash c \mid \Delta_0; (\alpha : S \to A)^{\leq 1}; (\alpha' : A)^{\leq 1}$ y $\pi_s \rhd_{\mathcal{E}} \Gamma_1 \vdash s : S \mid \Delta_1; (\alpha' : A)^{\leq 1}$. Luego, se concluye pues, por Lem. C.0.22, existe $\pi_{\{\!\{\alpha \backslash^{\alpha'} s\}\!\}c} \rhd_{\mathcal{E}} \Gamma' \vdash \{\!\{\alpha \backslash^{\alpha'} s\}\!\}c \mid \Delta'$ tal que $\Gamma' \subseteq \Gamma_0 \cup \Gamma_1 = \Gamma$ y $\Delta' \subseteq \Delta_0 \cup \Delta_1; \alpha' : A = \Delta$.

- $\mathtt{O} = \mathtt{T}\, u$. Luego, $o = t\, u$ y $o' = t'\, u$ con $t \to_{\Lambda M} t'$. Más aún, por (APP) se tienen $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta = \Delta_0 \cup \Delta_1$ con $\pi_t \rhd_{\mathcal{E}} \Gamma_0 \vdash t : A \to T \mid \Delta_0$ y $\pi_u \rhd_{\mathcal{E}} \Gamma_1 \vdash u : A \mid \Delta_1$. Por *h.i.* existe $\pi_{t'} \rhd_{\mathcal{E}} \Gamma_0' \vdash t' : A \to T \mid \Delta_0'$ con $\Gamma_0' \subseteq \Gamma_0$ y $\Delta_0' \subseteq \Delta_0$. Finalmente, se concluye por (APP), $\pi' \rhd_{\mathcal{E}} \Gamma' \vdash o' : T \mid \Delta'$ con $\Gamma' = \Gamma_0' \cup \Gamma_1 \subseteq \Gamma_0 \cup \Gamma_1 = \Gamma$ y $\Delta' = \Delta_0' \cup \Delta_1 \subseteq \Delta_0 \cup \Delta_1 = \Delta$.

- $\mathtt{O} = t\, \mathtt{T}$. Luego, $o = t\, u$ y $o' = t\, u'$ con $u \to_{\Lambda M} u'$. Más aún, por (APP) se tienen $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta = \Delta_0 \cup \Delta_1$ con $\pi_t \rhd_{\mathcal{E}} \Gamma_0 \vdash t : A \to T \mid \Delta_0$ y $\pi_u \rhd_{\mathcal{E}} \Gamma_1 \vdash u : A \mid \Delta_1$. Por *h.i.* existe $\pi_{u'} \rhd_{\mathcal{E}} \Gamma_1' \vdash u' : A \mid \Delta_1'$ con $\Gamma_1' \subseteq \Gamma_1$ y $\Delta_1' \subseteq \Delta_1$. Finalmente, se concluye por (APP), $\pi' \rhd_{\mathcal{E}} \Gamma' \vdash o' : T \mid \Delta'$ con $\Gamma' = \Gamma_0 \cup \Gamma_1' \subseteq \Gamma_0 \cup \Gamma_1 = \Gamma$ y $\Delta' = \Delta_0 \cup \Delta_1' \subseteq \Delta_0 \cup \Delta_1 = \Delta$.

- $\mathtt{O} = \lambda x.\mathtt{T}$. Luego, $o = \lambda x.t$ y $o' = \lambda x.t'$ con $t \to_{\Lambda M} t'$. Más aún, por (ABS) se tiene $T = A \to B$ con $\pi_t \rhd_{\mathcal{E}} \Gamma; (x : A)^{\leq 1} \vdash t : B \mid \Delta$. Por *h.i.* existe $\pi_{t'} \rhd_{\mathcal{E}} \Gamma'; (x : A)^{\leq 1} \vdash t' : B \mid \Delta'$ con $\Gamma'; (x : A)^{\leq 1} \subseteq \Gamma; (x : A)^{\leq 1}$ y $\Delta' \subseteq \Delta$. Finalmente, se concluye por (ABS), $\pi' \rhd_{\mathcal{E}} \Gamma' \vdash o' : T \mid \Delta'$ con $\Gamma' \subseteq \Gamma$ y $\Delta' \subseteq \Delta$.

- $\mathtt{O} = \mu\alpha.\mathtt{C}$. Luego, $o = \mu\alpha.c$ y $o' = \mu\alpha.c'$ con $c \to_{\Lambda M} c'$. Más aún, por (CONT) se tiene $\pi_c \rhd_{\mathcal{E}} \Gamma \vdash c \mid \Delta; (\alpha : T)^{\leq 1}$. Por *h.i.* existe una derivación $\pi_{c'} \rhd_{\mathcal{E}} \Gamma' \vdash c' \mid \Delta'; (\alpha : T)^{\leq 1}$ con $\Gamma' \subseteq \Gamma$ y $\Delta'; (\alpha : T)^{\leq 1} \subseteq \Delta; (\alpha : T)^{\leq 1}$. Finalmente, se concluye por (CONT), $\pi' \rhd_{\mathcal{E}} \Gamma' \vdash o' : T \mid \Delta'$ con $\Gamma' \subseteq \Gamma$ y $\Delta' \subseteq \Delta$.

- $\mathtt{O} = \mathtt{T}[x \setminus u]$. Luego, $o = t[x \setminus u]$ y $o' = t'[x \setminus u]$ con $t \to_{\Lambda M} t'$. Más aún, por (SUBS), $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta = \Delta_0 \cup \Delta_1$ con $\pi_t \rhd_{\mathcal{E}} \Gamma_0; (x : A)^{\leq 1} \vdash t : T \mid \Delta_0$ y $\pi_u \rhd_{\mathcal{E}} \Gamma_1 \vdash u : A \mid \Delta_1$. Por *h.i.* existe $\pi_{t'} \rhd_{\mathcal{E}} \Gamma_0'; (x : A)^{\leq 1} \vdash t' : T \mid \Delta_0'$ con $\Gamma_0'; (x : A)^{\leq 1} \subseteq \Gamma_0$ y $\Delta_0'; (x : A)^{\leq 1} \subseteq \Delta_0$. Finalmente, se concluye por (SUBS), $\pi' \rhd_{\mathcal{E}} \Gamma' \vdash o' : T \mid \Delta'$ con $\Gamma' = \Gamma_0' \cup \Gamma_1 \subseteq \Gamma_0 \cup \Gamma_1 = \Gamma$ y $\Delta' = \Delta_0' \cup \Delta_1 \subseteq \Delta_0 \cup \Delta_1 = \Delta$.

- $\mathtt{O} = t[x \setminus \mathtt{T}]$. Luego, $o = t[x \setminus u]$ y $o' = t[x \setminus u']$ con $u \to_{\Lambda M} u'$. Más aún, por (SUBS) se tienen $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta = \Delta_0 \cup \Delta_1$ con $\pi_t \rhd_{\mathcal{E}} \Gamma_0; (x : A)^{\leq 1} \vdash t : T \mid \Delta_0$ y $\pi_u \rhd_{\mathcal{E}} \Gamma_1 \vdash u : A \mid \Delta_1$. Por *h.i.* existe $\pi_{u'} \rhd_{\mathcal{E}} \Gamma_1' \vdash u' : A \mid \Delta_1'$ con $\Gamma_1' \subseteq \Gamma_1$ y $\Delta_1' \subseteq \Delta_1$. Finalmente, se concluye por (SUBS), $\pi' \rhd_{\mathcal{E}} \Gamma' \vdash o' : T \mid \Delta'$ con $\Gamma' = \Gamma_0 \cup \Gamma_1' \subseteq \Gamma_0 \cup \Gamma_1 = \Gamma$ y $\Delta' = \Delta_0 \cup \Delta_1' \subseteq \Delta_0 \cup \Delta_1 = \Delta$.

- $\mathtt{O} = [\alpha]\, \mathtt{T}$. Luego, $o = [\alpha]\, t$ y $o' = [\alpha]\, t'$ con $t \to_{\Lambda M} t'$. Más aún, por (NAME) se tiene $\Delta = \Delta_0; \alpha : A$ con $\pi_t \rhd_{\mathcal{E}} \Gamma \vdash t : A \mid \Delta_0; (\alpha : A)^{\leq 1}$. Por *h.i.* existe $\pi_{t'} \rhd_{\mathcal{E}} \Gamma' \vdash t' : A \mid \Delta_0'; (\alpha : A)^{\leq 1}$ con $\Gamma' \subseteq \Gamma$ y $\Delta_0'; (\alpha : A)^{\leq 1} \subseteq \Delta_0; (\alpha : A)^{\leq 1}$. Se concluye por (NAME), $\pi' \rhd_{\mathcal{E}} \Gamma' \vdash o' : T \mid \Delta_0'; \alpha : A$ con $\Gamma' \subseteq \Gamma$ y $\Delta' = \Delta_0'; \alpha : A \subseteq \Delta_0; \alpha : A = \Delta$.

- $\mathtt{O} = \mathtt{C}[\![\alpha \backslash^{\alpha'} s]\!]$. Luego, $o = c[\![\alpha \backslash^{\alpha'} s]\!]$ y $o' = c'[\![\alpha \backslash^{\alpha'} s]\!]$ con $c \to_{\Lambda M} c'$. Más aún, por (REPL) se tienen $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta = \Delta_0 \cup \Delta_1; \alpha' : A$ con $\pi_c \rhd_{\mathcal{E}} \Gamma_0 \vdash c \mid \Delta_0; (\alpha : S \to A)^{\leq 1}; (\alpha' : A)^{\leq 1}$

y $\pi_s \rhd_{\mathcal{E}} \Gamma_1 \vdash s : A \mid \Delta_1; (\alpha' : A)^{\leq 1}$. Por *h.i.* existe $\pi_{c'} \rhd_{\mathcal{E}} \Gamma_0' \vdash c' \mid \Delta_0'; (\alpha : S \to A)^{\leq 1}; (\alpha' : A)^{\leq 1}$ con $\Gamma_0' \subseteq \Gamma_0$ y $\Delta_0'; (\alpha : S \to A)^{\leq 1}; (\alpha' : A)^{\leq 1} \subseteq \Delta_0; (\alpha : S \to A)^{\leq 1}; (\alpha' : A)^{\leq 1}$. Finalmente, se concluye por (REPL), $\pi' \rhd_{\mathcal{E}} \Gamma' \vdash o' : T \mid \Delta'$ con $\Gamma' = \Gamma_0' \cup \Gamma_1 \subseteq \Gamma_0 \cup \Gamma_1 = \Gamma$ y $\Delta' = \Delta_0' \cup \Delta_1; \alpha' : A \subseteq \Delta_0 \cup \Delta_1; \alpha' : A = \Delta$.

- $\mathtt{O} = c[\![\alpha \setminus^{\alpha'} \mathtt{S}]\!]$. Luego, $o = c[\![\alpha \setminus^{\alpha'} s]\!]$ y $o' = c[\![\alpha \setminus^{\alpha'} s']\!]$ con $s \to_{\Lambda M} s'$. Más aún, por (REPL) se tienen $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta = \Delta_0 \cup \Delta_1; \alpha' : A$ con $\pi_c \rhd_{\mathcal{E}} \Gamma_0 \vdash c \mid \Delta_0; (\alpha : S \to A)^{\leq 1}; (\alpha' : A)^{\leq 1}$ y $\pi_s \rhd_{\mathcal{E}} \Gamma_1 \vdash s : A \mid \Delta_1; (\alpha' : A)^{\leq 1}$. Por *h.i.* existe una derivación $\pi_{s'} \rhd_{\mathcal{E}} \Gamma_1' \vdash s' : A \mid \Delta_1'; (\alpha' : A)^{\leq 1}$ con $\Gamma_1' \subseteq \Gamma_1$ y $\Delta_1'; (\alpha' : A)^{\leq 1} \subseteq \Delta_1; (\alpha' : A)^{\leq 1}$. Finalmente, se concluye por (REPL), $\pi' \rhd_{\mathcal{E}} \Gamma' \vdash o' : T \mid \Delta'$ con los contextos $\Gamma' = \Gamma_0 \cup \Gamma_1' \subseteq \Gamma_0 \cup \Gamma_1 = \Gamma$ y $\Delta' = \Delta_0 \cup \Delta_1'; \alpha' : A \subseteq \Delta_0 \cup \Delta_1; \alpha' : A = \Delta$.

- $\mathtt{O} = \mathtt{C}[\alpha \setminus \beta]$. Luego, $o = c[\alpha \setminus \beta]$ y $o' = c'[\alpha \setminus \beta]$ con $c \to_{\Lambda M} c'$. Más aún, por (REN) se tiene $\Delta = \Delta_0; \beta : A$ con $\pi_c \rhd_{\mathcal{E}} \Gamma \vdash c \mid \Delta_0; (\alpha : A)^{\leq 1}; (\beta : A)^{\leq 1}$. Por *h.i.* existe una derivación $\pi_{c'} \rhd_{\mathcal{E}} \Gamma' \vdash c' \mid \Delta_0'; (\alpha : A)^{\leq 1}; (\beta : A)^{\leq 1}$ con $\Gamma' \subseteq \Gamma$ y $\Delta_0'; (\alpha : A)^{\leq 1}; (\beta : A)^{\leq 1} \subseteq \Delta_0; (\alpha : A)^{\leq 1}; (\beta : A)^{\leq 1}$. Finalmente, se concluye por (REN), $\pi' \rhd_{\mathcal{E}} \Gamma' \vdash o' : T \mid \Delta'$ con $\Gamma' \subseteq \Gamma$ y $\Delta' = \Delta_0'; \beta : A \subseteq \Delta_0; \beta : A = \Delta$.

- $\mathtt{O} = \mathtt{T} \cdot s$. Luego, $o = t \cdot s$ y $o' = t' \cdot s$ con $t \to_{\Lambda M} t'$. Más aún, por (STK) se tienen $T = A \cdot S$, $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta = \Delta_0 \cup \Delta_1$ con $\pi_t \rhd_{\mathcal{E}} \Gamma_0 \vdash t : A \mid \Delta_0$ y $\pi_s \rhd_{\mathcal{E}} \Gamma_1 \vdash s : S \mid \Delta_1$. Por *h.i.* existe $\pi_{t'} \rhd_{\mathcal{E}} \Gamma_0' \vdash t' : A \mid \Delta_0'$ con $\Gamma_0' \subseteq \Gamma_0$ y $\Delta_0' \subseteq \Delta_0$. Finalmente, se concluye por (STK), $\pi' \rhd_{\mathcal{E}} \Gamma' \vdash o' : T \mid \Delta'$ con $\Gamma' = \Gamma_0' \cup \Gamma_1 \subseteq \Gamma_0 \cup \Gamma_1 = \Gamma$ y $\Delta' = \Delta_0' \cup \Delta_1 \subseteq \Delta_0 \cup \Delta_1 = \Delta$.

- $\mathtt{O} = t \cdot \mathtt{S}$. Luego, $o = t \cdot s$ y $o' = t \cdot s'$ con $s \to_{\Lambda M} s'$. Más aún, por (STK) se tienen $T = A \cdot S$, $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta = \Delta_0 \cup \Delta_1$ con $\pi_t \rhd_{\mathcal{E}} \Gamma_0 \vdash t : A \mid \Delta_0$ y $\pi_s \rhd_{\mathcal{E}} \Gamma_1 \vdash s : S \mid \Delta_1$. Por *h.i.* existe $\pi_{s'} \rhd_{\mathcal{E}} \Gamma_1' \vdash s' : S \mid \Delta_1'$ con $\Gamma_1' \subseteq \Gamma_1$ y $\Delta_1' \subseteq \Delta_1$. Finalmente, se concluye por (STK), $\pi' \rhd_{\mathcal{E}} \Gamma' \vdash o' : T \mid \Delta'$ con $\Gamma' = \Gamma_0 \cup \Gamma_1' \subseteq \Gamma_0 \cup \Gamma_1 = \Gamma$ y $\Delta' = \Delta_0 \cup \Delta_1' \subseteq \Delta_0 \cup \Delta_1 = \Delta$.

□

Lema C.0.23 (Renaming)**.** *Sea* $\pi_o \rhd_{\mathcal{E}} \Gamma \vdash o : T \mid \Delta; (\alpha : B)^{\leq 1}$. *Luego, existe una derivación* $\pi_{\{\alpha \setminus \beta\} o} \rhd_{\mathcal{E}} \Gamma \vdash \{\alpha \setminus \beta\} o : T \mid \Delta \cup (\beta : B)^{\leq 1}$.

Demostración. Por inducción en o.

- $o = x$. Luego, $\{\alpha \setminus \beta\} o = o$ y el resultado es inmediato con $(\alpha : B)^{\leq 1}$ y $(\beta : B)^{\leq 1}$ ambos vacíos.

- $o = t\,u$. Luego, $\{\alpha \setminus \beta\} o = \{\alpha \setminus \beta\} t\, \{\alpha \setminus \beta\} u$. Más aún, por (APP) se tienen $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta = \Delta_0 \cup \Delta_1$ con $\pi_t \rhd_{\mathcal{E}} \Gamma_0 \vdash t : A \to T \mid \Delta_0; (\alpha : B)^{\leq 1}$ y $\pi_u \rhd_{\mathcal{E}} \Gamma_1 \vdash u : A \mid \Delta_1; (\alpha : B)^{\leq 1}$. Por *h.i.* con π_t y π_u, existen dos derivaciones $\pi_{\{\alpha \setminus \beta\} t} \rhd_{\mathcal{E}} \Gamma_0 \vdash \{\alpha \setminus \beta\} t : A \to T \mid \Delta_0 \cup (\beta : B)^{\leq 1}$ y $\pi_{\{\alpha \setminus \beta\} u} \rhd_{\mathcal{E}} \Gamma_1 \vdash \{\alpha \setminus \beta\} u : A \mid \Delta_1 \cup (\beta : B)^{\leq 1}$. Finalmente, se concluye por (APP), $\pi_{\{\alpha \setminus \beta\} o} \rhd_{\mathcal{E}} \Gamma_0 \cup \Gamma_1 \vdash \{\alpha \setminus \beta\} o : T \mid \Delta_0 \cup \Delta_1 \cup (\beta : B)^{\leq 1}$.

- $o = \lambda x.t$. Luego, $\{\alpha \setminus \beta\} o = \lambda x. \{\alpha \setminus \beta\} t$. Más aún, por (ABS) se tiene $T = A \to A'$ con la derivación $\pi_t \rhd_{\mathcal{E}} \Gamma; (x : A)^{\leq 1} \vdash t : A' \mid \Delta; (\alpha : B)^{\leq 1}$. Por *h.i.* existe una derivación $\pi_{\{\alpha \setminus \beta\} t} \rhd_{\mathcal{E}} \Gamma; (x : A)^{\leq 1} \vdash \{\alpha \setminus \beta\} t : A' \mid \Delta \cup (\beta : B)^{\leq 1}$. Finalmente, se concluye aplicando (ABS), $\pi_{\{\alpha \setminus \beta\} o} \rhd_{\mathcal{E}} \Gamma \vdash \{\alpha \setminus \beta\} o : T \mid \Delta \cup (\beta : B)^{\leq 1}$.

- $o = \mu\gamma.c$ con $\gamma \neq \alpha$ y $\gamma \neq \beta$. Luego, $\{\alpha \setminus \beta\} o = \mu\gamma. \{\alpha \setminus \beta\} c$. Más aún, por (CONT) se tiene $\pi_c \rhd_{\mathcal{E}} \Gamma \vdash c \mid \Delta; (\gamma : T)^{\leq 1}; (\alpha : B)^{\leq 1}$. Por *h.i.* existe $\pi_{\{\alpha \setminus \beta\} c} \rhd_{\mathcal{E}} \Gamma \vdash c \mid (\Delta; (\gamma : T)^{\leq 1}) \cup (\beta : B)^{\leq 1}$. Finalmente, se concluye por (CONT), $\pi_{\{\alpha \setminus \beta\} o} \rhd_{\mathcal{E}} \Gamma \vdash \{\alpha \setminus \beta\} o : T \mid \Delta \cup (\beta : B)^{\leq 1}$.

- $o = t[x \setminus u]$. Luego, $\{\alpha \setminus \beta\}o = \{\alpha \setminus \beta\}t[x \setminus \{\alpha \setminus \beta\}u]$. Más aún, por (SUBS) se tienen $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta = \Delta_0 \cup \Delta_1$ con $\pi_t \rhd_{\mathcal{E}} \Gamma_0; (x : A)^{\leq 1} \vdash t : T \mid \Delta_0; (\alpha : B)^{\leq 1}$ y $\pi_u \rhd_{\mathcal{E}} \Gamma_1 \vdash u : A \mid \Delta_1; (\alpha : B)^{\leq 1}$. Por *h.i.* existen derivaciones $\pi_{\{\alpha\setminus\beta\}t} \rhd_{\mathcal{E}} \Gamma_0; (x : A)^{\leq 1} \vdash \{\alpha \setminus \beta\}t : T \mid \Delta_0 \cup (\beta : B)^{\leq 1}$ y $\pi_{\{\alpha\setminus\beta\}u} \rhd_{\mathcal{E}} \Gamma_1 \vdash \{\alpha \setminus \beta\}u : A \mid \Delta_1 \cup (\beta : B)^{\leq 1}$. Finalmente, se concluye con (SUBS) entre $\pi_{\{\alpha\setminus\beta\}t}$ y $\pi_{\{\alpha\setminus\beta\}u}$, $\pi_{\{\alpha\setminus\beta\}o} \rhd_{\mathcal{E}} \Gamma_0 \cup \Gamma_1 \vdash \{\alpha \setminus \beta\}o : T \mid \Delta_0 \cup \Delta_1 \cup (\beta : B)^{\leq 1}$.

- $o = [\alpha]\, t$. Luego, $\{\alpha \setminus \beta\}o = [\beta]\, \{\alpha \setminus \beta\}t$. Más aún, por (NAME), $(\alpha : B)^{\leq 1}$ es no vacío en π_o y $\pi_t \rhd_{\mathcal{E}} \Gamma \vdash t : B \mid \Delta; (\alpha : B)^{\leq 1}$. Por *h.i.* existe $\pi_{\{\alpha\setminus\beta\}t} \rhd_{\mathcal{E}} \Gamma \vdash \{\alpha \setminus \beta\}t : B \mid \Delta \cup (\beta : B)^{\leq 1}$. Se concluye por (NAME), con $\pi_{\{\alpha\setminus\beta\}o} \rhd_{\mathcal{E}} \Gamma \vdash \{\alpha \setminus \beta\}o : T \mid \Delta \cup \beta : B$.

- $o = [\gamma]\, t$ con $\gamma \neq \alpha$. Luego, $\{\alpha \setminus \beta\}o = [\gamma]\, \{\alpha \setminus \beta\}t$. Más aún, por (NAME), se tiene $\Delta = \Delta_0; \gamma : A$ con la derivación $\pi_t \rhd_{\mathcal{E}} \Gamma \vdash t : A \mid \Delta_0; (\gamma : A)^{\leq 1}; (\alpha : B)^{\leq 1}$. Por *h.i.* existe una derivación $\pi_{\{\alpha\setminus\beta\}t} \rhd_{\mathcal{E}} \Gamma \vdash \{\alpha \setminus \beta\}t : A \mid (\Delta_0; (\gamma : A)^{\leq 1}) \cup (\beta : B)^{\leq 1}$. Finalmente, se concluye por (NAME), $\pi_{\{\alpha\setminus\beta\}o} \rhd_{\mathcal{E}} \Gamma \vdash \{\alpha \setminus \beta\}o : T \mid \Delta \cup (\beta : B)^{\leq 1}$.

- $o = c[\![\gamma \setminus^{\alpha} s]\!]$ con $\alpha \neq \gamma$ y $\beta \neq \gamma$. Luego, $\{\alpha \setminus \beta\}o = \{\alpha \setminus \beta\}c[\![\gamma \setminus^{\beta} \{\alpha \setminus \beta\}s]\!]$. Más aún, por (REPL), se tiene $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta = \Delta_0 \cup \Delta_1$ y $(\alpha : B)^{\leq 1}$ no vacío en π_o con las derivaciones $\pi_c \rhd_{\mathcal{E}} \Gamma_0 \vdash c \mid \Delta_0; (\gamma : S \to B)^{\leq 1}; (\alpha : B)^{\leq 1}$ y $\pi_s \rhd_{\mathcal{E}} \Gamma_1 \vdash s : S \mid \Delta_1; (\alpha : B)^{\leq 1}$. Entoncens, por *h.i.* con π_c y π_s, existen dos derivaciones $\pi_{\{\alpha\setminus\beta\}c} \rhd_{\mathcal{E}} \Gamma_0 \vdash \{\alpha \setminus \beta\}c \mid (\Delta_0; (\gamma : S \to B)^{\leq 1}) \cup (\beta : B)^{\leq 1}$ y $\pi_{\{\alpha\setminus\beta\}s} \rhd_{\mathcal{E}} \Gamma_1 \vdash \{\alpha \setminus \beta\}s : S \mid \Delta_1 \cup (\beta : B)^{\leq 1}$. Finalmente, se concluye por (REPL), $\pi_{\{\alpha\setminus\beta\}o} \rhd_{\mathcal{E}} \Gamma \vdash \{\alpha \setminus \beta\}o : T \mid \Delta \cup \beta : B$.

- $o = c[\![\gamma \setminus^{\gamma'} s]\!]$ con $\alpha \neq \gamma$, $\alpha \neq \gamma'$, $\beta \neq \gamma$. Luego, $\{\alpha \setminus \beta\}o = \{\alpha \setminus \beta\}c[\![\gamma \setminus^{\gamma'} \{\alpha \setminus \beta\}s]\!]$. Más aún, por (REPL), se tienen contextos $\Gamma = \Gamma_0 \cup \Gamma_1$y $\Delta = \Delta_0 \cup \Delta_1; \gamma' : A$ con derivaciones $\pi_c \rhd_{\mathcal{E}} \Gamma_0 \vdash c \mid \Delta_0; (\gamma : S \to A)^{\leq 1}; (\gamma' : A)^{\leq 1}; (\alpha : B)^{\leq 1}$ y $\pi_s \rhd_{\mathcal{E}} \Gamma_1 \vdash s : S \mid \Delta_1; (\gamma' : A)^{\leq 1}; (\alpha : B)^{\leq 1}$. Por *h.i.* existen $\pi_{\{\alpha\setminus\beta\}c} \rhd_{\mathcal{E}} \Gamma_0 \vdash \{\alpha \setminus \beta\}c \mid (\Delta_0; (\gamma : S \to A)^{\leq 1}; (\gamma' : A)^{\leq 1}) \cup (\beta : B)^{\leq 1}$ y $\pi_{\{\alpha\setminus\beta\}s} \rhd_{\mathcal{E}} \Gamma_1 \vdash \{\alpha \setminus \beta\}s : S \mid (\Delta_1; (\gamma' : A)^{\leq 1}) \cup (\beta : B)^{\leq 1}$. Finalmente, se concluye por (REPL), $\pi_{\{\alpha\setminus\beta\}o} \rhd_{\mathcal{E}} \Gamma \vdash \{\alpha \setminus \beta\}o : T \mid \Delta \cup (\beta : B)^{\leq 1}$.

- $o = c[\gamma \setminus \alpha]$ con $\alpha \neq \gamma$ y $\beta \neq \gamma$. Luego, $\{\alpha \setminus \beta\}o = \{\alpha \setminus \beta\}c[\gamma \setminus \beta]$. Más aún, por (REN), $(\alpha : B)^{\leq 1}$ es no vacío en π_o y $\pi_c \rhd_{\mathcal{E}} \Gamma \vdash c \mid \Delta; (\gamma : B)^{\leq 1}; (\alpha : B)^{\leq 1}$. Por *h.i.* existe $\pi_{\{\alpha\setminus\beta\}c} \rhd_{\mathcal{E}} \Gamma \vdash \{\alpha \setminus \beta\}c \mid (\Delta; (\gamma : B)^{\leq 1}) \cup (\beta : B)^{\leq 1}$. Finalmente, se concluye aplicando (REN), $\pi_{\{\alpha\setminus\beta\}o} \rhd_{\mathcal{E}} \Gamma \vdash \{\alpha \setminus \beta\}o : T \mid \Delta \cup \beta : B$.

- $o = c[\gamma \setminus \gamma']$ con $\alpha \neq \gamma$, $\alpha \neq \gamma'$, $\beta \neq \gamma$. Luego, $\{\alpha \setminus \beta\}o = \{\alpha \setminus \beta\}c[\![\gamma \setminus^{\gamma'} \{\alpha \setminus \beta\}s]\!]$. Más aún, por (REN), se tiene $\Delta = \Delta_0; \gamma' : A$ con una derivación $\pi_c \rhd_{\mathcal{E}} \Gamma \vdash c \mid \Delta_0; (\gamma : A)^{\leq 1}; (\gamma' : A)^{\leq 1}; (\alpha : B)^{\leq 1}$. Por *h.i.* existe otra derivación $\pi_{\{\alpha\setminus\beta\}c} \rhd_{\mathcal{E}} \Gamma \vdash \{\alpha \setminus \beta\}c \mid (\Delta_0; (\gamma : A)^{\leq 1}; (\gamma' : A)^{\leq 1}) \cup (\beta : B)^{\leq 1}$. Finalmente, se concluye por (REN), $\pi_{\{\alpha\setminus\beta\}o} \rhd_{\mathcal{E}} \Gamma \vdash \{\alpha \setminus \beta\}o : T \mid \Delta \cup (\beta : B)^{\leq 1}$.

- $o = t \cdot s$. Luego, $\{\alpha \setminus \beta\}o = \{\alpha \setminus \beta\}t \cdot \{\alpha \setminus \beta\}s$. Más aún, por (STK) se tienen $T = A \cdot S$, $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta = \Delta_0 \cup \Delta_1$ con derivaciones $\pi_t \rhd_{\mathcal{E}} \Gamma_0 \vdash t : A \mid \Delta_0; (\alpha : B)^{\leq 1}$ y $\pi_s \rhd_{\mathcal{E}} \Gamma_1 \vdash s : S \mid \Delta_1; (\alpha : B)^{\leq 1}$. Entonces, por *h.i.* con π_t y π_s, existen derivaciones $\pi_{\{\alpha\setminus\beta\}t} \rhd_{\mathcal{E}} \Gamma_0 \vdash \{\alpha \setminus \beta\}t : A \mid \Delta_0 \cup (\beta : B)^{\leq 1}$ y $\pi_{\{\alpha\setminus\beta\}s} \rhd_{\mathcal{E}} \Gamma_1 \vdash \{\alpha \setminus \beta\}s : S \mid \Delta_1 \cup (\beta : B)^{\leq 1}$. Finalmente, se concluye por (STK), $\pi_{\{\alpha\setminus\beta\}o} \rhd_{\mathcal{E}} \Gamma_0 \cup \Gamma_1 \vdash \{\alpha \setminus \beta\}o : T \mid \Delta_0 \cup \Delta_1 \cup (\beta : B)^{\leq 1}$.

□

Lema C.0.24 (Anti-Renaming)**.** *Sea* $\pi_{\{\alpha\setminus\beta\}o} \rhd_{\mathcal{E}} \Gamma \vdash \{\alpha \setminus \beta\}o : T \mid \Delta \cup (\beta : B)^{\leq 1}$. *Luego, existe* $\pi_o \rhd_{\mathcal{E}} \Gamma \vdash o : T \mid \Delta; (\alpha : B)^{\leq 1}$.

Demostración. Por inducción en o.

- $o = x$. Luego, $\{\alpha \setminus \beta\}o = o$ y el resultado es inmediato con $(\alpha : B)^{\leq 1}$ y $(\beta : B)^{\leq 1}$ ambos vacíos.

- $o = t\,u$. Luego, $\{\alpha \setminus \beta\}o = \{\alpha \setminus \beta\}t\,\{\alpha \setminus \beta\}u$. Más aún, por (APP) se tienen contextos $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta = \Delta_0 \cup \Delta_1$ con derivaciones $\pi_{\{\alpha\setminus\beta\}t} \rhd_{\mathcal{E}} \Gamma_0 \vdash \{\alpha \setminus \beta\}t : A \to T \mid \Delta_0 \cup (\beta : B)^{\leq 1}$ y $\pi_{\{\alpha\setminus\beta\}u} \rhd_{\mathcal{E}} \Gamma_1 \vdash \{\alpha \setminus \beta\}u : A \mid \Delta_1 \cup (\beta : B)^{\leq 1}$. Entonces, por *h.i.* existen derivaciones $\pi_t \rhd_{\mathcal{E}} \Gamma_0 \vdash t : A \to T \mid \Delta_0; (\alpha : B)^{\leq 1}$ y $\pi_u \rhd_{\mathcal{E}} \Gamma_1 \vdash u : A \mid \Delta_1; (\alpha : B)^{\leq 1}$. Finalmente, se concluye por (APP), $\pi_o \rhd_{\mathcal{E}} \Gamma_0 \cup \Gamma_1 \vdash o : T \mid \Delta_0 \cup \Delta_1; (\alpha : B)^{\leq 1}$.

- $o = \lambda x.t$. Luego, $\{\alpha \setminus \beta\}o = \lambda x.\{\alpha \setminus \beta\}t$. Más aún, por (ABS) se tiene $T = A \to A'$ con $\pi_{\{\alpha\setminus\beta\}t} \rhd_{\mathcal{E}} \Gamma; (x : A)^{\leq 1} \vdash \{\alpha \setminus \beta\}t : A' \mid \Delta \cup (\beta : B)^{\leq 1}$. Por *h.i.* existe $\pi_t \rhd_{\mathcal{E}} \Gamma; (x : A)^{\leq 1} \vdash t : A' \mid \Delta; (\alpha : B)^{\leq 1}$. Se concluye por (ABS), $\pi_o \rhd_{\mathcal{E}} \Gamma \vdash o : T \mid \Delta; (\alpha : B)^{\leq 1}$.

- $o = \mu\gamma.c$ con $\gamma \neq \alpha$ y $\gamma \neq \beta$. Luego, $\{\alpha \setminus \beta\}o = \mu\gamma.\{\alpha \setminus \beta\}c$. Más aún, por (CONT) se tiene $\pi_{\{\alpha\setminus\beta\}c} \rhd_{\mathcal{E}} \Gamma \vdash c \mid (\Delta; (\gamma : T)^{\leq 1}) \cup (\beta : B)^{\leq 1}$. Por *h.i.* existe $\pi_c \rhd_{\mathcal{E}} \Gamma \vdash c \mid \Delta; (\gamma : T)^{\leq 1}; (\alpha : B)^{\leq 1}$. Se concluye por (CONT), $\pi_o \rhd_{\mathcal{E}} \Gamma \vdash o : T \mid \Delta; (\alpha : B)^{\leq 1}$.

- $o = t[x \setminus u]$. Luego, $\{\alpha \setminus \beta\}o = \{\alpha \setminus \beta\}t[x \setminus \{\alpha \setminus \beta\}u]$. Más aún, por (SUBS) se tienen contextos $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta = \Delta_0 \cup \Delta_1$ con derivaciones $\pi_{\{\alpha\setminus\beta\}t} \rhd_{\mathcal{E}} \Gamma_0; (x : A)^{\leq 1} \vdash \{\alpha \setminus \beta\}t : T \mid \Delta_0 \cup (\beta : B)^{\leq 1}$ y $\pi_{\{\alpha\setminus\beta\}u} \rhd_{\mathcal{E}} \Gamma_1 \vdash \{\alpha \setminus \beta\}u : A \mid \Delta_1 \cup (\beta : B)^{\leq 1}$. Entonces, por *h.i.* existen derivaciones $\pi_t \rhd_{\mathcal{E}} \Gamma_0; (x : A)^{\leq 1} \vdash t : T \mid \Delta_0; (\alpha : B)^{\leq 1}$ y $\pi_u \rhd_{\mathcal{E}} \Gamma_1 \vdash u : A \mid \Delta_1; (\alpha : B)^{\leq 1}$. Finalmente, se concluye por (SUBS), $\pi_o \rhd_{\mathcal{E}} \Gamma_0 \cup \Gamma_1 \vdash o : T \mid \Delta_0 \cup \Delta_1; (\alpha : B)^{\leq 1}$.

- $o = [\alpha]\,t$. Luego, $\{\alpha \setminus \beta\}o = [\beta]\,\{\alpha \setminus \beta\}t$. Más aún, por (NAME), se tiene $(\beta : B)^{\leq 1}$ no vacío en $\pi_{\{\alpha\setminus\beta\}o}$ y la derivación $\pi_{\{\alpha\setminus\beta\}t} \rhd_{\mathcal{E}} \Gamma \vdash \{\alpha \setminus \beta\}t : B \mid \Delta \cup (\beta : B)^{\leq 1}$. Entonces, por *h.i.* existe una derivación $\pi_t \rhd_{\mathcal{E}} \Gamma \vdash t : B \mid \Delta; (\alpha : B)^{\leq 1}$. Finalmente, se concluye por (NAME), con $\pi_o \rhd_{\mathcal{E}} \Gamma \vdash o : T \mid \Delta; \alpha : B$.

- $o = [\gamma]\,t$ con $\gamma \neq \alpha$. Luego, $\{\alpha \setminus \beta\}o = [\gamma]\,\{\alpha \setminus \beta\}t$. Más aún, por (NAME), se tiene $\Delta = \Delta_0; \gamma : A$ con la derivación $\pi_{\{\alpha\setminus\beta\}t} \rhd_{\mathcal{E}} \Gamma \vdash \{\alpha \setminus \beta\}t : A \mid (\Delta_0; (\gamma : A)^{\leq 1}) \cup (\beta : B)^{\leq 1}$. Entonces, por *h.i.* existe una derivación $\pi_t \rhd_{\mathcal{E}} \Gamma \vdash t : A \mid \Delta_0; (\gamma : A)^{\leq 1}; (\alpha : B)^{\leq 1}$. Finalmente, se concluye por (NAME), $\pi_o \rhd_{\mathcal{E}} \Gamma \vdash o : T \mid \Delta; (\alpha : B)^{\leq 1}$.

- $o = c[\![\gamma \setminus^{\alpha} s]\!]$ con $\alpha \neq \gamma$ y $\beta \neq \gamma$. Luego, $\{\alpha \setminus \beta\}o = \{\alpha \setminus \beta\}c[\![\gamma \setminus^{\beta} \{\alpha \setminus \beta\}s]\!]$. Más aún, por (REPL), se tienen contextos $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta = \Delta_0 \cup \Delta_1$ con $(\beta : B)^{\leq 1}$ no vacío en $\pi_{\pi_{\{\alpha\setminus\beta\}o}}$ y derivaciones $\pi_{\{\alpha\setminus\beta\}c} \rhd_{\mathcal{E}} \Gamma_0 \vdash \{\alpha \setminus \beta\}c \mid (\Delta_0; (\gamma : S \to B)^{\leq 1}) \cup (\beta : B)^{\leq 1}$ y $\pi_{\{\alpha\setminus\beta\}s} \rhd_{\mathcal{E}} \Gamma_1 \vdash \{\alpha \setminus \beta\}s : S \mid \Delta_1 \cup (\beta : B)^{\leq 1}$. Por *h.i.* existen $\pi_c \rhd_{\mathcal{E}} \Gamma_0 \vdash c \mid \Delta_0; (\gamma : S \to B)^{\leq 1}; (\alpha : B)^{\leq 1}$ y $\pi_s \rhd_{\mathcal{E}} \Gamma_1 \vdash s : S \mid \Delta_1; (\alpha : B)^{\leq 1}$. Finalmente, se concluye por (REPL), $\pi_o \rhd_{\mathcal{E}} \Gamma \vdash o : T \mid \Delta; \alpha : B$.

- $o = c[\![\gamma \setminus^{\gamma'} s]\!]$ con $\alpha \neq \gamma$, $\alpha \neq \gamma'$, $\beta \neq \gamma$. Luego, $\{\alpha \setminus \beta\}o = \{\alpha \setminus \beta\}c[\![\gamma \setminus^{\gamma'} \{\alpha \setminus \beta\}s]\!]$. Más aún, por (REPL), por las hipótesis se tienen contextos $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta = \Delta_0 \cup \Delta_1; \gamma' : A$ con derivaciones $\pi_{\{\alpha\setminus\beta\}c} \rhd_{\mathcal{E}} \Gamma_0 \vdash \{\alpha \setminus \beta\}c \mid (\Delta_0; (\gamma : S \to A)^{\leq 1}; (\gamma' : A)^{\leq 1}) \cup (\beta : B)^{\leq 1}$ y $\pi_{\{\alpha\setminus\beta\}s} \rhd_{\mathcal{E}} \Gamma_1 \vdash \{\alpha \setminus \beta\}s : S \mid (\Delta_1; (\gamma' : A)^{\leq 1}) \cup (\beta : B)^{\leq 1}$. Entonces, por *h.i.* existen dos derivaciones $\pi_c \rhd_{\mathcal{E}} \Gamma_0 \vdash c \mid \Delta_0; (\gamma : S \to A)^{\leq 1}; (\gamma' : A)^{\leq 1}; (\alpha : B)^{\leq 1}$ y $\pi_s \rhd_{\mathcal{E}} \Gamma_1 \vdash s : S \mid \Delta_1; (\gamma' : A)^{\leq 1}; (\alpha : B)^{\leq 1}$. Finalmente, se concluye por (REPL), $\pi_o \rhd_{\mathcal{E}} \Gamma \vdash o : T \mid \Delta; (\alpha : B)^{\leq 1}$.

- $o = c[\gamma \setminus \alpha]$ con $\alpha \neq \gamma$ y $\beta \neq \gamma$. Luego, $\{\alpha \setminus \beta\}o = \{\alpha \setminus \beta\}c[\gamma \setminus \beta]$. Más aún, por (REN), $(\beta : B)^{\leq 1}$ es no vacío en π_o y $\pi_{\{\alpha\setminus\beta\}c} \rhd_{\mathcal{E}} \Gamma \vdash \{\alpha \setminus \beta\}c \mid (\Delta; (\gamma : B)^{\leq 1}) \cup (\beta : B)^{\leq 1}$. Por *h.i.* existe $\pi_c \rhd_{\mathcal{E}} \Gamma \vdash c \mid \Delta; (\gamma : B)^{\leq 1}; (\alpha : B)^{\leq 1}$. Finalmente, se concluye por (REN), $\pi_o \rhd_{\mathcal{E}} \Gamma \vdash o : T \mid \Delta; \alpha : B$.

- $o = c[\gamma \setminus \gamma']$ con $\alpha \neq \gamma$, $\alpha \neq \gamma'$, $\beta \neq \gamma$. Luego, $\{\alpha \setminus \beta\}o = \{\alpha \setminus \beta\}c[\![\gamma \setminus^{\gamma'} \{\alpha \setminus \beta\}s]\!]$. Más aún, por (REN), por las hipótesis se tiene el contexto $\Delta = \Delta_0; \gamma' : A$ con la derivación $\pi_{\{\alpha\setminus\beta\}c} \rhd_{\mathcal{E}} \Gamma \vdash \{\alpha \setminus \beta\}c \mid (\Delta_0; (\gamma : A)^{\leq 1}; (\gamma' : A)^{\leq 1}) \cup (\beta : B)^{\leq 1}$. Entonces, por *h.i.* existe la derivación $\pi_c \rhd_{\mathcal{E}} \Gamma \vdash c \mid \Delta_0; (\gamma : A)^{\leq 1}; (\gamma' : A)^{\leq 1}; (\alpha : B)^{\leq 1}$. Se concluye pos (REN), $\pi_o \rhd_{\mathcal{E}} \Gamma \vdash o : T \mid \Delta; (\alpha : B)^{\leq 1}$.

- $o = t \cdot s$. Luego, $\{\alpha \setminus \beta\}o = \{\alpha \setminus \beta\}t \cdot \{\alpha \setminus \beta\}s$. Más aún, por (STK) se tienen $T = A \cdot S$, $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta = \Delta_0 \cup \Delta_1$ con derivaciones $\pi_{\{\alpha\setminus\beta\}t} \rhd_{\mathcal{E}} \Gamma_0 \vdash \{\alpha \setminus \beta\}t : A \mid \Delta_0 \cup (\beta : B)^{\leq 1}$ y $\pi_{\{\alpha\setminus\beta\}s} \rhd_{\mathcal{E}} \Gamma_1 \vdash \{\alpha \setminus \beta\}s : S \mid \Delta_1 \cup (\beta : B)^{\leq 1}$. Por *h.i.* existen $\pi_t \rhd_{\mathcal{E}} \Gamma_0 \vdash t : A \mid \Delta_0; (\alpha : B)^{\leq 1}$ y $\pi_s \rhd_{\mathcal{E}} \Gamma_1 \vdash s : S \mid \Delta_1; (\alpha : B)^{\leq 1}$. Se concluye por (STK), $\pi_o \rhd_{\mathcal{E}} \Gamma_0 \cup \Gamma_1 \vdash o : T \mid \Delta_0 \cup \Delta_1; (\alpha : B)^{\leq 1}$.

□

Lema 4.3.3. *Sea $o \in \mathbb{O}_{\lambda\mu}$. Si $\pi \rhd_{\mathcal{E}} \Gamma \vdash o : T \mid \Delta$ y $o \simeq_\sigma o'$, entonces existe $\pi' \rhd_{\mathcal{E}} \Gamma \vdash o' : T \mid \Delta'$.*

Demostración. La prueba es por inducción en $o \simeq_\sigma o'$. Por definición esto implica verificar cuatro condiciones: reflexividad, transitividad, simetría y congruencia (*i.e.* clausura por contextos):

Reflexividad Luego, $o' = o$ por lo que el resultado es inmediato.

Transitividad Luego, existe $p \in \mathbb{O}_{\lambda\mu}$ tal que $o \simeq_\sigma p$ y $p \simeq_\sigma o'$. Por *h.i.* con $o \simeq_\sigma p$ existe $\pi_p \rhd_{\mathcal{E}} \Gamma \vdash p : T \mid \Delta^*$. Más aún, por *h.i.* nuevamente, ahora con $p \simeq_\sigma o'$, existe $\pi' \rhd_{\mathcal{E}} \Gamma \vdash o' : T \mid \Delta'$ por lo que se concluye.

Simetría/Congruencia La simetría y congruencia se demuestran simultáneamente por inducción en el contexto de clausura $\mathtt{O}$ tal que $o = \mathtt{O}\langle p \rangle$ y $o' = \mathtt{O}\langle p' \rangle$ con $p \simeq_{\sigma_*} p'$, donde $\simeq_{\sigma_*}$ es una regla de la Fig. 4.10.

- $\mathtt{O} = \Box$. Luego, $o = p \simeq_{\sigma_*} p' = o'$. Más aún, los objetos son necesariamente términos. Se analiza la regla aplicada:
  - $\simeq_{\sigma_1}$. Luego, $o = (\lambda x.\lambda y.t)\, u$ y $o' = \lambda y.(\lambda x.t)\, u$ con $y \notin u$. Más aún, de π se tienen $T = A' \to B$, $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta = \Delta_0 \cup \Delta_1$ con $\pi_t \rhd_{\mathcal{E}} \Gamma_0; (x : A)^{\leq 1}; (y : A')^{\leq 1} \vdash t : B \mid \Delta_0$ y $\pi_u \rhd_{\mathcal{E}} \Gamma_1 \vdash u : A \mid \Delta_1$. Por Lem. 4.3.1, se tiene $y \notin \mathsf{dom}(\Gamma_1)$. Luego, por (ABS) con π_t se deriva $\rhd_{\mathcal{E}} \Gamma_0; (y : A')^{\leq 1} \vdash \lambda x.t : A \to B \mid \Delta_0$. Entonces, por (APP) con π_u, $\rhd_{\mathcal{E}} \Gamma; (y : A')^{\leq 1} \vdash (\lambda x.t)\, u : B \mid \Delta$. Por último, se concluye por (ABS) nuevamente, $\pi' \rhd_{\mathcal{E}} \Gamma \vdash o' : T \mid \Delta$. El caso simétrico es análogo.
  - $\simeq_{\sigma_2}$. Luego, $o = (\lambda x.t\, v)\, u$ y $o' = (\lambda x.t)\, u\, v$ con $x \notin v$. Más aún, de π se tienen $\Gamma = \Gamma_0 \cup \Gamma_1 \cup \Gamma_2$ y $\Delta = \Delta_0 \cup \Delta_1 \cup \Delta_2$ con derivaciones $\pi_t \rhd_{\mathcal{E}} \Gamma_0; (x : A)^{\leq 1} \vdash t : B \to T \mid \Delta_0$, $\pi_u \rhd_{\mathcal{E}} \Gamma_1 \vdash u : A \mid \Delta_1$ y $\pi_v \rhd_{\mathcal{E}} \Gamma_2 \vdash v : B \mid \Delta_2$. Por Lem. 4.3.1, se tiene $x \notin \mathsf{dom}(\Gamma_2)$. Luego, por (ABS) con π_t se deriva $\rhd_{\mathcal{E}} \Gamma_0 \vdash \lambda x.t : A \to B \to T \mid \Delta_0$. Por (APP) con π_u, $\rhd_{\mathcal{E}} \Gamma_0 \cup \Gamma_1 \vdash (\lambda x.t)\, u : B \to T \mid \Delta_0 \cup \Delta_1$ y se concluye por (APP) con π_v, $\pi' \rhd_{\mathcal{E}} \Gamma \vdash o' : T \mid \Delta$. El caso simétrico es análogo.
  - $\simeq_{\sigma_3}$. Luego, $o = (\lambda x.\mu\beta.[\alpha]\, t)\, u$ y $o' = \mu\beta.[\alpha]\, (\lambda x.t)\, u$ con $\beta \notin u$. Más aún, de π se tienen los contextos $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta = \Delta_0 \cup \Delta_1; (\alpha : T)^{\leq 1}$ con las derivaciones $\pi_t \rhd_{\mathcal{E}} \Gamma_0; (x : A)^{\leq 1} \vdash t : T \mid \Delta_0; (\alpha : T)^{\leq 1}; (\beta : T)^{\leq 1}$ y $\pi_u \rhd_{\mathcal{E}} \Gamma_1 \vdash u : A \mid \Delta_1; (\alpha : T)^{\leq 1}$. Por Lem. 4.3.1, se tiene $\beta \notin \mathsf{dom}(\Delta_1)$. Entonces, aplicando (ABS) con π_t, se deriva $\rhd_{\mathcal{E}} \Gamma_0 \vdash \lambda x.t : A \to T \mid \Delta_0; (\alpha : T)^{\leq 1}; (\beta : T)^{\leq 1}$. Luego, aplicando (APP) con π_u, $\rhd_{\mathcal{E}} \Gamma \vdash (\lambda x.t)\, u : T \mid \Delta; (\alpha : T)^{\leq 1}; (\beta : T)^{\leq 1}$. Finalmente, se concluye por (NAME) y (CONT), $\pi' \rhd_{\mathcal{E}} \Gamma \vdash o' : T \mid \Delta$. El caso simétrico es análogo.
- $\mathtt{O} = \boxdot$. Luego, $o = p \simeq_{\sigma_*} p' = o'$. Más aún, los objetos son necesariamente comandos. Se analiza la regla aplicada:
  - $\simeq_{\sigma_4}$. Luego, $o = [\alpha']\, (\mu\alpha.[\beta']\, (\mu\beta.c)\, v)\, u$ y $o' = [\beta']\, (\mu\beta.[\alpha']\, (\mu\alpha.c)\, u)\, v$ con $\alpha \notin v$, $\beta \notin u$, $\beta \neq \alpha'$ y $\alpha \neq \beta'$. Más aún, de π por las hipótesis se tienen los contextos $\Gamma = \Gamma_0 \cup \Gamma_1 \cup \Gamma_2$ y $\Delta = (\Delta_0 \cup \Delta_1 \cup \Delta_2); (\alpha' : B \cup \beta' : B')$ con una derivación $\pi_c \rhd_{\mathcal{E}} \Gamma_0 \vdash c \mid \Delta_0; (\alpha' : B)^{\leq 1} \cup (\beta' : B')^{\leq 1}; (\alpha : A \to B)^{\leq 1}; (\beta : A' \to B')^{\leq 1}$, otra derivación $\pi_u \rhd_{\mathcal{E}} \Gamma_1 \vdash u : A \mid \Delta_1; (\alpha' : B)^{\leq 1} \cup (\beta' : B')^{\leq 1}$, y una última derivación $\pi_v \rhd_{\mathcal{E}}$

$\Gamma_2 \vdash v : A' \mid \Delta_2; (\alpha' : B)^{\leq 1} \cup (\beta' : B')^{\leq 1}$, para c, u y v respectivamente. Por Lem. 4.3.1, se tiene $\alpha \notin \mathsf{dom}(\Delta_2)$ y $\beta \notin \mathsf{dom}(\Delta_1)$. Luego, aplicando (CONT) con π_c, se deriva $\rhd_{\mathcal{E}} \Gamma_0 \vdash \mu\alpha.c : A \to B \mid \Delta_0; (\alpha' : B)^{\leq 1} \cup (\beta' : B')^{\leq 1}; (\beta : A' \to B')^{\leq 1}$ y, por (APP) con π_u, se tiene $\rhd_{\mathcal{E}} \Gamma_0 \cup \Gamma_1 \vdash (\mu\alpha.c)\, u : B \mid \Delta_0 \cup \Delta_1; (\alpha' : B)^{\leq 1} \cup (\beta' : B')^{\leq 1}; (\beta : A' \to B')^{\leq 1}$. Más aún, aplicando (NAME) con el nombre α' seguida de (CONT) con β, se obtiene $\rhd_{\mathcal{E}} \Gamma_0 \cup \Gamma_1 \vdash \mu\beta.[\alpha'](\mu\alpha.c)\, u : A' \to B' \mid \Delta_0 \cup \Delta_1; \alpha' : B \cup (\beta' : B')^{\leq 1}$. Por (APP) con π_v, $\rhd_{\mathcal{E}} \Gamma_0 \cup \Gamma_1 \cup \Gamma_2 \vdash (\mu\beta.[\alpha'](\mu\alpha.c)\, u)\, v : B' \mid \Delta_0 \cup \Delta_1 \cup \Delta_2; \alpha' : B \cup (\beta' : B')^{\leq 1}$. Finalmente, se concluye por (NAME), $\pi' \rhd_{\mathcal{E}} \Gamma \vdash o' : T \mid \Delta$. El caso simétrico es análogo.

- $\simeq_{\sigma_5}$. Luego, $o = [\alpha'](\mu\alpha.[\beta']\,\lambda y.\mu\beta.c)\, u$ y $o' = [\beta']\,\lambda y.\mu\beta.[\alpha'](\mu\alpha.c)\, u$ con $y \notin u$, $\beta \notin u$, $\beta \neq \alpha'$ y $\alpha \neq \beta'$. Más aún, de π por las hipótesis se tienen los contextos $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta = (\Delta_0 \cup \Delta_1); (\alpha' : B \cup \beta' : A' \to B')$ con una derivación para el comando c, $\pi_c \rhd_{\mathcal{E}} \Gamma_0; (y : A')^{\leq 1} \vdash c \mid \Delta_0; (\alpha' : B)^{\leq 1} \cup (\beta' : A' \to B')^{\leq 1}; (\alpha : A \to B)^{\leq 1}; (\beta : B')^{\leq 1}$, y otra para el término u, $\pi_u \rhd_{\mathcal{E}} \Gamma_1 \vdash u : A \mid \Delta_1; (\alpha' : B)^{\leq 1} \cup (\beta' : A' \to B')^{\leq 1}$. Por Lem. 4.3.1, se tiene $y, \beta \notin \mathsf{dom}(\Delta_1)$. Entonces, aplicando (CONT) con π_c, se obtiene la derivación $\rhd_{\mathcal{E}} \Gamma_0; (y : A')^{\leq 1} \vdash \mu\alpha.c : A \to B \mid \Delta_0; (\alpha' : B)^{\leq 1} \cup (\beta' : A' \to B')^{\leq 1}; (\beta : B')^{\leq 1}$. Luego, basta aplicar la regla (APP) con la derivación anterior y π_u para obtener $\rhd_{\mathcal{E}} \Gamma_0 \cup \Gamma_1; (y : A')^{\leq 1} \vdash (\mu\alpha.c)\, u : B \mid \Delta_0 \cup \Delta_1; (\alpha' : B)^{\leq 1} \cup (\beta' : A' \to B')^{\leq 1}; (\beta : B')^{\leq 1}$. A su vez, aplicando (NAME) con el nombre α' seguida de (CONT) con β, se tiene $\rhd_{\mathcal{E}} \Gamma_0; (y : A')^{\leq 1} \vdash \mu\beta.[\alpha'](\mu\alpha.c)\, u : B' \mid \Delta_0 \cup \Delta_1; \alpha' : B \cup (\beta' : A' \to B')^{\leq 1}$. Entonces, por (ABS), $\rhd_{\mathcal{E}} \Gamma_0 \cup \Gamma_1 \vdash \lambda y.\mu\beta.[\alpha'](\mu\alpha.c)\, u : A' \to B' \mid \Delta_0 \cup \Delta_1; \alpha' : B \cup (\beta' : A' \to B')^{\leq 1}$. Finalmente, se concluye por (NAME), $\pi' \rhd_{\mathcal{E}} \Gamma \vdash o' : T \mid \Delta$. El caso simétrico es análogo.
- $\simeq_{\sigma_6}$. Luego, $o = [\alpha']\,\lambda x.\mu\alpha.[\beta']\,\lambda y.\mu\beta.c$ y $o' = [\beta']\,\lambda y.\mu\beta.[\alpha']\,\lambda x.\mu\alpha.c$ con $\beta \neq \alpha'$ y $\alpha \neq \beta'$. Más aún, de π se tiene el contexto $\Delta = \Delta_0; \alpha' : A \to B \cup \beta' : A' \to B'$ con $\pi_c \rhd_{\mathcal{E}} \Gamma; (x : A)^{\leq 1}; (y : A')^{\leq 1} \vdash c \mid \Delta_0; (\alpha' : A \to B)^{\leq 1} \cup (\beta' : A' \to B')^{\leq 1}; (\alpha : B)^{\leq 1}; (\beta : B')^{\leq 1}$. Por (CONT), (ABS) y (NAME) sobre α, x y α' respectivamente, se obtiene la derivación $\pi_c \rhd_{\mathcal{E}} \Gamma; (y : A')^{\leq 1} \vdash [\alpha']\,\lambda x.\mu\alpha.c \mid \Delta_0; \alpha' : A \to B \cup (\beta' : A' \to B')^{\leq 1}; (\beta : B')^{\leq 1}$. Así mismo, aplicando la misma secuencia de reglas con β, y y β' respectivamente se concluye, $\pi' \rhd_{\mathcal{E}} \Gamma \vdash o' : T \mid \Delta$. El caso simétrico es análogo.
- $\simeq_{\sigma_7}$. Luego, $o = [\beta]\,\mu\alpha.c$ y $o' = \{\alpha \setminus \beta\}c$. Más aún, de π se tiene $\Delta = \Delta_0; \beta : A$ y la derivación $\pi_c \rhd_{\mathcal{E}} \Gamma \vdash c \mid \Delta_0; (\alpha : A)^{\leq 1}; (\beta : A)^{\leq 1}$. Luego, concluye por Lem. C.0.23 con $\pi' \rhd_{\mathcal{E}} \Gamma \vdash \{\alpha \setminus \beta\}c \mid \Delta_0 \cup (\beta : A)^{\leq 1}$. Para el caso simétrico, se tiene $o = \{\alpha \setminus \beta\}c$ y $o' = [\mu\alpha.c]\,\beta$. De $\Delta = \Delta_0 \cup (\beta : A)^{\leq 1}$, por Lem. C.0.24, se obtiene la derivación $\pi_c \rhd_{\mathcal{E}} \Gamma \vdash c \mid \Delta_0; (\alpha : A)^{\leq 1}$. Finalmente, se concluye por (NAME) y (CONT), $\pi' \rhd_{\mathcal{E}} \Gamma \vdash o' \mid \Delta_0 \cup \beta : A$. Notar que $\alpha \notin \mathsf{fn}(o)$ implica $\alpha \notin \mathsf{dom}(\Delta_0) \subseteq \mathsf{dom}(\Delta)$, por Lem. 4.3.1.
- $\simeq_{\sigma_8}$. Luego, $o = \mu\alpha.[\alpha]\,t$ y $o' = t$ con $\alpha \notin t$. Más aún, de π se tiene la derivación $\rhd_{\mathcal{E}} \Gamma \vdash t : T \mid \Delta; (\alpha : T)^{\leq 1}$ y, por Lem. 4.3.1, $\alpha \notin \mathsf{dom}(\Delta; (\alpha : T)^{\leq 1})$. Luego, se concluye $\pi' \rhd_{\mathcal{E}} \Gamma \vdash o' : T \mid \Delta$. El caso simétrico es análogo.

- $\mathtt{O} = \mathtt{T}\, u$. Luego, $o = t\, u$ y $o' = t'\, u$ con $t \simeq_{\sigma} t'$. Más aún, por (APP) se tienen $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta = \Delta_0 \cup \Delta_1$ con $\pi_t \rhd_{\mathcal{E}} \Gamma_0 \vdash t : A \to T \mid \Delta_0$ y $\pi_u \rhd_{\mathcal{E}} \Gamma_1 \vdash u : A \mid \Delta_1$. Por *h.i.* existe $\pi_{t'} \rhd_{\mathcal{E}} \Gamma_0 \vdash t' : A \to T \mid \Delta_0'$. Finalmente, se concluye por (APP), $\pi' \rhd_{\mathcal{E}} \Gamma \vdash o' : T \mid \Delta'$ con $\Delta' = \Delta_0' \cup \Delta_1$. El caso simétrico es análogo.
- $\mathtt{O} = t\, \mathtt{T}$. Luego, $o = t\, u$ y $o' = t\, u'$ con $u \simeq_{\sigma} u'$. Más aún, por (APP) se tienen $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta = \Delta_0 \cup \Delta_1$ con $\pi_t \rhd_{\mathcal{E}} \Gamma_0 \vdash t : A \to T \mid \Delta_0$ y $\pi_u \rhd_{\mathcal{E}} \Gamma_1 \vdash u : A \mid \Delta_1$. Por *h.i.* existe $\pi_{u'} \rhd_{\mathcal{E}} \Gamma_1 \vdash u' : A \mid \Delta_1'$. Finalmente, se concluye por (APP), $\pi' \rhd_{\mathcal{E}} \Gamma \vdash o' : T \mid \Delta'$ con $\Delta' = \Delta_0 \cup \Delta_1'$. El caso simétrico es análogo.
- $\mathtt{O} = \lambda x.\mathtt{T}$. Luego, $o = \lambda x.t$ y $o' = \lambda x.t'$ con $t \simeq_{\sigma} t'$. Más aún, por (ABS) se tiene $T = A \to B$ con $\pi_t \rhd_{\mathcal{E}} \Gamma; (x : A)^{\leq 1} \vdash t : B \mid \Delta$. Por *h.i.* existe $\pi_{t'} \rhd_{\mathcal{E}} \Gamma; (x : A)^{\leq 1} \vdash t' : B \mid \Delta'$. Finalmente, se concluye por (ABS), $\pi' \rhd_{\mathcal{E}} \Gamma \vdash o' : T \mid \Delta'$. El caso simétrico es análogo.

- $\mathtt{O} = \mu\alpha.\mathtt{C}$. Luego, $o = \mu\alpha.c$ y $o' = \mu\alpha.c'$ con $c \simeq_\sigma c'$. Más aún, por (CONT) se tiene $\pi_c \rhd_{\mathcal{E}} \Gamma \vdash c \mid \Delta; (\alpha : T)^{\leq 1}$. Por *h.i.* existe $\pi_{c'} \rhd_{\mathcal{E}} \Gamma \vdash c' \mid \Delta'; (\alpha : T)^{\leq 1}$. Se concluye por (CONT), $\pi' \rhd_{\mathcal{E}} \Gamma \vdash o' : T \mid \Delta'$. El caso simétrico es análogo.

- $\mathtt{O} = [\alpha]\,\mathtt{T}$. Luego, $o = [\alpha]\,t$ y $o' = [\alpha]\,t'$ con $t \simeq_\sigma t'$. Más aún, por (NAME) se tiene $\Delta = \Delta_0; \alpha : A$ con $\pi_t \rhd_{\mathcal{E}} \Gamma \vdash t : A \mid \Delta_0; (\alpha : A)^{\leq 1}$. Por *h.i.* existe $\pi_{t'} \rhd_{\mathcal{E}} \Gamma \vdash t' : A \mid \Delta'_0; (\alpha : A)^{\leq 1}$. Se concluye por (NAME), $\pi' \rhd_{\mathcal{E}} \Gamma \vdash o' : T \mid \Delta'$ con $\Delta' = \Delta'_0; \alpha : A$. El caso simétrico es análogo.

□

Traducción a proof-nets

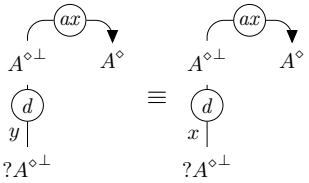
Dados $o \in \mathbb{O}_{\Lambda M}$ tipado, y variables x e y, se denota $(\pi_o)^{\Diamond, x, y}$ a la polarized proof-net resultante de agregar un nodo de contracción entre entre los hílos x e y si $x, y \in \mathsf{fv}(o)$. De lo contrario $(\pi_o)^{\Diamond, x, y} \triangleq (\pi_o)^{\Diamond}$. Análogamente para nombres α y β respectivamente. Notar que en particular, dado un comando c, resulta $(\pi_c)^{\Diamond, \alpha, \beta} = (\pi_{c[\alpha \setminus \beta]})^{\Diamond}$.

Lema C.0.25. *Sea $o \in \mathbb{O}_{\Lambda M}$ tipado. Luego, $(\pi_{\{x \setminus y\} o})^{\Diamond} \equiv (\pi_o)^{\Diamond, x, y}$ y $(\pi_{\{\alpha \setminus \beta\} o})^{\Diamond} \equiv (\pi_o)^{\Diamond, \alpha, \beta}$.*

Demostración. Por inducción en o.

- $o = x$. Luego, $\{x \setminus y\} o = y$. Más aún, $y \notin \mathsf{fv}(o)$, por lo que se tiene

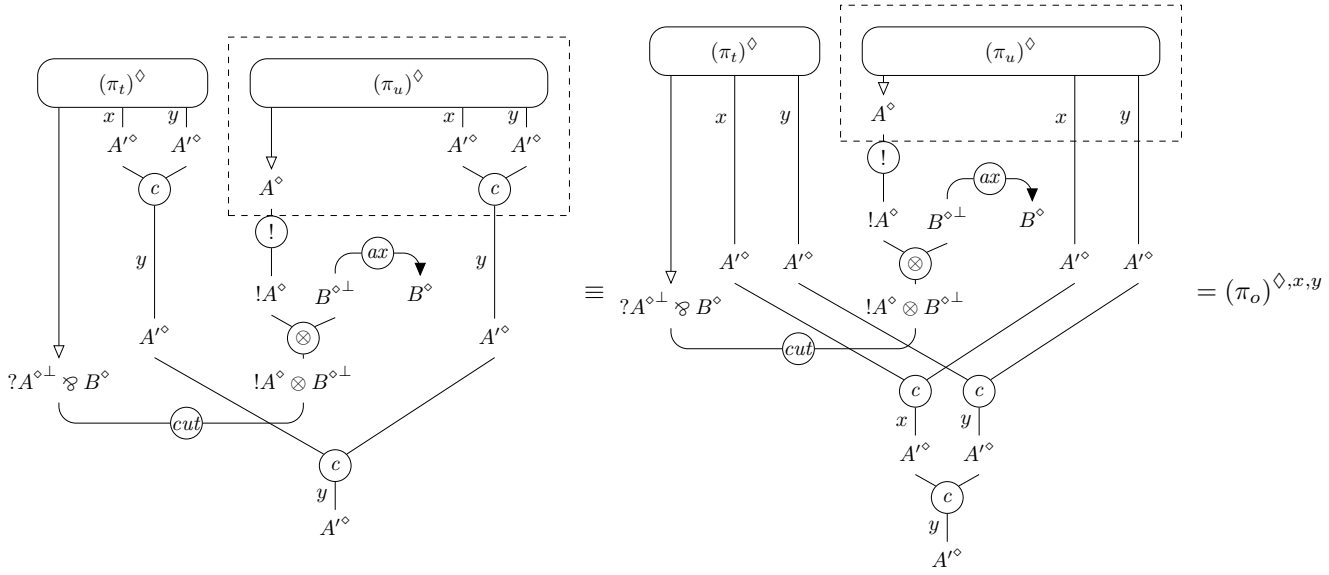
$$(\pi_{\{x \setminus y\} o})^{\Diamond} = \quad \equiv \quad = (\pi_o)^{\Diamond, x, y}$$



- $o = z \neq x$. Luego, $\{x \setminus y\} o = z$. El resultado es inmediato por reflexividad de $\equiv$, dado que $x \notin \mathsf{fv}(o)$ implica $(\pi_o)^{\Diamond, x, y} = (\pi_o)^{\Diamond}$ por definición. El caso para α y β es análogo.

- $o = t\,u$. Luego, $\{x \setminus y\} o = \{x \setminus y\} t\, \{x \setminus y\} u$. Por *h.i.* se tienen $(\pi_{\{x \setminus y\} t})^{\Diamond} \equiv (\pi_t)^{\Diamond, x, y}$ y $(\pi_{\{x \setminus u\} t})^{\Diamond} \equiv (\pi_u)^{\Diamond, x, y}$. Se ilustra únicamente el caso donde $x, y \in \mathsf{fv}(t)$ y $x, y \in \mathsf{fv}(u)$. Les restantes combina-

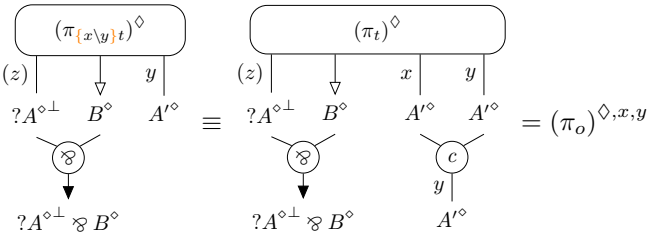
ciones resultan en PPNs más simples. Luego, se tiene

$(\pi_{\{x\setminus y\}o})^{\Diamond} \equiv$ 

$= (\pi_o)^{\Diamond,x,y}$

El caso para α y β es análogo.

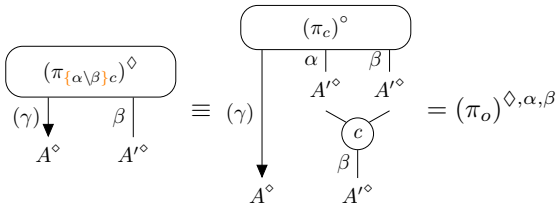
- $o = \lambda z.t$ con $z \neq x$ y $z \neq y$. Luego, $\{x \setminus y\}o = \lambda z.\{x \setminus y\}t$. Por *h.i.* se tiene $(\pi_{\{x\setminus y\}t})^{\Diamond} \equiv (\pi_t)^{\Diamond,x,y}$. Se ilustra únicamente el caso donde $x, y \in \mathsf{fv}(t)$. Les restantes combinaciones resultan en PPNs más simples. Luego, se tiene

$(\pi_{\{x\setminus y\}o})^{\Diamond} =$ 

$= (\pi_o)^{\Diamond,x,y}$

El caso para α y β es análogo.

- $o = \mu\gamma.c$ con $\gamma \neq \alpha$ y $\gamma \neq \beta$. Luego, $\{\alpha \setminus \beta\}o = \mu\gamma.\{\alpha \setminus \beta\}c$. Por *h.i.* se tiene $(\pi_{\{\alpha\setminus\beta\}c})^{\Diamond} \equiv (\pi_c)^{\Diamond,\alpha,\beta}$. Se ilustra únicamente el caso donde $\alpha, \beta, \gamma \in \mathsf{fn}(c)$. Les restantes combinaciones resultan en PPNs más simples. Luego, se tiene

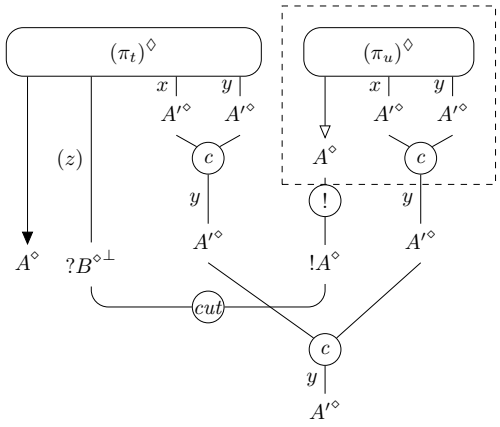
$(\pi_{\{\alpha\setminus\beta\}o})^{\Diamond} =$ 

$= (\pi_o)^{\Diamond,\alpha,\beta}$

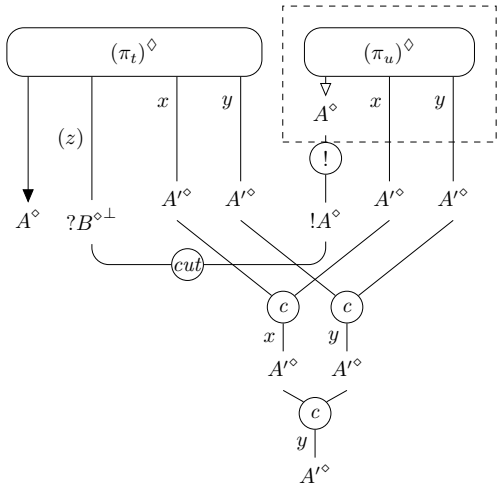
El caso para x e y es análogo.

- $o = t[z \setminus u]$ con $z \neq x$ y $z \neq y$. Luego, $\{x \setminus y\}o = \{x \setminus y\}t[z \setminus \{x \setminus y\}u]$. Por *h.i.* se tienen $(\pi_{\{x\setminus y\}t})^{\Diamond} \equiv (\pi_t)^{\Diamond,x,y}$ y $(\pi_{\{x\setminus u\}t})^{\Diamond} \equiv (\pi_u)^{\Diamond,x,y}$. Se ilustra únicamente el caso donde $x, y \in \mathsf{fv}(t)$ y $x, y \in \mathsf{fv}(u)$. Les restantes combinaciones resultan en PPNs más simples. Luego, se tiene

$(\pi_{\{x\setminus y\}o})^{\Diamond} \equiv$



$\equiv$

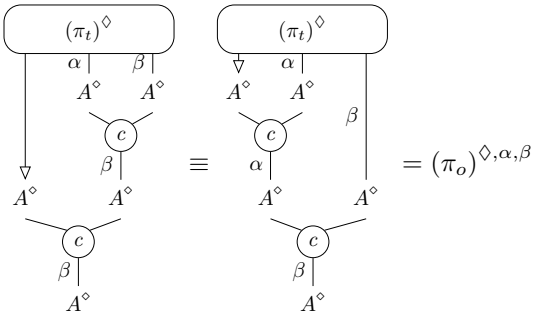


$= (\pi_o)^{\Diamond,x,y}$

El caso para α y β es análogo.

- $o = [\alpha]\, t$. Luego, $\{\alpha \setminus \beta\}o = [\beta]\, \{\alpha \setminus \beta\}t$. Por *h.i.* se tiene $(\pi_{\{\alpha\setminus\beta\}t})^{\Diamond} \equiv (\pi_t)^{\Diamond,\alpha,\beta}$. Se ilustra únicamente el caso donde $\alpha, \beta, \gamma \in \mathsf{fn}(t)$. Les restantes combinaciones resultan en PPNs más simples. Luego, se tiene

$(\pi_{\{\alpha\setminus\beta\}o})^{\Diamond} \equiv$

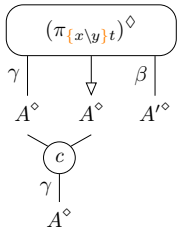


$\equiv$

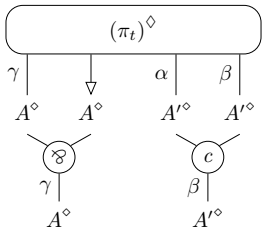
$= (\pi_o)^{\Diamond,\alpha,\beta}$

- $o = [\gamma]\, t$ con $\gamma \neq \alpha$. Luego, $\{\alpha \setminus \beta\}o = [\beta]\, \{\alpha \setminus \beta\}t$. Por *h.i.* se tiene $(\pi_{\{\alpha\setminus\beta\}t})^{\Diamond} \equiv (\pi_t)^{\Diamond,\alpha,\beta}$. Se ilustra únicamente el caso donde $\alpha, \beta, \gamma \in \mathsf{fn}(t)$. Les restantes combinaciones resultan en PPNs más simples. Luego, se tiene

$(\pi_{\{\alpha\setminus\beta\}o})^{\Diamond} =$



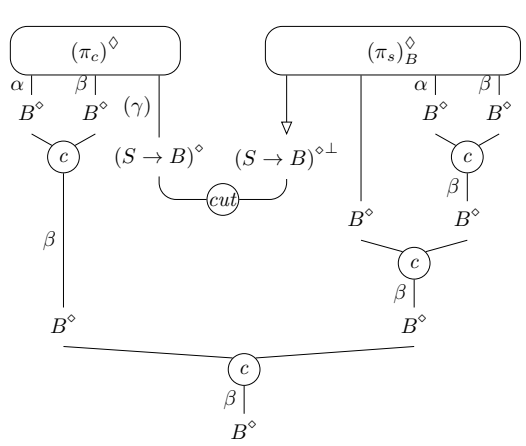
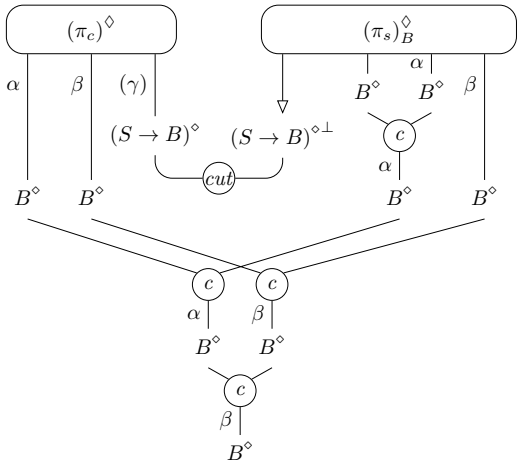
$\equiv$



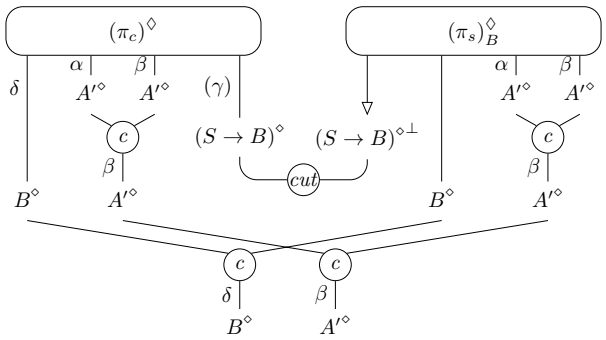
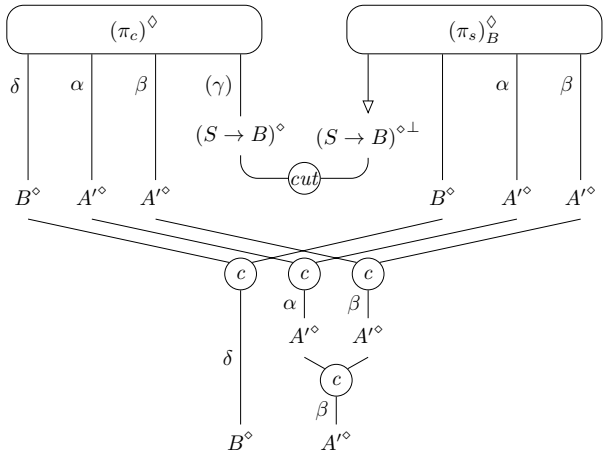
$= (\pi_o)^{\Diamond,\alpha,\beta}$

El caso para x e y es análogo.

- $o = c[\![\gamma \setminus^{\alpha} s]\!]$ con $\gamma \neq \alpha$ y $\gamma \neq \beta$. Luego, $\{\alpha \setminus \beta\}o = (\{\alpha \setminus \beta\}c)[\![\gamma \setminus^{\beta} \{\alpha \setminus \beta\}s]\!]$. Por *h.i.* se tienen $(\pi_{\{\alpha\setminus\beta\}c})^{\Diamond} \equiv (\pi_c)^{\Diamond,\alpha,\beta}$ y $(\pi_{\{\alpha\setminus\beta\}s})^{\Diamond} \equiv (\pi_s)^{\Diamond,\alpha,\beta}$. Se ilustra únicamente el caso donde $\alpha, \beta, \gamma \in \mathsf{fn}(c)$ y $\alpha, \beta \in \mathsf{fn}(s)$. Les restantes combinaciones resultan en PPNs más simples. Luego, se tiene

$(\pi_{\{\alpha\setminus\beta\}o})^{\Diamond} \equiv$  $\equiv$  $= (\pi_o)^{\Diamond,\alpha,\beta}$

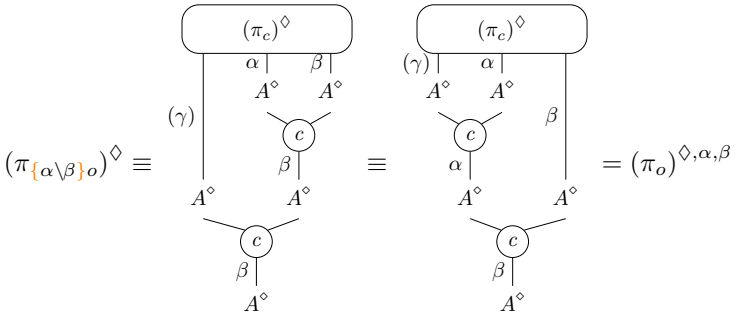
- $o = c[\![\gamma \setminus^{\delta} s]\!]$ con $\gamma \neq \alpha$, $\gamma \neq \beta$ y $\delta \neq \alpha$. Luego, $\{\alpha \setminus \beta\}o = (\{\alpha \setminus \beta\}c)[\![\gamma \setminus^{\delta} \{\alpha \setminus \beta\}s]\!]$. Por *h.i.* se tienen $(\pi_{\{\alpha\setminus\beta\}c})^{\Diamond} \equiv (\pi_c)^{\Diamond,\alpha,\beta}$ y $(\pi_{\{\alpha\setminus\beta\}s})^{\Diamond} \equiv (\pi_s)^{\Diamond,\alpha,\beta}$. Se ilustra únicamente el caso donde $\alpha, \beta, \gamma, \delta \in \mathsf{fn}(c)$, $\alpha, \beta \in \mathsf{fn}(s)$ y $\delta \notin \mathsf{fn}(s)$ con $\delta \neq \beta$. Les restantes combinaciones resultan en PPNs similares o más simples. Luego, se tiene

$(\pi_{\{\alpha\setminus\beta\}o})^{\Diamond} \equiv$  $\equiv$  $= (\pi_o)^{\Diamond,\alpha,\beta}$

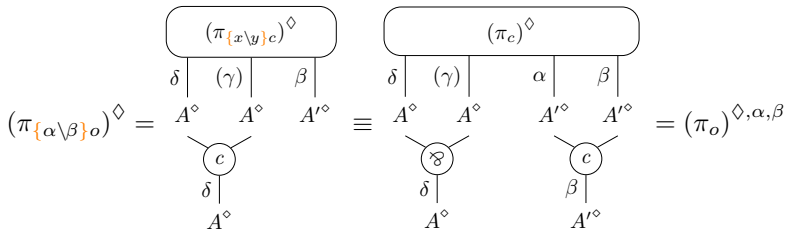
El caso para x e y es análogo.

- $o = c[\gamma \setminus \alpha]$. con $\gamma \neq \alpha$ y $\gamma \neq \beta$. Luego, $\{\alpha \setminus \beta\}o = (\{\alpha \setminus \beta\}c)[\gamma \setminus \beta]$. Por *h.i.* se tiene $(\pi_{\{\alpha\setminus\beta\}c})^{\Diamond} \equiv (\pi_c)^{\Diamond,\alpha,\beta}$. Se ilustra únicamente el caso donde $\alpha, \beta, \gamma \in \mathsf{fn}(c)$. Les restantes combi-

naciones resultan en PPNs más simples. Luego, se tiene



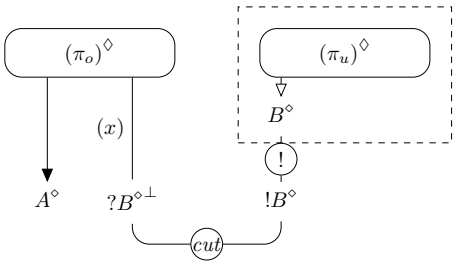
- $o = c[\gamma \setminus \delta]$ con $\gamma \neq \alpha$, $\gamma \neq \beta$ y $\delta \neq \alpha$. Luego, $\{\alpha \setminus \beta\}o = (\{\alpha \setminus \beta\}c)[\gamma \setminus \delta]$. Por *h.i.* se tienen $(\pi_{\{\alpha\setminus\beta\}c})^{\Diamond} \equiv (\pi_c)^{\Diamond,\alpha,\beta}$. Se ilustra únicamente el caso donde $\alpha, \beta, \gamma, \delta \in \mathsf{fn}(c)$ con $\delta \neq \beta$. Les restantes combinaciones resultan en PPNs similares o más simples. Luego, se tiene



El caso para x e y es análogo.

□

Lema C.0.26 (Sustitución)**.** *Sean $o \in \mathbb{O}_{\Lambda M}$ y $u \in \mathbb{T}_{\Lambda M}$ tipados tal que $\mathsf{fv}(o) \cap \mathsf{fv}(u) = \emptyset$. Luego,*

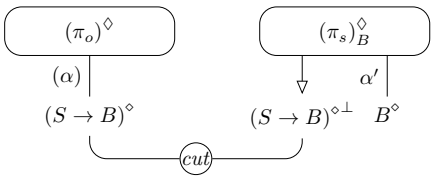


reduce a una PPN estructuralmente equivalente a $(\pi_{\{x\setminus u\}o})^{\Diamond}$.

Demostración. Por una simple inducción en o usando la Def 4.4.2.

□

Lema C.0.27 (Replacement)**.** *Sean $o, s \in \mathbb{C}_{\Lambda M}$ tipados con s un stack tal que $\mathsf{fn}(o) \cap \mathsf{fn}(s) = \emptyset$. Luego,*



reduce a una PPN estructuralmente equivalente a $(\pi_{\{\!\{\alpha\setminus\alpha' s\}\!\}c})^{\blacklozenge}$.

Demostración. Por una simple inducción en o usando la Def 4.4.2.

□

Teorema 4.4.3 (Simulación)**.** *Sean $o, p \in \mathbb{O}_{\Lambda M}$ tipados. Si $o \to_{\Lambda M} p$ y π_o, π_p son las derivaciones de tipos relacionadas correspondientes, entonces $(\pi_o)^{\blacklozenge} \twoheadrightarrow_{\mathtt{PPN}} (\pi_p)^{\blacklozenge}$.*

Demostración. Por una simple inducción en o apelando a los Lem. C.0.25, C.0.26 y C.0.27.

□

Reducción meaningful para ΛM-cálculo

Para demostrar la normalización fuerte de la relación de reducción canónica es preciso introducir una serie de medidas sobre objetos del ΛM-cálculo. En primer lugar se define el *tamaño* de un objeto como:

$$\begin{array}{rcl} \mathsf{sz}(x) & \triangleq & 1 \\ \mathsf{sz}(t\,u) & \triangleq & \mathsf{sz}(t)+\mathsf{sz}(u)+1 \\ \mathsf{sz}(\lambda x.t) & \triangleq & \mathsf{sz}(t)+1 \\ \mathsf{sz}(\mu\alpha.c) & \triangleq & \mathsf{sz}(c)+1 \\ \mathsf{sz}(t[x \setminus u]) & \triangleq & \mathsf{sz}(t)+\mathsf{sz}(u)+1 \end{array} \qquad \begin{array}{rcl} \mathsf{sz}([\alpha]\,t) & \triangleq & \mathsf{sz}(t)+1 \\ \mathsf{sz}(c[\![\alpha \setminus^{\alpha'} s]\!]) & \triangleq & \mathsf{sz}(c)+\mathsf{sz}(s)+1 \\ \mathsf{sz}(c[\alpha \setminus \beta]) & \triangleq & \mathsf{sz}(c)+1 \\ \mathsf{sz}(t \cdot s) & \triangleq & \mathsf{sz}(t)+\mathsf{sz}(s)+1 \end{array}$$

La noción de tamaño se extiende a contextos definiendo $\mathsf{sz}(\Box) \triangleq \mathsf{sz}(\boxdot) \triangleq 0$, de modo que $\mathsf{sz}(\mathtt{O}\langle o\rangle) = \mathsf{sz}(\mathtt{O}) + \mathsf{sz}(o)$.

Lema C.0.28. *Sea $o \in \mathbb{O}_{\Lambda M}$. Si $o \to_{\mathfrak{C}} o'$, entonces $\mathsf{sz}(o) \geq \mathsf{sz}(o')$.*

Demostración. Por definición, $o \to_{\mathfrak{C}} o'$ implica $o = \mathtt{O}\langle l\rangle$ y $o' = \mathtt{O}\langle r\rangle$ con $l \mapsto_* r$, $* \in \{\mathtt{dB}, \mathtt{dM}, \mathtt{N}, \mathtt{P}, \mathtt{W}\}$. La demostración es por inducción en $\mathtt{O}$.

- $\mathtt{O} = \Box$. Luego, se tienen dos casos posibles según la regla de reescritura aplicada al reducir.
  1. $\mathtt{dB}$. Luego, $o = \mathtt{L}\langle\lambda x.t\rangle\, u$ y $o' = \mathtt{L}\langle t[x \setminus u]\rangle$. Se concluye dado que $\mathsf{sz}(o) = \mathsf{sz}(\mathtt{L}) + (\mathsf{sz}(t)+1) + \mathsf{sz}(u) + 1 > \mathsf{sz}(\mathtt{L}) + \mathsf{sz}(t) + \mathsf{sz}(u) + 1 = \mathsf{sz}(o')$.
  2. $\mathtt{dM}$. Luego, $o = \mathtt{L}\langle\mu\alpha.c\rangle\, u$ y $o' = \mathtt{L}\langle\mu\alpha'.c[\![\alpha \setminus^{\alpha'} u]\!]\rangle$ con α' un nombre fresco. Se concluye dado que $\mathsf{sz}(o) = \mathsf{sz}(\mathtt{L}) + (\mathsf{sz}(c)+1) + \mathsf{sz}(u) + 1 = \mathsf{sz}(\mathtt{L}) + (\mathsf{sz}(c) + \mathsf{sz}(u) + 1) + 1 = \mathsf{sz}(o')$.
- $\mathtt{O} = \boxdot$. Luego, se tienen tres casos posibles según la regla de reescritura aplicada al reducir.
  1. $\mathtt{N}$. Luego, $o = \mathtt{LCC}\langle[\alpha]\,t\rangle[\![\alpha \setminus^{\alpha'} s]\!]$ y $o' = \mathtt{LCC}\langle[\alpha']\,t :: s\rangle$. Más aún, se tiene $\mathsf{sz}(t :: s) = \mathsf{sz}(t) + \mathsf{sz}(s) + 1$. Se concluye dado que $\mathsf{sz}(o) = (\mathsf{sz}(\mathtt{LCC}) + (\mathsf{sz}(t)+1)) + \mathsf{sz}(s) + 1 = \mathsf{sz}(\mathtt{LCC}) + ((\mathsf{sz}(t) + \mathsf{sz}(s) + 1) + 1) = \mathsf{sz}(o')$.
  2. $\mathtt{P}$. Luego, $o = \mathtt{LCC}\langle c[\![\beta \setminus^{\alpha} s']\!][\![\alpha \setminus^{\alpha'} s]\!]\rangle$ y $o' = \mathtt{LCC}\langle c[\![\beta \setminus^{\alpha'} s' \cdot s]\!]\rangle$. Se concluye dado que $\mathsf{sz}(o) = (\mathsf{sz}(\mathtt{LCC}) + (\mathsf{sz}(c) + \mathsf{sz}(s') + 1)) + \mathsf{sz}(s) + 1 = \mathsf{sz}(\mathtt{LCC}) + (\mathsf{sz}(c) + (\mathsf{sz}(s') + \mathsf{sz}(s) + 1) + 1) = \mathsf{sz}(o')$.
  3. $\mathtt{W}$. Luego, $o = \mathtt{LCC}\langle c[\beta \setminus \alpha]\rangle[\![\alpha \setminus^{\alpha'} s]\!]$ y $o' = \mathtt{LCC}\langle c[\![\beta \setminus^{\alpha} s]\!][\alpha \setminus \alpha']\rangle$. Se concluye dado que $\mathsf{sz}(o) = (\mathsf{sz}(\mathtt{LCC}) + (\mathsf{sz}(c)+1)) + \mathsf{sz}(s) + 1 = \mathsf{sz}(\mathtt{LCC}) + ((\mathsf{sz}(c) + \mathsf{sz}(s) + 1) + 1) = \mathsf{sz}(o')$.
- $\mathtt{O} = \mathtt{T}\,u$. Luego, $o = t\,u$ y $o' = t'\,u$ con $t \to_{\mathfrak{C}} t'$. Por *h.i.* se tiene $\mathsf{sz}(t) \geq \mathsf{sz}(t')$. Se concluye pues $\mathsf{sz}(o) = \mathsf{sz}(t) + \mathsf{sz}(u) + 1 \geq \mathsf{sz}(t') + \mathsf{sz}(u) + 1 = \mathsf{sz}(o')$.
- $\mathtt{O} = t\,\mathtt{T}$. Luego, $o = t\,u$ y $o' = t\,u'$ con $u \to_{\mathfrak{C}} u'$. Por *h.i.* se tiene $\mathsf{sz}(u) \geq \mathsf{sz}(u')$. Se concluye pues $\mathsf{sz}(o) = \mathsf{sz}(t) + \mathsf{sz}(u) + 1 \geq \mathsf{sz}(t) + \mathsf{sz}(u') + 1 = \mathsf{sz}(o')$.

- $\mathtt{O} = \lambda x.\mathtt{T}$. Luego, $o = \lambda x.t$ y $o' = \lambda x.t'$ con $t \to_{\mathfrak{C}} t'$. Por *h.i.* se tiene $\mathsf{sz}(t) \geq \mathsf{sz}(t')$. Se concluye pues $\mathsf{sz}(o) = \mathsf{sz}(t) + 1 \geq \mathsf{sz}(t') + 1 = \mathsf{sz}(o')$.
- $\mathtt{O} = \mu\alpha.\mathtt{C}$. Luego, $o = \mu\alpha.c$ y $o' = \mu\alpha.c'$ con $c \to_{\mathfrak{C}} c'$. Por *h.i.* se tiene $\mathsf{sz}(c) \geq \mathsf{sz}(c')$. Se concluye pues $\mathsf{sz}(o) = \mathsf{sz}(c) + 1 \geq \mathsf{sz}(c') + 1 = \mathsf{sz}(o')$.
- $\mathtt{O} = \mathtt{T}[x \setminus u]$. Luego, $o = t[x \setminus u]$ y $o' = t'[x \setminus u]$ con $t \to_{\mathfrak{C}} t'$. Por *h.i.* se tiene $\mathsf{sz}(t) \geq \mathsf{sz}(t')$. Se concluye pues $\mathsf{sz}(o) = \mathsf{sz}(t) + \mathsf{sz}(u) + 1 \geq \mathsf{sz}(t') + \mathsf{sz}(u) + 1 = \mathsf{sz}(o')$.
- $\mathtt{O} = t[x \setminus \mathtt{T}]$. Luego, $o = t[x \setminus u]$ y $o' = t[x \setminus u']$ con $u \to_{\mathfrak{C}} u'$. Por *h.i.* se tiene $\mathsf{sz}(u) \geq \mathsf{sz}(u')$. Se concluye pues $\mathsf{sz}(o) = \mathsf{sz}(t) + \mathsf{sz}(u) + 1 \geq \mathsf{sz}(t) + \mathsf{sz}(u') + 1 = \mathsf{sz}(o')$.
- $\mathtt{O} = [\alpha]\,\mathtt{T}$. Luego, $o = [\alpha]\,t$ y $o' = [\alpha]\,t'$ con $t \to_{\mathfrak{C}} t'$. Por *h.i.* se tiene $\mathsf{sz}(t) = \mathsf{sz}(t')$. Se concluye pues $\mathsf{sz}(o) \geq \mathsf{sz}(t) + 1 \geq \mathsf{sz}(t') + 1 = \mathsf{sz}(o')$.
- $\mathtt{O} = \mathtt{C}[\![\alpha \setminus^{\alpha'} s]\!]$. Luego, $o = c[\![\alpha \setminus^{\alpha'} s]\!]$ y $o' = c'[\![\alpha \setminus^{\alpha'} s]\!]$ con $c \to_{\mathfrak{C}} c'$. Por *h.i.* se tiene $\mathsf{sz}(c) \geq \mathsf{sz}(c')$. Se concluye pues $\mathsf{sz}(o) = \mathsf{sz}(c) + \mathsf{sz}(s) + 1 \geq \mathsf{sz}(c') + \mathsf{sz}(s) + 1 = \mathsf{sz}(o')$.
- $\mathtt{O} = c[\![\alpha \setminus^{\alpha'} \mathtt{S}]\!]$. Luego, $o = c[\![\alpha \setminus^{\alpha'} s]\!]$ y $o' = c[\![\alpha \setminus^{\alpha'} s']\!]$ con $s \to_{\mathfrak{C}} s'$. Por *h.i.* se tiene $\mathsf{sz}(s) \geq \mathsf{sz}(s')$. Se concluye pues $\mathsf{sz}(o) = \mathsf{sz}(c) + \mathsf{sz}(s) + 1 \geq \mathsf{sz}(c) + \mathsf{sz}(s') + 1 = \mathsf{sz}(o')$.
- $\mathtt{O} = \mathtt{C}[\alpha \setminus \beta]$. Luego, $o = c[\alpha \setminus \beta]$ y $o' = c'[\alpha \setminus \beta]$ con $c \to_{\mathfrak{C}} c'$. Por *h.i.* se tiene $\mathsf{sz}(c) \geq \mathsf{sz}(c')$. Se concluye pues $\mathsf{sz}(o) = \mathsf{sz}(c) + 1 \geq \mathsf{sz}(c') + 1 = \mathsf{sz}(o')$.
- $\mathtt{O} = \mathtt{T} \cdot s$. Luego, $o = t \cdot s$ y $o' = t' \cdot s$ con $t \to_{\mathfrak{C}} t'$. Por *h.i.* se tiene $\mathsf{sz}(t) \geq \mathsf{sz}(t')$. Se concluye pues $\mathsf{sz}(o) = \mathsf{sz}(t) + \mathsf{sz}(s) + 1 \geq \mathsf{sz}(t') + \mathsf{sz}(s) + 1 = \mathsf{sz}(o')$.
- $\mathtt{O} = t \cdot \mathtt{S}$. Luego, $o = t \cdot s$ y $o' = t \cdot s'$ con $s \to_{\mathfrak{C}} s'$. Por *h.i.* se tiene $\mathsf{sz}(s) \geq \mathsf{sz}(s')$. Se concluye pues $\mathsf{sz}(o) = \mathsf{sz}(t) + \mathsf{sz}(s) + 1 \geq \mathsf{sz}(t) + \mathsf{sz}(s') + 1 = \mathsf{sz}(o')$.

□

Esta medida permite demostrar que el tamaño de un objeto decrece al reducir, aunque este decrecimiento no es necesariamente estricto. Esto se debe a cuatro razones principales.

1. La regla `dM` descarta una aplicación al mismo tiempo que introduce un replacement explícito, preservando la cantidad de constructores del objeto.
2. La regla `N` descarta un replacement explícito lineal sobre un stack de n elementos reemplazándolo por n aplicaciones. La cantidad de constructores de un stack de n elementos resulta ser $n - 1$ que, sumados al replacement descartado, compensa las n aplicaciones introducidas.
3. La regla `P` descarta un replacement explícito lineal al combinarlo con otro, concatenando sus respectivos stacks. Esto introduce un nuevo constructor de stacks, preservando la cantidad de constructores del objeto.
4. La regla `W` simplemente re-ordena los constructores del objeto, empujando hacia adentro los replacement explícitos lineales.

Notar que la primera observación sugiere que las aplicaciones deberían tener más peso que los replacements para garantizar la terminación mediante un método de interpretación polinomial. Sin embargo, la segunda insinúa exactamente lo contrario. La tercera por su parte requiere que los replacements tengan más peso que la concatenación de stacks. En cuanto a la cuarta, simplemente sugiere que los replacements explícitos a izquierda de renamings deben tener menor peso que los aquellos a la derecha. Bajo estas observaciones, se considera dos medidas sobre los objetos de ΛM-cálculo: en

primer lugar una función de peso que lleva cuenta de los nombres usados en el objeto y el peso de sus potenciales argumentos, y en segundo lugar una interpretación de objetos en multi-conjuntos basada en la función de peso anterior y adecuada para probar la normalización fuerte de la relación de reducción canónica.

Se define entonces una medida sobre los objeto del ΛM-cálculo que resulta de contar la cantidad de μ-abstracciones, términos *name* y renaming explícitos en el objeto, asignándole a los términos *name* un peso acorde a la medida de los argumentos que el comando en cuestión eventualmente consumirá durante la evaluación del objeto a su forma canónica. Sea τ una función de nombres en listas de números naturales y l una lista de números naturales, la *función de peso* $|\_|_l^\tau$ para objetos del ΛM-cálculo se define inductivamente como:

$$
\begin{array}{rcll}
|x|_l^\tau & \triangleq & 0 & \\
|t\,u|_l^\tau & \triangleq & |t|^\tau_{|u|^\tau_{[]}:l} + |u|^\tau_{[]} & \\
|\lambda x.t|_l^\tau & \triangleq & |t|^\tau_{l'} & l' = \begin{cases} [] & \text{si } l = [] \\ ns & \text{si } l = n : ns \end{cases} \\
|\mu\alpha.c|_l^\tau & \triangleq & |c|^{\tau[\alpha:=l]}_{[]} + 1 & \\
|t[x \setminus u]|_l^\tau & \triangleq & |t|_l^\tau & \\
|[\alpha]\,t|_l^\tau & \triangleq & |t|^\tau_{\tau(\alpha)} + 1 + \sum \tau(\alpha) & \\
|c[\![\alpha \setminus^{\alpha'} s]\!]|_l^\tau & \triangleq & |c|^{\tau[\alpha:=|u_0|^\tau_{[]}:\ldots:|u_n|^\tau_{[]}:\tau(\alpha')]}_{[]} & s = u_0 \cdot \ldots \cdot u_n \\
|c[\alpha \setminus \beta]|_l^\tau & \triangleq & |c|^{\tau[\alpha:=\tau(\beta)]}_{[]} + 1 & \\
|t \cdot s|_l^\tau & \triangleq & |t|^\tau_{[]} + |s|^\tau_{[]} &
\end{array}
$$

Notar que se ignoran las ocurrencias en las sustituciones y replacements explícitos, puesto que éstas no intervienen en la reducción canónica. Las ocurrencias en el stack de un replacement explícito son, sin embargo, propagadas a traves de la función τ para ser eventualmente contabilizadas en el correspondiente término *name*, puesto que pueden eventualmente quedar involucradas en un `N`-redex. La lista l debe interpretarse como una pila con el peso de los argumentos al objeto en cuestión, y $\sum l$ denota la suma de todos los elementos de l. En particular, dado que los comandos y stacks no son aplicables en el ΛM-cálculo, la lista es ignorada en estos casos. A modo ilustrativo, considerar el término $t = (\mu\alpha.([\beta]\,x)[\![\beta \setminus^\alpha \mu\delta.[\delta]\,x]\!])\,(\mu\delta.[\delta]\,x)$, luego se tiene $|\mu\delta.[\delta]\,x|^\tau_{[]} = |[\delta]\,x|^{\tau[\delta:=[]]}_{[]} + 1 = 0 + 1 + \sum [] + 1 = 2$ y resulta $|t|^\tau_{[]} = |(\mu\alpha.([\beta]\,x)[\![\beta \setminus^\alpha \mu\delta.[\delta]\,x]\!])|^\tau_{[2]} + 2 = |([\beta]\,x)[\![\beta \setminus^\alpha \mu\delta.[\delta]\,x]\!]|^{\tau[\alpha:=[2]]}_{[]} + 2 + 1 = |[\beta]\,x|^{\tau[\alpha:=[2]][\beta:=[2,2]]}_{[]} + 3 = 0 + 1 + \sum [2,2] + 3 = 8$.

A continuación se presentan algunos resultados auxiliares sobre la función de peso. Para eso se considera la siguiente noción de orden entre listas de números naturales: sean $l = [a_1, \ldots, a_n]$ y $l' = [b_1, \ldots, b_m]$, entonces $l \geq l'$ sii $n = m$ y $a_i \geq b_i$ para todo $i \in [1, n]$. Esta noción de orden garantiza, en particular, que $l \geq l'$ implica $\sum l \geq \sum l'$.

Lema C.0.29. *Sea $o \in \mathbb{O}_{\Lambda M}$.*

1. *Si $\alpha \notin \mathsf{fn}(o)$, entonces $|o|_l^{\tau[\alpha:=l']} = |o|_l^\tau$.*
2. *Si $\tau(\alpha) \geq \tau'(\alpha)$ para todo $\alpha \in o$, entonces $|o|_l^\tau \geq |o|_l^{\tau'}$.*
3. *Si $l \geq l'$, entonces $|o|_l^\tau \geq |o|_{l'}^\tau$.*

Demostración. Se muestran las tres propiedades simultáneamente por inducción en o.

- $o = x$. Luego, los tres resultados son inmediatos dado que el peso de una variable es constante.

- $o = t\,u$.

  1. Luego, $|o|_l^{\tau[\alpha:=l']} = |t|_{|u|_{[]}^{\tau[\alpha:=l']}:l}^{\tau[\alpha:=l']} + |u|_{[]}^{\tau[\alpha:=l']}$ con $\alpha \notin \mathsf{fn}(t)$ y $\alpha \notin \mathsf{fn}(u)$. Por *h.i.* (1) se tienen $|u|_{[]}^{\tau[\alpha:=l']} = |u|_{[]}^{\tau}$ y $|t|_{|u|_{[]}^{\tau[\alpha:=l']}:l}^{\tau[\alpha:=l']} = |t|_{|u|_{[]}^{\tau[\alpha:=l']}:l}^{\tau}$. Más aún, $|t|_{|u|_{[]}^{\tau[\alpha:=l']}:l}^{\tau} = |t|_{|u|_{[]}^{\tau}:l}^{\tau}$. Se concluye entonces, $|t\,u|_l^{\tau[\alpha:=l']} = |t|_{|u|_{[]}^{\tau[\alpha:=l']}:l}^{\tau[\alpha:=l']} + |u|_{[]}^{\tau[\alpha:=l']} = |t|_{|u|_{[]}^{\tau}:l}^{\tau} + |u|_{[]}^{\tau} = |t\,u|_l^{\tau}$.
  2. Luego, $|o|_l^{\tau} = |t|_{|u|_{[]}^{\tau}:l}^{\tau} + |u|_{[]}^{\tau}$. Por *h.i.* (2) se tienen $|t|_{|u|_{[]}^{\tau}:l}^{\tau} \geq |t|_{|u|_{[]}^{\tau'}:l}^{\tau'}$ y $|u|_{[]}^{\tau} \geq |u|_{[]}^{\tau'}$. Más aún, se tiene entonces $|u|_{[]}^{\tau} : l \geq |u|_{[]}^{\tau'} : l$ y, por *h.i.* (3), $|t|_{|u|_{[]}^{\tau}:l}^{\tau'} \geq |t|_{|u|_{[]}^{\tau'}:l}^{\tau'}$. Se concluye entonces, $|t\,u|_l^{\tau} = |t|_{|u|_{[]}^{\tau}:l}^{\tau} + |u|_{[]}^{\tau} \geq |t|_{|u|_{[]}^{\tau}:l}^{\tau'} + |u|_{[]}^{\tau} \geq |t|_{|u|_{[]}^{\tau'}:l}^{\tau'} + |u|_{[]}^{\tau'} = |t\,u|_l^{\tau'}$.
  3. Luego, $|o|_l^{\tau} = |t|_{|u|_{[]}^{\tau}:l}^{\tau} + |u|_{[]}^{\tau}$. Más aún, $l \geq l'$ implica $|u|_{[]}^{\tau} : l \geq |u|_{[]}^{\tau} : l'$ y, por *h.i.* (3) se tiene $|t|_{|u|_{[]}^{\tau}:l}^{\tau} \geq |t|_{|u|_{[]}^{\tau}:l'}^{\tau}$. Se concluye entonces, $|t\,u|_l^{\tau} = |t|_{|u|_{[]}^{\tau}:l}^{\tau} + |u|_{[]}^{\tau} \geq |t|_{|u|_{[]}^{\tau}:l'}^{\tau} + |u|_{[]}^{\tau} = |t\,u|_{l'}^{\tau}$.

- $o = \lambda x.t$.

  1. Luego, $|o|_l^{\tau[\alpha:=l']} = |t|_{l_0}^{\tau[\alpha:=l']}$ con $l = n : l_0$ o $l = [] = l_0$, y $\alpha \notin \mathsf{fn}(t)$. Por *h.i.* (1) se tiene $|t|_{l_0}^{\tau[\alpha:=l']} = |t|_{l_0}^{\tau}$. Se concluye entonces, $|\lambda x.t|_l^{\tau[\alpha:=l']} = |t|_{l_0}^{\tau[\alpha:=l']} = |t|_{l_0}^{\tau} = |\lambda x.t|_l^{\tau}$.
  2. Luego, $|o|_l^{\tau} = |t|_{l_0}^{\tau}$ con $l = n : l_0$ o $l = [] = l_0$. Por *h.i.* (2) se tiene $|t|_{l_0}^{\tau} \geq |t|_{l_0}^{\tau'}$. Se concluye entonces, $|\lambda x.t|_l^{\tau} = |t|_{l_0}^{\tau} \geq |t|_{l_0}^{\tau'} = |\lambda x.t|_l^{\tau'}$.
  3. Luego, $|o|_l^{\tau} = |t|_{l_0}^{\tau}$ con $l = n : l_0$ o $l = [] = l_0$. Si $l = []$, entonces $l \geq l'$ implica $l' = [] = l_0$ también, por lo que se concluye inmediatamente $|\lambda x.t|_l^{\tau} = |t|_{l_0}^{\tau} = |\lambda x.t|_{l'}^{\tau}$. Si no (*i.e.* $l = n : l_0$), de la hipótesis $l \geq l'$ se tiene $l' = n' : l_0'$ con $n \geq n'$ y $l_0 \geq l_0'$ por definición. Luego, por *h.i.* (3), $|t|_{l_0}^{\tau} \geq |t|_{l_0'}^{\tau}$. Se concluye entonces, $|\lambda x.t|_l^{\tau} = |t|_{l_0}^{\tau} \geq |t|_{l_0'}^{\tau} = |\lambda x.t|_{l'}^{\tau}$.

- $o = \mu\gamma.c$.

  1. Luego, $|o|_l^{\tau[\alpha:=l']} = |c|_{[]}^{\tau[\alpha:=l'][\gamma:=l]} = |c|_{[]}^{\tau[\gamma:=l][\alpha:=l']}$ dado que, por α-conversión, $\gamma \neq \alpha$. Más aún, $\alpha \notin \mathsf{fn}(o)$ y $\gamma \neq \alpha$ implican $\alpha \notin \mathsf{fn}(c)$. Por *h.i.* (1) se tiene $|c|_{[]}^{\tau[\gamma:=l][\alpha:=l']} = |c|_{[]}^{\tau[\gamma:=l]}$. Se concluye entonces, $|\mu\gamma.c|_l^{\tau[\alpha:=l']} = |c|_{[]}^{\tau[\alpha:=l'][\gamma:=l]} = |c|_{[]}^{\tau[\gamma:=l]} = |\mu\gamma.c|_l^{\tau}$.
  2. Luego, $|o|_l^{\tau} = |c|_{[]}^{\tau[\gamma:=l]}$. Más aún, $\tau \geq \tau'$ implica $\tau[\gamma := l] \geq \tau'[\gamma := l]$. Por *h.i.* (2) se tiene $|c|_{[]}^{\tau[\gamma:=l]} \geq |c|_{[]}^{\tau'[\gamma:=l]}$. Se concluye entonces, $|\mu\gamma.c|_l^{\tau} = |c|_{[]}^{\tau[\gamma:=l]} \geq |c|_{[]}^{\tau'[\gamma:=l]} = |\mu\gamma.c|_l^{\tau'}$.
  3. Luego, $|o|_l^{\tau} = |c|_{[]}^{\tau[\gamma:=l]}$. Más aún, $l \geq l'$ implica $\tau[\gamma := l] \geq \tau[\gamma := l']$. Por *h.i.* (2) se tiene $|c|_{[]}^{\tau[\gamma:=l]} \geq |c|_{[]}^{\tau[\gamma:=l']}$. Se concluye entonces, $|\mu\gamma.c|_l^{\tau} = |c|_{[]}^{\tau[\gamma:=l]} \geq |c|_{[]}^{\tau[\gamma:=l']} = |\mu\gamma.c|_{l'}^{\tau}$.

- $o = t[x \setminus u]$.

  1. Luego, $|o|_l^{\tau[\alpha:=l']} = |t|_l^{\tau[\alpha:=l']}$ con $\alpha \notin \mathsf{fn}(t)$. Por *h.i.* (1) se tiene $|t|_l^{\tau[\alpha:=l']} = |t|_l^{\tau}$. Se concluye entonces, $|t[x \setminus u]|_l^{\tau[\alpha:=l']} = |t|_l^{\tau[\alpha:=l']} = |t|_l^{\tau} = |t[x \setminus u]|_l^{\tau}$.
  2. Luego, $|o|_l^{\tau} = |t|_l^{\tau}$. Por *h.i.* (2) se tiene $|t|_l^{\tau} \geq |t|_l^{\tau'}$. Se concluye entonces, $|t[x \setminus u]|_l^{\tau} = |t|_l^{\tau} \geq |t|_l^{\tau'} = |t[x \setminus u]|_l^{\tau'}$.
  3. Luego, $|o|_l^{\tau} = |t|_l^{\tau}$. Por *h.i.* (3) se tiene $|t|_l^{\tau} \geq |t|_{l'}^{\tau}$. Se concluye entonces, $|t[x \setminus u]|_l^{\tau} = |t|_l^{\tau} \geq |t|_{l'}^{\tau} = |t[x \setminus u]|_{l'}^{\tau}$.

- $o = [\delta]\, t$.

  1. Luego, $|o|_l^{\tau[\alpha:=l']} = |t|_{\tau[\alpha:=l'](\delta)}^{\tau[\alpha:=l']} + 1 + \sum \tau[\alpha := l'](\delta)$ con $\delta \neq \alpha$ y $\alpha \notin \mathsf{fn}(t)$. Más aún, $\delta \neq \alpha$ implica $\tau[\alpha := l'](\delta) = \tau(\delta)$. Por *h.i.* (1) se tiene $|t|_{\tau[\alpha:=l'](\delta)}^{\tau[\alpha:=l']} = |t|_{\tau[\alpha:=l'](\delta)}^{\tau} = |t|_{\tau(\delta)}^{\tau}$. Se concluye entonces, $|[\delta]\, t|_l^{\tau[\alpha:=l']} = |t|_{\tau[\alpha:=l'](\delta)}^{\tau[\alpha:=l']} + 1 + \sum \tau[\alpha := l'](\delta) = |t|_{\tau(\delta)}^{\tau} + 1 + \sum \tau(\delta) = |[\delta]\, t|_l^{\tau}$.
  2. Luego, $|o|_l^{\tau} = |t|_{\tau(\delta)}^{\tau} + 1 + \sum \tau(\delta)$. Por *h.i.* (2), se tiene $|t|_{\tau(\delta)}^{\tau} \geq |t|_{\tau(\delta)}^{\tau'}$. Más aún, por hipótesis $\tau(\delta) \geq \tau'(\delta)$, por lo que $\sum \tau(\delta) \geq \sum \tau'(\delta)$ y, por *h.i.* (3), $|t|_{\tau(\delta)}^{\tau'} \geq |t|_{\tau'(\delta)}^{\tau'}$. Se concluye entonces, $|[\delta]\, t|_l^{\tau} = |t|_{\tau(\delta)}^{\tau} + 1 + \sum \tau(\delta) \geq |t|_{\tau'(\delta)}^{\tau'} + 1 + \sum \tau'(\delta) = |[\delta]\, t|_l^{\tau'}$.
  3. Luego, $|o|_l^{\tau} = |t|_{\tau(\delta)}^{\tau} + 1 + \sum \tau(\delta) = |o|_{l'}^{\tau}$ por lo que el resultado es inmediato.

- $o = c[\![\gamma \setminus^{\delta} s]\!]$.

  1. Luego, $|o|_l^{\tau[\alpha:=l']} = |c|_{[]}^{\tau[\alpha:=l'][\gamma:=|u_0|_{[]}^{\tau[\alpha:=l']}:\ldots:|u_n|_{[]}^{\tau[\alpha:=l']}:\tau[\alpha:=l'](\delta)]}$ con $s = u_0 \cdot \ldots \cdot \cdot\, u_n$, $\alpha \neq \delta$, $\alpha \neq \mathsf{fn}(c)$ y $\alpha \neq \mathsf{fn}(s)$. Más aún, por α-conversión se asume $\gamma \neq \alpha$, por lo que $\alpha \notin \mathsf{fn}(o)$ implica $\alpha \notin \mathsf{fn}(c)$ y $\alpha \notin \mathsf{fn}(s)$ (*i.e.* $(\alpha \notin \mathsf{fn}(u_i))_{i<n}$). Por *h.i.* (1) se tienen $(|u_i|_{[]}^{\tau[\alpha:=l']} = |u_i|_{[]}^{\tau})_{i\leq n}$ y

  $$\begin{array}{rcl} |c|_{[]}^{\tau[\alpha:=l'][\gamma:=|u_0|_{[]}^{\tau[\alpha:=l']}:\ldots:|u_n|_{[]}^{\tau[\alpha:=l']}:\tau[\alpha:=l'](\delta)]} & = & |c|_{[]}^{\tau[\gamma:=|u_0|_{[]}^{\tau}:\ldots:|u_n|_{[]}^{\tau}:\tau[\alpha:=l'](\delta)][\alpha:=l']} \\ & = & |c|_{[]}^{\tau[\gamma:=|u_0|_{[]}^{\tau}:\ldots:|u_n|_{[]}^{\tau}:\tau[\alpha:=l'](\delta)]} \end{array}$$

  Notar que $\delta \neq \alpha$ implica $\tau[\alpha := l'](\delta) = \tau(\delta)$. Se concluye entonces, $|c[\![\gamma \setminus^{\delta} s]\!]|_l^{\tau[\alpha:=l']} = |c|_{[]}^{\tau[\alpha:=l'][\gamma:=|u_0|_{[]}^{\tau[\alpha:=l']}:\ldots:|u_n|_{[]}^{\tau[\alpha:=l']}:\tau[\alpha:=l'](\delta)]} = |c|_{[]}^{\tau[\gamma:=|u_0|_{[]}^{\tau}:\ldots:|u_n|_{[]}^{\tau}:\tau(\delta)]} = |c[\![\gamma \setminus^{\delta} s]\!]|_l^{\tau}$.
  2. Luego, $|o|_l^{\tau} = |c|_{[]}^{\tau[\gamma:=|u_0|_{[]}^{\tau}:\ldots:|u_n|_{[]}^{\tau}:\tau(\delta)]}$. Por *h.i.* (2) se tiene $(|u_i|_{[]}^{\tau} = |u_i|_{[]}^{\tau'})_{i\leq n}$. Más aún, por hipótesis $\tau(\delta) \geq \tau'(\delta)$, por lo que $l_0 = |u_0|_{[]}^{\tau} : \ldots : |u_n|_{[]}^{\tau} : \tau(\delta) \geq |u_0|_{[]}^{\tau'} : \ldots : |u_n|_{[]}^{\tau'} : \tau'(\delta) = l_0'$ y, por lo tanto, $\tau[\gamma := l_0] \geq \tau'[\gamma := l_0']$. Luego, por *h.i.* (2), $|c|_{[]}^{\tau[\gamma:=l_0]} \geq |c|_{[]}^{\tau'[\gamma:=l_0']}$. Se concluye entonces, $|c[\![\gamma \setminus^{\delta} s]\!]|_l^{\tau} = |c|_{[]}^{\tau[\gamma:=l_0]} \geq |c|_{[]}^{\tau'[\gamma:=l_0']} = |c[\![\gamma \setminus^{\delta} s]\!]|_l^{\tau'}$.
  3. Luego, $|o|_l^{\tau} = |c|_{[]}^{\tau[\gamma:=|u_0|_{[]}^{\tau}:\ldots:|u_n|_{[]}^{\tau}:\tau(\delta)]} = |o|_{l'}^{\tau}$ por lo que el resultado es inmediato.

- $o = c[\gamma \setminus \delta]$.

  1. Luego, $|o|_l^{\tau[\alpha:=l']} = |c|_{[]}^{\tau[\alpha:=l'][\gamma:=\tau[\alpha:=l'](\delta)]} + 1$ con $\alpha \neq \delta$ y $\alpha \neq \mathsf{fn}(c)$. Más aún, por α-conversión se asume $\gamma \neq \alpha$, por lo que $\alpha \notin \mathsf{fn}(o)$ implica $\alpha \notin \mathsf{fn}(c)$. Por *h.i.* (1) se tienen $|c|_{[]}^{\tau[\alpha:=l'][\gamma:=\tau[\alpha:=l'](\delta)]} = |c|_{[]}^{\tau[\gamma:=\tau[\alpha:=l'](\delta)][\alpha:=l']} = |c|_{[]}^{\tau[\gamma:=\tau[\alpha:=l'](\delta)]}$. Notar que $\delta \neq \alpha$ implica $\tau[\alpha := l'](\delta) = \tau(\delta)$. Se concluye entonces, $|c[\gamma \setminus \delta]|_l^{\tau[\alpha:=l']} = |c|_{[]}^{\tau[\alpha:=l'][\gamma:=\tau[\alpha:=l'](\delta)]} + 1 = |c|_{[]}^{\tau[\gamma:=\tau(\delta)]} + 1 = |c[\gamma \setminus \delta]|_l^{\tau}$.
  2. Luego, $|o|_l^{\tau} = |c|_{[]}^{\tau[\gamma:=\tau(\delta)]} + 1$. Más aún, por hipótesis $\tau(\delta) \geq \tau'(\delta)$, por lo que $\tau[\gamma := \tau(\delta)] \geq \tau'[\gamma := \tau'(\delta)]$ y, por *h.i.* (2), se tiene $|c|_{[]}^{\tau[\gamma:=\tau(\delta)]} \geq |c|_{[]}^{\tau'[\gamma:=\tau'(\delta)]}$. Se concluye entonces, $|c[\gamma \setminus \delta]|_l^{\tau} = |c|_{[]}^{\tau[\gamma:=\tau(\delta)]} + 1 \geq |c|_{[]}^{\tau'[\gamma:=\tau'(\delta)]} + 1 = |[\delta]\, t|_l^{\tau'}$.
  3. Luego, $|o|_l^{\tau} = |c|_{[]}^{\tau[\gamma:=\tau(\delta)]} + 1 = |o|_{l'}^{\tau}$ por lo que el resultado es inmediato.

- $o = t \cdot s$.
  1. Luego, $|o|_l^{\tau[\alpha:=l']} = |t|_{[]}^{\tau[\alpha:=l']} + |s|_{[]}^{\tau[\alpha:=l']}$ con $\alpha \notin \mathsf{fn}(t)$ y $\alpha \notin \mathsf{fn}(s)$. Por *h.i.* (1) se tienen $|t|_{[]}^{\tau[\alpha:=l']} = |t|_{[]}^{\tau}$ y $|s|_{[]}^{\tau[\alpha:=l']} = |s|_{[]}^{\tau}$. Se concluye entonces, $|t \cdot s|_l^{\tau[\alpha:=l']} = |t|_{[]}^{\tau[\alpha:=l']} + |s|_{[]}^{\tau[\alpha:=l']} = |t|_{[]}^{\tau} + |s|_{[]}^{\tau} = |t \cdot s|_l^{\tau}$.
  2. Luego, $|o|_l^{\tau} = |t|_{[]}^{\tau} + |s|_{[]}^{\tau}$. Por *h.i.* (2) se tienen $|t|_{[]}^{\tau} \geq |t|_{[]}^{\tau'}$ y $|s|_{[]}^{\tau} \geq |s|_{[]}^{\tau'}$. Se concluye entonces, $|t \cdot s|_l^{\tau} = |t|_{[]}^{\tau} + |s|_{[]}^{\tau} \geq |t|_{[]}^{\tau'} + |s|_{[]}^{\tau'} = |t \cdot s|_l^{\tau'}$.
  3. Luego, $|o|_l^{\tau} = |t|_{[]}^{\tau} + |s|_{[]}^{\tau} = |o|_{l'}^{\tau}$ por lo que el resultado es inmediato.

□

Lema C.0.30. *Sea $t \in \mathbb{T}_{\Lambda M}$. Luego, $|\mathtt{L}\langle t\rangle|_l^{\tau} = |t|_l^{\tau}$.*

Demostración. Por inducción en $\mathtt{L}$.

- $\mathtt{L} = \Box$. Luego, $\mathtt{L}\langle t\rangle = t$ y el resultado es inmediato.
- $\mathtt{L} = \mathtt{L}'[x \setminus u]$. Luego, $|\mathtt{L}\langle t\rangle|_l^{\tau} = |\mathtt{L}'\langle t\rangle[x \setminus u]|_l^{\tau} = |\mathtt{L}'\langle t\rangle|_l^{\tau}$ por definición. Se concluye por *h.i.*

□

Lema C.0.31. *Sea $t \in \mathbb{T}_{\Lambda M}$ y $s = u_0 \cdot \ldots \cdot u_n$ un stack. Luego, $|t :: s|_l^{\tau} = |t|_{|u_0|_{[]}^{\tau}:\ldots:|u_n|_{[]}^{\tau}:l}^{\tau} + |s|_{[]}^{\tau}$.*

Demostración. Por inducción en n.

- $n = 0$. Luego, $t :: s = t\, u_0$ y, por definición, $|t\, u_0|_l^{\tau} = |t|_{|u_0|_{[]}^{\tau}:l}^{\tau} + |u_0|_{[]}^{\tau}$, por lo que se concluye.
- $n > 0$. Luego, $s = u_0 \cdot s'$ con $t :: s = (t\, u_0) :: s'$ y $s' = u_1 \cdot \ldots \cdot u_n$. Por *h.i.* se tiene $|(t\, u_0) :: s'|_l^{\tau} = |t\, u_0|_{|u_1|_{[]}^{\tau}:\ldots:|u_n|_{[]}^{\tau}:l}^{\tau} + |s'|_{[]}^{\tau} = |t|_{|u_0|_{[]}^{\tau}:\ldots:|u_n|_{[]}^{\tau}:l}^{\tau} + |u_0|_{[]}^{\tau} + |s'|_{[]}^{\tau} = |t|_{|u_0|_{[]}^{\tau}:\ldots:|u_n|_{[]}^{\tau}:l}^{\tau} + |s|_{[]}^{\tau}$, por lo que se concluye.

□

Lema C.0.32 (Replacement)**.** *Sea $o \in \mathbb{O}_{\Lambda M}$ y $s = u_0 \cdot \ldots \cdot u_n$ un stack. Luego, $|\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o|_l^{\tau} = |o|_l^{\tau[\alpha:=|u_0|_{[]}^{\tau}:\ldots:|u_n|_{[]}^{\tau}:\tau(\alpha')]}$.*

Demostración. Por inducción en o. Sea $\tau' = \tau[\alpha := |u_0|_{[]}^{\tau} : \ldots : |u_n|_{[]}^{\tau} : \tau(\alpha')]$.

- $o = x$. Luego, $\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o = x$ y $|x|_l^{\tau} = 0 = |x|_l^{\tau'}$, por lo que se concluye.
- $o = t\, u$. Luego, $\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o = \{\!\{\alpha \setminus^{\alpha'} s\}\!\} t\, \{\!\{\alpha \setminus^{\alpha'} s\}\!\} u$. Por *h.i.* sobre u se tiene $|\{\!\{\alpha \setminus^{\alpha'} s\}\!\} u|_{[]}^{\tau} = |u|_{[]}^{\tau'}$. Por *h.i.* sobre t se tiene entonces $|\{\!\{\alpha \setminus^{\alpha'} s\}\!\} t|_{|\{\!\{\alpha \setminus^{\alpha'} s\}\!\} u|_{[]}^{\tau}:l}^{\tau} = |t|_{|\{\!\{\alpha \setminus^{\alpha'} s\}\!\} u|_{[]}^{\tau}:l}^{\tau'} = |t|_{|u|_{[]}^{\tau'}:l}^{\tau'}$. Se concluye dado que $|\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o|_l^{\tau} = |\{\!\{\alpha \setminus^{\alpha'} s\}\!\} t|_{|\{\!\{\alpha \setminus^{\alpha'} s\}\!\} u|_{[]}^{\tau}:l}^{\tau} + |\{\!\{\alpha \setminus^{\alpha'} s\}\!\} u|_{[]}^{\tau} = |t|_{|u|_{[]}^{\tau'}:l}^{\tau'} + |u|_{[]}^{\tau'} = |o|_l^{\tau'}$.
- $o = \lambda x.t$. Luego, $\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o = \lambda x.\{\!\{\alpha \setminus^{\alpha'} s\}\!\} t$ con $x \notin s$. Sea l_0 tal que $l = n : l_0$ o $l = [] = l_0$. Por *h.i.* se tiene $|\{\!\{\alpha \setminus^{\alpha'} s\}\!\} t|_{l_0}^{\tau} = |t|_{l_0}^{\tau'}$. Se concluye dado que $|\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o|_l^{\tau} = |\{\!\{\alpha \setminus^{\alpha'} s\}\!\} t|_{l_0}^{\tau} = |t|_{l_0}^{\tau'} = |o|_l^{\tau'}$.
- $o = \mu\gamma.c$. Luego, $\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o = \mu\gamma.\{\!\{\alpha \setminus^{\alpha'} s\}\!\} c$ con $\gamma \notin s$. Más aún, por α-conversión se asume $\gamma \neq \alpha$. Por *h.i.* se tiene entonces $|\{\!\{\alpha \setminus^{\alpha'} s\}\!\} c|_{[]}^{\tau[\gamma:=l]} = |c|_{[]}^{\tau[\gamma:=l][\alpha:=|u_0|_{[]}^{\tau}:\ldots:|u_n|_{[]}^{\tau}:\tau(\alpha')]} |c|_{[]}^{\tau'[\gamma:=l]}$. Se concluye dado que $|\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o|_l^{\tau} = |\{\!\{\alpha \setminus^{\alpha'} s\}\!\} c|_{[]}^{\tau[\gamma:=l]} + 1 = |c|_{[]}^{\tau'[\gamma:=l]} + 1 = |o|_l^{\tau'}$.

- $o = t[x \setminus u]$. Luego, $\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o = \{\!\{\alpha \setminus^{\alpha'} s\}\!\} t[x \setminus \{\!\{\alpha \setminus^{\alpha'} s\}\!\} u]$ con $x \notin s$. Por *h.i.* se tiene $|\{\!\{\alpha \setminus^{\alpha'} s\}\!\} t|_l^\tau = |t|_l^{\tau'}$. Se concluye dado que $|\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o|_l^\tau = |\{\!\{\alpha \setminus^{\alpha'} s\}\!\} t|_l^\tau = |t|_l^{\tau'} = |o|_l^{\tau'}$.

- $o = [\alpha]\, t$. Luego, $\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o = [\alpha']\, (\{\!\{\alpha \setminus^{\alpha'} s\}\!\} t) :: s$. Por Lem. C.0.31, se tiene $|\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o|_l^\tau = |(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} t) :: s|^\tau_{\tau(\alpha')} + 1 + \sum \tau(\alpha') = |\{\!\{\alpha \setminus^{\alpha'} s\}\!\} t|^\tau_{\tau'(\alpha)} + |s|^\tau_{[]} + 1 + \sum \tau(\alpha')$. Por *h.i.* se tiene $|\{\!\{\alpha \setminus^{\alpha'} s\}\!\} t|^\tau_{\tau'(\alpha)} = |t|^{\tau'}_{\tau'(\alpha)}$. Se concluye dado que $|\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o|_l^\tau = |\{\!\{\alpha \setminus^{\alpha'} s\}\!\} t|^\tau_{\tau'(\alpha)} + |s|^\tau_{[]} + 1 + \sum \tau(\alpha') = |t|^{\tau'}_{\tau'(\alpha)} + 1 + \sum \tau'(\alpha) = |o|_l^{\tau'}$.

- $o = [\delta]\, t$ con $\delta \neq \alpha$. Luego, $\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o = [\delta]\, \{\!\{\alpha \setminus^{\alpha'} s\}\!\} t$. Más aún, $\delta \neq \alpha$ implica $\tau(\delta) = \tau'(\delta)$. Por *h.i.* se tiene entonces $|\{\!\{\alpha \setminus^{\alpha'} s\}\!\} t|^\tau_{\tau(\delta)} = |t|^{\tau'}_{\tau'(\delta)}$. Se concluye dado que $|\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o|_l^\tau = |\{\!\{\alpha \setminus^{\alpha'} s\}\!\} t|^\tau_{\tau(\delta)} + 1 + \sum \tau(\delta) = |t|^{\tau'}_{\tau'(\delta)} + 1 + \sum \tau'(\delta) = |o|_l^{\tau'}$.

- $o = c[\![\gamma \setminus^\alpha s']\!]$. Luego, $\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o = (\{\!\{\alpha \setminus^{\alpha'} s\}\!\} c)[\![\gamma \setminus^{\alpha'} \{\!\{\alpha \setminus^{\alpha'} s\}\!\} s' \cdot s]\!]$ con $\gamma \notin s$, $\gamma \neq \alpha$ y $\gamma \neq \alpha'$. Sea $s' = v_0 \cdot \ldots \cdot v_m$. Por *h.i.* sobre s' se tiene $|\{\!\{\alpha \setminus^{\alpha'} s\}\!\} s'|^\tau_{[]} = |s'|^{\tau'}_{[]}$, lo que implica $(|\{\!\{\alpha \setminus^{\alpha'} s\}\!\} v_j|^\tau_{[]} = |v_j|^{\tau'}_{[]})_{j \leq m}$. Más aún, dado que $\gamma \neq \alpha$, por *h.i.* sobre c se tiene $|\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o|_l^\tau = |\{\!\{\alpha \setminus^{\alpha'} s\}\!\} c|^{\tau[\gamma := |v_0|^{\tau'}_{[]} :\ldots: |v_m|^{\tau'}_{[]} : |u_0|^\tau_{[]} :\ldots: |u_n|^\tau_{[]} : \tau(\alpha')]}_{[]} = |\{\!\{\alpha \setminus^{\alpha'} s\}\!\} c|^{\tau'[\gamma := |v_0|^{\tau'}_{[]} :\ldots: |v_m|^{\tau'}_{[]} : \tau'(\alpha)]}_{[]} = |o|_l^{\tau'}$, por lo que se concluye.

- $o = c[\![\gamma \setminus^\delta s']\!]$ con $\delta \neq \alpha$. Luego, $\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o = (\{\!\{\alpha \setminus^{\alpha'} s\}\!\} c)[\![\gamma \setminus^\delta \{\!\{\alpha \setminus^{\alpha'} s\}\!\} s']\!]$ con $\gamma \notin s$, $\gamma \neq \alpha$ y $\gamma \neq \alpha'$. Sea $s' = v_0 \cdot \ldots \cdot v_m$. Por *h.i.* sobre s' se tiene $|\{\!\{\alpha \setminus^{\alpha'} s\}\!\} s'|^\tau_{[]} = |s'|^{\tau'}_{[]}$, lo que implica $(|\{\!\{\alpha \setminus^{\alpha'} s\}\!\} v_j|^\tau_{[]} = |v_j|^{\tau'}_{[]})_{j \leq m}$. Más aún, dado que $\gamma \neq \alpha$, por *h.i.* sobre c se tiene $|\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o|_l^\tau = |\{\!\{\alpha \setminus^{\alpha'} s\}\!\} c|^{\tau[\gamma := |v_0|^{\tau'}_{[]} :\ldots: |v_m|^{\tau'}_{[]} : l]}_{[]} = |c|^{\tau'[\gamma := |v_0|^{\tau'}_{[]} :\ldots: |v_m|^{\tau'}_{[]} : l]}_{[]} = |o|_l^\tau$, por lo que se concluye.

- $o = c[\gamma \setminus \alpha]$. Luego, $\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o = (\{\!\{\alpha \setminus^{\alpha'} s\}\!\} c)[\![\gamma \setminus^\delta s]\!][\delta \setminus \alpha']$ con $\gamma \notin s$, $\gamma \neq \alpha$, $\gamma \neq \alpha'$ y δ un nombre fresco. Por Lem. C.0.29 (1) con $\delta \notin s$, se tiene $|s|^{\tau[\delta := \tau(\alpha')]}_{[]} = |s|^\tau_{[]}$, lo que implica $(|u_i|^{\tau[\delta := \tau(\alpha')]}_{[]} = |u_i|^\tau_{[]})_{i \leq n}$. Por Lem. C.0.29 (1), ahora con $\delta \notin c$, se tiene $|\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o|_l^\tau = |(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} c)[\![\gamma \setminus^\delta s]\!]|^{\tau[\delta := \tau(\alpha')]}_{[]} + 1 = |\{\!\{\alpha \setminus^{\alpha'} s\}\!\} c|^{\tau[\delta := \tau(\alpha')][\gamma := \tau'(\alpha)]}_{[]} + 1 = |\{\!\{\alpha \setminus^{\alpha'} s\}\!\} c|^{\tau[\gamma := \tau'(\alpha)]}_{[]} + 1$. Finalmente, por *h.i.* se tiene $|\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o|_l^\tau = |\{\!\{\alpha \setminus^{\alpha'} s\}\!\} c|^{\tau[\gamma := \tau'(\alpha)]}_{[]} + 1 = |c|^{\tau'[\gamma := \tau'(\alpha)]}_{[]} + 1 = |o|_l^{\tau'}$, por lo que se concluye.

- $o = c[\gamma \setminus \delta]$ con $\delta \neq \alpha$. Luego, $\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o = (\{\!\{\alpha \setminus^{\alpha'} s\}\!\} c)[\gamma \setminus \delta]$ con $\gamma \notin s$, $\gamma \neq \alpha$ y $\gamma \neq \alpha'$. Más aún, $\delta \neq \alpha$ implica $\tau(\delta) = \tau'(\delta)$. Entonces, dado que $\gamma \neq \alpha$, por *h.i.* se tiene $|\{\!\{\alpha \setminus^{\alpha'} s\}\!\} c|^{\tau[\gamma := \tau(\delta)]}_{[]} = |c|^{\tau'[\gamma := \tau'(\delta)]}_{[]}$. Se concluye dado que $|\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o|_l^\tau = |\{\!\{\alpha \setminus^{\alpha'} s\}\!\} c|^{\tau[\gamma := \tau(\delta)]}_{[]} + 1 = |c|^{\tau'[\gamma := \tau'(\delta)]}_{[]} + 1 = |o|_l^{\tau'}$.

- $o = t \cdot s'$. Luego, $\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o = \{\!\{\alpha \setminus^{\alpha'} s\}\!\} t \cdot \{\!\{\alpha \setminus^{\alpha'} s\}\!\} s'$. Por *h.i.* se tienen $|\{\!\{\alpha \setminus^{\alpha'} s\}\!\} t|^\tau_{[]} = |t|^{\tau'}_{[]}$ y $|\{\!\{\alpha \setminus^{\alpha'} s\}\!\} s'|^\tau_{[]} = |s'|^{\tau'}_{[]}$. Se concluye dado que $|\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o|_l^\tau = |\{\!\{\alpha \setminus^{\alpha'} s\}\!\} t|^\tau_{[]} + |\{\!\{\alpha \setminus^{\alpha'} s\}\!\} s'|^\tau_{[]} = |t|^{\tau'}_{[]} + |s'|^{\tau'}_{[]} = |o|_l^{\tau'}$.

□

Corolario C.0.33. *Sea $c \in \mathbb{C}_{\Lambda M}$ y $s = u_0 \cdot \ldots \cdot u_n$ un stack. Luego, $|c[\![\alpha \setminus^{\alpha'} s]\!]|_l^\tau = |\{\!\{\alpha \setminus^{\alpha'} s\}\!\} c|_l^\tau$.*

Demostración. Notar que $|\_|_l^\tau$ ignora la lista l en el caso de los comandos, por lo que $|c|_l^\tau = |c|_{l'}^\tau$ para listas cualesquiera l y l'. Luego, $|c[\![\alpha \setminus^{\alpha'} s]\!]|_l^\tau = |c|^{\tau[\alpha := |u_0|^\tau_{[]} :\ldots: |u_n|^\tau_{[]} : \tau(\alpha')]}_{[]} = |\{\!\{\alpha \setminus^{\alpha'} s\}\!\} c|^\tau_{[]} = |\{\!\{\alpha \setminus^{\alpha'} s\}\!\} c|_l^\tau$. □

La propiedad principal de la función de peso es la preservación del orden bajo reducción canónica.

Lema C.0.34. *Sea $o \in \mathbb{O}_{\Lambda M}$. Si $o \to_{\mathfrak{C}} o'$, entonces $|o|_l^\tau \geq |o'|_l^\tau$.*

Demostración. Por definición, $o \to_{\mathfrak{C}} o'$ implica $o = \mathtt{O}\langle l \rangle$ y $o' = \mathtt{O}\langle r \rangle$ con $l \mapsto_* r$, $* \in \{\mathtt{dB}, \mathtt{dM}, \mathtt{N}, \mathtt{P}, \mathtt{W}\}$. La demostración es por inducción en $\mathtt{O}$.

- $\mathtt{O} = \Box$. Luego, se tienen dos casos posibles según la regla de reescritura aplicada al reducir.
  1. $\mathtt{dB}$. Luego, $o = \mathtt{L}\langle \lambda x.t \rangle\, u$ y $o' = \mathtt{L}\langle t[x \setminus u] \rangle$. Por Lem. C.0.30, $|o|_l^\tau = |\mathtt{L}\langle \lambda x.t \rangle|^\tau_{|u|^\tau_{[]}:l} + |u|^\tau_{[]} = |\lambda x.t|^\tau_{|u|^\tau_{[]}:l} + |u|^\tau_{[]} = |t|_l^\tau + |u|^\tau_{[]}$. Se concluye apelando nuevamente al Lem. C.0.30, dado que $|o|_l^\tau = |t|_l^\tau + |u|^\tau_{[]} \geq |t|_l^\tau = |t[x \setminus u]|_l^\tau = |o'|_l^\tau$.
  2. $\mathtt{dM}$. Luego, $o = \mathtt{L}\langle \mu\alpha.c \rangle\, u$ y $o' = \mathtt{L}\langle \mu\alpha'.c[\![\alpha \setminus^{\alpha'} u]\!] \rangle$ con α' un nombre fresco. Por Lem. C.0.30, $|o|_l^\tau = |\mathtt{L}\langle \mu\alpha.c \rangle|^\tau_{|u|^\tau_{[]}:l} + |u|^\tau_{[]} = |\mu\alpha.c|^\tau_{|u|^\tau_{[]}:l} + |u|^\tau_{[]} = |c|^{\tau[\alpha:=|u|^\tau_{[]}:l]}_{[]} + 1 + |u|^\tau_{[]}$. Más aún, por Lem. C.0.29 (1) con α' fresco, se tiene $|c|^{\tau[\alpha:=|u|^\tau_{[]}:l]}_{[]} = |c|^{\tau[\alpha':=l][\alpha:=|u|^{\tau[\alpha':=l]}_{[]}:\tau[\alpha':=l](\alpha')]}_{[]}$. Se concluye apelando nuevamente al Lem. C.0.30, dado que $|o|_l^\tau = |c|^{\tau[\alpha:=|u|^\tau_{[]}:l]}_{[]} + 1 + |u|^\tau_{[]} \geq |c|^{\tau[\alpha':=l][\alpha:=|u|^{\tau[\alpha':=l]}_{[]}:\tau[\alpha':=l](\alpha')]}_{[]} + 1 = |c[\![\alpha \setminus^{\alpha'} u]\!]|^{\tau[\alpha':=l]}_{[]} + 1 = |\mu\alpha'.c[\![\alpha \setminus^{\alpha'} u]\!]|_l^\tau = |o'|_l^\tau$.
- $\mathtt{O} = \boxdot$. Luego, se tienen tres casos posibles según la regla de reescritura aplicada al reducir.
  1. $\mathtt{N}$. Luego, $o = \mathtt{LCC}\langle [\alpha]\, t \rangle [\![\alpha \setminus^{\alpha'} s]\!]$ y $o' = \mathtt{LCC}\langle [\alpha']\, t :: s \rangle$ con $\alpha \notin t$, $\alpha \notin \mathtt{LCC}$ y $\mathtt{fc}(s, \mathtt{LCC})$. Notar que $o' = \{\!\{\alpha \setminus^{\alpha'} s\}\!\}(\mathtt{LCC}\langle [\alpha]\, t \rangle)$. Se concluye entonces por Cor. C.0.33.
  2. $\mathtt{P}$. Luego, $o = \mathtt{LCC}\langle c[\![\beta \setminus^{\alpha} s']\!] \rangle [\![\alpha \setminus^{\alpha'} s]\!]$ y $o' = \mathtt{LCC}\langle c[\![\beta \setminus^{\alpha'} s' \cdot s]\!] \rangle$ con $\alpha \notin c$, $\alpha \notin \mathtt{LCC}$, $\alpha \notin s'$, $\mathtt{fc}(\alpha', \mathtt{LCC})$ y $\mathtt{fc}(s, \mathtt{LCC})$. Notar que $o' = \{\!\{\alpha \setminus^{\alpha'} s\}\!\}(\mathtt{LCC}\langle c[\![\beta \setminus^{\alpha} s']\!] \rangle)$. Se concluye entonces por Cor. C.0.33.
  3. $\mathtt{W}$. Luego, $o = \mathtt{LCC}\langle c[\beta \setminus \alpha] \rangle [\![\alpha \setminus^{\alpha'} s]\!]$ y $o' = \mathtt{LCC}\langle c[\![\beta \setminus^{\alpha} s]\!][\alpha \setminus \alpha'] \rangle$ con $\alpha \notin c$, $\alpha \notin \mathtt{LCC}$, $\mathtt{fc}(\alpha', \mathtt{LCC})$ y $\mathtt{fc}(s, \mathtt{LCC})$. Notar que $o' = \{\!\{\alpha \setminus^{\alpha'} s\}\!\}(\mathtt{LCC}\langle c[\beta \setminus \alpha] \rangle)$. Se concluye entonces por Cor. C.0.33.
- $\mathtt{O} = \mathtt{T}\, u$. Luego, $o = t\, u$ y $o' = t'\, u$ con $t \to_{\mathfrak{C}} t'$. Por *h.i.* se tiene $|t|^\tau_{|u|^\tau_{[]}:l} \geq |t'|^\tau_{|u|^\tau_{[]}:l}$. Se concluye dado que $|o|_l^\tau = |t|^\tau_{|u|^\tau_{[]}:l} + |u|^\tau_{[]} \geq |t'|^\tau_{|u|^\tau_{[]}:l} + |u|^\tau_{[]} = |o'|_l^\tau$.
- $\mathtt{O} = t\, \mathtt{T}$. Luego, $o = t\, u$ y $o' = t\, u'$ con $u \to_{\mathfrak{C}} u'$. Por *h.i.* se tiene $|u|^\tau_{[]} \geq |u'|^\tau_{[]}$. Más aún, por Lem. C.0.29 (3), se tiene también $|t|^\tau_{|u|^\tau_{[]}:l} \geq |t|^\tau_{|u'|^\tau_{[]}:l}$. Se concluye dado que $|o|_l^\tau = |t|^\tau_{|u|^\tau_{[]}:l} + |u|^\tau_{[]} \geq |t|^\tau_{|u'|^\tau_{[]}:l} + |u'|^\tau_{[]} = |o'|_l^\tau$.
- $\mathtt{O} = \lambda x.\mathtt{T}$. Luego, $o = \lambda x.t$ y $o' = \lambda x.t'$ con $t \to_{\mathfrak{C}} t'$. Sea l_0 tal que $l = n : l_0$ o $l = [] = l_0$. Por *h.i.* se tiene $|o|_l^\tau = |t|^\tau_{l_0} \geq |t'|^\tau_{l_0} = |o'|_l^\tau$, por lo que se concluye.
- $\mathtt{O} = \mu\alpha.\mathtt{C}$. Luego, $o = \mu\alpha.c$ y $o' = \mu\alpha.c'$ con $c \to_{\mathfrak{C}} c'$. Por *h.i.* se tiene $|c|^{\tau[\alpha:=l]}_{[]} \geq |c'|^{\tau[\alpha:=l]}_{[]}$. Se concluye dado que $|o|_l^\tau = |c|^{\tau[\alpha:=l]}_{[]} + 1 \geq |c'|^{\tau[\alpha:=l]}_{[]} + 1 = |o'|_l^\tau$.
- $\mathtt{O} = \mathtt{T}[x \setminus u]$. Luego, $o = t[x \setminus u]$ y $o' = t'[x \setminus u]$ con $t \to_{\mathfrak{C}} t'$. Por *h.i.* se tiene $|o|_l^\tau = |t|_l^\tau \geq |t'|_l^\tau = |o'|_l^\tau$, por lo que se concluye.
- $\mathtt{O} = t[x \setminus \mathtt{T}]$. Luego, $o = t[x \setminus u]$ y $o' = t[x \setminus u']$ con $u \to_{\mathfrak{C}} u'$. El resultado es inmediato dado $|o|_l^\tau = |t|_l^\tau = |o'|_l^\tau$.

- $\mathtt{O} = [\alpha]\,\mathtt{T}$. Luego, $o = [\alpha]\,t$ y $o' = [\alpha]\,t'$ con $t \to_{\mathfrak{C}} t'$. Por *h.i.* se tiene $|t|^{\tau}_{\tau(\alpha)} \geq |t'|^{\tau}_{\tau(\alpha)}$. Se concluye dado que $|o|^{\tau}_{l} = |t|^{\tau}_{\tau(\alpha)} + 1 + \sum \tau(\alpha) \geq |t'|^{\tau}_{\tau(\alpha)} + 1 + \sum \tau(\alpha) = |o'|^{\tau}_{l}$.

- $\mathtt{O} = \mathtt{C}[\![\alpha \setminus^{\alpha'} s]\!]$. Luego, $o = c[\![\alpha \setminus^{\alpha'} s]\!]$ y $o' = c'[\![\alpha \setminus^{\alpha'} s]\!]$ con $c \to_{\mathfrak{C}} c'$. Sean $s = u_0 \cdot \ldots \cdot u_n$ y $\tau' = \tau[\alpha := |u_0|^{\tau}_{[]} : \ldots : |u_n|^{\tau}_{[]} : \tau(\alpha')]$. Por *h.i.* se tiene $|o|^{\tau}_{l} = |c|^{\tau'}_{[]} \geq |c'|^{\tau'}_{[]} = |o'|^{\tau}_{l}$, por lo que se concluye.

- $\mathtt{O} = c[\![\alpha \setminus^{\alpha'} \mathtt{S}]\!]$. Luego, $o = c[\![\alpha \setminus^{\alpha'} s]\!]$ y $o' = c[\![\alpha \setminus^{\alpha'} s']\!]$ con $s \to_{\mathfrak{C}} s'$. Sea $s = u_0 \cdot \ldots \cdot u_n$. Entonces, $s \to_{\mathfrak{C}} s'$ implica $s' = u'_0 \cdot \ldots \cdot u'_n$ con $u_k \to_{\mathfrak{C}} u'_k$ para algún $k \in [0, n]$ y $(u_i = u'_i)_{i \leq n, i \neq k}$. Por *h.i.* se tiene $|u_k|^{\tau}_{[]} \geq |u'_k|^{\tau}_{[]}$. Sean $\tau' = \tau[\alpha := |u_0|^{\tau}_{[]} : \ldots : |u_n|^{\tau}_{[]} : \tau(\alpha')]$ y $\tau'' = \tau[\alpha := |u'_0|^{\tau}_{[]} : \ldots : |u'_n|^{\tau}_{[]} : \tau(\alpha')]$. Se tiene entonces $\tau' \geq \tau''$. Luego, por Lem. C.0.29 (2), $|c|^{\tau'}_{[]} \geq |c|^{\tau''}_{[]}$. Se concluye dado que $|o|^{\tau}_{l} = |c|^{\tau'}_{[]} \geq |c|^{\tau''}_{[]} = |o'|^{\tau}_{l}$.

- $\mathtt{O} = \mathtt{C}[\alpha \setminus \beta]$. Luego, $o = c[\alpha \setminus \beta]$ y $o' = c'[\alpha \setminus \beta]$ con $c \to_{\mathfrak{C}} c'$. Por *h.i.* se tiene $|c|^{\tau[\alpha:=\tau(\beta)]}_{[]} \geq |c'|^{\tau[\alpha:=\tau(\beta)]}_{[]}$. Se concluye dado que $|o|^{\tau}_{l} = |c|^{\tau[\alpha:=\tau(\beta)]}_{[]} + 1 \geq |c'|^{\tau[\alpha:=\tau(\beta)]}_{[]} + 1 = |o'|^{\tau}_{l}$.

- $\mathtt{O} = \mathtt{T} \cdot s$. Luego, $o = t \cdot s$ y $o' = t' \cdot s$ con $t \to_{\mathfrak{C}} t'$. Por *h.i.* se tiene $|t|^{\tau}_{[]} \geq |t'|^{\tau}_{[]}$. Se concluye dado que $|o|^{\tau}_{l} = |t|^{\tau}_{[]} + |s|^{\tau}_{[]} \geq |t'|^{\tau}_{[]} + |s|^{\tau}_{[]} = |o'|^{\tau}_{l}$.

- $\mathtt{O} = t \cdot \mathtt{S}$. Luego, $o = t \cdot s$ y $o' = t \cdot s'$ con $s \to_{\mathfrak{C}} s'$. Por *h.i.* se tiene $|s|^{\tau}_{[]} \geq |s'|^{\tau}_{[]}$. Se concluye dado que $|o|^{\tau}_{l} = |t|^{\tau}_{[]} + |s|^{\tau}_{[]} \geq |t|^{\tau}_{[]} + |s'|^{\tau}_{[]} = |o'|^{\tau}_{l}$.

□

Se define a continuación una interpretación de los objetos del ΛM-cálculo en multi-conjuntos de números naturales. La interpretación consiste esencialmente en considerar el peso de las aplicaciones y los renaming explícitos del objeto. Esto está inspirado en la interacción entre ambos constructores propuesta por las reglas de reducción $\mathtt{dM}$ y $\mathtt{N}$. Sea τ una función de nombres en listas de números naturales y l una lista de números naturales, la interpretación $(\!|\_|\!)^{\tau}_{l}$ de objetos del ΛM-cálculo en multi-conjuntos de números naturales se define inductivamente como:

$$
\begin{array}{rcl}
(\!|x|\!)^{\tau}_{l} & \triangleq & \{\!\{\}\!\} \\
(\!|t\,u|\!)^{\tau}_{l} & \triangleq & (\!|t|\!)^{\tau}_{|u|^{\tau}_{[]}:l} \sqcup (\!|u|\!)^{\tau}_{[]} \sqcup \{\!\{|t\,u|^{\tau}_{l}\}\!\} \\
(\!|\lambda x.t|\!)^{\tau}_{l} & \triangleq & (\!|t|\!)^{\tau}_{l'} \qquad\qquad l' = \begin{cases} [] & \text{si } l = [] \\ ns & \text{si } l = n : ns \end{cases} \\
(\!|\mu\alpha.c|\!)^{\tau}_{l} & \triangleq & (\!|c|\!)^{\tau[\alpha:=l]}_{[]} \\
(\!|t[x \setminus u]|\!)^{\tau}_{l} & \triangleq & (\!|t|\!)^{\tau}_{l} \sqcup (\!|u|\!)^{\tau}_{[]} \\
(\!|[\alpha]\,t|\!)^{\tau}_{l} & \triangleq & (\!|t|\!)^{\tau}_{\tau(\alpha)} \\
(\!|c[\![\alpha \setminus^{\alpha'} s]\!]|\!)^{\tau}_{l} & \triangleq & (\!|c|\!)^{\tau[\alpha:=|u_0|^{\tau}_{[]}:\ldots:|u_n|^{\tau}_{[]}:\tau(\alpha')]}_{[]} \sqcup (\!|s|\!)^{\tau}_{[]} \sqcup \{\!\{|c[\![\alpha \setminus^{\alpha'} s]\!]|^{\tau}_{l}\}\!\} \quad s = u_0 \cdot \ldots \cdot u_n \\
(\!|c[\alpha \setminus \beta]|\!)^{\tau}_{l} & \triangleq & (\!|c|\!)^{\tau[\alpha:=\tau(\beta)]}_{[]} \\
(\!|t \cdot s|\!)^{\tau}_{l} & \triangleq & (\!|t|\!)^{\tau}_{[]} \sqcup (\!|s|\!)^{\tau}_{[]}
\end{array}
$$

Notar que en este caso se inspeccionan recursivamente todos los sub-objetos, a diferencia de la función de peso que ignora las sustituciones y los replacements explícitos. La interpretación se extiende a contextos definiendo $(\!|\Box|\!)^{\tau}_{l} \triangleq (\!|\boxdot|\!)^{\tau}_{l} \triangleq \{\!\{\}\!\}$.

La interpretación $(\!|\_|\!)^{\tau}_{l}$ resulta de particular interés por la noción de orden de multi-conjunto subyacente [DM79]:

Definición C.0.35. *Dado un orden parcial $\langle \mathcal{S}, > \rangle$, el* orden de multi-conjuntos $>$ *sobre $\mathscr{M}(\mathcal{S})$ se define como: $\mathcal{M} > \mathcal{M}'$ sii existen multi-conjuntos $\mathcal{X}, \mathcal{Y} \in \mathscr{M}(\mathcal{S})$ tal que $\{\!\{\}\!\} \neq \mathcal{X} \subseteq \mathcal{M}$, $\mathcal{M}' = (\mathcal{M} \setminus \mathcal{X}) \sqcup \mathcal{Y}$ y para todo elemento $y \in \mathcal{Y}$ existe un elemento $x \in \mathcal{X}$ tal que $x > y$.*

Informalmente, $\mathcal{M} > \mathcal{M}'$ si $\mathcal{M}'$ resulta de remover al menos un elemento de $\mathcal{M}$, pudiendo agregarse una cantidad finita de elementos nuevos, siempre que estos sean menores a alguno de los eliminados. Por ejemplo, $\{\!\{2,3\}\!\} > \{\!\{2,1,1,1\}\!\}$. Notar que, en particular, $\mathcal{M} > \mathcal{M}'$ sii $\mathcal{M} \sqcup \mathcal{N} > \mathcal{M}' \sqcup \mathcal{N}$.

Teorema C.0.36 ([DM79])**.** *El orden de multi-conjunto $\langle \mathscr{M}(\mathcal{S}), > \rangle$ sobre $\langle \mathcal{S}, > \rangle$ es bien fundado si y solo si $\langle \mathcal{S}, > \rangle$ lo es.*

Esta noción de orden resulta conveniente para garantizar la terminación de la relación de reducción canónica.

Al igual que para la función de peso, se introducen primero algunos resultados auxiliares para el orden de multi-conjunto dado por la interpretación de los objetos del ΛM-cálculo.

Lema C.0.37. *Sea $o \in \mathbb{O}_{\Lambda M}$.*

1. *Si $\alpha \notin \mathsf{fn}(o)$, entonces $(\!|o|\!)_l^{\tau[\alpha:=l']} = (\!|o|\!)_l^{\tau}$.*
2. *Si $\tau(\alpha) \geq \tau'(\alpha)$ para todo $\alpha \in o$, entonces $(\!|o|\!)_l^{\tau} \geq (\!|o|\!)_l^{\tau'}$.*
3. *Si $l \geq l'$, entonces $(\!|o|\!)_l^{\tau} \geq (\!|o|\!)_{l'}^{\tau}$.*

Demostración. Por inducción en o usando el Lem C.0.29. □

Lema C.0.38. *Sea $t \in \mathbb{T}_{\Lambda M}$. Luego, $(\!|\mathtt{L}\langle t \rangle|\!)_l^{\tau} = (\!|\mathtt{L}|\!)_{[]}^{\tau} \sqcup (\!|t|\!)_l^{\tau}$.*

Demostración. Por inducción en $\mathtt{L}$.

- $\mathtt{L} = \Box$. Luego, $\mathtt{L}\langle t \rangle = t$ y $(\!|\mathtt{L}\langle t \rangle|\!)_l^{\tau} = (\!|t|\!)_l^{\tau} = \{\!\{\}\!\} \sqcup (\!|t|\!)_l^{\tau} = (\!|\mathtt{L}|\!)_{[]}^{\tau} \sqcup (\!|t|\!)_l^{\tau}$, por lo que se concluye.
- $\mathtt{L} = \mathtt{L}'[x \setminus u]$. Luego, $\mathtt{L}\langle t \rangle = \mathtt{L}'\langle t \rangle[x \setminus u]$. Se concluye directo de la *h.i.* dado que $(\!|\mathtt{L}\langle t \rangle|\!)_l^{\tau} = (\!|\mathtt{L}'\langle t \rangle|\!)_l^{\tau} \sqcup (\!|u|\!)_{[]}^{\tau} = (\!|\mathtt{L}'|\!)_{[]}^{\tau} \sqcup (\!|t|\!)_l^{\tau} \sqcup (\!|u|\!)_{[]}^{\tau} = (\!|\mathtt{L}|\!)_{[]}^{\tau} \sqcup (\!|t|\!)_l^{\tau}$.

□

Lema C.0.39. *Sea $t \in \mathbb{T}_{\Lambda M}$ y $s = u_0 \cdot \ldots \cdot u_n$ un stack. Luego, $(\!|t :: s|\!)_l^{\tau} = (\!|t|\!)_{|u_0|_{[]}^{\tau}:\ldots:|u_n|_{[]}^{\tau}:l}^{\tau} \sqcup (\!|s|\!)_{[]}^{\tau} \sqcup (\bigsqcup_{i=0}^{n} \{\!\{|t|_{|u_0|_{[]}^{\tau}:\ldots:|u_n|_{[]}^{\tau}:l}^{\tau} + \sum_{j=0}^{i} |u_j|_{[]}^{\tau}\}\!\})$.*

Demostración. Por inducción en n.

- $n = 0$. Luego, $t :: s = t\,u_0$ y, por definición, $(\!|t\,u_0|\!)_l^{\tau} = (\!|t|\!)_{|u_0|_{[]}^{\tau}:l}^{\tau} \sqcup (\!|u_0|\!)_{[]}^{\tau} \sqcup \{\!\{|t|_{|u_0|_{[]}^{\tau}:l}^{\tau} + |u_0|_{[]}^{\tau}\}\!\}$, por lo que se concluye.
- $n > 0$. Luego, $s = u_0 \cdot s'$ con $t :: s = (t\,u_0) :: s'$ y $s' = u_1 \cdot \ldots \cdot u_n$. Se concluye directo de la *h.i.* dado que

$$\begin{aligned}
(\!|(t\,u_0) :: s'|\!)_l^{\tau} &= (\!|t\,u_0|\!)_{|u_1|_{[]}^{\tau}:\ldots:|u_n|_{[]}^{\tau}:l}^{\tau} \sqcup (\!|s'|\!)_{[]}^{\tau} \sqcup (\bigsqcup_{i=1}^{n} \{\!\{|t :: u_0|_{|u_1|_{[]}^{\tau}:\ldots:|u_n|_{[]}^{\tau}:l}^{\tau} + \sum_{j=1}^{i} |u_j|_{[]}^{\tau}\}\!\}) \\
&= (\!|t|\!)_{|u_0|_{[]}^{\tau}:\ldots:|u_n|_{[]}^{\tau}:l}^{\tau} \sqcup (\!|u_0|\!)_{[]}^{\tau} \sqcup \{\!\{|t|_{|u_0|_{[]}^{\tau}:\ldots:|u_n|_{[]}^{\tau}:l}^{\tau} + |u_0|_{[]}^{\tau}\}\!\} \\
&\qquad \sqcup (\!|s'|\!)_{[]}^{\tau} \sqcup (\bigsqcup_{i=1}^{n} \{\!\{|t|_{|u_0|_{[]}^{\tau}:\ldots:|u_n|_{[]}^{\tau}:l}^{\tau} + |u_0|_{[]}^{\tau} + \sum_{j=1}^{i} |u_j|_{[]}^{\tau}\}\!\}) \\
&= (\!|t|\!)_{|u_0|_{[]}^{\tau}:\ldots:|u_n|_{[]}^{\tau}:l}^{\tau} \sqcup (\!|s|\!)_{[]}^{\tau} \sqcup (\bigsqcup_{i=0}^{n} \{\!\{|t|_{|u_0|_{[]}^{\tau}:\ldots:|u_n|_{[]}^{\tau}:l}^{\tau} + \sum_{j=0}^{i} |u_j|_{[]}^{\tau}\}\!\})
\end{aligned}$$

□

Se dice que un nombre α *ocurre linealmente* en un objeto o si o tiene una única ocurrencia de α en posición de un contexto lineal. Es decir, $\mathsf{fn}_\alpha(o) = 1$ y $o = \mathtt{LXC}\langle c\rangle$ con $c = [\alpha]\,t$, $c = c'[\![\gamma \setminus^\alpha s]\!]$ o $c = c'[\gamma \setminus \alpha]$, donde $\mathtt{LXC}$ denota un contexto $\mathtt{LTC}$ o $\mathtt{LCC}$. Dada esta noción, se tiene el siguiente lema de replacement lineal para la interpretación de los objetos del ΛM-cálculo.

Lema C.0.40 (Replacement lineal)**.** *Sea $o \in \mathbb{O}_{\Lambda M}$ y $s = u_0 \cdot \ldots \cdot u_n$ un stack. Si α ocurre linealmente en o, entonces $(\!|\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o|\!)^\tau_l < (\!|o|\!)^{\tau[\alpha:=|u_0|^\tau_{[]}:\ldots:|u_n|^\tau_{[]}:\tau(\alpha')]}_l \sqcup (\!|s|\!)^\tau_{[]} \sqcup \{\!\{|o|^{\tau[\alpha:=|u_0|^\tau_{[]}:\ldots:|u_n|^\tau_{[]}:\tau(\alpha')]}_l\}\!\}$.*

Demostración. Por inducción en o. Sea $\tau' = \tau[\alpha := |u_0|^\tau_{[]} : \ldots : |u_n|^\tau_{[]} : \tau(\alpha')]$.

- $o = x$. El resultado es inmediato dado que α no ocurre linealmente en o.
- $o = t\,u$. Luego, $\alpha \notin \mathsf{fn}(u)$ y $\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o = \{\!\{\alpha \setminus^{\alpha'} s\}\!\}t\,u$. Por *h.i.* $(\!|\{\!\{\alpha \setminus^{\alpha'} s\}\!\}t|\!)^\tau_{|u|^\tau_{[]}:l} < (\!|t|\!)^{\tau'}_{|u|^\tau_{[]}:l} \sqcup (\!|s|\!)^\tau_{[]} \sqcup \{\!\{|t|^{\tau'}_{|u|^\tau_{[]}:l}\}\!\}$. Más aún, $|t|^{\tau'}_{|u|^\tau_{[]}:l} \leq |t|^{\tau'}_{|u|^\tau_{[]}:l} + |u|^\tau_{[]} = |o|^{\tau'}_l$ y, por Lem. C.0.32, $|\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o|^\tau_l = |o|^{\tau'}_l$. Por Lem. C.0.37 (1) con $\alpha \notin \mathsf{fn}(u)$ también se tiene $(\!|u|\!)^\tau_{[]} = (\!|u|\!)^{\tau'}_{[]}$. Finalmente, se concluye dado que

$$\begin{array}{rcl}(\!|\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o|\!)^\tau_l & = & (\!|\{\!\{\alpha \setminus^{\alpha'} s\}\!\}t|\!)^\tau_{|u|^\tau_{[]}:l} \sqcup (\!|u|\!)^\tau_{[]} \sqcup \{\!\{|\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o|^\tau_l\}\!\} \\ & < & (\!|t|\!)^{\tau'}_{|u|^\tau_{[]}:l} \sqcup (\!|u|\!)^\tau_{[]} \sqcup (\!|s|\!)^\tau_{[]} \sqcup \{\!\{|t|^{\tau'}_{|u|^\tau_{[]}:l}, |\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o|^\tau_l\}\!\} \\ & \leq & (\!|t|\!)^{\tau'}_{|u|^\tau_{[]}:l} \sqcup (\!|u|\!)^{\tau'}_{[]} \sqcup (\!|s|\!)^\tau_{[]} \sqcup \{\!\{|o|^{\tau'}_l, |o|^{\tau'}_l\}\!\} \\ & = & (\!|o|\!)^{\tau'}_l \sqcup (\!|s|\!)^\tau_{[]} \sqcup \{\!\{|o|^{\tau'}_l\}\!\}\end{array}$$

- $o = \lambda x.t$. Luego, $\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o = \lambda x.\{\!\{\alpha \setminus^{\alpha'} s\}\!\}t$ con $x \notin s$. Sea l_0 tal que $l = n : l_0$ o $l = [] = l_0$. Por *h.i.* se tiene $(\!|\{\!\{\alpha \setminus^{\alpha'} s\}\!\}t|\!)^\tau_{l_0} < (\!|t|\!)^{\tau'}_{l_0} \sqcup (\!|s|\!)^\tau_{[]} \sqcup \{\!\{|t|^{\tau'}_{l_0}\}\!\}$. Se concluye dado que $(\!|\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o|\!)^\tau_l = (\!|\{\!\{\alpha \setminus^{\alpha'} s\}\!\}t|\!)^\tau_{l_0} < (\!|t|\!)^{\tau'}_{l_0} \sqcup (\!|s|\!)^\tau_{[]} \sqcup \{\!\{|t|^{\tau'}_{l_0}\}\!\} = (\!|o|\!)^{\tau'}_l \sqcup (\!|s|\!)^\tau_{[]} \sqcup \{\!\{|o|^{\tau'}_l\}\!\}$.
- $o = \mu\gamma.c$. Luego, $\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o = \mu\gamma.\{\!\{\alpha \setminus^{\alpha'} s\}\!\}c$ con $\gamma \notin s$. Por α-conversión se asume $\gamma \neq \alpha$. Por *h.i.* se tiene entonces $(\!|\{\!\{\alpha \setminus^{\alpha'} s\}\!\}c|\!)^{\tau[\gamma:=l]}_{[]} < (\!|c|\!)^{\tau'[\gamma:=l]}_{[]} \sqcup (\!|s|\!)^{\tau[\gamma:=l]}_{[]} \sqcup \{\!\{|c|^{\tau'[\gamma:=l]}_{[]}\}\!\}$. Más aún, $|c|^{\tau'[\gamma:=l]}_{[]} < |c|^{\tau'[\gamma:=l]}_{[]} + 1 = |o|^{\tau'}_l$ y, por Lem. C.0.37 (1) con $\gamma \notin s$ también se tiene $(\!|s|\!)^{\tau[\gamma:=l]}_{[]} = (\!|s|\!)^\tau_{[]}$. Se concluye dado que $(\!|\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o|\!)^\tau_l = (\!|\{\!\{\alpha \setminus^{\alpha'} s\}\!\}c|\!)^{\tau[\gamma:=l]}_{[]} < (\!|c|\!)^{\tau'[\gamma:=l]}_{[]} \sqcup (\!|s|\!)^{\tau[\gamma:=l]}_{[]} \sqcup \{\!\{|c|^{\tau'[\gamma:=l]}_{[]}\}\!\} < (\!|o|\!)^{\tau'}_l \sqcup (\!|s|\!)^\tau_{[]} \sqcup \{\!\{|o|^{\tau'}_l\}\!\}$.
- $o = t[x \setminus u]$. Luego, $\alpha \notin \mathsf{fn}(u)$ y $\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o = \{\!\{\alpha \setminus^{\alpha'} s\}\!\}t[x \setminus u]$. Por *h.i.* $(\!|\{\!\{\alpha \setminus^{\alpha'} s\}\!\}t|\!)^\tau_l < (\!|t|\!)^{\tau'}_l \sqcup (\!|s|\!)^\tau_{[]} \sqcup \{\!\{|t|^{\tau'}_l\}\!\}$. Más aún, $|t|^{\tau'}_l = |o|^{\tau'}_l$ y, por Lem. C.0.37 (1) con $\alpha \notin \mathsf{fn}(u)$ también se tiene $(\!|u|\!)^\tau_{[]} = (\!|u|\!)^{\tau'}_{[]}$. Se concluye dado que $(\!|\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o|\!)^\tau_l = (\!|\{\!\{\alpha \setminus^{\alpha'} s\}\!\}t|\!)^\tau_l \sqcup (\!|u|\!)^\tau_{[]} < (\!|t|\!)^{\tau'}_l \sqcup (\!|u|\!)^{\tau'}_{[]} \sqcup (\!|s|\!)^\tau_{[]} \sqcup \{\!\{|t|^{\tau'}_l\}\!\} = (\!|o|\!)^{\tau'}_l \sqcup (\!|s|\!)^\tau_{[]} \sqcup \{\!\{|o|^{\tau'}_l\}\!\}$.
- $o = [\alpha]\,t$. Luego, $\alpha \notin t$ y $\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o = [\alpha']\,t :: s$. Por Lem. C.0.39, se tiene $(\!|\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o|\!)^\tau_l = (\!|t|\!)^\tau_{\tau'(\alpha)} \sqcup (\!|s|\!)^\tau_{[]} \sqcup (\bigsqcup^n_{i=0} \{\!\{|t|^\tau_{\tau'(\alpha)} + \sum^i_{j=0} |u_j|^\tau_{[]}\}\!\})$. Más aún, $(\sum^i_{j=0} |u_j|^\tau_{[]} \leq |s|^\tau_{[]})_{i \leq n}$ y a su vez $|s|^\tau_{[]} \leq |s|^\tau_{[]} + \sum \tau(\alpha') = \sum \tau'(\alpha)$. Por Lem. C.0.37 (1) y Lem. C.0.29 (1) con $\alpha \notin \mathsf{fn}(t)$ se tienen $(\!|t|\!)^\tau_{\tau'(\alpha)} = (\!|t|\!)^{\tau'}_{\tau'(\alpha)}$ y $|t|^\tau_{\tau'(\alpha)} = |t|^{\tau'}_{\tau'(\alpha)}$ respectivamente. Finalmente, se concluye dado que

$$\begin{array}{rcl}(\!|\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o|\!)^\tau_l & = & (\!|t|\!)^\tau_{\tau'(\alpha)} \sqcup (\!|s|\!)^\tau_{[]} \sqcup (\bigsqcup^n_{i=0} \{\!\{|t|^\tau_{\tau'(\alpha)} + \sum^i_{j=0} |u_j|^\tau_{[]}\}\!\}) \\ & < & (\!|t|\!)^\tau_{\tau'(\alpha)} \sqcup (\!|s|\!)^\tau_{[]} \sqcup (\bigsqcup^n_{i=0} \{\!\{|t|^\tau_{\tau'(\alpha)} + 1 + \sum \tau'(\alpha)\}\!\}) \\ & = & (\!|t|\!)^{\tau'}_{\tau'(\alpha)} \sqcup (\!|s|\!)^\tau_{[]} \sqcup (\bigsqcup^n_{i=0} \{\!\{|t|^{\tau'}_{\tau'(\alpha)} + 1 + \sum \tau'(\alpha)\}\!\}) \\ & = & (\!|o|\!)^{\tau'}_l \sqcup (\!|s|\!)^\tau_{[]} \sqcup \{\!\{|o|^{\tau'}_l\}\!\}\end{array}$$

- $o = [\delta]\, t$ con $\delta \neq \alpha$. Luego, $\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o = [\delta]\, \{\!\{\alpha \setminus^{\alpha'} s\}\!\} t$. Más aún, $\delta \neq \alpha$ implica $\tau(\delta) = \tau'(\delta)$. Por *h.i.* se tiene entonces $(\!|\{\!\{\alpha \setminus^{\alpha'} s\}\!\} t|\!)^{\tau}_{\tau(\delta)} < (\!|t|\!)^{\tau'}_{\tau(\delta)} \sqcup (\!|s|\!)^{\tau}_{[]} \sqcup \{\!|\, |t|^{\tau'}_{\tau(\delta)} \,|\!\}$. Más aún, $|t|^{\tau'}_{\tau(\delta)} < |t|^{\tau'}_{\tau(\delta)} + 1 + \sum \tau(\delta) = |o|^{\tau'}_{l}$. Se concluye dado que $(\!|\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o|\!)^{\tau}_{l} = (\!|\{\!\{\alpha \setminus^{\alpha'} s\}\!\} t|\!)^{\tau}_{\tau(\delta)} < (\!|t|\!)^{\tau'}_{\tau(\delta)} \sqcup (\!|s|\!)^{\tau}_{[]} \sqcup \{\!|\, |t|^{\tau'}_{\tau(\delta)} \,|\!\} < (\!|o|\!)^{\tau'}_{l} \sqcup (\!|s|\!)^{\tau}_{[]} \sqcup \{\!|\, |o|^{\tau'}_{l} \,|\!\}$.

- $o = c[\![\gamma \setminus^{\alpha} s']\!]$. Luego, $\alpha \notin c$, $\alpha \notin s'$ y $\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o = c[\![\gamma \setminus^{\alpha'} s' \cdot s]\!]$ con $\gamma \notin s$, $\gamma \neq \alpha$ y $\gamma \neq \alpha'$. Sea $s' = v_0 \cdot \ldots \cdot v_m$. Más aún, por Lem. C.0.29 (1) con $\alpha \notin s'$, se tiene $|s'|^{\tau}_{[]} = |s'|^{\tau'}_{[]}$, lo que implica $(|v_j|^{\tau}_{[]} = |v_j|^{\tau'}_{[]})_{j \leq m}$. Luego, por Lem. C.0.37 (1) con $\alpha \notin c$ y $\alpha \notin s'$, se tienen $(\!|c|\!)^{\tau[\gamma := |v_0|^{\tau}_{[]} : \ldots : |v_m|^{\tau}_{[]} : |u_0|^{\tau}_{[]} : \ldots : |u_n|^{\tau}_{[]} : \tau(\alpha')]}_{[]} = (\!|c|\!)^{\tau'[\gamma := |v_0|^{\tau'}_{[]} : \ldots : |v_m|^{\tau'}_{[]} : \tau'(\alpha)]}_{[]}$ y $(\!|s'|\!)^{\tau}_{[]} = (\!|s'|\!)^{\tau'}_{[]}$ respectivamente. Del mismo modo, por Lem. C.0.29 (1) con $\alpha \notin c$, se tiene $|c|^{\tau[\gamma := |v_0|^{\tau}_{[]} : \ldots : |v_m|^{\tau}_{[]} : |u_0|^{\tau}_{[]} : \ldots : |u_n|^{\tau}_{[]} : \tau(\alpha')]}_{[]} = |c|^{\tau'[\gamma := |v_0|^{\tau'}_{[]} : \ldots : |v_m|^{\tau'}_{[]} : \tau'(\alpha)]}_{[]}$. Finalmente, se concluye dado que

$$
\begin{array}{rcl}
(\!|\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o|\!)^{\tau}_{l} & = & (\!|c|\!)^{\tau[\gamma := |v_0|^{\tau}_{[]} : \ldots : |v_m|^{\tau}_{[]} : |u_0|^{\tau}_{[]} : \ldots : |u_n|^{\tau}_{[]} : \tau(\alpha')]}_{[]} \sqcup (\!|s' \cdot s|\!)^{\tau}_{[]} \sqcup \{\!|\, |c|^{\tau[\gamma := |v_0|^{\tau}_{[]} : \ldots : |v_m|^{\tau}_{[]} : |u_0|^{\tau}_{[]} : \ldots : |u_n|^{\tau}_{[]} : \tau(\alpha')]}_{[]} \,|\!\} \\
& = & (\!|c|\!)^{\tau'[\gamma := |v_0|^{\tau'}_{[]} : \ldots : |v_m|^{\tau'}_{[]} : \tau'(\alpha)]}_{[]} \sqcup (\!|s'|\!)^{\tau'}_{[]} \sqcup (\!|s|\!)^{\tau}_{[]} \sqcup \{\!|\, |c|^{\tau'[\gamma := |v_0|^{\tau'}_{[]} : \ldots : |v_m|^{\tau'}_{[]} : \tau'(\alpha)]}_{[]} \,|\!\} \\
& = & (\!|o|\!)^{\tau'}_{l} \sqcup (\!|s|\!)^{\tau}_{[]} \\
& < & (\!|o|\!)^{\tau'}_{l} \sqcup (\!|s|\!)^{\tau}_{[]} \sqcup \{\!|\, |o|^{\tau'}_{l} \,|\!\}
\end{array}
$$

- $o = c[\![\gamma \setminus^{\delta} s']\!]$ con $\delta \neq \alpha$. Luego, $\alpha \notin s'$ y $\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o = (\{\!\{\alpha \setminus^{\alpha'} s\}\!\} c)[\![\gamma \setminus^{\delta} s']\!]$ con $\gamma \notin s$, $\gamma \neq \alpha$ y $\gamma \neq \alpha'$. Sea $s' = v_0 \cdot \ldots \cdot v_m$. Por *h.i.* se tiene $(\!|\{\!\{\alpha \setminus^{\alpha'} s\}\!\} c|\!)^{\tau[\gamma := |v_0|^{\tau}_{[]} : \ldots : |v_m|^{\tau}_{[]} : \tau(\delta)]}_{[]} < (\!|c|\!)^{\tau'[\gamma := |v_0|^{\tau}_{[]} : \ldots : |v_m|^{\tau}_{[]} : \tau(\delta)]}_{[]} \sqcup (\!|s|\!)^{\tau}_{[]} \sqcup \{\!|\, |c|^{\tau'[\gamma := |v_0|^{\tau}_{[]} : \ldots : |v_m|^{\tau}_{[]} : \tau(\delta)]}_{[]} \,|\!\}$. Más aún, por Lem. C.0.29 (1) con $\alpha \notin s'$, se tiene $|s'|^{\tau}_{[]} = |s'|^{\tau'}_{[]}$, lo que implica $(|v_j|^{\tau}_{[]} = |v_j|^{\tau'}_{[]})_{j \leq m}$, Luego, por $\delta \neq \alpha$ se tiene $|c|^{\tau'[\gamma := |v_0|^{\tau}_{[]} : \ldots : |v_m|^{\tau}_{[]} : \tau(\delta)]}_{[]} = |c|^{\tau'[\gamma := |v_0|^{\tau'}_{[]} : \ldots : |v_m|^{\tau'}_{[]} : \tau'(\delta)]}_{[]} = |o|^{\tau'}_{l}$ y, por Lem. C.0.32, $|\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o|^{\tau}_{l} = |o|^{\tau'}_{l}$. Por Lem. C.0.37 (1) con $\alpha \notin \mathsf{fn}(s')$ también se tiene $(\!|s'|\!)^{\tau}_{[]} = (\!|s'|\!)^{\tau'}_{[]}$. Finalmente, se concluye dado que

$$
\begin{array}{rcl}
(\!|\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o|\!)^{\tau}_{l} & = & (\!|\{\!\{\alpha \setminus^{\alpha'} s\}\!\} c|\!)^{\tau[\gamma := |v_0|^{\tau}_{[]} : \ldots : |v_m|^{\tau}_{[]} : \tau(\delta)]}_{[]} \sqcup (\!|s'|\!)^{\tau}_{[]} \sqcup \{\!|\, |\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o|^{\tau}_{l} \,|\!\} \\
& < & (\!|c|\!)^{\tau'[\gamma := |v_0|^{\tau}_{[]} : \ldots : |v_m|^{\tau}_{[]} : \tau(\delta)]}_{[]} \sqcup (\!|s|\!)^{\tau}_{[]} \sqcup (\!|s'|\!)^{\tau}_{[]} \sqcup \{\!|\, |c|^{\tau'[\gamma := |v_0|^{\tau}_{[]} : \ldots : |v_m|^{\tau}_{[]} : \tau(\delta)]}_{[]}, |\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o|^{\tau}_{l} \,|\!\} \\
& = & (\!|c|\!)^{\tau'[\gamma := |v_0|^{\tau'}_{[]} : \ldots : |v_m|^{\tau'}_{[]} : \tau'(\delta)]}_{[]} \sqcup (\!|s|\!)^{\tau}_{[]} \sqcup (\!|s'|\!)^{\tau'}_{[]} \sqcup \{\!|\, |c|^{\tau'[\gamma := |v_0|^{\tau'}_{[]} : \ldots : |v_m|^{\tau'}_{[]} : \tau'(\delta)]}_{[]}, |o|^{\tau'}_{l} \,|\!\} \\
& = & (\!|o|\!)^{\tau'}_{l} \sqcup (\!|s|\!)^{\tau}_{[]} \sqcup \{\!|\, |o|^{\tau'}_{l} \,|\!\}
\end{array}
$$

- $o = c[\gamma \setminus \alpha]$. Luego, $\alpha \notin c$ y $\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o = c[\![\gamma \setminus^{\delta} s]\!][\delta \setminus \alpha']$ con $\gamma \notin s$, $\gamma \neq \alpha$, $\gamma \neq \alpha'$ y δ un nombre fresco. Por Lem. C.0.29 (1) con δ fresco, se tiene $|s|^{\tau[\delta := \tau(\alpha')]}_{[]} = |s|^{\tau}_{[]}$, lo que implica $(|u_i|^{\tau[\delta := \tau(\alpha')]}_{[]} = |u_i|^{\tau}_{[]})_{i \leq n}$. Entonces, apelando al Lem. C.0.37 (1) con δ fresco, se tiene $(\!|c|\!)^{\tau[\delta := \tau(\alpha')][\gamma := |u_0|^{\tau[\delta := \tau(\alpha')]}_{[]} : \ldots : |u_n|^{\tau[\delta := \tau(\alpha')]}_{[]} : \tau[\delta := \tau(\alpha')](\delta)]}_{[]} = (\!|c|\!)^{\tau[\gamma := |u_0|^{\tau}_{[]} : \ldots : |u_n|^{\tau}_{[]} : \tau(\alpha')]}_{[]}$ junto con $(\!|s|\!)^{\tau[\delta := \tau(\alpha')]}_{[]} = (\!|s|\!)^{\tau}_{[]}$. Más aún, por Lem. C.0.37 (1) con $\alpha \notin c$, se tiene $(\!|c|\!)^{\tau[\gamma := |u_0|^{\tau}_{[]} : \ldots : |u_n|^{\tau}_{[]} : \tau(\alpha')]}_{[]} = (\!|c|\!)^{\tau'[\gamma := \tau'(\alpha)]}_{[]}$. Análogamente, Lem. C.0.29 (1) con δ fresco y $\alpha \notin c$ se tiene $|c[\![\gamma \setminus^{\delta} s]\!]|^{\tau[\delta := \tau(\alpha')]}_{[]} =$

$|c|_{[]}^{\tau'[\gamma:=\tau'(\alpha)]}$. Finalmente, se concluye dado que

$$\begin{aligned}
(\!|\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o|\!)_l^\tau &= (\!|c[\![\gamma \setminus^{\delta} s]\!]|\!)_{[]}^{\tau[\delta:=\tau(\alpha')]} \\
&= (\!|c|\!)_{[]}^{\tau[\delta:=\tau(\alpha')][\gamma:=|u_0|_{[]}^{\tau[\delta:=\tau(\alpha')]}:\ldots:|u_n|_{[]}^{\tau[\delta:=\tau(\alpha')]}:\tau[\delta:=\tau(\alpha')](\delta)]} \sqcup (\!|s|\!)_{[]}^{\tau[\delta:=\tau(\alpha')]} \sqcup \{\!\{|c[\![\gamma \setminus^{\delta} s]\!]|_{[]}^{\tau[\delta:=\tau(\alpha')]}\}\!\} \\
&= (\!|c|\!)_{[]}^{\tau'[\gamma:=\tau'(\alpha)]} \sqcup (\!|s|\!)_{[]}^{\tau} \sqcup \{\!\{|c|_{[]}^{\tau'[\gamma:=\tau'(\alpha)]}\}\!\} \\
&< (\!|c|\!)_{[]}^{\tau'[\gamma:=\tau'(\alpha)]} \sqcup (\!|s|\!)_{[]}^{\tau} \sqcup \{\!\{|c|_{[]}^{\tau'[\gamma:=\tau'(\alpha)]} + 1\}\!\} \\
&= (\!|o|\!)_l^{\tau'} \sqcup (\!|s|\!)_{[]}^{\tau} \sqcup \{\!\{|o|_l^{\tau'}\}\!\}
\end{aligned}$$

- $o = c[\gamma \setminus \delta]$ con $\delta \neq \alpha$. Luego, $\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o = (\{\!\{\alpha \setminus^{\alpha'} s\}\!\}c)[\gamma \setminus \delta]$ con $\gamma \notin s$, $\gamma \neq \alpha$ y $\gamma \neq \alpha'$. Más aún, $\delta \neq \alpha$ implica $\tau(\delta) = \tau'(\delta)$. Por *h.i.* se tiene entonces $(\!|\{\!\{\alpha \setminus^{\alpha'} s\}\!\}c|\!)_{[]}^{\tau[\gamma:=\tau(\delta)]} < (\!|c|\!)_{[]}^{\tau'[\gamma:=\tau(\delta)]} \sqcup (\!|s|\!)_{[]}^{\tau[\gamma:=\tau(\delta)]} \sqcup \{\!\{|c|_{[]}^{\tau'[\gamma:=\tau(\delta)]}\}\!\}$. Más aún, como $\delta \neq \alpha$, se tienen $(\!|c|\!)_{[]}^{\tau'[\gamma:=\tau(\delta)]} = (\!|c|\!)_{[]}^{\tau'[\gamma:=\tau'(\delta)]} = (\!|o|\!)_l^{\tau'}$ y $|c|_{[]}^{\tau'[\gamma:=\tau(\delta)]} = |c|_{[]}^{\tau'[\gamma:=\tau'(\delta)]} < |c|_{[]}^{\tau'[\gamma:=\tau'(\delta)]} + 1 = |o|_l^{\tau'}$. Además, por Lem. C.0.37 (1) con $\gamma \notin s$, se tiene $(\!|s|\!)_{[]}^{\tau[\gamma:=\tau(\delta)]} = (\!|s|\!)_{[]}^{\tau}$. Se concluye dado que $(\!|\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o|\!)_l^{\tau} = (\!|\{\!\{\alpha \setminus^{\alpha'} s\}\!\}c|\!)_{[]}^{\tau[\gamma:=\tau(\delta)]} < (\!|c|\!)_{[]}^{\tau'[\gamma:=\tau(\delta)]} \sqcup (\!|s|\!)_{[]}^{\tau[\gamma:=\tau(\delta)]} \sqcup \{\!\{|c|_{[]}^{\tau'[\gamma:=\tau(\delta)]}\}\!\} < (\!|o|\!)_l^{\tau'} \sqcup (\!|s|\!)_{[]}^{\tau} \sqcup \{\!\{|o|_l^{\tau'}\}\!\}$.
- $o = t \cdot s'$. El resultado es inmediato dado que α no ocurre linealmente en o.

□

Corolario C.0.41. *Sea $c \in \mathbb{C}_{\Lambda M}$ y $s = u_0 \cdot \ldots \cdot u_n$ un stack. Si α ocurre linealmente en c, entonces $(\!|c[\![\alpha \setminus^{\alpha'} s]\!]|\!)_l^\tau > (\!|\{\!\{\alpha \setminus^{\alpha'} s\}\!\}c|\!)_l^\tau$.*

Demostración. Al igual que $|\_|_l^\tau$, la interpretación $(\!|\_|\!)_l^\tau$ ignora la lista l en el caso de los comandos. Luego, $(\!|c[\![\alpha \setminus^{\alpha'} s]\!]|\!)_l^\tau = (\!|c|\!)_l^{\tau[\alpha:=|u_0|_{[]}^\tau:\ldots:|u_n|_{[]}^\tau:\tau(\alpha')]} \sqcup (\!|s|\!)_{[]}^\tau \sqcup \{\!\{|c|_l^{\tau[\alpha:=|u_0|_{[]}^\tau:\ldots:|u_n|_{[]}^\tau:\tau(\alpha')]}\}\!\} > (\!|\{\!\{\alpha \setminus^{\alpha'} s\}\!\}c|\!)_{[]}^\tau = (\!|\{\!\{\alpha \setminus^{\alpha'} s\}\!\}c|\!)_l^\tau$. □

Finalmente, se prueba que el orden orden de multi-conjuntos dado por la interpretación de los objetos del ΛM-cálculo decrece estrictamente al aplicar un paso de reducción canónica.

Lema C.0.42. *Sea $o \in \mathbb{O}_{\Lambda M}$. Si $o \to_{\mathfrak{C}} o'$, entonces $(\!|o|\!)_l^\tau > (\!|o'|\!)_l^\tau$.*

Demostración. Por definición, $o \to_{\mathfrak{C}} o'$ implica $o = \mathtt{O}\langle l\rangle$ y $o' = \mathtt{O}\langle r\rangle$ con $l \mapsto_* r$, $* \in \{\mathtt{dB}, \mathtt{dM}, \mathtt{N}, \mathtt{P}, \mathtt{W}\}$. La demostración es por inducción en $\mathtt{O}$.

- $\mathtt{O} = \Box$. Luego, se tienen dos casos posibles según la regla de reescritura aplicada al reducir.
  1. $\mathtt{dB}$. Luego, $o = \mathtt{L}\langle \lambda x.t\rangle\, u$ y $o' = \mathtt{L}\langle t[x \setminus u]\rangle$. Por Lem. C.0.38, $(\!|o|\!)_l^\tau = (\!|\mathtt{L}\langle \lambda x.t\rangle|\!)_{|u|_{[]}^\tau:l}^\tau \sqcup (\!|u|\!)_{[]}^\tau \sqcup \{\!\{|o|_l^\tau\}\!\} = (\!|\mathtt{L}|\!)_{[]}^\tau \sqcup (\!|\lambda x.t|\!)_{|u|_{[]}^\tau:l}^\tau \sqcup (\!|u|\!)_{[]}^\tau \sqcup \{\!\{|o|_l^\tau\}\!\} = (\!|\mathtt{L}|\!)_{[]}^\tau \sqcup (\!|t|\!)_l^\tau \sqcup (\!|u|\!)_{[]}^\tau \sqcup \{\!\{|o|_l^\tau\}\!\}$. Se concluye apelando nuevamente al Lem. C.0.38, dado que $(\!|o|\!)_l^\tau = (\!|\mathtt{L}|\!)_{[]}^\tau \sqcup (\!|t|\!)_l^\tau \sqcup (\!|u|\!)_{[]}^\tau \sqcup \{\!\{|o|_l^\tau\}\!\} > (\!|\mathtt{L}|\!)_{[]}^\tau \sqcup (\!|t|\!)_l^\tau \sqcup (\!|u|\!)_{[]}^\tau = (\!|\mathtt{L}|\!)_{[]}^\tau \sqcup (\!|t[x \setminus u]|\!)_l^\tau = (\!|o'|\!)_l^\tau$.
  2. $\mathtt{dM}$. Luego, $o = \mathtt{L}\langle \mu\alpha.c\rangle\, u$ y $o' = \mathtt{L}\langle \mu\alpha'.c[\![\alpha \setminus^{\alpha'} u]\!]\rangle$ con α' un nombre fresco. Por Lem. C.0.38, $(\!|o|\!)_l^\tau = (\!|\mathtt{L}\langle \mu\alpha.c\rangle|\!)_{|u|_{[]}^\tau:l}^\tau \sqcup (\!|u|\!)_{[]}^\tau \sqcup \{\!\{|o|_l^\tau\}\!\} = (\!|\mathtt{L}|\!)_{[]}^\tau \sqcup (\!|\mu\alpha.c|\!)_{|u|_{[]}^\tau:l}^\tau \sqcup (\!|u|\!)_{[]}^\tau \sqcup \{\!\{|o|_l^\tau\}\!\} = (\!|\mathtt{L}|\!)_{[]}^\tau \sqcup (\!|c|\!)_{[]}^{\tau[\alpha:=|u|_{[]}^\tau:l]} \sqcup (\!|u|\!)_{[]}^\tau \sqcup \{\!\{|o|_l^\tau\}\!\}$. y, por Lem. C.0.30, $|o|_l^\tau = |\mathtt{L}\langle \mu\alpha.c\rangle|_{|u|_{[]}^\tau:l}^\tau + |u|_{[]}^\tau = |\mu\alpha.c|_{|u|_{[]}^\tau:l}^\tau + |u|_{[]}^\tau = |c|_{[]}^{\tau[\alpha:=|u|_{[]}^\tau:l]} + 1 + |u|_{[]}^\tau$. Más aún, por Lem. C.0.29 (1) con α' fresco, se tienen $|u|_{[]}^\tau = |u|_{[]}^{\tau[\alpha':=l]}$ y $|c|_{[]}^{\tau[\alpha:=|u|_{[]}^\tau:l]} = |c|_{[]}^{\tau[\alpha':=l][\alpha:=|u|_{[]}^{\tau[\alpha':=l]}:l]} = |c[\![\alpha \setminus^{\alpha'} u]\!]|_{[]}^{\tau[\alpha':=l]}$ y por lo tanto $|o|_l^\tau >$

$|c[\![\alpha \setminus^{\alpha'} u]\!]|_{[]}^{\tau[\alpha':=l]}$. Análogamente, por Lem. C.0.37 (1) con α' fresco $(\!|u|\!)_{[]}^{\tau} = (\!|u|\!)_{[]}^{\tau[\alpha':=l]}$ y $(\!|c|\!)_{[]}^{\tau[\alpha:=|u|_{[]}^{\tau}:l]} = (\!|c|\!)_{[]}^{\tau[\alpha':=l][\alpha:=|u|_{[]}^{\tau[\alpha':=l]}:l]}$. Se concluye por Lem. C.0.38 nuevamente, dado que $(\!|o|\!)_l^{\tau} = (\!|\mathtt{L}|\!)_{[]}^{\tau} \sqcup (\!|c|\!)_{[]}^{\tau[\alpha':=l][\alpha:=|u|_{[]}^{\tau[\alpha':=l]}:l]} \sqcup (\!|u|\!)_{[]}^{\tau[\alpha':=l]} \sqcup \{\!\{|o|_l^{\tau}\}\!\} > (\!|\mathtt{L}|\!)_{[]}^{\tau} \sqcup (\!|c|\!)_{[]}^{\tau[\alpha':=l][\alpha:=|u|_{[]}^{\tau[\alpha':=l]}:l]} \sqcup (\!|u|\!)_{[]}^{\tau[\alpha':=l]} \sqcup \{\!\{|c[\![\alpha \setminus^{\alpha'} u]\!]|_{[]}^{\tau[\alpha':=l]}\}\!\} = (\!|\mathtt{L}|\!)_{[]}^{\tau} \sqcup (\!|c[\![\alpha \setminus^{\alpha'} u]\!]|\!)_l^{\tau} = (\!|o'|\!)_l^{\tau}$.

- $\mathtt{O} = \boxdot$. Luego, se tienen tres casos posibles según la regla de reescritura aplicada al reducir.

  1. $\mathtt{N}$. Luego, $o = \mathtt{LCC}\langle[\alpha]\, t\rangle[\![\alpha \setminus^{\alpha'} s]\!]$ y $o' = \mathtt{LCC}\langle[\alpha']\, t :: s\rangle$ con $\alpha \notin t$, $\alpha \notin \mathtt{LCC}$ y $\mathsf{fc}(s, \mathtt{LCC})$. Notar que $o' = \{\!\{\alpha \setminus^{\alpha'} s\}\!\}(\mathtt{LCC}\langle[\alpha]\, t\rangle)$ con α ocurriendo linealmente en $\mathtt{LCC}\langle[\alpha]\, t\rangle$. Se concluye entonces por Cor. C.0.41.
  2. $\mathtt{P}$. Luego, $o = \mathtt{LCC}\langle c[\![\beta \setminus^{\alpha} s']\!]\rangle[\![\alpha \setminus^{\alpha'} s]\!]$ y $o' = \mathtt{LCC}\langle c[\![\beta \setminus^{\alpha'} s' \cdot s]\!]\rangle$ con $\alpha \notin c$, $\alpha \notin \mathtt{LCC}$, $\alpha \notin s'$, $\mathsf{fc}(\alpha', \mathtt{LCC})$ y $\mathsf{fc}(s, \mathtt{LCC})$. Notar que $o' = \{\!\{\alpha \setminus^{\alpha'} s\}\!\}(\mathtt{LCC}\langle c[\![\beta \setminus^{\alpha} s']\!]\rangle)$ con α ocurriendo linealmente en $\mathtt{LCC}\langle c[\![\beta \setminus^{\alpha} s']\!]\rangle$. Se concluye entonces por Cor. C.0.41.
  3. $\mathtt{W}$. Luego, $o = \mathtt{LCC}\langle c[\beta \setminus \alpha]\rangle[\![\alpha \setminus^{\alpha'} s]\!]$ y $o' = \mathtt{LCC}\langle c[\![\beta \setminus^{\alpha} s]\!][\alpha \setminus \alpha']\rangle$ con $\alpha \notin c$, $\alpha \notin \mathtt{LCC}$, $\mathsf{fc}(\alpha', \mathtt{LCC})$ y $\mathsf{fc}(s, \mathtt{LCC})$. Notar que $o' = \{\!\{\alpha \setminus^{\alpha'} s\}\!\}(\mathtt{LCC}\langle c[\beta \setminus \alpha]\rangle)$ con α ocurriendo linealmente en $\mathtt{LCC}\langle c[\beta \setminus \alpha]\rangle$. Se concluye entonces por Cor. C.0.41.

- $\mathtt{O} = \mathtt{T}\, u$. Luego, $o = t\, u$ y $o' = t'\, u$ con $t \to_{\mathfrak{C}} t'$. Por *h.i.* se tiene $(\!|t|\!)_{|u|_{[]}^{\tau}:l}^{\tau} > (\!|t'|\!)_{|u|_{[]}^{\tau}:l}^{\tau}$. Más aún, por Lem. C.0.34, $|o|_l^{\tau} \geq |o'|_l^{\tau}$. Finalmente, se concluye dado que $(\!|o|\!)_l^{\tau} = (\!|t|\!)_{|u|_{[]}^{\tau}:l}^{\tau} \sqcup (\!|u|\!)_{[]}^{\tau} \sqcup \{\!\{|o|_l^{\tau}\}\!\} > (\!|t'|\!)_{|u|_{[]}^{\tau}:l}^{\tau} \sqcup (\!|u|\!)_{[]}^{\tau} \sqcup \{\!\{|o'|_l^{\tau}\}\!\} = (\!|o'|\!)_l^{\tau}$.

- $\mathtt{O} = t\, \mathtt{T}$. Luego, $o = t\, u$ y $o' = t\, u'$ con $u \to_{\mathfrak{C}} u'$. Por *h.i.* se tiene $(\!|u|\!)_{[]}^{\tau} > (\!|u'|\!)_{[]}^{\tau}$. Más aún, por Lem. C.0.34, se tienen $|o|_l^{\tau} \geq |o'|_l^{\tau}$ y $|u|_{[]}^{\tau} \geq |u'|_{[]}^{\tau}$. Entonces, por Lem. C.0.37 (3), se tiene también $(\!|t|\!)_{|u|_{[]}^{\tau}:l}^{\tau} \geq (\!|t|\!)_{|u'|_{[]}^{\tau}:l}^{\tau}$. Finalmente, se concluye dado que $(\!|o|\!)_l^{\tau} = (\!|t|\!)_{|u|_{[]}^{\tau}:l}^{\tau} \sqcup (\!|u|\!)_{[]}^{\tau} \sqcup \{\!\{|o|_l^{\tau}\}\!\} > (\!|t|\!)_{|u'|_{[]}^{\tau}:l}^{\tau} \sqcup (\!|u'|\!)_{[]}^{\tau} \sqcup \{\!\{|o'|_l^{\tau}\}\!\} = (\!|o'|\!)_l^{\tau}$.

- $\mathtt{O} = \lambda x.\mathtt{T}$. Luego, $o = \lambda x.t$ y $o' = \lambda x.t'$ con $t \to_{\mathfrak{C}} t'$. Sea l_0 tal que $l = n : l_0$ o $l = [] = l_0$. Por *h.i.* se tiene $(\!|o|\!)_l^{\tau} = (\!|t|\!)_{l_0}^{\tau} > (\!|t'|\!)_{l_0}^{\tau} = (\!|o'|\!)_l^{\tau}$, por lo que se concluye.

- $\mathtt{O} = \mu\alpha.\mathtt{C}$. Luego, $o = \mu\alpha.c$ y $o' = \mu\alpha.c'$ con $c \to_{\mathfrak{C}} c'$. Por *h.i.* se tiene $(\!|o|\!)_l^{\tau} = (\!|c|\!)_{[]}^{\tau[\alpha:=l]} > (\!|c'|\!)_{[]}^{\tau[\alpha:=l]} = (\!|o'|\!)_l^{\tau}$, por lo que se concluye.

- $\mathtt{O} = \mathtt{T}[x \setminus u]$. Luego, $o = t[x \setminus u]$ y $o' = t'[x \setminus u]$ con $t \to_{\mathfrak{C}} t'$. Por *h.i.* se tiene $(\!|t|\!)_l^{\tau} > (\!|t'|\!)_l^{\tau}$. Se concluye dado que $(\!|o|\!)_l^{\tau} = (\!|t|\!)_l^{\tau} \sqcup (\!|u|\!)_{[]}^{\tau} > (\!|t'|\!)_l^{\tau} \sqcup (\!|u|\!)_{[]}^{\tau} = (\!|o'|\!)_l^{\tau}$, por lo que se concluye.

- $\mathtt{O} = t[x \setminus \mathtt{T}]$. Luego, $o = t[x \setminus u]$ y $o' = t[x \setminus u']$ con $u \to_{\mathfrak{C}} u'$. Por *h.i.* se tiene $(\!|u|\!)_{[]}^{\tau} > (\!|u'|\!)_{[]}^{\tau}$. Se concluye dado que $(\!|o|\!)_l^{\tau} = (\!|t|\!)_l^{\tau} \sqcup (\!|u|\!)_{[]}^{\tau} > (\!|t|\!)_l^{\tau} \sqcup (\!|u'|\!)_{[]}^{\tau} = (\!|o'|\!)_l^{\tau}$, por lo que se concluye.

- $\mathtt{O} = [\alpha]\, \mathtt{T}$. Luego, $o = [\alpha]\, t$ y $o' = [\alpha]\, t'$ con $t \to_{\mathfrak{C}} t'$. Por *h.i.* se tiene $(\!|o|\!)_l^{\tau} = (\!|t|\!)_{\tau(\alpha)}^{\tau} > (\!|t'|\!)_{\tau(\alpha)}^{\tau} = (\!|o'|\!)_l^{\tau}$, por lo que se concluye.

- $\mathtt{O} = \mathtt{C}[\![\alpha \setminus^{\alpha'} s]\!]$. Luego, $o = c[\![\alpha \setminus^{\alpha'} s]\!]$ y $o' = c'[\![\alpha \setminus^{\alpha'} s]\!]$ con $c \to_{\mathfrak{C}} c'$. Sean $s = u_0 \cdot \ldots \cdot u_n$ y $\tau' = \tau[\alpha := |u_0|_{[]}^{\tau} : \ldots : |u_n|_{[]}^{\tau} : \tau(\alpha')]$. Por *h.i.* se tiene $(\!|c|\!)_{[]}^{\tau'} > (\!|c'|\!)_{[]}^{\tau'}$. Más aún, por Lem. C.0.34, $|o|_l^{\tau} \geq |o'|_l^{\tau}$. Finalmente, se concluye dado que $(\!|o|\!)_l^{\tau} = (\!|c|\!)_{[]}^{\tau'} \sqcup (\!|s|\!)_{[]}^{\tau} \sqcup \{\!\{|o|_l^{\tau}\}\!\} > (\!|c'|\!)_{[]}^{\tau'} \sqcup (\!|s|\!)_{[]}^{\tau} \sqcup \{\!\{|o'|_l^{\tau}\}\!\} = (\!|o'|\!)_l^{\tau}$.

- $\mathtt{O} = c[\![\alpha \setminus^{\alpha'} \mathtt{S}]\!]$. Luego, $o = c[\![\alpha \setminus^{\alpha'} s]\!]$ y $o' = c[\![\alpha \setminus^{\alpha'} s']\!]$ con $s \to_{\mathfrak{C}} s'$. Sea $s = u_0 \cdot \ldots \cdot u_n$. Entonces, $s \to_{\mathfrak{C}} s'$ implica $s' = u'_0 \cdot \ldots \cdot u'_n$ con $u_k \to_{\mathfrak{C}} u'_k$ para algún $k \in [0, n]$ y $(u_i = u'_i)_{i \leq n, i \neq k}$. Por *h.i.* se tiene $(\!|s|\!)^{\tau}_{[]} > (\!|s'|\!)^{\tau}_{[]}$. Más aún, por Lem. C.0.34, se tienen $|o|^{\tau}_{l} \geq |o'|^{\tau}_{l}$ y $|u_k|^{\tau}_{[]} \geq |u'_k|^{\tau}_{[]}$. Sean $\tau' = \tau[\alpha := |u_0|^{\tau}_{[]} : \ldots : |u_n|^{\tau}_{[]} : \tau(\alpha')]$ y $\tau'' = \tau[\alpha := |u'_0|^{\tau}_{[]} : \ldots : |u'_n|^{\tau}_{[]} : \tau(\alpha')]$. Se tiene entonces $\tau' \geq \tau''$. Luego, por Lem. C.0.37 (2), $(\!|c|\!)^{\tau'}_{[]} \geq (\!|c|\!)^{\tau''}_{[]}$. Se concluye dado que $(\!|o|\!)^{\tau}_{l} = (\!|c|\!)^{\tau'}_{[]} \sqcup (\!|s|\!)^{\tau}_{[]} \sqcup \{\!\{|o|^{\tau}_{l}\}\!\} > (\!|c|\!)^{\tau''}_{[]} \sqcup (\!|s'|\!)^{\tau}_{[]} \sqcup \{\!\{|o'|^{\tau}_{l}\}\!\} = (\!|o'|\!)^{\tau}_{l}$.

- $\mathtt{O} = \mathtt{C}[\alpha \setminus \beta]$. Luego, $o = c[\alpha \setminus \beta]$ y $o' = c'[\alpha \setminus \beta]$ con $c \to_{\mathfrak{C}} c'$. Por *h.i.* se tiene $(\!|o|\!)^{\tau}_{l} = (\!|c|\!)^{\tau[\alpha:=\tau(\beta)]}_{[]} > (\!|c'|\!)^{\tau[\alpha:=\tau(\beta)]}_{[]} = (\!|o'|\!)^{\tau}_{l}$, por lo que se concluye.

- $\mathtt{O} = \mathtt{T} \cdot s$. Luego, $o = t \cdot s$ y $o' = t' \cdot s$ con $t \to_{\mathfrak{C}} t'$. Por *h.i.* se tiene $(\!|t|\!)^{\tau}_{[]} > (\!|t'|\!)^{\tau}_{[]}$. Se concluye dado que $(\!|o|\!)^{\tau}_{l} = (\!|t|\!)^{\tau}_{[]} \sqcup (\!|s|\!)^{\tau}_{[]} > (\!|t'|\!)^{\tau}_{[]} \sqcup (\!|s|\!)^{\tau}_{[]} = (\!|o'|\!)^{\tau}_{l}$.

- $\mathtt{O} = t \cdot \mathtt{S}$. Luego, $o = t \cdot s$ y $o' = t \cdot s'$ con $s \to_{\mathfrak{C}} s'$. Por *h.i.* se tiene $(\!|s|\!)^{\tau}_{[]} > (\!|s'|\!)^{\tau}_{[]}$. Se concluye dado que $(\!|o|\!)^{\tau}_{l} = (\!|t|\!)^{\tau}_{[]} \sqcup (\!|s|\!)^{\tau}_{[]} > (\!|t|\!)^{\tau}_{[]} \sqcup (\!|s'|\!)^{\tau}_{[]} = (\!|o'|\!)^{\tau}_{l}$.

□

Volviendo sobre el ejemplo dado anteriormente con el término $t = (\mu\alpha.([\beta]\, x)[\![\beta \setminus^{\alpha} \mu\delta.[\delta]\, x]\!])\, (\mu\delta.[\delta]\, x)$, donde $|\mu\delta.[\delta]\, x|^{\tau}_{[]} = 2$ y $|t|^{\tau}_{[]} = 8$, considerar la siguiente secuencia de reducción canónica:

$$\begin{array}{lll} (\mu\alpha.([\beta]\, x)[\![\beta \setminus^{\alpha} \mu\delta.[\delta]\, x]\!])\, (\mu\delta.[\delta]\, x) & \to_{\mathtt{dM}} & \mu\alpha'.([\beta]\, x)[\![\beta \setminus^{\alpha} \mu\delta.[\delta]\, x]\!][\![\alpha \setminus^{\alpha'} \mu\delta.[\delta]\, x]\!] \\ & \to_{\mathtt{P}} & \mu\alpha'.([\beta]\, x)[\![\beta \setminus^{\alpha'} (\mu\delta.[\delta]\, x) \cdot (\mu\delta.[\delta]\, x)]\!] \\ & \to_{\mathtt{N}} & \mu\alpha'.[\beta]\, x\, (\mu\delta.[\delta]\, x)\, (\mu\delta.[\delta]\, x) \end{array}$$

Luego, $(\!|t|\!)^{\tau}_{[]} = \{\!\{5, 8\}\!\}$, mientras que $(\!|\mu\alpha'.([\beta]\, x)[\![\beta \setminus^{\alpha} \mu\delta.[\delta]\, x]\!][\![\alpha \setminus^{\alpha'} \mu\delta.[\delta]\, x]\!]|\!)^{\tau}_{[]} = \{\!\{5, 5\}\!\}$. Más aún, $(\!|\mu\alpha'.([\beta]\, x)[\![\beta \setminus^{\alpha'} (\mu\delta.[\delta]\, x) \cdot (\mu\delta.[\delta]\, x)]\!]|\!)^{\tau}_{[]} = \{\!\{5\}\!\}$ y por último $(\!|\mu\alpha'.[\beta]\, x\, (\mu\delta.[\delta]\, x)\, (\mu\delta.[\delta]\, x)|\!)^{\tau}_{[]} = \{\!\{4, 2\}\!\}$.

Se prueba entonces la normalización fuerte de la relación de reducción canónica apelando a los resultados anteriormente presentados.

Teorema 4.5.1. *La relación de reducción $\to_{\mathfrak{C}}$ es fuertemente normalizante (SN).*

Demostración. El resultado es una consecuencia inmediata del Lem. C.0.42 y el Teo. C.0.36. □

Teorema 4.5.2. *La relación de reducción $\to_{\mathfrak{C}}$ es confluente (CR).*

Demostración. Se prueba que la relación es *débilmente confluente* (WCR) mediante un análisis de pares críticos [BN98] y se concluye por Lema de Newman apelando al Teo. 4.5.1 anterior.

Luego, basta ver que todos los pares críticos entre las reglas `dB`, `dM`, `N`, `P`, `W` convergen. Se tienen únicamente tres pares críticos:

1. `N`–`P`. Luego, se tiene $o = \mathtt{LCC}_2\langle \mathtt{LCC}_1\langle [\delta]\, t\rangle [\![\delta \setminus^{\alpha} s']\!]\rangle [\![\alpha \setminus^{\alpha'} s]\!]$ con las condiciones $\alpha \notin \mathtt{LCC}_2$, $\alpha \notin \mathtt{LCC}_1\langle [\delta]\, t\rangle$, $\alpha \notin s$, $\mathtt{fc}(\alpha', \mathtt{LCC}_2)$, $\mathtt{fc}(s, \mathtt{LCC}_2)$ por la regla `P`, y las condiciones $\delta \notin t$, $\delta \notin \mathtt{LCC}_1$, $\mathtt{fc}(\alpha, \mathtt{LCC}_1)$ y $\mathtt{fc}(s', \mathtt{LCC}_1)$ para la regla `N`. Se concluye dado que $(t :: s') :: s = t :: (s' \cdot s)$, obteniendo el siguiente diagrama:

$$\begin{array}{ccc} \mathtt{LCC}_2\langle \mathtt{LCC}_1\langle [\delta]\, t\rangle [\![\delta \setminus^{\alpha} s']\!]\rangle [\![\alpha \setminus^{\alpha'} s]\!] & \to_{\mathtt{N}} & \mathtt{LCC}_2\langle \mathtt{LCC}_1\langle [\alpha]\, t :: s'\rangle\rangle [\![\alpha \setminus^{\alpha'} s]\!] \\ \downarrow_{\mathtt{P}} & & \downarrow_{\mathtt{N}} \\ \mathtt{LCC}_2\langle \mathtt{LCC}_1\langle [\delta]\, t\rangle [\![\delta \setminus^{\alpha'} s' \cdot s]\!]\rangle & \to_{\mathtt{N}} & \mathtt{LCC}_2\langle \mathtt{LCC}_1\langle [\alpha']\, (t :: s') :: s\rangle\rangle \end{array}$$

2. $\mathtt{P}$–$\mathtt{P}$. Luego, se tiene $o = \mathtt{LCC}_2\langle \mathtt{LCC}_1\langle c[\![\gamma \setminus^{\delta} s'']\!]\rangle [\![\delta \setminus^{\alpha} s']\!]\rangle [\![\alpha \setminus^{\alpha'} s]\!]$ con las condiciones $\alpha \notin \mathtt{LCC}_2$, $\alpha \notin \mathtt{LCC}_1\langle c[\![\gamma \setminus^{\delta} s'']\!]\rangle$, $\alpha \notin s$, $\mathsf{fc}(\alpha', \mathtt{LCC}_2)$, $\mathsf{fc}(s, \mathtt{LCC}_2)$ por la aplicación externa de la regla, y las condiciones $\delta \notin c$, $\delta \notin \mathtt{LCC}_1$, $\delta \notin s'$, $\mathsf{fc}(\alpha, \mathtt{LCC}_1)$ y $\mathsf{fc}(s', \mathtt{LCC}_1)$ para la interna. Se concluye dado que $s'' \cdot (s' \cdot s) = (s'' \cdot s') \cdot s$, obteniendo el siguiente diagrama:

$$\begin{array}{ccc} \mathtt{LCC}_2\langle \mathtt{LCC}_1\langle c[\![\gamma \setminus^{\delta} s'']\!]\rangle [\![\delta \setminus^{\alpha} s']\!]\rangle [\![\alpha \setminus^{\alpha'} s]\!] & \to_{\mathtt{P}} & \mathtt{LCC}_2\langle \mathtt{LCC}_1\langle c[\![\gamma \setminus^{\alpha} s'' \cdot s']\!]\rangle\rangle [\![\alpha \setminus^{\alpha'} s]\!] \\ \downarrow_{\mathtt{P}} & & \downarrow_{\mathtt{P}} \\ \mathtt{LCC}_2\langle \mathtt{LCC}_1\langle c[\![\gamma \setminus^{\delta} s'']\!]\rangle [\![\delta \setminus^{\alpha'} s' \cdot s]\!]\rangle & \to_{\mathtt{P}} & \mathtt{LCC}_2\langle \mathtt{LCC}_1\langle c[\![\gamma \setminus^{\alpha'} s'' \cdot s' \cdot s]\!]\rangle\rangle \end{array}$$

3. $\mathtt{W}$–$\mathtt{P}$. Luego, se tiene $o = \mathtt{LCC}_2\langle \mathtt{LCC}_1\langle c[\gamma \setminus \delta]\rangle [\![\delta \setminus^{\alpha} s']\!]\rangle [\![\alpha \setminus^{\alpha'} s]\!]$ con las condiciones $\alpha \notin \mathtt{LCC}_2$, $\alpha \notin \mathtt{LCC}_1\langle c[\gamma \setminus \delta]\rangle$, $\alpha \notin s$, $\mathsf{fc}(\alpha', \mathtt{LCC}_2)$, $\mathsf{fc}(s, \mathtt{LCC}_2)$ por la regla $\mathtt{P}$, y las condiciones $\delta \notin c$, $\delta \notin \mathtt{LCC}_1$, $\mathsf{fc}(\alpha, \mathtt{LCC}_1)$ y $\mathsf{fc}(s', \mathtt{LCC}_1)$ para la regla $\mathtt{W}$. Se concluye por α-conversion con β un nombre fresco, obteniendo el siguiente diagrama:

$$\begin{array}{ccc} \mathtt{LCC}_2\langle \mathtt{LCC}_1\langle c[\gamma \setminus \delta]\rangle [\![\delta \setminus^{\alpha} s']\!]\rangle [\![\alpha \setminus^{\alpha'} s]\!] & \to_{\mathtt{W}} & \mathtt{LCC}_2\langle \mathtt{LCC}_1\langle c[\![\gamma \setminus^{\delta} s']\!][\delta \setminus \alpha]\rangle\rangle [\![\alpha \setminus^{\alpha'} s]\!] \\ & & \downarrow_{\mathtt{W}} \\ \downarrow_{\mathtt{P}} & & \mathtt{LCC}_2\langle \mathtt{LCC}_1\langle c[\![\gamma \setminus^{\delta} s']\!][\![\delta \setminus^{\alpha} s]\!][\alpha \setminus \alpha']\rangle\rangle \\ & & \downarrow_{\mathtt{P}} \\ \mathtt{LCC}_2\langle \mathtt{LCC}_1\langle c[\gamma \setminus \delta]\rangle [\![\delta \setminus^{\alpha'} s' \cdot s]\!]\rangle & \to_{\mathtt{W}} & \mathtt{LCC}_2\langle \mathtt{LCC}_1\langle c[\![\gamma \setminus^{\beta} s' \cdot s]\!][\beta \setminus \alpha']\rangle\rangle \end{array}$$

□

Lema C.0.43 (Subject Reduction)**.** *Sea $o \in \mathbb{O}_{\Lambda M}$. Si $\pi \rhd_{\mathcal{E}} \Gamma \vdash o : T \mid \Delta$ y $o \to_{\mathfrak{C}} o'$, entonces existe $\pi' \rhd_{\mathcal{E}} \Gamma \vdash o' : T \mid \Delta$.*

Demostración. Por definición, $o \to_{\mathfrak{C}} o'$ implica $o = \mathtt{O}\langle l\rangle$ y $o' = \mathtt{O}\langle r\rangle$ con $l \mapsto_* r$, $* \in \{\mathtt{dB}, \mathtt{dM}, \mathtt{N}, \mathtt{P}, \mathtt{W}\}$. La demostración es por inducción en $\mathtt{O}$.

- $\mathtt{O} = \Box$. Luego, se tienen dos casos posibles según la regla de reescritura aplicada al reducir.

  1. $\mathtt{dB}$. Luego, $o = \mathtt{L}\langle \lambda x.t\rangle\, u$ y $o' = \mathtt{L}\langle t[x \setminus u]\rangle$. Por (APP) se tienen $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta = \Delta_0 \cup \Delta_1$ con $\pi_{\mathtt{L}\langle \lambda x.t\rangle} \rhd_{\mathcal{E}} \Gamma_0 \vdash \mathtt{L}\langle \lambda x.t\rangle : A \to T \mid \Delta_0$ y $\pi_u \rhd_{\mathcal{E}} \Gamma_1 \vdash u : A \mid \Delta_1$. Se procede por inducción en $\mathtt{L}$.
     - $\mathtt{L} = \Box$. Luego, de $\pi_{\mathtt{L}\langle \lambda x.t\rangle}$ se tiene $\pi_t \rhd_{\mathcal{E}} \Gamma_0; (x : A)^{\leq 1} \vdash t : T \mid \Delta_0$ por (ABS). Se concluye por (SUBS) con π_t y π_u, obteniendo $\pi' \rhd_{\mathcal{E}} \Gamma \vdash t[x \setminus u] : T \mid \Delta$.
     - $\mathtt{L} = \mathtt{L}'[y \setminus v]$ con $y \notin u$. Luego, por Lem. 4.3.1, $y \notin \mathsf{dom}(\Gamma_1)$. De $\pi_{\mathtt{L}\langle \lambda x.t\rangle}$ por (SUBS) se tienen $\Gamma_0 = \Gamma_0' \cup \Gamma_0''$ y $\Delta_0 = \Delta_0' \cup \Delta_0''$ con $\pi_{\mathtt{L}'\langle \lambda x.t\rangle} \rhd_{\mathcal{E}} \Gamma_0'; (y : B)^{\leq 1} \vdash \mathtt{L}'\langle \lambda x.t\rangle : A \to T \mid \Delta_0'$ y $\pi_v \rhd_{\mathcal{E}} \Gamma_0'' \vdash v : B \mid \Delta_0''$. Luego, por (APP) con $\pi_{\mathtt{L}'\langle \lambda x.t\rangle}$ y π_u se obtiene $\pi_{\mathtt{L}'\langle \lambda x.t\rangle\, u} \rhd_{\mathcal{E}} \Gamma_0' \cup \Gamma_1; (y : B)^{\leq 1} \vdash \mathtt{L}'\langle \lambda x.t\rangle\, u : T \mid \Delta_0' \cup \Delta_1$. Por *h.i.* existe una derivación $\pi_{\mathtt{L}'\langle t[x \setminus u]\rangle} \rhd_{\mathcal{E}} \Gamma_0' \cup \Gamma_1; (y : B)^{\leq 1} \vdash \mathtt{L}'\langle t[x \setminus u]\rangle : T \mid \Delta_0' \cup \Delta_1$. Finalmente, se concluye por (SUBS) con π_v, $\pi' \rhd_{\mathcal{E}} \Gamma \vdash \mathtt{L}\langle t[x \setminus u]\rangle : T \mid \Delta$.
  2. $\mathtt{dM}$. Luego, $o = \mathtt{L}\langle \mu\alpha.c\rangle\, u$ y $o' = \mathtt{L}\langle \mu\alpha'.c[\![\alpha \setminus^{\alpha'} u]\!]\rangle$ con α' un nombre fresco. Por (APP) se tienen $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta = \Delta_0 \cup \Delta_1$ con $\pi_{\mathtt{L}\langle \mu\alpha.c\rangle} \rhd_{\mathcal{E}} \Gamma_0 \vdash \mathtt{L}\langle \mu\alpha.c\rangle : A \to T \mid \Delta_0$ y $\pi_u \rhd_{\mathcal{E}} \Gamma_1 \vdash u : A \mid \Delta_1$. Se procede por inducción en $\mathtt{L}$.
     - $\mathtt{L} = \Box$. Luego, de $\pi_{\mathtt{L}\langle \mu\alpha.c\rangle}$ se tiene $\pi_c \rhd_{\mathcal{E}} \Gamma_1 \vdash c \mid \Delta_1; (\alpha : A \to T)^{\leq 1}$ por (CONT). Por (REPL) con π_c y π_u se obtiene $\pi_{c[\![\alpha \setminus^{\alpha'} u]\!]} \rhd_{\mathcal{E}} \Gamma \vdash c[\![\alpha \setminus^{\alpha'} u]\!] \mid \Delta; \alpha' : T$. Luego, se concluye por (CONT), $\pi' \rhd_{\mathcal{E}} \Gamma \vdash \mu\alpha'.c[\![\alpha \setminus^{\alpha'} u]\!] : T \mid \Delta$.

- $\mathtt{L} = \mathtt{L}'[y \setminus v]$ con $y \notin u$. Luego, por Lem. 4.3.1, $y \notin \mathsf{dom}(\Gamma_1)$. De $\pi_{\mathtt{L}\langle\mu\alpha.c\rangle}$ por (SUBS) se tienen contextos $\Gamma_0 = \Gamma_0' \cup \Gamma_0''$ y $\Delta_0 = \Delta_0' \cup \Delta_0''$ con derivaciones $\pi_{\mathtt{L}'\langle\mu\alpha.c\rangle} \rhd_{\mathcal{E}} \Gamma_0'; (y : B)^{\leq 1} \vdash \mathtt{L}'\langle\mu\alpha.c\rangle : A \to T \mid \Delta_0'$ y $\pi_v \rhd_{\mathcal{E}} \Gamma_0'' \vdash v : B \mid \Delta_0''$. Por (APP) con π_u, se obtiene la derivación $\pi_{\mathtt{L}'\langle\mu\alpha.c\rangle\, u} \rhd_{\mathcal{E}} \Gamma_0' \cup \Gamma_1; (y : B)^{\leq 1} \vdash \mathtt{L}'\langle\mu\alpha.c\rangle\, u : T \mid \Delta_0' \cup \Delta_1$. Entonces, por *h.i.* existe $\pi_{\mathtt{L}'\langle\mu\alpha'.c[\![\alpha \setminus^{\alpha'} u]\!]\rangle} \rhd_{\mathcal{E}} \Gamma_0' \cup \Gamma_1; (y : B)^{\leq 1} \vdash \mathtt{L}'\langle\mu\alpha'.c[\![\alpha \setminus^{\alpha'} u]\!]\rangle : T \mid \Delta_0' \cup \Delta_1$. Finalmente, se concluye por (SUBS) con π_v, $\pi' \rhd_{\mathcal{E}} \Gamma \vdash \mathtt{L}\langle\mu\alpha'.c[\![\alpha \setminus^{\alpha'} u]\!]\rangle : T \mid \Delta$.

- $\mathtt{O} = \boxdot$. Luego, se tienen tres casos posibles según la regla de reescritura aplicada al reducir.

  1. $\mathtt{N}$. Luego, $o = \mathtt{LCC}\langle[\alpha]\, t\rangle[\![\alpha \setminus^{\alpha'} s]\!]$ y $o' = \mathtt{LCC}\langle[\alpha']\, t :: s\rangle$. Por (REPL) se tienen $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta = \Delta_0 \cup \Delta_1; \alpha' : A$ con derivaciones $\pi_{\mathtt{LCC}\langle[\alpha]\, t\rangle} \rhd_{\mathcal{E}} \Gamma_0 \vdash \mathtt{LCC}\langle[\alpha]\, t\rangle \mid \Delta_0; (\alpha : S \to A)^{\leq 1}; (\alpha' : A)^{\leq 1}$ y $\pi_s \rhd_{\mathcal{E}} \Gamma_1 \vdash s : S \mid \Delta_1; (\alpha' : A)^{\leq 1}$. Se procede por inducción en $\mathtt{LCC}$.

     - $\mathtt{LCC} = \boxdot$. Luego, por (NAME) se tiene $(\alpha : S \to A)^{\leq 1}$ no vacío en $\pi_{\mathtt{LCC}\langle[\alpha]\, t\rangle}$ con $\pi_t \rhd_{\mathcal{E}} \Gamma_0 \vdash t : S \to A \mid \Delta_0; (\alpha : S \to A)^{\leq 1}; (\alpha' : A)^{\leq 1}$. Más aún, por hipótesis de la regla $\mathtt{N}$, se tiene $\alpha \notin t$. Luego, por Lem. 4.3.1, $(\alpha : S \to A)^{\leq 1}$ es vacío en π_t. Por Lem. C.0.20 (3), existe $\pi_{t::s} \rhd_{\mathcal{E}} \Gamma \vdash t :: s : A \mid \Delta_0 \cup \Delta_1; (\alpha' : A)^{\leq 1}$. Luego, se concluye por (NAME), $\pi_{o'} \rhd_{\mathcal{E}} \Gamma \vdash [\alpha']\, t :: s \mid \Delta$.
     - $\mathtt{LCC} = [\gamma]\, \mathtt{LTC}$. Luego, por (NAME), se tiene el contexto $\Delta_0 = \Delta_2; \gamma : B$ con la derivación $\pi_{\mathtt{LTC}\langle[\alpha]\, t\rangle} \rhd_{\mathcal{E}} \Gamma_0 \vdash \mathtt{LTC}\langle[\alpha]\, t\rangle : B \mid \Delta_2; (\alpha : S \to A)^{\leq 1}; ((\alpha' : A)^{\leq 1} \cup (\gamma : B)^{\leq 1})$. Se procede analizando la forma de $\mathtt{LTC}$.
       - $\mathtt{LTC} = \mathtt{LTC}'\, u$. Luego, por (APP), se tienen contextos $\Gamma_0 = \Gamma_0' \cup \Gamma_0''$ y $\Delta_2 = \Delta_2' \cup \Delta_2''$ con $\pi_{\mathtt{LTC}'\langle[\alpha]\, t\rangle} \rhd_{\mathcal{E}} \Gamma_0' \vdash \mathtt{LTC}'\langle[\alpha]\, t\rangle : B' \to B \mid \Delta_2'; (\alpha : S \to A)^{\leq 1}; ((\alpha' : A)^{\leq 1} \cup (\gamma : B)^{\leq 1})$ y $\pi_u \rhd_{\mathcal{E}} \Gamma_0'' \vdash u : B' \mid \Delta_2''; (\alpha : S \to A)^{\leq 1}; ((\alpha' : A)^{\leq 1} \cup (\gamma : B)^{\leq 1})$. Considerar $p = \mathtt{LCC}'\langle[\alpha]\, t\rangle[\![\alpha \setminus^{\alpha'} s]\!]$ donde $\mathtt{LCC}' = [\delta]\, \mathtt{LTC}'$ con δ un nombre fresco. Entonces, por (NAME) y (REPL) con π_s, $\pi_p \rhd_{\mathcal{E}} \Gamma_0' \cup \Gamma_1 \vdash p \mid \Delta_1 \cup \Delta_2'; \delta : B' \to B; (\alpha' : A \cup (\gamma : B)^{\leq 1})$. Más aún, como $p \to_{\mathtt{N}} p' = \mathtt{LCC}'\langle[\alpha']\, t :: s\rangle$, por *h.i.* existe una derivación $\pi_{p'} \rhd_{\mathcal{E}} \Gamma_0' \cup \Gamma_1 \vdash p' \mid \Delta_1 \cup \Delta_2'; \delta : B' \to B; (\alpha' : A \cup (\gamma : B)^{\leq 1})$. Luego, de (NAME), se tiene $\pi_{\mathtt{LTC}'\langle[\alpha']\, t::s\rangle} \rhd_{\mathcal{E}} \Gamma_0' \cup \Gamma_1 \vdash \mathtt{LTC}'\langle[\alpha']\, t :: s\rangle : B' \to B \mid \Delta_1 \cup \Delta_2'; (\alpha' : A \cup (\gamma : B)^{\leq 1})$, ya que δ es fresco. Finalmente, se concluye por (APP) con π_u y luego (NAME) con γ, $\pi_{o'} \rhd_{\mathcal{E}} \Gamma \vdash o' \mid \Delta$. Notar que, por hipótesis de la regla $\mathtt{N}$, $\alpha \notin u$ y, por Lem. 4.3.1, $(\alpha : S \to A)^{\leq 1}$ es vacío en π_u.
       - $\mathtt{LTC} = \lambda x.\mathtt{LTC}'$. Luego, por (ABS), se tienen $B = B' \to B''$ con $\pi_{\mathtt{LTC}'\langle[\alpha]\, t\rangle} \rhd_{\mathcal{E}} \Gamma_0; (x : B')^{\leq 1} \vdash \mathtt{LTC}'\langle[\alpha]\, t\rangle : B'' \mid \Delta_2; (\alpha : S \to A)^{\leq 1}; ((\alpha' : A)^{\leq 1} \cup (\gamma : B)^{\leq 1})$. Considerar $p = \mathtt{LCC}'\langle[\alpha]\, t\rangle[\![\alpha \setminus^{\alpha'} s]\!]$ donde $\mathtt{LCC}' = [\delta]\, \mathtt{LTC}'$ con δ un nombre fresco. Entonces, por (NAME) con el nombre δ y (REPL) con π_s, se obtiene la derivación $\pi_p \rhd_{\mathcal{E}} \Gamma; (x : B')^{\leq 1} \vdash p \mid \Delta_1 \cup \Delta_2; \delta : B''; (\alpha' : A \cup (\gamma : B)^{\leq 1})$. Más aún, como $p \to_{\mathtt{N}} p' = \mathtt{LCC}'\langle[\alpha']\, t :: s\rangle$, aplicando la *h.i.* se tiene que existe una derivación $\pi_{p'} \rhd_{\mathcal{E}} \Gamma; (x : B')^{\leq 1} \vdash p' \mid \Delta_1 \cup \Delta_2; \delta : B''; (\alpha' : A \cup (\gamma : B)^{\leq 1})$. Luego, de (NAME), se tiene $\pi_{\mathtt{LTC}'\langle[\alpha']\, t::s\rangle} \rhd_{\mathcal{E}} \Gamma; (x : B')^{\leq 1} \vdash \mathtt{LTC}'\langle[\alpha']\, t :: s\rangle : B'' \mid \Delta_1 \cup \Delta_2; (\alpha' : A \cup (\gamma : B)^{\leq 1})$, ya que δ es fresco. Finalmente, se concluye por (ABS) y (NAME) con γ, $\pi_{o'} \rhd_{\mathcal{E}} \Gamma \vdash o' \mid \Delta$.
       - $\mathtt{LTC} = \mu\delta.\mathtt{LCC}'$. Luego, de $\pi_{\mathtt{LTC}\langle[\alpha]\, t\rangle}$ por (CONT), se tiene la derivación $\pi_{\mathtt{LCC}'\langle[\alpha]\, t\rangle} \rhd_{\mathcal{E}} \Gamma_0 \vdash \mathtt{LCC}'\langle[\alpha]\, t\rangle \mid \Delta_0'; (\alpha : S \to A)^{\leq 1}; ((\alpha' : A)^{\leq 1} \cup (\gamma : B)^{\leq 1}); (\delta : B)^{\leq 1}$. Por lo tanto, aplicando la regla (REPL) con π_s, se obtiene la derivación $\pi_{\mathtt{LCC}'\langle[\alpha]\, t\rangle[\![\alpha \setminus^{\alpha'} s]\!]} \rhd_{\mathcal{E}} \Gamma \vdash \mathtt{LCC}'\langle[\alpha]\, t\rangle[\![\alpha \setminus^{\alpha'} s]\!] \mid \Delta; (\alpha' : A \cup (\gamma : B)^{\leq 1}); (\delta : B)^{\leq 1}$. Por *h.i.* existe una derivación $\pi_{\mathtt{LCC}'\langle[\alpha']\, t::s\rangle} \rhd_{\mathcal{E}} \Gamma \vdash \mathtt{LCC}'\langle[\alpha']\, t :: s\rangle \mid \Delta; (\alpha' : A \cup (\gamma : B)^{\leq 1}); (\delta : B)^{\leq 1}$. Finalmente, se concluye por (CONT) y (NAME), $\pi_{o'} \rhd_{\mathcal{E}} \Gamma \vdash o' \mid \Delta$.
       - $\mathtt{LTC} = \mathtt{LTC}'[x \setminus u]$. Luego, por (SUBS), $\Gamma_0 = \Gamma_0' \cup \Gamma_0''$ y $\Delta_2 = \Delta_2' \cup \Delta_2''$ con $\pi_{\mathtt{LTC}'\langle[\alpha]\, t\rangle} \rhd_{\mathcal{E}} \Gamma_0'; (x : B')^{\leq 1} \vdash \mathtt{LTC}'\langle[\alpha]\, t\rangle : B \mid \Delta_2'; (\alpha : S \to A)^{\leq 1}; ((\alpha' : A)^{\leq 1} \cup (\gamma : B)^{\leq 1})$ y $\pi_u \rhd_{\mathcal{E}} \Gamma_0'' \vdash u : B' \mid \Delta_2''; (\alpha : S \to A)^{\leq 1}; ((\alpha' : A)^{\leq 1} \cup (\gamma : B)^{\leq 1})$. Más aún, por hipótesis de

la regla N, se tiene $\alpha \notin u$. Luego, por Lem. 4.3.1, $(\alpha : S \to A)^{\leq 1}$ es vacío en π_u. Considerar $p = \mathtt{LCC}'\langle[\alpha]\, t\rangle[\![\alpha \setminus^{\alpha'} s]\!]$ donde $\mathtt{LCC}' = [\gamma]\,\mathtt{LTC}'$. Entonces, por (NAME) con γ y (REPL) con π_s, se tiene $\pi_p \rhd_{\mathcal{E}} \Gamma'_0 \cup \Gamma_1; (x : B')^{\leq 1} \vdash p \mid \Delta_1 \cup \Delta'_2; (\alpha' : A \cup \gamma : B)$. Más aún, como $p \to_{\mathtt{N}} p' = \mathtt{LCC}'\langle[\alpha']\, t :: s\rangle$, por *h.i.* existe una derivación $\pi_{p'} \rhd_{\mathcal{E}} \Gamma'_0 \cup \Gamma_1; (x : B')^{\leq 1} \vdash p' \mid \Delta_1 \cup \Delta'_2; (\alpha' : A \cup \gamma : B)$. Esta derivación necesariamente termina en una aplicación de (NAME) con γ. Luego, se tiene la premisa $\pi_{\mathtt{LTC}'\langle[\alpha']\, t::s\rangle} \rhd_{\mathcal{E}} \Gamma'_0 \cup \Gamma_1; (x : B')^{\leq 1} \vdash \mathtt{LTC}'\langle[\alpha']\, t :: s\rangle : B \mid \Delta_1 \cup \Delta'_2; (\alpha' : A \cup (\gamma : B)^{\leq 1})$. Finalmente, se concluye por (SUBS) con π_u y luego (NAME), $\pi_{o'} \rhd_{\mathcal{E}} \Gamma \vdash o' \mid \Delta$.

- $\mathtt{LCC} = \mathtt{LCC}'[\![\gamma \setminus^{\delta} s']\!]$. Luego, de $\pi_{\mathtt{LCC}\langle[\alpha]\, t\rangle}$ por la regla (REPL), se tienen los contextos $\Gamma_0 = \Gamma'_0 \cup \Gamma''_0$ y $\Delta_0 = \Delta'_0 \cup \Delta''_0; \delta : B$ con una derivación para $\pi_{\mathtt{LCC}'\langle[\alpha]\, t\rangle}$ por un lado $\pi_{\mathtt{LCC}'\langle[\alpha]\, t\rangle} \rhd_{\mathcal{E}} \Gamma'_0 \vdash \mathtt{LCC}'\langle[\alpha]\, t\rangle \mid \Delta'_0; (\alpha : S \to A)^{\leq 1}; (\gamma : S' \to B)^{\leq 1}; ((\alpha' : A)^{\leq 1} \cup (\delta : B)^{\leq 1})$ y otra para s' por otro $\pi_{s'} \rhd_{\mathcal{E}} \Gamma''_0 \vdash s' : S' \mid \Delta''_0; (\alpha : S \to A)^{\leq 1}; ((\alpha' : A)^{\leq 1} \cup (\delta : B)^{\leq 1})$. Entonces, aplicando la regla (REPL) con π_s, se obtiene la derivación $\pi_{\mathtt{LCC}'\langle[\alpha]\, t\rangle[\![\alpha \setminus^{\alpha'} s]\!]} \rhd_{\mathcal{E}} \Gamma'_0 \cup \Gamma_1 \vdash \mathtt{LCC}'\langle[\alpha]\, t\rangle[\![\alpha \setminus^{\alpha'} s]\!] \mid \Delta'_0 \cup \Delta_1; (\gamma : S' \to B)^{\leq 1}; ((\alpha' : A)^{\leq 1} \cup (\delta : B)^{\leq 1})$. Luego, por *h.i.* sobre $\mathtt{LCC}'$, se garantiza que existe una derivación para el reducto $\pi_{\mathtt{LCC}'\langle[\alpha']\, t::s\rangle} \rhd_{\mathcal{E}} \Gamma'_0 \cup \Gamma_1 \vdash \mathtt{LCC}'\langle[\alpha']\, t :: s\rangle \mid \Delta'_0 \cup \Delta_1; (\gamma : S' \to B)^{\leq 1}; ((\alpha' : A)^{\leq 1} \cup (\delta : B)^{\leq 1})$. Finalmente, se concluye por (REPL) con $\pi_{s'}$, $\pi_{o'} \rhd_{\mathcal{E}} \Gamma \vdash o' \mid \Delta$. Notar que, por hipótesis de la regla N, $\alpha \notin s'$ y, por Lem. 4.3.1, $(\alpha : S \to A)^{\leq 1}$ es vacío en $\pi_{s'}$.
- $\mathtt{LCC} = \mathtt{LCC}'[\gamma \backslash \delta]$. Luego, por (REN), se tiene $\Delta_0 = \Delta'_0; \delta : B$ con la derivación $\pi_{\mathtt{LCC}'\langle[\alpha]\, t\rangle} \rhd_{\mathcal{E}} \Gamma_0 \vdash \mathtt{LCC}'\langle[\alpha]\, t\rangle \mid \Delta_0; (\alpha : S \to A)^{\leq 1}; ((\alpha' : A)^{\leq 1} \cup (\delta : B)^{\leq 1}); (\gamma : B)^{\leq 1}$. Más aún, combinando esta derivación con π_s mediante la regla (REPL), se obtiene $\pi_{\mathtt{LCC}'\langle[\alpha]\, t\rangle[\![\alpha \setminus^{\alpha'} s]\!]} \rhd_{\mathcal{E}} \Gamma_0 \cup \Gamma_1 \vdash \mathtt{LCC}'\langle[\alpha]\, t\rangle[\![\alpha \setminus^{\alpha'} s]\!] \mid \Delta'_0 \cup \Delta_1; (\alpha' : A \cup (\delta : B)^{\leq 1}); (\gamma : B)^{\leq 1}$. Luego, aplicando *h.i.* sobre $\mathtt{LCC}'$, se tiene que existe una derivación para el reducto $\pi_{\mathtt{LCC}'\langle[\alpha']\, t::s\rangle} \rhd_{\mathcal{E}} \Gamma_0 \cup \Gamma_1 \vdash \mathtt{LCC}'\langle[\alpha']\, t :: s\rangle \mid \Delta'_0 \cup \Delta_1; (\alpha' : A \cup (\delta : B)^{\leq 1}); (\gamma : B)^{\leq 1}$. Finalmente, se concluye por (REN), $\pi_{o'} \rhd_{\mathcal{E}} \Gamma \vdash o' \mid \Delta$.

2. P. Luego, $o = \mathtt{LCC}\langle c[\![\beta \setminus^{\alpha} s']\!]\rangle[\![\alpha \setminus^{\alpha'} s]\!]$ y $o' = \mathtt{LCC}\langle c[\![\beta \setminus^{\alpha'} s' \cdot s]\!]\rangle$. Por (REPL), se tienen los contextos $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta = \Delta_0 \cup \Delta_1; \alpha' : A$ con las derivaciones $\pi_{\mathtt{LCC}\langle c[\![\beta \setminus^{\alpha} s']\!]\rangle} \rhd_{\mathcal{E}} \Gamma_0 \vdash \mathtt{LCC}\langle c[\![\beta \setminus^{\alpha} s']\!]\rangle \mid \Delta_0; (\alpha : S \to A)^{\leq 1}; (\alpha' : A)^{\leq 1}$ y $\pi_s \rhd_{\mathcal{E}} \Gamma_1 \vdash s : S \mid \Delta_1; (\alpha' : A)^{\leq 1}$. Se procede por inducción en LCC.

   - $\mathtt{LCC} = \boxdot$. Luego, por (REPL) se tienen los contextos $\Gamma_0 = \Gamma'_0 \cup \Gamma''_0$ y $\Delta = \Delta'_0 \cup \Delta''_0$ con $(\alpha : S \to A)^{\leq 1}$ no vacío en $\pi_{\mathtt{LCC}\langle c[\![\beta \setminus^{\alpha} s']\!]\rangle}$ y una derivación para c por un lado $\pi_c \rhd_{\mathcal{E}} \Gamma'_0 \vdash c \mid \Delta'_0; (\beta : S' \to S \to A)^{\leq 1}; (\alpha : S \to A)^{\leq 1}; (\alpha' : A)^{\leq 1}$ y otra para s' por otro $\pi_{s'} \rhd_{\mathcal{E}} \Gamma''_0 \vdash s' : S' \mid \Delta''_0; (\alpha : S \to A)^{\leq 1}; (\alpha' : A)^{\leq 1}$. Más aún, por hipótesis de la regla P, se tiene $\alpha \notin c$ y $\alpha \notin s'$. Luego, por Lem. 4.3.1, $(\alpha : S \to A)^{\leq 1}$ es vacío en π_c y $\pi_{s'}$. Por Lem. C.0.20 (1), existe $\pi_{s' \cdot s} \rhd_{\mathcal{E}} \Gamma''_0 \cup \Gamma_1 \vdash s' \cdot s : S' \cdot S \mid \Delta''_0 \cup \Delta_1; (\alpha' : A)^{\leq 1}$. Notar que $S' \to S \to A = S' \cdot S \to A$. Luego, se concluye por (REPL), $\pi_{o'} \rhd_{\mathcal{E}} \Gamma \vdash c[\![\beta \setminus^{\alpha'} s' \cdot s]\!] \mid \Delta$.
   - $\mathtt{LCC} = [\gamma]\,\mathtt{LTC}$. Luego, por (NAME), se tiene el contexto $\Delta_0 = \Delta_2; \gamma : B$ con la derivación $\pi_{\mathtt{LTC}\langle c[\![\beta \setminus^{\alpha} s']\!]\rangle} \rhd_{\mathcal{E}} \Gamma_0 \vdash \mathtt{LTC}\langle c[\![\beta \setminus^{\alpha} s']\!]\rangle : B \mid \Delta_2; (\alpha : S \to A)^{\leq 1}; ((\alpha' : A)^{\leq 1} \cup (\gamma : B)^{\leq 1})$. Se procede analizando la forma de LTC.
     - $\mathtt{LTC} = \mathtt{LTC}'\, u$. Luego, de $\pi_{\mathtt{LTC}\langle c[\![\beta \setminus^{\alpha} s']\!]\rangle}$ por (APP), se tienen los contextos $\Gamma_0 = \Gamma'_0 \cup \Gamma''_0$ y $\Delta_2 = \Delta'_2 \cup \Delta''_2$ con derivaciones para las premisas $\pi_{\mathtt{LTC}'\langle c[\![\beta \setminus^{\alpha} s']\!]\rangle} \rhd_{\mathcal{E}} \Gamma'_0 \vdash \mathtt{LTC}'\langle c[\![\beta \setminus^{\alpha} s']\!]\rangle : B' \to B \mid \Delta'_2; (\alpha : S \to A)^{\leq 1}; ((\alpha' : A)^{\leq 1} \cup (\gamma : B)^{\leq 1})$ y $\pi_u \rhd_{\mathcal{E}} \Gamma''_0 \vdash u : B' \mid \Delta''_2; (\alpha : S \to A)^{\leq 1}; ((\alpha' : A)^{\leq 1} \cup (\gamma : B)^{\leq 1})$. Considerar entonces $p = \mathtt{LCC}'\langle c[\![\beta \setminus^{\alpha} s']\!]\rangle[\![\alpha \setminus^{\alpha'} s]\!]$ donde $\mathtt{LCC}' = [\delta]\,\mathtt{LTC}'$ con δ un nombre fresco. Por (NAME) con δ y (REPL) con π_s, $\pi_p \rhd_{\mathcal{E}} \Gamma'_0 \cup \Gamma_1 \vdash p \mid \Delta_1 \cup \Delta'_2; \delta : B' \to B; (\alpha' : A \cup (\gamma : B)^{\leq 1})$. Más aún, como $p \to_{\mathtt{P}} p' = \mathtt{LCC}'\langle c[\![\beta \setminus^{\alpha'} s' \cdot s]\!]\rangle$, por *h.i.* existe una derivación $\pi_{p'} \rhd_{\mathcal{E}} \Gamma'_0 \cup \Gamma_1 \vdash p' \mid \Delta_1 \cup \Delta'_2; \delta : B' \to B; (\alpha' : A \cup (\gamma : B)^{\leq 1})$. Esta derivación necesariamente termina con la regla (NAME) sobre δ. Luego, se tiene $\pi_{\mathtt{LTC}'\langle c[\![\beta \setminus^{\alpha'} s' \cdot s]\!]\rangle} \rhd_{\mathcal{E}}$

$\Gamma'_0 \cup \Gamma_1 \vdash \mathtt{LTC}'\langle c[\![\beta \setminus^{\alpha'} s' \cdot s]\!]\rangle : B' \to B \mid \Delta_1 \cup \Delta'_2; (\alpha' : A \cup (\gamma : B)^{\leq 1})$, dado que δ es fresco. Finalmente, se concluye por (APP) con π_u y luego (NAME) con γ, $\pi_{o'} \rhd_{\mathcal{E}} \Gamma \vdash o' \mid \Delta$. Notar que, por hipótesis de la regla P, se tiene $\alpha \notin u$ y, por Lem. 4.3.1, $(\alpha : S \to A)^{\leq 1}$ es vacío en π_u.

- $\mathtt{LTC} = \lambda x.\mathtt{LTC}'$. Luego, por (ABS), se tiene $B = B' \to B''$ con $\pi_{\mathtt{LTC}'\langle c[\![\beta \setminus^{\alpha} s']\!]\rangle} \rhd_{\mathcal{E}} \Gamma_0; (x : B')^{\leq 1} \vdash \mathtt{LTC}'\langle c[\![\beta \setminus^{\alpha} s']\!]\rangle : B'' \mid \Delta_2; (\alpha : S \to A)^{\leq 1}; ((\alpha' : A)^{\leq 1} \cup (\gamma : B)^{\leq 1})$. Considerar $p = \mathtt{LCC}'\langle c[\![\beta \setminus^{\alpha} s']\!]\rangle[\![\alpha \setminus^{\alpha'} s]\!]$ donde $\mathtt{LCC}' = [\delta]\,\mathtt{LTC}'$ con δ un nombre fresco. Entonces, por (NAME) con δ y (REPL) con π_s, se obtiene la derivación $\pi_p \rhd_{\mathcal{E}} \Gamma; (x : B')^{\leq 1} \vdash p \mid \Delta_1 \cup \Delta_2; \delta : B''; (\alpha' : A \cup (\gamma : B)^{\leq 1})$. Más aún, se tiene $p \to_{\mathtt{P}} p' = \mathtt{LCC}'\langle c[\![\beta \setminus^{\alpha'} s' \cdot s]\!]\rangle$. Lueog, por *h.i.* existe una derivación $\pi_{p'} \rhd_{\mathcal{E}} \Gamma; (x : B')^{\leq 1} \vdash p' \mid \Delta_1 \cup \Delta_2; \delta : B''; (\alpha' : A \cup (\gamma : B)^{\leq 1})$. Esta derivación necesariamente termina con la regla (NAME) sobre δ. Luego, se tiene $\pi_{\mathtt{LTC}'\langle c[\![\beta \setminus^{\alpha'} s' \cdot s]\!]\rangle} \rhd_{\mathcal{E}} \Gamma; (x : B')^{\leq 1} \vdash \mathtt{LTC}'\langle c[\![\beta \setminus^{\alpha'} s' \cdot s]\!]\rangle : B'' \mid \Delta_1 \cup \Delta_2; (\alpha' : A \cup (\gamma : B)^{\leq 1})$, dado que δ es fresco. Finalmente, se concluye por (ABS) y (NAME) con γ, $\pi_{o'} \rhd_{\mathcal{E}} \Gamma \vdash o' \mid \Delta$.
- $\mathtt{LTC} = \mu\delta.\mathtt{LCC}'$. Luego, de $\pi_{\mathtt{LTC}\langle c[\![\beta \setminus^{\alpha} s']\!]\rangle}$ por (CONT), se tiene la premisa $\pi_{\mathtt{LCC}'\langle c[\![\beta \setminus^{\alpha} s']\!]\rangle} \rhd_{\mathcal{E}} \Gamma_0 \vdash \mathtt{LCC}'\langle c[\![\beta \setminus^{\alpha} s']\!]\rangle \mid \Delta'_0; (\alpha : S \to A)^{\leq 1}; ((\alpha' : A)^{\leq 1} \cup (\gamma : B)^{\leq 1}); (\delta : B)^{\leq 1}$. Entonces, aplicando la regla (REPL) con π_s, se obtiene la derivación $\pi_{\mathtt{LCC}'\langle c[\![\beta \setminus^{\alpha} s']\!]\rangle[\![\alpha \setminus^{\alpha'} s]\!]} \rhd_{\mathcal{E}} \Gamma \vdash \mathtt{LCC}'\langle c[\![\beta \setminus^{\alpha} s']\!]\rangle[\![\alpha \setminus^{\alpha'} s]\!] \mid \Delta; (\alpha' : A \cup (\gamma : B)^{\leq 1}); (\delta : B)^{\leq 1}$. Luego, por *h.i.* existe $\pi_{\mathtt{LCC}'\langle c[\![\beta \setminus^{\alpha'} s' \cdot s]\!]\rangle} \rhd_{\mathcal{E}} \Gamma \vdash \mathtt{LCC}'\langle c[\![\beta \setminus^{\alpha'} s' \cdot s]\!]\rangle \mid \Delta; (\alpha' : A \cup (\gamma : B)^{\leq 1}); (\delta : B)^{\leq 1}$. Finalmente, se concluye por (CONT) y (NAME), $\pi_{o'} \rhd_{\mathcal{E}} \Gamma \vdash o' \mid \Delta$.
- $\mathtt{LTC} = \mathtt{LTC}'[x \setminus u]$. Luego, por (SUBS), se tienen contextos $\Gamma_0 = \Gamma'_0 \cup \Gamma''_0$ y $\Delta_2 = \Delta'_2 \cup \Delta''_2$ con una derivación para $\mathtt{LTC}'\langle c[\![\beta \setminus^{\alpha} s']\!]\rangle$ por un lado $\pi_{\mathtt{LTC}'\langle c[\![\beta \setminus^{\alpha} s']\!]\rangle} \rhd_{\mathcal{E}} \Gamma'_0; (x : B')^{\leq 1} \vdash \mathtt{LTC}'\langle c[\![\beta \setminus^{\alpha} s']\!]\rangle : B \mid \Delta'_2; (\alpha : S \to A)^{\leq 1}; ((\alpha' : A)^{\leq 1} \cup (\gamma : B)^{\leq 1})$, y una para u por otro $\pi_u \rhd_{\mathcal{E}} \Gamma''_0 \vdash u : B' \mid \Delta''_2; (\alpha : S \to A)^{\leq 1}; ((\alpha' : A)^{\leq 1} \cup (\gamma : B)^{\leq 1})$. Más aún, por hipótesis de la regla P, se tiene $\alpha \notin u$. Luego, por Lem. 4.3.1, $(\alpha : S \to A)^{\leq 1}$ es vacío en π_u. Considerar $p = \mathtt{LCC}'\langle c[\![\beta \setminus^{\alpha} s']\!]\rangle[\![\alpha \setminus^{\alpha'} s]\!]$ donde $\mathtt{LCC}' = [\gamma]\,\mathtt{LTC}'$. Entonces, por (NAME) con γ y (REPL) con π_s, se tiene la derivación $\pi_p \rhd_{\mathcal{E}} \Gamma'_0 \cup \Gamma_1; (x : B')^{\leq 1} \vdash p \mid \Delta_1 \cup \Delta'_2; (\alpha' : A \cup \gamma : B)$. Incluso más, se tiene $p \to_{\mathtt{P}} p' = \mathtt{LCC}'\langle c[\![\beta \setminus^{\alpha'} s' \cdot s]\!]\rangle$. Luego, por *h.i.* sobre $\mathtt{LCC}'$, se garantiza que existe una derivación para el reducto $\pi_{p'} \rhd_{\mathcal{E}} \Gamma'_0 \cup \Gamma_1; (x : B')^{\leq 1} \vdash p' \mid \Delta_1 \cup \Delta'_2; (\alpha' : A \cup \gamma : B)$. Esta derivación necesariamente termina con (NAME) sobre γ. Luego, $\pi_{\mathtt{LTC}'\langle c[\![\beta \setminus^{\alpha'} s' \cdot s]\!]\rangle} \rhd_{\mathcal{E}} \Gamma'_0 \cup \Gamma_1; (x : B')^{\leq 1} \vdash \mathtt{LTC}'\langle c[\![\beta \setminus^{\alpha'} s' \cdot s]\!]\rangle : B \mid \Delta_1 \cup \Delta'_2; (\alpha' : A \cup (\gamma : B)^{\leq 1})$. Por último, se concluye aplicando la regla (SUBS) con π_u seguida de (NAME) con γ, $\pi_{o'} \rhd_{\mathcal{E}} \Gamma \vdash o' \mid \Delta$.

- $\mathtt{LCC} = \mathtt{LCC}'[\![\gamma \setminus^{\delta} s'']\!]$. Luego, de $\pi_{\mathtt{LCC}\langle c[\![\beta \setminus^{\alpha} s']\!]\rangle}$ por (REPL), se tienen contextos $\Gamma_0 = \Gamma'_0 \cup \Gamma''_0$ y $\Delta_0 = \Delta'_0 \cup \Delta''_0; \delta : B$ con derivaciones para las premisas $\pi_{\mathtt{LCC}'\langle c[\![\beta \setminus^{\alpha} s']\!]\rangle} \rhd_{\mathcal{E}} \Gamma'_0 \vdash \mathtt{LCC}'\langle c[\![\beta \setminus^{\alpha} s']\!]\rangle \mid \Delta'_0; (\alpha : S \to A)^{\leq 1}; (\gamma : S'' \to B)^{\leq 1}; ((\alpha' : A)^{\leq 1} \cup (\delta : B)^{\leq 1})$ y $\pi_{s''} \rhd_{\mathcal{E}} \Gamma''_0 \vdash s'' : S'' \mid \Delta''_0; (\alpha : S \to A)^{\leq 1}; ((\alpha' : A)^{\leq 1} \cup (\delta : B)^{\leq 1})$. Entonces, combinando las derivaciones $\pi_{\mathtt{LCC}'\langle c[\![\beta \setminus^{\alpha} s']\!]\rangle}$ y π_s mediante la regla (REPL), se obtiene $\pi_{\mathtt{LCC}'\langle c[\![\beta \setminus^{\alpha} s']\!]\rangle[\![\alpha \setminus^{\alpha'} s]\!]} \rhd_{\mathcal{E}} \Gamma'_0 \cup \Gamma_1 \vdash \mathtt{LCC}'\langle c[\![\beta \setminus^{\alpha} s']\!]\rangle[\![\alpha \setminus^{\alpha'} s]\!] \mid \Delta'_0 \cup \Delta_1; (\gamma : S'' \to B)^{\leq 1}; ((\alpha' : A)^{\leq 1} \cup (\delta : B)^{\leq 1})$. Luego, por *h.i.* sobre $\mathtt{LCC}'$, existe una derivación para el reducto $\pi_{\mathtt{LCC}'\langle c[\![\beta \setminus^{\alpha'} s' \cdot s]\!]\rangle} \rhd_{\mathcal{E}} \Gamma'_0 \cup \Gamma_1 \vdash \mathtt{LCC}'\langle c[\![\beta \setminus^{\alpha'} s' \cdot s]\!]\rangle \mid \Delta'_0 \cup \Delta_1; (\gamma : S'' \to B)^{\leq 1}; ((\alpha' : A)^{\leq 1} \cup (\delta : B)^{\leq 1})$. Finalmente, se concluye por (REPL) con $\pi_{s''}$, $\pi_{o'} \rhd_{\mathcal{E}} \Gamma \vdash o' \mid \Delta$. Notar que, por hipótesis de la regla P, $\alpha \notin s''$ y, por Lem. 4.3.1, $(\alpha : S \to A)^{\leq 1}$ es vacío en $\pi_{s''}$.
- $\mathtt{LCC} = \mathtt{LCC}'[\gamma \setminus \delta]$. Luego, por (REN) se tiene $\Delta_0 = \Delta'_0; \delta : B$ con la premisa $\pi_{\mathtt{LCC}'\langle c[\![\beta \setminus^{\alpha} s']\!]\rangle} \rhd_{\mathcal{E}} \Gamma_0 \vdash \mathtt{LCC}'\langle c[\![\beta \setminus^{\alpha} s']\!]\rangle \mid \Delta_0; (\alpha : S \to A)^{\leq 1}; ((\alpha' : A)^{\leq 1} \cup (\delta : B)^{\leq 1}); (\gamma : B)^{\leq 1}$. Entonces, aplicando (REPL) con $\pi_{\mathtt{LCC}'\langle c[\![\beta \setminus^{\alpha} s']\!]\rangle}$ y π_s, se obtiene la derivación $\pi_{\mathtt{LCC}'\langle c[\![\beta \setminus^{\alpha} s']\!]\rangle[\![\alpha \setminus^{\alpha'} s]\!]} \rhd_{\mathcal{E}}$

$\Gamma_0 \cup \Gamma_1 \vdash \mathtt{LCC}'\langle c[\![\beta \setminus^{\alpha} s']\!]\rangle[\![\alpha \setminus^{\alpha'} s]\!] \mid \Delta'_0 \cup \Delta_1; (\alpha' : A \cup (\delta : B)^{\leq 1}); (\gamma : B)^{\leq 1}$. Luego, por *h.i.* sobre $\mathtt{LCC}'$, se garantiza que existe una derivación para el reducto $\pi_{\mathtt{LCC}'\langle c[\![\beta \setminus^{\alpha'} s' \cdot s]\!]\rangle} \rhd_{\mathcal{E}} \Gamma_0 \cup \Gamma_1 \vdash \mathtt{LCC}'\langle c[\![\beta \setminus^{\alpha'} s' \cdot s]\!]\rangle \mid \Delta'_0 \cup \Delta_1; (\alpha' : A \cup (\delta : B)^{\leq 1}); (\gamma : B)^{\leq 1}$. Se concluye aplicando (REN) con γ y δ, $\pi_{o'} \rhd_{\mathcal{E}} \Gamma \vdash o' \mid \Delta$.

3. $\mathtt{W}$. Luego, $o = \mathtt{LCC}\langle c[\beta \setminus \alpha]\rangle[\![\alpha \setminus^{\alpha'} s]\!]$ y $o' = \mathtt{LCC}\langle c[\![\beta \setminus^{\alpha} s]\!][\alpha \setminus \alpha']\rangle$. Por (REPL) se tienen los contextos $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta = \Delta_0 \cup \Delta_1; \alpha' : A$ con derivaciones para premisas $\pi_{\mathtt{LCC}\langle c[\beta\setminus\alpha]\rangle} \rhd_{\mathcal{E}} \Gamma_0 \vdash \mathtt{LCC}\langle c[\beta \setminus \alpha]\rangle \mid \Delta_0; (\alpha : S \to A)^{\leq 1}; (\alpha' : A)^{\leq 1}$ y $\pi_s \rhd_{\mathcal{E}} \Gamma_1 \vdash s : S \mid \Delta_1; (\alpha' : A)^{\leq 1}$. Se procede por inducción en $\mathtt{LCC}$.

   - $\mathtt{LCC} = \boxdot$. Luego, de $\pi_{\mathtt{LCC}\langle c[\beta\setminus\alpha]\rangle}$ por (REN), se tiene la derivación para la premisa $\pi_c \rhd_{\mathcal{E}} \Gamma_0 \vdash c \mid \Delta_0; (\beta : S \to A)^{\leq 1}; (\alpha : S \to A)^{\leq 1}; (\alpha' : A)^{\leq 1}$. Más aún, por hipótesis de la regla $\mathtt{W}$, se tiene $\alpha \notin c$. Luego, por Lem. 4.3.1, $(\alpha : S \to A)^{\leq 1}$ es vacío en π_c. Entonces, α es fresca y, por (REPL) con π_s, se obtiene $\pi_{c[\![\beta\setminus^{\alpha} s]\!]} \rhd_{\mathcal{E}} \Gamma \vdash c[\![\beta \setminus^{\alpha} s]\!] \mid \Delta; \alpha : A; (\alpha' : A)^{\leq 1}$. Luego, se concluye por (REPL), $\pi_{o'} \rhd_{\mathcal{E}} \Gamma \vdash c[\![\beta \setminus^{\alpha'} s]\!][\alpha \setminus \alpha'] \mid \Delta$.
   - $\mathtt{LCC} = [\gamma]\,\mathtt{LTC}$. Luego, por (NAME), se tiene el contexto $\Delta_0 = \Delta_2; \gamma : B$ con la derivación $\pi_{\mathtt{LTC}\langle c[\beta\setminus\alpha]\rangle} \rhd_{\mathcal{E}} \Gamma_0 \vdash \mathtt{LTC}\langle c[\beta \setminus \alpha]\rangle : B \mid \Delta_2; (\alpha : S \to A)^{\leq 1}; ((\alpha' : A)^{\leq 1} \cup (\gamma : B)^{\leq 1})$. Se procede analizando la forma de $\mathtt{LTC}$.
     - $\mathtt{LTC} = \mathtt{LTC}'\,u$. Luego, de $\pi_{\mathtt{LTC}\langle c[\beta\setminus\alpha]\rangle}$ por (APP), se tienen los contextos $\Gamma_0 = \Gamma'_0 \cup \Gamma''_0$ y $\Delta_2 = \Delta'_2 \cup \Delta''_2$ con una derivación para $\mathtt{LTC}'\langle c[\beta \setminus \alpha]\rangle$ por un lado $\pi_{\mathtt{LTC}'\langle c[\beta\setminus\alpha]\rangle} \rhd_{\mathcal{E}} \Gamma'_0 \vdash \mathtt{LTC}'\langle c[\beta \setminus \alpha]\rangle : B' \to B \mid \Delta'_2; (\alpha : S \to A)^{\leq 1}; ((\alpha' : A)^{\leq 1} \cup (\gamma : B)^{\leq 1})$, y una para u por otro $\pi_u \rhd_{\mathcal{E}} \Gamma''_0 \vdash u : B' \mid \Delta''_2; (\alpha : S \to A)^{\leq 1}; ((\alpha' : A)^{\leq 1} \cup (\gamma : B)^{\leq 1})$. Considerar $p = \mathtt{LCC}'\langle c[\beta \setminus \alpha]\rangle[\![\alpha \setminus^{\alpha'} s]\!]$ donde $\mathtt{LCC}' = [\delta]\,\mathtt{LTC}'$ con δ un nombre fresco. Entonces, aplicando (NAME) con δ y (REPL) con π_s, se obtiene la derivación $\pi_p \rhd_{\mathcal{E}} \Gamma'_0 \cup \Gamma_1 \vdash p \mid \Delta_1 \cup \Delta'_2; (\alpha' : A \cup (\gamma : B)^{\leq 1}); \delta : B' \to B$. Más aún, se tiene $p \to_{\mathtt{W}} p' = \mathtt{LCC}'\langle c[\![\beta \setminus^{\alpha} s]\!][\alpha \setminus \alpha']\rangle$. Luego, por *h.i.* existe una derivación para el reducto $\pi_{p'} \rhd_{\mathcal{E}} \Gamma'_0 \cup \Gamma_1 \vdash p' \mid \Delta_1 \cup \Delta'_2; (\alpha' : A \cup (\gamma : B)^{\leq 1}); \delta : B' \to B$. Esta derivación necesariamente termina con la regla (NAME) sobre δ. Luego, se tiene $\pi_{\mathtt{LTC}'\langle c[\![\beta\setminus^{\alpha} s]\!][\alpha\setminus\alpha']\rangle} \rhd_{\mathcal{E}} \Gamma'_0 \cup \Gamma_1 \vdash \mathtt{LTC}'\langle c[\![\beta \setminus^{\alpha} s]\!][\alpha \setminus \alpha']\rangle : B' \to B \mid \Delta_1 \cup \Delta'_2; (\alpha' : A \cup (\gamma : B)^{\leq 1})$, dado que δ es fresco. Finalmente, se concluye por (APP) con π_u y luego (NAME) con γ, $\pi_{o'} \rhd_{\mathcal{E}} \Gamma \vdash o' \mid \Delta$. Notar que, por hipótesis de la regla $\mathtt{W}$, $\alpha \notin u$ y, por Lem. 4.3.1, $(\alpha : S \to A)^{\leq 1}$ es vacío en π_u.
     - $\mathtt{LTC} = \lambda x.\mathtt{LTC}'$. Luego, por (ABS), se tiene $B = B' \to B''$ con $\pi_{\mathtt{LTC}'\langle c[\beta\setminus\alpha]\rangle} \rhd_{\mathcal{E}} \Gamma_0; (x : B')^{\leq 1} \vdash \mathtt{LTC}'\langle c[\beta \setminus \alpha]\rangle : B'' \mid \Delta_2; (\alpha : S \to A)^{\leq 1}; ((\alpha' : A)^{\leq 1} \cup (\gamma : B)^{\leq 1})$. Sea $p = \mathtt{LCC}'\langle c[\beta \setminus \alpha]\rangle[\![\alpha \setminus^{\alpha'} s]\!]$ donde $\mathtt{LCC}' = [\delta]\,\mathtt{LTC}'$ con δ un nombre fresco. Por (NAME) y (REPL) con π_s, se tiene $\pi_p \rhd_{\mathcal{E}} \Gamma; (x : B')^{\leq 1} \vdash p \mid \Delta_1 \cup \Delta_2; (\alpha' : A \cup (\gamma : B)^{\leq 1}); \delta : B''$. Más aún, como $p \to_{\mathtt{W}} p' = \mathtt{LCC}'\langle c[\![\beta \setminus^{\alpha} s]\!][\alpha \setminus \alpha']\rangle$, por *h.i.* existe una derivación $\pi_{p'} \rhd_{\mathcal{E}} \Gamma; (x : B')^{\leq 1} \vdash p' \mid \Delta_1 \cup \Delta_2; (\alpha' : A \cup (\gamma : B)^{\leq 1}); \delta : B''$. Esta derivación necesariamente termina con la regla (NAME) sobre δ. Luego, se tiene $\pi_{\mathtt{LTC}'\langle c[\![\beta\setminus^{\alpha} s]\!][\alpha\setminus\alpha']\rangle} \rhd_{\mathcal{E}} \Gamma; (x : B')^{\leq 1} \vdash \mathtt{LTC}'\langle c[\![\beta \setminus^{\alpha} s]\!][\alpha \setminus \alpha']\rangle : B'' \mid \Delta_1 \cup \Delta_2; (\alpha' : A \cup (\gamma : B)^{\leq 1})$, dado que δ es fresco. Finalmente, se concluye por (ABS) y (NAME) con γ, $\pi_{o'} \rhd_{\mathcal{E}} \Gamma \vdash o' \mid \Delta$.
     - $\mathtt{LTC} = \mu\delta.\mathtt{LCC}'$. Luego, de $\pi_{\mathtt{LTC}\langle c[\beta\setminus\alpha]\rangle}$ por (CONT), se tiene la derivación $\pi_{\mathtt{LCC}'\langle c[\beta\setminus\alpha]\rangle} \rhd_{\mathcal{E}} \Gamma_0 \vdash \mathtt{LCC}'\langle c[\beta \setminus \alpha]\rangle \mid \Delta'_0; (\alpha : S \to A)^{\leq 1}; ((\alpha' : A)^{\leq 1} \cup (\gamma : B)^{\leq 1}); (\delta : B)^{\leq 1}$. Más aún, aplicando la regla (REPL) con π_s, se obtiene la derivación $\pi_{\mathtt{LCC}'\langle c[\beta\setminus\alpha]\rangle[\![\alpha\setminus^{\alpha'} s]\!]} \rhd_{\mathcal{E}} \Gamma \vdash \mathtt{LCC}'\langle c[\beta \setminus \alpha]\rangle[\![\alpha \setminus^{\alpha'} s]\!] \mid \Delta; (\alpha' : A \cup (\gamma : B)^{\leq 1}); (\delta : B)^{\leq 1}$. Luego, por *h.i.* sobre $\mathtt{LCC}'$, se garantiza que existe una derivación para el reducto $\pi_{\mathtt{LCC}'\langle c[\![\beta\setminus^{\alpha} s]\!][\alpha\setminus\alpha']\rangle} \rhd_{\mathcal{E}} \Gamma \vdash \mathtt{LCC}'\langle c[\![\beta \setminus^{\alpha} s]\!][\alpha \setminus \alpha']\rangle \mid \Delta; (\alpha' : A \cup (\gamma : B)^{\leq 1}); (\delta : B)^{\leq 1}$. Finalmente, se concluye por (CONT) y (NAME), $\pi_{o'} \rhd_{\mathcal{E}} \Gamma \vdash o' \mid \Delta$.
     - $\mathtt{LTC} = \mathtt{LTC}'[x \setminus u]$. Luego, de $\pi_{\mathtt{LTC}\langle c[\beta\setminus\alpha]\rangle}$ por (SUBS), se tienen contextos $\Gamma_0 = \Gamma'_0 \cup \Gamma''_0$ y $\Delta_2 = \Delta'_2 \cup \Delta''_2$ con una derivación para $\pi_{\mathtt{LTC}'\langle c[\beta\setminus\alpha]\rangle}$ por un lado $\pi_{\mathtt{LTC}'\langle c[\beta\setminus\alpha]\rangle} \rhd_{\mathcal{E}}$

$\Gamma'_0; (x : B')^{\leq 1} \vdash \mathtt{LTC}'\langle c[\beta \setminus \alpha]\rangle : B \mid \Delta'_2; (\alpha : S \to A)^{\leq 1}; ((\alpha' : A)^{\leq 1} \cup (\gamma : B)^{\leq 1})$, y una para u por otro $\pi_u \rhd_{\mathcal{E}} \Gamma''_0 \vdash u : B' \mid \Delta''_2; (\alpha : S \to A)^{\leq 1}; ((\alpha' : A)^{\leq 1} \cup (\gamma : B)^{\leq 1})$. Más aún, por hipótesis de la regla $\mathtt{W}$, se tiene $\alpha \notin u$. Luego, por Lem. 4.3.1, $(\alpha : S \to A)^{\leq 1}$ es vacío en π_u. Considerar $p = \mathtt{LCC}'\langle c[\beta \setminus \alpha]\rangle[\![\alpha \setminus^{\alpha'} s]\!]$ donde $\mathtt{LCC}' = [\gamma]\,\mathtt{LTC}'$. Entonces, aplicando la regla (NAME) con γ y (REPL) con π_s, se obtiene la derivación $\pi_p \rhd_{\mathcal{E}} \Gamma'_0 \cup \Gamma_1; (x : B')^{\leq 1} \vdash p \mid \Delta_1 \cup \Delta'_2; (\alpha' : A \cup \gamma : B)$. Más aún, se tiene $p \to_{\mathtt{W}} p' = \mathtt{LCC}'\langle c[\![\beta \setminus^{\alpha} s]\!][\alpha \setminus \alpha']\rangle$. Luego, por *h.i.* existe la derivación para el reducto $\pi_{p'} \rhd_{\mathcal{E}} \Gamma'_0 \cup \Gamma_1; (x : B')^{\leq 1} \vdash p' \mid \Delta_1 \cup \Delta'_2; (\alpha' : A \cup \gamma : B)$. Esta derivación necesariamente termina con (NAME) sobre γ. Luego, se tiene la premisa $\pi_{\mathtt{LTC}'\langle c[\![\beta \setminus^{\alpha} s]\!][\alpha \setminus \alpha']\rangle} \rhd_{\mathcal{E}} \Gamma'_0 \cup \Gamma_1; (x : B')^{\leq 1} \vdash \mathtt{LTC}'\langle c[\![\beta \setminus^{\alpha} s]\!][\alpha \setminus \alpha']\rangle : B \mid \Delta_1 \cup \Delta'_2; (\alpha' : A \cup (\gamma : B)^{\leq 1})$. Se concluye aplicando (SUBS) con π_u y (NAME) con γ, $\pi_{o'} \rhd_{\mathcal{E}} \Gamma \vdash o' \mid \Delta$.

- $\mathtt{LCC} = \mathtt{LCC}'[\![\gamma \setminus^{\delta} s']\!]$. Luego, por (REPL), se tienen los contextos $\Gamma_0 = \Gamma'_0 \cup \Gamma''_0$ y $\Delta_0 = \Delta'_0 \cup \Delta''_0; \delta : B$ con una derivación para $\mathtt{LCC}'\langle c[\beta \setminus \alpha]\rangle$ por un lado $\pi_{\mathtt{LCC}'\langle c[\beta\setminus\alpha]\rangle} \rhd_{\mathcal{E}} \Gamma'_0 \vdash \mathtt{LCC}'\langle c[\beta \setminus \alpha]\rangle \mid \Delta'_0; (\alpha : S \to A)^{\leq 1}; (\gamma : S' \to B)^{\leq 1}; ((\alpha' : A)^{\leq 1} \cup (\delta : B)^{\leq 1})$, y una para s' por otro $\pi_{s'} \rhd_{\mathcal{E}} \Gamma''_0 \vdash s' : S' \mid \Delta''_0; (\alpha : S \to A)^{\leq 1}; ((\alpha' : A)^{\leq 1} \cup (\delta : B)^{\leq 1})$. Entonces, combinando $\pi_{\mathtt{LCC}'\langle c[\beta\setminus\alpha]\rangle}$ y π_s mediante la regla (REPL), se tiene $\pi_{\mathtt{LCC}'\langle c[\beta\setminus\alpha]\rangle[\![\alpha\setminus^{\alpha'} s]\!]} \rhd_{\mathcal{E}} \Gamma'_0 \cup \Gamma_1 \vdash \mathtt{LCC}'\langle c[\beta \setminus \alpha]\rangle[\![\alpha \setminus^{\alpha'} s]\!] \mid \Delta'_0 \cup \Delta_1; (\gamma : S' \to B)^{\leq 1}; (\alpha' : A \cup (\delta : B)^{\leq 1})$. Luego, la *h.i.* sobre $\mathtt{LCC}'$ garantiza que existe una derivación para el reducto $\pi_{\mathtt{LCC}'\langle c[\![\beta\setminus^{\alpha} s]\!][\alpha\setminus\alpha']\rangle} \rhd_{\mathcal{E}} \Gamma'_0 \cup \Gamma_1 \vdash \mathtt{LCC}'\langle c[\![\beta \setminus^{\alpha} s]\!][\alpha \setminus \alpha']\rangle \mid \Delta'_0 \cup \Delta_1; (\gamma : S' \to B)^{\leq 1}; (\alpha' : A \cup (\delta : B)^{\leq 1})$. Por último, se concluye por (REPL) con $\pi_{s'}$, $\pi_{o'} \rhd_{\mathcal{E}} \Gamma \vdash o' \mid \Delta$. Notar que, por hipótesis de la regla $\mathtt{W}$, $\alpha \notin s'$ y, por Lem. 4.3.1, $(\alpha : S \to A)^{\leq 1}$ es vacío en $\pi_{s'}$.
- $\mathtt{LCC} = \mathtt{LCC}'[\gamma\backslash\delta]$. Luego, por (REN), se tiene el contexto $\Delta_0 = \Delta'_0; \delta : B$ con $\pi_{\mathtt{LCC}'\langle c[\beta\setminus\alpha]\rangle} \rhd_{\mathcal{E}} \Gamma_0 \vdash \mathtt{LCC}'\langle c[\beta \setminus \alpha]\rangle \mid \Delta_0; (\alpha : S \to A)^{\leq 1}; ((\alpha' : A)^{\leq 1} \cup (\delta : B)^{\leq 1}); (\gamma : B)^{\leq 1}$. Mediante la regla (REPL), combinando $\pi_{\mathtt{LCC}'\langle c[\beta\setminus\alpha]\rangle}$ y π_s, se obtiene la derivación $\pi_{\mathtt{LCC}'\langle c[\beta\setminus\alpha]\rangle[\![\alpha\setminus^{\alpha'} s]\!]} \rhd_{\mathcal{E}} \Gamma_0 \cup \Gamma_1 \vdash \mathtt{LCC}'\langle c[\beta \setminus \alpha]\rangle[\![\alpha \setminus^{\alpha'} s]\!] \mid \Delta_0 \cup \Delta_1; (\alpha' : A \cup (\delta : B)^{\leq 1}); (\gamma : B)^{\leq 1}$. Luego, la *h.i.* sobre $\mathtt{LCC}'$ garantiza que existe una derivación para el reducto $\pi_{\mathtt{LCC}'\langle c[\![\beta\setminus^{\alpha} s]\!][\alpha\setminus\alpha']\rangle} \rhd_{\mathcal{E}} \Gamma_0 \cup \Gamma_1 \vdash \mathtt{LCC}'\langle c[\![\beta \setminus^{\alpha} s]\!][\alpha \setminus \alpha']\rangle \mid \Delta_0 \cup \Delta_1; (\alpha' : A \cup (\delta : B)^{\leq 1}); (\gamma : B)^{\leq 1}$. Por último, se concluye aplicando (REN) con γ, $\pi_{o'} \rhd_{\mathcal{E}} \Gamma \vdash o' \mid \Delta$.

- $\mathtt{O} = \mathtt{T}\,u$. Luego, $o = t\,u$ y $o' = t'\,u$ con $t \to_{\mathfrak{C}} t'$. Más aún, por (APP) se tienen $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta = \Delta_0 \cup \Delta_1$ con $\pi_t \rhd_{\mathcal{E}} \Gamma_0 \vdash t : A \to T \mid \Delta_0$ y $\pi_u \rhd_{\mathcal{E}} \Gamma_1 \vdash u : A \mid \Delta_1$. Por *h.i.* existe $\pi_{t'} \rhd_{\mathcal{E}} \Gamma_0 \vdash t' : A \to T \mid \Delta_0$. Finalmente, se concluye por (APP), $\pi' \rhd_{\mathcal{E}} \Gamma \vdash o' : T \mid \Delta$.
- $\mathtt{O} = t\,\mathtt{T}$. Luego, $o = t\,u$ y $o' = t\,u'$ con $u \to_{\mathfrak{C}} u'$. Más aún, por (APP) se tienen $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta = \Delta_0 \cup \Delta_1$ con $\pi_t \rhd_{\mathcal{E}} \Gamma_0 \vdash t : A \to T \mid \Delta_0$ y $\pi_u \rhd_{\mathcal{E}} \Gamma_1 \vdash u : A \mid \Delta_1$. Por *h.i.* existe $\pi_{u'} \rhd_{\mathcal{E}} \Gamma_1 \vdash u' : A \mid \Delta_1$. Finalmente, se concluye por (APP), $\pi' \rhd_{\mathcal{E}} \Gamma \vdash o' : T \mid \Delta$.
- $\mathtt{O} = \lambda x.\mathtt{T}$. Luego, $o = \lambda x.t$ y $o' = \lambda x.t'$ con $t \to_{\mathfrak{C}} t'$. Más aún, por (ABS) se tiene $T = A \to B$ con $\pi_t \rhd_{\mathcal{E}} \Gamma; (x : A)^{\leq 1} \vdash t : B \mid \Delta$. Por *h.i.* existe $\pi_{t'} \rhd_{\mathcal{E}} \Gamma; (x : A)^{\leq 1} \vdash t' : B \mid \Delta$. Finalmente, se concluye por (ABS), $\pi' \rhd_{\mathcal{E}} \Gamma \vdash o' : T \mid \Delta$.
- $\mathtt{O} = \mu\alpha.\mathtt{C}$. Luego, $o = \mu\alpha.c$ y $o' = \mu\alpha.c'$ con $c \to_{\mathfrak{C}} c'$. Más aún, por (CONT) se tiene $\pi_c \rhd_{\mathcal{E}} \Gamma \vdash c \mid \Delta; (\alpha : T)^{\leq 1}$. Por *h.i.* existe una derivación $\pi_{c'} \rhd_{\mathcal{E}} \Gamma \vdash c' \mid \Delta; (\alpha : T)^{\leq 1}$. Finalmente, se concluye por (CONT), $\pi' \rhd_{\mathcal{E}} \Gamma \vdash o' : T \mid \Delta$.
- $\mathtt{O} = \mathtt{T}[x \setminus u]$. Luego, $o = t[x \setminus u]$ y $o' = t'[x \setminus u]$ con $t \to_{\mathfrak{C}} t'$. Más aún, por (SUBS), $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta = \Delta_0 \cup \Delta_1$ con $\pi_t \rhd_{\mathcal{E}} \Gamma_0; (x : A)^{\leq 1} \vdash t : T \mid \Delta_0$ y $\pi_u \rhd_{\mathcal{E}} \Gamma_1 \vdash u : A \mid \Delta_1$. Por *h.i.* existe $\pi_{t'} \rhd_{\mathcal{E}} \Gamma_0; (x : A)^{\leq 1} \vdash t' : T \mid \Delta_0$. Finalmente, se concluye por (SUBS), $\pi' \rhd_{\mathcal{E}} \Gamma \vdash o' : T \mid \Delta$.
- $\mathtt{O} = t[x \setminus \mathtt{T}]$. Luego, $o = t[x \setminus u]$ y $o' = t[x \setminus u']$ con $u \to_{\mathfrak{C}} u'$. Más aún, por (SUBS) se tienen $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta = \Delta_0 \cup \Delta_1$ con $\pi_t \rhd_{\mathcal{E}} \Gamma_0; (x : A)^{\leq 1} \vdash t : T \mid \Delta_0$ y $\pi_u \rhd_{\mathcal{E}} \Gamma_1 \vdash u : A \mid \Delta_1$. Por *h.i.* existe $\pi_{u'} \rhd_{\mathcal{E}} \Gamma_1 \vdash u' : A \mid \Delta_1$. Finalmente, se concluye por (SUBS), $\pi' \rhd_{\mathcal{E}} \Gamma \vdash o' : T \mid \Delta$.

- $\mathtt{O} = [\alpha]\,\mathtt{T}$. Luego, $o = [\alpha]\,t$ y $o' = [\alpha]\,t'$ con $t \to_{\mathfrak{C}} t'$. Más aún, por (NAME) se tiene $\Delta = \Delta_0; \alpha : A$ con $\pi_t \rhd_{\mathcal{E}} \Gamma \vdash t : A \mid \Delta_0; (\alpha : A)^{\leq 1}$. Por *h.i.* existe $\pi_{t'} \rhd_{\mathcal{E}} \Gamma \vdash t' : A \mid \Delta_0; (\alpha : A)^{\leq 1}$. Se concluye por (NAME), $\pi' \rhd_{\mathcal{E}} \Gamma \vdash o' : T \mid \Delta_0; \alpha : A$.

- $\mathtt{O} = \mathtt{C}[\![\alpha \setminus^{\alpha'} s]\!]$. Luego, $o = c[\![\alpha \setminus^{\alpha'} s]\!]$ y $o' = c'[\![\alpha \setminus^{\alpha'} s]\!]$ con $c \to_{\mathfrak{C}} c'$. Más aún, por (REPL) se tienen $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta = \Delta_0 \cup \Delta_1; \alpha' : A$ con $\pi_c \rhd_{\mathcal{E}} \Gamma_0 \vdash c \mid \Delta_0; (\alpha : S \to A)^{\leq 1}; (\alpha' : A)^{\leq 1}$ y $\pi_s \rhd_{\mathcal{E}} \Gamma_1 \vdash s : A \mid \Delta_1; (\alpha' : A)^{\leq 1}$. Por *h.i.* existe $\pi_{c'} \rhd_{\mathcal{E}} \Gamma_0 \vdash c' \mid \Delta_0; (\alpha : S \to A)^{\leq 1}; (\alpha' : A)^{\leq 1}$. Finalmente, se concluye por (REPL), $\pi' \rhd_{\mathcal{E}} \Gamma \vdash o' : T \mid \Delta$.

- $\mathtt{O} = c[\![\alpha \setminus^{\alpha'} \mathtt{S}]\!]$. Luego, $o = c[\![\alpha \setminus^{\alpha'} s]\!]$ y $o' = c[\![\alpha \setminus^{\alpha'} s']\!]$ con $s \to_{\mathfrak{C}} s'$. Más aún, por (REPL) se tienen $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta = \Delta_0 \cup \Delta_1; \alpha' : A$ con $\pi_c \rhd_{\mathcal{E}} \Gamma_0 \vdash c \mid \Delta_0; (\alpha : S \to A)^{\leq 1}; (\alpha' : A)^{\leq 1}$ y $\pi_s \rhd_{\mathcal{E}} \Gamma_1 \vdash s : A \mid \Delta_1; (\alpha' : A)^{\leq 1}$. Por *h.i.* existe una derivación $\pi_{s'} \rhd_{\mathcal{E}} \Gamma_1 \vdash s' : A \mid \Delta_1; (\alpha' : A)^{\leq 1}$. Finalmente, se concluye por (REPL), $\pi' \rhd_{\mathcal{E}} \Gamma \vdash o' : T \mid \Delta$.

- $\mathtt{O} = \mathtt{C}[\alpha \setminus \beta]$. Luego, $o = c[\alpha \setminus \beta]$ y $o' = c'[\alpha \setminus \beta]$ con $c \to_{\mathfrak{C}} c'$. Más aún, por (REN) se tiene $\Delta = \Delta_0; \beta : A$ con $\pi_c \rhd_{\mathcal{E}} \Gamma \vdash c \mid \Delta_0; (\alpha : A)^{\leq 1}; (\beta : A)^{\leq 1}$. Por *h.i.* existe una derivación $\pi_{c'} \rhd_{\mathcal{E}} \Gamma \vdash c' \mid \Delta_0; (\alpha : A)^{\leq 1}; (\beta : A)^{\leq 1}$. Finalmente, se concluye por (REN), $\pi' \rhd_{\mathcal{E}} \Gamma \vdash o' : T \mid \Delta$.

- $\mathtt{O} = \mathtt{T} \cdot s$. Luego, $o = t \cdot s$ y $o' = t' \cdot s$ con $t \to_{\mathfrak{C}} t'$. Más aún, por (STK) se tienen $T = A \cdot S$, $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta = \Delta_0 \cup \Delta_1$ con $\pi_t \rhd_{\mathcal{E}} \Gamma_0 \vdash t : A \mid \Delta_0$ y $\pi_s \rhd_{\mathcal{E}} \Gamma_1 \vdash s : S \mid \Delta_1$. Por *h.i.* existe $\pi_{t'} \rhd_{\mathcal{E}} \Gamma_0 \vdash t' : A \mid \Delta_0$. Finalmente, se concluye por (STK), $\pi' \rhd_{\mathcal{E}} \Gamma \vdash o' : T \mid \Delta$.

- $\mathtt{O} = t \cdot \mathtt{S}$. Luego, $o = t \cdot s$ y $o' = t \cdot s'$ con $s \to_{\mathfrak{C}} s'$. Más aún, por (STK) se tienen $T = A \cdot S$, $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta = \Delta_0 \cup \Delta_1$ con $\pi_t \rhd_{\mathcal{E}} \Gamma_0 \vdash t : A \mid \Delta_0$ y $\pi_s \rhd_{\mathcal{E}} \Gamma_1 \vdash s : S \mid \Delta_1$. Por *h.i.* existe $\pi_{s'} \rhd_{\mathcal{E}} \Gamma_1 \vdash s' : S \mid \Delta_1$. Finalmente, se concluye por (STK), $\pi' \rhd_{\mathcal{E}} \Gamma \vdash o' : T \mid \Delta$.

□

Lema C.0.44 (Subject Expansion)**.** *Sea $o \in \mathbb{O}_{\Lambda M}$. Si $\pi \rhd_{\mathcal{E}} \Gamma \vdash o : T \mid \Delta$ y $o' \to_{\mathfrak{C}} o$, entonces existe $\pi' \rhd_{\mathcal{E}} \Gamma \vdash o' : T \mid \Delta$.*

Demostración. Por definición, $o' \to_{\mathfrak{C}} o$ implica $o' = \mathtt{O}\langle l \rangle$ y $o = \mathtt{O}\langle r \rangle$ con $l \mapsto_* r$, $* \in \{\mathtt{dB}, \mathtt{dM}, \mathtt{N}, \mathtt{P}, \mathtt{W}\}$. La demostración es por inducción en $\mathtt{O}$.

- $\mathtt{O} = \Box$. Luego, se tienen dos casos posibles según la regla de reescritura aplicada al reducir.

  1. $\mathtt{dB}$. Luego, $o = \mathtt{L}\langle t[x \setminus u] \rangle$ y $o' = \mathtt{L}\langle \lambda x.t \rangle\, u$. Se procede por inducción en $\mathtt{L}$.

     - $\mathtt{L} = \Box$. Luego, por (SUBS) se tienen $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta = \Delta_0 \cup \Delta_1$ con las derivaciones $\pi_t \rhd_{\mathcal{E}} \Gamma_0; (x : A)^{\leq 1} \vdash t : T \mid \Delta_0$ y $\pi_u \rhd_{\mathcal{E}} \Gamma_1 \vdash u : A \mid \Delta_1$. Por (ABS) con π_t se obtiene $\pi_{\mathtt{L}\langle \lambda x.t \rangle} \rhd_{\mathcal{E}} \Gamma_0 \vdash \mathtt{L}\langle \lambda x.t \rangle : A \to T \mid \Delta_0$. Se concluye por (APP) con π_u, $\pi' \rhd_{\mathcal{E}} \Gamma \vdash \mathtt{L}\langle \lambda x.t \rangle\, u : T \mid \Delta$.
     - $\mathtt{L} = \mathtt{L}'[y \setminus v]$ con $y \notin u$. Luego, por (SUBS) se tienen $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta = \Delta_0 \cup \Delta_1$ con $\pi_{\mathtt{L}'\langle t[x \setminus u] \rangle} \rhd_{\mathcal{E}} \Gamma_0; (y : B)^{\leq 1} \vdash \mathtt{L}'\langle t[x \setminus u] \rangle : T \mid \Delta_0$ y $\pi_v \rhd_{\mathcal{E}} \Gamma_1 \vdash v : B \mid \Delta_1$. Por *h.i.* existe una derivación $\pi_{\mathtt{L}'\langle \lambda x.t \rangle\, u} \rhd_{\mathcal{E}} \Gamma_0; (y : B)^{\leq 1} \vdash \mathtt{L}'\langle \lambda x.t \rangle\, u : T \mid \Delta_0$. Más aún, de (APP) se tienen $\Gamma_0 = \Gamma_0' \cup \Gamma_0''$ y $\Delta_0 = \Delta_0' \cup \Delta_0''$ con derivaciones $\pi_{\mathtt{L}'\langle \lambda x.t \rangle} \rhd_{\mathcal{E}} \Gamma_0'; (y : B)^{\leq 1} \vdash \mathtt{L}'\langle \lambda x.t \rangle : A \to T \mid \Delta_0'$ y $\pi_u \rhd_{\mathcal{E}} \Gamma_0'' \vdash u : A \mid \Delta_0''$. Notar que, por Lem. 4.3.1, $y \notin \mathsf{dom}(\Gamma_0'')$. Luego, por (SUBS) con $\pi_{\mathtt{L}'\langle \lambda x.t \rangle}$ y π_v se obtiene la derivación $\pi_{\mathtt{L}\langle \lambda x.t \rangle} \rhd_{\mathcal{E}} \Gamma_0' \cup \Gamma_1 \vdash \mathtt{L}\langle \lambda x.t \rangle : A \to T \mid \Delta_0' \cup \Delta_1$. Finalmente, se concluye por (APP) con π_u, $\pi' \rhd_{\mathcal{E}} \Gamma \vdash \mathtt{L}\langle \lambda x.t \rangle\, u : T \mid \Delta$.

  2. $\mathtt{dM}$. Luego, $o = \mathtt{L}\langle \mu\alpha'.c[\![\alpha \setminus^{\alpha'} u]\!] \rangle$ y $o' = \mathtt{L}\langle \mu\alpha.c \rangle\, u$ con α' un nombre fresco. Se procede por inducción en $\mathtt{L}$.

- $\mathtt{L} = \square$. Luego, por (CONT) se tiene la derivación $\pi_{c[\![\alpha \setminus^{\alpha'} u]\!]} \rhd_{\mathcal{E}} \Gamma \vdash c[\![\alpha \setminus^{\alpha'} u]\!] \mid \Delta; \alpha' : T$. Más aún, por (REPL), se tienen los contextos $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta = \Delta_0 \cup \Delta_1$ con las derivaciones $\pi_c \rhd_{\mathcal{E}} \Gamma_0 \vdash c \mid \Delta_0; (\alpha : A \to T)^{\leq 1}; (\alpha' : T)^{\leq 1}$ y $\pi_u \rhd_{\mathcal{E}} \Gamma_1 \vdash u : A \mid \Delta_1$. Notar que, por Lem. 4.3.1, α' fresca implica $(\alpha' : T)^{\leq 1}$ vacío en π_c. Luego, por (CONT) se obtiene $\pi_{\mu\alpha.c} \rhd_{\mathcal{E}} \Gamma_0 \vdash \mu\alpha.c : A \to T \mid \Delta_0$. Finalmente, se concluye por (APP) con π_u, $\pi' \rhd_{\mathcal{E}} \Gamma \vdash \mathtt{L}\langle \mu\alpha.c \rangle\, u : T \mid \Delta$.
- $\mathtt{L} = \mathtt{L}'[y \setminus v]$ con $y \notin u$. Luego, por (SUBS), $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta = \Delta_0 \cup \Delta_1$ con $\pi_{\mathtt{L}'\langle \mu\alpha'.c[\![\alpha \setminus^{\alpha'} u]\!] \rangle} \rhd_{\mathcal{E}} \Gamma_0; (y : B)^{\leq 1} \vdash \mathtt{L}'\langle \mu\alpha'.c[\![\alpha \setminus^{\alpha'} u]\!] \rangle : T \mid \Delta_0$ y $\pi_v \rhd_{\mathcal{E}} \Gamma_1 \vdash v : B \mid \Delta_1$. Por *h.i.* existe una derivación $\pi_{\mathtt{L}'\langle \mu\alpha.c \rangle\, u} \rhd_{\mathcal{E}} \Gamma_0; (y : B)^{\leq 1} \vdash \mathtt{L}'\langle \mu\alpha.c \rangle\, u : T \mid \Delta_0$. Más aún, de (APP) se tienen $\Gamma_0 = \Gamma_0' \cup \Gamma_0''$ y $\Delta_0 = \Delta_0' \cup \Delta_0''$ con derivaciones $\pi_{\mathtt{L}'\langle \mu\alpha.c \rangle} \rhd_{\mathcal{E}} \Gamma_0'; (y : B)^{\leq 1} \vdash \mathtt{L}'\langle \mu\alpha.c \rangle : A \to T \mid \Delta_0'$ y $\pi_u \rhd_{\mathcal{E}} \Gamma_0'' \vdash u : A \mid \Delta_0''$. Notar que, por Lem. 4.3.1, $y \notin \mathsf{dom}(\Gamma_0'')$. Luego, por (SUBS) con $\pi_{\mathtt{L}'\langle \mu\alpha.c \rangle}$ y π_v se obtiene la derivación $\pi_{\mathtt{L}\langle \mu\alpha.c \rangle} \rhd_{\mathcal{E}} \Gamma_0' \cup \Gamma_1 \vdash \mathtt{L}\langle \mu\alpha.c \rangle : A \to T \mid \Delta_0' \cup \Delta_1$. Finalmente, se concluye por (APP) con π_u, $\pi' \rhd_{\mathcal{E}} \Gamma \vdash \mathtt{L}\langle \mu\alpha.c \rangle\, u : T \mid \Delta$.

- $\mathtt{O} = \boxdot$. Luego, se tienen tres casos posibles según la regla de reescritura aplicada al reducir.

  1. N. Luego, $o = \mathtt{LCC}\langle [\alpha']\, t :: s \rangle$ y $o' = \mathtt{LCC}\langle [\alpha]\, t \rangle [\![\alpha \setminus^{\alpha'} s]\!]$. Se procede por inducción en LCC.
     - $\mathtt{LCC} = \boxdot$. Luego, por (NAME) se tiene el contexto $\Delta = \Delta'; \alpha' : A$ con la derivación $\pi_{t::s} \rhd_{\mathcal{E}} \Gamma \vdash t :: s : A \mid \Delta'; (\alpha' : A)^{\leq 1}$. Por Lem. C.0.20 (4), se tienen $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta' = \Delta_0 \cup \Delta_1$ con $\pi_t \rhd_{\mathcal{E}} \Gamma_0 \vdash t : S \to A \mid \Delta_0; (\alpha' : A)^{\leq 1}$ y $\pi_s \rhd_{\mathcal{E}} \Gamma_1 \vdash s : S \mid \Delta_1; (\alpha' : A)^{\leq 1}$. Más aún, por hipótesis de la regla N, $\alpha \notin t$. Luego, por (NAME) con α y π_t, se tiene $\pi_{[\alpha]\, t} \rhd_{\mathcal{E}} \Gamma_0 \vdash [\alpha]\, t \mid \Delta_0; \alpha : S \to A; (\alpha' : A)^{\leq 1}$. Finalmente, se concluye por (REPL) con π_s, $\pi_{o'} \rhd_{\mathcal{E}} \Gamma \vdash o' \mid \Delta$.
     - $\mathtt{LCC} = [\gamma]\, \mathtt{LTC}$. Luego, por Lem. 4.3.1 con $\alpha' \in o$ y (NAME), se tiene el contexto $\Delta = \Delta'; (\alpha' : A \cup \gamma : B)$ con $\pi_{\mathtt{LTC}\langle [\alpha']\, t ::s \rangle} \rhd_{\mathcal{E}} \Gamma \vdash \mathtt{LTC}\langle [\alpha']\, t :: s \rangle : B \mid \Delta'; (\alpha' : A \cup (\gamma : B)^{\leq 1})$. Se procede analizando la forma de LTC.
       - $\mathtt{LTC} = \mathtt{LTC}'\, u$. Luego, por (APP), se tienen contextos $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta' = \Delta_0 \cup \Delta_1$ con derivaciones $\pi_{\mathtt{LTC}'\langle [\alpha']\, t ::s \rangle} \rhd_{\mathcal{E}} \Gamma_0 \vdash \mathtt{LTC}'\langle [\alpha']\, t :: s \rangle : B' \to B \mid \Delta_0; (\alpha' : A \cup (\gamma : B)^{\leq 1})$ y $\pi_u \rhd_{\mathcal{E}} \Gamma_1 \vdash u : B' \mid \Delta_1; (\alpha' : A \cup (\gamma : B)^{\leq 1})$. Considerar $p = \mathtt{LCC}'\langle [\alpha']\, t :: s \rangle$ donde $\mathtt{LCC}' = [\delta]\, \mathtt{LTC}'$ con δ un nombre fresco. Luego, por (NAME), se obtiene $\pi_p \rhd_{\mathcal{E}} \Gamma_0 \vdash p \mid \Delta_0; (\alpha' : A \cup (\gamma : B)^{\leq 1}); \delta : B' \to B$. Más aún, como $p' = \mathtt{LCC}'\langle [\alpha]\, t \rangle [\![\alpha \setminus^{\alpha'} s]\!] \to_{\mathtt{N}} p$, por *h.i.* existe $\pi_{p'} \rhd_{\mathcal{E}} \Gamma_0 \vdash p' \mid \Delta_0; (\alpha' : A \cup (\gamma : B)^{\leq 1}); \delta : B' \to B$. Esta derivación necesariamente termina con aplicaciones de las reglas (REPL) y (NAME) con δ. Luego, se tienen los contextos $\Gamma_0 = \Gamma_0' \cup \Gamma_0''$ y $\Delta_0 = \Delta_0' \cup \Delta_0''$ con las premisas $\pi_{\mathtt{LTC}'\langle [\alpha]\, t \rangle} \rhd_{\mathcal{E}} \Gamma_0' \vdash \mathtt{LTC}'\langle [\alpha]\, t \rangle : B' \to B \mid \Delta_0'; \alpha : S \to A; ((\alpha' : A)^{\leq 1} \cup (\gamma : B)^{\leq 1})$ y $\pi_s \rhd_{\mathcal{E}} \Gamma_0'' \vdash s : S \mid \Delta_0''; ((\alpha' : A)^{\leq 1} \cup (\gamma : B)^{\leq 1})$, dado que δ es fresco. Luego, aplicando (APP) con π_u seguido de (NAME) con γ, se obtiene la derivación $\pi_{\mathtt{LCC}\langle [\alpha]\, t \rangle} \rhd_{\mathcal{E}} \Gamma_0' \cup \Gamma_1 \vdash \mathtt{LCC}\langle [\alpha]\, t \rangle \mid \Delta_0' \cup \Delta_1; \alpha : S \to A; ((\alpha' : A)^{\leq 1} \cup \gamma : B)$. Finalmente, se concluye por (REPL) con π_s, $\pi_{o'} \rhd_{\mathcal{E}} \Gamma \vdash o' \mid \Delta$.
       - $\mathtt{LTC} = \lambda x.\mathtt{LTC}'$. Luego, por (ABS), se tiene $B = B' \to B''$ con la derivación $\pi_{\mathtt{LTC}'\langle [\alpha']\, t ::s \rangle} \rhd_{\mathcal{E}} \Gamma; (x : B')^{\leq 1} \vdash \mathtt{LTC}'\langle [\alpha']\, t :: s \rangle : B'' \mid \Delta'; (\alpha' : A \cup (\gamma : B)^{\leq 1})$. Considerar $p = \mathtt{LCC}'\langle [\alpha']\, t :: s \rangle$ donde $\mathtt{LCC}' = [\delta]\, \mathtt{LTC}'$ con δ un nombre fresco. Entonces, por (NAME), se obtiene $\pi_p \rhd_{\mathcal{E}} \Gamma; (x : B')^{\leq 1} \vdash p \mid \Delta'; (\alpha' : A \cup (\gamma : B)^{\leq 1}); \delta : B''$. Más aún, como $p' = \mathtt{LCC}'\langle [\alpha]\, t \rangle [\![\alpha \setminus^{\alpha'} s]\!] \to_{\mathtt{N}} p$, por *h.i.* existe una derivación $\pi_{p'} \rhd_{\mathcal{E}} \Gamma; (x : B')^{\leq 1} \vdash p' \mid \Delta'; (\alpha' : A \cup (\gamma : B)^{\leq 1}); \delta : B''$. De (REPL) y (NAME), dado que δ es fresco, se tienen los contextos $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta' = \Delta_0 \cup \Delta_1$ con $\pi_{\mathtt{LTC}'\langle [\alpha]\, t \rangle} \rhd_{\mathcal{E}} \Gamma_0; (x : B')^{\leq 1} \vdash \mathtt{LTC}'\langle [\alpha]\, t \rangle : B'' \mid \Delta_0; \alpha : S \to A; ((\alpha' : A)^{\leq 1} \cup (\gamma : B)^{\leq 1})$ y también $\pi_s \rhd_{\mathcal{E}} \Gamma_1 \vdash s : S \mid \Delta_1; ((\alpha' : A)^{\leq 1} \cup (\gamma : B)^{\leq 1})$. Luego, por (ABS) seguido de (NAME)

con γ, se obtiene $\pi_{\texttt{LCC}\langle[\alpha]\,t\rangle} \rhd_{\mathcal{E}} \Gamma_0 \vdash \texttt{LCC}\langle[\alpha]\,t\rangle \mid \Delta_0; \alpha : S \to A; ((\alpha' : A)^{\leq 1} \cup \gamma : B)$. Por último, se concluye por (REPL) con π_s, $\pi_{o'} \rhd_{\mathcal{E}} \Gamma \vdash o' \mid \Delta$.

- $\texttt{LTC} = \mu\delta.\texttt{LCC}'$. Luego, de $\pi_{\texttt{LTC}\langle[\alpha']\,t::s\rangle}$ por (CONT), se tiene la premisa $\pi_{\texttt{LCC}'\langle[\alpha']\,t::s\rangle} \rhd_{\mathcal{E}} \Gamma \vdash \texttt{LCC}'\langle[\alpha']\,t :: s\rangle \mid \Delta'; (\alpha' : A \cup (\gamma : B)^{\leq 1}); (\delta : B)^{\leq 1}$. Más aún, por *h.i.* sobre $\texttt{LCC}'$, existe $\pi_{\texttt{LCC}'\langle[\alpha]\,t\rangle[\![\alpha \setminus^{\alpha'} s]\!]} \rhd_{\mathcal{E}} \Gamma \vdash \texttt{LCC}'\langle[\alpha]\,t\rangle[\![\alpha \setminus^{\alpha'} s]\!] \mid \Delta'; (\alpha' : A \cup (\gamma : B)^{\leq 1}); (\delta : B)^{\leq 1}$. De (REPL), se tiene $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta' = \Delta_0 \cup \Delta_1$ con la premisa $\pi_{\texttt{LCC}'\langle[\alpha]\,t\rangle} \rhd_{\mathcal{E}} \Gamma_0 \vdash \texttt{LCC}'\langle[\alpha]\,t\rangle \mid \Delta_0; \alpha : S \to A; ((\alpha' : A)^{\leq 1} \cup (\gamma : B)^{\leq 1}); (\delta : B)^{\leq 1}$ y también $\pi_s \rhd_{\mathcal{E}} \Gamma_1 \vdash s : S \mid \Delta_1; ((\alpha' : A)^{\leq 1} \cup (\gamma : B)^{\leq 1}); (\delta : B)^{\leq 1}$. Por (CONT) con δ y (NAME) con γ, se obtiene $\pi_{\texttt{LCC}\langle[\alpha]\,t\rangle} \rhd_{\mathcal{E}} \Gamma_0 \vdash \texttt{LCC}\langle[\alpha]\,t\rangle \mid \Delta_0; \alpha : S \to A; ((\alpha' : A)^{\leq 1} \cup \gamma : B)$. Por último, se concluye por (REPL) con π_s, $\pi_{o'} \rhd_{\mathcal{E}} \Gamma \vdash o' \mid \Delta$.
- $\texttt{LTC} = \texttt{LTC}'[x \backslash u]$. Luego, por (SUBS), se tienen $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta' = \Delta_0 \cup \Delta_1$ con derivaciones $\pi_{\texttt{LTC}'\langle[\alpha']\,t::s\rangle} \rhd_{\mathcal{E}} \Gamma_0; (x : B')^{\leq 1} \vdash \texttt{LTC}'\langle[\alpha']\,t :: s\rangle : B \mid \Delta_0; (\alpha' : A \cup (\gamma : B)^{\leq 1})$ y $\pi_u \rhd_{\mathcal{E}} \Gamma_1 \vdash u : B' \mid \Delta_1; ((\alpha' : A)^{\leq 1} \cup (\gamma : B)^{\leq 1})$. Considerar $p = \texttt{LCC}'\langle[\alpha']\,t :: s\rangle$ donde $\texttt{LCC}' = [\gamma]\,\texttt{LTC}'$. Por (NAME), $\pi_p \rhd_{\mathcal{E}} \Gamma_0; (x : B')^{\leq 1} \vdash p \mid \Delta_0; (\alpha' : A \cup (\gamma : B)^{\leq 1})$. Más aún, como $p' = \texttt{LCC}'\langle[\alpha]\,t\rangle[\![\alpha \setminus^{\alpha'} s]\!] \to_{\mathtt{N}} p$, por *h.i.* existe una derivación $\pi_{p'} \rhd_{\mathcal{E}} \Gamma_0; (x : B')^{\leq 1} \vdash p' \mid \Delta_0; (\alpha' : A \cup (\gamma : B)^{\leq 1})$. Esta derivación necesariamente termina con aplicaciones de las reglas (REPL) y (NAME) con γ. Luego, se tienen los contextos $\Gamma_0 = \Gamma_0' \cup \Gamma_0''$ y $\Delta_0 = \Delta_0' \cup \Delta_0''$ con las premisas $\pi_{\texttt{LTC}'\langle[\alpha]\,t\rangle} \rhd_{\mathcal{E}} \Gamma_0'; (x : B')^{\leq 1} \vdash \texttt{LTC}'\langle[\alpha]\,t\rangle : B \mid \Delta_0'; \alpha : S \to A; ((\alpha' : A)^{\leq 1} \cup (\gamma : B)^{\leq 1})$ y $\pi_s \rhd_{\mathcal{E}} \Gamma_0'' \vdash s : S \mid \Delta_0''; ((\alpha' : A)^{\leq 1} \cup (\gamma : B)^{\leq 1})$. Por (SUBS) con π_u seguida de (NAME) con γ, se tiene $\pi_{\texttt{LCC}\langle[\alpha]\,t\rangle} \rhd_{\mathcal{E}} \Gamma_0' \cup \Gamma_1 \vdash \texttt{LCC}\langle[\alpha]\,t\rangle \mid \Delta_0' \cup \Delta_1; \alpha : S \to A; ((\alpha' : A)^{\leq 1} \cup \gamma : B)$. Finalmente, se concluye por (REPL) con π_s, $\pi_{o'} \rhd_{\mathcal{E}} \Gamma \vdash o' \mid \Delta$.

- $\texttt{LCC} = \texttt{LCC}'[\![\gamma \setminus^{\delta} s']\!]$. Luego, por Lem. 4.3.1 con $\alpha' \in o$ y (REPL), se tienen los contextos $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta = \Delta_0 \cup \Delta_1; (\alpha' : A \cup \delta : B)$ con la premisa $\pi_{\texttt{LCC}'\langle[\alpha']\,t::s\rangle} \rhd_{\mathcal{E}} \Gamma_0 \vdash \texttt{LCC}'\langle[\alpha']\,t :: s\rangle \mid \Delta_0; (\gamma : S' \to B)^{\leq 1}; (\alpha' : A \cup (\delta : B)^{\leq 1})$ junto a la premisa $\pi_{s'} \rhd_{\mathcal{E}} \Gamma_1 \vdash s' : S' \mid \Delta_1; ((\alpha' : A)^{\leq 1} \cup (\delta : B)^{\leq 1})$. Por *h.i.* existe la derivación $\pi_{\texttt{LCC}'\langle[\alpha]\,t\rangle[\![\alpha \setminus^{\alpha'} s]\!]} \rhd_{\mathcal{E}} \Gamma_0 \vdash \texttt{LCC}'\langle[\alpha]\,t\rangle[\![\alpha \setminus^{\alpha'} s]\!] \mid \Delta_0; (\gamma : S' \to B)^{\leq 1}; (\alpha' : A \cup (\delta : B)^{\leq 1})$. Esta derivación necesariamente termina con una aplicación de de la regla (REPL). Luego, se tienen los contextos $\Gamma_0 = \Gamma_0' \cup \Gamma_0''$ y $\Delta_0 = \Delta_0' \cup \Delta_0''$ con derivaciones para $\texttt{LCC}'\langle[\alpha]\,t\rangle$, $\pi_{\texttt{LCC}'\langle[\alpha]\,t\rangle} \rhd_{\mathcal{E}} \Gamma_0' \vdash \texttt{LCC}'\langle[\alpha]\,t\rangle \mid \Delta_0'; (\alpha : S \to A)^{\leq 1}; (\gamma : S' \to B)^{\leq 1}; ((\alpha' : A)^{\leq 1} \cup (\delta : B)^{\leq 1})$; y para s, $\pi_s \rhd_{\mathcal{E}} \Gamma_0'' \vdash s : S \mid \Delta_0'; (\gamma : S' \to B)^{\leq 1}; ((\alpha' : A)^{\leq 1} \cup (\delta : B)^{\leq 1})$. Entonces, aplicando la regla (REPL) con las premisas $\pi_{\texttt{LCC}'\langle[\alpha]\,t\rangle}$ y $\pi_{s'}$, se obtiene la derivación $\pi_{\texttt{LCC}\langle[\alpha]\,t\rangle} \rhd_{\mathcal{E}} \Gamma_0' \cup \Gamma_1 \vdash \texttt{LCC}\langle[\alpha]\,t\rangle \mid \Delta_0' \cup \Delta_1; (\alpha : S \to A)^{\leq 1}; ((\alpha' : A)^{\leq 1} \cup \delta : B)$. Finalmente, se concluye por (REPL) con π_s, $\pi_{o'} \rhd_{\mathcal{E}} \Gamma \vdash o' \mid \Delta$. Notar que, sin pérdida de generalidad, puede asumirse $\gamma \notin s$ y, por Lem. 4.3.1, se tiene $(\gamma : S' \to B)^{\leq 1}$ vacío en π_s.
- $\texttt{LCC} = \texttt{LCC}'[\gamma \backslash \delta]$. Luego, por Lem. 4.3.1 con $\alpha' \in o$ y (REN), $\Delta = \Delta'; (\alpha' : A \cup \delta : B)$ y se tiene la derivación $\pi_{\texttt{LCC}'\langle[\alpha']\,t::s\rangle} \rhd_{\mathcal{E}} \Gamma \vdash \texttt{LCC}' : [\alpha']\,t :: s \mid \Delta'; (\alpha' : A \cup (\delta : B)^{\leq 1}); (\gamma : B)^{\leq 1}$. Entonces, por *h.i.* sobre $\texttt{LCC}'$, se garantiza que existe una derivación $\pi_{\texttt{LCC}'\langle[\alpha]\,t\rangle[\![\alpha \setminus^{\alpha'} s]\!]} \rhd_{\mathcal{E}} \Gamma \vdash \texttt{LCC}'\langle[\alpha]\,t\rangle[\![\alpha \setminus^{\alpha'} s]\!] \mid \Delta; (\alpha' : A \cup (\delta : B)^{\leq 1}); (\gamma : B)^{\leq 1}$. Esta derivación necesariamente termina la regla (REPL). Luego, se tienen $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta' = \Delta_0 \cup \Delta_1$ con la premisa $\pi_{\texttt{LCC}'\langle[\alpha]\,t\rangle} \rhd_{\mathcal{E}} \Gamma_0 \vdash \texttt{LCC}'\langle[\alpha]\,t\rangle \mid \Delta_0; (\alpha : S \to A)^{\leq 1}; ((\alpha' : A)^{\leq 1} \cup (\delta : B)^{\leq 1}); (\gamma : B)^{\leq 1}$ y junto a la premisa $\pi_s \rhd_{\mathcal{E}} \Gamma_1 \vdash s : S \mid \Delta_1; ((\alpha' : A)^{\leq 1} \cup (\delta : B)^{\leq 1}); (\gamma : B)^{\leq 1}$. Luego, aplicando la regla (REN) sobre $\pi_{\texttt{LCC}'\langle[\alpha]\,t\rangle}$ con γ y δ, se obtiene la derivación $\pi_{\texttt{LCC}\langle[\alpha]\,t\rangle} \rhd_{\mathcal{E}} \Gamma_0 \vdash \texttt{LCC}\langle[\alpha]\,t\rangle \mid \Delta_0; (\alpha : S \to A)^{\leq 1}; ((\alpha' : A)^{\leq 1} \cup \delta : B)$. Finalmente, se concluye aplicando (REPL) con π_s, $\pi_{o'} \rhd_{\mathcal{E}} \Gamma \vdash o' \mid \Delta$. Notar que, sin pérdida de generalidad, puede asumirse $\gamma \notin s$ y, por Lem. 4.3.1, se tiene $(\gamma : S'' \to B)^{\leq 1}$ vacío en π_s.

2. P. Luego, $o = \texttt{LCC}\langle c[\![\beta \setminus^{\alpha'} s' \cdot s]\!]\rangle$ y $o' = \texttt{LCC}\langle c[\![\beta \setminus^{\alpha} s']\!]\rangle[\![\alpha \setminus^{\alpha'} s]\!]$. Se procede por inducción en LCC.

- $\texttt{LCC} = \boxdot$. Luego, por (REPL) se tienen $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta = \Delta_0 \cup \Delta_1; \alpha' : A$ con $\pi_c \rhd_{\mathcal{E}} \Gamma_0 \vdash c \mid \Delta_0; (\beta : S'' \to A)^{\leq 1}; (\alpha' : A)^{\leq 1}$ y $\pi_{s' \cdot s} \rhd_{\mathcal{E}} \Gamma_1 \vdash s' \cdot s : S'' \mid \Delta_1; (\alpha' : A)^{\leq 1}$. Por Lem. C.0.20 (2), se tienen $S'' = S' \cdot S$, $\Gamma_1 = \Gamma_1' \cup \Gamma_1''$ y $\Delta_1 = \Delta_1' \cup \Delta_1''$ con derivaciones $\pi_{s'} \rhd_{\mathcal{E}} \Gamma_1' \vdash s' : S' \mid \Delta_1'; (\alpha' : A)^{\leq 1}$ y $\pi_s \rhd_{\mathcal{E}} \Gamma_1'' \vdash s : S \mid \Delta_1''; (\alpha' : A)^{\leq 1}$. Notar que $S'' \to A = S' \to S \to A$. Más aún, por hipótesis de la regla P, $\alpha \notin c$. Luego, puede agregarse la hipótesis $(\alpha : S \to A)^{\leq 1}$ vacía al contexto de nombres de π_c. Entonces, por (REPL) con $\pi_{s'}$, $\pi_{c[\![\beta \setminus^{\alpha} s']\!]} \rhd_{\mathcal{E}} \Gamma_0 \cup \Gamma_1' \vdash c[\![\beta \setminus^{\alpha} s']\!] \mid \Delta_0 \cup \Delta_1'; \alpha : S \to A; (\alpha' : A)^{\leq 1}$. Finalmente, se concluye por (REPL) con π_s, $\pi_{o'} \rhd_{\mathcal{E}} \Gamma \vdash o' \mid \Delta$.
- $\texttt{LCC} = [\gamma]\,\texttt{LTC}$. Luego, por Lem. 4.3.1 con $\alpha' \in o$ y (NAME), $\Delta = \Delta'; (\alpha' : A \cup \gamma : B)$ y se tiene la derivación $\pi_{\texttt{LTC}\langle c[\![\beta \setminus^{\alpha'} s' \cdot s]\!]\rangle} \rhd_{\mathcal{E}} \Gamma \vdash \texttt{LTC}\langle c[\![\beta \setminus^{\alpha'} s' \cdot s]\!]\rangle : B \mid \Delta'; (\alpha' : A \cup (\gamma : B)^{\leq 1})$. Se procede analizando la forma de LTC.
  - $\texttt{LTC} = \texttt{LTC}'\,u$. Luego, por (APP), se tienen contextos $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta' = \Delta_0 \cup \Delta_1$ con $\pi_{\texttt{LTC}'\langle c[\![\beta \setminus^{\alpha'} s' \cdot s]\!]\rangle} \rhd_{\mathcal{E}} \Gamma_0 \vdash \texttt{LTC}'\langle c[\![\beta \setminus^{\alpha'} s' \cdot s]\!]\rangle : B' \to B \mid \Delta_0; (\alpha' : A \cup (\gamma : B)^{\leq 1})$ y $\pi_u \rhd_{\mathcal{E}} \Gamma_1 \vdash u : B' \mid \Delta_1; (\alpha' : A \cup (\gamma : B)^{\leq 1})$. Considerar $p = \texttt{LCC}'\langle c[\![\beta \setminus^{\alpha'} s' \cdot s]\!]\rangle$ donde $\texttt{LCC}' = [\delta]\,\texttt{LTC}'$ con δ un nombre fresco. Entonces, por (NAME), se tiene la derivación $\pi_p \rhd_{\mathcal{E}} \Gamma_0 \vdash p \mid \Delta_0; (\alpha' : A \cup (\gamma : B)^{\leq 1}); \delta : B' \to B$. Más aún, se tiene $p' = \texttt{LCC}'\langle c[\![\beta \setminus^{\alpha} s']\!]\rangle[\![\alpha \setminus^{\alpha'} s]\!] \to_{\texttt{P}} p$. Luego, por *h.i.* existe una derivación $\pi_{p'} \rhd_{\mathcal{E}} \Gamma_0 \vdash p' \mid \Delta_0; (\alpha' : A \cup (\gamma : B)^{\leq 1}); \delta : B' \to B$. De (REPL) y (NAME), dado que δ es fresco, se tienen los contextos $\Gamma_0 = \Gamma_0' \cup \Gamma_0''$ y $\Delta_0 = \Delta_0' \cup \Delta_0''$ con $\pi_{\texttt{LTC}'\langle c[\![\beta \setminus^{\alpha} s']\!]\rangle} \rhd_{\mathcal{E}} \Gamma_0' \vdash \texttt{LTC}'\langle c[\![\beta \setminus^{\alpha} s']\!]\rangle : B' \to B \mid \Delta_0'; \alpha : S \to A; ((\alpha' : A)^{\leq 1} \cup (\gamma : B)^{\leq 1})$ y la premisa $\pi_s \rhd_{\mathcal{E}} \Gamma_0'' \vdash s : S \mid \Delta_0''; ((\alpha' : A)^{\leq 1} \cup (\gamma : B)^{\leq 1})$. Luego, aplicando la regla (APP) con π_u seguida de la regla (NAME) con γ, se obtiene la derivación $\pi_{\texttt{LCC}\langle c[\![\beta \setminus^{\alpha} s']\!]\rangle} \rhd_{\mathcal{E}} \Gamma_0' \cup \Gamma_1 \vdash \texttt{LCC}\langle c[\![\beta \setminus^{\alpha} s']\!]\rangle \mid \Delta_0' \cup \Delta_1; \alpha : S \to A; ((\alpha' : A)^{\leq 1} \cup \gamma : B)$. Finalmente, se concluye por (REPL) con π_s, $\pi_{o'} \rhd_{\mathcal{E}} \Gamma \vdash o' \mid \Delta$.
  - $\texttt{LTC} = \lambda x.\texttt{LTC}'$. Luego, por (ABS), se tiene $B = B' \to B''$ con $\pi_{\texttt{LTC}'\langle c[\![\beta \setminus^{\alpha'} s' \cdot s]\!]\rangle} \rhd_{\mathcal{E}} \Gamma; (x : B')^{\leq 1} \vdash \texttt{LTC}'\langle c[\![\beta \setminus^{\alpha'} s' \cdot s]\!]\rangle : B'' \mid \Delta'; (\alpha' : A \cup (\gamma : B)^{\leq 1})$. Considerar el objeto $p = \texttt{LCC}'\langle c[\![\beta \setminus^{\alpha'} s' \cdot s]\!]\rangle$ donde $\texttt{LCC}' = [\delta]\,\texttt{LTC}'$ con δ un nombre fresco. Entonces, por (NAME), se tiene $\pi_p \rhd_{\mathcal{E}} \Gamma; (x : B')^{\leq 1} \vdash p \mid \Delta'; (\alpha' : A \cup (\gamma : B)^{\leq 1}); \delta : B''$. Más aún, como $p' = \texttt{LCC}'\langle c[\![\beta \setminus^{\alpha} s']\!]\rangle[\![\alpha \setminus^{\alpha'} s]\!] \to_{\texttt{P}} p$, por *h.i.* existe una derivación $\pi_{p'} \rhd_{\mathcal{E}} \Gamma; (x : B')^{\leq 1} \vdash p' \mid \Delta'; (\alpha' : A \cup (\gamma : B)^{\leq 1}); \delta : B''$. De (REPL) y (NAME), dado que δ es fresco, se tienen los contextos $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta' = \Delta_0 \cup \Delta_1$ con $\pi_{\texttt{LTC}'\langle c[\![\beta \setminus^{\alpha} s']\!]\rangle} \rhd_{\mathcal{E}} \Gamma_0; (x : B')^{\leq 1} \vdash \texttt{LTC}'\langle c[\![\beta \setminus^{\alpha} s']\!]\rangle : B'' \mid \Delta_0; \alpha : S \to A; ((\alpha' : A)^{\leq 1} \cup (\gamma : B)^{\leq 1})$ junto a $\pi_s \rhd_{\mathcal{E}} \Gamma_1 \vdash s : S \mid \Delta_1; ((\alpha' : A)^{\leq 1} \cup (\gamma : B)^{\leq 1})$. Luego, aplicando la regla (ABS) seguida de una aplicación de (NAME) con γ, se obtiene la derivación $\pi_{\texttt{LCC}\langle c[\![\beta \setminus^{\alpha} s']\!]\rangle} \rhd_{\mathcal{E}} \Gamma_0 \vdash \texttt{LCC}\langle c[\![\beta \setminus^{\alpha} s']\!]\rangle \mid \Delta_0; \alpha : S \to A; ((\alpha' : A)^{\leq 1} \cup \gamma : B)$. Por último, se concluye por (REPL) con π_s, $\pi_{o'} \rhd_{\mathcal{E}} \Gamma \vdash o' \mid \Delta$.
  - $\texttt{LTC} = \mu\delta.\texttt{LCC}'$. Luego, de $\pi_{\texttt{LTC}\langle c[\![\beta \setminus^{\alpha'} s' \cdot s]\!]\rangle}$ por (CONT), se tiene una derivación para la premisa $\pi_{\texttt{LCC}'\langle c[\![\beta \setminus^{\alpha'} s' \cdot s]\!]\rangle} \rhd_{\mathcal{E}} \Gamma \vdash \texttt{LCC}'\langle c[\![\beta \setminus^{\alpha'} s' \cdot s]\!]\rangle \mid \Delta'; (\alpha' : A \cup (\gamma : B)^{\leq 1}); (\delta : B)^{\leq 1}$. Por *h.i.* sobre LCC′, se garantiza que existe una derivación $\pi_{\texttt{LCC}'\langle c[\![\beta \setminus^{\alpha} s']\!]\rangle[\![\alpha \setminus^{\alpha'} s]\!]} \rhd_{\mathcal{E}} \Gamma \vdash \texttt{LCC}'\langle c[\![\beta \setminus^{\alpha} s']\!]\rangle[\![\alpha \setminus^{\alpha'} s]\!] \mid \Delta'; (\alpha' : A \cup (\gamma : B)^{\leq 1}); (\delta : B)^{\leq 1}$. Esta derivación necesariamente termina con una aplicación de de la regla (REPL). Luego, se tienen los contextos $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta' = \Delta_0 \cup \Delta_1$ con la derivación $\pi_{\texttt{LCC}'\langle c[\![\beta \setminus^{\alpha} s']\!]\rangle} \rhd_{\mathcal{E}} \Gamma_0 \vdash \texttt{LCC}'\langle c[\![\beta \setminus^{\alpha} s']\!]\rangle \mid \Delta_0; \alpha : S \to A; ((\alpha' : A)^{\leq 1} \cup (\gamma : B)^{\leq 1}); (\delta : B)^{\leq 1}$ junto a la derivación $\pi_s \rhd_{\mathcal{E}} \Gamma_1 \vdash s : S \mid \Delta_1; ((\alpha' : A)^{\leq 1} \cup (\gamma : B)^{\leq 1}); (\delta : B)^{\leq 1}$. Luego, aplicando (CONT) con δ seguida de (NAME) con γ, se obtiene la derivación $\pi_{\texttt{LCC}\langle c[\![\beta \setminus^{\alpha} s']\!]\rangle} \rhd_{\mathcal{E}} \Gamma_0 \vdash \texttt{LCC}\langle c[\![\beta \setminus^{\alpha} s']\!]\rangle \mid \Delta_0; \alpha : S \to A; ((\alpha' : A)^{\leq 1} \cup \gamma : B)$. Por último, se concluye por (REPL) con π_s, $\pi_{o'} \rhd_{\mathcal{E}} \Gamma \vdash o' \mid \Delta$.
  - $\texttt{LTC} = \texttt{LTC}'[x \setminus u]$. Luego, por (SUBS), se tienen $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta' = \Delta_0 \cup \Delta_1$ con

$\pi_{\mathtt{LTC}'\langle c[\![\beta \setminus^{\alpha'} s' \cdot s]\!]\rangle} \rhd_{\mathcal{E}} \Gamma_0; (x : B')^{\leq 1} \vdash \mathtt{LTC}'\langle c[\![\beta \setminus^{\alpha'} s' \cdot s]\!]\rangle : B \mid \Delta_0; (\alpha' : A \cup (\gamma : B)^{\leq 1})$ y $\pi_u \rhd_{\mathcal{E}} \Gamma_1 \vdash u : B' \mid \Delta_1; ((\alpha' : A)^{\leq 1} \cup (\gamma : B)^{\leq 1})$. Considerar $p = \mathtt{LCC}'\langle c[\![\beta \setminus^{\alpha'} s' \cdot s]\!]\rangle$ donde $\mathtt{LCC}' = [\gamma]\,\mathtt{LTC}'$. Por (NAME), $\pi_p \rhd_{\mathcal{E}} \Gamma_0; (x : B')^{\leq 1} \vdash p \mid \Delta_0; (\alpha' : A \cup (\gamma : B)^{\leq 1})$. Más aún, se tiene $p' = \mathtt{LCC}'\langle c[\![\beta \setminus^{\alpha} s']\!]\rangle[\![\alpha \setminus^{\alpha'} s]\!] \to_{\mathtt{p}} p$. Luego, por *h.i.* existe una derivación $\pi_{p'} \rhd_{\mathcal{E}} \Gamma_0; (x : B')^{\leq 1} \vdash p' \mid \Delta_0; (\alpha' : A \cup (\gamma : B)^{\leq 1})$. De (REPL) y (NAME), se tienen los contextos $\Gamma_0 = \Gamma_0' \cup \Gamma_0''$ y $\Delta_0 = \Delta_0' \cup \Delta_0''$ con las premisas $\pi_{\mathtt{LTC}'\langle c[\![\beta \setminus^{\alpha} s']\!]\rangle} \rhd_{\mathcal{E}} \Gamma_0'; (x : B')^{\leq 1} \vdash \mathtt{LTC}'\langle c[\![\beta \setminus^{\alpha} s']\!]\rangle : B \mid \Delta_0'; \alpha : S \to A; ((\alpha' : A)^{\leq 1} \cup (\gamma : B)^{\leq 1})$ y $\pi_s \rhd_{\mathcal{E}} \Gamma_0'' \vdash s : S \mid \Delta_0''; ((\alpha' : A)^{\leq 1} \cup (\gamma : B)^{\leq 1})$. Entonces, aplicando la regla (SUBS) con π_u seguida de la regla (NAME) con γ, se obtiene la derivación $\pi_{\mathtt{LCC}\langle c[\![\beta \setminus^{\alpha} s']\!]\rangle} \rhd_{\mathcal{E}} \Gamma_0' \cup \Gamma_1 \vdash \mathtt{LCC}\langle c[\![\beta \setminus^{\alpha} s']\!]\rangle \mid \Delta_0' \cup \Delta_1; \alpha : S \to A; ((\alpha' : A)^{\leq 1} \cup \gamma : B)$. Finalmente, se concluye por (REPL) con π_s, $\pi_{o'} \rhd_{\mathcal{E}} \Gamma \vdash o' \mid \Delta$.

- $\mathtt{LCC} = \mathtt{LCC}'[\![\gamma \setminus^{\delta} s'']\!]$. Luego, por Lem. 4.3.1 con $\alpha' \in o$ y (REPL), se tienen los contextos $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta = \Delta_0 \cup \Delta_1; (\alpha' : A \cup \delta : B)$ con la derivación $\pi_{\mathtt{LCC}'\langle c[\![\beta \setminus^{\alpha'} s' \cdot s]\!]\rangle} \rhd_{\mathcal{E}} \Gamma_0 \vdash \mathtt{LCC}'\langle c[\![\beta \setminus^{\alpha'} s' \cdot s]\!]\rangle \mid \Delta_0; (\gamma : S'' \to B)^{\leq 1}; (\alpha' : A \cup (\delta : B)^{\leq 1})$ y la derivación $\pi_{s''} \rhd_{\mathcal{E}} \Gamma_1 \vdash s'' : S'' \mid \Delta_1; ((\alpha' : A)^{\leq 1} \cup (\delta : B)^{\leq 1})$. Entonces, por *h.i.* existe $\pi_{\mathtt{LCC}'\langle c[\![\beta \setminus^{\alpha} s']\!]\rangle[\![\alpha \setminus^{\alpha'} s]\!]} \rhd_{\mathcal{E}} \Gamma_0 \vdash \mathtt{LCC}'\langle c[\![\beta \setminus^{\alpha} s']\!]\rangle[\![\alpha \setminus^{\alpha'} s]\!] \mid \Delta_0; (\gamma : S'' \to B)^{\leq 1}; (\alpha' : A \cup (\delta : B)^{\leq 1})$. Por (REPL) se tienen los contextos $\Gamma_0 = \Gamma_0' \cup \Gamma_0''$ y $\Delta_0 = \Delta_0' \cup \Delta_0''$ con la premisa $\pi_{\mathtt{LCC}'\langle c[\![\beta \setminus^{\alpha} s']\!]\rangle} \rhd_{\mathcal{E}} \Gamma_0' \vdash \mathtt{LCC}'\langle c[\![\beta \setminus^{\alpha} s']\!]\rangle \mid \Delta_0'; (\alpha : S \to A)^{\leq 1}; (\gamma : S'' \to B)^{\leq 1}; ((\alpha' : A)^{\leq 1} \cup (\delta : B)^{\leq 1})$ y la premisa $\pi_s \rhd_{\mathcal{E}} \Gamma_0'' \vdash s : S \mid \Delta_0'; (\gamma : S'' \to B)^{\leq 1}; ((\alpha' : A)^{\leq 1} \cup (\delta : B)^{\leq 1})$. Luego, aplicando la regla (REPL) con $\pi_{\mathtt{LCC}'\langle c[\![\beta \setminus^{\alpha} s']\!]\rangle}$ y $\pi_{s''}$, se obtiene la derivación $\pi_{\mathtt{LCC}\langle c[\![\beta \setminus^{\alpha} s']\!]\rangle} \rhd_{\mathcal{E}} \Gamma_0' \cup \Gamma_1 \vdash \mathtt{LCC}\langle c[\![\beta \setminus^{\alpha} s']\!]\rangle \mid \Delta_0' \cup \Delta_1; (\alpha : S \to A)^{\leq 1}; ((\alpha' : A)^{\leq 1} \cup \delta : B)$. Finalmente, se concluye por (REPL) con π_s, $\pi_{o'} \rhd_{\mathcal{E}} \Gamma \vdash o' \mid \Delta$. Notar que, sin pérdida de generalidad, puede asumirse $\gamma \notin s$ y, por Lem. 4.3.1, se tiene $(\gamma : S'' \to B)^{\leq 1}$ vacío en π_s.
- $\mathtt{LCC} = \mathtt{LCC}'[\gamma \setminus \delta]$. Luego, por Lem. 4.3.1 con $\alpha' \in o$ y (REN), $\Delta = \Delta'; (\alpha' : A \cup \delta : B)$ con $\pi_{\mathtt{LCC}'\langle c[\![\beta \setminus^{\alpha'} s' \cdot s]\!]\rangle} \rhd_{\mathcal{E}} \Gamma \vdash \mathtt{LCC}'\langle c[\![\beta \setminus^{\alpha'} s' \cdot s]\!]\rangle \mid \Delta'; (\alpha' : A \cup (\delta : B)^{\leq 1}); (\gamma : B)^{\leq 1}$. Entonces, por *h.i.* sobre $\mathtt{LCC}'$, se garantiza que existe una derivación $\pi_{\mathtt{LCC}'\langle c[\![\beta \setminus^{\alpha} s']\!]\rangle[\![\alpha \setminus^{\alpha'} s]\!]} \rhd_{\mathcal{E}} \Gamma \vdash \mathtt{LCC}'\langle c[\![\beta \setminus^{\alpha} s']\!]\rangle[\![\alpha \setminus^{\alpha'} s]\!] \mid \Delta; (\alpha' : A \cup (\delta : B)^{\leq 1}); (\gamma : B)^{\leq 1}$. Más aún, por (REPL), se tienen los contextos $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta' = \Delta_0 \cup \Delta_1$ con la premisa $\pi_{\mathtt{LCC}'\langle c[\![\beta \setminus^{\alpha} s']\!]\rangle} \rhd_{\mathcal{E}} \Gamma_0 \vdash \mathtt{LCC}'\langle c[\![\beta \setminus^{\alpha} s']\!]\rangle \mid \Delta_0; (\alpha : S \to A)^{\leq 1}; ((\alpha' : A)^{\leq 1} \cup (\delta : B)^{\leq 1}); (\gamma : B)^{\leq 1}$ y la premisa $\pi_s \rhd_{\mathcal{E}} \Gamma_1 \vdash s : S \mid \Delta_1; ((\alpha' : A)^{\leq 1} \cup (\delta : B)^{\leq 1}); (\gamma : B)^{\leq 1}$. Luego, aplicando la regla (REN) sobre $\pi_{\mathtt{LCC}'\langle c[\![\beta \setminus^{\alpha} s']\!]\rangle}$ con los nombres γ y δ, se obtiene la derivación $\pi_{\mathtt{LCC}\langle c[\![\beta \setminus^{\alpha} s']\!]\rangle} \rhd_{\mathcal{E}} \Gamma_0 \vdash \mathtt{LCC}\langle c[\![\beta \setminus^{\alpha} s']\!]\rangle \mid \Delta_0; (\alpha : S \to A)^{\leq 1}; ((\alpha' : A)^{\leq 1} \cup \delta : B)$. Por último, se concluye por (REPL) con π_s, $\pi_{o'} \rhd_{\mathcal{E}} \Gamma \vdash o' \mid \Delta$. Notar que, sin pérdida de generalidad, puede asumirse $\gamma \notin s$ y, por Lem. 4.3.1, se tiene $(\gamma : S'' \to B)^{\leq 1}$ vacío en π_s.

3. W. Luego, $o = \mathtt{LCC}\langle c[\![\beta \setminus^{\alpha} s]\!][\alpha \setminus \alpha']\rangle$ y $o' = \mathtt{LCC}\langle c[\beta \setminus \alpha]\rangle[\![\alpha \setminus^{\alpha'} s]\!]$. Se procede por inducción en LCC.

   - $\mathtt{LCC} = \boxdot$. Luego, por (REN) y (REPL), $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta = \Delta_0 \cup \Delta_1; \alpha' : A$ con $\pi_c \rhd_{\mathcal{E}} \Gamma_0 \vdash c \mid \Delta_0; (\alpha : A)^{\leq 1}; (\alpha' : A)^{\leq 1}; (\beta : S \to A)^{\leq 1}$ y $\pi_s \rhd_{\mathcal{E}} \Gamma_1 \vdash s : S \mid \Delta_1; (\alpha' : A)^{\leq 1}$. Notar que, por hipótesis de la regla W, $\alpha \notin c$. Luego, por Lem. 4.3.1, $(\alpha : A)^{\leq 1}$ es vacío en π_c. Más aún, puede agregarse la hipótesis $(\alpha : S \to A)^{\leq 1}$ vacía también a π_c y, por (REN), se obtiene la derivación $\pi_{c[\beta \setminus \alpha]} \rhd_{\mathcal{E}} \Gamma_0 \vdash c[\beta \setminus \alpha] \mid \Delta_0; \alpha : S \to A; (\alpha' : A)^{\leq 1}$. Finalmente, se concluye por (REPL) con π_s, $\pi_{o'} \rhd_{\mathcal{E}} \Gamma \vdash o' \mid \Delta$.
   - $\mathtt{LCC} = [\gamma]\,\mathtt{LTC}$. Luego, por Lem. 4.3.1 con $\alpha' \in o$ y (NAME), $\Delta = \Delta'; (\alpha' : A \cup \gamma : B)$ con la derivación $\pi_{\mathtt{LTC}\langle c[\![\beta \setminus^{\alpha} s]\!][\alpha \setminus \alpha']\rangle} \rhd_{\mathcal{E}} \Gamma \vdash \mathtt{LTC}\langle c[\![\beta \setminus^{\alpha} s]\!][\alpha \setminus \alpha']\rangle : B \mid \Delta'; (\alpha' : A \cup (\gamma : B)^{\leq 1})$. Se procede analizando la forma de LTC.
     - $\mathtt{LTC} = \mathtt{LTC}'\,u$. Luego, por (APP), se tienen $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta' = \Delta_0 \cup \Delta_1$ con premisas $\pi_{\mathtt{LTC}'\langle c[\![\beta \setminus^{\alpha} s]\!][\alpha \setminus \alpha']\rangle} \rhd_{\mathcal{E}} \Gamma_0 \vdash \mathtt{LTC}'\langle c[\![\beta \setminus^{\alpha} s]\!][\alpha \setminus \alpha']\rangle : B' \to B \mid \Delta_0; (\alpha' : A \cup (\gamma : B)^{\leq 1})$

y $\pi_u \rhd_{\mathcal{E}} \Gamma_1 \vdash u : B' \mid \Delta_1; (\alpha' : A \cup (\gamma : B)^{\leq 1})$. Considerar $p = \mathtt{LCC}'\langle c[\![\beta \setminus^{\alpha} s]\!][\alpha \setminus \alpha']\rangle$ donde $\mathtt{LCC}' = [\delta]\,\mathtt{LTC}'$ con δ un nombre fresco. Entonces, por (NAME), se tiene la derivación $\pi_p \rhd_{\mathcal{E}} \Gamma_0 \vdash p \mid \Delta_0; (\alpha' : A \cup (\gamma : B)^{\leq 1}); \delta : B' \to B$. Más aún, se tiene $p' = \mathtt{LCC}'\langle c[\beta \setminus \alpha]\rangle[\![\alpha \setminus^{\alpha'} s]\!] \to_{\mathtt{W}} p$. Luego, por *h.i.* se garantiza que existe una derivación $\pi_{p'} \rhd_{\mathcal{E}} \Gamma_0 \vdash p' \mid \Delta_0; (\alpha' : A \cup (\gamma : B)^{\leq 1}); \delta : B' \to B$. De (REPL) y (NAME), dado que δ es fresco, se tienen los contextos $\Gamma_0 = \Gamma_0' \cup \Gamma_0''$ y $\Delta_0 = \Delta_0' \cup \Delta_0''$ con las derivaciones $\pi_{\mathtt{LTC}'\langle c[\beta\setminus\alpha]\rangle} \rhd_{\mathcal{E}} \Gamma_0' \vdash \mathtt{LTC}'\langle c[\beta \setminus \alpha]\rangle : B' \to B \mid \Delta_0'; \alpha : S \to A; ((\alpha' : A)^{\leq 1} \cup (\gamma : B)^{\leq 1})$ y $\pi_s \rhd_{\mathcal{E}} \Gamma_0'' \vdash s : S \mid \Delta_0''; ((\alpha' : A)^{\leq 1} \cup (\gamma : B)^{\leq 1})$. Luego, aplicando la regla (APP) con π_u seguida de una aplicación de (NAME) con γ, se obtiene la derivación $\pi_{\mathtt{LCC}\langle c[\beta\setminus\alpha]\rangle} \rhd_{\mathcal{E}} \Gamma_0' \cup \Gamma_1 \vdash \mathtt{LCC}\langle c[\beta \setminus \alpha]\rangle \mid \Delta_0' \cup \Delta_1; \alpha : S \to A; ((\alpha' : A)^{\leq 1} \cup \gamma : B)$. Se concluye por (REPL) con π_s, $\pi_{o'} \rhd_{\mathcal{E}} \Gamma \vdash o' \mid \Delta$.

- $\mathtt{LTC} = \lambda x.\mathtt{LTC}'$. Por (ABS), se tiene $B = B' \to B''$ con $\pi_{\mathtt{LTC}'\langle c[\![\beta\setminus^{\alpha} s]\!][\alpha\setminus\alpha']\rangle} \rhd_{\mathcal{E}} \Gamma; (x : B')^{\leq 1} \vdash \mathtt{LTC}'\langle c[\![\beta \setminus^{\alpha} s]\!][\alpha \setminus \alpha']\rangle : B'' \mid \Delta'; (\alpha' : A \cup (\gamma : B)^{\leq 1})$. Considerar $p = \mathtt{LCC}'\langle c[\![\beta \setminus^{\alpha} s]\!][\alpha \setminus \alpha']\rangle$ donde $\mathtt{LCC}' = [\delta]\,\mathtt{LTC}'$ con δ un nombre fresco. Entonces, por (NAME), se tiene la derivación $\pi_p \rhd_{\mathcal{E}} \Gamma; (x : B')^{\leq 1} \vdash p \mid \Delta'; (\alpha' : A \cup (\gamma : B)^{\leq 1}); \delta : B''$. Más aún, se tiene $p' = \mathtt{LCC}'\langle c[\beta \setminus \alpha]\rangle[\![\alpha \setminus^{\alpha'} s]\!] \to_{\mathtt{W}} p$. Luego, por *h.i.* existe una derivación $\pi_{p'} \rhd_{\mathcal{E}} \Gamma; (x : B')^{\leq 1} \vdash p' \mid \Delta'; (\alpha' : A \cup (\gamma : B)^{\leq 1}); \delta : B''$. De (REPL) y (NAME), dado que δ es fresco, se tienen $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta' = \Delta_0 \cup \Delta_1$ con $\pi_{\mathtt{LTC}'\langle c[\beta\setminus\alpha]\rangle} \rhd_{\mathcal{E}} \Gamma_0; (x : B')^{\leq 1} \vdash \mathtt{LTC}'\langle c[\beta \setminus \alpha]\rangle : B'' \mid \Delta_0; \alpha : S \to A; ((\alpha' : A)^{\leq 1} \cup (\gamma : B)^{\leq 1})$ y $\pi_s \rhd_{\mathcal{E}} \Gamma_1 \vdash s : S \mid \Delta_1; ((\alpha' : A)^{\leq 1} \cup (\gamma : B)^{\leq 1})$. Entonces, aplicando la regla (ABS) seguida de una aplicación de (NAME) con el nombre γ, se obtiene la derivación $\pi_{\mathtt{LCC}\langle c[\beta\setminus\alpha]\rangle} \rhd_{\mathcal{E}} \Gamma_0 \vdash \mathtt{LCC}\langle c[\beta \setminus \alpha]\rangle \mid \Delta_0; \alpha : S \to A; ((\alpha' : A)^{\leq 1} \cup \gamma : B)$. Finalmente, se concluye por (REPL) con π_s, $\pi_{o'} \rhd_{\mathcal{E}} \Gamma \vdash o' \mid \Delta$.
- $\mathtt{LTC} = \mu\delta.\mathtt{LCC}'$. Luego, por (CONT), se tiene la derivación $\pi_{\mathtt{LCC}'\langle c[\![\beta\setminus^{\alpha} s]\!][\alpha\setminus\alpha']\rangle} \rhd_{\mathcal{E}} \Gamma \vdash \mathtt{LCC}'\langle c[\![\beta \setminus^{\alpha} s]\!][\alpha \setminus \alpha']\rangle \mid \Delta'; (\alpha' : A \cup (\gamma : B)^{\leq 1}); (\delta : B)^{\leq 1}$. Más aún, por *h.i.* sobre $\mathtt{LCC}'$ existe una derivación para la expansión del objeto $\pi_{\mathtt{LCC}'\langle c[\beta\setminus\alpha]\rangle[\![\alpha\setminus^{\alpha'} s]\!]} \rhd_{\mathcal{E}} \Gamma \vdash \mathtt{LCC}'\langle c[\beta \setminus \alpha]\rangle[\![\alpha \setminus^{\alpha'} s]\!] \mid \Delta'; (\alpha' : A \cup (\gamma : B)^{\leq 1}); (\delta : B)^{\leq 1}$. Esta derivación necesariamente termina con una aplicación de de la regla (REPL). Luego, se tienen los contextos $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta' = \Delta_0 \cup \Delta_1$ con derivaciones para la premisa $\pi_{\mathtt{LCC}'\langle c[\beta\setminus\alpha]\rangle} \rhd_{\mathcal{E}} \Gamma_0 \vdash \mathtt{LCC}'\langle c[\beta \setminus \alpha]\rangle \mid \Delta_0; \alpha : S \to A; ((\alpha' : A)^{\leq 1} \cup (\gamma : B)^{\leq 1}); (\delta : B)^{\leq 1}$ y la premisa $\pi_s \rhd_{\mathcal{E}} \Gamma_1 \vdash s : S \mid \Delta_1; ((\alpha' : A)^{\leq 1} \cup (\gamma : B)^{\leq 1}); (\delta : B)^{\leq 1}$. Entonces, aplicando la regla (CONT) con δ y luego la regla (NAME) con γ, se obtiene la derivación $\pi_{\mathtt{LCC}\langle c[\beta\setminus\alpha]\rangle} \rhd_{\mathcal{E}} \Gamma_0 \vdash \mathtt{LCC}\langle c[\beta \setminus \alpha]\rangle \mid \Delta_0; \alpha : S \to A; ((\alpha' : A)^{\leq 1} \cup \gamma : B)$. Por último, se concluye por (REPL) con π_s, $\pi_{o'} \rhd_{\mathcal{E}} \Gamma \vdash o' \mid \Delta$.
- $\mathtt{LTC} = \mathtt{LTC}'[x \setminus u]$. Luego, de $\pi_{\mathtt{LTC}\langle c[\![\beta\setminus^{\alpha} s]\!][\alpha\setminus\alpha']\rangle}$ por (SUBS), se tienen los contextos $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta' = \Delta_0 \cup \Delta_1$ con derivaciones para la premisa $\pi_{\mathtt{LTC}'\langle c[\![\beta\setminus^{\alpha} s]\!][\alpha\setminus\alpha']\rangle} \rhd_{\mathcal{E}} \Gamma_0; (x : B')^{\leq 1} \vdash \mathtt{LTC}'\langle c[\![\beta \setminus^{\alpha} s]\!][\alpha \setminus \alpha']\rangle : B \mid \Delta_0; (\alpha' : A \cup (\gamma : B)^{\leq 1})$ y para la premisa $\pi_u \rhd_{\mathcal{E}} \Gamma_1 \vdash u : B' \mid \Delta_1; ((\alpha' : A)^{\leq 1} \cup (\gamma : B)^{\leq 1})$. Considerar entonces el objeto $p = \mathtt{LCC}'\langle c[\![\beta \setminus^{\alpha} s]\!][\alpha \setminus \alpha']\rangle$ donde $\mathtt{LCC}' = [\gamma]\,\mathtt{LTC}'$. Aplicando (NAME) con γ, se obtiene $\pi_p \rhd_{\mathcal{E}} \Gamma_0; (x : B')^{\leq 1} \vdash p \mid \Delta_0; (\alpha' : A \cup (\gamma : B)^{\leq 1})$. Más aún, se tiene $p' = \mathtt{LCC}'\langle c[\beta \setminus \alpha]\rangle[\![\alpha \setminus^{\alpha'} s]\!] \to_{\mathtt{W}} p$. Luego, por *h.i.* existe una derivación para la expansión de p, $\pi_{p'} \rhd_{\mathcal{E}} \Gamma_0; (x : B')^{\leq 1} \vdash p' \mid \Delta_0; (\alpha' : A \cup (\gamma : B)^{\leq 1})$. Esta derivación necesariamente termina con aplicaciones de las reglas (REPL) y (NAME) con γ. Luego, se tienen los contextos $\Gamma_0 = \Gamma_0' \cup \Gamma_0''$ y $\Delta_0 = \Delta_0' \cup \Delta_0''$ con las premisas $\pi_{\mathtt{LTC}'\langle c[\beta\setminus\alpha]\rangle} \rhd_{\mathcal{E}} \Gamma_0'; (x : B')^{\leq 1} \vdash \mathtt{LTC}'\langle c[\beta \setminus \alpha]\rangle : B \mid \Delta_0'; \alpha : S \to A; ((\alpha' : A)^{\leq 1} \cup (\gamma : B)^{\leq 1})$ y $\pi_s \rhd_{\mathcal{E}} \Gamma_0'' \vdash s : S \mid \Delta_0''; ((\alpha' : A)^{\leq 1} \cup (\gamma : B)^{\leq 1})$. Entonces, aplicando la regla (SUBS) con π_u seguida de una aplicación de (NAME) con γ, se obtiene la derivación $\pi_{\mathtt{LCC}\langle c[\beta\setminus\alpha]\rangle} \rhd_{\mathcal{E}} \Gamma_0' \cup \Gamma_1 \vdash \mathtt{LCC}\langle c[\beta \setminus \alpha]\rangle \mid \Delta_0' \cup \Delta_1; \alpha : S \to A; ((\alpha' : A)^{\leq 1} \cup \gamma : B)$. Se concluye por

(REPL) con π_s, $\pi_{o'} \rhd_{\mathcal{E}} \Gamma \vdash o' \mid \Delta$.

- $\mathtt{LCC} = \mathtt{LCC}'[\![\gamma \setminus^{\delta} s']\!]$. Luego, por Lem. 4.3.1 con $\alpha' \in o$ y (REPL), se tienen los contextos $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta = \Delta_0 \cup \Delta_1; (\alpha' : A \cup \delta : B)$ con la premisa $\pi_{\mathtt{LCC}'\langle c[\![\beta \setminus^{\alpha} s]\!][\alpha \setminus \alpha']\rangle} \rhd_{\mathcal{E}} \Gamma_0 \vdash \mathtt{LCC}'\langle c[\![\beta \setminus^{\alpha} s]\!][\alpha \setminus \alpha']\rangle \mid \Delta_0; (\gamma : S' \to B)^{\leq 1}; (\alpha' : A \cup (\delta : B)^{\leq 1})$ y la premisa $\pi_{s'} \rhd_{\mathcal{E}} \Gamma_1 \vdash s' : S' \mid \Delta_1; ((\alpha' : A)^{\leq 1} \cup (\delta : B)^{\leq 1})$. Por *h.i.* sobre $\mathtt{LCC}'$, existe $\pi_{\mathtt{LCC}'\langle c[\beta \setminus \alpha]\rangle[\![\alpha \setminus^{\alpha'} s]\!]} \rhd_{\mathcal{E}} \Gamma_0 \vdash \mathtt{LCC}'\langle c[\beta \setminus \alpha]\rangle[\![\alpha \setminus^{\alpha'} s]\!] \mid \Delta_0; (\gamma : S' \to B)^{\leq 1}; (\alpha' : A \cup (\delta : B)^{\leq 1})$. Esta derivación necesariamente termina con una aplicación de de la regla (REPL). Luego, se tienen los contextos $\Gamma_0 = \Gamma_0' \cup \Gamma_0''$ y $\Delta_0 = \Delta_0' \cup \Delta_0''$ con derivaciones para las premisas $\pi_{\mathtt{LCC}'\langle c[\beta \setminus \alpha]\rangle} \rhd_{\mathcal{E}} \Gamma_0' \vdash \mathtt{LCC}'\langle c[\beta \setminus \alpha]\rangle \mid \Delta_0'; (\alpha : S \to A)^{\leq 1}; (\gamma : S' \to B)^{\leq 1}; ((\alpha' : A)^{\leq 1} \cup (\delta : B)^{\leq 1})$ y $\pi_s \rhd_{\mathcal{E}} \Gamma_0'' \vdash s : S \mid \Delta_0'; (\gamma : S' \to B)^{\leq 1}; ((\alpha' : A)^{\leq 1} \cup (\delta : B)^{\leq 1})$. Entonces, puede aplicarse la regla (REPL) con las premisas $\pi_{\mathtt{LCC}'\langle c[\beta \setminus \alpha]\rangle}$ y $\pi_{s'}$, obteniendo la derivación $\pi_{\mathtt{LCC}\langle c[\beta \setminus \alpha]\rangle} \rhd_{\mathcal{E}} \Gamma_0' \cup \Gamma_1 \vdash \mathtt{LCC}\langle c[\beta \setminus \alpha]\rangle \mid \Delta_0' \cup \Delta_1; (\alpha : S \to A)^{\leq 1}; ((\alpha' : A)^{\leq 1} \cup \delta : B)$. Se concluye por (REPL) con π_s, $\pi_{o'} \rhd_{\mathcal{E}} \Gamma \vdash o' \mid \Delta$. Notar que, sin pérdida de generalidad, puede asumirse $\gamma \notin s$ y, por Lem. 4.3.1, se tiene $(\gamma : S' \to B)^{\leq 1}$ vacío en π_s.
- $\mathtt{LCC} = \mathtt{LCC}'[\gamma \setminus \delta]$. Luego, por Lem. 4.3.1 con $\alpha' \in o$ y (REN), $\Delta = \Delta'; (\alpha' : A \cup \delta : B)$ y se tiene $\pi_{\mathtt{LCC}'\langle c[\![\beta \setminus^{\alpha} s]\!][\alpha \setminus \alpha']\rangle} \rhd_{\mathcal{E}} \Gamma \vdash \mathtt{LCC}'\langle c[\![\beta \setminus^{\alpha} s]\!][\alpha \setminus \alpha']\rangle \mid \Delta'; (\alpha' : A \cup (\delta : B)^{\leq 1}); (\gamma : B)^{\leq 1}$. Entonces, por *h.i.* sobre $\mathtt{LCC}'$, se garantiza que existe una derivación $\pi_{\mathtt{LCC}'\langle c[\beta \setminus \alpha]\rangle[\![\alpha \setminus^{\alpha'} s]\!]} \rhd_{\mathcal{E}} \Gamma \vdash \mathtt{LCC}'\langle c[\beta \setminus \alpha]\rangle[\![\alpha \setminus^{\alpha'} s]\!] \mid \Delta; (\alpha' : A \cup (\delta : B)^{\leq 1}); (\gamma : B)^{\leq 1}$. Esta derivación necesariamente termina con una aplicación de de la regla (REPL). Luego, se tienen los contextos $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta' = \Delta_0 \cup \Delta_1$ con derivaciones para la premisa $\pi_{\mathtt{LCC}'\langle c[\beta \setminus \alpha]\rangle} \rhd_{\mathcal{E}} \Gamma_0 \vdash \mathtt{LCC}'\langle c[\beta \setminus \alpha]\rangle \mid \Delta_0; (\alpha : S \to A)^{\leq 1}; ((\alpha' : A)^{\leq 1} \cup (\delta : B)^{\leq 1}); (\gamma : B)^{\leq 1}$ y la premisa $\pi_s \rhd_{\mathcal{E}} \Gamma_1 \vdash s : S \mid \Delta_1; ((\alpha' : A)^{\leq 1} \cup (\delta : B)^{\leq 1}); (\gamma : B)^{\leq 1}$. Entonces, aplicando la regla (REN) con $\pi_{\mathtt{LCC}'\langle c[\beta \setminus \alpha]\rangle}$ y los nombres γ y δ, se obtiene la derivación $\pi_{\mathtt{LCC}\langle c[\beta \setminus \alpha]\rangle} \rhd_{\mathcal{E}} \Gamma_0 \vdash \mathtt{LCC}\langle c[\beta \setminus \alpha]\rangle \mid \Delta_0; (\alpha : S \to A)^{\leq 1}; ((\alpha' : A)^{\leq 1} \cup \delta : B)$. Por último, se concluye por (REPL) con π_s, $\pi_{o'} \rhd_{\mathcal{E}} \Gamma \vdash o' \mid \Delta$. Notar que, sin pérdida de generalidad, puede asumirse $\gamma \notin s$ y, por Lem. 4.3.1, se tiene $(\gamma : S'' \to B)^{\leq 1}$ vacío en π_s.

- $\mathtt{O} = \mathtt{T}\,u$. Luego, $o = t\,u$ y $o' = t'\,u$ con $t' \to_{\mathfrak{C}} t$. Más aún, por (APP) se tienen $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta = \Delta_0 \cup \Delta_1$ con $\pi_t \rhd_{\mathcal{E}} \Gamma_0 \vdash t : A \to T \mid \Delta_0$ y $\pi_u \rhd_{\mathcal{E}} \Gamma_1 \vdash u : A \mid \Delta_1$. Por *h.i.* existe $\pi_{t'} \rhd_{\mathcal{E}} \Gamma_0 \vdash t' : A \to T \mid \Delta_0$. Finalmente, se concluye por (APP), $\pi' \rhd_{\mathcal{E}} \Gamma \vdash o' : T \mid \Delta$.

- $\mathtt{O} = t\,\mathtt{T}$. Luego, $o = t\,u$ y $o' = t\,u'$ con $u' \to_{\mathfrak{C}} u$. Más aún, por (APP) se tienen $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta = \Delta_0 \cup \Delta_1$ con $\pi_t \rhd_{\mathcal{E}} \Gamma_0 \vdash t : A \to T \mid \Delta_0$ y $\pi_u \rhd_{\mathcal{E}} \Gamma_1 \vdash u : A \mid \Delta_1$. Por *h.i.* existe $\pi_{u'} \rhd_{\mathcal{E}} \Gamma_1 \vdash u' : A \mid \Delta_1$. Finalmente, se concluye por (APP), $\pi' \rhd_{\mathcal{E}} \Gamma \vdash o' : T \mid \Delta$.

- $\mathtt{O} = \lambda x.\mathtt{T}$. Luego, $o = \lambda x.t$ y $o' = \lambda x.t'$ con $t' \to_{\mathfrak{C}} t$. Más aún, por (ABS) se tiene $T = A \to B$ con $\pi_t \rhd_{\mathcal{E}} \Gamma; (x : A)^{\leq 1} \vdash t : B \mid \Delta$. Por *h.i.* existe $\pi_{t'} \rhd_{\mathcal{E}} \Gamma; (x : A)^{\leq 1} \vdash t' : B \mid \Delta$. Finalmente, se concluye por (ABS), $\pi' \rhd_{\mathcal{E}} \Gamma \vdash o' : T \mid \Delta$.

- $\mathtt{O} = \mu\alpha.\mathtt{C}$. Luego, $o = \mu\alpha.c$ y $o' = \mu\alpha.c'$ con $c' \to_{\mathfrak{C}} c$. Más aún, por (CONT) se tiene $\pi_c \rhd_{\mathcal{E}} \Gamma \vdash c \mid \Delta; (\alpha : T)^{\leq 1}$. Por *h.i.* existe una derivación $\pi_{c'} \rhd_{\mathcal{E}} \Gamma \vdash c' \mid \Delta; (\alpha : T)^{\leq 1}$. Finalmente, se concluye por (CONT), $\pi' \rhd_{\mathcal{E}} \Gamma \vdash o' : T \mid \Delta$.

- $\mathtt{O} = \mathtt{T}[x \setminus u]$. Luego, $o = t[x \setminus u]$ y $o' = t'[x \setminus u]$ con $t' \to_{\mathfrak{C}} t$. Más aún, por (SUBS), $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta = \Delta_0 \cup \Delta_1$ con $\pi_t \rhd_{\mathcal{E}} \Gamma_0; (x : A)^{\leq 1} \vdash t : T \mid \Delta_0$ y $\pi_u \rhd_{\mathcal{E}} \Gamma_1 \vdash u : A \mid \Delta_1$. Por *h.i.* existe $\pi_{t'} \rhd_{\mathcal{E}} \Gamma_0; (x : A)^{\leq 1} \vdash t' : T \mid \Delta_0$. Finalmente, se concluye por (SUBS), $\pi' \rhd_{\mathcal{E}} \Gamma \vdash o' : T \mid \Delta$.

- $\mathtt{O} = t[x \setminus \mathtt{T}]$. Luego, $o = t[x \setminus u]$ y $o' = t[x \setminus u']$ con $u' \to_{\mathfrak{C}} u$. Más aún, por (SUBS) se tienen $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta = \Delta_0 \cup \Delta_1$ con $\pi_t \rhd_{\mathcal{E}} \Gamma_0; (x : A)^{\leq 1} \vdash t : T \mid \Delta_0$ y $\pi_u \rhd_{\mathcal{E}} \Gamma_1 \vdash u : A \mid \Delta_1$. Por *h.i.* existe $\pi_{u'} \rhd_{\mathcal{E}} \Gamma_1 \vdash u' : A \mid \Delta_1$. Finalmente, se concluye por (SUBS), $\pi' \rhd_{\mathcal{E}} \Gamma \vdash o' : T \mid \Delta$.

- $\mathtt{O} = [\alpha]\,\mathtt{T}$. Luego, $o = [\alpha]\,t$ y $o' = [\alpha]\,t'$ con $t' \to_{\mathfrak{C}} t$. Más aún, por (NAME) se tiene $\Delta = \Delta_0; \alpha : A$ con $\pi_t \rhd_{\mathcal{E}} \Gamma \vdash t : A \mid \Delta_0; (\alpha : A)^{\leq 1}$. Por *h.i.* existe $\pi_{t'} \rhd_{\mathcal{E}} \Gamma \vdash t' : A \mid \Delta_0; (\alpha : A)^{\leq 1}$. Se concluye por (NAME), $\pi' \rhd_{\mathcal{E}} \Gamma \vdash o' : T \mid \Delta_0; \alpha : A$.

- $\mathtt{O} = \mathtt{C}[\![\alpha \setminus^{\alpha'} s]\!]$. Luego, $o = c[\![\alpha \setminus^{\alpha'} s]\!]$ y $o' = c'[\![\alpha \setminus^{\alpha'} s]\!]$ con $c' \to_{\mathfrak{C}} c$. Más aún, por (REPL) se tienen $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta = \Delta_0 \cup \Delta_1; \alpha' : A$ con $\pi_c \rhd_{\mathcal{E}} \Gamma_0 \vdash c \mid \Delta_0; (\alpha : S \to A)^{\leq 1}; (\alpha' : A)^{\leq 1}$ y $\pi_s \rhd_{\mathcal{E}} \Gamma_1 \vdash s : A \mid \Delta_1; (\alpha' : A)^{\leq 1}$. Por *h.i.* existe $\pi_{c'} \rhd_{\mathcal{E}} \Gamma_0 \vdash c' \mid \Delta_0; (\alpha : S \to A)^{\leq 1}; (\alpha' : A)^{\leq 1}$. Finalmente, se concluye por (REPL), $\pi' \rhd_{\mathcal{E}} \Gamma \vdash o' : T \mid \Delta$.

- $\mathtt{O} = c[\![\alpha \setminus^{\alpha'} \mathtt{S}]\!]$. Luego, $o = c[\![\alpha \setminus^{\alpha'} s]\!]$ y $o' = c[\![\alpha \setminus^{\alpha'} s']\!]$ con $s' \to_{\mathfrak{C}} s$. Más aún, por (REPL) se tienen $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta = \Delta_0 \cup \Delta_1; \alpha' : A$ con $\pi_c \rhd_{\mathcal{E}} \Gamma_0 \vdash c \mid \Delta_0; (\alpha : S \to A)^{\leq 1}; (\alpha' : A)^{\leq 1}$ y $\pi_s \rhd_{\mathcal{E}} \Gamma_1 \vdash s : A \mid \Delta_1; (\alpha' : A)^{\leq 1}$. Por *h.i.* existe una derivación $\pi_{s'} \rhd_{\mathcal{E}} \Gamma_1 \vdash s' : A \mid \Delta_1; (\alpha' : A)^{\leq 1}$. Finalmente, se concluye por (REPL), $\pi' \rhd_{\mathcal{E}} \Gamma \vdash o' : T \mid \Delta$.

- $\mathtt{O} = \mathtt{C}[\alpha \setminus \beta]$. Luego, $o = c[\alpha \setminus \beta]$ y $o' = c'[\alpha \setminus \beta]$ con $c' \to_{\mathfrak{C}} c$. Más aún, por (REN) se tiene $\Delta = \Delta_0; \beta : A$ con $\pi_c \rhd_{\mathcal{E}} \Gamma \vdash c \mid \Delta_0; (\alpha : A)^{\leq 1}; (\beta : A)^{\leq 1}$. Por *h.i.* existe una derivación $\pi_{c'} \rhd_{\mathcal{E}} \Gamma \vdash c' \mid \Delta_0; (\alpha : A)^{\leq 1}; (\beta : A)^{\leq 1}$. Finalmente, se concluye por (REN), $\pi' \rhd_{\mathcal{E}} \Gamma \vdash o' : T \mid \Delta$.

- $\mathtt{O} = \mathtt{T} \cdot s$. Luego, $o = t \cdot s$ y $o' = t' \cdot s$ con $t' \to_{\mathfrak{C}} t$. Más aún, por (STK) se tienen $T = A \cdot S$, $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta = \Delta_0 \cup \Delta_1$ con $\pi_t \rhd_{\mathcal{E}} \Gamma_0 \vdash t : A \mid \Delta_0$ y $\pi_s \rhd_{\mathcal{E}} \Gamma_1 \vdash s : S \mid \Delta_1$. Por *h.i.* existe $\pi_{t'} \rhd_{\mathcal{E}} \Gamma_0 \vdash t' : A \mid \Delta_0$. Finalmente, se concluye por (STK), $\pi' \rhd_{\mathcal{E}} \Gamma \vdash o' : T \mid \Delta$.

- $\mathtt{O} = t \cdot \mathtt{S}$. Luego, $o = t \cdot s$ y $o' = t \cdot s'$ con $s' \to_{\mathfrak{C}} s$. Más aún, por (STK) se tienen $T = A \cdot S$, $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta = \Delta_0 \cup \Delta_1$ con $\pi_t \rhd_{\mathcal{E}} \Gamma_0 \vdash t : A \mid \Delta_0$ y $\pi_s \rhd_{\mathcal{E}} \Gamma_1 \vdash s : S \mid \Delta_1$. Por *h.i.* existe $\pi_{s'} \rhd_{\mathcal{E}} \Gamma_1 \vdash s' : S \mid \Delta_1$. Finalmente, se concluye por (STK), $\pi' \rhd_{\mathcal{E}} \Gamma \vdash o' : T \mid \Delta$.

□

Equivalencia estructural para ΛM-cálculo

Para probar el Lem 4.6.2 se introducen dos resultados auxiliares sobre contextos LTT y LCC respectivamente.

Lema C.0.45. *Sea* $t \in \mathbb{T}_{\Lambda M}$ *y* $\mathcal{S} \triangleq \{\langle \mathtt{l}\langle \mathtt{LTT}\langle t\rangle\rangle, \mathtt{LTT}\langle \mathtt{l}\langle t\rangle\rangle\rangle \mid \mathsf{bv}(\mathtt{l}) \notin \mathtt{LTT}, \mathsf{fc}(\mathtt{l}, \mathtt{LTT})\}$. *Luego,* $\mathcal{S}$ *es una bisimulación fuerte respecto a la relación de reducción canónica.*

Demostración. Se prueba que

$$\begin{array}{ccc} \mathtt{L}\langle \mathtt{LTT}\langle t\rangle\rangle & \mathcal{S} & \mathtt{LTT}\langle \mathtt{L}\langle t\rangle\rangle \\ \mathfrak{e}\downarrow & & \mathfrak{e}\downarrow \\ \mathtt{L}'\langle \mathtt{LTT}'\langle t'\rangle\rangle & \mathcal{S} & \mathtt{LTT}'\langle \mathtt{L}'\langle t'\rangle\rangle \end{array}$$

para algún $\mathtt{L}'$, $\mathtt{LTT}'$ y t'. El análisis para el caso simétrico

$$\begin{array}{ccc} \mathtt{L}\langle \mathtt{LTT}\langle t\rangle\rangle & \mathcal{S} & \mathtt{LTT}\langle \mathtt{L}\langle t\rangle\rangle \\ \mathfrak{e}\downarrow & & \mathfrak{e}\downarrow \\ \mathtt{L}'\langle \mathtt{LTT}'\langle t'\rangle\rangle & \mathcal{S} & \mathtt{LTT}'\langle \mathtt{L}'\langle t'\rangle\rangle \end{array}$$

es similar y se omite. Los posibles solapamientos entre pasos $\mathfrak{C}$ y $\mathtt{L}\langle \mathtt{LTT}\langle t\rangle\rangle$ llevan a analizar los siguientes casos:

- La reducción ocurre completamente dentro de t, *i.e.* $t \to_{\mathfrak{C}} t'$. Luego, basta tomar $\mathtt{L}' = \mathtt{L}$ y $\mathtt{LTT}' = \mathtt{LTT}$.
- La reducción ocurre completamente dentro de $\mathtt{L}$, *i.e.* $\mathtt{L} \to_{\mathfrak{C}} \mathtt{L}'$. Luego, basta tomar $t' = t$ y $\mathtt{LTT}' = \mathtt{LTT}$.
- La reducción ocurre completamente dentro de $\mathtt{LTT}$, *i.e.* $\mathtt{LTT} \to_{\mathfrak{C}} \mathtt{LTT}'$. Luego, basta tomar $t' = t$ y $\mathtt{L}' = \mathtt{L}$.
- La reducción se solapa con $\mathtt{LTT}$. Hay cuatro casos posibles según la regla de reducción aplicada:

1. $\mathtt{dB}$. Luego, hay dos posibilidades.
   a) Se superpone con t. Luego, $t = \mathtt{L}_1\langle \lambda x.v\rangle$, $\mathtt{LTT} = \mathtt{LTT}_2\langle \mathtt{L}_2\, u\rangle$ y el lado izquierdo de la regla $\mathtt{dB}$ es $\mathtt{L}_2\langle \mathtt{L}_1\langle \lambda x.v\rangle\rangle\, u$. Se concluye con $t' = \mathtt{L}_1\langle v[x \setminus u]\rangle$, $\mathtt{L}' = \mathtt{L}$ y $\mathtt{LTT}' = \mathtt{LTT}_2\langle \mathtt{L}_2\rangle$:
   $$\begin{array}{ccc} \mathtt{L}\langle \mathtt{LTT}\langle t\rangle\rangle & \mathcal{S} & \mathtt{LTT}_2\langle \mathtt{L}_2\langle \mathtt{L}\langle \mathtt{L}_1\langle \lambda x.v\rangle\rangle\rangle\, u\rangle \\ \mathtt{dB}\downarrow & & \mathtt{dB}\vdots \\ \mathtt{L}'\langle \mathtt{LTT}'\langle t'\rangle\rangle & \mathcal{S} & \mathtt{LTT}_2\langle \mathtt{L}_2\langle \mathtt{L}'\langle \mathtt{L}_1\langle v[x \setminus u]\rangle\rangle\rangle\rangle \end{array}$$
   b) No se superpone con t. Luego, $\mathtt{LTT} = \mathtt{LTT}_2\langle \mathtt{L}_2\langle \lambda x.\mathtt{LTT}_1\rangle\, u\rangle$. Se concluye con $t' = t$, $\mathtt{L}' = \mathtt{L}$ y $\mathtt{LTT}' = \mathtt{LTT}_2\langle \mathtt{L}_2\langle \mathtt{LTT}_1[x \setminus u]\rangle\rangle$:
   $$\begin{array}{ccc} \mathtt{L}\langle \mathtt{LTT}\langle t\rangle\rangle & \mathcal{S} & \mathtt{LTT}_2\langle \mathtt{L}_2\langle \lambda x.\mathtt{LTT}_1\langle \mathtt{L}\langle t\rangle\rangle\rangle\, u\rangle \\ \mathtt{dB}\downarrow & & \mathtt{dB}\vdots \\ \mathtt{L}'\langle \mathtt{LTT}'\langle t'\rangle\rangle & \mathcal{S} & \mathtt{LTT}_2\langle \mathtt{L}_2\langle \mathtt{LTT}_1\langle \mathtt{L}'\langle t'\rangle\rangle[x \setminus u]\rangle\rangle \end{array}$$
2. $\mathtt{dM}$. Luego, hay dos posibilidades.
   a) Se superpone con t. Luego, $t = \mathtt{L}_1\langle \mu\alpha.c\rangle$, $\mathtt{LTT} = \mathtt{LTT}_2\langle \mathtt{L}_2\, u\rangle$ y el lado izquierdo de la regla $\mathtt{dM}$ es $\mathtt{L}_2\langle \mathtt{L}_1\langle \mu\alpha.c\rangle\rangle\, u$. Se concluye con $t' = \mathtt{L}_1\langle \mu\alpha'.c[\![\alpha \setminus^{\alpha'} u]\!]\rangle$, $\mathtt{L}' = \mathtt{L}$ y $\mathtt{LTT}' = \mathtt{LTT}_2\langle \mathtt{L}_2\rangle$:
   $$\begin{array}{ccc} \mathtt{L}\langle \mathtt{LTT}\langle t\rangle\rangle & \mathcal{S} & \mathtt{LTT}_2\langle \mathtt{L}_2\langle \mathtt{L}\langle \mathtt{L}_1\langle \mu\alpha.c\rangle\rangle\rangle\, u\rangle \\ \mathtt{dM}\downarrow & & \mathtt{dM}\vdots \\ \mathtt{L}'\langle \mathtt{LTT}'\langle t'\rangle\rangle & \mathcal{S} & \mathtt{LTT}_2\langle \mathtt{L}_2\langle \mathtt{L}'\langle \mathtt{L}_1\langle \mu\alpha'.c[\![\alpha \setminus^{\alpha'} u]\!]\rangle\rangle\rangle\rangle \end{array}$$
   b) No se superpone con t. Luego, $\mathtt{LTT} = \mathtt{LTT}_2\langle \mathtt{L}_2\langle \mu\alpha.\mathtt{LCT}\rangle\, u\rangle$. Se concluye con $t' = t$, $\mathtt{L}' = \mathtt{L}$ y $\mathtt{LTT}' = \mathtt{LTT}_2\langle \mathtt{L}_2\langle \mu\alpha'.\mathtt{LCT}[\![\alpha \setminus^{\alpha'} u]\!]\rangle\rangle$:
   $$\begin{array}{ccc} \mathtt{L}\langle \mathtt{LTT}\langle t\rangle\rangle & \mathcal{S} & \mathtt{LTT}_2\langle \mathtt{L}_2\langle \mu\alpha.\mathtt{LCT}\langle \mathtt{L}\langle t\rangle\rangle\rangle\, u\rangle \\ \mathtt{dM}\downarrow & & \mathtt{dM}\vdots \\ \mathtt{L}'\langle \mathtt{LTT}'\langle t'\rangle\rangle & \mathcal{S} & \mathtt{LTT}_2\langle \mathtt{L}_2\langle \mu\alpha'.\mathtt{LCT}\langle \mathtt{L}'\langle t'\rangle\rangle[\![\alpha \setminus^{\alpha'} u]\!]\rangle\rangle \end{array}$$
3. $\mathtt{N}$. Luego, hay dos posibilidades.
   a) Se superpone con t. Luego, $t = \mathtt{LTT}_1\langle \mu\gamma.\mathtt{LCC}_1\langle [\alpha]\, u\rangle\rangle$, $\mathtt{LTT} = \mathtt{LTT}_2\langle \mu\delta.\mathtt{LCC}_2\langle \mathtt{LCT}[\![\alpha \setminus^{\alpha'} s]\!]\rangle\rangle$ y el lado izquierdo de la regla $\mathtt{N}$ es $\mathtt{LCT}\langle \mathtt{LTT}_1\langle \mu\gamma.\mathtt{LCC}_1\langle [\alpha]\, u\rangle\rangle[\![\alpha \setminus^{\alpha'} s]\!]$. Se concluye tomando $t' = \mathtt{LTT}_1\langle \mu\gamma.\mathtt{LCC}_1\langle [\alpha']\, u :: s\rangle\rangle$, $\mathtt{L}' = \mathtt{L}$ y $\mathtt{LTT}' = \mathtt{LTT}_2\langle \mu\delta.\mathtt{LCC}_2\langle \mathtt{LCT}\rangle\rangle$:
   $$\begin{array}{ccc} \mathtt{L}\langle \mathtt{LTT}\langle t\rangle\rangle & \mathcal{S} & \mathtt{LTT}_2\langle \mu\delta.\mathtt{LCC}_2\langle \mathtt{LCT}\langle \mathtt{L}\langle \mathtt{LTT}_1\langle \mu\gamma.\mathtt{LCC}_1\langle [\alpha]\, u\rangle\rangle\rangle\rangle[\![\alpha \setminus^{\alpha'} s]\!]\rangle\rangle \\ \mathtt{N}\downarrow & & \mathtt{N}\vdots \\ \mathtt{L}'\langle \mathtt{LTT}'\langle t'\rangle\rangle & \mathcal{S} & \mathtt{LTT}_2\langle \mu\delta.\mathtt{LCC}_2\langle \mathtt{LCT}\langle \mathtt{L}'\langle \mathtt{LTT}_1\langle \mu\gamma.\mathtt{LCC}_1\langle [\alpha']\, u :: s\rangle\rangle\rangle\rangle\rangle\rangle \end{array}$$

b) No se superpone con t. Luego, $\mathtt{LTT} = \mathtt{LTT}_2\langle\mu\gamma.\mathtt{LCC}_2\langle\mathtt{LCC}\langle[\alpha]\,\mathtt{LCT}\rangle[\![\alpha \setminus^{\alpha'} s]\!]\rangle\rangle$. Se concluye con $t' = t$, $\mathtt{L}' = \mathtt{L}$ y $\mathtt{LTT}' = \mathtt{LTT}_2\langle\mu\gamma.\mathtt{LCC}_2\langle\mathtt{LCC}\langle[\alpha']\,\mathtt{LTT}_1 :: s\rangle\rangle\rangle$:

$$\begin{array}{ccc} \mathtt{L}\langle\mathtt{LTT}\langle t\rangle\rangle & \mathcal{S} & \mathtt{LTT}_2\langle\mu\gamma.\mathtt{LCC}_2\langle\mathtt{LCC}\langle[\alpha]\,\mathtt{LCT}\langle\mathtt{L}\langle t\rangle\rangle\rangle[\![\alpha \setminus^{\alpha'} s]\!]\rangle\rangle \\ \downarrow{\scriptstyle \mathtt{N}} & & \vdots{\scriptstyle \mathtt{N}} \\ \mathtt{L}'\langle\mathtt{LTT}'\langle t'\rangle\rangle & \mathcal{S} & \mathtt{LTT}_2\langle\mu\gamma.\mathtt{LCC}_2\langle\mathtt{LCC}\langle[\alpha']\,\mathtt{LTT}_1\langle\mathtt{L}'\langle t'\rangle\rangle :: s\rangle\rangle\rangle \end{array}$$

4. $\mathtt{P}$. Luego, hay dos posibilidades.

a) Se superpone con t. Luego, $t = \mathtt{LTT}_1\langle\mu\gamma.\mathtt{LCC}_1\langle c[\![\beta \setminus^{\alpha} s']\!]\rangle\rangle$, $\mathtt{LTT} = \mathtt{LTT}_2\langle\mu\delta.\mathtt{LCC}_2\langle\mathtt{LCT}[\![\alpha \setminus^{\alpha'} s]\!]\rangle\rangle$ y el lado izquierdo de la regla $\mathtt{P}$ es $\mathtt{LCT}\langle\mathtt{LTT}_1\langle\mu\gamma.\mathtt{LCC}_1\langle c[\![\beta \setminus^{\alpha} s']\!]\rangle\rangle\rangle[\![\alpha \setminus^{\alpha'} s]\!]$. Se concluye tomando $t' = \mathtt{LTT}_1\langle\mu\gamma.\mathtt{LCC}_1\langle c[\![\beta \setminus^{\alpha'} s' \cdot s]\!]\rangle\rangle$, $\mathtt{L}' = \mathtt{L}$ y $\mathtt{LTT}' = \mathtt{LTT}_2\langle\mu\delta.\mathtt{LCC}_2\langle\mathtt{LCT}\rangle\rangle$:

$$\begin{array}{ccc} \mathtt{L}\langle\mathtt{LTT}\langle t\rangle\rangle & \mathcal{S} & \mathtt{LTT}_2\langle\mu\delta.\mathtt{LCC}_2\langle\mathtt{LCT}\langle\mathtt{L}\langle\mathtt{LTT}_1\langle\mu\gamma.\mathtt{LCC}_1\langle c[\![\beta \setminus^{\alpha} s']\!]\rangle\rangle\rangle\rangle[\![\alpha \setminus^{\alpha'} s]\!]\rangle\rangle \\ \downarrow{\scriptstyle \mathtt{P}} & & \vdots{\scriptstyle \mathtt{P}} \\ \mathtt{L}'\langle\mathtt{LTT}'\langle t'\rangle\rangle & \mathcal{S} & \mathtt{LTT}_2\langle\mu\delta.\mathtt{LCC}_2\langle\mathtt{LCT}\langle\mathtt{L}'\langle\mathtt{LTT}_1\langle\mu\gamma.\mathtt{LCC}_1\langle c[\![\beta \setminus^{\alpha'} s' \cdot s]\!]\rangle\rangle\rangle\rangle\rangle\rangle \end{array}$$

b) No se superpone con t. Luego, $\mathtt{LTT} = \mathtt{LTT}_2\langle\mu\gamma.\mathtt{LCC}_2\langle\mathtt{LCC}\langle\mathtt{LCT}[\![\beta \setminus^{\alpha} s']\!]\rangle[\![\alpha \setminus^{\alpha'} s]\!]\rangle\rangle$. Se concluye con $t' = t$, $\mathtt{L}' = \mathtt{L}$ y $\mathtt{LTT}' = \mathtt{LTT}_2\langle\mu\gamma.\mathtt{LCC}_2\langle\mathtt{LCC}\langle\mathtt{LCT}[\![\beta \setminus^{\alpha'} s' \cdot s]\!]\rangle\rangle\rangle$:

$$\begin{array}{ccc} \mathtt{L}\langle\mathtt{LTT}\langle t\rangle\rangle & \mathcal{S} & \mathtt{LTT}_2\langle\mu\gamma.\mathtt{LCC}_2\langle\mathtt{LCC}\langle\mathtt{LCT}\langle\mathtt{L}\langle t\rangle\rangle[\![\beta \setminus^{\alpha} s']\!]\rangle[\![\alpha \setminus^{\alpha'} s]\!]\rangle\rangle \\ \downarrow{\scriptstyle \mathtt{P}} & & \vdots{\scriptstyle \mathtt{P}} \\ \mathtt{L}'\langle\mathtt{LTT}'\langle t'\rangle\rangle & \mathcal{S} & \mathtt{LTT}_2\langle\mu\gamma.\mathtt{LCC}_2\langle\mathtt{LCC}\langle\mathtt{LCT}\langle\mathtt{L}'\langle t'\rangle\rangle[\![\beta \setminus^{\alpha'} s' \cdot s]\!]\rangle\rangle\rangle \end{array}$$

5. $\mathtt{W}$. Luego, hay dos posibilidades.

a) Se superpone con t. Luego, $t = \mathtt{LTT}_1\langle\mu\gamma.\mathtt{LCC}_1\langle c[\beta \setminus \alpha]\rangle\rangle$, $\mathtt{LTT} = \mathtt{LTT}_2\langle\mu\delta.\mathtt{LCC}_2\langle\mathtt{LCT}[\![\alpha \setminus^{\alpha'} s]\!]\rangle\rangle$ y el lado izquierdo de la regla $\mathtt{W}$ es $\mathtt{LCT}\langle\mathtt{LTT}_1\langle\mu\gamma.\mathtt{LCC}_1\langle c[\beta \setminus \alpha]\rangle\rangle\rangle[\![\alpha \setminus^{\alpha'} s]\!]$. Se concluye tomando $t' = \mathtt{LTT}_1\langle\mu\gamma.\mathtt{LCC}_1\langle c[\![\beta \setminus^{\alpha} s][\alpha \setminus \alpha']\!]\rangle\rangle$, $\mathtt{L}' = \mathtt{L}$ y $\mathtt{LTT}' = \mathtt{LTT}_2\langle\mu\delta.\mathtt{LCC}_2\langle\mathtt{LCT}\rangle\rangle$:

$$\begin{array}{ccc} \mathtt{L}\langle\mathtt{LTT}\langle t\rangle\rangle & \mathcal{S} & \mathtt{LTT}_2\langle\mu\delta.\mathtt{LCC}_2\langle\mathtt{LCT}\langle\mathtt{L}\langle\mathtt{LTT}_1\langle\mu\gamma.\mathtt{LCC}_1\langle c[\beta \setminus \alpha]\rangle\rangle\rangle\rangle[\![\alpha \setminus^{\alpha'} s]\!]\rangle\rangle \\ \downarrow{\scriptstyle \mathtt{W}} & & \vdots{\scriptstyle \mathtt{W}} \\ \mathtt{L}'\langle\mathtt{LTT}'\langle t'\rangle\rangle & \mathcal{S} & \mathtt{LTT}_2\langle\mu\delta.\mathtt{LCC}_2\langle\mathtt{LCT}\langle\mathtt{L}'\langle\mathtt{LTT}_1\langle\mu\gamma.\mathtt{LCC}_1\langle c[\![\beta \setminus^{\alpha} s][\alpha \setminus \alpha']\!]\rangle\rangle\rangle\rangle\rangle\rangle \end{array}$$

b) No se superpone con t. Luego, $\mathtt{LTT} = \mathtt{LTT}_2\langle\mu\gamma.\mathtt{LCC}_2\langle\mathtt{LCC}\langle\mathtt{LCT}[\beta \setminus \alpha]\rangle[\![\alpha \setminus^{\alpha'} s]\!]\rangle\rangle$. Se concluye con $t' = t$, $\mathtt{L}' = \mathtt{L}$ y $\mathtt{LTT}' = \mathtt{LTT}_2\langle\mu\gamma.\mathtt{LCC}_2\langle\mathtt{LCC}\langle\mathtt{LCT}[\![\beta \setminus^{\alpha} s][\alpha \setminus \alpha']\!]\rangle\rangle\rangle$:

$$\begin{array}{ccc} \mathtt{L}\langle\mathtt{LTT}\langle t\rangle\rangle & \mathcal{S} & \mathtt{LTT}_2\langle\mu\gamma.\mathtt{LCC}_2\langle\mathtt{LCC}\langle\mathtt{LCT}\langle\mathtt{L}\langle t\rangle\rangle[\beta \setminus \alpha]\rangle[\![\alpha \setminus^{\alpha'} s]\!]\rangle\rangle \\ \downarrow{\scriptstyle \mathtt{W}} & & \vdots{\scriptstyle \mathtt{W}} \\ \mathtt{L}'\langle\mathtt{LTT}'\langle t'\rangle\rangle & \mathcal{S} & \mathtt{LTT}_2\langle\mu\gamma.\mathtt{LCC}_2\langle\mathtt{LCC}\langle\mathtt{LCT}\langle\mathtt{L}'\langle t'\rangle\rangle[\![\beta \setminus^{\alpha} s][\alpha \setminus \alpha']\!]\rangle\rangle\rangle \end{array}$$

- No hay más casos posibles dadas las hipótesis de la relación $\mathcal{S}$.

□

Lema C.0.46. *Sea $c \in \mathbb{C}_{\Lambda M}$ y $\mathcal{R} \triangleq \{\langle \mathtt{R}\langle\mathtt{LCC}\langle c\rangle\rangle, \mathtt{LCC}\langle\mathtt{R}\langle c\rangle\rangle\rangle \mid \mathsf{bn}(\mathtt{R}) \notin \mathtt{LCC}, \mathsf{fc}(\mathtt{R}, \mathtt{LCC})\}$. Luego, $\mathcal{R}$ es una bisimulación fuerte respecto a la relación de reducción canónica.*

Demostración. Se prueba que

$$\begin{array}{ccc} \mathtt{R}\langle \mathtt{LCC}\langle c\rangle\rangle & \mathcal{R} & \mathtt{LCC}\langle \mathtt{R}\langle c\rangle\rangle \\ \mathfrak{C}\downarrow & & \mathfrak{C}\vdots \\ \mathtt{R}'\langle \mathtt{LCC}'\langle c'\rangle\rangle & \mathcal{R} & \mathtt{LCC}'\langle \mathtt{R}'\langle c'\rangle\rangle \end{array}$$

para algún $\mathtt{R}'$, $\mathtt{LCC}'$ y c'. El análisis para el caso simétrico

$$\begin{array}{ccc} \mathtt{R}\langle \mathtt{LCC}\langle c\rangle\rangle & \mathcal{R} & \mathtt{LCC}\langle \mathtt{R}\langle c\rangle\rangle \\ \mathfrak{C}\vdots & & \mathfrak{C}\downarrow \\ \mathtt{R}'\langle \mathtt{LCC}'\langle c'\rangle\rangle & \mathcal{R} & \mathtt{LCC}'\langle \mathtt{R}'\langle c'\rangle\rangle \end{array}$$

es similar y se omite. Los posibles solapamientos entre pasos $\mathfrak{C}$ y $\mathtt{R}\langle \mathtt{LCC}\langle c\rangle\rangle$ llevan a analizar los siguientes casos:

- La reducción ocurre completamente dentro de c, *i.e.* $c \to_{\mathfrak{C}} c'$. Luego, basta tomar $\mathtt{R}' = \mathtt{R}$ y $\mathtt{LCC}' = \mathtt{LCC}$.
- La reducción ocurre completamente dentro de $\mathtt{R}$, *i.e.* $\mathtt{R} \to_{\mathfrak{C}} \mathtt{R}'$. Luego, basta tomar $c' = c$ y $\mathtt{LCC}' = \mathtt{LCC}$.
- La reducción ocurre completamente dentro de $\mathtt{LCC}$, *i.e.* $\mathtt{LCC} \to_{\mathfrak{C}} \mathtt{LCC}'$. Luego, basta tomar $c' = c$ y $\mathtt{R}' = \mathtt{R}$.
- La reducción se solapa con $\mathtt{LCC}$. Hay cuatro casos posibles según la regla de reducción aplicada:
  1. $\mathtt{dB}$. Luego, $\mathtt{LCC} = \mathtt{LCC}_1\langle[\gamma]\,\mathtt{LTC}_1\langle \mathtt{L}\langle\lambda x.\mathtt{LTC}_2\rangle\, u\rangle\rangle$. Se concluye con $c' = c$, $\mathtt{R}' = \mathtt{R}$ y $\mathtt{LCC}' = \mathtt{LCC}_1\langle[\gamma]\,\mathtt{LTC}_1\langle \mathtt{L}\langle\mathtt{LTC}_2[x \setminus u]\rangle\rangle\rangle$:
  $$\begin{array}{ccc} \mathtt{R}\langle \mathtt{LCC}\langle c\rangle\rangle & \mathcal{R} & \mathtt{LCC}_1\langle[\gamma]\,\mathtt{LTC}_1\langle \mathtt{L}\langle\lambda x.\mathtt{LTC}_2\langle \mathtt{R}\langle c\rangle\rangle\rangle\, u\rangle\rangle \\ \mathtt{dB}\downarrow & & \mathtt{dB}\vdots \\ \mathtt{R}'\langle \mathtt{LCC}'\langle c'\rangle\rangle & \mathcal{R} & \mathtt{LCC}_1\langle[\gamma]\,\mathtt{LTC}_1\langle \mathtt{L}\langle\mathtt{LTC}_2\langle \mathtt{R}'\langle c'\rangle\rangle[x \setminus u]\rangle\rangle\rangle \end{array}$$
  2. $\mathtt{dM}$. Luego, $\mathtt{LCC} = \mathtt{LCC}_1\langle[\gamma]\,\mathtt{LTC}_1\langle \mathtt{L}\langle\mu\alpha.\mathtt{LTC}_2\rangle\, u\rangle\rangle$. Se concluye con $c' = c$, $\mathtt{R}' = \mathtt{R}$ y $\mathtt{LCC}' = \mathtt{LCC}_1\langle[\gamma]\,\mathtt{LTC}_1\langle \mathtt{L}\langle\mu\alpha'.\mathtt{LTC}_2[\![\alpha \setminus^{\alpha'} u]\!]\rangle\rangle\rangle$:
  $$\begin{array}{ccc} \mathtt{R}\langle \mathtt{LCC}\langle c\rangle\rangle & \mathcal{R} & \mathtt{LCC}_1\langle[\gamma]\,\mathtt{LTC}_1\langle \mathtt{L}\langle\mu\alpha.\mathtt{LTC}_2\langle \mathtt{R}\langle c\rangle\rangle\rangle\, u\rangle\rangle \\ \mathtt{dM}\downarrow & & \mathtt{dM}\vdots \\ \mathtt{R}'\langle \mathtt{LCC}'\langle c'\rangle\rangle & \mathcal{R} & \mathtt{LCC}_1\langle[\gamma]\,\mathtt{LTC}_1\langle \mathtt{L}\langle\mu\alpha'.\mathtt{LTC}_2\langle \mathtt{R}'\langle c'\rangle\rangle[\![\alpha \setminus^{\alpha'} u]\!]\rangle\rangle\rangle \end{array}$$
  3. $\mathtt{N}$. Luego, hay dos posibilidades.
     a) Se superpone con c. Luego, se tienen $c = \mathtt{LCC}_1\langle[\alpha]\, t\rangle$, $\mathtt{LCC} = \mathtt{LCC}_3\langle\mathtt{LCC}_2[\![\alpha \setminus^{\alpha'} s]\!]\rangle$ y la regla $\mathtt{N}$ aplica sobre $\mathtt{LCC}_2\langle\mathtt{LCC}_1\langle[\alpha]\, t\rangle\rangle[\![\alpha \setminus^{\alpha'} s]\!]$. Se concluye tomando $c' = \mathtt{LCC}_1\langle[\alpha']\, t :: sc_1\rangle$, $\mathtt{R}' = \mathtt{R}$ y $\mathtt{LCC}' = \mathtt{LCC}_3\langle\mathtt{LCC}_2\rangle$:
     $$\begin{array}{ccc} \mathtt{R}\langle \mathtt{LCC}\langle c\rangle\rangle & \mathcal{R} & \mathtt{LCC}_3\langle\mathtt{LCC}_2\langle \mathtt{R}\langle\mathtt{LCC}_1\langle[\alpha]\, t\rangle\rangle\rangle[\![\alpha \setminus^{\alpha'} s]\!]\rangle \\ \mathtt{N}\downarrow & & \mathtt{N}\vdots \\ \mathtt{R}'\langle \mathtt{LCC}'\langle c'\rangle\rangle & \mathcal{R} & \mathtt{LCC}_3\langle\mathtt{LCC}_2\langle \mathtt{R}'\langle\mathtt{LCC}_1\langle[\alpha']\, t :: s\rangle\rangle\rangle\rangle \end{array}$$
     b) No se superpone con c. Luego, $\mathtt{LCC} = \mathtt{LCC}_3\langle\mathtt{LCC}_2\langle[\alpha]\,\mathtt{LTC}\rangle[\![\alpha \setminus^{\alpha'} s]\!]\rangle$. Se concluye con $c' = c$, $\mathtt{R}' = \mathtt{R}$ y $\mathtt{LCC}' = \mathtt{LCC}_3\langle\mathtt{LCC}_2\langle[\alpha']\,\mathtt{LTC} :: s\rangle\rangle$:

$$\begin{array}{ccc} \mathtt{R}\langle \mathtt{LCC}\langle c\rangle\rangle & \mathcal{R} & \mathtt{LCC}_3\langle \mathtt{LCC}_2\langle [\alpha]\,\mathtt{LTC}\langle \mathtt{R}\langle c\rangle\rangle\rangle [\![\alpha \setminus^{\alpha'} s]\!]\rangle \\ \downarrow{\scriptstyle \mathtt{N}} & & \vdots{\scriptstyle \mathtt{N}} \\ \mathtt{R}'\langle \mathtt{LCC}'\langle c'\rangle\rangle & \mathcal{R} & \mathtt{LCC}_3\langle \mathtt{LCC}_2\langle [\alpha']\,\mathtt{LTC}\langle \mathtt{R}'\langle c'\rangle\rangle :: s\rangle\rangle \end{array}$$

4. P. Luego, hay dos posibilidades.

   a) Se superpone con c. Luego, se tienen $c = \mathtt{LCC}_1\langle c_1 [\![\beta \setminus^{\alpha} s']\!]\rangle$, $\mathtt{LCC} = \mathtt{LCC}_3\langle \mathtt{LCC}_2 [\![\alpha \setminus^{\alpha'} s]\!]\rangle$ y la regla P aplica sobre $\mathtt{LCC}_2\langle \mathtt{LCC}_1\langle c_1 [\![\beta \setminus^{\alpha} s']\!]\rangle\rangle [\![\alpha \setminus^{\alpha'} s]\!]$. Se concluye tomando $c' = \mathtt{LCC}_1\langle c_1 [\![\beta \setminus^{\alpha'} s' \cdot s]\!]\rangle$, $\mathtt{R}' = \mathtt{R}$ y $\mathtt{LCC}' = \mathtt{LCC}_3\langle \mathtt{LCC}_2\rangle$:

$$\begin{array}{ccc} \mathtt{R}\langle \mathtt{LCC}\langle c\rangle\rangle & \mathcal{R} & \mathtt{LCC}_3\langle \mathtt{LCC}_2\langle \mathtt{R}\langle \mathtt{LCC}_1\langle c_1 [\![\beta \setminus^{\alpha} s']\!]\rangle\rangle\rangle [\![\alpha \setminus^{\alpha'} s]\!]\rangle \\ \downarrow{\scriptstyle \mathtt{P}} & & \vdots{\scriptstyle \mathtt{P}} \\ \mathtt{R}'\langle \mathtt{LCC}'\langle c'\rangle\rangle & \mathcal{R} & \mathtt{LCC}_3\langle \mathtt{LCC}_2\langle \mathtt{R}'\langle \mathtt{LCC}_1\langle c_1 [\![\beta \setminus^{\alpha'} s' \cdot s]\!]\rangle\rangle\rangle\rangle \end{array}$$

   b) No se superpone con c. Luego, $\mathtt{LCC} = \mathtt{LCC}_3\langle \mathtt{LCC}_2\langle \mathtt{LCC}_1 [\![\beta \setminus^{\alpha} s']\!]\rangle [\![\alpha \setminus^{\alpha'} s]\!]\rangle$. Se concluye con $c' = c$, $\mathtt{R}' = \mathtt{R}$ y $\mathtt{LCC}' = \mathtt{LCC}_3\langle \mathtt{LCC}_2\langle \mathtt{LCC}_1 [\![\beta \setminus^{\alpha'} s' \cdot s]\!]\rangle\rangle$:

$$\begin{array}{ccc} \mathtt{R}\langle \mathtt{LCC}\langle c\rangle\rangle & \mathcal{R} & \mathtt{LCC}_3\langle \mathtt{LCC}_2\langle \mathtt{LCC}_1\langle \mathtt{R}\langle c\rangle\rangle [\![\beta \setminus^{\alpha} s']\!]\rangle [\![\alpha \setminus^{\alpha'} s]\!]\rangle \\ \downarrow{\scriptstyle \mathtt{P}} & & \vdots{\scriptstyle \mathtt{P}} \\ \mathtt{R}'\langle \mathtt{LCC}'\langle c'\rangle\rangle & \mathcal{R} & \mathtt{LCC}_3\langle \mathtt{LCC}_2\langle \mathtt{LCC}_1\langle \mathtt{R}'\langle c'\rangle\rangle [\![\beta \setminus^{\alpha'} s' \cdot s]\!]\rangle\rangle \end{array}$$

5. W. Luego, hay dos posibilidades.

   a) Se superpone con c. Luego, se tienen $c = \mathtt{LCC}_1\langle c_1 [\beta \setminus \alpha]\rangle$, $\mathtt{LCC} = \mathtt{LCC}_3\langle \mathtt{LCC}_2 [\![\alpha \setminus^{\alpha'} s]\!]\rangle$ y la regla W aplica sobre $\mathtt{LCC}_2\langle \mathtt{LCC}_1\langle c_1 [\beta \setminus \alpha]\rangle\rangle [\![\alpha \setminus^{\alpha'} s]\!]$. Se concluye tomando $c' = \mathtt{LCC}_1\langle c [\![\beta \setminus^{\alpha} s]\!] [\alpha \setminus \alpha']\rangle$, $\mathtt{R}' = \mathtt{R}$ y $\mathtt{LCC}' = \mathtt{LCC}_3\langle \mathtt{LCC}_2\rangle$:

$$\begin{array}{ccc} \mathtt{R}\langle \mathtt{LCC}\langle c\rangle\rangle & \mathcal{R} & \mathtt{LCC}_3\langle \mathtt{LCC}_2\langle \mathtt{R}\langle \mathtt{LCC}_1\langle c_1 [\beta \setminus \alpha]\rangle\rangle\rangle [\![\alpha \setminus^{\alpha'} s]\!]\rangle \\ \downarrow{\scriptstyle \mathtt{W}} & & \vdots{\scriptstyle \mathtt{W}} \\ \mathtt{R}'\langle \mathtt{LCC}'\langle c'\rangle\rangle & \mathcal{R} & \mathtt{LCC}_3\langle \mathtt{LCC}_2\langle \mathtt{R}'\langle \mathtt{LCC}_1\langle c [\![\beta \setminus^{\alpha} s]\!] [\alpha \setminus \alpha']\rangle\rangle\rangle\rangle \end{array}$$

   b) No se superpone con c. Luego, $\mathtt{LCC} = \mathtt{LCC}_3\langle \mathtt{LCC}_2\langle \mathtt{LCC}_1 [\beta \setminus \alpha]\rangle [\![\alpha \setminus^{\alpha'} s]\!]\rangle$. Se concluye con $c' = c$, $\mathtt{R}' = \mathtt{R}$ y $\mathtt{LCC}' = \mathtt{LCC}_3\langle \mathtt{LCC}_2\langle \mathtt{LCC}_1 [\![\beta \setminus^{\alpha} s]\!] [\alpha \setminus \alpha']\rangle\rangle$:

$$\begin{array}{ccc} \mathtt{R}\langle \mathtt{LCC}\langle c\rangle\rangle & \mathcal{R} & \mathtt{LCC}_3\langle \mathtt{LCC}_2\langle \mathtt{LCC}_1\langle \mathtt{R}\langle c\rangle\rangle [\beta \setminus \alpha]\rangle [\![\alpha \setminus^{\alpha'} s]\!]\rangle \\ \downarrow{\scriptstyle \mathtt{W}} & & \vdots{\scriptstyle \mathtt{W}} \\ \mathtt{R}'\langle \mathtt{LCC}'\langle c'\rangle\rangle & \mathcal{R} & \mathtt{LCC}_3\langle \mathtt{LCC}_2\langle \mathtt{LCC}_1\langle \mathtt{R}'\langle c'\rangle\rangle [\![\beta \setminus^{\alpha} s]\!] [\alpha \setminus \alpha']\rangle\rangle \end{array}$$

- No hay más casos posibles dadas las hipótesis de la relación $\mathcal{R}$.

□

Lema 4.6.2.

1. *Sea* $t \in \mathbb{T}_{\Lambda M}$. *Luego,* $\mathfrak{C}(\mathtt{L}\langle \mathtt{LTT}\langle t\rangle\rangle) \simeq \mathfrak{C}(\mathtt{LTT}\langle \mathtt{L}\langle t\rangle\rangle)$ *si* $\mathsf{bv}(\mathtt{L}) \notin \mathtt{LTT}$ *y* $\mathsf{fc}(\mathtt{L}, \mathtt{LTT})$.
2. *Sea* $c \in \mathbb{C}_{\Lambda M}$. *Luego,* $\mathfrak{C}(\mathtt{R}\langle \mathtt{LCC}\langle c\rangle\rangle) \simeq \mathfrak{C}(\mathtt{LCC}\langle \mathtt{R}\langle c\rangle\rangle)$ *si* $\mathsf{bn}(\mathtt{R}) \notin \mathtt{LCC}$ *y* $\mathsf{fc}(\mathtt{R}, \mathtt{LCC})$.

Demostración.

1. Sea $\mathtt{L}\langle\mathtt{LTT}\langle t\rangle\rangle \in \mathbb{T}_{\Lambda M}$ con $\mathsf{bv}(\mathtt{L}) \notin \mathtt{LTT}$ y $\mathsf{fc}(\mathtt{L},\mathtt{LTT})$. Por Lem. C.0.45, $\langle \mathtt{L}\langle\mathtt{LTT}\langle t\rangle\rangle, \mathtt{LTT}\langle\mathtt{L}\langle t\rangle\rangle\rangle \in \mathcal{S}$. Más aún, $\mathcal{S}$ es una bisumulación fuerte con respecto a la relación de reducción canónica. Luego, $\mathfrak{C}(\mathtt{L}\langle\mathtt{LTT}\langle t\rangle\rangle) = \mathtt{L}'\langle\mathtt{LTT}'\langle t'\rangle\rangle$. Del mismo modo, $\mathfrak{C}(\mathtt{LTT}\langle\mathtt{L}\langle t\rangle\rangle) = \mathtt{LTT}'\langle\mathtt{L}'\langle t'\rangle\rangle$. Por una simple inducción en $\mathtt{L}$ utilizando $\simeq_{\mathtt{exsubs}}$ se concluye $\mathtt{L}'\langle\mathtt{LTT}'\langle t'\rangle\rangle \simeq \mathtt{LTT}'\langle\mathtt{L}'\langle t'\rangle\rangle$.

2. Sea $\mathtt{R}\langle\mathtt{LCC}\langle c\rangle\rangle \in \mathbb{C}_{\Lambda M}$ con $\mathsf{bn}(\mathtt{R}) \notin \mathtt{LCC}$ y $\mathsf{fc}(\mathtt{R},\mathtt{LCC})$. Por Lem. C.0.46, $\langle \mathtt{R}\langle\mathtt{LCC}\langle c\rangle\rangle, \mathtt{LCC}\langle\mathtt{R}\langle c\rangle\rangle\rangle \in \mathcal{R}$. Más aún, $\mathcal{R}$ es una bisumulación fuerte con respecto a la relación de reducción canónica. Luego, $\mathfrak{C}(\mathtt{R}\langle\mathtt{LCC}\langle c\rangle\rangle) = \mathtt{R}'\langle\mathtt{LCC}'\langle c'\rangle\rangle$. Del mismo modo, $\mathfrak{C}(\mathtt{LCC}\langle\mathtt{R}\langle c\rangle\rangle) = \mathtt{LCC}'\langle\mathtt{R}'\langle c'\rangle\rangle$. Por una simple inducción en $\mathtt{R}$ utilizando $\simeq_{\mathtt{exrepl}}$ se concluye $\mathtt{R}'\langle\mathtt{LCC}'\langle c'\rangle\rangle \simeq \mathtt{LCC}'\langle\mathtt{R}'\langle c'\rangle\rangle$.

□

Lema 4.6.3 (Preservación de tipos para $\simeq$)**.** *Sea $o \in \mathbb{O}_{\Lambda M}$. Si $\pi \rhd_{\mathcal{E}} \Gamma \vdash o : T \mid \Delta$ y $o \simeq o'$, entonces existe $\pi' \rhd_{\mathcal{E}} \Gamma \vdash o' : T \mid \Delta'$.*

Demostración. La prueba es por inducción en $o \simeq o'$. Por definición esto implica verificar cuatro condiciones: reflexividad, transitividad, simetría y congruencia (*i.e.* clausura por contextos):

Reflexividad Luego, $o' = o$ por lo que el resultado es inmediato.

Transitividad Luego, existe $p \in \mathbb{O}_{\Lambda M}$ tal que $o \simeq p$ y $p \simeq o'$. Por *h.i.* con $o \simeq p$ existe $\pi_p \rhd_{\mathcal{E}} \Gamma \vdash p : T \mid \Delta^*$. Más aún, por *h.i.* nuevamente, ahora con $p \simeq o'$, existe $\pi' \rhd_{\mathcal{E}} \Gamma \vdash o' : T \mid \Delta'$ por lo que se concluye.

Simetría/Congruencia La simetría y congruencia se demuestran simultáneamente por inducción en el contexto de clausura $\mathtt{O}$ tal que $o = \mathtt{O}\langle p\rangle$ y $o' = \mathtt{O}\langle p'\rangle$ con $p \simeq_* p'$, donde $\simeq_*$ es una regla de la Fig. 4.13.

- $\mathtt{O} = \Box$. Luego, $o = p \simeq_* p' = o'$. Más aún, los objetos son necesariamente términos. Se analiza la regla aplicada:
  - $\simeq_{\mathtt{exsubs}}$. Luego, $o = \mathtt{LTT}\langle t\rangle[x \setminus u]$ y $o' = \mathtt{LTT}\langle t[x \setminus u]\rangle$ con $x \notin \mathtt{LTT}$ y $\mathsf{fc}(u,\mathtt{LTT})$. Se realiza a continuación el análisis sobre o. El caso simétrico es análogo y se omite. De π por (SUBS) se tienen $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta = \Delta_0 \cup \Delta_1$ con $\pi_{\mathtt{LTT}\langle t\rangle} \rhd_{\mathcal{E}} \Gamma_0; (x : A)^{\leq 1} \vdash \mathtt{LTT}\langle t\rangle : T \mid \Delta_0$ y $\pi_u \rhd_{\mathcal{E}} \Gamma_1 \vdash u : A \mid \Delta_1$. Se procede por inducción en $\mathtt{LTT}$.
    - $\mathtt{LTT} = \Box$. Luego, $o = o'$ y el resultado es inmediato.
    - $\mathtt{LTT} = \mathtt{LTT}'\,u'$. Luego, por (APP), $\Gamma_0 = \Gamma_0' \cup \Gamma_0''$ y $\Delta_0 = \Delta_0' \cup \Delta_0''$ con $\pi_{\mathtt{LTT}'\langle t\rangle} \rhd_{\mathcal{E}} \Gamma_0'; (x : A)^{\leq 1} \vdash \mathtt{LTT}'\langle t\rangle : B \to T \mid \Delta_0'$ y $\pi_{u'} \rhd_{\mathcal{E}} \Gamma_0''; (x : A)^{\leq 1} \vdash u' : B \mid \Delta_0''$. Considerar los objetos $q = \mathtt{LTT}'\langle t\rangle[x \setminus u]$ y $q' = \mathtt{LTT}'\langle t[x \setminus u]\rangle$. Por (SUBS) con π_u, se obtiene $\pi_q \rhd_{\mathcal{E}} \Gamma_0' \cup \Gamma_1 \vdash q : B \to T \mid \Delta_0' \cup \Delta_1$. Luego, por *h.i.* existe una derivación $\pi_{q'} \rhd_{\mathcal{E}} \Gamma_0' \cup \Gamma_1 \vdash q' : B \to T \mid \Delta_0' \cup \Delta_1$. Se concluye por (APP) con $\pi_{u'}, \pi_{o'} \rhd_{\mathcal{E}} \Gamma \vdash o' : T \mid \Delta$. Notar que, por hipótesis de la regla $\simeq_{\mathtt{exsubs}}$, $x \notin u'$. Luego, por Lem. 4.3.1, $(x : A)^{\leq 1}$ es vacío en $\pi_{u'}$.
    - $\mathtt{LTT} = \lambda y.\mathtt{LTT}'$. Luego, por (APP) se tiene $T = B \to A'$ con la derivación $\pi_{\mathtt{LTT}'\langle t\rangle} \rhd_{\mathcal{E}} \Gamma_0; (x : A)^{\leq 1}; (y : B)^{\leq 1} \vdash \mathtt{LTT}'\langle t\rangle : A' \mid \Delta_0$. Considerar los objetos $q = \mathtt{LTT}'\langle t\rangle[x \setminus u]$ y $q' = \mathtt{LTT}'\langle t[x \setminus u]\rangle$. Por (SUBS) con π_u, se obtiene $\pi_q \rhd_{\mathcal{E}} \Gamma; (y : B)^{\leq 1} \vdash q : A' \mid \Delta$. Luego, por *h.i.* existe $\pi_{q'} \rhd_{\mathcal{E}} \Gamma; (y : B)^{\leq 1} \vdash q' : A' \mid \Delta$. Se concluye por (ABS), $\pi_{o'} \rhd_{\mathcal{E}} \Gamma \vdash o' : T \mid \Delta$.
    - $\mathtt{LTT} = \mu\alpha.\mathtt{LCT}$. Luego, por (CONT), se tiene la derivación para la premisa $\pi_{\mathtt{LCT}\langle t\rangle} \rhd_{\mathcal{E}} \Gamma_0; (x : A)^{\leq 1} \vdash \mathtt{LCT}\langle t\rangle \mid \Delta_0; (\alpha : T)^{\leq 1}$. Se procede analizando la forma de $\mathtt{LCT}$.

◇ $\mathtt{LCT} = [\gamma]\,\mathtt{LTT}'$. Luego, de (NAME), se tiene $\Delta_0 = \Delta^*; (\gamma : B)^{\leq 1}$ con la derivación $\pi_{\mathtt{LTT}'\langle t\rangle} \rhd_{\mathcal{E}} \Gamma_0; (x : A)^{\leq 1} \vdash \mathtt{LTT}'\langle t\rangle : B \mid \Delta^*; ((\alpha : T)^{\leq 1} \cup (\gamma : B)^{\leq 1})$. Considerar los objetos $q = \mathtt{LTT}'\langle t\rangle[x \setminus u]$ y $q' = \mathtt{LTT}'\langle t[x \setminus u]\rangle$. Por (SUBS) con π_u, $\pi_q \rhd_{\mathcal{E}} \Gamma \vdash q : B \mid \Delta^*; ((\alpha : T)^{\leq 1} \cup (\gamma : B)^{\leq 1})$. Por *h.i.* existe una derivación $\pi_{q'} \rhd_{\mathcal{E}} \Gamma \vdash q' : B \mid \Delta^*; ((\alpha : T)^{\leq 1} \cup (\gamma : B)^{\leq 1})$. Se concluye por (NAME) con γ seguido de (CONT) con α, $\pi_{o'} \rhd_{\mathcal{E}} \Gamma \vdash o' : T \mid \Delta$.

◇ $\mathtt{LCT} = \mathtt{LCT}'[\![\gamma \setminus^{\delta} s]\!]$. De (REPL) se tiene $\Gamma_0 = \Gamma_0' \cup \Gamma_0''$ y $\Delta_0 = \Delta_0' \cup \Delta_0''; (\delta : B)^{\leq 1}$ con derivaciones $\pi_{\mathtt{LCT}'\langle t\rangle} \rhd_{\mathcal{E}} \Gamma_0' \vdash \mathtt{LCT}'\langle t\rangle \mid \Delta_0'; (\gamma : S \to B)^{\leq 1}; ((\alpha : T)^{\leq 1} \cup (\delta : B)^{\leq 1})$ y $\pi_s \rhd_{\mathcal{E}} \Gamma_0'' \vdash s : S \mid \Delta_0''; (\alpha : T)^{\leq 1}$. Considerar los objetos $q = \mathtt{LTT}'\langle t\rangle[x \setminus u]$ y $q' = \mathtt{LTT}'\langle t[x \setminus u]\rangle$ donde $\mathtt{LTT}' = \mu\alpha.\mathtt{LCT}'$. Por (CONT) seguido de (SUBS) con π_u, se obtiene $\pi_q \rhd_{\mathcal{E}} \Gamma_0' \cup \Gamma_1 \vdash q : T \mid \Delta_0' \cup \Delta_1; (\gamma : S \to B)^{\leq 1}; (\delta : B)^{\leq 1}$. Por *h.i.* existe una derivación $\pi_{q'} \rhd_{\mathcal{E}} \Gamma_0' \cup \Gamma_1 \vdash q' : T \mid \Delta_0' \cup \Delta_1; (\gamma : S \to B)^{\leq 1}; (\delta : B)^{\leq 1}$. Esta derivación necesariamente termina con la regla (CONT). Luego, se tiene $\pi_{\mathtt{LCT}'\langle t[x\setminus u]\rangle} \rhd_{\mathcal{E}} \Gamma_0' \cup \Gamma_1 \vdash \mathtt{LCT}'\langle t[x \setminus u]\rangle \mid \Delta_0' \cup \Delta_1; (\gamma : S \to B)^{\leq 1}; ((\alpha : T)^{\leq 1} \cup (\delta : B)^{\leq 1})$. Se concluye por (REPL) con π_s seguido de (CONT) con α, $\pi_{o'} \rhd_{\mathcal{E}} \Gamma \vdash o' : T \mid \Delta$.

◇ $\mathtt{LCT} = \mathtt{LCT}'[\gamma \setminus \delta]$. De (REN) se tiene $\Delta_0 = \Delta^*; (\delta : B)^{\leq 1}$ con la derivación $\pi_{\mathtt{LCT}'\langle t\rangle} \rhd_{\mathcal{E}} \Gamma_0 \vdash \mathtt{LCT}'\langle t\rangle \mid \Delta^*; (\gamma : B)^{\leq 1}; ((\alpha : T)^{\leq 1} \cup (\delta : B)^{\leq 1})$. Considerar los objetos $q = \mathtt{LTT}'\langle t\rangle[x \setminus u]$ y $q' = \mathtt{LTT}'\langle t[x \setminus u]\rangle$ donde $\mathtt{LTT}' = \mu\alpha.\mathtt{LCT}'$. Por (CONT) seguido de (SUBS) con π_u, se obtiene $\pi_q \rhd_{\mathcal{E}} \Gamma \vdash q : T \mid \Delta^* \cup \Delta_1; (\gamma : B)^{\leq 1}; (\delta : B)^{\leq 1}$. Por *h.i.* existe una derivación $\pi_{q'} \rhd_{\mathcal{E}} \Gamma \vdash q' : T \mid \Delta^* \cup \Delta_1; (\gamma : B)^{\leq 1}; (\delta : B)^{\leq 1}$. Esta derivación necesariamente termina con la regla (CONT). Luego, se tiene $\pi_{\mathtt{LCT}'\langle t[x\setminus u]\rangle} \rhd_{\mathcal{E}} \Gamma \vdash \mathtt{LCT}'\langle t[x \setminus u]\rangle \mid \Delta^* \cup \Delta_1; (\gamma : B)^{\leq 1}; ((\alpha : T)^{\leq 1} \cup (\delta : B)^{\leq 1})$. Se concluye por (REN) con γ y δ seguido de (CONT) con α, $\pi_{o'} \rhd_{\mathcal{E}} \Gamma \vdash o' : T \mid \Delta$.

○ $\mathtt{LTT} = \mathtt{LTT}'[y \setminus u']$. Luego, por (SUBS), se tienen los contextos $\Gamma_0 = \Gamma_0' \cup \Gamma_0''$ y $\Delta_0 = \Delta_0' \cup \Delta_0''$ con las derivaciones $\pi_{\mathtt{LTT}'\langle t\rangle} \rhd_{\mathcal{E}} \Gamma_0'; (x : A)^{\leq 1}; (y : B)^{\leq 1} \vdash \mathtt{LTT}'\langle t\rangle : T \mid \Delta_0'$ y $\pi_{u'} \rhd_{\mathcal{E}} \Gamma_0''; (x : A)^{\leq 1} \vdash u' : B \mid \Delta_0''$. Considerar $q = \mathtt{LTT}'\langle t\rangle[x \setminus u]$ y $q' = \mathtt{LTT}'\langle t[x \setminus u]\rangle$. Por (SUBS) con π_u, se obtiene la derivación $\pi_q \rhd_{\mathcal{E}} \Gamma_0' \cup \Gamma_1; (y : B)^{\leq 1} \vdash q : T \mid \Delta_0' \cup \Delta_1$. Luego, por *h.i.* existe $\pi_{q'} \rhd_{\mathcal{E}} \Gamma_0' \cup \Gamma_1; (y : B)^{\leq 1} \vdash q' : T \mid \Delta_0' \cup \Delta_1$. Se concluye por (APP) con $\pi_{u'}$, $\pi_{o'} \rhd_{\mathcal{E}} \Gamma \vdash o' : T \mid \Delta$. Notar que, por hipótesis de la regla $\simeq_{\mathtt{exsubs}}$, $x \notin u'$. Luego, por Lem. 4.3.1, $(x : A)^{\leq 1}$ es vacío en $\pi_{u'}$.

- $\simeq_\theta$. Luego, $o = \mu\alpha.[\alpha]\,t$ y $o' = t$ con $\alpha \notin t$. Más aún, de π por (CONT) se tiene la derivación $\pi_{[\alpha]\,t} \rhd_{\mathcal{E}} \Gamma \vdash [\alpha]\,t \mid \Delta; \alpha : T$ y, por (NAME), $\pi_t \rhd_{\mathcal{E}} \Gamma \vdash t \mid \Delta; (\alpha : T)^{\leq 1}$. Se concluye por Lem. 4.3.1, pues $\alpha \notin t$ implica $(\alpha : T)^{\leq 1}$ vacío. Para el caso simétrico, se parte de π_t y se concluye aplicando (NAME) seguido de (CONT).

▪ $\mathtt{O} = \boxdot$. Luego, $o = p \simeq_* p' = o'$. Más aún, los objetos son necesariamente comandos. Se analiza la regla aplicada:

- $\simeq_{\mathtt{exrepl}}$. Luego, $o = \mathtt{LCC}\langle c\rangle[\![\alpha \setminus^{\alpha'} s]\!]$ y $o' = \mathtt{LCC}\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle$ con $\alpha \notin \mathtt{LCC}$, $\mathsf{fc}(\alpha', \mathtt{LCC})$ y $\mathsf{fc}(s, \mathtt{LCC})$. Se realiza a continuación el análisis sobre o. El caso simétrico es análogo y se omite. De π por (REPL) se tienen $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta = \Delta_0 \cup \Delta_1; \alpha' : A$ con $\pi_{\mathtt{LCC}\langle c\rangle} \rhd_{\mathcal{E}} \Gamma_0 \vdash \mathtt{LCC}\langle c\rangle \mid \Delta_0; (\alpha : S \to A)^{\leq 1}; (\alpha' : A)^{\leq 1}$ y $\pi_s \rhd_{\mathcal{E}} \Gamma_1 \vdash s : S \mid \Delta_1; (\alpha' : A)^{\leq 1}$. Se procede por inducción en LCC.

○ $\mathtt{LCC} = \boxdot$. Luego, $o = o'$ y el resultado es inmediato.

○ $\mathtt{LCC} = [\gamma]\,\mathtt{LTC}$. Luego, por (NAME), $\Delta_0 = \Delta_2; \gamma : B$ con la derivación $\pi_{\mathtt{LTC}\langle c\rangle} \rhd_{\mathcal{E}} \Gamma_0 \vdash \mathtt{LTC}\langle c\rangle : B \mid \Delta_2; (\alpha : S \to A)^{\leq 1}; ((\alpha' : A)^{\leq 1} \cup (\gamma : B)^{\leq 1})$. Se procede analizando la forma de LTC.

◇ $\mathtt{LTC} = \mathtt{LTC}'\,u$. Luego, por (APP), se tienen contextos $\Gamma_0 = \Gamma_0' \cup \Gamma_0''$ y $\Delta_2 = \Delta_2' \cup \Delta_2''$ con $\pi_{\mathtt{LTC}'\langle c\rangle} \rhd_{\mathcal{E}} \Gamma_0' \vdash \mathtt{LTC}'\langle c\rangle : B' \to B \mid \Delta_2'; (\alpha : S \to A)^{\leq 1}; ((\alpha' : A)^{\leq 1} \cup (\gamma : B)^{\leq 1})$ y $\pi_u \rhd_{\mathcal{E}} \Gamma_0'' \vdash u : B' \mid \Delta_2''; (\alpha : S \to A)^{\leq 1}; ((\alpha' : A)^{\leq 1} \cup (\gamma : B)^{\leq 1})$. Considerar $q =$

$\mathtt{LCC}'\langle c\rangle[\![\alpha \setminus^{\alpha'} s]\!]$ y $q' = \mathtt{LCC}'\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle$ donde $\mathtt{LCC}' = [\delta]\,\mathtt{LTC}'$ con δ un nombre fresco. Entonces, por (NAME) y (REPL) con π_s se obtiene la derivación $\pi_q \rhd_{\mathcal{E}} \Gamma'_0 \cup \Gamma_1 \vdash q \mid \Delta_1 \cup \Delta'_2; \delta : B' \to B; (\alpha' : A \cup (\gamma : B)^{\leq 1})$. Por *h.i.* existe una derivación $\pi_{q'} \rhd_{\mathcal{E}} \Gamma'_0 \cup \Gamma_1 \vdash q' \mid \Delta_1 \cup \Delta'_2; \delta : B' \to B; (\alpha' : A \cup (\gamma : B)^{\leq 1})$. Esta derivación necesariamente termina con la regla (NAME). Luego, se tiene $\pi_{\mathtt{LTC}'\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle} \rhd_{\mathcal{E}} \Gamma'_0 \cup \Gamma_1 \vdash \mathtt{LTC}'\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle : B' \to B \mid \Delta_1 \cup \Delta'_2; (\alpha' : A \cup (\gamma : B)^{\leq 1})$, dado que δ es fresco. Finalmente, se concluye por (APP) con π_u y luego (NAME) con γ, $\pi_{o'} \rhd_{\mathcal{E}} \Gamma \vdash o' \mid \Delta$. Notar que, por hipótesis de la regla $\simeq_{\mathtt{exrepl}}$, $\alpha \notin u$. Luego, por Lem. 4.3.1, $(\alpha : S \to A)^{\leq 1}$ es vacío en π_u.

- $\mathtt{LTC} = \lambda x.\mathtt{LTC}'$. Luego, por (ABS), se tiene $B = B' \to B''$ y la derivación $\pi_{\mathtt{LTC}'\langle c\rangle} \rhd_{\mathcal{E}} \Gamma_0; (x : B')^{\leq 1} \vdash \mathtt{LTC}'\langle c\rangle : B'' \mid \Delta_2; (\alpha : S \to A)^{\leq 1}; ((\alpha' : A)^{\leq 1} \cup (\gamma : B)^{\leq 1})$. Considerar $q = \mathtt{LCC}'\langle c\rangle[\![\alpha \setminus^{\alpha'} s]\!]$ y $q' = \mathtt{LCC}'\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle$ donde $\mathtt{LCC}' = [\delta]\,\mathtt{LTC}'$ con δ un nombre fresco. Entonces, por (NAME) y (REPL) con π_s se obtiene la derivación $\pi_q \rhd_{\mathcal{E}} \Gamma; (x : B')^{\leq 1} \vdash q \mid \Delta_1 \cup \Delta_2; (\alpha' : A \cup (\gamma : B)^{\leq 1}); \delta : B''$. Por *h.i.* existe una derivación $\pi_{q'} \rhd_{\mathcal{E}} \Gamma; (x : B')^{\leq 1} \vdash q' \mid \Delta_1 \cup \Delta_2; (\alpha' : A \cup (\gamma : B)^{\leq 1}); \delta : B''$. Esta derivación necesariamente termina con (NAME). Luego, $\pi_{\mathtt{LTC}'\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle} \rhd_{\mathcal{E}} \Gamma; (x : B')^{\leq 1} \vdash \mathtt{LTC}'\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle : B'' \mid \Delta_1 \cup \Delta_2; (\alpha' : A \cup (\gamma : B)^{\leq 1})$, dado que δ es fresco. Finalmente, se concluye por (ABS) y (NAME) con γ, $\pi_{o'} \rhd_{\mathcal{E}} \Gamma \vdash o' \mid \Delta$.
- $\mathtt{LTC} = \mu\delta.\mathtt{LCC}'$. Luego, por (CONT), se tiene la derivación para la premisa $\pi_{\mathtt{LCC}'\langle c\rangle} \rhd_{\mathcal{E}} \Gamma_0 \vdash \mathtt{LCC}'\langle c\rangle \mid \Delta'_0; (\alpha : S \to A)^{\leq 1}; ((\alpha' : A)^{\leq 1} \cup (\gamma : B)^{\leq 1}); (\delta : B)^{\leq 1}$. Aplicando la regla (REPL) con las premisas $\pi_{\mathtt{LCC}'\langle c\rangle}$ y π_s, se obtiene la derivación $\pi_{\mathtt{LCC}'\langle c\rangle[\![\alpha \setminus^{\alpha'} s]\!]} \rhd_{\mathcal{E}} \Gamma \vdash \mathtt{LCC}'\langle c\rangle[\![\alpha \setminus^{\alpha'} s]\!] \mid \Delta; (\alpha' : A \cup (\gamma : B)^{\leq 1}); (\delta : B)^{\leq 1}$. Por *h.i.* existe una derivación $\pi_{\mathtt{LCC}'\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle} \rhd_{\mathcal{E}} \Gamma \vdash \mathtt{LCC}'\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle \mid \Delta; (\alpha' : A \cup (\gamma : B)^{\leq 1}); (\delta : B)^{\leq 1}$. Finalmente, se concluye por (CONT) y (NAME), $\pi_{o'} \rhd_{\mathcal{E}} \Gamma \vdash o' \mid \Delta$.
- $\mathtt{LTC} = \mathtt{LTC}'[x \setminus u]$. Luego, por (SUBS), $\Gamma_0 = \Gamma'_0 \cup \Gamma''_0$ y $\Delta_2 = \Delta'_2 \cup \Delta''_2$ con $\pi_{\mathtt{LTC}'\langle c\rangle} \rhd_{\mathcal{E}} \Gamma'_0; (x : B')^{\leq 1} \vdash \mathtt{LTC}'\langle c\rangle : B \mid \Delta'_2; (\alpha : S \to A)^{\leq 1}; ((\alpha' : A)^{\leq 1} \cup (\gamma : B)^{\leq 1})$ y $\pi_u \rhd_{\mathcal{E}} \Gamma''_0 \vdash u : B' \mid \Delta''_2; (\alpha : S \to A)^{\leq 1}; ((\alpha' : A)^{\leq 1} \cup (\gamma : B)^{\leq 1})$. Sean $q = \mathtt{LCC}'\langle c\rangle[\![\alpha \setminus^{\alpha'} s]\!]$ y $q' = \mathtt{LCC}'\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle$ donde $\mathtt{LCC}' = [\gamma]\,\mathtt{LTC}'$. Entonces, por (NAME) y (REPL) con π_s, $\pi_q \rhd_{\mathcal{E}} \Gamma'_0 \cup \Gamma_1; (x : B')^{\leq 1} \vdash q \mid \Delta_1 \cup \Delta'_2; (\alpha' : A \cup \gamma : B)$. Por *h.i.* existe una derivación $\pi_{q'} \rhd_{\mathcal{E}} \Gamma'_0 \cup \Gamma_1; (x : B')^{\leq 1} \vdash q' \mid \Delta_1 \cup \Delta'_2; (\alpha' : A \cup \gamma : B)$. Esta derivación necesariamente termina con la regla (NAME). Luego, se tiene $\pi_{\mathtt{LTC}'\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle} \rhd_{\mathcal{E}} \Gamma'_0 \cup \Gamma_1; (x : B')^{\leq 1} \vdash \mathtt{LTC}'\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle : B \mid \Delta_1 \cup \Delta'_2; (\alpha' : A \cup (\gamma : B)^{\leq 1})$. Por último, se concluye por (SUBS) con π_u y luego (NAME), $\pi_{o'} \rhd_{\mathcal{E}} \Gamma \vdash o' \mid \Delta$.

- $\mathtt{LCC} = \mathtt{LCC}'[\![\gamma \setminus^{\delta} s']\!]$. Luego, por (REPL), $\Gamma_0 = \Gamma'_0 \cup \Gamma''_0$ y $\Delta_0 = \Delta'_0 \cup \Delta''_0; \delta : B$ con $\pi_{\mathtt{LCC}'\langle c\rangle} \rhd_{\mathcal{E}} \Gamma'_0 \vdash \mathtt{LCC}'\langle c\rangle \mid \Delta'_0; (\alpha : S \to A)^{\leq 1}; (\gamma : S' \to B)^{\leq 1}; ((\alpha' : A)^{\leq 1} \cup (\delta : B)^{\leq 1})$ y $\pi_{s'} \rhd_{\mathcal{E}} \Gamma''_0 \vdash s' : S' \mid \Delta''_0; (\alpha : S \to A)^{\leq 1}; ((\alpha' : A)^{\leq 1} \cup (\delta : B)^{\leq 1})$. Entonces, aplicando la regla (REPL) con la premisa π_s, se obtiene la derivación $\pi_{\mathtt{LCC}'\langle c\rangle[\![\alpha \setminus^{\alpha'} s]\!]} \rhd_{\mathcal{E}} \Gamma'_0 \cup \Gamma_1 \vdash \mathtt{LCC}'\langle c\rangle[\![\alpha \setminus^{\alpha'} s]\!] \mid \Delta'_0 \cup \Delta_1; (\gamma : S' \to B)^{\leq 1}; (\alpha' : A \cup (\delta : B)^{\leq 1})$. Más aún, aplicando la *h.i.* sobre $\mathtt{LCC}'$, se garantiza que existe una derivación $\pi_{\mathtt{LCC}'\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle} \rhd_{\mathcal{E}} \Gamma'_0 \cup \Gamma_1 \vdash \mathtt{LCC}'\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle \mid \Delta'_0 \cup \Delta_1; (\gamma : S' \to B)^{\leq 1}; (\alpha' : A \cup (\delta : B)^{\leq 1})$. Finalmente, se concluye por (REPL) con $\pi_{s'}$, $\pi_{o'} \rhd_{\mathcal{E}} \Gamma \vdash o' \mid \Delta$. Notar que, por hipótesis de la regla P, $\alpha \notin s'$ y, por Lem. 4.3.1, $(\alpha : S \to A)^{\leq 1}$ es vacío en $\pi_{s'}$.
- $\mathtt{LCC} = \mathtt{LCC}'[\gamma \setminus \delta]$. Por (REN), se tienen $\Delta_0 = \Delta'_0; \delta : B$ con la derivación $\pi_{\mathtt{LCC}'\langle c\rangle} \rhd_{\mathcal{E}} \Gamma_0 \vdash \mathtt{LCC}'\langle c\rangle \mid \Delta_0; (\alpha : S \to A)^{\leq 1}; ((\alpha' : A)^{\leq 1} \cup (\delta : B)^{\leq 1}); (\gamma : B)^{\leq 1}$. Entonces, aplicando la regla (REPL) con la premisa π_s se obtiene la derivación $\pi_{\mathtt{LCC}'\langle c\rangle[\![\alpha \setminus^{\alpha'} s]\!]} \rhd_{\mathcal{E}} \Gamma_0 \cup \Gamma_1 \vdash \mathtt{LCC}'\langle c\rangle[\![\alpha \setminus^{\alpha'} s]\!] \mid \Delta'_0 \cup \Delta_1; (\alpha' : A \cup (\delta : B)^{\leq 1}); (\gamma : B)^{\leq 1}$. Más aún, aplicando la *h.i.* sobre $\mathtt{LCC}'$, se garantiza que existe una derivación $\pi_{\mathtt{LCC}'\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle} \rhd_{\mathcal{E}}$

$\Gamma_0 \cup \Gamma_1 \vdash \mathtt{LCC}'\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle \mid \Delta_0' \cup \Delta_1; (\alpha' : A \cup (\delta : B)^{\leq 1}); (\gamma : B)^{\leq 1}$. Finalmente, se concluye por (REN), $\pi_{o'} \rhd_{\mathcal{E}} \Gamma \vdash o' \mid \Delta$.

- $\simeq_{\mathtt{exren}}$. Luego, $o = \mathtt{LCC}\langle c\rangle[\alpha \setminus \beta]$ y $o' = \mathtt{LCC}\langle c[\alpha \setminus \beta]\rangle$ con $\alpha \notin \mathtt{LCC}$ y $\mathsf{fc}(\beta, \mathtt{LCC})$. Se realiza a continuación el análisis sobre o. El caso simétrico es análogo y se omite. De π por (REN) se tiene $\Delta = \Delta^*; \beta : A$ con $\pi_{\mathtt{LCC}\langle c\rangle} \rhd_{\mathcal{E}} \Gamma \vdash \mathtt{LCC}\langle c\rangle \mid \Delta^*; (\alpha : A)^{\leq 1}; (\beta : A)^{\leq 1}$. Se procede por inducción en LCC.
  - $\mathtt{LCC} = \boxdot$. Luego, $o = o'$ y el resultado es inmediato.
  - $\mathtt{LCC} = [\gamma]\,\mathtt{LTC}$. Luego, por (NAME), se tienen $\Delta^* = \Delta'; \gamma : B$ con la derivación $\pi_{\mathtt{LTC}\langle c\rangle} \rhd_{\mathcal{E}} \Gamma \vdash \mathtt{LTC}\langle c\rangle : B \mid \Delta'; (\alpha : A)^{\leq 1}; ((\beta : A)^{\leq 1} \cup (\gamma : B)^{\leq 1})$. Se procede analizando la forma de LTC.
    - $\mathtt{LTC} = \mathtt{LTC}'\,u$. Luego, por (APP), se tienen contextos $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta' = \Delta_0 \cup \Delta_1$ con $\pi_{\mathtt{LTC}'\langle c\rangle} \rhd_{\mathcal{E}} \Gamma_0 \vdash \mathtt{LTC}'\langle c\rangle : B' \to B \mid \Delta_0; (\alpha : A)^{\leq 1}; ((\beta : A)^{\leq 1} \cup (\gamma : B)^{\leq 1})$ y $\pi_u \rhd_{\mathcal{E}} \Gamma_1 \vdash u : B' \mid \Delta_1; (\alpha : A)^{\leq 1}; ((\beta : A)^{\leq 1} \cup (\gamma : B)^{\leq 1})$. Sean $q = \mathtt{LCC}'\langle c\rangle[\alpha \setminus \beta]$ y $q' = \mathtt{LCC}'\langle c[\alpha \setminus \beta]\rangle$ donde $\mathtt{LCC}' = [\delta]\,\mathtt{LTC}'$ con δ un nombre fresco. Entonces, aplicando la regla (NAME) con $\pi_{\mathtt{LTC}'\langle c\rangle}$ y δ seguida de (REN) con α y β, se obtiene la derivación $\pi_q \rhd_{\mathcal{E}} \Gamma_0 \vdash q \mid \Delta_0; (\beta : A \cup (\gamma : B)^{\leq 1}); \delta : B' \to B$. Por *h.i.* existe $\pi_{q'} \rhd_{\mathcal{E}} \Gamma_0 \vdash q' \mid \Delta_0; (\beta : A \cup (\gamma : B)^{\leq 1}); \delta : B' \to B$. De (NAME), dado que δ es fresco, se tiene la $\pi_{\mathtt{LTC}'\langle c[\alpha\setminus\beta]\rangle} \rhd_{\mathcal{E}} \Gamma_0 \vdash \mathtt{LTC}'\langle c[\alpha \setminus \beta]\rangle : B' \to B \mid \Delta_0; (\beta : A \cup (\gamma : B)^{\leq 1})$. Finalmente, se concluye por (APP) con π_u y luego (NAME) con γ, $\pi_{o'} \rhd_{\mathcal{E}} \Gamma \vdash o' \mid \Delta$.
    - $\mathtt{LTC} = \lambda x.\mathtt{LTC}'$. Luego, por (ABS), se tiene $B = B' \to B''$ con $\pi_{\mathtt{LTC}'\langle c\rangle} \rhd_{\mathcal{E}} \Gamma; (x : B')^{\leq 1} \vdash \mathtt{LTC}'\langle c\rangle : B'' \mid \Delta'; (\alpha : A)^{\leq 1}; ((\beta : A)^{\leq 1} \cup (\gamma : B)^{\leq 1})$. Más aún, sean $q = \mathtt{LCC}'\langle c\rangle[\alpha \setminus \beta]$ y $q' = \mathtt{LCC}'\langle c[\alpha \setminus \beta]\rangle$ donde $\mathtt{LCC}' = [\delta]\,\mathtt{LTC}'$ con δ un nombre fresco. Entonces, de $\pi_{\mathtt{LTC}'\langle c\rangle}$ por (NAME) y (REN) se obtiene la derivación $\pi_q \rhd_{\mathcal{E}} \Gamma; (x : B')^{\leq 1} \vdash q \mid \Delta'; (\beta : A \cup (\gamma : B)^{\leq 1}); \delta : B''$. Por *h.i.* existe una derivación $\pi_{q'} \rhd_{\mathcal{E}} \Gamma; (x : B')^{\leq 1} \vdash q' \mid \Delta'; (\beta : A \cup (\gamma : B)^{\leq 1}); \delta : B''$. Esta derivación necesariamente termina con la regla (NAME). Luego, se tiene la premisa $\pi_{\mathtt{LTC}'\langle c[\alpha\setminus\beta]\rangle} \rhd_{\mathcal{E}} \Gamma; (x : B')^{\leq 1} \vdash \mathtt{LTC}'\langle c[\alpha \setminus \beta]\rangle : B'' \mid \Delta'; (\beta : A \cup (\gamma : B)^{\leq 1})$, dado que δ es fresco. Se concluye por (ABS) y (NAME) con γ, $\pi_{o'} \rhd_{\mathcal{E}} \Gamma \vdash o' \mid \Delta$.
    - $\mathtt{LTC} = \mu\delta.\mathtt{LCC}'$. Luego, por (CONT), se tiene la derivación para la premisa $\pi_{\mathtt{LCC}'\langle c\rangle} \rhd_{\mathcal{E}} \Gamma \vdash \mathtt{LCC}'\langle c\rangle \mid \Delta'; (\alpha : A)^{\leq 1}; ((\beta : A)^{\leq 1} \cup (\gamma : B)^{\leq 1}); (\delta : B)^{\leq 1}$. Por (REN) con α y β, se obtiene $\pi_{\mathtt{LCC}'\langle c\rangle[\alpha\setminus\beta]} \rhd_{\mathcal{E}} \Gamma \vdash \mathtt{LCC}'\langle c\rangle[\alpha \setminus \beta] \mid \Delta; (\beta : A \cup (\gamma : B)^{\leq 1}); (\delta : B)^{\leq 1}$. Por *h.i.* existe $\pi_{\mathtt{LCC}'\langle c[\alpha\setminus\beta]\rangle} \rhd_{\mathcal{E}} \Gamma \vdash \mathtt{LCC}'\langle c[\alpha \setminus \beta]\rangle \mid \Delta; (\beta : A \cup (\gamma : B)^{\leq 1}); (\delta : B)^{\leq 1}$. Finalmente, se concluye por (CONT) y (NAME), $\pi_{o'} \rhd_{\mathcal{E}} \Gamma \vdash o' \mid \Delta$.
    - $\mathtt{LTC} = \mathtt{LTC}'[x \setminus u]$. Luego, por (SUBS), se tienen $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta' = \Delta_0 \cup \Delta_1$ con $\pi_{\mathtt{LTC}'\langle c\rangle} \rhd_{\mathcal{E}} \Gamma_0; (x : B')^{\leq 1} \vdash \mathtt{LTC}'\langle c\rangle : B \mid \Delta_0; (\alpha : A)^{\leq 1}; ((\beta : A)^{\leq 1} \cup (\gamma : B)^{\leq 1})$ y $\pi_u \rhd_{\mathcal{E}} \Gamma_1 \vdash u : B' \mid \Delta_1; ((\beta : A)^{\leq 1} \cup (\gamma : B)^{\leq 1})$. Considerar $q = \mathtt{LCC}'\langle c\rangle[\alpha \setminus \beta]$ y $q' = \mathtt{LCC}'\langle c[\alpha \setminus \beta]\rangle$ donde $\mathtt{LCC}' = [\gamma]\,\mathtt{LTC}'$. Entonces, por (NAME) y (REN) se obtiene $\pi_q \rhd_{\mathcal{E}} \Gamma_0; (x : B')^{\leq 1} \vdash q \mid \Delta_0; (\beta : A \cup \gamma : B)$. Por *h.i.* existe una derivación $\pi_{q'} \rhd_{\mathcal{E}} \Gamma_0; (x : B')^{\leq 1} \vdash q' \mid \Delta_0; (\beta : A \cup \gamma : B)$. Esta derivación necesariamente termina con la regla (NAME). Luego, se tiene la premisa $\pi_{\mathtt{LTC}'\langle c[\alpha\setminus\beta]\rangle} \rhd_{\mathcal{E}} \Gamma_0; (x : B')^{\leq 1} \vdash \mathtt{LTC}'\langle c[\alpha \setminus \beta]\rangle : B \mid \Delta_0; (\beta : A \cup (\gamma : B)^{\leq 1})$. Finalmente, se concluye por (SUBS) con π_u y luego (NAME), $\pi_{o'} \rhd_{\mathcal{E}} \Gamma \vdash o' \mid \Delta$.
  - $\mathtt{LCC} = \mathtt{LCC}'[\![\gamma \setminus^{\delta} s]\!]$. Luego, por (REPL), $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta^* = \Delta_0 \cup \Delta_1; \delta : B$ con $\pi_{\mathtt{LCC}'\langle c\rangle} \rhd_{\mathcal{E}} \Gamma_0 \vdash \mathtt{LCC}'\langle c\rangle \mid \Delta_0; (\alpha : A)^{\leq 1}; (\gamma : S \to B)^{\leq 1}; ((\beta : A)^{\leq 1} \cup (\delta : B)^{\leq 1})$ y $\pi_s \rhd_{\mathcal{E}} \Gamma_1 \vdash s : S \mid \Delta_1; ((\beta : A)^{\leq 1} \cup (\delta : B)^{\leq 1})$. Por (REN), se obtiene $\pi_{\mathtt{LCC}'\langle c\rangle[\alpha\setminus\beta]} \rhd_{\mathcal{E}} \Gamma_0 \vdash \mathtt{LCC}'\langle c\rangle[\alpha \setminus \beta] \mid \Delta_0; (\beta : A \cup (\delta : B)^{\leq 1}); (\gamma : S \to B)^{\leq 1}$. Entonces, por *h.i.* sobre $\mathtt{LCC}'$, existe $\pi_{\mathtt{LCC}'\langle c[\alpha\setminus\beta]\rangle} \rhd_{\mathcal{E}} \Gamma_0 \vdash \mathtt{LCC}'\langle c[\alpha \setminus \beta]\rangle \mid \Delta_0; (\beta : A \cup (\delta : B)^{\leq 1}); (\gamma : S \to B)^{\leq 1}$. Finalmente, se concluye por (REPL) con π_s, $\pi_{o'} \rhd_{\mathcal{E}} \Gamma \vdash o' \mid \Delta$.

- $\mathtt{LCC} = \mathtt{LCC}'[\gamma \setminus \delta]$. Luego, por (REN), se tiene $\Delta^* = \Delta'; \delta : B$ con la derivación $\pi_{\mathtt{LCC}'\langle c\rangle} \rhd_{\mathcal{E}} \Gamma \vdash \mathtt{LCC}'\langle c\rangle \mid \Delta'; (\alpha : A)^{\leq 1}; ((\beta : A)^{\leq 1} \cup (\delta : B)^{\leq 1}); (\gamma : B)^{\leq 1}$. Por (REN), se obtiene $\pi_{\mathtt{LCC}'\langle c\rangle[\alpha\setminus\beta]} \rhd_{\mathcal{E}} \Gamma \vdash \mathtt{LCC}'\langle c\rangle[\alpha \setminus \beta] \mid \Delta'; (\beta : A \cup (\delta : B)^{\leq 1}); (\gamma : B)^{\leq 1}$. Más aún, aplicando la *h.i.* sobre $\mathtt{LCC}'$, se garantiza que existe una derivación $\pi_{\mathtt{LCC}'\langle c[\alpha\setminus\beta]\rangle} \rhd_{\mathcal{E}} \Gamma \vdash \mathtt{LCC}'\langle c[\alpha \setminus \beta]\rangle \mid \Delta'; (\beta : A \cup (\delta : B)^{\leq 1}); (\gamma : B)^{\leq 1}$. Finalmente, se concluye por (REN) con γ y δ, $\pi_{o'} \rhd_{\mathcal{E}} \Gamma \vdash o' \mid \Delta$.

- $\simeq_{\mathtt{pp}}$. Luego, $o = [\alpha']\,\lambda x.\mu\alpha.[\beta']\,\lambda y.\mu\beta.c$ y $o' = [\beta']\,\lambda y.\mu\beta.[\alpha']\,\lambda x.\mu\alpha.c$ con $\beta \neq \alpha'$ y $\alpha \neq \beta'$. Más aún, de π se tiene el contexto $\Delta = \Delta^*; \alpha' : A \to B \cup \beta' : A' \to B'$ con $\pi_c \rhd_{\mathcal{E}} \Gamma; (x : A)^{\leq 1}; (y : A')^{\leq 1} \vdash c \mid \Delta^*; (\alpha' : A \to B)^{\leq 1} \cup (\beta' : A' \to B')^{\leq 1}; (\alpha : B)^{\leq 1}; (\beta : B')^{\leq 1}$. Por (CONT), (ABS) y (NAME) sobre α, x y α' respectivamente, se obtiene la derivación $\pi_c \rhd_{\mathcal{E}} \Gamma; (y : A')^{\leq 1} \vdash [\alpha']\,\lambda x.\mu\alpha.c \mid \Delta^*; \alpha' : A \to B \cup (\beta' : A' \to B')^{\leq 1}; (\beta : B')^{\leq 1}$. Del mismo modo, aplicando la misma secuencia de reglas con β, y y β' respectivamente se concluye, $\pi' \rhd_{\mathcal{E}} \Gamma \vdash o' : T \mid \Delta$. El caso simétrico es análogo.

- $\simeq_{\rho}$. Luego, $o = [\beta]\,\mu\alpha.c$ y $o' = c[\alpha \setminus \beta]$. Más aún, de π por (NAME) se tiene $\Delta = \Delta^*; \beta : A$ con $\pi_{\mu\alpha.c} \rhd_{\mathcal{E}} \Gamma \vdash \mu\alpha.c : A \mid \Delta^*; (\beta : A)^{\leq 1}$ y, por (CONT), $\pi_c \rhd_{\mathcal{E}} \Gamma \vdash c \mid \Delta^*; (\alpha : A)^{\leq 1}; (\beta : A)^{\leq 1}$. Se concluye por (REN), obteniendo $\pi_{c[\alpha\setminus\beta]} \rhd_{\mathcal{E}} \Gamma \vdash c[\alpha \setminus \beta] \mid \Delta$. Para el caso simétrico, se parte de $\pi_{c[\alpha\setminus\beta]}$ descomponiendo por (REN) para obtener π_c. Finalmente, se concluye por (CONT) seguido de (NAME).

- $\mathtt{O} = \mathtt{T}\,u$. Luego, $o = t\,u$ y $o' = t'\,u$ con $t \simeq t'$. Más aún, por (APP) se tienen $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta = \Delta_0 \cup \Delta_1$ con $\pi_t \rhd_{\mathcal{E}} \Gamma_0 \vdash t : A \to T \mid \Delta_0$ y $\pi_u \rhd_{\mathcal{E}} \Gamma_1 \vdash u : A \mid \Delta_1$. Por *h.i.* existe $\pi_{t'} \rhd_{\mathcal{E}} \Gamma_0 \vdash t' : A \to T \mid \Delta_0'$. Finalmente, se concluye por (APP), $\pi' \rhd_{\mathcal{E}} \Gamma \vdash o' : T \mid \Delta'$ con $\Delta' = \Delta_0' \cup \Delta_1$. El caso simétrico es análogo.
- $\mathtt{O} = t\,\mathtt{T}$. Luego, $o = t\,u$ y $o' = t\,u'$ con $u \simeq u'$. Más aún, por (APP) se tienen $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta = \Delta_0 \cup \Delta_1$ con $\pi_t \rhd_{\mathcal{E}} \Gamma_0 \vdash t : A \to T \mid \Delta_0$ y $\pi_u \rhd_{\mathcal{E}} \Gamma_1 \vdash u : A \mid \Delta_1$. Por *h.i.* existe $\pi_{u'} \rhd_{\mathcal{E}} \Gamma_1 \vdash u' : A \mid \Delta_1'$. Finalmente, se concluye por (APP), $\pi' \rhd_{\mathcal{E}} \Gamma \vdash o' : T \mid \Delta'$ con $\Delta' = \Delta_0 \cup \Delta_1'$. El caso simétrico es análogo.
- $\mathtt{O} = \lambda x.\mathtt{T}$. Luego, $o = \lambda x.t$ y $o' = \lambda x.t'$ con $t \simeq t'$. Más aún, por (ABS) se tiene $T = A \to B$ con $\pi_t \rhd_{\mathcal{E}} \Gamma; (x : A)^{\leq 1} \vdash t : B \mid \Delta$. Por *h.i.* existe $\pi_{t'} \rhd_{\mathcal{E}} \Gamma; (x : A)^{\leq 1} \vdash t' : B \mid \Delta'$. Finalmente, se concluye por (ABS), $\pi' \rhd_{\mathcal{E}} \Gamma \vdash o' : T \mid \Delta'$. El caso simétrico es análogo.
- $\mathtt{O} = \mu\alpha.\mathtt{C}$. Luego, $o = \mu\alpha.c$ y $o' = \mu\alpha.c'$ con $c \simeq c'$. Más aún, por (CONT) se tiene $\pi_c \rhd_{\mathcal{E}} \Gamma \vdash c \mid \Delta; (\alpha : T)^{\leq 1}$. Por *h.i.* existe $\pi_{c'} \rhd_{\mathcal{E}} \Gamma \vdash c' \mid \Delta'; (\alpha : T)^{\leq 1}$. Se concluye por (CONT), $\pi' \rhd_{\mathcal{E}} \Gamma \vdash o' : T \mid \Delta'$. El caso simétrico es análogo.
- $\mathtt{O} = \mathtt{T}[x \setminus u]$. Luego, $o = t[x \setminus u]$ y $o' = t'[x \setminus u]$ con $t \simeq t'$. Más aún, por (SUBS) se tienen $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta = \Delta_0 \cup \Delta_1$ con $\pi_t \rhd_{\mathcal{E}} \Gamma_0; (x : A)^{\leq 1} \vdash t : T \mid \Delta_0$ y $\pi_u \rhd_{\mathcal{E}} \Gamma_1 \vdash u : A \mid \Delta_1$. Por *h.i.* existe $\pi_{t'} \rhd_{\mathcal{E}} \Gamma_0; (x : A)^{\leq 1} \vdash t' : T \mid \Delta_0'$. Finalmente, se concluye por (SUBS), $\pi' \rhd_{\mathcal{E}} \Gamma \vdash o' : T \mid \Delta'$ con $\Delta' = \Delta_0' \cup \Delta_1$. El caso simétrico es análogo.
- $\mathtt{O} = t[x \setminus \mathtt{T}]$. Luego, $o = t[x \setminus u]$ y $o' = t[x \setminus u']$ con $u \simeq u'$. Más aún, por (SUBS) se tienen $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta = \Delta_0 \cup \Delta_1$ con $\pi_t \rhd_{\mathcal{E}} \Gamma_0; (x : A)^{\leq 1} \vdash t : T \mid \Delta_0$ y $\pi_u \rhd_{\mathcal{E}} \Gamma_1 \vdash u : A \mid \Delta_1$. Por *h.i.* existe $\pi_{u'} \rhd_{\mathcal{E}} \Gamma_1 \vdash u' : A \mid \Delta_1'$. Se concluye por (SUBS), $\pi' \rhd_{\mathcal{E}} \Gamma \vdash o' : T \mid \Delta'$ con $\Delta' = \Delta_0 \cup \Delta_1'$. El caso simétrico es análogo.
- $\mathtt{O} = [\alpha]\,\mathtt{T}$. Luego, $o = [\alpha]\,t$ y $o' = [\alpha]\,t'$ con $t \simeq t'$. Más aún, por (NAME) se tiene $\Delta = \Delta_0; \alpha : A$ con $\pi_t \rhd_{\mathcal{E}} \Gamma \vdash t : A \mid \Delta_0; (\alpha : A)^{\leq 1}$. Por *h.i.* existe $\pi_{t'} \rhd_{\mathcal{E}} \Gamma \vdash t' : A \mid \Delta_0'; (\alpha : A)^{\leq 1}$. Se concluye por (NAME), $\pi' \rhd_{\mathcal{E}} \Gamma \vdash o' : T \mid \Delta'$ con $\Delta' = \Delta_0'; \alpha : A$. El caso simétrico es análogo.
- $\mathtt{O} = \mathtt{C}[\![\alpha \setminus^{\alpha'} s]\!]$. Luego, $o = c[\![\alpha \setminus^{\alpha'} s]\!]$ y $o' = c'[\![\alpha \setminus^{\alpha'} s]\!]$ con $c \simeq c'$. Más aún, por (REPL) se tienen $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta = \Delta_0 \cup \Delta_1; \alpha' : A$ con $\pi_c \rhd_{\mathcal{E}} \Gamma_0 \vdash c \mid \Delta_0; (\alpha : S \to A)^{\leq 1}; (\alpha' : A)^{\leq 1}$ y $\pi_s \rhd_{\mathcal{E}} \Gamma_1 \vdash s : A \mid \Delta_1; (\alpha' : A)^{\leq 1}$. Por *h.i.* existe $\pi_{c'} \rhd_{\mathcal{E}} \Gamma_0 \vdash c' \mid \Delta_0'; (\alpha : S \to A)^{\leq 1}; (\alpha' : A)^{\leq 1}$. Se concluye por (REPL), $\pi' \rhd_{\mathcal{E}} \Gamma \vdash o' : T \mid \Delta'$ con $\Delta' = \Delta_0' \cup \Delta_1; \alpha' : A$. El caso simétrico es análogo.

- $\mathtt{O} = c[\![\alpha \backslash^{\alpha'} \mathtt{S}]\!]$. Luego, $o = c[\![\alpha \backslash^{\alpha'} s]\!]$ y $o' = c[\![\alpha \backslash^{\alpha'} s']\!]$ con $s \simeq s'$. Más aún, por (REPL) se tienen $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta = \Delta_0 \cup \Delta_1; \alpha' : A$ con $\pi_c \rhd_{\mathcal{E}} \Gamma_0 \vdash c \mid \Delta_0; (\alpha : S \to A)^{\leq 1}; (\alpha' : A)^{\leq 1}$ y $\pi_s \rhd_{\mathcal{E}} \Gamma_1 \vdash s : A \mid \Delta_1; (\alpha' : A)^{\leq 1}$. Por *h.i.* existe una derivación $\pi_{s'} \rhd_{\mathcal{E}} \Gamma_1 \vdash s' : A \mid \Delta_1'; (\alpha' : A)^{\leq 1}$. Se concluye por (REPL), $\pi' \rhd_{\mathcal{E}} \Gamma \vdash o' : T \mid \Delta'$ con $\Delta' = \Delta_0 \cup \Delta_1'; \alpha' : A$. El caso simétrico es análogo.
- $\mathtt{O} = \mathtt{C}[\alpha \setminus \beta]$. Luego, $o = c[\alpha \setminus \beta]$ y $o' = c'[\alpha \setminus \beta]$ con $c \simeq c'$. Más aún, por (REN), $\Delta = \Delta_0; \beta : A$ con $\pi_c \rhd_{\mathcal{E}} \Gamma \vdash c \mid \Delta_0; (\alpha : A)^{\leq 1}; (\beta : A)^{\leq 1}$. Por *h.i.* existe una derivación $\pi_{c'} \rhd_{\mathcal{E}} \Gamma \vdash c' \mid \Delta_0'; (\alpha : A)^{\leq 1}; (\beta : A)^{\leq 1}$. Finalmente, se concluye por (REN), $\pi' \rhd_{\mathcal{E}} \Gamma \vdash o' : T \mid \Delta'$ con $\Delta' = \Delta_0'; \beta : A$. El caso simétrico es análogo.
- $\mathtt{O} = \mathtt{T} \cdot s$. Luego, $o = t \cdot s$ y $o' = t' \cdot s$ con $t \simeq t'$. Más aún, por (STK) se tienen $T = A \cdot S$, $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta = \Delta_0 \cup \Delta_1$ con $\pi_t \rhd_{\mathcal{E}} \Gamma_0 \vdash t : A \mid \Delta_0$ y $\pi_s \rhd_{\mathcal{E}} \Gamma_1 \vdash s : S \mid \Delta_1$. Por *h.i.* existe $\pi_{t'} \rhd_{\mathcal{E}} \Gamma_0 \vdash t' : A \mid \Delta_0'$. Finalmente, se concluye por (STK), $\pi' \rhd_{\mathcal{E}} \Gamma \vdash o' : T \mid \Delta'$ con $\Delta' = \Delta_0' \cup \Delta_1$. El caso simétrico es análogo.
- $\mathtt{O} = t \cdot \mathtt{S}$. Luego, $o = t \cdot s$ y $o' = t \cdot s'$ con $s \simeq s'$. Más aún, por (STK) se tienen $T = A \cdot S$, $\Gamma = \Gamma_0 \cup \Gamma_1$ y $\Delta = \Delta_0 \cup \Delta_1$ con $\pi_t \rhd_{\mathcal{E}} \Gamma_0 \vdash t : A \mid \Delta_0$ y $\pi_s \rhd_{\mathcal{E}} \Gamma_1 \vdash s : S \mid \Delta_1$. Por *h.i.* existe $\pi_{s'} \rhd_{\mathcal{E}} \Gamma_1 \vdash s' : S \mid \Delta_1'$. Finalmente, se concluye por (STK), $\pi' \rhd_{\mathcal{E}} \Gamma' \vdash o' : T \mid \Delta'$ con $\Delta' = \Delta_0 \cup \Delta_1'$. El caso simétrico es análogo.

□

Dos resultados de correspondencia

Para demostrar la correción de $\simeq_\sigma$ respecto a $\simeq_{\mathtt{er}}$ es preciso enunciar anteriormente una serie de resultados auxiliares.

Lema C.0.47. *Sean $o, o' \in \mathbb{O}_{\Lambda M}$ tal que $o \simeq_* o'$ con $\simeq_*$ una regla de Fig. 4.13 o $\simeq_{\mathtt{ren}}$. Si $o = \mathtt{LXC}\langle c \rangle$ con $\alpha \notin \mathtt{LXC}$, $\mathsf{fc}(\alpha', \mathtt{LXC})$ y $\mathsf{fc}(s, \mathtt{LXC})$, luego $o' = \mathtt{LXC}'\langle c' \rangle$ tal que $\alpha \notin \mathtt{LXC}'$, $\mathsf{fc}(\alpha', \mathtt{LXC}')$, $\mathsf{fc}(s, \mathtt{LXC}')$ y $\mathfrak{C}(\mathtt{LXC}\langle c[\![\alpha \backslash^{\alpha'} s]\!]\rangle) \simeq_{\mathtt{er}} \mathfrak{C}(\mathtt{LXC}'\langle c'[\![\alpha \backslash^{\alpha'} s]\!]\rangle)$.*

Demostración. Por análisis de casos sobre $\mathtt{LXC}$.

- $\mathtt{LXC} = \boxdot$. Luego, $o = c$. Hay cinco posibles reglas para comandos:
  - $\simeq_{\mathtt{exrepl}}$. Luego, $o = \mathtt{LCC}\langle c' \rangle[\![\gamma \backslash^{\delta} s']\!]$ y $o' = \mathtt{LCC}\langle c'[\![\gamma \backslash^{\delta} s']\!]\rangle$ con $\gamma \notin \mathtt{LCC}$, $\mathsf{fc}(\delta, \mathtt{LCC})$ y $\mathsf{fc}(s', \mathtt{LCC})$. Por α-conversión se asumen también $\mathsf{fc}(\alpha', \mathtt{LCC})$ y $\mathsf{fc}(s, \mathtt{LCC})$. Luego, se tienen cuatro casos posibles:
    1. $o[\![\alpha \backslash^{\alpha'} s]\!] \to_{\mathtt{p}} \mathtt{LCC}\langle c \rangle[\![\gamma \backslash^{\alpha'} s' \cdot s]\!]$ (*i.e.* $\alpha = \delta$, $\delta \notin c$, $\delta \notin \mathtt{LCC}$ y $\delta \notin s'$). Luego, $\mathfrak{C}(o[\![\alpha \backslash^{\alpha'} s]\!]) = \mathfrak{C}(\mathtt{LCC}\langle c \rangle[\![\gamma \backslash^{\alpha'} s' \cdot s]\!])$. Más aún, se tiene $o'[\![\alpha \backslash^{\alpha'} s]\!] \to_{\mathtt{p}} \mathtt{LCC}\langle c[\![\gamma \backslash^{\alpha'} s' \cdot s]\!]\rangle$, por lo que $\mathfrak{C}(o'[\![\alpha \backslash^{\alpha'} s]\!]) = \mathfrak{C}(\mathtt{LCC}\langle c[\![\gamma \backslash^{\alpha'} s' \cdot s]\!]\rangle)$. Luego, por Lem. 4.6.2 (2), se tiene $\mathfrak{C}(\mathtt{LCC}\langle c \rangle[\![\gamma \backslash^{\alpha'} s' \cdot s]\!]) \simeq \mathfrak{C}(\mathtt{LCC}\langle c[\![\gamma \backslash^{\alpha'} s' \cdot s]\!]\rangle)$. Se concluye dado que $\simeq \subseteq \simeq_{\mathtt{er}}$, con $\mathtt{LXC}' = \boxdot$ y $c' = o'$.
    2. $o[\![\alpha \backslash^{\alpha'} s]\!] \to_{\mathtt{NPW}} \mathtt{LCC}'\langle c \rangle[\![\gamma \backslash^{\delta} s']\!]$ (*i.e.* $\mathsf{fn}_\alpha(o) = 1$, $\alpha \neq \delta$, $\alpha \notin c$ y $\alpha \notin s'$). Luego, $\mathfrak{C}(o[\![\alpha \backslash^{\alpha'} s]\!]) = \mathfrak{C}(\mathtt{LCC}'\langle c \rangle[\![\gamma \backslash^{\delta} s']\!])$. Más aún, también se tiene $o'[\![\alpha \backslash^{\alpha'} s]\!] \to_{\mathtt{NPW}} \mathtt{LCC}'\langle c[\![\gamma \backslash^{\delta} s']\!]\rangle$, por lo que $\mathfrak{C}(o'[\![\alpha \backslash^{\alpha'} s]\!]) = \mathfrak{C}(\mathtt{LCC}'\langle c[\![\gamma \backslash^{\delta} s']\!]\rangle)$. Por Lem. 4.6.2 (2), se tiene $\mathfrak{C}(\mathtt{LCC}'\langle c \rangle[\![\gamma \backslash^{\delta} s']\!]) \simeq \mathfrak{C}(\mathtt{LCC}'\langle c[\![\gamma \backslash^{\delta} s']\!]\rangle)$. Se concluye dado que $\simeq \subseteq \simeq_{\mathtt{er}}$, con $\mathtt{LXC}' = \boxdot$ y $c' = o'$.
    3. $o[\![\alpha \backslash^{\alpha'} s]\!] \to_{\mathtt{NPW}} \mathtt{LCC}\langle c' \rangle[\![\gamma \backslash^{\delta} s']\!]$ (*i.e.* $\mathsf{fn}_\alpha(o) = 1$, $\alpha \neq \delta$, $\alpha \notin \mathtt{LCC}$ y $\alpha \notin s'$). Este caso es análogo al anterior.
    4. De lo contrario, $\mathfrak{C}(o[\![\alpha \backslash^{\alpha'} s]\!]) = o[\![\alpha \backslash^{\alpha'} \mathfrak{C}(s)]\!] \simeq_{\mathtt{er}} o'[\![\alpha \backslash^{\alpha'} \mathfrak{C}(s)]\!] = \mathfrak{C}(o'[\![\alpha \backslash^{\alpha'} s]\!])$, pues $o \simeq_{\mathtt{er}} o'$ implica $\mathfrak{C}(o) = o$ y $\mathfrak{C}(o') = o'$. Se concluye con $\mathtt{LXC}' = \boxdot$ y $c' = o'$.

- $\simeq_{\mathtt{exren}}$. Luego, $o = \mathtt{LCC}\langle c\rangle[\gamma \backslash \delta]$ y $o' = \mathtt{LCC}\langle c[\gamma \setminus \delta]\rangle$ con $\gamma \notin \mathtt{LCC}$ y $\mathsf{fc}(\delta, \mathtt{LCC})$. Por α-conversión se asumen también $\mathsf{fc}(\alpha', \mathtt{LCC})$ y $\mathsf{fc}(s, \mathtt{LCC})$. Luego, se tienen cuatro casos posibles:
  1. $o[\![\alpha \backslash^{\alpha'} s]\!] \to_{\mathtt{W}} \mathtt{LCC}\langle c\rangle[\![\gamma \backslash^{\alpha} s]\!][\alpha \setminus \alpha']$ (*i.e.* $\alpha = \delta$, $\delta \notin c$ y $\delta \notin \mathtt{LCC}$). Luego, $\mathfrak{C}(o[\![\alpha \backslash^{\alpha'} s]\!]) = \mathfrak{C}(\mathtt{LCC}\langle c\rangle[\![\gamma \backslash^{\alpha} s]\!][\alpha \setminus \alpha'])$. Más aún, también se tiene $o'[\![\alpha \backslash^{\alpha'} s]\!] \to_{\mathtt{W}} \mathtt{LCC}\langle c[\![\gamma \backslash^{\alpha} s]\!][\alpha \setminus \alpha']\rangle$, por lo que $\mathfrak{C}(o'[\![\alpha \backslash^{\alpha'} s]\!]) = \mathfrak{C}(\mathtt{LCC}\langle c[\![\gamma \backslash^{\alpha} s]\!][\alpha \setminus \alpha']\rangle)$. Luego, por Lem. 4.6.2 (2), se tiene $\mathfrak{C}(\mathtt{LCC}\langle c\rangle[\![\gamma \backslash^{\alpha} s]\!][\alpha \setminus \alpha']) \simeq \mathfrak{C}(\mathtt{LCC}\langle c[\![\gamma \backslash^{\alpha} s]\!][\alpha \setminus \alpha']\rangle)$. Se concluye dado que $\simeq \subseteq \simeq_{\mathtt{er}}$, con $\mathtt{LXC}' = \boxdot$ y $c' = o'$.
  2. $o[\![\alpha \backslash^{\alpha'} s]\!] \to_{\mathtt{NPW}} \mathtt{LCC}'\langle c\rangle[\gamma \setminus \delta]$ (*i.e.* $\mathsf{fn}_{\alpha}(o) = 1$, $\alpha \neq \delta$ y $\alpha \notin c$). Luego, $\mathfrak{C}(o[\![\alpha \backslash^{\alpha'} s]\!]) = \mathfrak{C}(\mathtt{LCC}'\langle c\rangle[\gamma \setminus \delta])$. Más aún, también se tiene $o'[\![\alpha \backslash^{\alpha'} s]\!] \to_{\mathtt{NPW}} \mathtt{LCC}'\langle c[\gamma \setminus \delta]\rangle$, por lo que $\mathfrak{C}(o'[\![\alpha \backslash^{\alpha'} s]\!]) = \mathfrak{C}(\mathtt{LCC}'\langle c[\gamma \setminus \delta]\rangle)$. Por Lem. 4.6.2 (2), se tiene $\mathfrak{C}(\mathtt{LCC}'\langle c\rangle[\gamma \setminus \delta]) \simeq \mathfrak{C}(\mathtt{LCC}'\langle c[\gamma \setminus \delta]\rangle)$. Se concluye dado que $\simeq \subseteq \simeq_{\mathtt{er}}$, con $\mathtt{LXC}' = \boxdot$ y $c' = o'$.
  3. $o[\![\alpha \backslash^{\alpha'} s]\!] \to_{\mathtt{NPW}} \mathtt{LCC}\langle c'\rangle[\gamma \setminus \delta]$ (*i.e.* $\mathsf{fn}_{\alpha}(o) = 1$, $\alpha \neq \delta$ y $\alpha \notin \mathtt{LCC}$). Este caso es análogo al anterior.
  4. De lo contrario, $\mathfrak{C}(o[\![\alpha \backslash^{\alpha'} s]\!]) = o[\![\alpha \backslash^{\alpha'} \mathfrak{C}(s)]\!] \simeq_{\mathtt{er}} o'[\![\alpha \backslash^{\alpha'} \mathfrak{C}(s)]\!] = \mathfrak{C}(o'[\![\alpha \backslash^{\alpha'} s]\!])$, pues $o \simeq_{\mathtt{er}} o'$ implica $\mathfrak{C}(o) = o$ y $\mathfrak{C}(o') = o'$. Se concluye con $\mathtt{LXC}' = \boxdot$ y $c' = o'$.
- $\simeq_{\mathtt{pp}}$. Luego, $o = [\gamma']\,\lambda x.\mu\gamma.[\delta']\,\lambda y.\mu\delta.c$ y $o' = [\delta']\,\lambda y.\mu\delta.[\gamma']\,\lambda x.\mu\gamma.c$ con $\delta \neq \gamma'$ y $\gamma \neq \delta'$. Por α-conversión se asumen también $x \notin s$, $y \notin s$, $\gamma \neq \alpha'$, $\gamma \notin s$, $\delta \neq \alpha'$ y $\delta \notin s$. Luego, se tienen cuatro casos posibles.
  1. $o[\![\alpha \backslash^{\alpha'} s]\!] \to_{\mathtt{N}} [\alpha']\,(\lambda x.\mu\gamma.[\delta']\,\lambda y.\mu\delta.c) :: s$ (*i.e.* $\alpha = \gamma'$, $\alpha \neq \delta'$ y $\gamma' \notin c$). Asumir $s = u \cdot s'$ (el caso $s = u$ es ligeramente más simple) y considerar nombres frescos γ'', δ''. Luego, $\mathfrak{C}(o[\![\alpha \backslash^{\alpha'} s]\!]) = \mathfrak{C}([\alpha']\,(\mu\gamma''.([\delta']\,\lambda y.\mu\delta.c)[\![\gamma \backslash^{\gamma''} s']\!])[x \setminus u])$. Del mismo modo, $\mathfrak{C}(o'[\![\alpha \backslash^{\alpha'} s]\!]) = \mathfrak{C}([\delta']\,\lambda y.\mu\delta.[\alpha']\,(\mu\gamma''.c'[\![\gamma \backslash^{\gamma''} s']\!])[x \setminus u])$. En el caso $s = u$ simplemente se evita el replacement explícito. Luego, por Lem. 4.6.2, se tiene por un lado $\mathfrak{C}(o[\![\alpha \backslash^{\alpha'} s]\!]) \simeq \mathfrak{C}([\alpha']\,\mu\gamma''.[\delta']\,(\lambda y.\mu\delta.c'[\![\gamma \backslash^{\gamma''} s']\!])[x \setminus u])$ y por otro $\mathfrak{C}(o'[\![\alpha \backslash^{\alpha'} s]\!]) \simeq \mathfrak{C}([\delta']\,(\lambda y.\mu\delta.[\alpha']\,\mu\gamma''.c[\![\gamma \backslash^{\gamma''} s']\!])[x \setminus u])$, apelando al ítem (1) del lema para posicionar la sustitución explícita y al ítem (2) para el replacement. Finalmente, aplicando dos veces $\simeq_{\rho}$ y Lem. 4.6.2 (2) para posicionar adecuadamente el renaming explícito introducido por la regla, se tiene

$$\begin{array}{rcl}
\mathfrak{C}(o[\![\alpha \backslash^{\alpha'} s]\!]) & \simeq & \mathfrak{C}([\alpha']\,\mu\gamma''.[\delta']\,(\lambda y.\mu\delta.c'[\![\gamma \backslash^{\gamma''} s']\!])[x \setminus u]) \\
& = & [\alpha']\,\mu\gamma''.[\delta']\,(\lambda y.\mu\delta.\mathfrak{C}(c'[\![\gamma \backslash^{\gamma''} s']\!]))[x \setminus \mathfrak{C}(u)] \\
& \simeq_{\rho} & ([\delta']\,(\lambda y.\mu\delta.\mathfrak{C}(c'[\![\gamma \backslash^{\gamma''} s']\!]))[x \setminus \mathfrak{C}(u)])[\gamma'' \setminus \alpha'] \\
& \simeq & [\delta']\,(\lambda y.\mu\delta.\mathfrak{C}(c'[\![\gamma \backslash^{\gamma''} s']\!])[\gamma'' \setminus \alpha'])[x \setminus \mathfrak{C}(u)] \\
& \simeq_{\rho} & [\delta']\,(\lambda y.\mu\delta.[\alpha']\,\mu\gamma''.\mathfrak{C}(c'[\![\gamma \backslash^{\gamma''} s']\!]))[x \setminus \mathfrak{C}(u)] \\
& = & \mathfrak{C}([\delta']\,(\lambda y.\mu\delta.[\alpha']\,\mu\gamma''.c'[\![\gamma \backslash^{\gamma''} s']\!])[x \setminus u]) \\
& \simeq & \mathfrak{C}(o'[\![\alpha \backslash^{\alpha'} s]\!])
\end{array}$$

     Se concluye dado que $\simeq \subseteq \simeq_{\mathtt{er}}$, con $\mathtt{LXC}' = \boxdot$ y $c' = o'$.
  2. $o[\![\alpha \backslash^{\alpha'} s]\!] \to_{\mathtt{N}} [\gamma']\,\lambda x.\mu\gamma.[\alpha']\,(\lambda y.\mu\delta.c) :: s$ (*i.e.* $\alpha \neq \gamma'$, $\alpha = \delta'$ y $\delta' \notin c$). Este caso es análogo al anterior.
  3. $o[\![\alpha \backslash^{\alpha'} s]\!] \to_{\mathtt{NPW}} [\gamma']\,\lambda x.\mu\gamma.[\delta']\,\lambda y.\mu\delta.c'$ (*i.e.* $\mathsf{fn}_{\alpha}(o) = 1$, $\alpha \neq \gamma'$ y $\alpha \neq \delta'$). Luego, $\mathfrak{C}(o[\![\alpha \backslash^{\alpha'} s]\!]) = [\gamma']\,\lambda x.\mu\gamma.[\delta']\,\lambda y.\mu\delta.\mathfrak{C}(c') \simeq_{\mathtt{pp}} [\delta']\,\lambda y.\mu\delta.[\gamma']\,\lambda x.\mu\gamma.\mathfrak{C}(c') = \mathfrak{C}(o'[\![\alpha \backslash^{\alpha'} s]\!])$. Se concluye con $\mathtt{LXC}' = \boxdot$ y $c' = o'$.
  4. De lo contrario, $\mathfrak{C}(o[\![\alpha \backslash^{\alpha'} s]\!]) = o[\![\alpha \backslash^{\alpha'} \mathfrak{C}(s)]\!] \simeq_{\mathtt{er}} o'[\![\alpha \backslash^{\alpha'} \mathfrak{C}(s)]\!] = \mathfrak{C}(o'[\![\alpha \backslash^{\alpha'} s]\!])$, pues $o \simeq_{\mathtt{er}} o'$ implica $\mathfrak{C}(o) = o$ y $\mathfrak{C}(o') = o'$. Se concluye con $\mathtt{LXC}' = \boxdot$ y $c' = o'$.
- $\simeq_{\rho}$. Luego, $o = [\delta]\,\mu\gamma.c$ y $o' = c[\gamma \setminus \delta]$. Por α-conversión se asumen $\gamma \neq \alpha'$ y $\gamma \notin s$. Luego, se tienen tres casos posibles.

1. $o[\![\alpha \setminus^{\alpha'} s]\!] \to_{\mathtt{N}} [\alpha'](\mu\gamma.c) :: s$ (*i.e.* $\alpha = \delta$ y $\delta \notin c$). Luego, se tiene $\mathfrak{C}(o[\![\alpha \setminus^{\alpha'} s]\!]) = [\alpha']\,\mathfrak{C}((\mu\gamma.c) :: s) = [\alpha']\,\mu\alpha.\mathfrak{C}(c[\![\gamma \setminus^{\alpha} s]\!])$. Notar que α es fresca en $(\mu\gamma.c) :: s$. Más aún, se tiene también $o'[\![\alpha \setminus^{\alpha'} s]\!] \to_{\mathtt{W}} c[\![\gamma \setminus^{\alpha} s]\!][\alpha \setminus \alpha']$. Se concluye por $\simeq_\rho$ con $\mathtt{LXC}' = \boxdot$ y $c' = o'$, dado que $\mathfrak{C}(o[\![\alpha \setminus^{\alpha'} s]\!]) \simeq_\rho \mathfrak{C}(c[\![\gamma \setminus^{\alpha} s]\!])[\alpha \setminus \alpha'] = \mathfrak{C}(c[\![\gamma \setminus^{\alpha} s]\!][\alpha \setminus \alpha']) = \mathfrak{C}(o'[\![\alpha \setminus^{\alpha'} s]\!])$
2. $o[\![\alpha \setminus^{\alpha'} s]\!] \to_{\mathtt{NPW}} [\delta]\,\mu\gamma.c'$ (*i.e.* $\mathsf{fn}_\alpha(o) = 1$ y $\alpha \neq \delta$). Luego, $\mathfrak{C}(o[\![\alpha \setminus^{\alpha'} s]\!]) = [\delta]\,\mu\gamma.\mathfrak{C}(c') \simeq_\rho \mathfrak{C}(c')[\gamma \setminus \delta] = \mathfrak{C}(o'[\![\alpha \setminus^{\alpha'} s]\!])$. Se concluye con $\mathtt{LXC}' = \boxdot$ y $c' = o'$.
3. De lo contrario, $\mathfrak{C}(o[\![\alpha \setminus^{\alpha'} s]\!]) = o[\![\alpha \setminus^{\alpha'} \mathfrak{C}(s)]\!] \simeq_{\mathtt{er}} o'[\![\alpha \setminus^{\alpha'} \mathfrak{C}(s)]\!] = \mathfrak{C}(o'[\![\alpha \setminus^{\alpha'} s]\!])$, pues $o \simeq_{\mathtt{er}} o'$ implica $\mathfrak{C}(o) = o$ y $\mathfrak{C}(o') = o'$. Se concluye con $\mathtt{LXC}' = \boxdot$ y $c' = o'$.

- $\simeq_{\mathtt{ren}}$. Luego, $o = c[\gamma \setminus \delta]$ y $o' = \{\gamma \setminus \delta\}c$. Por α-conversión se asumen $\gamma \neq \alpha'$ y $\gamma \notin s$. Luego, se tienen tres casos posibles.
  1. $o[\![\alpha \setminus^{\alpha'} s]\!] \to_{\mathtt{W}} c[\![\gamma \setminus^{\alpha} s]\!][\alpha \setminus \alpha']$ (*i.e.* $\alpha = \delta$ y $\delta \notin c$). Luego, $\mathfrak{C}(o[\![\alpha \setminus^{\alpha'} s]\!]) = \mathfrak{C}(c[\![\gamma \setminus^{\alpha} s]\!])[\alpha \setminus \alpha'] \simeq_{\mathtt{ren}} \{\alpha \setminus \alpha'\}\mathfrak{C}(c[\![\gamma \setminus^{\alpha} s]\!])$. Más aún, $\alpha \notin s$ por α-conversión y $\alpha = \delta \notin c$ por hipótesis de W. Luego, $\{\alpha \setminus \alpha'\}\mathfrak{C}(c[\![\gamma \setminus^{\alpha} s]\!]) = \mathfrak{C}(c[\![\gamma \setminus^{\alpha'} s]\!])$. Por otro lado, $\mathfrak{C}(o'[\![\alpha \setminus^{\alpha'} s]\!]) = \mathfrak{C}((\{\gamma \setminus \alpha\}c)[\![\alpha \setminus^{\alpha'} s]\!])$. Pero, como $\alpha = \delta \notin c$, esto es α-equivalente a $\mathfrak{C}(c[\![\gamma \setminus^{\alpha'} s]\!])$, por lo que se concluye con $\mathtt{LXC}' = \boxdot$ y $c' = o'$.
  2. $o[\![\alpha \setminus^{\alpha'} s]\!] \to_{\mathtt{NPW}} c'[\gamma \setminus \delta]$ (*i.e.* $\mathsf{fn}_\alpha(o) = 1$ y $\alpha \neq \delta$). Entonces, $\mathfrak{C}(o[\![\alpha \setminus^{\alpha'} s]\!]) = \mathfrak{C}(c')[\gamma \setminus \delta] \simeq_{\mathtt{ren}} \{\gamma \setminus \delta\}\mathfrak{C}(c') = \mathfrak{C}(o'[\![\alpha \setminus^{\alpha'} s]\!])$. Se concluye con $\mathtt{LXC}' = \boxdot$ y $c' = o'$.
  3. De lo contrario, $\mathfrak{C}(o[\![\alpha \setminus^{\alpha'} s]\!]) = o[\![\alpha \setminus^{\alpha'} \mathfrak{C}(s)]\!] \simeq_{\mathtt{er}} o'[\![\alpha \setminus^{\alpha'} \mathfrak{C}(s)]\!] = \mathfrak{C}(o'[\![\alpha \setminus^{\alpha'} s]\!])$, pues $o \simeq_{\mathtt{er}} o'$ implica $\mathfrak{C}(o) = o$ y $\mathfrak{C}(o') = o'$. Se concluye con $\mathtt{LXC}' = \boxdot$ y $c' = o'$.

- $\mathtt{LXC} = \mathtt{LTC}\,u$. Este caso es inmediato dado que no hay solapamiento con ninguna regla.

- $\mathtt{LXC} = \lambda x.\mathtt{LTC}$. Este caso es inmediato dado que no hay solapamiento con ninguna regla.

- $\mathtt{LXC} = \mu\gamma.\mathtt{LCC}$. Luego, la única regla aplicable es $\simeq_\theta$. Entonces, $\mathtt{LCC} = [\gamma]\,\mathtt{LTC}$ con $\gamma \notin c$ y $\gamma \notin \mathtt{LTC}$. Más aún, por hipótesis $\mathsf{fc}(\alpha', \mathtt{LXC})$ y $\mathsf{fc}(s, \mathtt{LXC})$, lo que implica $\gamma \neq \alpha'$ y $\gamma \notin s$. Se concluye por $\simeq_\theta$ con $\mathtt{LXC}' = \mathtt{LTC}$ y $c' = c$, $\mathfrak{C}(\mathtt{LXC}\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle) = \mu\gamma.[\gamma]\,\mathfrak{C}(\mathtt{LTC}\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle) \simeq_\theta \mathfrak{C}(\mathtt{LTC}\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle) = \mathfrak{C}(\mathtt{LXC}'\langle c'[\![\alpha \setminus^{\alpha'} s]\!]\rangle)$.

- $\mathtt{LXC} = \mathtt{LTC}[x \setminus u]$. Luego, la única regla aplicable es $\simeq_{\mathtt{exsubs}}$. Se tiene entonces $\mathtt{LTC}\langle c\rangle = \mathtt{LTT}\langle t\rangle$ con $x \notin \mathtt{LTT}$ y $\mathsf{fc}(u, \mathtt{LTT})$. Más aún, $\alpha \notin \mathtt{LXC}$ y $\mathsf{fc}(s, \mathtt{LXC})$ implican $\alpha \notin u$ y $x \notin \mathsf{fv}(s)$ respectivamente. Hay dos posibles casos:

  1. $t = \mathtt{LTC}_1\langle c\rangle$. Luego, $\mathtt{LXC} = \mathtt{LTT}\langle \mathtt{LTC}_1\rangle[x \setminus u]$. Más aún, se tiene $o' = \mathtt{LTT}\langle \mathtt{LTC}_1\langle c\rangle[x \setminus u]\rangle$: *i.e.* $\mathtt{LXC}' = \mathtt{LTT}\langle \mathtt{LTC}_1[x \setminus u]\rangle$ y $c' = c$. Por Lem. 4.6.2 (1), se tiene $\mathfrak{C}(\mathtt{LXC}\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle) = \mathfrak{C}(\mathtt{LTT}\langle \mathtt{LTC}_1\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle\rangle[x \setminus u]) \simeq \mathfrak{C}(\mathtt{LTT}\langle \mathtt{LTC}_1\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle[x \setminus u]\rangle) = \mathfrak{C}(\mathtt{LXC}'\langle c'[\![\alpha \setminus^{\alpha'} s]\!]\rangle)$. Se concluye dado que $\simeq \subseteq \simeq_{\mathtt{er}}$.
  2. $c = \mathtt{LCT}\langle t\rangle$ con $x \notin \mathtt{LCT}$ y $\mathsf{fc}(u, \mathtt{LCT})$. Luego, $o' = \mathtt{LTC}\langle \mathtt{LCT}\langle t[x \setminus u]\rangle\rangle$: *i.e.* $\mathtt{LXC}' = \mathtt{LTC}$ y $c' = \mathtt{LCT}\langle t[x \setminus u]\rangle$. Por Lem. 4.6.2 (1), $\mathfrak{C}(\mathtt{LXC}\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle) = \mathfrak{C}(\mathtt{LTC}\langle \mathtt{LCT}\langle t\rangle[\![\alpha \setminus^{\alpha'} s]\!]\rangle[x \setminus u]) \simeq \mathfrak{C}(\mathtt{LTC}\langle \mathtt{LCT}\langle t[x \setminus u]\rangle[\![\alpha \setminus^{\alpha'} s]\!]\rangle) = \mathfrak{C}(\mathtt{LXC}'\langle c'[\![\alpha \setminus^{\alpha'} s]\!]\rangle)$. Se concluye dado que $\simeq \subseteq \simeq_{\mathtt{er}}$.

- $\mathtt{LXC} = [\delta]\,\mathtt{LTC}$. Hay dos posibles reglas a analizar:

  - $\simeq_{\mathtt{pp}}$. Se tienen dos posibles instancias de la regla según la forma de LXC:
    1. $\mathtt{LXC} = [\delta]\,\lambda x.\mu\delta'.\boxdot$. Luego, $c = [\gamma]\,\lambda y.\mu\gamma'.c_0$ y $o' = [\gamma]\,\lambda y.\mu\gamma'.[\delta]\,\lambda x.\mu\delta'.c_0$. Más aún, $\alpha \notin \mathtt{LXC}$, $\mathsf{fc}(\alpha', \mathtt{LXC})$ y $\mathsf{fc}(s, \mathtt{LXC})$ implican $\alpha \neq \delta$, $\alpha' \neq \delta'$, $x \notin \mathsf{fv}(s)$ y $\delta' \notin \mathsf{fn}(s)$ respectivamente. Por α-conversión se asumen también $y \notin s$, $\gamma' \neq \alpha'$ y $\gamma' \notin \mathsf{fn}(s)$. Hay tres posibilidades para γ:

a) $\alpha = \gamma$ y $\gamma \notin c_0$. Luego, $c[\![\alpha \setminus^{\alpha'} s]\!] \to_{\mathtt{N}} [\alpha'](\lambda y.\mu\gamma'.c_0) :: s$. Asumir $s = u \cdot s'$ (si $s = u$ el análisis es ligeramente más simple) y considerar nombres frescos γ'', δ''. Luego, $\mathfrak{C}(\mathtt{LXC}\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle) = \mathfrak{C}([\delta]\,\lambda x.\mu\delta'.[\alpha']\,(\mu\gamma''.c_0[\![\gamma' \setminus^{\gamma''} s']\!])[y \setminus u])$. Por otro lado, considerar $\mathtt{LXC}' = \boxdot$ y $c' = o'$. Análogamente, se tiene $\mathfrak{C}(\mathtt{LXC}'\langle c'[\![\alpha \setminus^{\alpha'} s]\!]\rangle) = \mathfrak{C}([\alpha']\,(\mu\gamma''.([\delta]\,\lambda x.\mu\delta'.c_0)[\![\gamma' \setminus^{\gamma''} s']\!])[y \setminus u])$. Luego, por Lem. 4.6.2, se tiene por un lado $\mathfrak{C}(\mathtt{LXC}\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle) \simeq \mathfrak{C}([\delta]\,(\lambda x.\mu\delta'.[\alpha']\,\mu\gamma''.c_0[\![\gamma' \setminus^{\gamma''} s']\!])[y \setminus u])$ y por el otro $\mathfrak{C}(\mathtt{LXC}'\langle c'[\![\alpha \setminus^{\alpha'} s]\!]\rangle) \simeq \mathfrak{C}([\alpha']\,\mu\gamma''.[\delta]\,(\lambda x.\mu\delta'.c_0[\![\gamma' \setminus^{\gamma''} s']\!])[y \setminus u])$, apelando al ítem (1) del lema para posicionar la sustitución explícita y al ítem (2) para el replacement. Finalmente, aplicando dos veces $\simeq_\rho$ y Lem. 4.6.2 (2) para posicionar adecuadamente el renaming explícito introducido por la regla, se tiene

$$\begin{array}{rcl}
\mathfrak{C}(\mathtt{LXC}\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle) & \simeq & \mathfrak{C}([\delta]\,(\lambda x.\mu\delta'.[\alpha']\,\mu\gamma''.c_0[\![\gamma' \setminus^{\gamma''} s']\!])[y \setminus u]) \\
& = & [\delta]\,(\lambda x.\mu\delta'.[\alpha']\,\mu\gamma''.\mathfrak{C}(c_0[\![\gamma' \setminus^{\gamma''} s']\!]))[y \setminus \mathfrak{C}(u)] \\
& \simeq_\rho & [\delta]\,(\lambda x.\mu\delta'.\mathfrak{C}(c_0[\![\gamma' \setminus^{\gamma''} s']\!])[\gamma'' \setminus \alpha'])[y \setminus \mathfrak{C}(u)] \\
& \simeq & ([\delta]\,(\lambda x.\mu\delta'.\mathfrak{C}(c_0[\![\gamma' \setminus^{\gamma''} s']\!]))[y \setminus \mathfrak{C}(u)])[\gamma'' \setminus \alpha'] \\
& \simeq_\rho & [\alpha']\,\mu\gamma''.[\delta]\,(\lambda x.\mu\delta'.\mathfrak{C}(c_0[\![\gamma' \setminus^{\gamma''} s']\!]))[y \setminus \mathfrak{C}(u)] \\
& = & \mathfrak{C}([\alpha']\,\mu\gamma''.[\delta]\,(\lambda x.\mu\delta'.c_0[\![\gamma' \setminus^{\gamma''} s']\!])[y \setminus u]) \\
& \simeq & \mathfrak{C}(\mathtt{LXC}'\langle c'[\![\alpha \setminus^{\alpha'} s]\!]\rangle)
\end{array}$$

Se concluye dado que $\simeq \subseteq \simeq_{\mathtt{er}}$.

b) $\alpha = \gamma$ y $\gamma \in c_0$. Luego, $\mathfrak{C}(c[\![\alpha \setminus^{\alpha'} s]\!]) = c[\![\alpha \setminus^{\alpha'} \mathfrak{C}(s)]\!]$. Más aún, por Lem. 4.6.2 (2), se tiene $\mathfrak{C}(\mathtt{LXC}\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle) = \mathfrak{C}([\delta]\,\lambda x.\mu\delta'.c[\![\alpha \setminus^{\alpha'} s]\!]) \simeq \mathfrak{C}(([\delta]\,\lambda x.\mu\delta'.c)[\![\alpha \setminus^{\alpha'} s]\!])$. Considerar $\mathtt{LXC}' = \boxdot$ y $c' = o'$. Entonces, aplicando la regla $\simeq_{\mathtt{pp}}$ se tiene

$$\begin{array}{rcl}
\mathfrak{C}(\mathtt{LXC}'\langle c'[\![\alpha \setminus^{\alpha'} s]\!]\rangle) & \simeq & \mathfrak{C}(([\delta]\,\lambda x.\mu\delta'.c)[\![\alpha \setminus^{\alpha'} s]\!]) \\
& = & ([\delta]\,\lambda x.\mu\delta'.[\gamma]\,\lambda y.\mu\gamma'.c_0)[\![\alpha \setminus^{\alpha'} \mathfrak{C}(s)]\!] \\
& \simeq & ([\gamma]\,\lambda y.\mu\gamma'.[\delta]\,\lambda x.\mu\delta'.c_0)[\![\alpha \setminus^{\alpha'} \mathfrak{C}(s)]\!] \\
& = & \mathfrak{C}(\mathtt{LXC}'\langle c'[\![\alpha \setminus^{\alpha'} s]\!]\rangle)
\end{array}$$

Se concluye dado que $\simeq \subseteq \simeq_{\mathtt{er}}$.

c) $\alpha \neq \gamma$. Por Lem. 4.6.2 (2), $\mathfrak{C}(\mathtt{LXC}\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle) = \mathfrak{C}([\delta]\,\lambda x.\mu\delta'.[\gamma]\,\lambda y.\mu\gamma'.c_0[\![\alpha \setminus^{\alpha'} s]\!]) \simeq \mathfrak{C}([\delta]\,\lambda x.\mu\delta'.[\gamma]\,\lambda y.\mu\gamma'.c_0[\![\alpha \setminus^{\alpha'} s]\!])$. Considerar $\mathtt{LXC}' = [\gamma]\,\lambda y.\mu\gamma'.[\delta]\,\lambda x.\mu\delta'.\boxdot$ y $c' = c_0$. Luego, se tiene $\mathfrak{C}(\mathtt{LXC}'\langle c'[\![\alpha \setminus^{\alpha'} s]\!]\rangle) \simeq [\delta]\,\lambda x.\mu\delta'.[\gamma]\,\lambda y.\mu\gamma'.c_0[\![\alpha \setminus^{\alpha'} \mathfrak{C}(s)]\!] \simeq_{\mathtt{pp}} [\gamma]\,\lambda y.\mu\gamma'.[\delta]\,\lambda x.\mu\delta'.c_0[\![\alpha \setminus^{\alpha'} \mathfrak{C}(s)]\!] = \mathfrak{C}(\mathtt{LXC}'\langle c'[\![\alpha \setminus^{\alpha'} s]\!]\rangle)$. Se concluye dado que $\simeq \subseteq \simeq_{\mathtt{er}}$.

2. $\mathtt{LXC} = [\delta]\,\lambda x.\mu\delta'.[\gamma]\,\lambda y.\mu\gamma'.\mathtt{LCC}$. Luego, $o' = [\gamma]\,\lambda y.\mu\gamma'.[\delta]\,\lambda x.\mu\delta'.\mathtt{LCC}\langle c\rangle$: *i.e.* $\mathtt{LXC}' = [\gamma]\,\lambda y.\mu\gamma'.[\delta]\,\lambda x.\mu\delta'.\mathtt{LCC}$ y $c' = c$. Más aún, $\alpha \notin \mathtt{LXC}$ implica $\alpha \neq \delta$ y $\alpha \neq \gamma$; $\mathsf{fc}(\alpha', \mathtt{LXC})$ implica $\alpha' \neq \delta'$ y $\alpha' \neq \gamma'$; y $\mathsf{fc}(s, \mathtt{LXC})$ implica $x, y \notin \mathsf{fv}(s)$ y $\delta', \gamma' \notin \mathsf{fn}(s)$. Finalmente, se concluye por $\simeq_{\mathtt{pp}}$, $\mathfrak{C}(\mathtt{LXC}\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle) = [\delta]\,\lambda x.\mu\delta'.[\gamma]\,\lambda y.\mu\gamma'.\mathfrak{C}(\mathtt{LCC}\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle) \simeq_{\mathtt{pp}} [\gamma]\,\lambda y.\mu\gamma'.[\delta]\,\lambda x.\mu\delta'.\mathfrak{C}(\mathtt{LCC}\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle) = \mathfrak{C}(\mathtt{LXC}'\langle c'[\![\alpha \setminus^{\alpha'} s]\!]\rangle)$.

- $\simeq_\rho$. Luego, $\mathtt{LXC} = [\delta]\,\mu\gamma.\mathtt{LCC}$ y $o' = \mathtt{LCC}\langle c\rangle[\gamma \setminus \delta]$: *i.e.* $\mathtt{LXC}' = \mathtt{LCC}[\gamma \setminus \delta]$ y $c' = c$. Más aún, $\alpha \notin \mathtt{LXC}$, $\mathsf{fc}(\alpha', \mathtt{LXC})$ y $\mathsf{fc}(s, \mathtt{LXC})$ implican $\alpha \neq \delta$, $\alpha' \neq \gamma$ y $\gamma \notin \mathsf{fn}(s)$ respectivamente. Se concluye por $\simeq_\rho$, $\mathfrak{C}(\mathtt{LXC}\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle) = [\delta]\,\mu\gamma.\mathfrak{C}(\mathtt{LCC}\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle) \simeq_\rho \mathfrak{C}(\mathtt{LCC}\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle)[\gamma \setminus \delta] = \mathfrak{C}(\mathtt{LXC}'\langle c'[\![\alpha \setminus^{\alpha'} s]\!]\rangle)$.

- $\mathtt{LXC} = \mathtt{LCC}[\![\gamma \setminus^{\delta} s']\!]$. Luego, la única regla aplicable es $\simeq_{\mathtt{exrepl}}$. Se tiene entonces $\mathtt{LCC}\langle c\rangle = \mathtt{LCC}_0\langle c_0\rangle$ con $\gamma \notin \mathtt{LCC}_0$, $\mathsf{fc}(\delta, \mathtt{LCC}_0)$ y $\mathsf{fc}(s', \mathtt{LCC}_0)$. Más aún, $\alpha \notin \mathtt{LXC}$ y $\mathsf{fc}(s, \mathtt{LXC})$ implican $\alpha \neq \delta$, $\alpha \notin s'$ y $\gamma \notin \mathsf{fn}(s)$. Hay dos posibles casos:

1. $c_0 = \mathtt{LCC}_1\langle c\rangle$. Luego, $\mathtt{LXC} = \mathtt{LCC}_0\langle \mathtt{LCC}_1\rangle[\![\gamma \setminus^{\delta} s']\!]$. Más aún, se tiene $o' = \mathtt{LCC}_0\langle \mathtt{LCC}_1\langle c\rangle[\![\gamma \setminus^{\delta} s']\!]\rangle$: *i.e.* $\mathtt{LXC}' = \mathtt{LCC}_0\langle \mathtt{LCC}_1[\![\gamma \setminus^{\delta} s']\!]\rangle$ y $c' = c$. Por Lem. 4.6.2 (2), se tiene $\mathfrak{C}(\mathtt{LXC}\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle) = \mathfrak{C}(\mathtt{LCC}_0\langle \mathtt{LCC}_1\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle\rangle[\![\gamma \setminus^{\delta} s']\!]) \simeq \mathfrak{C}(\mathtt{LCC}_0\langle \mathtt{LCC}_1\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle[\![\gamma \setminus^{\delta} s']\!]\rangle) = \mathfrak{C}(\mathtt{LXC}'\langle c'[\![\alpha \setminus^{\alpha'} s]\!]\rangle)$. Se concluye dado que $\simeq \subseteq \simeq_{\mathtt{er}}$.
2. $c = \mathtt{LCC}_1\langle c_0\rangle$ con $\gamma \notin \mathtt{LCC}_1$ y $\mathsf{fc}(s', \mathtt{LCC}_1)$. Luego, $o' = \mathtt{LCC}\langle \mathtt{LCC}_1\langle c_0[\![\gamma \setminus^{\delta} s']\!]\rangle\rangle$: *i.e.* $\mathtt{LXC}' = \mathtt{LCC}$ y $c' = \mathtt{LCC}_1\langle c_0[\![\gamma \setminus^{\delta} s']\!]\rangle$. Más aún, por Lem. 4.6.2 (2), se tiene $\mathfrak{C}(\mathtt{LXC}\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle) = \mathfrak{C}(\mathtt{LCC}\langle \mathtt{LCC}_1\langle c_0\rangle[\![\alpha \setminus^{\alpha'} s]\!]\rangle[\![\gamma \setminus^{\delta} s']\!]) \simeq \mathfrak{C}(\mathtt{LCC}\langle \mathtt{LCC}_1\langle c_0[\![\gamma \setminus^{\delta} s']\!]\rangle[\![\alpha \setminus^{\alpha'} s]\!]\rangle) = \mathfrak{C}(\mathtt{LXC}'\langle c'[\![\alpha \setminus^{\alpha'} s]\!]\rangle)$. Se concluye dado que $\simeq \subseteq \simeq_{\mathtt{er}}$.

- $\mathtt{LXC} = \mathtt{LCC}[\gamma \setminus \delta]$. Luego, hay dos posibles reglas a analizar:
  - $\simeq_{\mathtt{exren}}$. Luego, $\mathtt{LCC}\langle c\rangle = \mathtt{LCC}_0\langle c_0\rangle$ con $\gamma \notin \mathtt{LCC}_0$ y $\mathsf{fc}(\delta, \mathtt{LCC}_0)$. Más aún, $\alpha \notin \mathtt{LXC}$, $\mathsf{fc}(\alpha', \mathtt{LXC})$ y $\mathsf{fc}(s, \mathtt{LXC})$ implican $\alpha \neq \delta$, $\alpha' \neq \gamma$ y $\gamma \notin \mathsf{fn}(s)$ respectivamente. Hay dos posibles casos:
    1. $c_0 = \mathtt{LCC}_1\langle c\rangle$. Luego, $\mathtt{LXC} = \mathtt{LCC}_0\langle \mathtt{LCC}_1\rangle[\gamma \setminus \delta]$. Más aún, se tiene $o' = \mathtt{LCC}_0\langle \mathtt{LCC}_1\langle c\rangle[\gamma \setminus \delta]\rangle$: *i.e.* $\mathtt{LXC}' = \mathtt{LCC}_0\langle \mathtt{LCC}_1[\gamma \setminus \delta]\rangle$ y $c' = c$. Por Lem. 4.6.2 (2), se tiene $\mathfrak{C}(\mathtt{LXC}\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle) = \mathfrak{C}(\mathtt{LCC}_0\langle \mathtt{LCC}_1\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle\rangle[\gamma \setminus \delta]) \simeq \mathfrak{C}(\mathtt{LCC}_0\langle \mathtt{LCC}_1\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle[\gamma \setminus \delta]\rangle) = \mathfrak{C}(\mathtt{LXC}'\langle c'[\![\alpha \setminus^{\alpha'} s]\!]\rangle)$. Se concluye dado que $\simeq \subseteq \simeq_{\mathtt{er}}$.
    2. $c = \mathtt{LCC}_1\langle c_0\rangle$ con $\gamma \notin \mathtt{LCC}_1$ y $\mathsf{fc}(s', \mathtt{LCC}_1)$. Luego, $o' = \mathtt{LCC}\langle \mathtt{LCC}_1\langle c_0[\gamma \setminus \delta]\rangle\rangle$: *i.e.* $\mathtt{LXC}' = \mathtt{LCC}$ y $c' = \mathtt{LCC}_1\langle c_0[\gamma \setminus \delta]\rangle$. Más aún, por Lem. 4.6.2 (2), se tiene $\mathfrak{C}(\mathtt{LXC}\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle) = \mathfrak{C}(\mathtt{LCC}\langle \mathtt{LCC}_1\langle c_0\rangle[\![\alpha \setminus^{\alpha'} s]\!]\rangle[\gamma \setminus \delta]) \simeq \mathfrak{C}(\mathtt{LCC}\langle \mathtt{LCC}_1\langle c_0[\gamma \setminus \delta]\rangle[\![\alpha \setminus^{\alpha'} s]\!]\rangle) = \mathfrak{C}(\mathtt{LXC}'\langle c'[\![\alpha \setminus^{\alpha'} s]\!]\rangle)$. Se concluye dado que $\simeq \subseteq \simeq_{\mathtt{er}}$.
  - $\simeq_{\mathtt{ren}}$. Luego, $o' = \{\gamma \setminus \delta\}\mathtt{LCC}\langle c\rangle$: *i.e.* $\mathtt{LXC}' = \{\gamma \setminus \delta\}\mathtt{LCC}$ y $c' = \{\gamma \setminus \delta\}c$. Más aún, $\alpha \notin \mathtt{LXC}$, $\mathsf{fc}(\alpha', \mathtt{LXC})$ y $\mathsf{fc}(s, \mathtt{LXC})$ implican $\alpha \neq \delta$, $\alpha' \neq \gamma$ y $\gamma \notin \mathsf{fn}(s)$ respectivamente. Se concluye por $\simeq_{\mathtt{ren}}$, $\mathfrak{C}(\mathtt{LXC}\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle) = \mathfrak{C}(\mathtt{LCC}\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle)[\gamma \setminus \delta] \simeq_{\mathtt{ren}} \{\gamma \setminus \delta\}\mathfrak{C}(\mathtt{LCC}\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle) = \mathfrak{C}(\mathtt{LXC}'\langle c'[\![\alpha \setminus^{\alpha'} s]\!]\rangle)$.

□

Lema C.0.48. *Sean $o, o' \in \mathbb{O}_{\Lambda M}$ tal que $o \simeq_{\mathtt{er}} o'$.*

1. *Si $o, o' \in \mathbb{T}_{\Lambda M}$, entonces $\mathfrak{C}(o :: s) \simeq_{\mathtt{er}} \mathfrak{C}(o' :: s)$.*
2. *Si $o, o' \in \mathbb{C}_{\Lambda M}$, entonces $\mathfrak{C}(o[\![\alpha \setminus^{\alpha'} s]\!]) \simeq_{\mathtt{er}} \mathfrak{C}(o'[\![\alpha \setminus^{\alpha'} s]\!])$.*

Demostración. Se prueban ambos items en simultaneo por inducción en $o \simeq_{\mathtt{er}} o'$. Por definición esto implica verificar cuatro condiciones: reflexividad, transitividad, simetría y congruencia (*i.e.* clausura por contextos):

Reflexividad Luego, $o' = o$ por lo que $\mathfrak{C}(o :: s) = \mathfrak{C}(o' :: s)$ y $\mathfrak{C}(o[\![\alpha \setminus^{\alpha'} s]\!]) = \mathfrak{C}(o'[\![\alpha \setminus^{\alpha'} s]\!])$. Ambos resultados son inmediatos por reflexividad de $\simeq_{\mathtt{er}}$.

Transitividad Luego, existe $p \in \mathbb{O}_{\Lambda M}$ tal que $o \simeq_{\mathtt{er}} p$ y $p \simeq_{\mathtt{er}} o'$.

1. Si $o, o' \in \mathbb{T}_{\Lambda M}$, entonces $p \in \mathbb{T}_{\Lambda M}$. Por *h.i.* se tienen $\mathfrak{C}(o :: s) \simeq_{\mathtt{er}} \mathfrak{C}(p :: s)$ y $\mathfrak{C}(p :: s) \simeq_{\mathtt{er}} \mathfrak{C}(o' :: s)$. Se concluye por transitividad de $\simeq_{\mathtt{er}}$.
2. Si $o, o' \in \mathbb{C}_{\Lambda M}$, entonces $p \in \mathbb{C}_{\Lambda M}$. Por *h.i.* se tienen $\mathfrak{C}(o[\![\alpha \setminus^{\alpha'} s]\!]) \simeq_{\mathtt{er}} \mathfrak{C}(p[\![\alpha \setminus^{\alpha'} s]\!])$ y $\mathfrak{C}(p[\![\alpha \setminus^{\alpha'} s]\!]) \simeq_{\mathtt{er}} \mathfrak{C}(o'[\![\alpha \setminus^{\alpha'} s]\!])$. Se concluye por transitividad de $\simeq_{\mathtt{er}}$.

Simetría Luego, $o' \simeq_{\mathtt{er}} o$.

1. Si $o, o' \in \mathbb{T}_{\Lambda M}$, por *h.i.* se tiene $\mathfrak{C}(o' :: s) \simeq_{\mathtt{er}} \mathfrak{C}(o :: s)$. Se concluye por simetría de $\simeq_{\mathtt{er}}$.

2. Si $o, o' \in \mathbb{C}_{\Lambda M}$, por *h.i.* se tiene $\mathfrak{C}(o'[\![\alpha \setminus^{\alpha'} s]\!]) \simeq_{\mathtt{er}} \mathfrak{C}(o[\![\alpha \setminus^{\alpha'} s]\!])$. Se concluye por simetría de $\simeq_{\mathtt{er}}$.

Congruencia La congruencia se demuestra por inducción en el contexto de clausura $\mathtt{O}$ tal que $o = \mathtt{O}\langle p\rangle$ y $o' = \mathtt{O}\langle p'\rangle$ con $p \simeq_* p'$, donde $\simeq_*$ es una regla de la Fig. 4.13 o $\simeq_{\mathtt{ren}}$.

- $\mathtt{O} = \Box$. Luego, $o = p \simeq_* p' = o'$. Más aún, es el caso de $o, o' \in \mathbb{T}_{\Lambda M}$. Se tienen solo dos reglas aplicables a términos:
  - $\simeq_{\mathtt{exsubs}}$. Luego, $o = \mathtt{LTT}\langle t\rangle[x \setminus v]$ y $o' = \mathtt{LTT}\langle t[x \setminus v]\rangle$ con $x \notin \mathtt{LTT}$ y $\mathsf{fc}(v, \mathtt{LTT})$. Por Lem. 4.6.2 (1), se tiene $\mathfrak{C}(\mathtt{LTT}\langle t\rangle[x \setminus v] :: s) \simeq \mathfrak{C}((\mathtt{LTT}\langle t\rangle :: s)[x \setminus v]) \simeq \mathfrak{C}(\mathtt{LTT}\langle t[x \setminus v]\rangle :: s)$. Notar que $\Box :: s$ constituye un contexto lineal. Se concluye dado que $\simeq \subseteq \simeq_{\mathtt{er}}$.
  - $\simeq_\theta$. Luego, $o = \mu\gamma.[\gamma]\, t$ y $o' = t$ con $\gamma \notin t$. Más aún, sea δ un nombre fresco, entonces $\mathfrak{C}((\mu\gamma.[\gamma]\, t) :: s) = \mu\delta.\mathfrak{C}(([\gamma]\, t)[\![\gamma \setminus^{\delta} s]\!]) = \mu\delta.[\delta]\, \mathfrak{C}(t :: s) \simeq_\theta \mathfrak{C}(t :: s)$.
- $\mathtt{O} = \boxdot$. Luego, $o = p \simeq_* p' = o'$. Más aún, es el caso de $o, o' \in \mathbb{C}_{\Lambda M}$. Por Lem. C.0.47 con $\mathtt{LXC} = \boxdot$ se tiene $o' = \mathtt{LCC}\langle c\rangle$ tal que $\alpha \notin \mathtt{LCC}$, $\mathsf{fc}(\alpha', \mathtt{LCC})$, $\mathsf{fc}(s, \mathtt{LCC})$ y $\mathfrak{C}(o[\![\alpha \setminus^{\alpha'} s]\!]) \simeq_{\mathtt{er}} \mathfrak{C}(\mathtt{LCC}\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle)$. Más aún, por Lem. 4.6.2 (2), $\mathfrak{C}(\mathtt{LCC}\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle) \simeq \mathfrak{C}(\mathtt{LCC}\langle c\rangle[\![\alpha \setminus^{\alpha'} s]\!]) = \mathfrak{C}(o'[\![\alpha \setminus^{\alpha'} s]\!])$. Se concluye dado que $\simeq \subseteq \simeq_{\mathtt{er}}$.
- $\mathtt{O} = \mathtt{T}\, v$. Luego, $o = \mathtt{T}\langle p\rangle\, v$ y $o' = \mathtt{T}\langle p'\rangle\, v$. Sean $t = \mathtt{T}\langle p\rangle$ y $t' = \mathtt{T}\langle p'\rangle$. Se tiene entonces $t \simeq_{\mathtt{er}} t'$. Más aún, $o \simeq_{\mathtt{er}} o'$ implica $\mathfrak{C}(o) = o$ y $\mathfrak{C}(o') = o'$. Luego, $\mathfrak{C}(o :: s) = (t\, v) :: \mathfrak{C}(s) \simeq_{\mathtt{er}} (t'\, v) :: \mathfrak{C}(s) = \mathfrak{C}(o' :: s)$, donde $\mathfrak{C}(s) = \mathfrak{C}(u_0) \cdot \ldots \cdot \mathfrak{C}(u_n)$ para $s = u_0 \cdot \ldots \cdot u_n$.
- $\mathtt{O} = t\, \mathtt{T}$. Luego, $o = t\, v$ y $o' = t\, v'$ con $v \simeq_{\mathtt{er}} v'$. Más aún, $o \simeq_{\mathtt{er}} o'$ implica $\mathfrak{C}(o) = o$ y $\mathfrak{C}(o') = o'$. Luego, $\mathfrak{C}(o :: s) = (t\, v) :: \mathfrak{C}(s) \simeq_{\mathtt{er}} (t\, v') :: \mathfrak{C}(s) = \mathfrak{C}(o' :: s)$.
- $\mathtt{O} = \lambda x.\mathtt{T}$. Luego, $o = \lambda x.t$ y $o' = \lambda x.t'$ con $t \simeq_{\mathtt{er}} t'$. Más aún, $t \simeq_{\mathtt{er}} t'$ implica $\mathfrak{C}(t) = t$ y $\mathfrak{C}(t') = t'$. Se procede por inducción en s:
  - $s = u$. Luego, $\mathfrak{C}(o\, u) = t[x \setminus \mathfrak{C}(u)] \simeq_{\mathtt{er}} t'[x \setminus \mathfrak{C}(u)] = \mathfrak{C}(o'\, u)$.
  - $s = u \cdot s'$. Luego, $\mathfrak{C}(o :: s) = \mathfrak{C}(t[x \setminus u] :: s')$. Por Lem. 4.6.2 (1) y $\simeq \subseteq \simeq_{\mathtt{er}}$, se tiene $\mathfrak{C}(o :: s) \simeq_{\mathtt{er}} \mathfrak{C}(t :: s')[x \setminus \mathfrak{C}(u)]$. Análogamente, $\mathfrak{C}(o' :: s) \simeq_{\mathtt{er}} \mathfrak{C}(t' :: s')[x \setminus \mathfrak{C}(u)]$. Por *h.i.* se tiene $\mathfrak{C}(t :: s) \simeq_{\mathtt{er}} \mathfrak{C}(t' :: s)$, por lo que se concluye.
- $\mathtt{O} = \mu\gamma.\mathtt{C}$. Luego, $o = \mu\gamma.c$ y $o' = \mu\gamma.c'$ con $c \simeq_{\mathtt{er}} c'$. Más aún, $\mathfrak{C}(o\, s) = \mu\delta.\mathfrak{C}(c[\![\gamma \setminus^{\delta} s]\!])$ y $\mathfrak{C}(o'\, s) = \mu\delta.\mathfrak{C}(c'[\![\gamma \setminus^{\delta} s]\!])$. Por *h.i.* se tiene $\mathfrak{C}(c[\![\gamma \setminus^{\delta} s]\!]) \simeq_{\mathtt{er}} \mathfrak{C}(c'[\![\gamma \setminus^{\delta} s]\!])$, por lo que se concluye.
- $\mathtt{O} = \mathtt{T}[x \setminus v]$. Luego, $o = t[x \setminus v]$ y $o' = t'[x \setminus v]$ con $t \simeq_{\mathtt{er}} t'$. Por Lem. 4.6.2 (1) y $\simeq \subseteq \simeq_{\mathtt{er}}$, se tiene $\mathfrak{C}(o) = \mathfrak{C}(t[x \setminus v] :: s) \simeq_{\mathtt{er}} \mathfrak{C}(t :: s)[x \setminus v]$. Análogamente, $\mathfrak{C}(o') \simeq_{\mathtt{er}} \mathfrak{C}(t' :: s)[x \setminus v]$. Por *h.i.* se tiene $\mathfrak{C}(t :: s) \simeq_{\mathtt{er}} \mathfrak{C}(t' :: s)$, por lo que se concluye.
- $\mathtt{O} = t[x \setminus \mathtt{T}]$. Luego, $o = t[x \setminus v]$ y $o' = t[x \setminus v']$ con $v \simeq_{\mathtt{er}} v'$. Más aún, $v \simeq_{\mathtt{er}} v'$ implica $\mathfrak{C}(v) = v$ y $\mathfrak{C}(v') = v'$. Se concluye por Lem. 4.6.2 (1) y $\simeq \subseteq \simeq_{\mathtt{er}}$, teniendo $\mathfrak{C}(o :: s) = \mathfrak{C}(t[x \setminus v] :: s) \simeq_{\mathtt{er}} \mathfrak{C}(t :: s)[x \setminus v] \simeq_{\mathtt{er}} \mathfrak{C}(t :: s)[x \setminus v'] \simeq_{\mathtt{er}} \mathfrak{C}(t[x \setminus v'] :: s) = \mathfrak{C}(o' :: s)$.
- $\mathtt{O} = [\delta]\, \mathtt{T}$. Luego, $o = [\delta]\, \mathtt{T}\langle p\rangle$ y $o' = [\delta]\, \mathtt{T}\langle p'\rangle$. Sean $t = \mathtt{T}\langle p\rangle$ y $t' = \mathtt{T}\langle p'\rangle$. Se tiene entonces $t \simeq_{\mathtt{er}} t'$. Se tienen tres casos posibles:
  1. $\alpha = \delta$ y $\alpha \notin t$. Luego, se tiene también $\alpha \notin t'$. Más aún, $\mathfrak{C}(o[\![\alpha \setminus^{\alpha'} s]\!]) = [\alpha']\, \mathfrak{C}(t :: s)$ y $\mathfrak{C}(o'[\![\alpha \setminus^{\alpha'} s]\!]) = [\alpha']\, \mathfrak{C}(t' :: s)$. Por *h.i.* se tiene $\mathfrak{C}(t :: s) \simeq_{\mathtt{er}} \mathfrak{C}(t' :: s)$, por lo que se concluye.
  2. $o[\![\alpha \setminus^{\alpha'} s]\!] \to_{\mathtt{NPW}} [\delta]\, u$ (*i.e.* $\alpha \neq \delta$ y $\alpha \in t$). Notar que, dado que $t = \mathtt{T}\langle p\rangle$ y $\mathsf{fn}_\alpha(t) = 1$ por hipótesis de las reglas `N`, `P` y `W` hay solo dos casos posibles:
     a) $u = \mathtt{T}'\langle p\rangle$. Luego, se analiza individualmente la regla de redución aplicada:

- $\mathtt{N}$. Luego, $\mathtt{T} = \mathtt{LTC}\langle[\alpha]\,\mathtt{T}_1\rangle$ y $\mathtt{T}' = \mathtt{LTC}\langle[\alpha']\,\mathtt{T}_1 :: s\rangle$ con $\alpha \notin \mathtt{LTC}$ y $\alpha \notin \mathtt{T}_1$. Más aún, $\mathfrak{C}(o[\![\alpha \setminus^{\alpha'} s]\!]) = [\delta]\,\mathfrak{C}(\mathtt{T}'\langle p\rangle) = [\delta]\,\mathtt{LTC}\langle[\alpha']\,\mathfrak{C}(\mathtt{T}_1\langle p\rangle :: s)\rangle$. Por otro lado, se tiene $o'[\![\alpha \setminus^{\alpha'} s]\!] \to_{\mathtt{N}} [\delta]\,\mathtt{T}'\langle p'\rangle$, por lo que $\mathfrak{C}(o'[\![\alpha \setminus^{\alpha'} s]\!]) = [\delta]\,\mathtt{LTC}\langle[\alpha']\,\mathfrak{C}(\mathtt{T}_1\langle p'\rangle :: s)\rangle$. Por *h.i.* se tiene $\mathfrak{C}(\mathtt{T}_1\langle p\rangle :: s) \simeq_{\mathtt{er}} \mathfrak{C}(\mathtt{T}_1\langle p'\rangle :: s)$, por lo que se concluye.
- $\mathtt{P}$. Luego, $\mathtt{T} = \mathtt{LTC}\langle\mathtt{C}[\![\gamma \setminus^{\alpha} s']\!]\rangle$ y $\mathtt{T}' = \mathtt{LTC}\langle\mathtt{C}[\![\gamma \setminus^{\alpha'} s' \cdot s]\!]\rangle$ con $\alpha \notin \mathtt{LTC}$, $\alpha \notin \mathtt{C}$ y $\alpha \notin s'$. Más aún, $\mathfrak{C}(o[\![\alpha \setminus^{\alpha'} s]\!]) = [\delta]\,\mathfrak{C}(\mathtt{T}'\langle p\rangle) = [\delta]\,\mathtt{LTC}\langle\mathfrak{C}(\mathtt{C}\langle p\rangle[\![\gamma \setminus^{\alpha'} s' \cdot s]\!])\rangle$. Por otro lado, se tiene $o'[\![\alpha \setminus^{\alpha'} s]\!] \to_{\mathtt{P}} [\delta]\,\mathtt{T}'\langle p'\rangle$, por lo que $\mathfrak{C}(o'[\![\alpha \setminus^{\alpha'} s]\!]) = [\delta]\,\mathtt{LTC}\langle\mathfrak{C}(\mathtt{C}\langle p'\rangle[\![\gamma \setminus^{\alpha'} s' \cdot s]\!])\rangle$. Por *h.i.* $\mathfrak{C}(\mathtt{C}\langle p\rangle[\![\gamma \setminus^{\alpha'} s' \cdot s]\!]) \simeq_{\mathtt{er}} \mathfrak{C}(\mathtt{C}\langle p'\rangle[\![\gamma \setminus^{\alpha'} s' \cdot s]\!])$, por lo que se concluye.
- $\mathtt{W}$. Luego, $\mathtt{T} = \mathtt{LTC}\langle\mathtt{C}[\gamma \setminus \alpha]\rangle$ y $\mathtt{T}' = \mathtt{LTC}\langle\mathtt{C}[\![\gamma \setminus^{\alpha} s]\!][\alpha \setminus \alpha']\rangle$ con $\alpha \notin \mathtt{LTC}$ y $\alpha \notin \mathtt{C}$. Más aún, $\mathfrak{C}(o[\![\alpha \setminus^{\alpha'} s]\!]) = [\delta]\,\mathfrak{C}(\mathtt{T}'\langle p\rangle) = [\delta]\,\mathtt{LTC}\langle\mathfrak{C}(\mathtt{C}\langle p\rangle[\![\gamma \setminus^{\alpha} s]\!])[\alpha \setminus \alpha']\rangle$. Por otro lado, $o'[\![\alpha \setminus^{\alpha'} s]\!] \to_{\mathtt{W}} [\delta]\,\mathtt{T}'\langle p'\rangle$, por lo que $\mathfrak{C}(o'[\![\alpha \setminus^{\alpha'} s]\!]) = [\delta]\,\mathtt{LTC}\langle\mathfrak{C}(\mathtt{C}\langle p'\rangle[\![\gamma \setminus^{\alpha} s]\!])[\alpha \setminus \alpha']\rangle$. Por *h.i.* se tiene $\mathfrak{C}(\mathtt{C}\langle p\rangle[\![\gamma \setminus^{\alpha} s]\!]) \simeq_{\mathtt{er}} \mathfrak{C}(\mathtt{C}\langle p'\rangle[\![\gamma \setminus^{\alpha} s]\!])$, por lo que se concluye.

b) $u = \mathtt{T}\langle q\rangle$. Luego, se tiene un contexto lineal $\mathtt{LTC}$ tal que $\mathtt{LTC} = \mathtt{LTX}\langle\mathtt{LXC}\rangle$, $\mathtt{T} = \mathtt{LTX}$ y $p = \mathtt{LXC}\langle c\rangle$ con $\alpha \neq \delta$, $\alpha \notin \mathtt{LTC}$ y $\alpha \in c$. Por α-conversión se asumen también $\mathsf{fc}(\alpha', \mathtt{LTC})$ y $\mathsf{fc}(s, \mathtt{LTC})$, de modo que $c[\![\alpha \setminus^{\alpha'} s]\!] \to_{\mathtt{NPW}} q$. Luego, por Lem. C.0.47 con $p \simeq_* p'$ se tiene $p' = \mathtt{LXC}'\langle c'\rangle$ tal que $\alpha \notin \mathtt{LXC}'$, $\mathsf{fc}(\alpha', \mathtt{LXC}')$, $\mathsf{fc}(s, \mathtt{LXC}')$ y $\mathfrak{C}(\mathtt{LXC}\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle) \simeq_{\mathtt{er}} \mathfrak{C}(\mathtt{LXC}'\langle c'[\![\alpha \setminus^{\alpha'} s]\!]\rangle)$. Finalmente, se tiene $o = [\delta]\,\mathtt{T}\langle p\rangle = [\delta]\,\mathtt{LTX}\langle\mathtt{LXC}\langle c\rangle\rangle$ y $o' = [\delta]\,\mathtt{T}\langle p'\rangle = [\delta]\,\mathtt{LTX}\langle\mathtt{LXC}'\langle c'\rangle\rangle$, y por Lem. 4.6.2 (2), se obtiene

$$\begin{array}{rcl}
\mathfrak{C}(o[\![\alpha \setminus^{\alpha'} s]\!]) & \simeq & \mathfrak{C}([\delta]\,\mathtt{LTX}\langle\mathtt{LXC}\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle\rangle) \\
& = & [\delta]\,\mathtt{LTX}\langle\mathfrak{C}(\mathtt{LXC}\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle)\rangle \\
& \simeq_{\mathtt{er}} & [\delta]\,\mathtt{LTX}\langle\mathfrak{C}(\mathtt{LXC}'\langle c'[\![\alpha \setminus^{\alpha'} s]\!]\rangle)\rangle \\
& = & \mathfrak{C}([\delta]\,\mathtt{LTX}\langle\mathtt{LXC}'\langle c'[\![\alpha \setminus^{\alpha'} s]\!]\rangle\rangle) \\
& \simeq & \mathfrak{C}(o'[\![\alpha \setminus^{\alpha'} s]\!])
\end{array}$$

Se concluye dado que $\simeq \subseteq \simeq_{\mathtt{er}}$.

3. De lo contrario, $\mathfrak{C}(o[\![\alpha \setminus^{\alpha'} s]\!]) = o[\![\alpha \setminus^{\alpha'} \mathfrak{C}(s)]\!] \simeq_{\mathtt{er}} o'[\![\alpha \setminus^{\alpha'} \mathfrak{C}(s)]\!] = \mathfrak{C}(o'[\![\alpha \setminus^{\alpha'} s]\!])$, dado que $o \simeq_{\mathtt{er}} o'$ implica $\mathfrak{C}(o) = o$ y $\mathfrak{C}(o') = o'$.

- $\mathtt{O} = \mathtt{C}[\![\gamma \setminus^{\delta} s']\!]$. Luego, $o = c[\![\gamma \setminus^{\delta} s']\!]$ y $o' = c'[\![\gamma \setminus^{\delta} s']\!]$ con $c \simeq_{\mathtt{er}} c'$. Más aún, $o \simeq_{\mathtt{er}} o'$ implica $\mathfrak{C}(o) = o$ y $\mathfrak{C}(o') = o'$. Luego, se tienen tres casos posibles:
  1. $\alpha = \delta$, $\alpha \notin c$ y $\alpha \notin s'$. Luego, se tiene también $\alpha \notin c'$. Más aún, $\mathfrak{C}(o[\![\alpha \setminus^{\alpha'} s]\!]) = c[\![\gamma \setminus^{\alpha'} s' \cdot \mathfrak{C}(s)]\!] \simeq_{\mathtt{er}} c'[\![\gamma \setminus^{\alpha'} s' \cdot \mathfrak{C}(s)]\!] = \mathfrak{C}(o'[\![\alpha \setminus^{\alpha'} s]\!])$, por lo que se concluye.
  2. $\alpha \neq \delta$ y $\alpha \notin s'$. Por α-conversión se asumen también $\alpha' \neq \gamma$ y $\gamma \notin \mathsf{fn}(s)$. Luego, por Lem. 4.6.2 (2) con $\simeq \subseteq \simeq_{\mathtt{er}}$ se tienen $\mathfrak{C}(o[\![\alpha \setminus^{\alpha'} s]\!]) \simeq_{\mathtt{er}} \mathfrak{C}(c[\![\alpha \setminus^{\alpha'} s]\!])[\![\gamma \setminus^{\delta} s']\!]$ y $\mathfrak{C}(o'[\![\alpha \setminus^{\alpha'} s]\!]) \simeq_{\mathtt{er}} \mathfrak{C}(c'[\![\alpha \setminus^{\alpha'} s]\!])[\![\gamma \setminus^{\delta} s']\!]$. Finalmente, por *h.i.* se tiene $\mathfrak{C}(c[\![\alpha \setminus^{\alpha'} s]\!]) \simeq_{\mathtt{er}} \mathfrak{C}(c'[\![\alpha \setminus^{\alpha'} s]\!])$, por lo que se concluye.
  3. De lo contrario, $\mathfrak{C}(o[\![\alpha \setminus^{\alpha'} s]\!]) = o[\![\alpha \setminus^{\alpha'} \mathfrak{C}(s)]\!] \simeq_{\mathtt{er}} o'[\![\alpha \setminus^{\alpha'} \mathfrak{C}(s)]\!] = \mathfrak{C}(o'[\![\alpha \setminus^{\alpha'} s]\!])$.
- $\mathtt{O} = c[\![\gamma \setminus^{\delta} \mathtt{S}]\!]$. Luego, $o = c[\![\gamma \setminus^{\delta} s']\!]$ y $o' = c[\![\gamma \setminus^{\delta} s'']\!]$ con $s' \simeq_{\mathtt{er}} s''$. Más aún, $o \simeq_{\mathtt{er}} o'$ implica $\mathfrak{C}(o) = o$ y $\mathfrak{C}(o') = o'$. Luego, se tienen tres casos posibles:
  1. $\alpha = \delta$, $\alpha \notin c$ y $\alpha \notin s'$. Luego, se tiene también $\alpha \notin s''$. Más aún, $\mathfrak{C}(o[\![\alpha \setminus^{\alpha'} s]\!]) = c[\![\gamma \setminus^{\alpha'} s' \cdot \mathfrak{C}(s)]\!] \simeq_{\mathtt{er}} c[\![\gamma \setminus^{\alpha'} s'' \cdot \mathfrak{C}(s)]\!] = \mathfrak{C}(o'[\![\alpha \setminus^{\alpha'} s]\!])$, por lo que se concluye.
  2. $\alpha \neq \delta$ y $\alpha \notin s'$. Luego, se tiene también $\alpha \notin s''$. Por α-conversión se asumen también $\alpha' \neq \gamma$ y $\gamma \notin \mathsf{fn}(s)$. Se concluye por Lem. 4.6.2 (2) y $\simeq \subseteq \simeq_{\mathtt{er}}$, teniendo $\mathfrak{C}(o[\![\alpha \setminus^{\alpha'} s]\!]) \simeq_{\mathtt{er}} \mathfrak{C}(c[\![\alpha \setminus^{\alpha'} s]\!])[\![\gamma \setminus^{\delta} s']\!] \simeq_{\mathtt{er}} \mathfrak{C}(c[\![\alpha \setminus^{\alpha'} s]\!])[\![\gamma \setminus^{\delta} s'']\!] \simeq_{\mathtt{er}} \mathfrak{C}(o'[\![\alpha \setminus^{\alpha'} s]\!])$.
  3. De lo contrario, $\mathfrak{C}(o[\![\alpha \setminus^{\alpha'} s]\!]) = o[\![\alpha \setminus^{\alpha'} \mathfrak{C}(s)]\!] \simeq_{\mathtt{er}} o'[\![\alpha \setminus^{\alpha'} \mathfrak{C}(s)]\!] = \mathfrak{C}(o'[\![\alpha \setminus^{\alpha'} s]\!])$.
- $\mathtt{O} = \mathtt{C}[\gamma \setminus \delta]$. Luego, $o = c[\gamma \setminus \delta]$ y $o' = c'[\gamma \setminus \delta]$ con $c \simeq_{\mathtt{er}} c'$. Más aún, $o \simeq_{\mathtt{er}} o'$ implica $\mathfrak{C}(o) = o$ y $\mathfrak{C}(o') = o'$. Luego, se tienen tres casos posibles:

1. $\alpha = \delta$ y $\alpha \notin c$. Luego, se tiene también $\alpha \notin c'$. Más aún, $\mathfrak{C}(o[\![\alpha \setminus^{\alpha'} s]\!]) = \mathfrak{C}(c[\![\gamma \setminus^{\alpha} s]\!])[\alpha' \setminus \alpha]$ y $\mathfrak{C}(o'[\![\alpha \setminus^{\alpha'} s]\!]) = \mathfrak{C}(c'[\![\gamma \setminus^{\alpha} s]\!])[\alpha' \setminus \alpha]$. Finalmente, por *h.i.* se tiene $\mathfrak{C}(c[\![\gamma \setminus^{\alpha'} s]\!]) \simeq_{\mathtt{er}} \mathfrak{C}(c'[\![\gamma \setminus^{\alpha'} s]\!])$, por lo que se concluye.
2. $\alpha \neq \delta$. Luego, por Lem. 4.6.2 (2) con $\simeq \subseteq \simeq_{\mathtt{er}}$ valen $\mathfrak{C}(o[\![\alpha \setminus^{\alpha'} s]\!]) \simeq_{\mathtt{er}} \mathfrak{C}(c[\![\alpha \setminus^{\alpha'} s]\!])[\gamma \setminus \delta]$ y $\mathfrak{C}(o'[\![\alpha \setminus^{\alpha'} s]\!]) \simeq_{\mathtt{er}} \mathfrak{C}(c'[\![\alpha \setminus^{\alpha'} s]\!])[\gamma \setminus \delta]$. Finalmente, por *h.i.* se tiene $\mathfrak{C}(c[\![\alpha \setminus^{\alpha'} s]\!]) \simeq_{\mathtt{er}} \mathfrak{C}(c'[\![\alpha \setminus^{\alpha'} s]\!])$, por lo que se concluye.
3. De lo contrario, $\mathfrak{C}(o[\![\alpha \setminus^{\alpha'} s]\!]) = o[\![\alpha \setminus^{\alpha'} \mathfrak{C}(s)]\!] \simeq_{\mathtt{er}} o'[\![\alpha \setminus^{\alpha'} \mathfrak{C}(s)]\!] = \mathfrak{C}(o'[\![\alpha \setminus^{\alpha'} s]\!])$.

- $\mathtt{O} = \mathtt{T} \cdot s$. El resultado es inmediato pues no se satisfacen las hipótesis de ninguno de los dos casos.
- $\mathtt{O} = t \cdot \mathtt{S}$. El resultado es inmediato pues no se satisfacen las hipótesis de ninguno de los dos casos.

□

Lema C.0.49. *Sea $o \in \mathbb{O}_{\Lambda M}$. Si $o \simeq_{\mathtt{er}} o'$, entonces para todo contexto $\mathtt{O}$ del sort adecuado se tiene $\mathfrak{C}(\mathtt{O}\langle o\rangle) \simeq_{\mathtt{er}} \mathfrak{C}(\mathtt{O}\langle o'\rangle)$.*

Demostración. Por inducción en $\mathtt{O}$.

- $\mathtt{O} = \Box$. El resultado es inmediato por hipótesis pues, por definición, $o \simeq_{\mathtt{er}} o'$ implica $\mathfrak{C}(o) = o$ y $\mathfrak{C}(o') = o'$.
- $\mathtt{O} = \boxdot$. El resultado es inmediato por hipótesis pues, por definición, $o \simeq_{\mathtt{er}} o'$ implica $\mathfrak{C}(o) = o$ y $\mathfrak{C}(o') = o'$.
- $\mathtt{O} = \mathtt{T}\, u$. Luego, $\mathfrak{C}(\mathtt{O}\langle o\rangle) = \mathfrak{C}(\mathfrak{C}(\mathtt{T}\langle o\rangle)\, u)$ y $\mathfrak{C}(\mathtt{O}\langle o'\rangle) = \mathfrak{C}(\mathfrak{C}(\mathtt{T}\langle o'\rangle)\, u)$. Por *h.i.* se tiene $\mathfrak{C}(\mathtt{T}\langle o\rangle) \simeq_{\mathtt{er}} \mathfrak{C}(\mathtt{T}\langle o'\rangle)$. Luego, se concluye por Lem. C.0.48 (1).
- $\mathtt{O} = t\, \mathtt{T}$. Luego, se tienen tres casos posible:
  1. $\mathfrak{C}(t) = \mathtt{L}\langle \lambda x.u\rangle$. Luego, $\mathfrak{C}(\mathtt{O}\langle o\rangle) = \mathfrak{C}(\mathtt{L}\langle \lambda x.u\rangle\, \mathtt{T}\langle o\rangle) = \mathtt{L}\langle u[x \setminus \mathfrak{C}(\mathtt{T}\langle o\rangle)]\rangle$. Del mismo modo, $\mathfrak{C}(\mathtt{O}\langle o'\rangle) = \mathtt{L}\langle u[x \setminus \mathfrak{C}(\mathtt{T}\langle o'\rangle)]\rangle$. Se concluye pues por *h.i.* se tiene $\mathfrak{C}(\mathtt{T}\langle o\rangle) \simeq_{\mathtt{er}} \mathfrak{C}(\mathtt{T}\langle o'\rangle)$.
  2. $\mathfrak{C}(t) = \mathtt{L}\langle \mu\alpha.c\rangle$. Luego, $\mathfrak{C}(\mathtt{O}\langle o\rangle) = \mathfrak{C}(\mathtt{L}\langle \mu\alpha.c\rangle\, \mathtt{T}\langle o\rangle) = \mathtt{L}\langle \mu\alpha'.\mathfrak{C}(c[\![\alpha \setminus^{\alpha'} \mathtt{T}\langle o\rangle]\!])\rangle$. Del mismo modo, $\mathfrak{C}(\mathtt{O}\langle o'\rangle) = \mathtt{L}\langle \mu\alpha'.\mathfrak{C}(c[\![\alpha \setminus^{\alpha'} \mathtt{T}\langle o'\rangle]\!])\rangle$. Por Lem. C.0.28, $\mathsf{sz}(\mathtt{O}\langle o\rangle) = \mathsf{sz}(t\, \mathtt{T}\langle o\rangle) \geq \mathsf{sz}(\mathtt{L}\langle \mu\alpha.c\rangle\, \mathtt{T}\langle o\rangle) \geq \mathsf{sz}(\mathtt{L}\langle \mu\alpha'.c[\![\alpha \setminus^{\alpha'} \mathtt{T}\langle o\rangle]\!]\rangle)$, por lo que $\mathsf{sz}(\mathtt{O}) > \mathsf{sz}(c[\![\alpha \setminus^{\alpha'} \mathtt{T}]\!])$. Finalmente, se concluye pues por *h.i.* se tiene $\mathfrak{C}(c[\![\alpha \setminus^{\alpha'} \mathtt{T}\langle o\rangle]\!]) \simeq_{\mathtt{er}} \mathfrak{C}(c[\![\alpha \setminus^{\alpha'} \mathtt{T}\langle o'\rangle]\!])$.
  3. De lo contrario, $\mathfrak{C}(\mathtt{O}\langle o\rangle) = \mathfrak{C}(t)\, \mathfrak{C}(\mathtt{T}\langle o\rangle)$ y $\mathfrak{C}(\mathtt{O}\langle o'\rangle) = \mathfrak{C}(t)\, \mathfrak{C}(\mathtt{T}\langle o'\rangle)$. Se concluye pues por *h.i.* se tiene $\mathfrak{C}(\mathtt{T}\langle o\rangle) \simeq_{\mathtt{er}} \mathfrak{C}(\mathtt{T}\langle o'\rangle)$.
- $\mathtt{O} = \lambda x.\mathtt{T}$. Luego, $\mathfrak{C}(\mathtt{O}\langle o\rangle) = \lambda x.\mathfrak{C}(\mathtt{T}\langle o\rangle)$ y $\mathfrak{C}(\mathtt{O}\langle o'\rangle) = \lambda x.\mathfrak{C}(\mathtt{T}\langle o'\rangle)$. Se concluye pues por *h.i.* se tiene $\mathfrak{C}(\mathtt{T}\langle o\rangle) \simeq_{\mathtt{er}} \mathfrak{C}(\mathtt{T}\langle o'\rangle)$.
- $\mathtt{O} = \mu\alpha.\mathtt{C}$. Luego, $\mathfrak{C}(\mathtt{O}\langle o\rangle) = \mu\alpha.\mathfrak{C}(\mathtt{C}\langle o\rangle)$ y $\mathfrak{C}(\mathtt{O}\langle o'\rangle) = \mu\alpha.\mathfrak{C}(\mathtt{C}\langle o'\rangle)$. Se concluye pues por *h.i.* se tiene $\mathfrak{C}(\mathtt{C}\langle o\rangle) \simeq_{\mathtt{er}} \mathfrak{C}(\mathtt{C}\langle o'\rangle)$.
- $\mathtt{O} = \mathtt{T}[x \setminus u]$. Luego, $\mathfrak{C}(\mathtt{O}\langle o\rangle) = \mathfrak{C}(\mathtt{T}\langle o\rangle)[x \setminus u]$ y $\mathfrak{C}(\mathtt{O}\langle o'\rangle) = \mathfrak{C}(\mathtt{T}\langle o'\rangle)[x \setminus u]$. Se concluye pues por *h.i.* se tiene $\mathfrak{C}(\mathtt{T}\langle o\rangle) \simeq_{\mathtt{er}} \mathfrak{C}(\mathtt{T}\langle o'\rangle)$.
- $\mathtt{O} = t[x \setminus \mathtt{T}]$. Luego, $\mathfrak{C}(\mathtt{O}\langle o\rangle) = t[x \setminus \mathfrak{C}(\mathtt{T}\langle o\rangle)]$ y $\mathfrak{C}(\mathtt{O}\langle o'\rangle) = t[x \setminus \mathfrak{C}(\mathtt{T}\langle o'\rangle)]$. Se concluye pues por *h.i.* se tiene $\mathfrak{C}(\mathtt{T}\langle o\rangle) \simeq_{\mathtt{er}} \mathfrak{C}(\mathtt{T}\langle o'\rangle)$.
- $\mathtt{O} = [\alpha]\, \mathtt{T}$. Luego, $\mathfrak{C}(\mathtt{O}\langle o\rangle) = [\alpha]\, \mathfrak{C}(\mathtt{T}\langle o\rangle)$ y $\mathfrak{C}(\mathtt{O}\langle o'\rangle) = [\alpha]\, \mathfrak{C}(\mathtt{T}\langle o'\rangle)$. Se concluye pues por *h.i.* se tiene $\mathfrak{C}(\mathtt{T}\langle o\rangle) \simeq_{\mathtt{er}} \mathfrak{C}(\mathtt{T}\langle o'\rangle)$.

- $\mathtt{O} = \mathtt{C}[\![\alpha \setminus^{\alpha'} s]\!]$. Luego, se tienen $\mathfrak{C}(\mathtt{O}\langle o\rangle) = \mathfrak{C}(\mathfrak{C}(\mathtt{C}\langle o\rangle)[\![\alpha \setminus^{\alpha'} s]\!])$ y $\mathfrak{C}(\mathtt{O}\langle o'\rangle) = \mathfrak{C}(\mathfrak{C}(\mathtt{C}\langle o'\rangle)[\![\alpha \setminus^{\alpha'} s]\!])$. Por *h.i.* $\mathfrak{C}(\mathtt{C}\langle o\rangle) \simeq_{\mathtt{er}} \mathfrak{C}(\mathtt{C}\langle o'\rangle)$. Finalmente, se concluye por Lem. C.0.48 (2).

- $\mathtt{O} = c[\![\alpha \setminus^{\alpha'} \mathtt{S}]\!]$. Luego, se tienen cuatro casos posible:

  1. $\mathfrak{C}(c) = \mathtt{LCC}\langle[\alpha]\, t\rangle$ con $\alpha \notin t$, $\alpha \notin \mathtt{LCC}$, $\mathsf{fc}(\alpha', \mathtt{LCC})$ y $\mathsf{fc}(\mathtt{S}\langle o\rangle, \mathtt{LCC})$. Más aún, se tiene también $\mathsf{fc}(\mathtt{S}\langle o'\rangle, \mathtt{LCC})$. Luego, $\mathfrak{C}(\mathtt{O}\langle o\rangle) = \mathfrak{C}(\mathtt{LCC}\langle[\alpha]\, t\rangle[\![\alpha \setminus^{\alpha'} \mathtt{S}\langle o\rangle]\!]) = \mathtt{LCC}\langle[\alpha']\, \mathfrak{C}(t :: \mathtt{S}\langle o\rangle)\rangle$. Del mismo modo, $\mathfrak{C}(\mathtt{O}\langle o'\rangle) = \mathtt{LCC}\langle[\alpha']\, \mathfrak{C}(t :: \mathtt{S}\langle o'\rangle)\rangle$. Por Lem. C.0.28, $\mathsf{sz}(\mathtt{O}\langle o\rangle) = \mathsf{sz}(c[\![\alpha \setminus^{\alpha'} \mathtt{S}\langle o\rangle]\!]) \geq \mathsf{sz}(\mathtt{LCC}\langle[\alpha]\, t\rangle[\![\alpha \setminus^{\alpha'} \mathtt{S}\langle o\rangle]\!]) \geq \mathsf{sz}(\mathtt{LCC}\langle[\alpha']\, t :: \mathtt{S}\langle o'\rangle\rangle)$, por lo que $\mathsf{sz}(\mathtt{O}) > \mathsf{sz}(t :: \mathtt{S})$. Finalmente, se concluye pues por *h.i.* se tiene $\mathfrak{C}(t :: \mathtt{S}\langle o\rangle) \simeq_{\mathtt{er}} \mathfrak{C}(t :: \mathtt{S}\langle o'\rangle)$.
  2. $\mathfrak{C}(c) = \mathtt{LCC}\langle c'[\![\beta \setminus^{\alpha} s]\!]\rangle$ con $\alpha \notin c'$, $\alpha \notin \mathtt{LCC}$, $\alpha \notin s$, $\mathsf{fc}(\alpha', \mathtt{LCC})$ y $\mathsf{fc}(\mathtt{S}\langle o\rangle, \mathtt{LCC})$. Más aún, se tiene también $\mathsf{fc}(\mathtt{S}\langle o'\rangle, \mathtt{LCC})$. Luego, se tiene $\mathfrak{C}(\mathtt{O}\langle o\rangle) = \mathfrak{C}(\mathtt{LCC}\langle c'[\![\beta \setminus^{\alpha} s]\!]\rangle[\![\alpha \setminus^{\alpha'} \mathtt{S}\langle o\rangle]\!]) = \mathtt{LCC}\langle c'[\![\beta \setminus^{\alpha'} s \cdot \mathfrak{C}(\mathtt{S}\langle o\rangle)]\!]\rangle$. Del mismo modo, $\mathfrak{C}(\mathtt{O}\langle o'\rangle) = \mathtt{LCC}\langle c'[\![\beta \setminus^{\alpha'} s \cdot \mathfrak{C}(\mathtt{S}\langle o'\rangle)]\!]\rangle$. Se concluye pues por *h.i.* se tiene $\mathfrak{C}(\mathtt{S}\langle o\rangle) \simeq_{\mathtt{er}} \mathfrak{C}(\mathtt{S}\langle o'\rangle)$.
  3. $\mathfrak{C}(c) = \mathtt{LCC}\langle c'[\beta \setminus \alpha]\rangle$ con $\alpha \notin c'$, $\alpha \notin \mathtt{LCC}$, $\mathsf{fc}(\alpha', \mathtt{LCC})$ y $\mathsf{fc}(\mathtt{S}\langle o\rangle, \mathtt{LCC})$. Más aún, se tiene también $\mathsf{fc}(\mathtt{S}\langle o'\rangle, \mathtt{LCC})$. Por lo tanto, se tiene $\mathfrak{C}(\mathtt{O}\langle o\rangle) = \mathfrak{C}(\mathtt{LCC}\langle c'[\beta \setminus \alpha]\rangle[\![\alpha \setminus^{\alpha'} \mathtt{S}]\!]) = \mathtt{LCC}\langle \mathfrak{C}(c'[\![\beta \setminus^{\alpha} \mathtt{S}\langle o\rangle]\!])[\alpha \setminus \alpha']\rangle$. Del mismo modo, $\mathfrak{C}(\mathtt{O}\langle o'\rangle) = \mathtt{LCC}\langle \mathfrak{C}(c'[\![\beta \setminus^{\alpha} \mathtt{S}\langle o'\rangle]\!])[\alpha \setminus \alpha']\rangle$. Además, por Lem. C.0.28, se tiene $\mathsf{sz}(\mathtt{O}\langle o\rangle) = \mathsf{sz}(c[\![\alpha \setminus^{\alpha'} \mathtt{S}\langle o\rangle]\!]) \geq \mathsf{sz}(\mathtt{LCC}\langle c'[\beta \setminus \alpha]\rangle[\![\alpha \setminus^{\alpha'} \mathtt{S}]\!]) \geq \mathsf{sz}(\mathtt{LCC}\langle c'[\![\beta \setminus^{\alpha} \mathtt{S}\langle o\rangle]\!][\alpha \setminus \alpha']\rangle)$, por lo que $\mathsf{sz}(\mathtt{O}) > \mathsf{sz}(c'[\![\beta \setminus^{\alpha} \mathtt{S}]\!])$. Finalmente, se concluye pues por *h.i.* se tiene $\mathfrak{C}(c'[\![\beta \setminus^{\alpha} \mathtt{S}\langle o\rangle]\!]) \simeq_{\mathtt{er}} \mathfrak{C}(c'[\![\beta \setminus^{\alpha} \mathtt{S}\langle o\rangle]\!])$.
  4. De lo contrario, se tienen $\mathfrak{C}(\mathtt{O}\langle o\rangle) = \mathfrak{C}(c)[\![\alpha \setminus^{\alpha'} \mathfrak{C}(\mathtt{S}\langle o\rangle)]\!]$ y $\mathfrak{C}(\mathtt{O}\langle o'\rangle) = \mathfrak{C}(c)[\![\alpha \setminus^{\alpha'} \mathfrak{C}(\mathtt{S}\langle o'\rangle)]\!]$. Se concluye pues por *h.i.* se tiene $\mathfrak{C}(\mathtt{S}\langle o\rangle) \simeq_{\mathtt{er}} \mathfrak{C}(\mathtt{S}\langle o'\rangle)$.

- $\mathtt{O} = \mathtt{C}[\alpha \setminus \beta]$. Luego, $\mathfrak{C}(\mathtt{O}\langle o\rangle) = \mathfrak{C}(\mathtt{C}\langle o\rangle)[\alpha \setminus \beta]$ y $\mathfrak{C}(\mathtt{O}\langle o'\rangle) = \mathfrak{C}(\mathtt{C}\langle o'\rangle)[\alpha \setminus \beta]$. Se concluye pues por *h.i.* se tiene $\mathfrak{C}(\mathtt{C}\langle o\rangle) \simeq_{\mathtt{er}} \mathfrak{C}(\mathtt{C}\langle o'\rangle)$.

- $\mathtt{O} = \mathtt{T} \cdot s$. Luego, $\mathfrak{C}(\mathtt{O}\langle o\rangle) = \mathfrak{C}(\mathtt{T}\langle o\rangle) \cdot s$ y $\mathfrak{C}(\mathtt{O}\langle o'\rangle) = \mathfrak{C}(\mathtt{T}\langle o'\rangle) \cdot s$. Se concluye pues por *h.i.* se tiene $\mathfrak{C}(\mathtt{T}\langle o\rangle) \simeq_{\mathtt{er}} \mathfrak{C}(\mathtt{T}\langle o'\rangle)$.

- $\mathtt{O} = t \cdot \mathtt{S}$. Luego, $\mathfrak{C}(\mathtt{O}\langle o\rangle) = t \cdot \mathfrak{C}(\mathtt{S}\langle o\rangle)$ y $\mathfrak{C}(\mathtt{O}\langle o'\rangle) = t \cdot \mathfrak{C}(\mathtt{S}\langle o'\rangle)$. Se concluye pues por *h.i.* se tiene $\mathfrak{C}(\mathtt{S}\langle o\rangle) \simeq_{\mathtt{er}} \mathfrak{C}(\mathtt{S}\langle o'\rangle)$.

□

Lema 4.7.1. *Sean $o, p \in \mathbb{O}_{\lambda\mu}$. Si $o \simeq_\sigma p$, entonces $\mathfrak{C}(o) \simeq_{\mathtt{er}} \mathfrak{C}(p)$.*

Demostración. La prueba es por inducción en $o \simeq_\sigma p$. Por definición esto implica verificar cuatro condiciones: reflexividad, transitividad, simetría y congruencia (*i.e.* clausura por contextos):

Reflexividad Luego, $p = o$ por lo que el resultado es inmediato por reflexividad de $\simeq_{\mathtt{er}}$.

Transitividad Luego, existe $q \in \mathbb{O}_{\lambda\mu}$ tal que $o \simeq_\sigma q$ y $q \simeq_\sigma p$. Por *h.i.* se tienen $\mathfrak{C}(o) \simeq_{\mathtt{er}} \mathfrak{C}(q)$ y $\mathfrak{C}(q) \simeq_{\mathtt{er}} \mathfrak{C}(p)$. Se concluye por transitividad de $\simeq_{\mathtt{er}}$.

Simetría Luego, $p \simeq_\sigma o$ y por *h.i.* se tiene $\mathfrak{C}(p) \simeq_{\mathtt{er}} \mathfrak{C}(o)$. Se concluye por simetría de $\simeq_{\mathtt{er}}$.

Congruencia La congruencia se demuestra por inducción en el contexto de clausura $\mathtt{O}$ tal que $o = \mathtt{O}\langle o'\rangle$ y $o' = \mathtt{O}\langle p'\rangle$ con $o' \simeq_{\sigma_*} p'$, donde $\simeq_{\sigma_*}$ es una regla de la Fig. 4.10.

- $\mathtt{O} = \Box$. Luego, $o = o' \simeq_{\sigma_*} p' = p$. Más aún, los objetos son necesariamente términos. Se analiza la regla aplicada:

- $\simeq_{\sigma_1}$. Luego, $o = (\lambda x.\lambda y.t)\,u$ y $p = \lambda y.(\lambda x.t)\,u$ con $y \notin u$. Más aún, se tiene $\mathfrak{C}(o) = \mathfrak{C}((\lambda y.t)[x \setminus u])$ y $\mathfrak{C}(p) = \mathfrak{C}(\lambda y.t[x \setminus u])$. Por Lem. 4.6.2 (1), se tiene $\mathfrak{C}((\lambda y.t)[x \setminus u]) \simeq \mathfrak{C}(\lambda y.t[x \setminus u])$. Se concluye dado que $\simeq \subseteq \simeq_{\mathtt{er}}$.
- $\simeq_{\sigma_2}$. Luego, $o = (\lambda x.t\,v)\,u$ y $p = (\lambda x.t)\,u\,v$ con $x \notin v$. Más aún, $\mathfrak{C}(o) = \mathfrak{C}((t\,v)[x \setminus u])$ y $\mathfrak{C}(p) = \mathfrak{C}(t[x \setminus u]\,v)$. Se concluye pues, por Lem. 4.6.2 (1), $\mathfrak{C}((t\,v)[x \setminus u]) \simeq \mathfrak{C}(t[x \setminus u]\,v)$ y $\simeq \subseteq \simeq_{\mathtt{er}}$.
- $\simeq_{\sigma_3}$. Luego, $o = (\lambda x.\mu\beta.[\alpha]\,t)\,u$ y $p = \mu\beta.[\alpha]\,(\lambda x.t)\,u$ con $\beta \notin u$. Más aún, $\mathfrak{C}(o) = \mathfrak{C}((\mu\beta.[\alpha]\,t)[x \setminus u])$ y $\mathfrak{C}(p) = \mathfrak{C}(\mu\beta.[\alpha]\,t[x \setminus u])$. Se concluye por Lem. 4.6.2 (1) y el hecho de que $\simeq \subseteq \simeq_{\mathtt{er}}$, $\mathfrak{C}((\mu\beta.[\alpha]\,t)[x \setminus u]) \simeq_{\mathtt{er}} \mathfrak{C}(\mu\beta.[\alpha]\,t[x \setminus u])$.

- $\mathtt{O} = \boxdot$. Luego, $o = o' \simeq_{\sigma_*} p' = p$. Más aún, los objetos son necesariamente comandos. Se analiza la regla aplicada:
  - $\simeq_{\sigma_4}$. Luego, $o = [\alpha']\,(\mu\alpha.[\beta']\,(\mu\beta.c)\,v)\,u$ y $p = [\beta']\,(\mu\beta.[\alpha']\,(\mu\alpha.c)\,u)\,v$ con $\alpha \notin v$, $\beta \notin u$, $\beta \neq \alpha'$ y $\alpha \neq \beta'$. Más aún, sean α'' y β'' nombres fresco, entonces $\mathfrak{C}(o) = \mathfrak{C}([\alpha']\,\mu\alpha''.([\beta']\,\mu\beta''.c[\![\beta \setminus^{\beta''} v]\!][\![\alpha \setminus^{\alpha''} u]\!]) \simeq_\rho \mathfrak{C}((c[\![\beta \setminus^{\beta''} v]\!][\beta'' \setminus \beta'])[\![\alpha \setminus^{\alpha''} u]\!][\alpha'' \setminus \alpha'])$. A su vez, sobre p también se tiene $\mathfrak{C}(p) = \mathfrak{C}([\beta']\,\mu\beta''.([\alpha']\,\mu\alpha''.c[\![\alpha \setminus^{\alpha''} u]\!])[\![\beta \setminus^{\beta''} v]\!]) \simeq_\rho \mathfrak{C}((c[\![\alpha \setminus^{\alpha''} u]\!][\alpha'' \setminus \alpha'])[\![\beta \setminus^{\beta''} v]\!][\beta'' \setminus \beta'])$. Finalmente, por Lem. 4.6.2 (2), se obtiene $\mathfrak{C}((c[\![\beta \setminus^{\beta''} v]\!][\beta'' \setminus \beta'])[\![\alpha \setminus^{\alpha''} u]\!][\alpha'' \setminus \alpha']) \simeq \mathfrak{C}((c[\![\alpha \setminus^{\alpha''} u]\!][\alpha'' \setminus \alpha'])[\![\beta \setminus^{\beta''} v]\!][\beta'' \setminus \beta'])$. Se concluye dado que $\simeq \subseteq \simeq_{\mathtt{er}}$.
  - $\simeq_{\sigma_5}$. Luego, $o = [\alpha']\,(\mu\alpha.[\beta']\,\lambda y.\mu\beta.c)\,u$ y $p = [\beta']\,\lambda y.\mu\beta.[\alpha']\,(\mu\alpha.c)\,u$ con $y \notin u$, $\beta \notin u$, $\beta \neq \alpha'$ y $\alpha \neq \beta'$. Más aún, sea α'' un nombre fresco, entonces $\mathfrak{C}(o) = \mathfrak{C}([\alpha']\,\mu\alpha''.([\beta']\,\lambda y.\mu\beta.c)[\![\alpha \setminus^{\alpha''} u]\!]) \simeq_\rho \mathfrak{C}(([\beta']\,\lambda y.\mu\beta.c)[\![\alpha \setminus^{\alpha''} u]\!][\alpha'' \setminus \alpha'])$. Del mismo modo, se tiene $\mathfrak{C}(p) = \mathfrak{C}([\beta']\,\lambda y.\mu\beta.[\alpha']\,\mu\alpha''.c[\![\alpha \setminus^{\alpha''} u]\!]) \simeq_\rho \mathfrak{C}([\beta']\,\lambda y.\mu\beta.c[\![\alpha \setminus^{\alpha''} u]\!][\alpha'' \setminus \alpha'])$. Finalmente, por Lem. 4.6.2 (2), se tiene también $\mathfrak{C}(([\beta']\,\lambda y.\mu\beta.c)[\![\alpha \setminus^{\alpha''} u]\!][\alpha'' \setminus \alpha']) \simeq \mathfrak{C}([\beta']\,\lambda y.\mu\beta.c[\![\alpha \setminus^{\alpha''} u]\!][\alpha'' \setminus \alpha'])$. Se concluye dado que $\simeq \subseteq \simeq_{\mathtt{er}}$.
  - $\simeq_{\sigma_6}$. Luego, $o = [\alpha']\,\lambda x.\mu\alpha.[\beta']\,\lambda y.\mu\beta.c$ y $p = [\beta']\,\lambda y.\mu\beta.[\alpha']\,\lambda x.\mu\alpha.c$ con $\beta \neq \alpha'$ y $\alpha \neq \beta'$. Más aún, $\mathfrak{C}(o) = [\alpha']\,\lambda x.\mu\alpha.[\beta']\,\lambda y.\mu\beta.\mathfrak{C}(c)$ y $\mathfrak{C}(p) = [\beta']\,\lambda y.\mu\beta.[\alpha']\,\lambda x.\mu\alpha.\mathfrak{C}(c)$. Se concluye por $\simeq_{\mathtt{pp}}$.
  - $\simeq_{\sigma_7}$. Luego, $o = [\beta]\,\mu\alpha.c$ y $p = \{\alpha \setminus \beta\}c$. Más aún, $\mathfrak{C}(o) = [\beta]\,\mu\alpha.\mathfrak{C}(c)$ y $\mathfrak{C}(p) = \{\alpha \setminus \beta\}\mathfrak{C}(c)$. Se concluye por $\simeq_\rho$ y $\simeq_{\mathtt{ren}}$. Notar que éste es el único caso es que es necesario apelar a la nueva reglar $\simeq_{\mathtt{ren}}$.
  - $\simeq_{\sigma_8}$. Luego, $o = \mu\alpha.[\alpha]\,t$ y $p = t$ con $\alpha \notin t$. Más aún, $\mathfrak{C}(o) = \mu\alpha.[\alpha]\,\mathfrak{C}(t)$. Se concluye por $\simeq_\theta$.
- $\mathtt{O} = \mathtt{T}\,u$. Luego, $o = t\,u$ y $p = t'\,u$ con $t \simeq_\sigma t'$. Por *h.i.* se tiene $\mathfrak{C}(t) \simeq_{\mathtt{er}} \mathfrak{C}(t')$. Se concluye por Lem. C.0.49 con $\mathtt{O}$. Notar que, por Teo. 4.5.2, para todo contexto $\mathtt{O}$ y objecto o vale $\mathfrak{C}(\mathtt{O}\langle\mathfrak{C}(o)\rangle) = \mathfrak{C}(\mathtt{O}\langle o\rangle)$.
- $\mathtt{O} = t\,\mathtt{T}$. Luego, $o = t\,u$ y $p = t\,u'$ con $u \simeq_\sigma u'$. Por *h.i.* se tiene $\mathfrak{C}(u) \simeq_{\mathtt{er}} \mathfrak{C}(u')$. Se concluye por Lem. C.0.49 con $\mathtt{O}$.
- $\mathtt{O} = \lambda x.\mathtt{T}$. Luego, $o = \lambda x.t$ y $p = \lambda x.t'$ con $t \simeq_\sigma t'$. Por *h.i.* se tiene $\mathfrak{C}(t) \simeq_{\mathtt{er}} \mathfrak{C}(t')$. Se concluye por Lem. C.0.49 con $\mathtt{O}$.
- $\mathtt{O} = \mu\alpha.\mathtt{C}$. Luego, $o = \mu\alpha.c$ y $p = \mu\alpha.c'$ con $c \simeq_\sigma c'$. Por *h.i.* se tiene $\mathfrak{C}(c) \simeq_{\mathtt{er}} \mathfrak{C}(c')$. Se concluye por Lem. C.0.49 con $\mathtt{O}$.
- $\mathtt{O} = [\alpha]\,\mathtt{T}$. Luego, $o = [\alpha]\,t$ y $p = [\alpha]\,t'$ con $t \simeq_\sigma t'$. Por *h.i.* se tiene $\mathfrak{C}(t) \simeq_{\mathtt{er}} \mathfrak{C}(t')$. Se concluye por Lem. C.0.49 con $\mathtt{O}$.

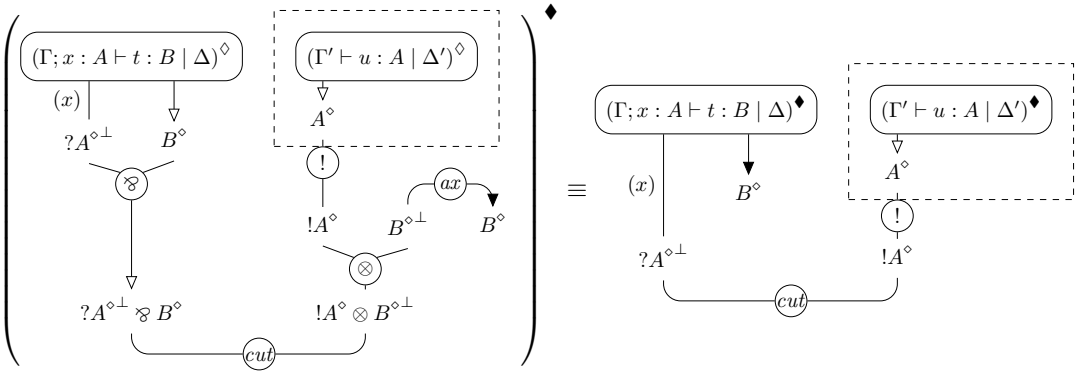
□

Lema 4.7.3. *Sean $o, p \in \mathbb{O}_{\Lambda M}$ tipados. Si $o \to_{\mathfrak{C}} p$, entonces $(\pi_o)^{\blacklozenge} \equiv (\pi_p)^{\blacklozenge}$, donde π_o y π_p son las correspondientes derivaciones de tipo relacionadas.*

Demostración. Por definición, $o \to_{\mathfrak{C}} p$ implica $o = \mathtt{O}\langle l\rangle$ y $p = \mathtt{O}\langle r\rangle$ con $l \mapsto_* r$, $* \in \{\mathtt{dB}, \mathtt{dM}, \mathtt{P}, \mathtt{W}\}$. La demostración es por inducción en $\mathtt{O}$. Se detallan únicamente los casos base con $\mathtt{O} = \Box$ y $\mathtt{O} = \boxdot$, los restantes concluyen directamente de la *h.i.* por equivalencia estructural de proof-nets.

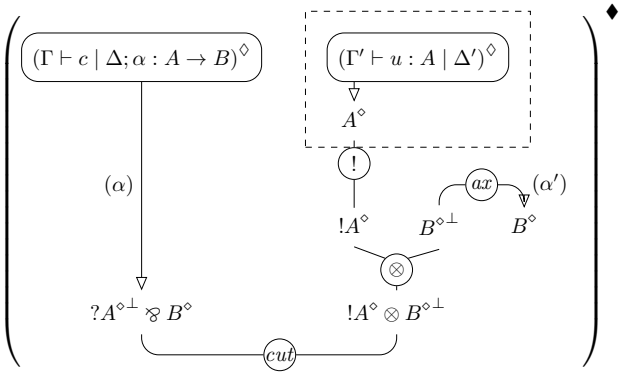
- $\mathtt{O} = \Box$. Luego, se tienen dos casos posibles según la regla de reescritura aplicada al reducir.

  1. $\mathtt{dB}$. Luego, $o = \mathtt{L}\langle\lambda x.t\rangle\, u$ y $o' = \mathtt{L}\langle t[x \setminus u]\rangle$ con $\mathtt{fc}(u, \mathtt{L})$. Se procede por inducción en $\mathtt{L}$. Se ilustra a continuación el caso base con $\mathtt{L} = \Box$ y $x \in \mathtt{fv}(t)$. El análisis para $x \notin \mathtt{fv}(t)$ es similar. El caso $\mathtt{L} \neq \Box$ se sigue inmediatamente de la *h.i.* por equivalencia estructural de proof-nets.



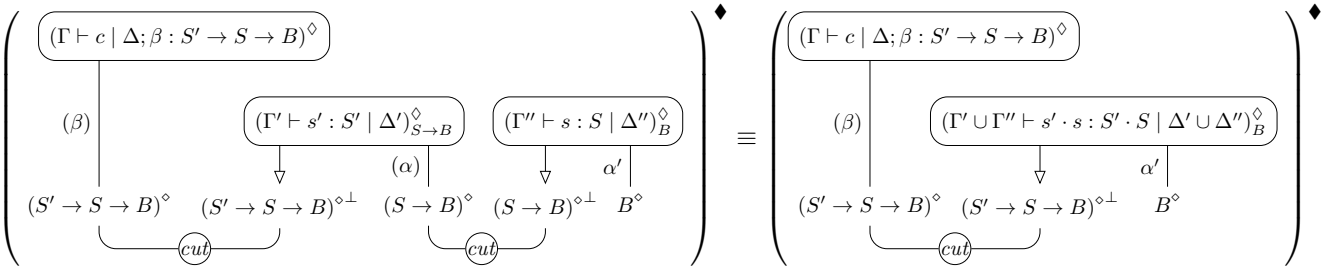
  donde cada lado de la equivalencia estrutural es la expansión de $(\pi_o)^{\blacklozenge}$ y $(\pi_p)^{\blacklozenge}$ respectivamente. Notar que el cut multiplicativo $\mathtt{C}(\otimes, \parr)$ en $(\pi_o)^{\blacklozenge}$ es contraído por la traducción, resultando en el cut exponencial $\mathtt{C}(!, d)$, que se ilustra en $(\pi_p)^{\blacklozenge}$, y en un segundo cut multiplicavo $\mathtt{C}(ax)$ entre las fórmulas $B^{\Diamond}$ y $B^{\Diamond\perp}$, que es a su vez contraído por la traducción.

  2. $\mathtt{dM}$. Luego, $o = \mathtt{L}\langle\mu\alpha.c\rangle\, u$ y $o' = \mathtt{L}\langle\mu\alpha'.c[\![\alpha \setminus^{\alpha'} u]\!]\rangle$ con α' un nombre fresco. Se procede por inducción en $\mathtt{L}$. Se ilustra a continuación el caso base con $\mathtt{L} = \Box$ y $\alpha \in \mathtt{fn}(c)$. El análisis para $\alpha \notin \mathtt{fn}(c)$ es similar. El caso $\mathtt{L} \neq \Box$ se sigue inmediatamente de la *h.i.* por equivalencia estructural de proof-nets. En este caso, $(\pi_o)^{\blacklozenge}$ y $(\pi_o)^{\blacklozenge}$ son ambos estructuralmente equivalentes, por definición, a



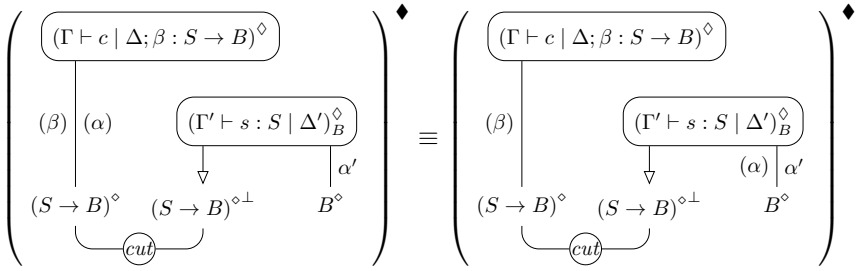
- $\mathtt{O} = \boxdot$. Luego, se tienen dos casos posibles según la regla de reescritura aplicada al reducir.

1. $\mathtt{N}$. Luego, $o = \mathtt{LCC}\langle[\alpha]\,t\rangle[\![\alpha \setminus^{\alpha'} s]\!]$ y $o' = \mathtt{LCC}\langle[\alpha']\,t :: s\rangle$ con $\alpha \notin \mathtt{LCC}$ y $\alpha \notin t$. Se procede por inducción en $\mathtt{LCC}$. Se ilustra en la Fig. C.1 el caso base con $\mathtt{LCC} = \boxdot$ y $\alpha' \notin \mathsf{fn}(s)$. El análisis para los casos base restantes es similar. Los casos con $\mathtt{LCC} \neq \Box$ se siguen inmediatamente de la *h.i.* por equivalencia estructural de proof-nets. Sea $s = u_0 \cdot \ldots \cdot u_n$, comienza con $(\Gamma; \Gamma_0; \ldots; \Gamma_n \vdash ([\alpha]\,t)[\![\alpha \setminus^{\alpha'} s]\!] \mid \Delta; \Delta_0; \ldots; \Delta_n; \alpha' : B)^{\blacklozenge}$ que se expande por definicion a la primera proof-net de la figura, mientras que la última proof-net ilustra la expansión de $(\Gamma; \Gamma_0; \ldots; \Gamma_n \vdash [\alpha']\,t :: s \mid \Delta; \Delta_0; \ldots; \Delta_n; \alpha' : B)^{\blacklozenge}$.
2. $\mathtt{P}$. Luego, $o = \mathtt{LCC}\langle c[\![\beta \setminus^{\alpha} s']\!]\rangle[\![\alpha \setminus^{\alpha'} s]\!]$ y $o' = \mathtt{LCC}\langle c[\![\beta \setminus^{\alpha'} s' \cdot s]\!]\rangle$ con $\alpha \notin \mathtt{LCC}$, $\alpha \notin c$ y $\alpha \notin s'$. Se procede por inducción en $\mathtt{LCC}$. Se ilustra a continuación el caso base con $\mathtt{LCC} = \boxdot$, $\beta \in \mathsf{fn}(c)$ y $\alpha' \notin \mathsf{fn}(s)$. El análisis para los casos base restantes es similar. Los casos con $\mathtt{LCC} \neq \Box$ se siguen inmediatamente de la *h.i.* por equivalencia estructural de proof-nets.



donde cada lado de la equivalencia estrutural es la expansión de $(\pi_o)^{\blacklozenge}$ y $(\pi_p)^{\blacklozenge}$ respectivamente. Vale la pena notar que el cut entre el hilo α en $(\Gamma' \vdash s' : S' \mid \Delta')^{\Diamond}_{S \to B}$ y la conclusión distinguida de $(\Gamma'' \vdash s : S \mid \Delta'')^{\Diamond}_{B}$ es multiplicativo. Recordar que los stack son interpretados en $\otimes$-trees, cuyo caso base es un nodo axioma. Este cut es contraído al computar la forma normal multiplicativa de las proof-nets en cuestión, resultando en un nuevo $\otimes$-tree equivalente a $(\Gamma' \cup \Gamma'' \vdash s' \cdot s : S' \cdot S \mid \Delta' \cup \Delta'')^{\Diamond}_{B}$.

3. $\mathtt{W}$. Luego, $o = \mathtt{LCC}\langle c[\beta \setminus \alpha]\rangle[\![\alpha \setminus^{\alpha'} s]\!]$ y $o' = \mathtt{LCC}\langle c[\![\beta \setminus^{\alpha} s]\!][\alpha \setminus \alpha']\rangle$ con $\alpha \notin \mathtt{LCC}$, $\alpha \notin c$. Se ilustra a continuación el caso base con $\mathtt{LCC} = \boxdot$, $\beta \in \mathsf{fn}(c)$ y $\alpha' \notin \mathsf{fn}(s)$. El análisis para los casos base restantes es similar. Los casos con $\mathtt{LCC} \neq \Box$ se siguen inmediatamente de la *h.i.* por equivalencia estructural de proof-nets.



donde cada lado de la equivalencia estrutural es la expansión de $(\pi_o)^{\blacklozenge}$ y $(\pi_p)^{\blacklozenge}$ respectivamente. Notar que la única diferencia entre ambas proof-nets es la ubicación original del nombre borrado α.

□

Lema 4.7.4. *Sean $o, p \in \mathbb{O}_{\Lambda M}$ tipados. Si $o \simeq_{\mathtt{er}} p$, entonces $(\pi_o)^{\blacklozenge} \equiv (\pi_p)^{\blacklozenge}$, donde π_o y π_p son las correspondientes derivaciones de tipo relacionadas.*

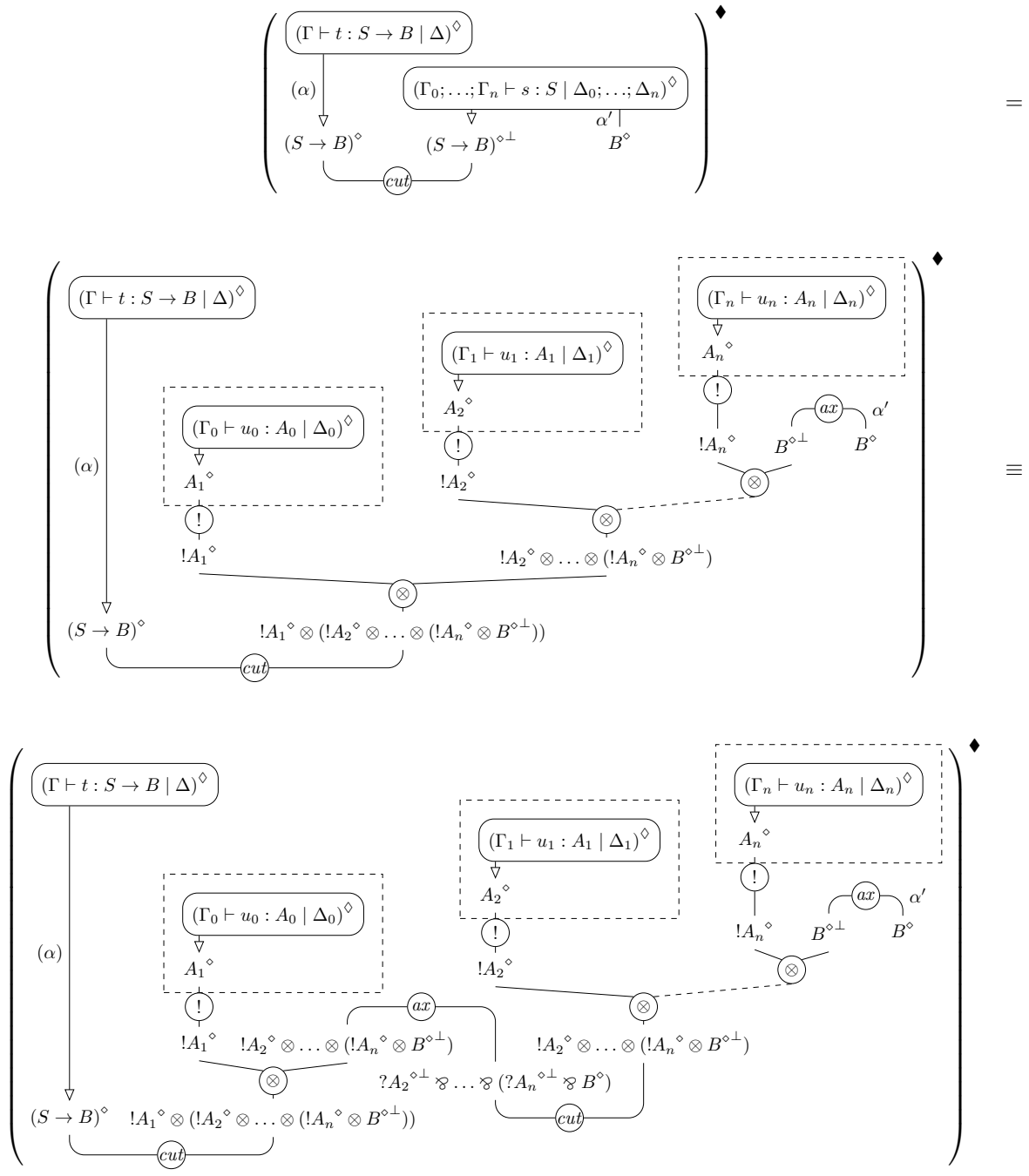


Figura C.1: Caso para la regla N de la prueba del Lem. 4.7.3.

Demostración. La prueba es por inducción en $o \simeq_{\mathtt{er}} p$. Por definición esto implica verificar cuatro condiciones: reflexividad, transitividad, simetría y congruencia (*i.e.* clausura por contextos):

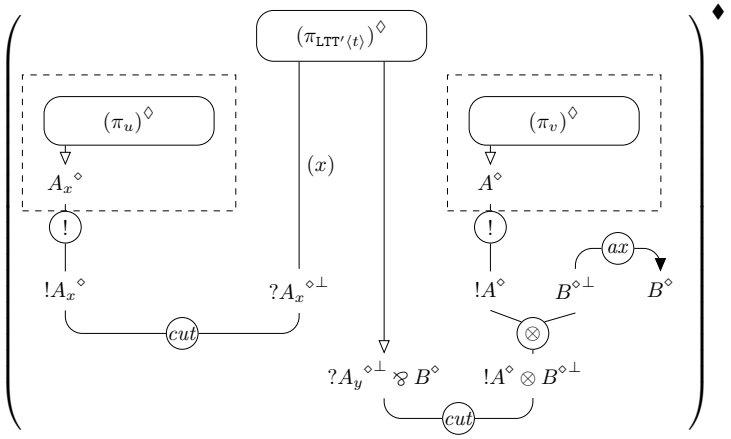
Reflexividad Luego, $p = o$ por lo que el resultado es inmediato por reflexividad de $\equiv$.

Transitividad Luego, existe $q \in \mathbb{O}_{\Lambda M}$ tal que $o \simeq_{\mathtt{er}} q$ y $q \simeq_{\mathtt{er}} p$. Por *h.i.* se tienen $(\pi_o)^{\blacklozenge} \equiv (\pi_q)^{\blacklozenge}$ y $(\pi_q)^{\blacklozenge} \equiv (\pi_p)^{\blacklozenge}$. Se concluye por transitivadad de $\equiv$.

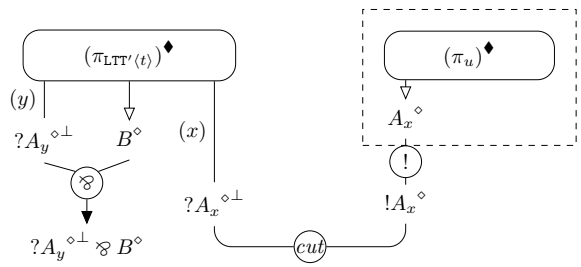
Simetría Luego, $p \simeq_{\mathtt{er}} o$ y por *h.i.* se tiene $(\pi_p)^{\blacklozenge} \equiv (\pi_o)^{\blacklozenge}$. Se concluye por simetría de $\equiv$.

Congruencia La congruencia se demuestra por inducción en el contexto de clausura $\mathtt{O}$ tal que $o = \mathtt{O}\langle o' \rangle$ y $p = \mathtt{O}\langle p' \rangle$ con $o' \simeq_* p'$, donde $\simeq_*$ es una regla de la Fig. 4.13 o $\simeq_{\mathtt{ren}}$. Se detallan únicamente los casos base con $\mathtt{O} = \Box$ y $\mathtt{O} = \boxdot$, los restantes concluyen directamente de la *h.i.* por equivalencia estructural de proof-nets.

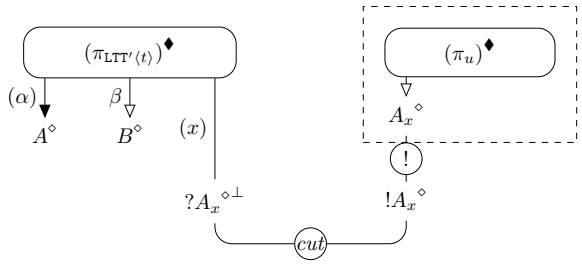
- $\mathtt{O} = \Box$. Luego, $o = o' \simeq_* p' = p$. Más aún, los objetos son necesariamente términos. Se analiza la regla aplicada:
  - $\simeq_{\mathtt{exsubs}}$. Luego, $o = \mathtt{LTT}\langle t \rangle[x \setminus u]$ y $p = \mathtt{LTT}\langle t[x \setminus u] \rangle$ con $x \notin \mathtt{LTT}$ y $\mathsf{fc}(u, \mathtt{LTT})$. Se procede analizando la forma de $\mathtt{LTT}$:
    - $\mathtt{LTT} = \Box$. Luego, se tiene $o = p$ y el resultado es inmediato por reflexividad de $\equiv$.
    - $\mathtt{LTT} = \mathtt{LTT}'\, v$ con $x \notin v$ y $\mathsf{fc}(u, v)$. Se ilustra a continuación el caso $x \in \mathsf{fv}(t)$. El análisis para $x \notin \mathsf{fv}(t)$ es similar. Luego, $(\pi_o)^{\blacklozenge}$ y $(\pi_p)^{\blacklozenge}$ son ambos estructuralmente equivalentes, por definición, a



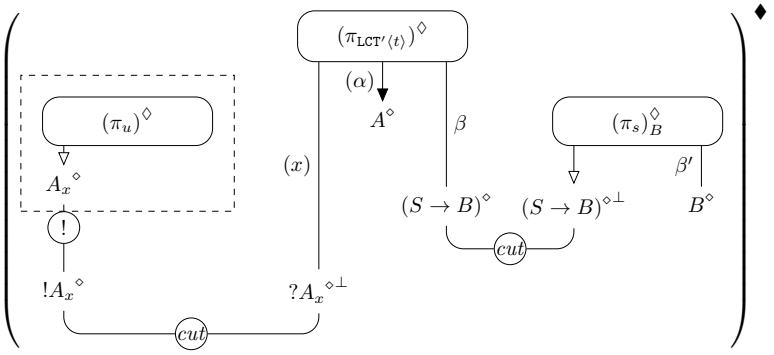
    - $\mathtt{LTT} = \lambda y.\mathtt{LTT}'$ con $y \notin \mathsf{fv}(u)$. Se ilustra a continuación el caso $x, y \in \mathsf{fv}(t)$. El análisis para los casos restantes es similar. Luego, $(\pi_o)^{\blacklozenge}$ y $(\pi_p)^{\blacklozenge}$ son ambos estructuralmente equivalentes, por definición, a



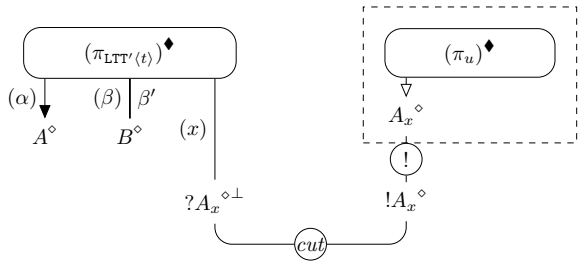
- $\mathtt{LTT} = \mu\alpha.\mathtt{LCT}$. Se analiza la forma de $\mathtt{LCT}$:
  - $\mathtt{LCT} = [\beta]\,\mathtt{LTT}'$ con $\alpha, \beta \notin \mathsf{fn}(u)$. Se ilustra a continuación el caso $x \in \mathsf{fv}(t)$, $\alpha \in \mathsf{fn}(t)$ y $\beta \notin \mathsf{fn}(t)$. El análisis para los casos restantes es similar. Luego, $(\pi_o)^{\blacklozenge}$ y $(\pi_p)^{\blacklozenge}$ son ambos estructuralmente equivalentes, por definición, a



  - $\mathtt{LCT} = \mathtt{LCT}'[\![\beta \setminus^{\beta'} s]\!]$ con $x \notin s$ y $\alpha, \beta \notin \mathsf{fn}(u)$. Se ilustra a continuación el caso donde β' es fresca. El análisis para los casos restantes es similar. Luego, $(\pi_o)^{\blacklozenge}$ y $(\pi_p)^{\blacklozenge}$ son ambos estructuralmente equivalentes, por definición, a

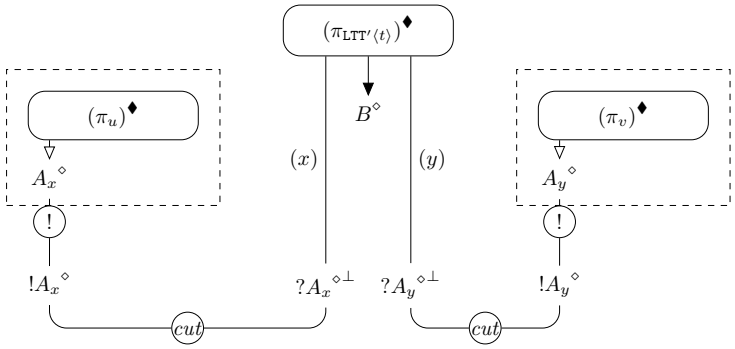


  - $\mathtt{LCT} = \mathtt{LCT}'[\beta \setminus \beta']$ con $\alpha, \beta \notin \mathsf{fn}(u)$. Se ilustra a continuación el caso $x \in \mathsf{fv}(t)$, $\alpha \in \mathsf{fn}(t)$ y $\beta \notin \mathsf{fn}(t)$. El análisis para los casos restantes es similar. Luego, $(\pi_o)^{\blacklozenge}$ y $(\pi_p)^{\blacklozenge}$ son ambos estructuralmente equivalentes, por definición, a



- $\mathtt{LTT} = \mathtt{LTT}'[x \setminus v]$ con $x \notin v$ e $y \notin \mathsf{fv}(u)$. Se ilustra a continuación el caso $x, y \in \mathsf{fv}(t)$. El análisis para los casos restantes es similar. Luego, $(\pi_o)^{\blacklozenge}$ y $(\pi_p)^{\blacklozenge}$ son ambos

estructuralmente equivalentes, por definición, a



- $\simeq_\theta$. Luego, $o = \mu\alpha.[\alpha]\, t$ y $p = t$ con $\alpha \notin t$. El resultado es inmediato dado que, por definición, se tiene $(\pi_{\mu\alpha.[\alpha]\, t})^{\Diamond} = (\pi_t)^{\Diamond}$.

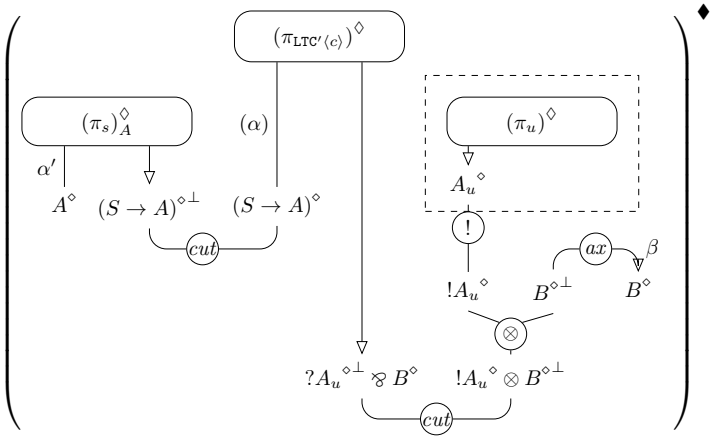
- $\mathtt{0} = \boxdot$. Luego, $o = o' \simeq_* p' = p$. Más aún, los objetos son necesariamente comandos. Se analiza la regla aplicada:

  - $\simeq_{\mathtt{exrepl}}$. Luego, $o = \mathtt{LCC}\langle c\rangle[\![\alpha \setminus^{\alpha'} s]\!]$ y $p = \mathtt{LCC}\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle$ con $\alpha \notin \mathtt{LCC}$, $\mathsf{fc}(\alpha', \mathtt{LCC})$ y $\mathsf{fc}(s, \mathtt{LCC})$. Se procede analizando la forma de $\mathtt{LCC}$:

    - $\mathtt{LCC} = \boxdot$. Luego, se tiene $o = p$ y el resultado es inmediato por reflexividad de $\equiv$.

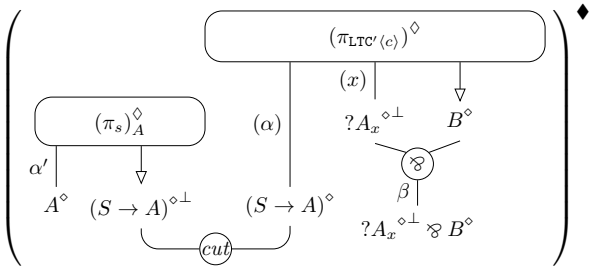
    - $\mathtt{LCC} = [\beta]\, \mathtt{LTC}$. Se analiza la forma de $\mathtt{LTC}$:

      - $\mathtt{LTC} = \mathtt{LTC}'\, u$ con $\alpha \notin u$. Se ilustra a continuación el caso $\alpha \in \mathsf{fn}(c)$, $\alpha' \notin \mathsf{fn}(c)$ y $\beta \neq \gamma$. El análisis para los casos restantes es similar. Luego, $(\pi_o)^{\blacklozenge}$ y $(\pi_p)^{\blacklozenge}$ son ambos estructuralmente equivalentes, por definición, a

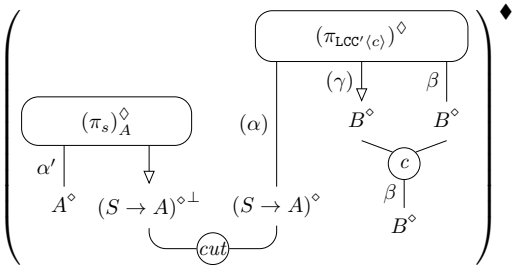


      - $\mathtt{LTC} = \lambda x.\mathtt{LTC}'$ con $x \notin \mathsf{fv}(s)$. Se ilustra a continuación el caso $x \in \mathsf{fv}(c)$, $\alpha \in \mathsf{fn}(c)$ y $\alpha' \notin \mathsf{fn}(c)$. El análisis para los casos restantes es similar. Luego, $(\pi_o)^{\blacklozenge}$ y $(\pi_p)^{\blacklozenge}$

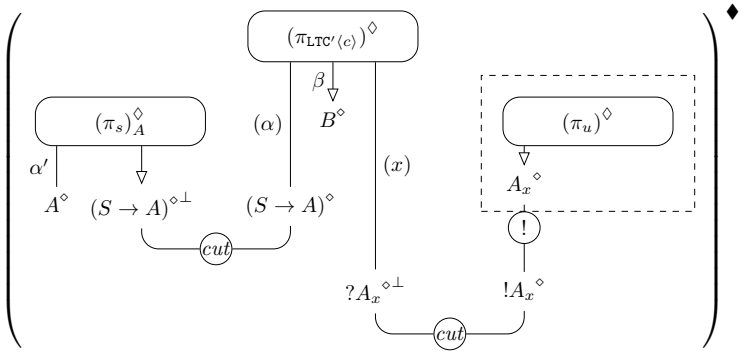
son ambos estructuralmente equivalentes, por definición, a



- $\mathtt{LTC} = \mu\gamma.\mathtt{LCC}'$ con $\gamma \notin s$. Se ilustra a continuación el caso $\alpha, \beta, \gamma \in \mathsf{fn}(c)$, $\alpha' \notin \mathsf{fn}(c)$ y $\beta \neq \gamma$. El análisis para los casos restantes es similar. Luego, $(\pi_o)^{\blacklozenge}$ y $(\pi_p)^{\blacklozenge}$ son ambos estructuralmente equivalentes, por definición, a



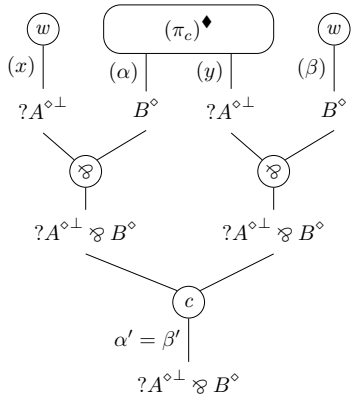
- $\mathtt{LTC} = \mathtt{LTC}'[x \setminus u]$ con $\alpha \notin u$ y $x \notin \mathsf{fv}(s)$. Se ilustra a continuación el caso $x \in \mathsf{fv}(c)$, $\alpha \in \mathsf{fn}(c)$ y $\alpha' \notin \mathsf{fn}(c)$. El análisis para los casos restantes es similar. Luego, $(\pi_o)^{\blacklozenge}$ y $(\pi_p)^{\blacklozenge}$ son ambos estructuralmente equivalentes, por definición, a



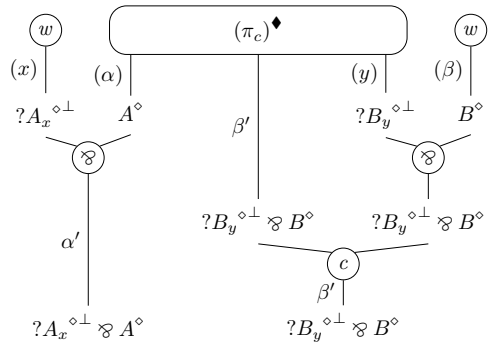
- $\simeq_{\mathtt{exren}}$. Luego, $o = \mathtt{LCC}\langle c\rangle[\alpha \setminus \beta]$ y $p = \mathtt{LCC}\langle c[\alpha \setminus \beta]\rangle$ con $\alpha \notin \mathtt{LCC}$ y $\mathsf{fc}(\beta, \mathtt{LCC})$. El único caso interesante es cuando $\alpha \in \mathsf{fn}(c)$ y $\beta \in \mathsf{fn}(\mathtt{LCC})$. El resultado se sigue del hecho que la traducción del renaming explícito solo agrega un nodo de contracción entre α y β, el cual puede ser permutado por equivalencia estructural de proof-nets obteniendo $(\pi_o)^{\blacklozenge} \equiv (\pi_p)^{\blacklozenge}$. Los casos restantes son inmediatos puesto que solo se etiquetan hílos agregando, a lo sumo, un nodo de weakening final.
- $\simeq_{\mathtt{pp}}$. Luego, $o = [\alpha']\,\lambda x.\mu\alpha.[\beta']\,\lambda y.\mu\beta.c$ y $p = [\beta']\,\lambda y.\mu\beta.[\alpha']\,\lambda x.\mu\alpha.c$ con $\alpha \neq \beta'$ y $\beta \neq \alpha'$. Se tienen diferentes posibilidades dependiendo de la ocurrencia o no de cada variable/nombre en el comando c. Se ilustran a continuación solo dos casos donde $x \notin \mathsf{fv}(c)$, $y \in \mathsf{fv}(c)$, $\alpha \in \mathsf{fn}(c)$ y $\beta \notin \mathsf{fn}(c)$. El análisis para los casos restantes es similar.

Notar que la rama que involucra a x, α y α' es paralela a la de y, β y β', y solo se contraen al final si $\alpha' = \beta'$.

1. $\alpha' = \beta'$ con $\alpha' \notin \mathsf{fn}(c)$. Luego, $(\pi_o)^{\blacklozenge}$ y $(\pi_p)^{\blacklozenge}$ son ambos iguales, por definición, a



2. $\alpha' \neq \beta'$ con $\alpha' \notin \mathsf{fn}(c)$ y $\beta' \in \mathsf{fn}(c)$. Luego, $(\pi_o)^{\blacklozenge}$ y $(\pi_p)^{\blacklozenge}$ son ambos iguales, por definición, a



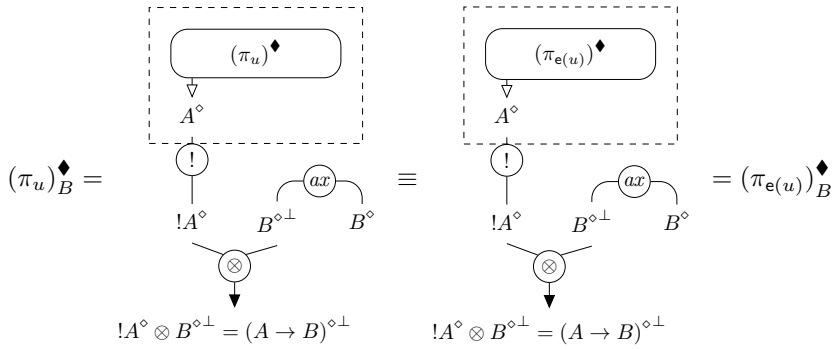
- $\simeq_\rho$. Luego, $o = [\alpha]\,\mu\beta.c$ y $p = c[\alpha\backslash\beta]$. El resultado es inmediato dado que, por definición, se tiene $(\pi_{[\alpha]\,\mu\beta.c})^{\Diamond} = (\pi_{c[\alpha\backslash\beta]})^{\Diamond}$.
- $\simeq_{\mathtt{ren}}$. Luego, $o = c[\alpha\backslash\beta]$ y $p = \{\alpha \setminus \beta\}c$. Se concluye por Lem. C.0.25, pues $(\pi_{\{\alpha\backslash\beta\}c})^{\Diamond} = (\pi_c)^{\Diamond,\alpha,\beta} = (\pi_{c[\alpha\backslash\beta]})^{\Diamond}$ por definición.

□

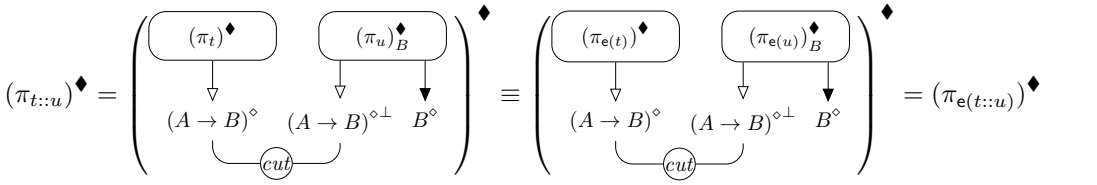
Lema C.0.50. *Sean $t \in \mathbb{T}_{\Lambda M}$ y $s \in \mathbb{O}_{\Lambda M}$ un stack tal que existen derivaciones π_t y π_s asignando tipos $S \to B$ y S respectivamente. Luego, $(\pi_t)^{\blacklozenge} \equiv (\pi_{\mathsf{e}(t)})^{\blacklozenge}$ y $(\pi_s)^{\blacklozenge}_B \equiv (\pi_{\mathsf{e}(s)})^{\blacklozenge}_B$ implican $(\pi_{t::s})^{\blacklozenge} \equiv (\pi_{\mathsf{e}(t::s)})^{\blacklozenge}$.*

Demostración. Por inducción en s.

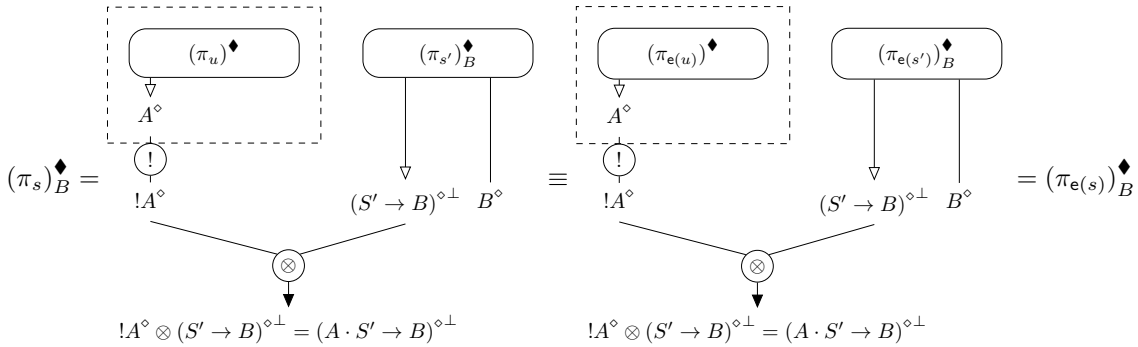
- $s = u$. Luego, $t :: u = t\,u$ y $S = A$. Más aún, por Def. 4.4.2, se tiene



  lo que, en particular, implica $(\pi_u)^{\blacklozenge} \equiv (\pi_{\mathsf{e}(u)})^{\blacklozenge}$. Luego, se concluye dado que $\mathsf{e}(t :: u) = \mathsf{e}(t)\,\mathsf{e}(u)$



- $s = u \cdot s'$. Luego, $t :: s = (t\,u) :: s'$ y $S = A \cdot S'$. Más aún, por Def. 4.4.2, se tiene



  lo que implica $(\pi_u)^{\blacklozenge} \equiv (\pi_{\mathsf{e}(u)})^{\blacklozenge}$ y $(\pi_{s'})^{\blacklozenge}_B \equiv (\pi_{\mathsf{e}(s')})^{\blacklozenge}_B$. Notar que $\mathsf{e}(t :: s) = \mathsf{e}(t) :: (\mathsf{e}(u) \cdot \mathsf{e}(s')) = (\mathsf{e}(t)\,\mathsf{e}(u)) :: \mathsf{e}(s') = \mathsf{e}((t\,u) :: s')$. Luego, de $(\pi_t)^{\blacklozenge} \equiv (\pi_{\mathsf{e}(t)})^{\blacklozenge}$ y $(\pi_u)^{\blacklozenge} \equiv (\pi_{\mathsf{e}(u)})^{\blacklozenge}$ se obtiene $(\pi_{t\,u})^{\blacklozenge} \equiv (\pi_{\mathsf{e}(t\,u)})^{\blacklozenge}$ por Def. 4.4.2. Finalmente, por *h.i.* con $(\pi_{s'})^{\blacklozenge}_B \equiv (\pi_{\mathsf{e}(s')})^{\blacklozenge}_B$, se concluye $(\pi_{t::s})^{\blacklozenge}_B \equiv (\pi_{\mathsf{e}(t::s)})^{\blacklozenge}_B$.

□

Lema 4.7.5.

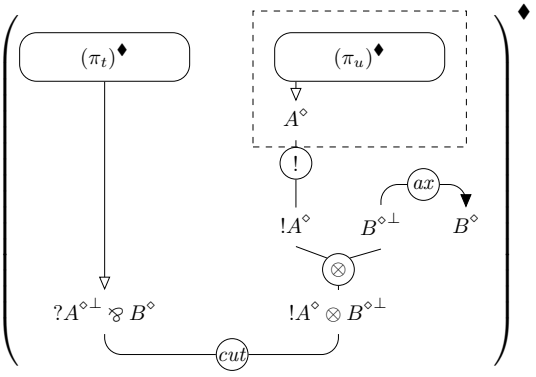
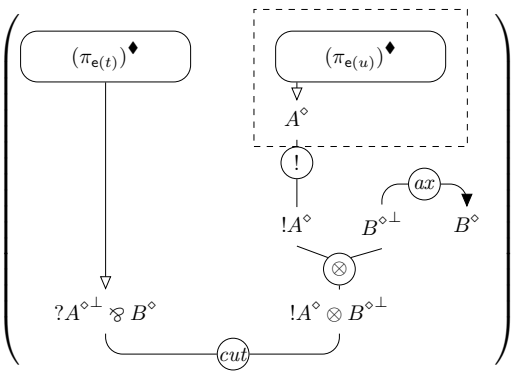
1. *Sea $o \in \mathbb{O}_{\lambda\mu}$. Luego, $(\pi_o)^{\blacklozenge} \equiv (\pi_{\mathfrak{C}(o)})^{\blacklozenge}$, donde π_o y $\pi_{\mathfrak{C}(o)}$ son las correspondientes derivaciones de tipo relacionadas.*
2. *Sea $o \in \mathbb{O}_{\Lambda M}$. Luego, $(\pi_o)^{\blacklozenge} \equiv (\pi_{\mathsf{e}(o)})^{\blacklozenge}$, donde π_o y $\pi_{\mathsf{e}(o)}$ son las correspondientes derivaciones de tipo relacionadas.*

Demostración. 1. Por inducción en el número n de pasos de reducción aplicados para computar $\mathfrak{C}(o)$: *i.e.* $o \to_{\mathfrak{C}}^{n} \mathfrak{C}(o)$. Recordar que $\mathbb{T}_{\lambda\mu} \subset \mathbb{T}_{\Lambda M}$ por definición.

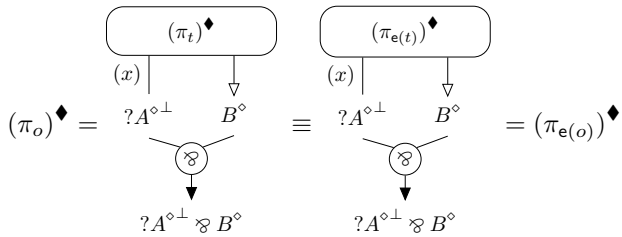
- $n = 0$. Luego, $o = \mathfrak{C}(o)$ y se concluye por reflexividad de $\equiv$.
- $n > n$. Luego, $o \to_{\mathfrak{C}} o' \to_{\mathfrak{C}}^{n-1} \mathfrak{C}(o)$. Por *h.i.* se tiene $\pi_{o'} \equiv \pi_{\mathfrak{C}(o)}$. Más aún, por Lem. 4.7.3, $\pi_o \equiv \pi_{o'}$. Se concluye por transitividad de $\equiv$.

2. Por inducción en o.

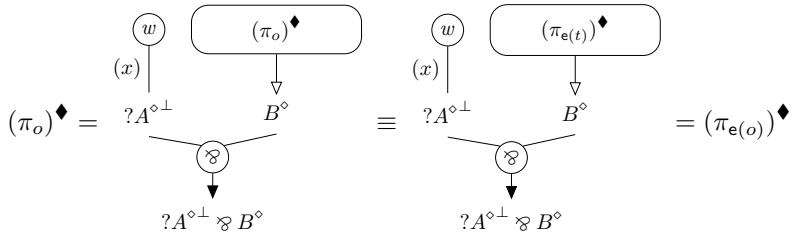
- $o = x$. Luego, $o = \mathsf{e}(o)$ y se concluye por reflexividad de $\equiv$.
- $o = t\,u$. Luego, $\mathsf{e}(o) = \mathsf{e}(t)\,\mathsf{e}(u)$. Por *h.i.* se tienen $(\pi_t)^{\blacklozenge} \equiv (\pi_{\mathsf{e}(t)})^{\blacklozenge}$ y $(\pi_u)^{\blacklozenge} \equiv (\pi_{\mathsf{e}(u)})^{\blacklozenge}$, por lo que se concluye

$(\pi_o)^{\blacklozenge} =$  $\equiv$  $= (\pi_{\mathsf{e}(o)})^{\blacklozenge}$

- $o = \lambda x.t$. Luego, $\mathsf{e}(o) = \lambda x.\mathsf{e}(t)$. Por *h.i.* se tiene $(\pi_t)^{\blacklozenge} \equiv (\pi_{\mathsf{e}(t)})^{\blacklozenge}$. Si $x \in \mathsf{fv}(t)$, entonces $x \in \mathsf{fv}(\mathsf{e}(t))$ por lo que

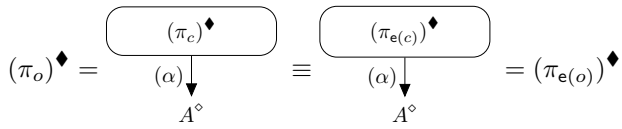


Si no (*i.e.* $x \notin \mathsf{fv}(t)$), se tiene también $x \notin \mathsf{fv}(\mathsf{e}(t))$. Luego, se concluye

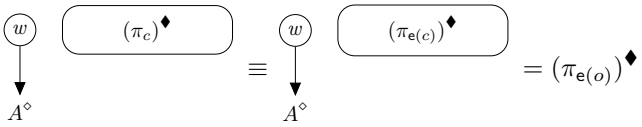


- $o = \mu\alpha.c$. Luego, $\mathsf{e}(o) = \mu\alpha.\mathsf{e}(c)$. Por *h.i.* se tiene $(\pi_c)^{\blacklozenge} \equiv (\pi_{\mathsf{e}(c)})^{\blacklozenge}$. Si $\alpha \in \mathsf{fn}(c)$, entonces

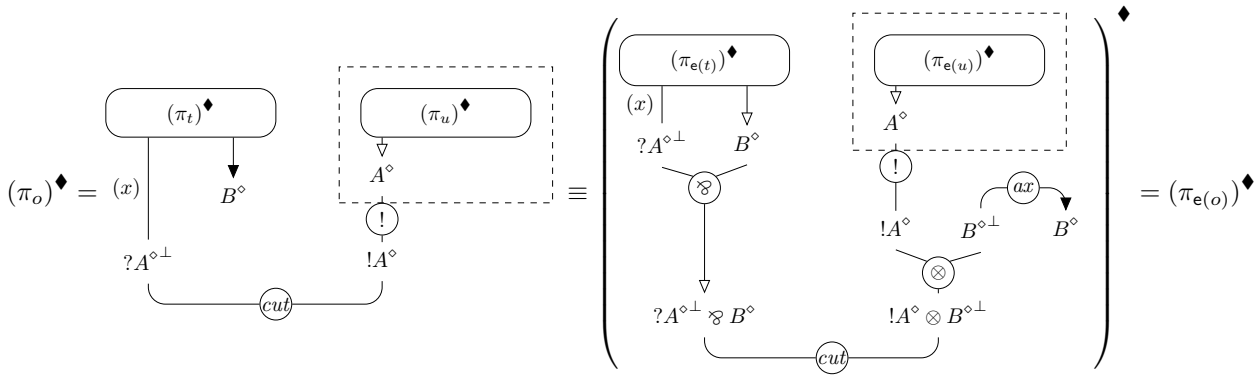
$\alpha \in \mathsf{fv}(\mathsf{e}(c))$ por lo que



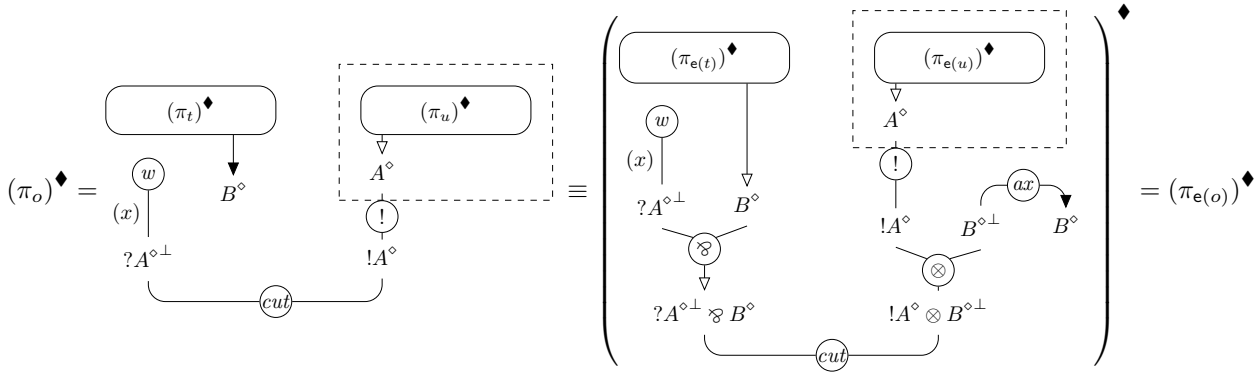
Si no (*i.e.* $\alpha \notin \mathsf{fn}(c)$), se tiene también $\alpha \notin \mathsf{fn}(\mathsf{e}(c))$. Luego, se concluye



- $o = t[x \setminus u]$. Luego, $\mathsf{e}(o) = (\lambda x.\mathsf{e}(t))\,\mathsf{e}(u)$. Por *h.i.* se tienen $(\pi_t)^\blacklozenge \equiv (\pi_{\mathsf{e}(t)})^\blacklozenge$ y $(\pi_u)^\blacklozenge \equiv (\pi_{\mathsf{e}(u)})^\blacklozenge$. Si $x \in \mathsf{fv}(t)$, entonces $x \in \mathsf{fv}(\mathsf{e}(t))$ por lo que

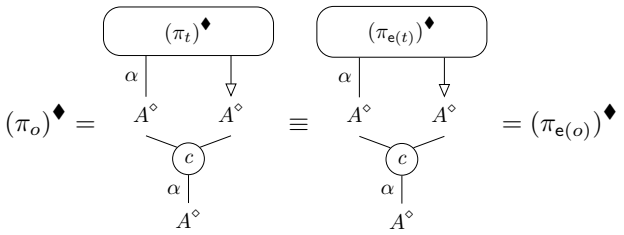


Notar que el cut multiplicativo $\mathtt{C}(\otimes, \parr)$ en $(\pi_{\mathsf{e}(o)})^\blacklozenge$ es contraído por la traducción, resultando en el cut exponencial que se ilustra en $(\pi_o)^\blacklozenge$, y en un segundo cut multiplicavo $\mathtt{C}(ax)$ entre las fórmulas $B^\diamond$ y $B^{\diamond\perp}$. Este último es a su vez contraído por la traducción. Si $x \notin \mathsf{fv}(t)$, se tiene también $x \notin \mathsf{fv}(\mathsf{e}(t))$ y la situación es similar

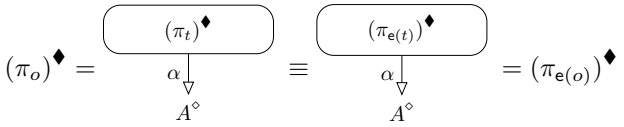


- $o = [\alpha]\,t$. Luego, $\mathsf{e}(o) = [\alpha]\,\mathsf{e}(t)$. Por *h.i.* se tiene $(\pi_t)^\blacklozenge \equiv (\pi_{\mathsf{e}(t)})^\blacklozenge$. Si $\alpha \in \mathsf{fn}(t)$, entonces

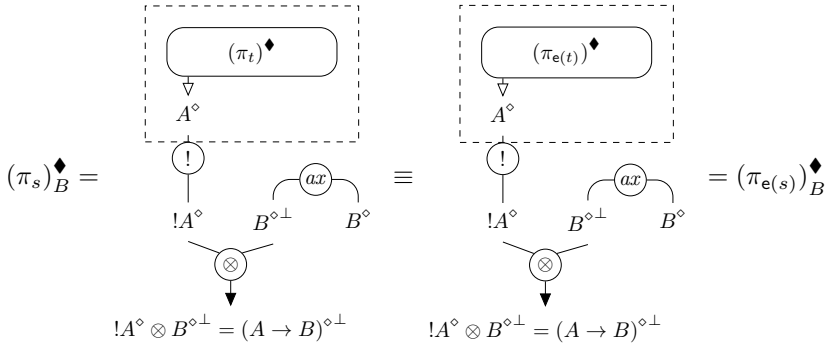
$\alpha \in \mathsf{fv}(\mathsf{e}(t))$ por lo que



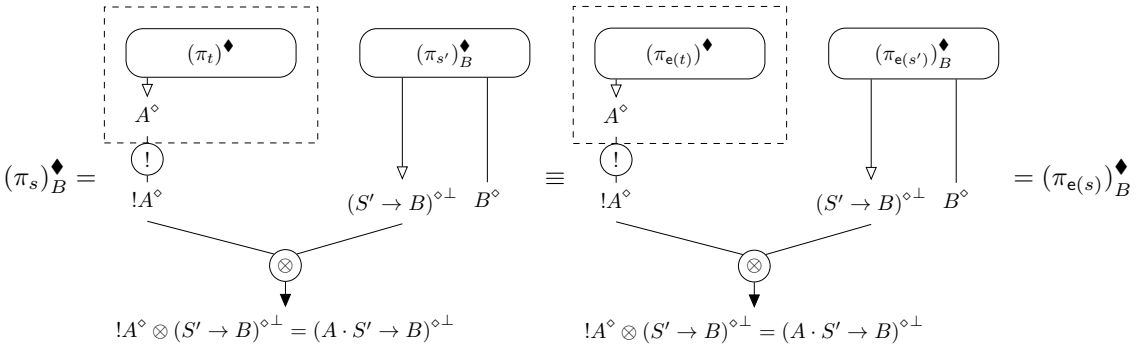
Si no (*i.e.* $\alpha \notin \mathsf{fn}(t)$), se tiene también $\alpha \notin \mathsf{fn}(\mathsf{e}(t))$. Luego, se concluye



- $o = c[\![\alpha \setminus^{\alpha'} s]\!]$. Luego, $\mathsf{e}(o) = [\alpha']\,(\mu\alpha.\mathsf{e}(c)) :: \mathsf{e}(s)$. Por *h.i.* se tiene $(\pi_c)^{\blacklozenge} \equiv (\pi_{\mathsf{e}(c)})^{\blacklozenge}$. Más aún, por α-conversión se asume $\alpha \notin s$. Se prueba $s^{\blacklozenge}_B \equiv \mathsf{e}(s)^{\blacklozenge}_B$ por inducción en s.
  - $s = t$. Por *h.i.* se tiene $(\pi_t)^{\blacklozenge} \equiv (\pi_{\mathsf{e}(t)})^{\blacklozenge}$. Luego

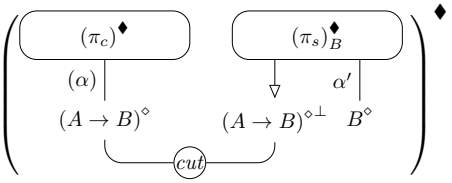


  - $s = t \cdot s'$. Aplicando ambas *h.i.* se tienen $(\pi_t)^{\blacklozenge} \equiv (\pi_{\mathsf{e}(t)})^{\blacklozenge}$ y $(\pi_{s'})^{\blacklozenge}_B \equiv (\pi_{\mathsf{e}(s')})^{\blacklozenge}_B$ respectivamente. Luego

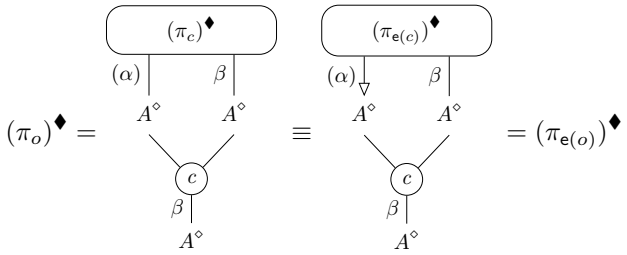


  Luego, se tienen $(\pi_c)^{\blacklozenge} \equiv (\pi_{\mathsf{e}(c)})^{\blacklozenge}$ y $s^{\blacklozenge}_B \equiv \mathsf{e}(s)^{\blacklozenge}_B$. Más aún, por Def. 4.4.2, de $\mathsf{e}(\mu\alpha.c) = \mu\alpha.\mathsf{e}(c)$ se obtiene $(\pi_{\mu\alpha.c})^{\blacklozenge} \equiv (\pi_{\mathsf{e}(\mu\alpha.c)})^{\blacklozenge}$. Por Lem. C.0.50 se tiene entonces $(\pi_{(\mu\alpha.c)::s})^{\blacklozenge} \equiv$

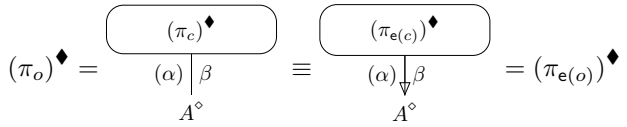
$(\pi_{\mathsf{e}((\mu\alpha.c)::s)})^{\blacklozenge}$, y nuevamente por Def. 4.4.2 se obtiene $(\pi_{[\alpha']\,(\mu\alpha.c)::s})^{\blacklozenge} \equiv (\pi_{\mathsf{e}([\alpha']\,(\mu\alpha.c)::s)})^{\blacklozenge} = (\pi_{\mathsf{e}(o)})^{\blacklozenge}$. Se concluye pues, $\pi_o{}^{\blacklozenge}$ y $(\pi_{[\alpha']\,(\mu\alpha.c)::s})^{\blacklozenge}$ son ambos estructuralmente equivalentes a



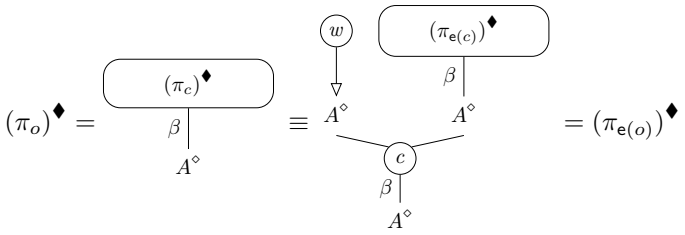
- $o = c[\alpha \setminus \beta]$. Luego, $\mathsf{e}(o) = [\beta]\,\mu\alpha.\mathsf{e}(c)$. Por *h.i.* se tiene $(\pi_c)^{\blacklozenge} \equiv (\pi_{\mathsf{e}(c)})^{\blacklozenge}$. Si $\alpha, \beta \in \mathsf{fn}(c)$, entonces $\alpha, \beta \in \mathsf{fn}(\mathsf{e}(c))$ por lo que



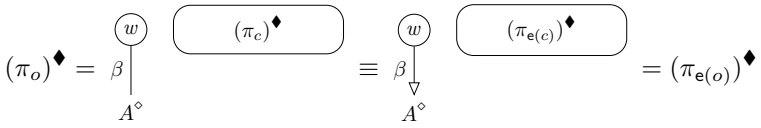
Así mismo, si $\alpha \in \mathsf{fn}(c)$ y $\beta \notin \mathsf{fn}(c)$, esto implica $\alpha \in \mathsf{fn}(\mathsf{e}(c))$ y $\beta \notin \mathsf{fn}(\mathsf{e}(c))$. Luego



Si $\alpha \notin \mathsf{fn}(c)$ y $\beta \in \mathsf{fn}(c)$, se tienen $\alpha \notin \mathsf{fn}(\mathsf{e}(c))$ y $\beta \in \mathsf{fn}(\mathsf{e}(c))$, por lo que



Finalmente, si $\alpha, \beta \notin \mathsf{fn}(c)$, entonces $\alpha, \beta \notin \mathsf{fn}(\mathsf{e}(c))$ y se concluye



□

El resultado de bisimulación fuerte

Lema C.0.51. *Sean $o, o' \in \mathbb{O}_{\Lambda M}$ tal que $o \simeq_* o'$ con $\simeq_*$ una regla de Fig. 4.13. Si $o = \mathtt{LXC}\langle c\rangle$ con $\alpha \notin \mathtt{LXC}$, $\mathsf{fc}(\alpha', \mathtt{LXC})$ y $\mathsf{fc}(s, \mathtt{LXC})$, luego $o' = \mathtt{LXC}'\langle c'\rangle$ tal que $\alpha \notin \mathtt{LXC}'$, $\mathsf{fc}(\alpha', \mathtt{LXC}')$, $\mathsf{fc}(s, \mathtt{LXC}')$ y $\mathfrak{C}(\mathtt{LXC}\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle) \simeq \mathfrak{C}(\mathtt{LXC}'\langle c'[\![\alpha \setminus^{\alpha'} s]\!]\rangle)$.*

Demostración. Por análisis de casos sobre $\mathtt{LXC}$.

- $\mathtt{LXC} = \boxdot$. Luego, $o = c$. Hay cuatro posibles reglas para comandos:

  - $\simeq_{\mathtt{exrepl}}$. Luego, $o = \mathtt{LCC}\langle c'\rangle[\![\gamma \setminus^{\delta} s']\!]$ y $o' = \mathtt{LCC}\langle c'[\![\gamma \setminus^{\delta} s']\!]\rangle$ con $\gamma \notin \mathtt{LCC}$, $\mathsf{fc}(\delta, \mathtt{LCC})$ y $\mathsf{fc}(s', \mathtt{LCC})$. Por α-conversión se asumen también $\mathsf{fc}(\alpha', \mathtt{LCC})$ y $\mathsf{fc}(s, \mathtt{LCC})$. Luego, se tienen cuatro casos posibles:

    1. $o[\![\alpha \setminus^{\alpha'} s]\!] \to_{\mathtt{p}} \mathtt{LCC}\langle c\rangle[\![\gamma \setminus^{\alpha'} s' \cdot s]\!]$ (*i.e.* $\alpha = \delta$, $\delta \notin c$, $\delta \notin \mathtt{LCC}$ y $\delta \notin s'$). Luego, $\mathfrak{C}(o[\![\alpha \setminus^{\alpha'} s]\!]) = \mathfrak{C}(\mathtt{LCC}\langle c\rangle[\![\gamma \setminus^{\alpha'} s' \cdot s]\!])$. Más aún, se tiene $o'[\![\alpha \setminus^{\alpha'} s]\!] \to_{\mathtt{p}} \mathtt{LCC}\langle c[\![\gamma \setminus^{\alpha'} s' \cdot s]\!]\rangle$, por lo que $\mathfrak{C}(o'[\![\alpha \setminus^{\alpha'} s]\!]) = \mathfrak{C}(\mathtt{LCC}\langle c[\![\gamma \setminus^{\alpha'} s' \cdot s]\!]\rangle)$. Luego, por Lem. 4.6.2 (2), se tiene $\mathfrak{C}(\mathtt{LCC}\langle c\rangle[\![\gamma \setminus^{\alpha'} s' \cdot s]\!]) \simeq \mathfrak{C}(\mathtt{LCC}\langle c[\![\gamma \setminus^{\alpha'} s' \cdot s]\!]\rangle)$. Se concluye con $\mathtt{LXC}' = \boxdot$ y $c' = o'$.
    2. $o[\![\alpha \setminus^{\alpha'} s]\!] \to_{\mathtt{NPW}} \mathtt{LCC}'\langle c\rangle[\![\gamma \setminus^{\delta} s']\!]$ (*i.e.* $\mathsf{fn}_{\alpha}(o) = 1$, $\alpha \neq \delta$, $\alpha \notin c$ y $\alpha \notin s'$). Luego, $\mathfrak{C}(o[\![\alpha \setminus^{\alpha'} s]\!]) = \mathfrak{C}(\mathtt{LCC}'\langle c\rangle[\![\gamma \setminus^{\delta} s']\!])$. Más aún, también se tiene $o'[\![\alpha \setminus^{\alpha'} s]\!] \to_{\mathtt{NPW}} \mathtt{LCC}'\langle c[\![\gamma \setminus^{\delta} s']\!]\rangle$, por lo que $\mathfrak{C}(o'[\![\alpha \setminus^{\alpha'} s]\!]) = \mathfrak{C}(\mathtt{LCC}'\langle c[\![\gamma \setminus^{\delta} s']\!]\rangle)$. Por Lem. 4.6.2 (2), se tiene $\mathfrak{C}(\mathtt{LCC}'\langle c\rangle[\![\gamma \setminus^{\delta} s']\!]) \simeq \mathfrak{C}(\mathtt{LCC}'\langle c[\![\gamma \setminus^{\delta} s']\!]\rangle)$. Se concluye con $\mathtt{LXC}' = \boxdot$ y $c' = o'$.
    3. $o[\![\alpha \setminus^{\alpha'} s]\!] \to_{\mathtt{NPW}} \mathtt{LCC}\langle c'\rangle[\![\gamma \setminus^{\delta} s']\!]$ (*i.e.* $\mathsf{fn}_{\alpha}(o) = 1$, $\alpha \neq \delta$, $\alpha \notin \mathtt{LCC}$ y $\alpha \notin s'$). Este caso es análogo al anterior.
    4. De lo contrario, $\mathfrak{C}(o[\![\alpha \setminus^{\alpha'} s]\!]) = o[\![\alpha \setminus^{\alpha'} \mathfrak{C}(s)]\!] \simeq o'[\![\alpha \setminus^{\alpha'} \mathfrak{C}(s)]\!] = \mathfrak{C}(o'[\![\alpha \setminus^{\alpha'} s]\!])$, pues $o \simeq o'$ implica $\mathfrak{C}(o) = o$ y $\mathfrak{C}(o') = o'$. Se concluye con $\mathtt{LXC}' = \boxdot$ y $c' = o'$.

  - $\simeq_{\mathtt{exren}}$. Luego, $o = \mathtt{LCC}\langle c\rangle[\gamma \setminus \delta]$ y $o' = \mathtt{LCC}\langle c[\gamma \setminus \delta]\rangle$ con $\gamma \notin \mathtt{LCC}$ y $\mathsf{fc}(\delta, \mathtt{LCC})$. Por α-conversión se asumen también $\mathsf{fc}(\alpha', \mathtt{LCC})$ y $\mathsf{fc}(s, \mathtt{LCC})$. Luego, se tienen cuatro casos posibles:

    1. $o[\![\alpha \setminus^{\alpha'} s]\!] \to_{\mathtt{W}} \mathtt{LCC}\langle c\rangle[\![\gamma \setminus^{\alpha} s]\!][\alpha \setminus \alpha']$ (*i.e.* $\alpha = \delta$, $\delta \notin c$ y $\delta \notin \mathtt{LCC}$). Luego, $\mathfrak{C}(o[\![\alpha \setminus^{\alpha'} s]\!]) = \mathfrak{C}(\mathtt{LCC}\langle c\rangle[\![\gamma \setminus^{\alpha} s]\!][\alpha \setminus \alpha'])$. Más aún, también se tiene $o'[\![\alpha \setminus^{\alpha'} s]\!] \to_{\mathtt{W}} \mathtt{LCC}\langle c[\![\gamma \setminus^{\alpha} s]\!][\alpha \setminus \alpha']\rangle$, por lo que $\mathfrak{C}(o'[\![\alpha \setminus^{\alpha'} s]\!]) = \mathfrak{C}(\mathtt{LCC}\langle c[\![\gamma \setminus^{\alpha} s]\!][\alpha \setminus \alpha']\rangle)$. Luego, por Lem. 4.6.2 (2), se tiene $\mathfrak{C}(\mathtt{LCC}\langle c\rangle[\![\gamma \setminus^{\alpha} s]\!][\alpha \setminus \alpha']) \simeq \mathfrak{C}(\mathtt{LCC}\langle c[\![\gamma \setminus^{\alpha} s]\!][\alpha \setminus \alpha']\rangle)$. Se concluye con $\mathtt{LXC}' = \boxdot$ y $c' = o'$.
    2. $o[\![\alpha \setminus^{\alpha'} s]\!] \to_{\mathtt{NPW}} \mathtt{LCC}'\langle c\rangle[\gamma \setminus \delta]$ (*i.e.* $\mathsf{fn}_{\alpha}(o) = 1$, $\alpha \neq \delta$ y $\alpha \notin c$). Luego, $\mathfrak{C}(o[\![\alpha \setminus^{\alpha'} s]\!]) = \mathfrak{C}(\mathtt{LCC}'\langle c\rangle[\gamma \setminus \delta])$. Más aún, también se tiene $o'[\![\alpha \setminus^{\alpha'} s]\!] \to_{\mathtt{NPW}} \mathtt{LCC}'\langle c[\gamma \setminus \delta]\rangle$, por lo que $\mathfrak{C}(o'[\![\alpha \setminus^{\alpha'} s]\!]) = \mathfrak{C}(\mathtt{LCC}'\langle c[\gamma \setminus \delta]\rangle)$. Por Lem. 4.6.2 (2), se tiene $\mathfrak{C}(\mathtt{LCC}'\langle c\rangle[\gamma \setminus \delta]) \simeq \mathfrak{C}(\mathtt{LCC}'\langle c[\gamma \setminus \delta]\rangle)$. Se concluye con $\mathtt{LXC}' = \boxdot$ y $c' = o'$.
    3. $o[\![\alpha \setminus^{\alpha'} s]\!] \to_{\mathtt{NPW}} \mathtt{LCC}\langle c'\rangle[\gamma \setminus \delta]$ (*i.e.* $\mathsf{fn}_{\alpha}(o) = 1$, $\alpha \neq \delta$ y $\alpha \notin \mathtt{LCC}$). Este caso es análogo al anterior.
    4. De lo contrario, $\mathfrak{C}(o[\![\alpha \setminus^{\alpha'} s]\!]) = o[\![\alpha \setminus^{\alpha'} \mathfrak{C}(s)]\!] \simeq o'[\![\alpha \setminus^{\alpha'} \mathfrak{C}(s)]\!] = \mathfrak{C}(o'[\![\alpha \setminus^{\alpha'} s]\!])$, pues $o \simeq o'$ implica $\mathfrak{C}(o) = o$ y $\mathfrak{C}(o') = o'$. Se concluye con $\mathtt{LXC}' = \boxdot$ y $c' = o'$.

  - $\simeq_{\mathtt{pp}}$. Luego, $o = [\gamma']\,\lambda x.\mu\gamma.[\delta']\,\lambda y.\mu\delta.c$ y $o' = [\delta']\,\lambda y.\mu\delta.[\gamma']\,\lambda x.\mu\gamma.c$ con $\delta \neq \gamma'$ y $\gamma \neq \delta'$. Por α-conversión se asumen también $x \notin s$, $y \notin s$, $\gamma \neq \alpha'$, $\gamma \notin s$, $\delta \neq \alpha'$ y $\delta \notin s$. Luego, se tienen cuatro casos posibles.

    1. $o[\![\alpha \setminus^{\alpha'} s]\!] \to_{\mathtt{N}} [\alpha']\,(\lambda x.\mu\gamma.[\delta']\,\lambda y.\mu\delta.c) :: s$ (*i.e.* $\alpha = \gamma'$, $\alpha \neq \delta'$ y $\gamma' \notin c$). Asumir $s = u \cdot s'$ (el caso $s = u$ es ligeramente más simple) y considerar nombres frescos γ'', δ''. Luego, $\mathfrak{C}(o[\![\alpha \setminus^{\alpha'} s]\!]) = \mathfrak{C}([\alpha']\,(\mu\gamma''.([\delta']\,\lambda y.\mu\delta.c)[\![\gamma \setminus^{\gamma''} s']\!])[x \setminus u])$. Del mismo modo, $\mathfrak{C}(o'[\![\alpha \setminus^{\alpha'} s]\!]) = \mathfrak{C}([\delta']\,\lambda y.\mu\delta.[\alpha']\,(\mu\gamma''.c'[\![\gamma \setminus^{\gamma''} s']\!])[x \setminus u])$. En el caso $s = u$ simplemente se evita el replacement explícito. Luego, por Lem. 4.6.2, se tiene por un lado $\mathfrak{C}(o[\![\alpha \setminus^{\alpha'} s]\!]) \simeq \mathfrak{C}([\alpha']\,\mu\gamma''.[\delta']\,(\lambda y.\mu\delta.c'[\![\gamma \setminus^{\gamma''} s']\!])[x \setminus u])$ y por otro $\mathfrak{C}(o'[\![\alpha \setminus^{\alpha'} s]\!]) \simeq \mathfrak{C}([\delta']\,(\lambda y.\mu\delta.[\alpha']\,\mu\gamma''.c[\![\gamma \setminus^{\gamma''} s']\!])[x \setminus u])$, apelando al ítem (1) del lema para posicionar la sustitución explícita y al ítem (2) para el replacement. Finalmente, aplicando dos veces $\simeq_{\rho}$ y Lem. 4.6.2 (2) para posicionar adecuadamente el renaming explícito introducido

por la regla, se tiene

$$\begin{array}{rcl}
\mathfrak{C}(o[\![\alpha \setminus^{\alpha'} s]\!]) & \simeq & \mathfrak{C}([\alpha']\,\mu\gamma''.[\delta']\,(\lambda y.\mu\delta.c'[\![\gamma \setminus^{\gamma''} s']\!])[x \setminus u]) \\
& = & [\alpha']\,\mu\gamma''.[\delta']\,(\lambda y.\mu\delta.\mathfrak{C}(c'[\![\gamma \setminus^{\gamma''} s']\!]))[x \setminus \mathfrak{C}(u)] \\
& \simeq_\rho & ([\delta']\,(\lambda y.\mu\delta.\mathfrak{C}(c'[\![\gamma \setminus^{\gamma''} s']\!]))[x \setminus \mathfrak{C}(u)])[\gamma'' \setminus \alpha'] \\
& \simeq & [\delta']\,(\lambda y.\mu\delta.\mathfrak{C}(c'[\![\gamma \setminus^{\gamma''} s']\!])[\gamma'' \setminus \alpha'])[x \setminus \mathfrak{C}(u)] \\
& \simeq_\rho & [\delta']\,(\lambda y.\mu\delta.[\alpha']\,\mu\gamma''.\mathfrak{C}(c'[\![\gamma \setminus^{\gamma''} s']\!]))[x \setminus \mathfrak{C}(u)] \\
& = & \mathfrak{C}([\delta']\,(\lambda y.\mu\delta.[\alpha']\,\mu\gamma''.c'[\![\gamma \setminus^{\gamma''} s']\!])[x \setminus u]) \\
& \simeq & \mathfrak{C}(o'[\![\alpha \setminus^{\alpha'} s]\!])
\end{array}$$

Se concluye con $\mathtt{LXC}' = \boxdot$ y $c' = o'$.

2. $o[\![\alpha \setminus^{\alpha'} s]\!] \to_{\mathtt{N}} [\gamma']\,\lambda x.\mu\gamma.[\alpha']\,(\lambda y.\mu\delta.c) :: s$ (*i.e.* $\alpha \neq \gamma'$, $\alpha = \delta'$ y $\delta' \notin c$). Este caso es análogo al anterior.
3. $o[\![\alpha \setminus^{\alpha'} s]\!] \to_{\mathtt{NPW}} [\gamma']\,\lambda x.\mu\gamma.[\delta']\,\lambda y.\mu\delta.c'$ (*i.e.* $\mathsf{fn}_\alpha(o) = 1$, $\alpha \neq \gamma'$ y $\alpha \neq \delta'$). Luego, $\mathfrak{C}(o[\![\alpha \setminus^{\alpha'} s]\!]) = [\gamma']\,\lambda x.\mu\gamma.[\delta']\,\lambda y.\mu\delta.\mathfrak{C}(c') \simeq_{\mathtt{pp}} [\delta']\,\lambda y.\mu\delta.[\gamma']\,\lambda x.\mu\gamma.\mathfrak{C}(c') = \mathfrak{C}(o'[\![\alpha \setminus^{\alpha'} s]\!])$. Se concluye con $\mathtt{LXC}' = \boxdot$ y $c' = o'$.
4. De lo contrario, $\mathfrak{C}(o[\![\alpha \setminus^{\alpha'} s]\!]) = o[\![\alpha \setminus^{\alpha'} \mathfrak{C}(s)]\!] \simeq o'[\![\alpha \setminus^{\alpha'} \mathfrak{C}(s)]\!] = \mathfrak{C}(o'[\![\alpha \setminus^{\alpha'} s]\!])$, pues $o \simeq o'$ implica $\mathfrak{C}(o) = o$ y $\mathfrak{C}(o') = o'$. Se concluye con $\mathtt{LXC}' = \boxdot$ y $c' = o'$.

- $\simeq_\rho$. Luego, $o = [\delta]\,\mu\gamma.c$ y $o' = c[\gamma \setminus \delta]$. Por α-conversión se asumen $\gamma \neq \alpha'$ y $\gamma \notin s$. Luego, se tienen tres casos posibles.
  1. $o[\![\alpha \setminus^{\alpha'} s]\!] \to_{\mathtt{N}} [\alpha']\,(\mu\gamma.c) :: s$ (*i.e.* $\alpha = \delta$ y $\delta \notin c$). Luego, se tiene $\mathfrak{C}(o[\![\alpha \setminus^{\alpha'} s]\!]) = [\alpha']\,\mathfrak{C}((\mu\gamma.c) :: s) = [\alpha']\,\mu\alpha.\mathfrak{C}(c[\![\gamma \setminus^{\alpha} s]\!])$. Notar que α es fresca en $(\mu\gamma.c) :: s$. Más aún, se tiene también $o'[\![\alpha \setminus^{\alpha'} s]\!] \to_{\mathtt{W}} c[\![\gamma \setminus^{\alpha} s]\!][\alpha \setminus \alpha']$. Se concluye por $\simeq_\rho$ con $\mathtt{LXC}' = \boxdot$ y $c' = o'$, dado que $\mathfrak{C}(o[\![\alpha \setminus^{\alpha'} s]\!]) \simeq_\rho \mathfrak{C}(c[\![\gamma \setminus^{\alpha} s]\!])[\alpha \setminus \alpha'] = \mathfrak{C}(c[\![\gamma \setminus^{\alpha} s]\!][\alpha \setminus \alpha']) = \mathfrak{C}(o'[\![\alpha \setminus^{\alpha'} s]\!])$
  2. $o[\![\alpha \setminus^{\alpha'} s]\!] \to_{\mathtt{NPW}} [\delta]\,\mu\gamma.c'$ (*i.e.* $\mathsf{fn}_\alpha(o) = 1$ y $\alpha \neq \delta$). Luego, $\mathfrak{C}(o[\![\alpha \setminus^{\alpha'} s]\!]) = [\delta]\,\mu\gamma.\mathfrak{C}(c') \simeq_\rho \mathfrak{C}(c')[\gamma \setminus \delta] = \mathfrak{C}(o'[\![\alpha \setminus^{\alpha'} s]\!])$. Se concluye con $\mathtt{LXC}' = \boxdot$ y $c' = o'$.
  3. De lo contrario, $\mathfrak{C}(o[\![\alpha \setminus^{\alpha'} s]\!]) = o[\![\alpha \setminus^{\alpha'} \mathfrak{C}(s)]\!] \simeq o'[\![\alpha \setminus^{\alpha'} \mathfrak{C}(s)]\!] = \mathfrak{C}(o'[\![\alpha \setminus^{\alpha'} s]\!])$, pues $o \simeq o'$ implica $\mathfrak{C}(o) = o$ y $\mathfrak{C}(o') = o'$. Se concluye con $\mathtt{LXC}' = \boxdot$ y $c' = o'$.

- $\mathtt{LXC} = \mathtt{LTC}\,u$. Este caso es inmediato dado que no hay solapamiento con ninguna regla.
- $\mathtt{LXC} = \lambda x.\mathtt{LTC}$. Este caso es inmediato dado que no hay solapamiento con ninguna regla.
- $\mathtt{LXC} = \mu\gamma.\mathtt{LCC}$. Luego, la única regla aplicable es $\simeq_\theta$. Entonces, $\mathtt{LCC} = [\gamma]\,\mathtt{LTC}$ con $\gamma \notin c$ y $\gamma \notin \mathtt{LTC}$. Más aún, por hipótesis $\mathsf{fc}(\alpha', \mathtt{LXC})$ y $\mathsf{fc}(s, \mathtt{LXC})$, lo que implica $\gamma \neq \alpha'$ y $\gamma \notin s$. Se concluye por $\simeq_\theta$ con $\mathtt{LXC}' = \mathtt{LTC}$ y $c' = c$, $\mathfrak{C}(\mathtt{LXC}\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle) = \mu\gamma.[\gamma]\,\mathfrak{C}(\mathtt{LTC}\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle) \simeq_\theta \mathfrak{C}(\mathtt{LTC}\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle) = \mathfrak{C}(\mathtt{LXC}'\langle c'[\![\alpha \setminus^{\alpha'} s]\!]\rangle)$.
- $\mathtt{LXC} = \mathtt{LTC}[x \setminus u]$. Luego, la única regla aplicable es $\simeq_{\mathtt{exsubs}}$. Se tiene entonces $\mathtt{LTC}\langle c\rangle = \mathtt{LTT}\langle t\rangle$ con $x \notin \mathtt{LTT}$ y $\mathsf{fc}(u, \mathtt{LTT})$. Más aún, $\alpha \notin \mathtt{LXC}$ y $\mathsf{fc}(s, \mathtt{LXC})$ implican $\alpha \notin u$ y $x \notin \mathsf{fv}(s)$ respectivamente. Hay dos posibles casos:
  1. $t = \mathtt{LTC}_1\langle c\rangle$. Luego, $\mathtt{LXC} = \mathtt{LTT}\langle \mathtt{LTC}_1\rangle[x \setminus u]$. Más aún, se tiene $o' = \mathtt{LTT}\langle \mathtt{LTC}_1\langle c\rangle[x \setminus u]\rangle$: *i.e.* $\mathtt{LXC}' = \mathtt{LTT}\langle \mathtt{LTC}_1[x \setminus u]\rangle$ y $c' = c$. Por Lem. 4.6.2 (1), se tiene $\mathfrak{C}(\mathtt{LXC}\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle) = \mathfrak{C}(\mathtt{LTT}\langle \mathtt{LTC}_1\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle\rangle[x \setminus u]) \simeq \mathfrak{C}(\mathtt{LTT}\langle \mathtt{LTC}_1\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle[x \setminus u]\rangle) = \mathfrak{C}(\mathtt{LXC}'\langle c'[\![\alpha \setminus^{\alpha'} s]\!]\rangle)$.
  2. $c = \mathtt{LCT}\langle t\rangle$ con $x \notin \mathtt{LCT}$ y $\mathsf{fc}(u, \mathtt{LCT})$. Luego, $o' = \mathtt{LTC}\langle \mathtt{LCT}\langle t[x \setminus u]\rangle\rangle$: *i.e.* $\mathtt{LXC}' = \mathtt{LTC}$ y $c' = \mathtt{LCT}\langle t[x \setminus u]\rangle$. Por Lem. 4.6.2 (1), $\mathfrak{C}(\mathtt{LXC}\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle) = \mathfrak{C}(\mathtt{LTC}\langle \mathtt{LCT}\langle t\rangle[\![\alpha \setminus^{\alpha'} s]\!]\rangle[x \setminus u]) \simeq \mathfrak{C}(\mathtt{LTC}\langle \mathtt{LCT}\langle t[x \setminus u]\rangle[\![\alpha \setminus^{\alpha'} s]\!]\rangle) = \mathfrak{C}(\mathtt{LXC}'\langle c'[\![\alpha \setminus^{\alpha'} s]\!]\rangle)$.
- $\mathtt{LXC} = [\delta]\,\mathtt{LTC}$. Hay dos posibles reglas a analizar:

- $\simeq_{\mathtt{pp}}$. Se tienen dos posibles instancias de la regla según la forma de $\mathtt{LXC}$:
  1. $\mathtt{LXC} = [\delta]\,\lambda x.\mu\delta'.\boxdot$. Luego, $c = [\gamma]\,\lambda y.\mu\gamma'.c_0$ y $o' = [\gamma]\,\lambda y.\mu\gamma'.[\delta]\,\lambda x.\mu\delta'.c_0$. Más aún, $\alpha \notin \mathtt{LXC}$, $\mathsf{fc}(\alpha', \mathtt{LXC})$ y $\mathsf{fc}(s, \mathtt{LXC})$ implican $\alpha \neq \delta$, $\alpha' \neq \delta'$, $x \notin \mathsf{fv}(s)$ y $\delta' \notin \mathsf{fn}(s)$ respectivamente. Por α-conversión se asumen también $y \notin s$, $\gamma' \neq \alpha'$ y $\gamma' \notin \mathsf{fn}(s)$. Hay tres posibilidades para γ:
     a) $\alpha = \gamma$ y $\gamma \notin c_0$. Luego, $c[\![\alpha \setminus^{\alpha'} s]\!] \to_{\mathtt{N}} [\alpha']\,(\lambda y.\mu\gamma'.c_0) :: s$. Asumir $s = u \cdot s'$ (si $s = u$ el análisis es ligeramente más simple) y considerar nombres frescos γ'', δ''. Luego, $\mathfrak{C}(\mathtt{LXC}\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle) = \mathfrak{C}([\delta]\,\lambda x.\mu\delta'.[\alpha']\,(\mu\gamma''.c_0[\![\gamma' \setminus^{\gamma''} s']\!])[y \setminus u])$. Por otro lado, considerar $\mathtt{LXC}' = \boxdot$ y $c' = o'$. Análogamente, se tiene $\mathfrak{C}(\mathtt{LXC}'\langle c'[\![\alpha \setminus^{\alpha'} s]\!]\rangle) = \mathfrak{C}([\alpha']\,(\mu\gamma''.([\delta]\,\lambda x.\mu\delta'.c_0)[\![\gamma' \setminus^{\gamma''} s']\!])[y \setminus u])$. Luego, por Lem. 4.6.2, se tiene por un lado $\mathfrak{C}(\mathtt{LXC}\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle) \simeq \mathfrak{C}([\delta]\,(\lambda x.\mu\delta'.[\alpha']\,\mu\gamma''.c_0[\![\gamma' \setminus^{\gamma''} s']\!])[y \setminus u])$ y por el otro $\mathfrak{C}(\mathtt{LXC}'\langle c'[\![\alpha \setminus^{\alpha'} s]\!]\rangle) \simeq \mathfrak{C}([\alpha']\,\mu\gamma''.[\delta]\,(\lambda x.\mu\delta'.c_0[\![\gamma' \setminus^{\gamma''} s']\!])[y \setminus u])$, apelando al ítem (1) del lema para posicionar la sustitución explícita y al ítem (2) para el replacement. Finalmente, aplicando dos veces $\simeq_\rho$ y Lem. 4.6.2 (2) para posicionar adecuadamente el renaming explícito introducido por la regla, se tiene

$$\begin{array}{rcl}
\mathfrak{C}(\mathtt{LXC}\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle) & \simeq & \mathfrak{C}([\delta]\,(\lambda x.\mu\delta'.[\alpha']\,\mu\gamma''.c_0[\![\gamma' \setminus^{\gamma''} s']\!])[y \setminus u]) \\
& = & [\delta]\,(\lambda x.\mu\delta'.[\alpha']\,\mu\gamma''.\mathfrak{C}(c_0[\![\gamma' \setminus^{\gamma''} s']\!]))[y \setminus \mathfrak{C}(u)] \\
& \simeq_\rho & [\delta]\,(\lambda x.\mu\delta'.\mathfrak{C}(c_0[\![\gamma' \setminus^{\gamma''} s']\!])[\gamma'' \setminus \alpha'])[y \setminus \mathfrak{C}(u)] \\
& \simeq & ([\delta]\,(\lambda x.\mu\delta'.\mathfrak{C}(c_0[\![\gamma' \setminus^{\gamma''} s']\!]))[y \setminus \mathfrak{C}(u)])[\gamma'' \setminus \alpha'] \\
& \simeq_\rho & [\alpha']\,\mu\gamma''.[\delta]\,(\lambda x.\mu\delta'.\mathfrak{C}(c_0[\![\gamma' \setminus^{\gamma''} s']\!]))[y \setminus \mathfrak{C}(u)] \\
& = & \mathfrak{C}([\alpha']\,\mu\gamma''.[\delta]\,(\lambda x.\mu\delta'.c_0[\![\gamma' \setminus^{\gamma''} s']\!])[y \setminus u]) \\
& \simeq & \mathfrak{C}(\mathtt{LXC}'\langle c'[\![\alpha \setminus^{\alpha'} s]\!]\rangle)
\end{array}$$

     b) $\alpha = \gamma$ y $\gamma \in c_0$. Luego, $\mathfrak{C}(c[\![\alpha \setminus^{\alpha'} s]\!]) = c[\![\alpha \setminus^{\alpha'} \mathfrak{C}(s)]\!]$. Más aún, por Lem. 4.6.2 (2), se tiene $\mathfrak{C}(\mathtt{LXC}\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle) = \mathfrak{C}([\delta]\,\lambda x.\mu\delta'.c[\![\alpha \setminus^{\alpha'} s]\!]) \simeq \mathfrak{C}(([\delta]\,\lambda x.\mu\delta'.c)[\![\alpha \setminus^{\alpha'} s]\!])$. Considerar $\mathtt{LXC}' = \boxdot$ y $c' = o'$. Entonces, aplicando la regla $\simeq_{\mathtt{pp}}$ se tiene

$$\begin{array}{rcl}
\mathfrak{C}(\mathtt{LXC}'\langle c'[\![\alpha \setminus^{\alpha'} s]\!]\rangle) & \simeq & \mathfrak{C}(([\delta]\,\lambda x.\mu\delta'.c)[\![\alpha \setminus^{\alpha'} s]\!]) \\
& = & ([\delta]\,\lambda x.\mu\delta'.[\gamma]\,\lambda y.\mu\gamma'.c_0)[\![\alpha \setminus^{\alpha'} \mathfrak{C}(s)]\!] \\
& \simeq & ([\gamma]\,\lambda y.\mu\gamma'.[\delta]\,\lambda x.\mu\delta'.c_0)[\![\alpha \setminus^{\alpha'} \mathfrak{C}(s)]\!] \\
& = & \mathfrak{C}(\mathtt{LXC}'\langle c'[\![\alpha \setminus^{\alpha'} s]\!]\rangle)
\end{array}$$

     c) $\alpha \neq \gamma$. Por Lem. 4.6.2 (2), $\mathfrak{C}(\mathtt{LXC}\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle) = \mathfrak{C}([\delta]\,\lambda x.\mu\delta'.[\gamma]\,\lambda y.\mu\gamma'.c_0[\![\alpha \setminus^{\alpha'} s]\!]) \simeq \mathfrak{C}([\delta]\,\lambda x.\mu\delta'.[\gamma]\,\lambda y.\mu\gamma'.c_0[\![\alpha \setminus^{\alpha'} s]\!])$. Considerar $\mathtt{LXC}' = [\gamma]\,\lambda y.\mu\gamma'.[\delta]\,\lambda x.\mu\delta'.\boxdot$ y $c' = c_0$. Luego, se tiene $\mathfrak{C}(\mathtt{LXC}'\langle c'[\![\alpha \setminus^{\alpha'} s]\!]\rangle) \simeq [\delta]\,\lambda x.\mu\delta'.[\gamma]\,\lambda y.\mu\gamma'.c_0[\![\alpha \setminus^{\alpha'} \mathfrak{C}(s)]\!] \simeq_{\mathtt{pp}} [\gamma]\,\lambda y.\mu\gamma'.[\delta]\,\lambda x.\mu\delta'.c_0[\![\alpha \setminus^{\alpha'} \mathfrak{C}(s)]\!] = \mathfrak{C}(\mathtt{LXC}'\langle c'[\![\alpha \setminus^{\alpha'} s]\!]\rangle)$.
  2. $\mathtt{LXC} = [\delta]\,\lambda x.\mu\delta'.[\gamma]\,\lambda y.\mu\gamma'.\mathtt{LCC}$. Luego, $o' = [\gamma]\,\lambda y.\mu\gamma'.[\delta]\,\lambda x.\mu\delta'.\mathtt{LCC}\langle c\rangle$: *i.e.* $\mathtt{LXC}' = [\gamma]\,\lambda y.\mu\gamma'.[\delta]\,\lambda x.\mu\delta'.\mathtt{LCC}$ y $c' = c$. Más aún, $\alpha \notin \mathtt{LXC}$ implica $\alpha \neq \delta$ y $\alpha \neq \gamma$; $\mathsf{fc}(\alpha', \mathtt{LXC})$ implica $\alpha' \neq \delta'$ y $\alpha' \neq \gamma'$; y $\mathsf{fc}(s, \mathtt{LXC})$ implica $x, y \notin \mathsf{fv}(s)$ y $\delta', \gamma' \notin \mathsf{fn}(s)$. Finalmente, se concluye por $\simeq_{\mathtt{pp}}$, $\mathfrak{C}(\mathtt{LXC}\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle) = [\delta]\,\lambda x.\mu\delta'.[\gamma]\,\lambda y.\mu\gamma'.\mathfrak{C}(\mathtt{LCC}\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle) \simeq_{\mathtt{pp}} [\gamma]\,\lambda y.\mu\gamma'.[\delta]\,\lambda x.\mu\delta'.\mathfrak{C}(\mathtt{LCC}\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle) = \mathfrak{C}(\mathtt{LXC}'\langle c'[\![\alpha \setminus^{\alpha'} s]\!]\rangle)$.
- $\simeq_\rho$. Luego, $\mathtt{LXC} = [\delta]\,\mu\gamma.\mathtt{LCC}$ y $o' = \mathtt{LCC}\langle c\rangle[\gamma \setminus \delta]$: *i.e.* $\mathtt{LXC}' = \mathtt{LCC}[\gamma \setminus \delta]$ y $c' = c$. Más aún, $\alpha \notin \mathtt{LXC}$, $\mathsf{fc}(\alpha', \mathtt{LXC})$ y $\mathsf{fc}(s, \mathtt{LXC})$ implican $\alpha \neq \delta$, $\alpha' \neq \gamma$ y $\gamma \notin \mathsf{fn}(s)$ respectivamente. Se concluye por $\simeq_\rho$, $\mathfrak{C}(\mathtt{LXC}\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle) = [\delta]\,\mu\gamma.\mathfrak{C}(\mathtt{LCC}\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle) \simeq_\rho \mathfrak{C}(\mathtt{LCC}\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle)[\gamma \setminus \delta] = \mathfrak{C}(\mathtt{LXC}'\langle c'[\![\alpha \setminus^{\alpha'} s]\!]\rangle)$.

- $\mathtt{LXC} = \mathtt{LCC}[\![\gamma \setminus^{\delta} s']\!]$. Luego, la única regla aplicable es $\simeq_{\mathtt{exrepl}}$. Se tiene entonces $\mathtt{LCC}\langle c\rangle = \mathtt{LCC}_0\langle c_0\rangle$ con $\gamma \notin \mathtt{LCC}_0$, $\mathsf{fc}(\delta, \mathtt{LCC}_0)$ y $\mathsf{fc}(s', \mathtt{LCC}_0)$. Más aún, $\alpha \notin \mathtt{LXC}$ y $\mathsf{fc}(s, \mathtt{LXC})$ implican $\alpha \neq \delta$, $\alpha \notin s'$ y $\gamma \notin \mathsf{fn}(s)$. Hay dos posibles casos:

1. $c_0 = \mathtt{LCC}_1\langle c\rangle$. Luego, $\mathtt{LXC} = \mathtt{LCC}_0\langle\mathtt{LCC}_1\rangle[\![\gamma \setminus^{\delta} s']\!]$. Más aún, se tiene $o' = \mathtt{LCC}_0\langle\mathtt{LCC}_1\langle c\rangle[\![\gamma \setminus^{\delta} s']\!]\rangle$: *i.e.* $\mathtt{LXC}' = \mathtt{LCC}_0\langle\mathtt{LCC}_1[\![\gamma \setminus^{\delta} s']\!]\rangle$ y $c' = c$. Por Lem. 4.6.2 (2), se tiene $\mathfrak{C}(\mathtt{LXC}\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle) = \mathfrak{C}(\mathtt{LCC}_0\langle\mathtt{LCC}_1\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle[\![\gamma \setminus^{\delta} s']\!]) \simeq \mathfrak{C}(\mathtt{LCC}_0\langle\mathtt{LCC}_1\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle[\![\gamma \setminus^{\delta} s']\!]) = \mathfrak{C}(\mathtt{LXC}'\langle c'[\![\alpha \setminus^{\alpha'} s]\!]\rangle)$.
2. $c = \mathtt{LCC}_1\langle c_0\rangle$ con $\gamma \notin \mathtt{LCC}_1$ y $\mathsf{fc}(s', \mathtt{LCC}_1)$. Luego, $o' = \mathtt{LCC}\langle\mathtt{LCC}_1\langle c_0[\![\gamma \setminus^{\delta} s']\!]\rangle\rangle$: *i.e.* $\mathtt{LXC}' = \mathtt{LCC}$ y $c' = \mathtt{LCC}_1\langle c_0[\![\gamma \setminus^{\delta} s']\!]\rangle$. Más aún, por Lem. 4.6.2 (2), se tiene $\mathfrak{C}(\mathtt{LXC}\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle) = \mathfrak{C}(\mathtt{LCC}\langle\mathtt{LCC}_1\langle c_0\rangle[\![\alpha \setminus^{\alpha'} s]\!]\rangle[\![\gamma \setminus^{\delta} s']\!]) \simeq \mathfrak{C}(\mathtt{LCC}\langle\mathtt{LCC}_1\langle c_0[\![\gamma \setminus^{\delta} s']\!]\rangle[\![\alpha \setminus^{\alpha'} s]\!]\rangle) = \mathfrak{C}(\mathtt{LXC}'\langle c'[\![\alpha \setminus^{\alpha'} s]\!]\rangle)$.

- $\mathtt{LXC} = \mathtt{LCC}[\gamma \setminus \delta]$. Luego, la única regla aplicable es $\simeq_{\mathtt{exren}}$. Se tiene entonces $\mathtt{LCC}\langle c\rangle = \mathtt{LCC}_0\langle c_0\rangle$ con $\gamma \notin \mathtt{LCC}_0$ y $\mathsf{fc}(\delta, \mathtt{LCC}_0)$. Más aún, $\alpha \notin \mathtt{LXC}$, $\mathsf{fc}(\alpha', \mathtt{LXC})$ y $\mathsf{fc}(s, \mathtt{LXC})$ implican $\alpha \neq \delta$, $\alpha' \neq \gamma$ y $\gamma \notin \mathsf{fn}(s)$ respectivamente. Hay dos posibles casos:

  1. $c_0 = \mathtt{LCC}_1\langle c\rangle$. Luego, $\mathtt{LXC} = \mathtt{LCC}_0\langle\mathtt{LCC}_1\rangle[\gamma \setminus \delta]$. Más aún, se tiene $o' = \mathtt{LCC}_0\langle\mathtt{LCC}_1\langle c\rangle[\gamma \setminus \delta]\rangle$: *i.e.* $\mathtt{LXC}' = \mathtt{LCC}_0\langle\mathtt{LCC}_1[\gamma \setminus \delta]\rangle$ y $c' = c$. Por Lem. 4.6.2 (2), se tiene $\mathfrak{C}(\mathtt{LXC}\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle) = \mathfrak{C}(\mathtt{LCC}_0\langle\mathtt{LCC}_1\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle\rangle[\gamma \setminus \delta]) \simeq \mathfrak{C}(\mathtt{LCC}_0\langle\mathtt{LCC}_1\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle[\gamma \setminus \delta]\rangle) = \mathfrak{C}(\mathtt{LXC}'\langle c'[\![\alpha \setminus^{\alpha'} s]\!]\rangle)$.
  2. $c = \mathtt{LCC}_1\langle c_0\rangle$ con $\gamma \notin \mathtt{LCC}_1$ y $\mathsf{fc}(s', \mathtt{LCC}_1)$. Luego, $o' = \mathtt{LCC}\langle\mathtt{LCC}_1\langle c_0[\gamma \setminus \delta]\rangle\rangle$: *i.e.* $\mathtt{LXC}' = \mathtt{LCC}$ y $c' = \mathtt{LCC}_1\langle c_0[\gamma \setminus \delta]\rangle$. Más aún, por Lem. 4.6.2 (2), se tiene $\mathfrak{C}(\mathtt{LXC}\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle) = \mathfrak{C}(\mathtt{LCC}\langle\mathtt{LCC}_1\langle c_0\rangle[\![\alpha \setminus^{\alpha'} s]\!]\rangle[\gamma \setminus \delta]) \simeq \mathfrak{C}(\mathtt{LCC}\langle\mathtt{LCC}_1\langle c_0[\gamma \setminus \delta]\rangle[\![\alpha \setminus^{\alpha'} s]\!]\rangle) = \mathfrak{C}(\mathtt{LXC}'\langle c'[\![\alpha \setminus^{\alpha'} s]\!]\rangle)$.

□

Lema C.0.52. *Sean $o, o' \in \mathbb{O}_{\Lambda M}$ tal que $o \simeq o'$.*

1. *Si $o, o' \in \mathbb{T}_{\Lambda M}$, entonces $\mathfrak{C}(o :: s) \simeq \mathfrak{C}(o' :: s)$.*
2. *Si $o, o' \in \mathbb{C}_{\Lambda M}$, entonces $\mathfrak{C}(o[\![\alpha \setminus^{\alpha'} s]\!]) \simeq \mathfrak{C}(o'[\![\alpha \setminus^{\alpha'} s]\!])$.*

Demostración. Se prueban ambos items en simultaneo por inducción en $o \simeq o'$. Por definición esto implica verificar cuatro condiciones: reflexividad, transitividad, simetría y congruencia (*i.e.* clausura por contextos):

Reflexividad Luego, $o' = o$ por lo que $\mathfrak{C}(o :: s) = \mathfrak{C}(o' :: s)$ y $\mathfrak{C}(o[\![\alpha \setminus^{\alpha'} s]\!]) = \mathfrak{C}(o'[\![\alpha \setminus^{\alpha'} s]\!])$. Ambos resultados son inmediatos por reflexividad de $\simeq$.

Transitividad Luego, existe $p \in \mathbb{O}_{\Lambda M}$ tal que $o \simeq p$ y $p \simeq o'$.

1. Si $o, o' \in \mathbb{T}_{\Lambda M}$, entonces $p \in \mathbb{T}_{\Lambda M}$. Por *h.i.* se tienen $\mathfrak{C}(o :: s) \simeq \mathfrak{C}(p :: s)$ y $\mathfrak{C}(p :: s) \simeq \mathfrak{C}(o' :: s)$. Se concluye por transitividad de $\simeq$.
2. Si $o, o' \in \mathbb{C}_{\Lambda M}$, entonces $p \in \mathbb{C}_{\Lambda M}$. Por *h.i.* se tienen $\mathfrak{C}(o[\![\alpha \setminus^{\alpha'} s]\!]) \simeq \mathfrak{C}(p[\![\alpha \setminus^{\alpha'} s]\!])$ y $\mathfrak{C}(p[\![\alpha \setminus^{\alpha'} s]\!]) \simeq \mathfrak{C}(o'[\![\alpha \setminus^{\alpha'} s]\!])$. Se concluye por transitividad de $\simeq$.

Simetría Luego, $o' \simeq o$.

1. Si $o, o' \in \mathbb{T}_{\Lambda M}$, por *h.i.* se tiene $\mathfrak{C}(o' :: s) \simeq \mathfrak{C}(o :: s)$. Se concluye por simetría de $\simeq$.
2. Si $o, o' \in \mathbb{C}_{\Lambda M}$, por *h.i.* se tiene $\mathfrak{C}(o'[\![\alpha \setminus^{\alpha'} s]\!]) \simeq \mathfrak{C}(o[\![\alpha \setminus^{\alpha'} s]\!])$. Se concluye por simetría de $\simeq$.

Congruencia La congruencia se demuestra por inducción en el contexto de clausura $\mathtt{O}$ tal que $o = \mathtt{O}\langle p\rangle$ y $o' = \mathtt{O}\langle p'\rangle$ con $p \simeq_* p'$, donde $\simeq_*$ es una regla de la Fig. 4.13.

- $\mathtt{O} = \Box$. Luego, $o = p \simeq_* p' = o'$. Más aún, es el caso de $o, o' \in \mathbb{T}_{\Lambda M}$. Se tienen solo dos reglas aplicables a términos:

- $\simeq_{\mathtt{exsubs}}$. Luego, $o = \mathtt{LTT}\langle t\rangle[x \setminus v]$ y $o' = \mathtt{LTT}\langle t[x \setminus v]\rangle$ con $x \notin \mathtt{LTT}$ y $\mathsf{fc}(v, \mathtt{LTT})$. Por Lem. 4.6.2 (1), se tiene $\mathfrak{C}(\mathtt{LTT}\langle t\rangle[x \setminus v] :: s) \simeq \mathfrak{C}((\mathtt{LTT}\langle t\rangle :: s)[x \setminus v]) \simeq \mathfrak{C}(\mathtt{LTT}\langle t[x \setminus v]\rangle :: s)$. Notar que $\Box :: s$ constituye un contexto lineal.
- $\simeq_\theta$. Luego, $o = \mu\gamma.[\gamma]\, t$ y $o' = t$ con $\gamma \notin t$. Más aún, sea δ un nombre fresco, entonces $\mathfrak{C}((\mu\gamma.[\gamma]\, t) :: s) = \mu\delta.\mathfrak{C}(([\gamma]\, t)[\![\gamma \setminus^{\delta} s]\!]) = \mu\delta.[\delta]\, \mathfrak{C}(t :: s) \simeq_\theta \mathfrak{C}(t :: s)$.

- $\mathtt{O} = \boxdot$. Luego, $o = p \simeq_* p' = o'$. Más aún, es el caso de $o, o' \in \mathbb{C}_{\Lambda M}$. Por Lem. C.0.51 con $\mathtt{LXC} = \boxdot$ se tiene $o' = \mathtt{LCC}\langle c\rangle$ tal que $\alpha \notin \mathtt{LCC}$, $\mathsf{fc}(\alpha', \mathtt{LCC})$, $\mathsf{fc}(s, \mathtt{LCC})$ y $\mathfrak{C}(o[\![\alpha \setminus^{\alpha'} s]\!]) \simeq \mathfrak{C}(\mathtt{LCC}\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle)$. Más aún, por Lem. 4.6.2 (2), $\mathfrak{C}(\mathtt{LCC}\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle) \simeq \mathfrak{C}(\mathtt{LCC}\langle c\rangle[\![\alpha \setminus^{\alpha'} s]\!]) = \mathfrak{C}(o'[\![\alpha \setminus^{\alpha'} s]\!])$.
- $\mathtt{O} = \mathtt{T}\, v$. Luego, $o = \mathtt{T}\langle p\rangle\, v$ y $o' = \mathtt{T}\langle p'\rangle\, v$. Sean $t = \mathtt{T}\langle p\rangle$ y $t' = \mathtt{T}\langle p'\rangle$. Se tiene entonces $t \simeq t'$. Más aún, $o \simeq o'$ implica $\mathfrak{C}(o) = o$ y $\mathfrak{C}(o') = o'$. Luego, $\mathfrak{C}(o :: s) = (t\, v) :: \mathfrak{C}(s) \simeq (t'\, v) :: \mathfrak{C}(s) = \mathfrak{C}(o' :: s)$, donde $\mathfrak{C}(s) = \mathfrak{C}(u_0) \cdot \ldots \cdot \mathfrak{C}(u_n)$ para $s = u_0 \cdot \ldots \cdot u_n$.
- $\mathtt{O} = t\, \mathtt{T}$. Luego, $o = t\, v$ y $o' = t\, v'$ con $v \simeq v'$. Más aún, $o \simeq o'$ implica $\mathfrak{C}(o) = o$ y $\mathfrak{C}(o') = o'$. Luego, $\mathfrak{C}(o :: s) = (t\, v) :: \mathfrak{C}(s) \simeq (t\, v') :: \mathfrak{C}(s) = \mathfrak{C}(o' :: s)$.
- $\mathtt{O} = \lambda x.\mathtt{T}$. Luego, $o = \lambda x.t$ y $o' = \lambda x.t'$ con $t \simeq t'$. Más aún, $t \simeq t'$ implica $\mathfrak{C}(t) = t$ y $\mathfrak{C}(t') = t'$. Se procede por inducción en s:
  - $s = u$. Luego, $\mathfrak{C}(o\, u) = t[x \setminus \mathfrak{C}(u)] \simeq t'[x \setminus \mathfrak{C}(u)] = \mathfrak{C}(o'\, u)$.
  - $s = u \cdot s'$. Luego, $\mathfrak{C}(o :: s) = \mathfrak{C}(t[x \setminus u] :: s')$. Por Lem. 4.6.2 (1), se tiene $\mathfrak{C}(o :: s) \simeq \mathfrak{C}(t :: s')[x \setminus \mathfrak{C}(u)]$. Análogamente, $\mathfrak{C}(o' :: s) \simeq \mathfrak{C}(t' :: s')[x \setminus \mathfrak{C}(u)]$. Por *h.i.* se tiene $\mathfrak{C}(t :: s) \simeq \mathfrak{C}(t' :: s)$, por lo que se concluye.
- $\mathtt{O} = \mu\gamma.\mathtt{C}$. Luego, $o = \mu\gamma.c$ y $o' = \mu\gamma.c'$ con $c \simeq c'$. Más aún, $\mathfrak{C}(o\, s) = \mu\delta.\mathfrak{C}(c[\![\gamma \setminus^{\delta} s]\!])$ y $\mathfrak{C}(o'\, s) = \mu\delta.\mathfrak{C}(c'[\![\gamma \setminus^{\delta} s]\!])$. Por *h.i.* se tiene $\mathfrak{C}(c[\![\gamma \setminus^{\delta} s]\!]) \simeq \mathfrak{C}(c'[\![\gamma \setminus^{\delta} s]\!])$, por lo que se concluye.
- $\mathtt{O} = \mathtt{T}[x \setminus v]$. Luego, $o = t[x \setminus v]$ y $o' = t'[x \setminus v]$ con $t \simeq t'$. Por Lem. 4.6.2 (1), se tiene $\mathfrak{C}(o) = \mathfrak{C}(t[x \setminus v] :: s) \simeq \mathfrak{C}(t :: s)[x \setminus v]$. Análogamente, $\mathfrak{C}(o') \simeq \mathfrak{C}(t' :: s)[x \setminus v]$. Por *h.i.* se tiene $\mathfrak{C}(t :: s) \simeq \mathfrak{C}(t' :: s)$, por lo que se concluye.
- $\mathtt{O} = t[x \setminus \mathtt{T}]$. Luego, $o = t[x \setminus v]$ y $o' = t[x \setminus v']$ con $v \simeq v'$. Más aún, $v \simeq v'$ implica $\mathfrak{C}(v) = v$ y $\mathfrak{C}(v') = v'$. Se concluye por Lem. 4.6.2 (1), teniendo $\mathfrak{C}(o :: s) = \mathfrak{C}(t[x \setminus v] :: s) \simeq \mathfrak{C}(t :: s)[x \setminus v] \simeq \mathfrak{C}(t :: s)[x \setminus v'] \simeq \mathfrak{C}(t[x \setminus v'] :: s) = \mathfrak{C}(o' :: s)$.
- $\mathtt{O} = [\delta]\, \mathtt{T}$. Luego, $o = [\delta]\, \mathtt{T}\langle p\rangle$ y $o' = [\delta]\, \mathtt{T}\langle p'\rangle$. Sean $t = \mathtt{T}\langle p\rangle$ y $t' = \mathtt{T}\langle p'\rangle$. Se tiene entonces $t \simeq t'$. Se tienen tres casos posibles:
  1. $\alpha = \delta$ y $\alpha \notin t$. Luego, se tiene también $\alpha \notin t'$. Más aún, $\mathfrak{C}(o[\![\alpha \setminus^{\alpha'} s]\!]) = [\alpha']\, \mathfrak{C}(t :: s)$ y $\mathfrak{C}(o'[\![\alpha \setminus^{\alpha'} s]\!]) = [\alpha']\, \mathfrak{C}(t' :: s)$. Por *h.i.* se tiene $\mathfrak{C}(t :: s) \simeq \mathfrak{C}(t' :: s)$, por lo que se concluye.
  2. $o[\![\alpha \setminus^{\alpha'} s]\!] \to_{\mathtt{NPW}} [\delta]\, u$ (*i.e.* $\alpha \neq \delta$ y $\alpha \in t$). Notar que, dado que $t = \mathtt{T}\langle p\rangle$ y $\mathsf{fn}_\alpha(t) = 1$ por hipótesis de las reglas `N`, `P` y `W` hay solo dos casos posibles:
     a) $u = \mathtt{T}'\langle p\rangle$. Luego, se analiza individualmente la regla de redución aplicada:
     - `N`. Luego, $\mathtt{T} = \mathtt{LTC}\langle[\alpha]\, \mathtt{T}_1\rangle$ y $\mathtt{T}' = \mathtt{LTC}\langle[\alpha']\, \mathtt{T}_1 :: s\rangle$ con $\alpha \notin \mathtt{LTC}$ y $\alpha \notin \mathtt{T}_1$. Más aún, $\mathfrak{C}(o[\![\alpha \setminus^{\alpha'} s]\!]) = [\delta]\, \mathfrak{C}(\mathtt{T}'\langle p\rangle) = [\delta]\, \mathtt{LTC}\langle[\alpha']\, \mathfrak{C}(\mathtt{T}_1\langle p\rangle :: s)\rangle$. Por otro lado, se tiene $o'[\![\alpha \setminus^{\alpha'} s]\!] \to_{\mathtt{N}} [\delta]\, \mathtt{T}'\langle p'\rangle$, por lo que $\mathfrak{C}(o'[\![\alpha \setminus^{\alpha'} s]\!]) = [\delta]\, \mathtt{LTC}\langle[\alpha']\, \mathfrak{C}(\mathtt{T}_1\langle p'\rangle :: s)\rangle$. Por *h.i.* se tiene $\mathfrak{C}(\mathtt{T}_1\langle p\rangle :: s) \simeq \mathfrak{C}(\mathtt{T}_1\langle p'\rangle :: s)$, por lo que se concluye.
     - `P`. Luego, $\mathtt{T} = \mathtt{LTC}\langle\mathtt{C}[\![\gamma \setminus^{\alpha} s']\!]\rangle$ y $\mathtt{T}' = \mathtt{LTC}\langle\mathtt{C}[\![\gamma \setminus^{\alpha'} s' \cdot s]\!]\rangle$ con $\alpha \notin \mathtt{LTC}$, $\alpha \notin \mathtt{C}$ y $\alpha \notin s'$. Más aún, $\mathfrak{C}(o[\![\alpha \setminus^{\alpha'} s]\!]) = [\delta]\, \mathfrak{C}(\mathtt{T}'\langle p\rangle) = [\delta]\, \mathtt{LTC}\langle\mathfrak{C}(\mathtt{C}\langle p\rangle[\![\gamma \setminus^{\alpha'} s' \cdot s]\!])\rangle$. Por otro lado, se tiene $o'[\![\alpha \setminus^{\alpha'} s]\!] \to_{\mathtt{P}} [\delta]\, \mathtt{T}'\langle p'\rangle$, por lo que $\mathfrak{C}(o'[\![\alpha \setminus^{\alpha'} s]\!]) = [\delta]\, \mathtt{LTC}\langle\mathfrak{C}(\mathtt{C}\langle p'\rangle[\![\gamma \setminus^{\alpha'} s' \cdot s]\!])\rangle$. Por *h.i.* $\mathfrak{C}(\mathtt{C}\langle p\rangle[\![\gamma \setminus^{\alpha'} s' \cdot s]\!]) \simeq \mathfrak{C}(\mathtt{C}\langle p'\rangle[\![\gamma \setminus^{\alpha'} s' \cdot s]\!])$, por lo que se concluye.

- $\mathtt{W}$. Luego, $\mathtt{T} = \mathtt{LTC}\langle \mathtt{C}[\gamma \setminus \alpha]\rangle$ y $\mathtt{T}' = \mathtt{LTC}\langle \mathtt{C}[\![\gamma \setminus^{\alpha} s]\!][\alpha \setminus \alpha']\rangle$ con $\alpha \notin \mathtt{LTC}$ y $\alpha \notin \mathtt{C}$. Más aún, $\mathfrak{C}(o[\![\alpha \setminus^{\alpha'} s]\!]) = [\delta]\,\mathfrak{C}(\mathtt{T}'\langle p\rangle) = [\delta]\,\mathtt{LTC}\langle \mathfrak{C}(\mathtt{C}\langle p\rangle[\![\gamma \setminus^{\alpha} s]\!])[\alpha \setminus \alpha']\rangle$. Por otro lado, $o'[\![\alpha \setminus^{\alpha'} s]\!] \to_{\mathtt{W}} [\delta]\,\mathtt{T}'\langle p'\rangle$, por lo que $\mathfrak{C}(o'[\![\alpha \setminus^{\alpha'} s]\!]) = [\delta]\,\mathtt{LTC}\langle \mathfrak{C}(\mathtt{C}\langle p'\rangle[\![\gamma \setminus^{\alpha} s]\!])[\alpha \setminus \alpha']\rangle$. Por *h.i.* se tiene $\mathfrak{C}(\mathtt{C}\langle p\rangle[\![\gamma \setminus^{\alpha} s]\!]) \simeq \mathfrak{C}(\mathtt{C}\langle p'\rangle[\![\gamma \setminus^{\alpha} s]\!])$, por lo que se concluye.

b) $u = \mathtt{T}\langle q\rangle$. Luego, se tiene un contexto lineal $\mathtt{LTC}$ tal que $\mathtt{LTC} = \mathtt{LTX}\langle \mathtt{LXC}\rangle$, $\mathtt{T} = \mathtt{LTX}$ y $p = \mathtt{LXC}\langle c\rangle$ con $\alpha \neq \delta$, $\alpha \notin \mathtt{LTC}$ y $\alpha \in c$. Por α-conversión se asumen también $\mathsf{fc}(\alpha', \mathtt{LTC})$ y $\mathsf{fc}(s, \mathtt{LTC})$, de modo que $c[\![\alpha \setminus^{\alpha'} s]\!] \to_{\mathtt{NPW}} q$. Luego, por Lem. C.0.51 con $p \simeq_* p'$ se tiene $p' = \mathtt{LXC}'\langle c'\rangle$ tal que $\alpha \notin \mathtt{LXC}'$, $\mathsf{fc}(\alpha', \mathtt{LXC}')$, $\mathsf{fc}(s, \mathtt{LXC}')$ y $\mathfrak{C}(\mathtt{LXC}\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle) \simeq \mathfrak{C}(\mathtt{LXC}'\langle c'[\![\alpha \setminus^{\alpha'} s]\!]\rangle)$. Entonces, $o = [\delta]\,\mathtt{T}\langle p\rangle = [\delta]\,\mathtt{LTX}\langle \mathtt{LXC}\langle c\rangle\rangle$ y $o' = [\delta]\,\mathtt{T}\langle p'\rangle = [\delta]\,\mathtt{LTX}\langle \mathtt{LXC}'\langle c'\rangle\rangle$. Finalmente, por Lem. 4.6.2 (2), se obtiene

$$\begin{array}{rcl}
\mathfrak{C}(o[\![\alpha \setminus^{\alpha'} s]\!]) & \simeq & \mathfrak{C}([\delta]\,\mathtt{LTX}\langle \mathtt{LXC}\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle\rangle) \\
& = & [\delta]\,\mathtt{LTX}\langle \mathfrak{C}(\mathtt{LXC}\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle)\rangle \\
& \simeq & [\delta]\,\mathtt{LTX}\langle \mathfrak{C}(\mathtt{LXC}'\langle c'[\![\alpha \setminus^{\alpha'} s]\!]\rangle)\rangle \\
& = & \mathfrak{C}([\delta]\,\mathtt{LTX}\langle \mathtt{LXC}'\langle c'[\![\alpha \setminus^{\alpha'} s]\!]\rangle\rangle) \\
& \simeq & \mathfrak{C}(o'[\![\alpha \setminus^{\alpha'} s]\!])
\end{array}$$

3. De lo contrario, $\mathfrak{C}(o[\![\alpha \setminus^{\alpha'} s]\!]) = o[\![\alpha \setminus^{\alpha'} \mathfrak{C}(s)]\!] \simeq o'[\![\alpha \setminus^{\alpha'} \mathfrak{C}(s)]\!] = \mathfrak{C}(o'[\![\alpha \setminus^{\alpha'} s]\!])$, dado que $o \simeq o'$ implica $\mathfrak{C}(o) = o$ y $\mathfrak{C}(o') = o'$.

- $\mathtt{O} = \mathtt{C}[\![\gamma \setminus^{\delta} s']\!]$. Luego, $o = c[\![\gamma \setminus^{\delta} s']\!]$ y $o' = c'[\![\gamma \setminus^{\delta} s']\!]$ con $c \simeq c'$. Más aún, $o \simeq o'$ implica $\mathfrak{C}(o) = o$ y $\mathfrak{C}(o') = o'$. Luego, se tienen tres casos posibles:
  1. $\alpha = \delta$, $\alpha \notin c$ y $\alpha \notin s'$. Luego, se tiene también $\alpha \notin c'$. Más aún, $\mathfrak{C}(o[\![\alpha \setminus^{\alpha'} s]\!]) = c[\![\gamma \setminus^{\alpha'} s' \cdot \mathfrak{C}(s)]\!] \simeq c'[\![\gamma \setminus^{\alpha'} s' \cdot \mathfrak{C}(s)]\!] = \mathfrak{C}(o'[\![\alpha \setminus^{\alpha'} s]\!])$, por lo que se concluye.
  2. $\alpha \neq \delta$ y $\alpha \notin s'$. Por α-conversión se asumen también $\alpha' \neq \gamma$ y $\gamma \notin \mathsf{fn}(s)$. Luego, por Lem. 4.6.2 (2) se tienen $\mathfrak{C}(o[\![\alpha \setminus^{\alpha'} s]\!]) \simeq \mathfrak{C}(c[\![\alpha \setminus^{\alpha'} s]\!])[\![\gamma \setminus^{\delta} s']\!]$ y $\mathfrak{C}(o'[\![\alpha \setminus^{\alpha'} s]\!]) \simeq \mathfrak{C}(c'[\![\alpha \setminus^{\alpha'} s]\!])[\![\gamma \setminus^{\delta} s']\!]$. Finalmente, por *h.i.* se tiene $\mathfrak{C}(c[\![\alpha \setminus^{\alpha'} s]\!]) \simeq \mathfrak{C}(c'[\![\alpha \setminus^{\alpha'} s]\!])$, por lo que se concluye.
  3. De lo contrario, $\mathfrak{C}(o[\![\alpha \setminus^{\alpha'} s]\!]) = o[\![\alpha \setminus^{\alpha'} \mathfrak{C}(s)]\!] \simeq o'[\![\alpha \setminus^{\alpha'} \mathfrak{C}(s)]\!] = \mathfrak{C}(o'[\![\alpha \setminus^{\alpha'} s]\!])$.
- $\mathtt{O} = c[\![\gamma \setminus^{\delta} \mathtt{S}]\!]$. Luego, $o = c[\![\gamma \setminus^{\delta} s']\!]$ y $o' = c[\![\gamma \setminus^{\delta} s'']\!]$ con $s' \simeq s''$. Más aún, $o \simeq o'$ implica $\mathfrak{C}(o) = o$ y $\mathfrak{C}(o') = o'$. Luego, se tienen tres casos posibles:
  1. $\alpha = \delta$, $\alpha \notin c$ y $\alpha \notin s'$. Luego, se tiene también $\alpha \notin s''$. Más aún, $\mathfrak{C}(o[\![\alpha \setminus^{\alpha'} s]\!]) = c[\![\gamma \setminus^{\alpha'} s' \cdot \mathfrak{C}(s)]\!] \simeq c[\![\gamma \setminus^{\alpha'} s'' \cdot \mathfrak{C}(s)]\!] = \mathfrak{C}(o'[\![\alpha \setminus^{\alpha'} s]\!])$, por lo que se concluye.
  2. $\alpha \neq \delta$ y $\alpha \notin s'$. Luego, se tiene también $\alpha \notin s''$. Por α-conversión se asumen también $\alpha' \neq \gamma$ y $\gamma \notin \mathsf{fn}(s)$. Se concluye por Lem. 4.6.2 (2), teniendo $\mathfrak{C}(o[\![\alpha \setminus^{\alpha'} s]\!]) \simeq \mathfrak{C}(c[\![\alpha \setminus^{\alpha'} s]\!])[\![\gamma \setminus^{\delta} s']\!] \simeq \mathfrak{C}(c[\![\alpha \setminus^{\alpha'} s]\!])[\![\gamma \setminus^{\delta} s'']\!] \simeq \mathfrak{C}(o'[\![\alpha \setminus^{\alpha'} s]\!])$.
  3. De lo contrario, $\mathfrak{C}(o[\![\alpha \setminus^{\alpha'} s]\!]) = o[\![\alpha \setminus^{\alpha'} \mathfrak{C}(s)]\!] \simeq o'[\![\alpha \setminus^{\alpha'} \mathfrak{C}(s)]\!] = \mathfrak{C}(o'[\![\alpha \setminus^{\alpha'} s]\!])$.
- $\mathtt{O} = \mathtt{C}[\gamma \setminus \delta]$. Luego, $o = c[\gamma \setminus \delta]$ y $o' = c'[\gamma \setminus \delta]$ con $c \simeq c'$. Más aún, $o \simeq o'$ implica $\mathfrak{C}(o) = o$ y $\mathfrak{C}(o') = o'$. Luego, se tienen tres casos posibles:
  1. $\alpha = \delta$ y $\alpha \notin c$. Luego, se tiene también $\alpha \notin c'$. Más aún, $\mathfrak{C}(o[\![\alpha \setminus^{\alpha'} s]\!]) = \mathfrak{C}(c[\![\gamma \setminus^{\alpha} s]\!])[\alpha' \setminus \alpha]$ y $\mathfrak{C}(o'[\![\alpha \setminus^{\alpha'} s]\!]) = \mathfrak{C}(c'[\![\gamma \setminus^{\alpha} s]\!])[\alpha' \setminus \alpha]$. Finalmente, por *h.i.* se tiene $\mathfrak{C}(c[\![\gamma \setminus^{\alpha'} s]\!]) \simeq \mathfrak{C}(c'[\![\gamma \setminus^{\alpha'} s]\!])$, por lo que se concluye.
  2. $\alpha \neq \delta$. Luego, por Lem. 4.6.2 (2) se tienen $\mathfrak{C}(o[\![\alpha \setminus^{\alpha'} s]\!]) \simeq \mathfrak{C}(c[\![\alpha \setminus^{\alpha'} s]\!])[\gamma \setminus \delta]$ y $\mathfrak{C}(o'[\![\alpha \setminus^{\alpha'} s]\!]) \simeq \mathfrak{C}(c'[\![\alpha \setminus^{\alpha'} s]\!])[\gamma \setminus \delta]$. Finalmente, por *h.i.* $\mathfrak{C}(c[\![\alpha \setminus^{\alpha'} s]\!]) \simeq \mathfrak{C}(c'[\![\alpha \setminus^{\alpha'} s]\!])$, por lo que se concluye.
  3. De lo contrario, $\mathfrak{C}(o[\![\alpha \setminus^{\alpha'} s]\!]) = o[\![\alpha \setminus^{\alpha'} \mathfrak{C}(s)]\!] \simeq o'[\![\alpha \setminus^{\alpha'} \mathfrak{C}(s)]\!] = \mathfrak{C}(o'[\![\alpha \setminus^{\alpha'} s]\!])$.
- $\mathtt{O} = \mathtt{T} \cdot s$. El resultado es inmediato pues no se satisfacen las hipótesis de ninguno de los dos casos.

- $\mathtt{O} = t \cdot \mathtt{S}$. El resultado es inmediato pues no se satisfacen las hipótesis de ninguno de los dos casos.

□

Lema 4.8.1. *Sea $o \in \mathbb{O}_{\Lambda M}$. Si $o \simeq o'$, entonces para todo contexto $\mathtt{O}$ del sort adecuado se tiene $\mathfrak{C}(\mathtt{O}\langle o\rangle) \simeq \mathfrak{C}(\mathtt{O}\langle o'\rangle)$.*

Demostración. Por inducción en $\mathtt{O}$.

- $\mathtt{O} = \Box$. El resultado es inmediato por hipótesis pues, por definición, $o \simeq o'$ implica $\mathfrak{C}(o) = o$ y $\mathfrak{C}(o') = o'$.
- $\mathtt{O} = \boxdot$. El resultado es inmediato por hipótesis pues, por definición, $o \simeq o'$ implica $\mathfrak{C}(o) = o$ y $\mathfrak{C}(o') = o'$.
- $\mathtt{O} = \mathtt{T}\, u$. Luego, $\mathfrak{C}(\mathtt{O}\langle o\rangle) = \mathfrak{C}(\mathfrak{C}(\mathtt{T}\langle o\rangle)\, u)$ y $\mathfrak{C}(\mathtt{O}\langle o'\rangle) = \mathfrak{C}(\mathfrak{C}(\mathtt{T}\langle o'\rangle)\, u)$. Por *h.i.* se tiene $\mathfrak{C}(\mathtt{T}\langle o\rangle) \simeq \mathfrak{C}(\mathtt{T}\langle o'\rangle)$. Luego, se concluye por Lem. C.0.52 (1).
- $\mathtt{O} = t\, \mathtt{T}$. Luego, se tienen tres casos posible:
  1. $\mathfrak{C}(t) = \mathtt{L}\langle \lambda x.u\rangle$. Luego, $\mathfrak{C}(\mathtt{O}\langle o\rangle) = \mathfrak{C}(\mathtt{L}\langle \lambda x.u\rangle\, \mathtt{T}\langle o\rangle) = \mathtt{L}\langle u[x \setminus \mathfrak{C}(\mathtt{T}\langle o\rangle)]\rangle$. Del mismo modo, $\mathfrak{C}(\mathtt{O}\langle o'\rangle) = \mathtt{L}\langle u[x \setminus \mathfrak{C}(\mathtt{T}\langle o'\rangle)]\rangle$. Se concluye pues por *h.i.* se tiene $\mathfrak{C}(\mathtt{T}\langle o\rangle) \simeq \mathfrak{C}(\mathtt{T}\langle o'\rangle)$.
  2. $\mathfrak{C}(t) = \mathtt{L}\langle \mu\alpha.c\rangle$. Luego, $\mathfrak{C}(\mathtt{O}\langle o\rangle) = \mathfrak{C}(\mathtt{L}\langle \mu\alpha.c\rangle\, \mathtt{T}\langle o\rangle) = \mathtt{L}\langle \mu\alpha'.\mathfrak{C}(c[\![\alpha \setminus^{\alpha'} \mathtt{T}\langle o\rangle]\!])\rangle$. Del mismo modo, $\mathfrak{C}(\mathtt{O}\langle o'\rangle) = \mathtt{L}\langle \mu\alpha'.\mathfrak{C}(c[\![\alpha \setminus^{\alpha'} \mathtt{T}\langle o'\rangle]\!])\rangle$. Por Lem. C.0.28, $\mathsf{sz}(\mathtt{O}\langle o\rangle) = \mathsf{sz}(t\, \mathtt{T}\langle o\rangle) \geq \mathsf{sz}(\mathtt{L}\langle \mu\alpha.c\rangle\, \mathtt{T}\langle o\rangle) \geq \mathsf{sz}(\mathtt{L}\langle \mu\alpha'.c[\![\alpha \setminus^{\alpha'} \mathtt{T}\langle o\rangle]\!]\rangle)$, por lo que $\mathsf{sz}(\mathtt{O}) > \mathsf{sz}(c[\![\alpha \setminus^{\alpha'} \mathtt{T}]\!])$. Finalmente, se concluye pues por *h.i.* se tiene $\mathfrak{C}(c[\![\alpha \setminus^{\alpha'} \mathtt{T}\langle o\rangle]\!]) \simeq \mathfrak{C}(c[\![\alpha \setminus^{\alpha'} \mathtt{T}\langle o'\rangle]\!])$.
  3. De lo contrario, $\mathfrak{C}(\mathtt{O}\langle o\rangle) = \mathfrak{C}(t)\, \mathfrak{C}(\mathtt{T}\langle o\rangle)$ y $\mathfrak{C}(\mathtt{O}\langle o'\rangle) = \mathfrak{C}(t)\, \mathfrak{C}(\mathtt{T}\langle o'\rangle)$. Se concluye pues por *h.i.* se tiene $\mathfrak{C}(\mathtt{T}\langle o\rangle) \simeq \mathfrak{C}(\mathtt{T}\langle o'\rangle)$.
- $\mathtt{O} = \lambda x.\mathtt{T}$. Luego, $\mathfrak{C}(\mathtt{O}\langle o\rangle) = \lambda x.\mathfrak{C}(\mathtt{T}\langle o\rangle)$ y $\mathfrak{C}(\mathtt{O}\langle o'\rangle) = \lambda x.\mathfrak{C}(\mathtt{T}\langle o'\rangle)$. Se concluye pues por *h.i.* se tiene $\mathfrak{C}(\mathtt{T}\langle o\rangle) \simeq \mathfrak{C}(\mathtt{T}\langle o'\rangle)$.
- $\mathtt{O} = \mu\alpha.\mathtt{C}$. Luego, $\mathfrak{C}(\mathtt{O}\langle o\rangle) = \mu\alpha.\mathfrak{C}(\mathtt{C}\langle o\rangle)$ y $\mathfrak{C}(\mathtt{O}\langle o'\rangle) = \mu\alpha.\mathfrak{C}(\mathtt{C}\langle o'\rangle)$. Se concluye pues por *h.i.* se tiene $\mathfrak{C}(\mathtt{C}\langle o\rangle) \simeq \mathfrak{C}(\mathtt{C}\langle o'\rangle)$.
- $\mathtt{O} = \mathtt{T}[x \setminus u]$. Luego, $\mathfrak{C}(\mathtt{O}\langle o\rangle) = \mathfrak{C}(\mathtt{T}\langle o\rangle)[x \setminus u]$ y $\mathfrak{C}(\mathtt{O}\langle o'\rangle) = \mathfrak{C}(\mathtt{T}\langle o'\rangle)[x \setminus u]$. Se concluye pues por *h.i.* se tiene $\mathfrak{C}(\mathtt{T}\langle o\rangle) \simeq \mathfrak{C}(\mathtt{T}\langle o'\rangle)$.
- $\mathtt{O} = t[x \setminus \mathtt{T}]$. Luego, $\mathfrak{C}(\mathtt{O}\langle o\rangle) = t[x \setminus \mathfrak{C}(\mathtt{T}\langle o\rangle)]$ y $\mathfrak{C}(\mathtt{O}\langle o'\rangle) = t[x \setminus \mathfrak{C}(\mathtt{T}\langle o'\rangle)]$. Se concluye pues por *h.i.* se tiene $\mathfrak{C}(\mathtt{T}\langle o\rangle) \simeq \mathfrak{C}(\mathtt{T}\langle o'\rangle)$.
- $\mathtt{O} = [\alpha]\, \mathtt{T}$. Luego, $\mathfrak{C}(\mathtt{O}\langle o\rangle) = [\alpha]\, \mathfrak{C}(\mathtt{T}\langle o\rangle)$ y $\mathfrak{C}(\mathtt{O}\langle o'\rangle) = [\alpha]\, \mathfrak{C}(\mathtt{T}\langle o'\rangle)$. Se concluye pues por *h.i.* se tiene $\mathfrak{C}(\mathtt{T}\langle o\rangle) \simeq \mathfrak{C}(\mathtt{T}\langle o'\rangle)$.
- $\mathtt{O} = \mathtt{C}[\![\alpha \setminus^{\alpha'} s]\!]$. Luego, se tienen $\mathfrak{C}(\mathtt{O}\langle o\rangle) = \mathfrak{C}(\mathfrak{C}(\mathtt{C}\langle o\rangle)[\![\alpha \setminus^{\alpha'} s]\!])$ y $\mathfrak{C}(\mathtt{O}\langle o'\rangle) = \mathfrak{C}(\mathfrak{C}(\mathtt{C}\langle o'\rangle)[\![\alpha \setminus^{\alpha'} s]\!])$. Por *h.i.* $\mathfrak{C}(\mathtt{C}\langle o\rangle) \simeq \mathfrak{C}(\mathtt{C}\langle o'\rangle)$. Finalmente, se concluye por Lem. C.0.52 (2).
- $\mathtt{O} = c[\![\alpha \setminus^{\alpha'} \mathtt{S}]\!]$. Luego, se tienen cuatro casos posible:
  1. $\mathfrak{C}(c) = \mathtt{LCC}\langle [\alpha]\, t\rangle$ con $\alpha \notin t$, $\alpha \notin \mathtt{LCC}$, $\mathsf{fc}(\alpha', \mathtt{LCC})$ y $\mathsf{fc}(\mathtt{S}\langle o\rangle, \mathtt{LCC})$. Más aún, se tiene también $\mathsf{fc}(\mathtt{S}\langle o'\rangle, \mathtt{LCC})$. Luego, $\mathfrak{C}(\mathtt{O}\langle o\rangle) = \mathfrak{C}(\mathtt{LCC}\langle [\alpha]\, t\rangle[\![\alpha \setminus^{\alpha'} \mathtt{S}\langle o\rangle]\!]) = \mathtt{LCC}\langle [\alpha']\, \mathfrak{C}(t :: \mathtt{S}\langle o\rangle)\rangle$. Del mismo modo, $\mathfrak{C}(\mathtt{O}\langle o'\rangle) = \mathtt{LCC}\langle [\alpha']\, \mathfrak{C}(t :: \mathtt{S}\langle o'\rangle)\rangle$. Por Lem. C.0.28, $\mathsf{sz}(\mathtt{O}\langle o\rangle) = \mathsf{sz}(c[\![\alpha \setminus^{\alpha'} \mathtt{S}\langle o\rangle]\!]) \geq \mathsf{sz}(\mathtt{LCC}\langle [\alpha]\, t\rangle[\![\alpha \setminus^{\alpha'} \mathtt{S}\langle o\rangle]\!]) \geq \mathsf{sz}(\mathtt{LCC}\langle [\alpha']\, t :: \mathtt{S}\langle o'\rangle\rangle)$, por lo que $\mathsf{sz}(\mathtt{O}) > \mathsf{sz}(t :: \mathtt{S})$. Finalmente, se concluye pues por *h.i.* se tiene $\mathfrak{C}(t :: \mathtt{S}\langle o\rangle) \simeq \mathfrak{C}(t :: \mathtt{S}\langle o'\rangle)$.

2. $\mathfrak{C}(c) = \mathtt{LCC}\langle c'[\![\beta \setminus^{\alpha} s]\!]\rangle$ con $\alpha \notin c'$, $\alpha \notin \mathtt{LCC}$, $\alpha \notin s$, $\mathsf{fc}(\alpha', \mathtt{LCC})$ y $\mathsf{fc}(\mathtt{S}\langle o\rangle, \mathtt{LCC})$. Más aún, se tiene también $\mathsf{fc}(\mathtt{S}\langle o'\rangle, \mathtt{LCC})$. Luego, se tiene $\mathfrak{C}(\mathtt{O}\langle o\rangle) = \mathfrak{C}(\mathtt{LCC}\langle c'[\![\beta \setminus^{\alpha} s]\!]\rangle[\![\alpha \setminus^{\alpha'} \mathtt{S}\langle o\rangle]\!]) = \mathtt{LCC}\langle c'[\![\beta \setminus^{\alpha'} s \cdot \mathfrak{C}(\mathtt{S}\langle o\rangle)]\!]\rangle$. Del mismo modo, $\mathfrak{C}(\mathtt{O}\langle o'\rangle) = \mathtt{LCC}\langle c'[\![\beta \setminus^{\alpha'} s \cdot \mathfrak{C}(\mathtt{S}\langle o'\rangle)]\!]\rangle$. Se concluye pues por *h.i.* se tiene $\mathfrak{C}(\mathtt{S}\langle o\rangle) \simeq \mathfrak{C}(\mathtt{S}\langle o'\rangle)$.
3. $\mathfrak{C}(c) = \mathtt{LCC}\langle c'[\beta \setminus \alpha]\rangle$ con $\alpha \notin c'$, $\alpha \notin \mathtt{LCC}$, $\mathsf{fc}(\alpha', \mathtt{LCC})$ y $\mathsf{fc}(\mathtt{S}\langle o\rangle, \mathtt{LCC})$. Más aún, se tiene también $\mathsf{fc}(\mathtt{S}\langle o'\rangle, \mathtt{LCC})$. Por lo tanto, se tiene $\mathfrak{C}(\mathtt{O}\langle o\rangle) = \mathfrak{C}(\mathtt{LCC}\langle c'[\beta \setminus \alpha]\rangle[\![\alpha \setminus^{\alpha'} \mathtt{S}]\!]) = \mathtt{LCC}\langle \mathfrak{C}(c'[\![\beta \setminus^{\alpha} \mathtt{S}\langle o\rangle]\!])[\alpha \setminus \alpha']\rangle$. Del mismo modo, $\mathfrak{C}(\mathtt{O}\langle o'\rangle) = \mathtt{LCC}\langle \mathfrak{C}(c'[\![\beta \setminus^{\alpha} \mathtt{S}\langle o'\rangle]\!])[\alpha \setminus \alpha']\rangle$. Además, por Lem. C.0.28, se tiene $\mathsf{sz}(\mathtt{O}\langle o\rangle) = \mathsf{sz}(c[\![\alpha \setminus^{\alpha'} \mathtt{S}\langle o\rangle]\!]) \geq \mathsf{sz}(\mathtt{LCC}\langle c'[\beta \setminus \alpha]\rangle[\![\alpha \setminus^{\alpha'} \mathtt{S}]\!]) \geq \mathsf{sz}(\mathtt{LCC}\langle c'[\![\beta \setminus^{\alpha} \mathtt{S}\langle o\rangle]\!][\alpha \setminus \alpha']\rangle)$, por lo que $\mathsf{sz}(\mathtt{O}) > \mathsf{sz}(c'[\![\beta \setminus^{\alpha} \mathtt{S}]\!])$. Finalmente, se concluye pues por *h.i.* se tiene $\mathfrak{C}(c'[\![\beta \setminus^{\alpha} \mathtt{S}\langle o\rangle]\!]) \simeq \mathfrak{C}(c'[\![\beta \setminus^{\alpha} \mathtt{S}\langle o\rangle]\!])$.
4. De lo contrario, se tienen $\mathfrak{C}(\mathtt{O}\langle o\rangle) = \mathfrak{C}(c)[\![\alpha \setminus^{\alpha'} \mathfrak{C}(\mathtt{S}\langle o\rangle)]\!]$ y $\mathfrak{C}(\mathtt{O}\langle o'\rangle) = \mathfrak{C}(c)[\![\alpha \setminus^{\alpha'} \mathfrak{C}(\mathtt{S}\langle o'\rangle)]\!]$. Se concluye pues por *h.i.* se tiene $\mathfrak{C}(\mathtt{S}\langle o\rangle) \simeq \mathfrak{C}(\mathtt{S}\langle o'\rangle)$.

- $\mathtt{O} = \mathtt{C}[\alpha \setminus \beta]$. Luego, $\mathfrak{C}(\mathtt{O}\langle o\rangle) = \mathfrak{C}(\mathtt{C}\langle o\rangle)[\alpha \setminus \beta]$ y $\mathfrak{C}(\mathtt{O}\langle o'\rangle) = \mathfrak{C}(\mathtt{C}\langle o'\rangle)[\alpha \setminus \beta]$. Se concluye pues por *h.i.* se tiene $\mathfrak{C}(\mathtt{C}\langle o\rangle) \simeq \mathfrak{C}(\mathtt{C}\langle o'\rangle)$.
- $\mathtt{O} = \mathtt{T} \cdot s$. Luego, $\mathfrak{C}(\mathtt{O}\langle o\rangle) = \mathfrak{C}(\mathtt{T}\langle o\rangle) \cdot s$ y $\mathfrak{C}(\mathtt{O}\langle o'\rangle) = \mathfrak{C}(\mathtt{T}\langle o'\rangle) \cdot s$. Se concluye pues por *h.i.* se tiene $\mathfrak{C}(\mathtt{T}\langle o\rangle) \simeq \mathfrak{C}(\mathtt{T}\langle o'\rangle)$.
- $\mathtt{O} = t \cdot \mathtt{S}$. Luego, $\mathfrak{C}(\mathtt{O}\langle o\rangle) = t \cdot \mathfrak{C}(\mathtt{S}\langle o\rangle)$ y $\mathfrak{C}(\mathtt{O}\langle o'\rangle) = t \cdot \mathfrak{C}(\mathtt{S}\langle o'\rangle)$. Se concluye pues por *h.i.* se tiene $\mathfrak{C}(\mathtt{S}\langle o\rangle) \simeq \mathfrak{C}(\mathtt{S}\langle o'\rangle)$.

□

Lema C.0.53. *Sean $o, o' \in \mathbb{O}_{\Lambda M}$ y $u, u' \in \mathbb{T}_{\Lambda M}$ tal que $o \simeq o'$ y $u \simeq u'$.*

1. *Luego, $\mathfrak{C}(\{x \setminus u\}o) \simeq \mathfrak{C}(\{x \setminus u\}o')$.*
2. *Luego, $\mathfrak{C}(\{x \setminus u\}o) \simeq \mathfrak{C}(\{x \setminus u'\}o)$.*

Demostración. 1. La prueba es por inducción en $o \simeq o'$. Por definición esto implica verificar cuatro condiciones: reflexividad, transitividad, simetría y congruencia (*i.e.* clausura por contextos):

Reflexividad Luego, $o' = o$ por lo que $\mathfrak{C}(\{x \setminus u\}o) = \mathfrak{C}(\{x \setminus u\}o')$. El resultado es inmediato por reflexividad de $\simeq$.

Transitividad Luego, existe $p \in \mathbb{O}_{\Lambda M}$ tal que $o \simeq p$ y $p \simeq o'$. Por *h.i.* se tienen $\mathfrak{C}(\{x \setminus u\}o) \simeq \mathfrak{C}(\{x \setminus u\}p)$ y $\mathfrak{C}(\{x \setminus u\}p) \simeq \mathfrak{C}(\{x \setminus u\}o')$. Se concluye por transitividad de $\simeq$.

Simetría Luego, $o' \simeq o$. Por *h.i.* se tiene $\mathfrak{C}(\{x \setminus u\}o') \simeq \mathfrak{C}(\{x \setminus u\}o)$. Se concluye por simetría de $\simeq$.

Congruencia La congruencia se demuestra por inducción en el contexto de clausura $\mathtt{O}$ tal que $o = \mathtt{O}\langle p\rangle$ y $o' = \mathtt{O}\langle p'\rangle$ con $p \simeq_* p'$, donde $\simeq_*$ es una regla de la Fig. 4.13.

- $\mathtt{O} = \Box$. Luego, $o = p \simeq_* p' = o'$. Más aún, es el caso de $o, o' \in \mathbb{T}_{\Lambda M}$. Se tienen solo dos reglas aplicables a términos:
  - $\simeq_{\mathtt{exsubs}}$. Luego, $o = \mathtt{LTT}\langle t\rangle[y \setminus v]$ y $o' = \mathtt{LTT}\langle t[y \setminus v]\rangle$ con $y \notin \mathtt{LTT}$ y $\mathsf{fc}(v, \mathtt{LTT})$. Sin perdida de generalidad, se asume $y \notin \mathsf{fv}(u)$ y $\mathsf{fc}(u, \mathtt{LTT})$. Luego, $\{x \setminus u\}o = \mathtt{LTT}'\langle t'\rangle[y \setminus v']$ y $\{x \setminus u\}o' = \mathtt{LTT}'\langle t'[y \setminus v']\rangle$ con $y \notin \mathtt{LTT}'$ y $\mathsf{fc}(v', \mathtt{LTT}')$, donde $t' = \{x \setminus u\}t$, $\mathtt{LTT}' = \{x \setminus u\}\mathtt{LTT}'$ y $v' = \{x \setminus u\}v$. Se concluye por Lem. 4.6.2 (1), pues $\mathfrak{C}(\mathtt{LTT}'\langle t'\rangle[y \setminus v']) \simeq \mathfrak{C}(\mathtt{LTT}'\langle t'[y \setminus v']\rangle)$.
  - $\simeq_{\theta}$. Luego, $o = \mu\alpha.[\alpha]\, t$ y $o' = t$ con $\alpha \notin t$. Sin perdida de generalidad se asume $\alpha \notin u$, por lo que $\{x \setminus u\}o = \mu\alpha.[\alpha]\, \{x \setminus u\}t$. Finalmente, se concluye $\mathfrak{C}(\{x \setminus u\}o) = \mu\alpha.[\alpha]\, \mathfrak{C}(\{x \setminus u\}t) \simeq_{\theta} \mathfrak{C}(\{x \setminus u\}t) = \mathfrak{C}(\{x \setminus u\}o')$.

- $\mathtt{O} = \boxdot$. Luego, $o = p \simeq_* p' = o'$. Más aún, es el caso de $o, o' \in \mathbb{C}_{\Lambda M}$. Se tienen cuatro reglas aplicables a comandos:
  - $\simeq_{\mathtt{exrepl}}$. Luego, $o = \mathtt{LCC}\langle c\rangle[\![\alpha \setminus^{\alpha'} s]\!]$ y $o' = \mathtt{LCC}\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle$ con $\alpha \notin \mathtt{LCC}$, $\mathsf{fc}(\alpha', \mathtt{LCC})$ y $\mathsf{fc}(s, \mathtt{LCC})$. Sin perdida de generalidad, se asume $\alpha \notin \mathsf{fn}(u)$ y $\mathsf{fc}(u, \mathtt{LCC})$. Luego, $\{x \setminus u\}o = \mathtt{LCC}'\langle c'\rangle[\![\alpha \setminus^{\alpha'} s']\!]$ y $\{x \setminus u\}o' = \mathtt{LCC}'\langle c'[\![\alpha \setminus^{\alpha'} s']\!]\rangle$ con $\alpha \notin \mathtt{LCC}'$ y $\mathsf{fc}(s', \mathtt{LCC}')$, donde $c' = \{x \setminus u\}t$, $\mathtt{LCC}' = \{x \setminus u\}\mathtt{LCC}$ y $s' = \{x \setminus u\}s$. Finalmente, se concluye por Lem. 4.6.2 (2), pues $\mathfrak{C}(\mathtt{LCC}'\langle c'\rangle[\![\alpha \setminus^{\alpha'} s']\!]) \simeq \mathfrak{C}(\mathtt{LCC}'\langle c'[\![\alpha \setminus^{\alpha'} s']\!]\rangle)$.
  - $\simeq_{\mathtt{exren}}$. Luego, $o = \mathtt{LCC}\langle c\rangle[\alpha \backslash \beta]$ y $o' = \mathtt{LCC}\langle c[\alpha \setminus \beta]\rangle$ con $\alpha \notin \mathtt{LCC}$ y $\mathsf{fc}(\beta, \mathtt{LCC})$. Sin perdida de generalidad, se asume $\alpha \notin \mathsf{fn}(u)$ y $\mathsf{fc}(u, \mathtt{LCC})$. Luego, $\{x \setminus u\}o = \mathtt{LCC}'\langle c'\rangle[\alpha \backslash \beta]$ y $\{x \setminus u\}o' = \mathtt{LCC}'\langle c'[\alpha \setminus \beta]\rangle$ con $\alpha \notin \mathtt{LCC}'$, donde $c' = \{x \setminus u\}c$, $\mathtt{LCC}' = \{x \setminus u\}\mathtt{LCC}$. Se concluye por Lem. 4.6.2 (2), pues $\mathfrak{C}(\mathtt{LCC}'\langle c'\rangle[\alpha \setminus \beta]) \simeq \mathfrak{C}(\mathtt{LCC}'\langle c'[\alpha \setminus \beta]\rangle)$.
  - $\simeq_{\mathtt{pp}}$. Luego, $o = [\alpha']\,\lambda z.\mu\alpha.[\beta']\,\lambda y.\mu\beta.c$ y $o' = [\beta']\,\lambda y.\mu\beta.[\alpha']\,\lambda z.\mu\alpha.c$ con $\beta \neq \alpha'$ y $\alpha \neq \beta'$. Sin perdida de generalidad, se asume también $z \notin u$, $y \notin u$, $\alpha \notin u$ y $\beta \notin u$. Luego, $\{x \setminus u\}o = [\alpha']\,\lambda z.\mu\alpha.[\beta']\,\lambda y.\mu\beta.c'$ y $\{x \setminus u\}o' = [\beta']\,\lambda y.\mu\beta.[\alpha']\,\lambda z.\mu\alpha.c'$ con $c' = \{x \setminus u\}c$. Finalmente, se concluye $\mathfrak{C}(\{x \setminus u\}o) = [\alpha']\,\lambda z.\mu\alpha.[\beta']\,\lambda y.\mu\beta.\mathfrak{C}(c') \simeq_{\mathtt{pp}} [\beta']\,\lambda y.\mu\beta.[\alpha']\,\lambda z.\mu\alpha.\mathfrak{C}(c') = \mathfrak{C}(\{x \setminus u\}o')$.
  - $\simeq_{\rho}$. Luego, $o = [\beta]\,\mu\alpha.c$ y $o' = c[\alpha \setminus \beta]$. Sin perdida de generalidad, se asume $\alpha \notin u$. Entonces, $\{x \setminus u\}o = [\beta]\,\mu\alpha.c'$ y $\{x \setminus u\}o' = c'[\alpha \setminus \beta]$ con $c' = \{x \setminus u\}c$. Finalmente, se concluye $\mathfrak{C}(\{x \setminus u\}o) = [\beta]\,\mu\alpha.\mathfrak{C}(c') \simeq_{\rho} \mathfrak{C}(c')[\alpha \setminus \beta] = \mathfrak{C}(\{x \setminus u\}o')$.
- $\mathtt{O} = \mathtt{T}\,v$. Luego, $o = \mathtt{T}\langle p\rangle\,v$ y $o' = \mathtt{T}\langle p'\rangle\,v$. Sean $t = \mathtt{T}\langle p\rangle$ y $t' = \mathtt{T}\langle p'\rangle$. Se tiene entonces $t \simeq t'$. Por *h.i.* se tiene $\mathfrak{C}(\{x \setminus u\}t) \simeq \mathfrak{C}(\{x \setminus u\}t')$. Finalmente, se apela al Lem. 4.8.1 para concluir $\mathfrak{C}(\{x \setminus u\}o) = \mathfrak{C}(\{x \setminus u\}t\,\{x \setminus u\}v) = \mathfrak{C}(\mathfrak{C}(\{x \setminus u\}t)\,\{x \setminus u\}v) \simeq \mathfrak{C}(\mathfrak{C}(\{x \setminus u\}t')\,\{x \setminus u\}v) = \mathfrak{C}(\{x \setminus u\}o')$.
- $\mathtt{O} = t\,\mathtt{T}$. Luego, $o = t\,v$ y $o' = t\,v'$ con $v \simeq v'$. Por *h.i.* se tiene $\mathfrak{C}(\{x \setminus u\}v) \simeq \mathfrak{C}(\{x \setminus u\}v')$. Se concluye por Lem. 4.8.1, pues $\mathfrak{C}(\{x \setminus u\}o) = \mathfrak{C}(\{x \setminus u\}t\,\mathfrak{C}(\{x \setminus u\}v)) \simeq \mathfrak{C}(\{x \setminus u\}t\,\mathfrak{C}(\{x \setminus u\}v')) = \mathfrak{C}(\{x \setminus u\}o')$.
- $\mathtt{O} = \lambda y.\mathtt{T}$. Luego, $o = \lambda y.t$ y $o' = \lambda y.t'$ con $t \simeq t'$. Por *h.i.* se tiene $\mathfrak{C}(\{x \setminus u\}t) \simeq \mathfrak{C}(\{x \setminus u\}t')$, por lo que se concluye $\mathfrak{C}(\{x \setminus u\}o) = \lambda y.\mathfrak{C}(\{x \setminus u\}t) \simeq \lambda y.\mathfrak{C}(\{x \setminus u\}t') = \mathfrak{C}(\{x \setminus u\}o')$.
- $\mathtt{O} = \mu\alpha.\mathtt{C}$. Luego, $o = \mu\alpha.c$ y $o' = \mu\alpha.c'$ con $c \simeq c'$. Por *h.i.* se tiene $\mathfrak{C}(\{x \setminus u\}c) \simeq \mathfrak{C}(\{x \setminus u\}c')$, por lo que se concluye $\mathfrak{C}(\{x \setminus u\}o) = \mu\alpha.\mathfrak{C}(\{x \setminus u\}c) \simeq \mu\alpha.\mathfrak{C}(\{x \setminus u\}c') = \mathfrak{C}(\{x \setminus u\}o')$.
- $\mathtt{O} = \mathtt{T}[y \setminus v]$. Luego, $o = t[y \setminus v]$ y $o' = t'[y \setminus v]$ con $t \simeq t'$. Por *h.i.* se tiene $\mathfrak{C}(\{x \setminus u\}t) \simeq \mathfrak{C}(\{x \setminus u\}t')$ y se concluye $\mathfrak{C}(\{x \setminus u\}o) = \mathfrak{C}(\{x \setminus u\}t)[y \setminus \mathfrak{C}(\{x \setminus u\}v)] \simeq \mathfrak{C}(\{x \setminus u\}t')[y \setminus \mathfrak{C}(\{x \setminus u\}v)] = \mathfrak{C}(\{x \setminus u\}o')$.
- $\mathtt{O} = t[y \setminus \mathtt{T}]$. Luego, $o = t[y \setminus v]$ y $o' = t[y \setminus v']$ con $v \simeq v'$. Por *h.i.* se tiene $\mathfrak{C}(\{x \setminus u\}v) \simeq \mathfrak{C}(\{x \setminus u\}v')$ y se concluye $\mathfrak{C}(\{x \setminus u\}o) = \mathfrak{C}(\{x \setminus u\}t)[y \setminus \mathfrak{C}(\{x \setminus u\}v)] \simeq \mathfrak{C}(\{x \setminus u\}t)[y \setminus \mathfrak{C}(\{x \setminus u\}v')] = \mathfrak{C}(\{x \setminus u\}o')$.
- $\mathtt{O} = [\alpha]\,\mathtt{T}$. Luego, $o = [\alpha]\,t$ y $o' = [\alpha]\,r'$ con $t \simeq t'$. Por *h.i.* se tiene $\mathfrak{C}(\{x \setminus u\}t) \simeq \mathfrak{C}(\{x \setminus u\}t')$, por lo que se concluye $\mathfrak{C}(\{x \setminus u\}o) = [\alpha]\,\mathfrak{C}(\{x \setminus u\}t) \simeq \mu\alpha.\mathfrak{C}(\{x \setminus u\}t') = \mathfrak{C}(\{x \setminus u\}o')$.
- $\mathtt{O} = \mathtt{C}[\![\alpha \setminus^{\alpha'} s]\!]$. Luego, $o = c[\![\alpha \setminus^{\alpha'} s]\!]$ y $o' = c'[\![\alpha \setminus^{\alpha'} s]\!]$ con $c \simeq c'$. Por *h.i.* se tiene $\mathfrak{C}(\{x \setminus u\}c) \simeq \mathfrak{C}(\{x \setminus u\}c')$. Más aún, dado que la sustitución no puede generar nuevos NPW-redexes con α, se concluye $\mathfrak{C}(\{x \setminus u\}o) = \mathfrak{C}(\{x \setminus u\}c)[\![\alpha \setminus^{\alpha'} \mathfrak{C}(\{x \setminus u\}s)]\!] \simeq \mathfrak{C}(\{x \setminus u\}c')[\![\alpha \setminus^{\alpha'} \mathfrak{C}(\{x \setminus u\}s)]\!] = \mathfrak{C}(\{x \setminus u\}o')$.
- $\mathtt{O} = c[\![\alpha \setminus^{\alpha'} \mathtt{S}]\!]$. Luego, $o = c[\![\alpha \setminus^{\alpha'} s]\!]$ y $o' = c[\![\alpha \setminus^{\alpha'} s']\!]$ con $s \simeq s'$. Por *h.i.* se tiene $\mathfrak{C}(\{x \setminus u\}s) \simeq \mathfrak{C}(\{x \setminus u\}s')$. Más aún, dado que la sustitución no puede generar

nuevos NPW-redexes con α, se concluye $\mathfrak{C}(\{x \setminus u\}o) = \mathfrak{C}(\{x \setminus u\}c)[\![\alpha \setminus^{\alpha'} \mathfrak{C}(\{x \setminus u\}s)]\!] \simeq \mathfrak{C}(\{x \setminus u\}c)[\![\alpha \setminus^{\alpha'} \mathfrak{C}(\{x \setminus u\}s')]\!] = \mathfrak{C}(\{x \setminus u\}o')$.

- $\mathtt{O} = \mathtt{C}[\alpha \setminus \beta]$. Luego, $o = c[\alpha \setminus \beta]$ y $o' = c'[\alpha \setminus \beta]$ con $c \simeq c'$. Por *h.i.* se tiene $\mathfrak{C}(\{x \setminus u\}c) \simeq \mathfrak{C}(\{x \setminus u\}c')$, por lo que se concluye $\mathfrak{C}(\{x \setminus u\}o) = \mathfrak{C}(\{x \setminus u\}c)[\alpha \setminus \beta] \simeq \mathfrak{C}(\{x \setminus u\}c')[\alpha \setminus \beta] = \mathfrak{C}(\{x \setminus u\}o')$.
- $\mathtt{O} = \mathtt{T} \cdot s$. Luego, $o = t \cdot s$ y $o' = t' \cdot s$ con $t \simeq t'$. Por *h.i.* se tiene $\mathfrak{C}(\{x \setminus u\}t) \simeq \mathfrak{C}(\{x \setminus u\}t')$, por lo que se concluye $\mathfrak{C}(\{x \setminus u\}o) = \mathfrak{C}(\{x \setminus u\}t) \cdot \mathfrak{C}(\{x \setminus u\}s) \simeq \mathfrak{C}(\{x \setminus u\}t') \cdot \mathfrak{C}(\{x \setminus u\}s) = \mathfrak{C}(\{x \setminus u\}o')$.
- $\mathtt{O} = t \cdot \mathtt{S}$. Luego, $o = t \cdot s$ y $o' = t \cdot s'$ con $s \simeq s'$. Por *h.i.* se tiene $\mathfrak{C}(\{x \setminus u\}s) \simeq \mathfrak{C}(\{x \setminus u\}s')$, por lo que se concluye $\mathfrak{C}(\{x \setminus u\}o) = \mathfrak{C}(\{x \setminus u\}t) \cdot \mathfrak{C}(\{x \setminus u\}s) \simeq \mathfrak{C}(\{x \setminus u\}t) \cdot \mathfrak{C}(\{x \setminus u\}s') = \mathfrak{C}(\{x \setminus u\}o')$.

2. Por inducción en o.

- $o = x$. Luego, $\{x \setminus u\}o = u$ y $\{x \setminus u'\}o = u'$. Se concluye dado que $u \simeq u'$ implica $\mathfrak{C}(u) = u$ y $\mathfrak{C}(u') = u'$. Por lo tanto, $\mathfrak{C}(\{x \setminus u\}o) = u \simeq u' = \mathfrak{C}(\{x \setminus u\}o')$.
- $o = y \neq x$. Luego, $\{x \setminus u\}o = y$ y $\{x \setminus u'\}o = y$. Se concluye por reflexividad de $\simeq$.
- $o = t\,v$. Luego, $\{x \setminus u\}o = \{x \setminus u\}t\,\{x \setminus u\}v$ y $\{x \setminus u'\}o = \{x \setminus u'\}t\,\{x \setminus u'\}v$. Por *h.i.* se tienen $\mathfrak{C}(\{x \setminus u\}t) \simeq \mathfrak{C}(\{x \setminus u'\}t)$ y $\mathfrak{C}(\{x \setminus u\}v) \simeq \mathfrak{C}(\{x \setminus u'\}v)$. Finalmente, se concluye por Lem. 4.8.1, pues $\mathfrak{C}(\{x \setminus u\}o) = \mathfrak{C}(\{x \setminus u\}t\,\{x \setminus u\}v) = \mathfrak{C}(\mathfrak{C}(\{x \setminus u\}t)\,\mathfrak{C}(\{x \setminus u\}v)) \simeq \mathfrak{C}(\mathfrak{C}(\{x \setminus u'\}t)\,\mathfrak{C}(\{x \setminus u'\}v)) = \mathfrak{C}(\{x \setminus u'\}o)$.
- $o = \lambda y.t$. Luego, $\{x \setminus u\}o = \lambda y.\{x \setminus u\}t$ y $\{x \setminus u'\}o = \lambda y.\{x \setminus u'\}t$. Por *h.i.* se tiene $\mathfrak{C}(\{x \setminus u\}t) \simeq \mathfrak{C}(\{x \setminus u'\}t)$ y se concluye $\mathfrak{C}(\{x \setminus u\}o) = \lambda y.\mathfrak{C}(\{x \setminus u\}t) \simeq \lambda y.\mathfrak{C}(\{x \setminus u'\}t) = \mathfrak{C}(\{x \setminus u'\}o)$.
- $o = \mu\alpha.c$. Luego, $\{x \setminus u\}o = \mu\alpha.\{x \setminus u\}c$ y $\{x \setminus u'\}o = \mu\alpha.\{x \setminus u'\}c$. Por *h.i.* se tiene $\mathfrak{C}(\{x \setminus u\}c) \simeq \mathfrak{C}(\{x \setminus u'\}c)$ y se concluye $\mathfrak{C}(\{x \setminus u\}o) = \mu\alpha.\mathfrak{C}(\{x \setminus u\}c) \simeq \mu\alpha.\mathfrak{C}(\{x \setminus u'\}c) = \mathfrak{C}(\{x \setminus u'\}o)$.
- $o = t[y \setminus v]$. Luego, $\{x \setminus u\}o = \{x \setminus u\}t[y \setminus \{x \setminus u\}v]$ y $\{x \setminus u'\}o = \{x \setminus u'\}t[y \setminus \{x \setminus u'\}v]$. Por *h.i.* se tienen $\mathfrak{C}(\{x \setminus u\}t) \simeq \mathfrak{C}(\{x \setminus u'\}t)$ y $\mathfrak{C}(\{x \setminus u\}v) \simeq \mathfrak{C}(\{x \setminus u'\}v)$. Finalmente, se concluye pues $\mathfrak{C}(\{x \setminus u\}o) = \mathfrak{C}(\{x \setminus u\}t)[y \setminus \mathfrak{C}(\{x \setminus u\}v)] \simeq \mathfrak{C}(\{x \setminus u'\}t)[y \setminus \mathfrak{C}(\{x \setminus u'\}v)] = \mathfrak{C}(\{x \setminus u'\}o)$.
- $o = [\alpha]\,t$. Luego, $\{x \setminus u\}o = [\alpha]\,\{x \setminus u\}t$ y $\{x \setminus u'\}o = [\alpha]\,\{x \setminus u'\}t$. Por *h.i.* $\mathfrak{C}(\{x \setminus u\}t) \simeq \mathfrak{C}(\{x \setminus u'\}t)$, por lo que se concluye $\mathfrak{C}(\{x \setminus u\}o) = [\alpha]\,\mathfrak{C}(\{x \setminus u\}t) \simeq [\alpha]\,\mathfrak{C}(\{x \setminus u'\}t) = \mathfrak{C}(\{x \setminus u'\}o)$.
- $o = c[\![\alpha \setminus^{\alpha'} s]\!]$. Luego, $\{x \setminus u\}o = \{x \setminus u\}c[\![\alpha \setminus^{\alpha'} \{x \setminus u\}s]\!]$ y $\{x \setminus u'\}o = \{x \setminus u'\}c[\![\alpha \setminus^{\alpha'} \{x \setminus u'\}s]\!]$. Por *h.i.* se tienen $\mathfrak{C}(\{x \setminus u\}c) \simeq \mathfrak{C}(\{x \setminus u'\}c)$ y $\mathfrak{C}(\{x \setminus u\}s) \simeq \mathfrak{C}(\{x \setminus u'\}s)$. Más aún, la sustitución no puede generar nuevos NPW-redexes con α, por lo que se concluye $\mathfrak{C}(\{x \setminus u\}o) = \mathfrak{C}(\{x \setminus u\}c)[\![\alpha \setminus^{\alpha'} \mathfrak{C}(\{x \setminus u\}s)]\!] \simeq \mathfrak{C}(\{x \setminus u'\}c)[\![\alpha \setminus^{\alpha'} \mathfrak{C}(\{x \setminus u'\}s)]\!] = \mathfrak{C}(\{x \setminus u\}o')$.
- $o = c[\alpha \setminus \beta]$. Luego, $\{x \setminus u\}o = \{x \setminus u\}c[\alpha \setminus \beta]$ y $\{x \setminus u'\}o = \{x \setminus u'\}c[\alpha \setminus \beta]$. Por *h.i.* se tiene $\mathfrak{C}(\{x \setminus u\}c) \simeq \mathfrak{C}(\{x \setminus u'\}c)$. Más aún, la sustitución no puede generar nuevos NPW-redexes con α, por lo que se concluye $\mathfrak{C}(\{x \setminus u\}o) = \mathfrak{C}(\{x \setminus u\}c)[\alpha \setminus \beta] \simeq \mathfrak{C}(\{x \setminus u'\}c)[\alpha \setminus \beta] = \mathfrak{C}(\{x \setminus u\}o')$.
- $o = t \cdot s$. Luego, $\{x \setminus u\}o = \{x \setminus u\}t \cdot \{x \setminus u\}s$ y $\{x \setminus u'\}o = \{x \setminus u'\}t \cdot \{x \setminus u'\}s$. Por *h.i.* se tienen $\mathfrak{C}(\{x \setminus u\}t) \simeq \mathfrak{C}(\{x \setminus u'\}t)$ y $\mathfrak{C}(\{x \setminus u\}s) \simeq \mathfrak{C}(\{x \setminus u'\}s)$. Finalmente, se concluye pues $\mathfrak{C}(\{x \setminus u\}o) = \mathfrak{C}(\{x \setminus u\}t) \cdot \mathfrak{C}(\{x \setminus u\}s) \simeq \mathfrak{C}(\{x \setminus u'\}t) \cdot \mathfrak{C}(\{x \setminus u'\}s) = \mathfrak{C}(\{x \setminus u'\}o)$.

□

Lema C.0.54. *Sean $o, o', s, s' \in \mathbb{O}_{\Lambda M}$ con s y s' stacks tal que $o \simeq o'$ y $s \simeq s'$.*

1. *Luego,* $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o) \simeq \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o')$.
2. *Luego,* $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o) \simeq \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s'\}\!\} o)$.

Demostración. 1. La prueba es por inducción en $o \simeq o'$. Por definición esto implica verificar cuatro condiciones: reflexividad, transitividad, simetría y congruencia (*i.e.* clausura por contextos):

Reflexividad Luego, $o' = o$ por lo que $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o) = \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o')$. El resultado es inmediato por reflexividad de $\simeq$.

Transitividad Luego, existe $p \in \mathbb{O}_{\Lambda M}$ tal que $o \simeq p$ y $p \simeq o'$. Por *h.i.* se tienen $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o) \simeq \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} p)$ y $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} p) \simeq \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o')$. Se concluye por transitividad de $\simeq$.

Simetría Luego, $o' \simeq o$. Por *h.i.* se tiene $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o') \simeq \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o)$. Se concluye por simetría de $\simeq$.

Congruencia La congruencia se demuestra por inducción en el contexto de clausura $\mathtt{O}$ tal que $o = \mathtt{O}\langle p\rangle$ y $o' = \mathtt{O}\langle p'\rangle$ con $p \simeq_* p'$, donde $\simeq_*$ es una regla de la Fig. 4.13.

- $\mathtt{O} = \Box$. Luego, $o = p \simeq_* p' = o'$. Más aún, es el caso de $o, o' \in \mathbb{T}_{\Lambda M}$. Se tienen solo dos reglas aplicables a términos:
  - $\simeq_{\mathtt{exsubs}}$. Luego, $o = \mathtt{LTT}\langle t\rangle[y \setminus v]$ y $o' = \mathtt{LTT}\langle t[y \setminus v]\rangle$ con $y \notin \mathtt{LTT}$ y $\mathsf{fc}(v, \mathtt{LTT})$. Sin perdida de generalidad, se asume $y \notin \mathsf{fv}(s)$, $\mathsf{fc}(s, \mathtt{LTT})$ y $\mathsf{fc}(\alpha', \mathtt{LTT})$. Luego, $\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o = \mathtt{LTT}'\langle t'\rangle[y \setminus v']$ y $\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o' = \mathtt{LTT}'\langle t'[y \setminus v']\rangle$ con $y \notin \mathtt{LTT}'$ y $\mathsf{fc}(v', \mathtt{LTT}')$, donde $t' = \{\!\{\alpha \setminus^{\alpha'} s\}\!\} t$, $\mathtt{LTT}' = \{\!\{\alpha \setminus^{\alpha'} s\}\!\} \mathtt{LTT}'$ y $v' = \{\!\{\alpha \setminus^{\alpha'} s\}\!\} v$. Se concluye por Lem. 4.6.2 (1), pues $\mathfrak{C}(\mathtt{LTT}'\langle t'\rangle[y \setminus v']) \simeq \mathfrak{C}(\mathtt{LTT}'\langle t'[y \setminus v']\rangle)$.
  - $\simeq_\theta$. Luego, $o = \mu\gamma.[\gamma]\, t$ y $o' = t$ con $\gamma \notin t$. Sin perdida de generalidad se asume $\gamma \neq \alpha'$ y $\gamma \notin s$, por lo que $\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o = \mu\alpha.[\alpha]\, \{\!\{\alpha \setminus^{\alpha'} s\}\!\} t$. Finalmente, se concluye $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o) = \mu\alpha.[\alpha]\, \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} t) \simeq_\theta \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} t) = \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o')$.
- $\mathtt{O} = \boxdot$. Luego, $o = p \simeq_* p' = o'$. Más aún, es el caso de $o, o' \in \mathbb{C}_{\Lambda M}$. Se tienen cinco reglas aplicables a comandos:
  - $\simeq_{\mathtt{exrepl}}$. Luego, $o = \mathtt{LCC}\langle c\rangle[\![\gamma \setminus^{\delta} s']\!]$ y $o' = \mathtt{LCC}\langle c[\![\gamma \setminus^{\delta} s']\!]\rangle$ con $\gamma \notin \mathtt{LCC}$, $\mathsf{fc}(\delta, \mathtt{LCC})$ y $\mathsf{fc}(s', \mathtt{LCC})$. Sin perdida de generalidad, se asume $\gamma \neq \alpha'$, $\gamma \notin \mathsf{fn}(s)$ y $\mathsf{fc}(s, \mathtt{LCC})$. Hay dos casos posibles:
    a) $\delta = \alpha$. Luego, se tienen $\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o = \mathtt{LCC}'\langle c'\rangle[\![\gamma \setminus^{\alpha'} s'' \cdot s]\!]$ y $\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o' = \mathtt{LCC}'\langle c'[\![\gamma \setminus^{\alpha'} s'' \cdot s]\!]\rangle$ con $\gamma \notin \mathtt{LCC}'$ y $\mathsf{fc}(s'', \mathtt{LCC}')$, donde $c' = \{\!\{\alpha \setminus^{\alpha'} s\}\!\} c$, $\mathtt{LCC}' = \{\!\{\alpha \setminus^{\alpha'} s\}\!\} \mathtt{LCC}$ y $s'' = \{\!\{\alpha \setminus^{\alpha'} s\}\!\} s'$. Entonces, se concluye por Lem. 4.6.2 (2), pues $\mathfrak{C}(\mathtt{LCC}'\langle c'\rangle[\![\gamma \setminus^{\alpha'} s'' \cdot s]\!]) \simeq \mathtt{LCC}'\langle c'[\![\gamma \setminus^{\alpha'} s'' \cdot s]\!]\rangle$.
    b) $\delta \neq \alpha$. Luego, $\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o = \mathtt{LCC}'\langle c'\rangle[\![\gamma \setminus^{\delta} s'']\!]$ y $\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o' = \mathtt{LCC}'\langle c'[\![\gamma \setminus^{\delta} s'']\!]\rangle$ con $\gamma \notin \mathtt{LCC}'$ y $\mathsf{fc}(s'', \mathtt{LCC}')$, donde $c' = \{\!\{\alpha \setminus^{\alpha'} s\}\!\} c$, $\mathtt{LCC}' = \{\!\{\alpha \setminus^{\alpha'} s\}\!\} \mathtt{LCC}$ y $s'' = \{\!\{\alpha \setminus^{\alpha'} s\}\!\} s'$. Como en el caso anterior, se concluye por Lem. 4.6.2 (2), dado que $\mathfrak{C}(\mathtt{LCC}'\langle c'\rangle[\![\gamma \setminus^{\delta} s'']\!]) \simeq \mathtt{LCC}'\langle c'[\![\gamma \setminus^{\delta} s'']\!]\rangle$.
  - $\simeq_{\mathtt{exren}}$. Luego, $o = \mathtt{LCC}\langle c\rangle[\gamma \setminus \delta]$ y $o' = \mathtt{LCC}\langle c[\gamma \setminus \delta]\rangle$ con $\gamma \notin \mathtt{LCC}$ y $\mathsf{fc}(\delta, \mathtt{LCC})$. Sin perdida de generalidad, se asume $\gamma \neq \alpha'$, $\gamma \notin \mathsf{fn}(s)$ y $\mathsf{fc}(s, \mathtt{LCC})$. Hay dos casos posibles:
    a) $\delta = \alpha$. Luego, se tienen $\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o = \mathtt{LCC}'\langle c'\rangle[\![\gamma \setminus^{\beta} s]\!][\beta \setminus \alpha']$ y $\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o' = \mathtt{LCC}'\langle c'[\![\gamma \setminus^{\beta} s]\!][\beta \setminus \alpha']\rangle$ con β un nombre fresco y $\gamma \notin \mathtt{LCC}'$, donde $c' = \{\!\{\alpha \setminus^{\alpha'} s\}\!\} c$, $\mathtt{LCC}' = \{\!\{\alpha \setminus^{\alpha'} s\}\!\} \mathtt{LCC}$. Entonces, se concluye apelando al Lem. 4.6.2 (2), pues se tiene $\mathfrak{C}(\mathtt{LCC}'\langle c'\rangle[\![\gamma \setminus^{\beta} s]\!][\beta \setminus \alpha']) \simeq \mathfrak{C}(\mathtt{LCC}'\langle c'[\![\gamma \setminus^{\beta} s]\!][\beta \setminus \alpha']\rangle)$.
    b) $\delta \neq \alpha$. Luego, $\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o = \mathtt{LCC}'\langle c'\rangle[\gamma \setminus \delta]$ y $\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o' = \mathtt{LCC}'\langle c'[\gamma \setminus \delta]\rangle$ con $\gamma \notin \mathtt{LCC}'$, donde $c' = \{\!\{\alpha \setminus^{\alpha'} s\}\!\} c$, $\mathtt{LCC}' = \{\!\{\alpha \setminus^{\alpha'} s\}\!\} \mathtt{LCC}$. Como en el caso anterior, se concluye por Lem. 4.6.2 (2), pues $\mathfrak{C}(\mathtt{LCC}'\langle c'\rangle[\gamma \setminus \delta]) \simeq \mathfrak{C}(\mathtt{LCC}'\langle c'[\gamma \setminus \delta]\rangle)$.

- $\simeq_{\mathtt{pp}}$. Luego, $o = [\gamma']\,\lambda x.\mu\gamma.[\delta']\,\lambda y.\mu\delta.c$ y $o' = [\delta']\,\lambda y.\mu\delta.[\gamma']\,\lambda x.\mu\gamma.c$ con $\delta \neq \gamma'$ y $\gamma \neq \delta'$. Sin perdida de generalidad, se asume también $x \notin s$, $y \notin s$, $\gamma \neq \alpha'$, $\delta \neq \alpha'$, $\gamma \notin s$ y $\delta \notin s$. Hay cuatro casos posibles:
  a) $\gamma' = \alpha$ y $\delta' = \alpha$. Luego, $\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o = [\alpha']\,(\lambda x.\mu\gamma.[\alpha']\,(\lambda y.\mu\delta.c') :: s) :: s$ y $\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o' = [\alpha']\,(\lambda y.\mu\delta.[\alpha']\,(\lambda x.\mu\gamma.c') :: s) :: s$ con $c' = \{\!\{\alpha \setminus^{\alpha'} s\}\!\}c$. Asumir $s = u \cdot s'$ (el caso $s = u$ es ligeramente más simple) y considerar nombres frescos γ'', δ''. Luego, $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o) = \mathfrak{C}([\alpha']\,(\mu\gamma''.([\alpha']\,(\mu\delta''.c'[\![\delta \setminus^{\delta''} s']\!])[y \setminus u])[\![\gamma \setminus^{\gamma''} s']\!])[x \setminus u])$, donde las sustituciones explícitas resultan de contraer los $\mathtt{dB}$-redexes sobre u; y los replacements explícitos son consecuencia de resolver los $\mathtt{dM}$-redexes sobre s' seguidos de los $\mathtt{P}$-redexes resultantes. Si $s = u$, no hay $\mathtt{dM}$-redexes a contraer, y solo se tienen las sustituciones explícitas. Análogamente, $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o') = \mathfrak{C}([\alpha']\,(\mu\delta''.([\alpha']\,(\mu\gamma''.c'[\![\gamma \setminus^{\gamma''} s']\!])[x \setminus u])[\![\delta \setminus^{\delta''} s']\!])[y \setminus u])$. Más aún, tomar $c'' = c'[\![\gamma \setminus^{\gamma''} s']\!][\![\delta \setminus^{\delta''} s']\!]$. Entonces, por Lem. 4.6.2, por un lado se tiene $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o) \simeq \mathfrak{C}([\alpha']\,\mu\gamma''.[\alpha']\,(\mu\delta''.c'')[x \setminus u][y \setminus u])$, mientras que por otro vale $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o') \simeq \mathfrak{C}([\alpha']\,\mu\delta''.[\alpha']\,(\mu\gamma''.c'')[x \setminus u][y \setminus u])$; apelando al ítem (1) del lema para posicionar las sustituciones explícitas y al ítem (2) para los replacements. Luego, se concluye aplicando dos veces $\simeq_\rho$ y Lem. 4.6.2 (2) para posicionar adecuadamente el renaming explícito introducido por la regla

$$\begin{array}{rcl}
\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o) & \simeq & \mathfrak{C}([\alpha']\,\mu\gamma''.[\alpha']\,(\mu\delta''.c'')[x \setminus u][y \setminus u]) \\
& = & [\alpha']\,\mu\gamma''.[\alpha']\,(\mu\delta''.\mathfrak{C}(c''))[x \setminus \mathfrak{C}(u)][y \setminus \mathfrak{C}(u)] \\
& \simeq_\rho & ([\alpha']\,(\mu\delta''.\mathfrak{C}(c''))[x \setminus \mathfrak{C}(u)][y \setminus \mathfrak{C}(u)])[\gamma'' \setminus \alpha'] \\
& \simeq & [\alpha']\,(\mu\delta''.\mathfrak{C}(c'')[\gamma'' \setminus \alpha'])[x \setminus \mathfrak{C}(u)][y \setminus \mathfrak{C}(u)] \\
& \simeq_\rho & [\alpha']\,(\mu\delta''.[\alpha']\,\mu\gamma''.\mathfrak{C}(c''))[x \setminus \mathfrak{C}(u)][y \setminus \mathfrak{C}(u)] \\
& = & \mathfrak{C}([\alpha']\,\mu\delta''.[\alpha']\,(\mu\gamma''.c'')[x \setminus u][y \setminus u]) \\
& \simeq & \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o')
\end{array}$$

  b) $\gamma' = \alpha$ y $\delta' \neq \alpha$. Luego, $\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o = [\alpha']\,(\lambda x.\mu\gamma.[\delta']\,\lambda y.\mu\delta.c') :: s$ y $\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o' = [\delta']\,\lambda y.\mu\delta.[\alpha']\,(\lambda x.\mu\gamma.c') :: s$ con $c' = \{\!\{\alpha \setminus^{\alpha'} s\}\!\}c$. Del mismo modo que en el caso anterior, sea $s = u \cdot s'$ con γ'' un nombre fresco. Luego, se tiene $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o) = \mathfrak{C}([\alpha']\,(\mu\gamma''.([\delta']\,\lambda y.\mu\delta.c')[\![\gamma \setminus^{\gamma''} s']\!])[x \setminus u])$. Más aún, por otro lado se tiene también $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o') = \mathfrak{C}([\delta']\,\lambda y.\mu\delta.[\alpha']\,(\mu\gamma''.c'[\![\gamma \setminus^{\gamma''} s']\!])[x \setminus u])$. En el caso $s = u$ simplemente se evita el replacement explícito. Por Lem. 4.6.2, de un lado se tiene $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o) \simeq \mathfrak{C}([\alpha']\,\mu\gamma''.[\delta']\,(\lambda y.\mu\delta.c'[\![\gamma \setminus^{\gamma''} s']\!])[x \setminus u])$ mientras que del otro $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o') \simeq \mathfrak{C}([\delta']\,(\lambda y.\mu\delta.[\alpha']\,\mu\gamma''.c'[\![\gamma \setminus^{\gamma''} s']\!])[x \setminus u])$, apelando al ítem (1) del lema para posicionar la sustitución explícita y al ítem (2) para el replacement. Finalmente, se concluye aplicando dos veces $\simeq_\rho$ y Lem. 4.6.2 (2) para posicionar adecuadamente el renaming explícito introducido por la regla

$$\begin{array}{rcl}
\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o) & \simeq & \mathfrak{C}([\alpha']\,\mu\gamma''.[\delta']\,(\lambda y.\mu\delta.c'[\![\gamma \setminus^{\gamma''} s']\!])[x \setminus u]) \\
& = & [\alpha']\,\mu\gamma''.[\delta']\,(\lambda y.\mu\delta.\mathfrak{C}(c'[\![\gamma \setminus^{\gamma''} s']\!]))[x \setminus \mathfrak{C}(u)] \\
& \simeq_\rho & ([\delta']\,(\lambda y.\mu\delta.\mathfrak{C}(c'[\![\gamma \setminus^{\gamma''} s']\!]))[x \setminus \mathfrak{C}(u)])[\gamma'' \setminus \alpha'] \\
& \simeq & [\delta']\,(\lambda y.\mu\delta.\mathfrak{C}(c'[\![\gamma \setminus^{\gamma''} s']\!])[\gamma'' \setminus \alpha'])[x \setminus \mathfrak{C}(u)] \\
& \simeq_\rho & [\delta']\,(\lambda y.\mu\delta.[\alpha']\,\mu\gamma''.\mathfrak{C}(c'[\![\gamma \setminus^{\gamma''} s']\!]))[x \setminus \mathfrak{C}(u)] \\
& = & \mathfrak{C}([\delta']\,(\lambda y.\mu\delta.[\alpha']\,\mu\gamma''.c'[\![\gamma \setminus^{\gamma''} s']\!])[x \setminus u]) \\
& \simeq & \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o')
\end{array}$$

  c) $\gamma' \neq \alpha$ y $\delta' = \alpha$. Este caso es análogo al anterior.
  d) $\gamma' \neq \alpha$ y $\delta' \neq \alpha$. Luego, $\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o = [\gamma']\,\lambda x.\mu\gamma.[\delta']\,\lambda y.\mu\delta.c'$ y $\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o' = [\delta']\,\lambda y.\mu\delta.[\gamma']\,\lambda x.\mu\gamma.c'$ con $c' = \{\!\{\alpha \setminus^{\alpha'} s\}\!\}c$. Se concluye entonces, $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o) = [\gamma']\,\lambda x.\mu\gamma.[\delta']\,\lambda y.\mu\delta.\mathfrak{C}(c') \simeq_{\mathtt{pp}} [\delta']\,\lambda y.\mu\delta.[\gamma']\,\lambda x.\mu\gamma.\mathfrak{C}(c') = \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o')$.

- $\simeq_\rho$. Luego, $o = [\delta]\,\mu\gamma.c$ y $o' = c[\gamma \setminus \delta]$. Sin perdida de generalidad, se asume $\gamma \neq \alpha'$ y $\gamma \notin s$. Hay dos casos posible:
  a) $\delta = \alpha$. Luego, $\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o = [\alpha]\,(\mu\gamma.c') :: s$ y $\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o' = c'[\![\gamma \setminus^{\gamma'} s]\!][\gamma' \setminus \alpha']$ con $c' = \{\!\{\alpha \setminus^{\alpha'} s\}\!\}c$ y γ' un nombre fresco. Más aún, se tienen $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o) = \mathfrak{C}([\alpha']\,\mu\gamma'.c'[\![\gamma \setminus^{\gamma'} s]\!])$, donde el replacement explícito resulta de contraer los dM-redexes sobre s seguidos de los (posibles) P-redexes resultantes. Luego, se concluye $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o) = [\alpha']\,\mu\gamma'.\mathfrak{C}(c'[\![\gamma \setminus^{\gamma'} s]\!]) \simeq_\rho \mathfrak{C}(c'[\![\gamma \setminus^{\gamma'} s]\!])[\gamma' \setminus \alpha'] = \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o')$.
  b) $\delta \neq \alpha$. Luego, $\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o = [\delta]\,\mu\gamma.c'$ y $\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o' = c'[\gamma \setminus \delta]$ con $c' = \{\!\{\alpha \setminus^{\alpha'} s\}\!\}c$. Se concluye entonces $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o) = [\delta]\,\mu\gamma.\mathfrak{C}(c') \simeq_\rho \mathfrak{C}(c')[\gamma \setminus \delta] = \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o')$.

- $\square = \mathtt{T}\,u$. Luego, $o = \mathtt{T}\langle p\rangle\,u$ y $o' = \mathtt{T}\langle p'\rangle\,u$. Sean $t = \mathtt{T}\langle p\rangle$ y $t' = \mathtt{T}\langle p'\rangle$. Se tiene entonces $t \simeq t'$. Por *h.i.* se tiene $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}t) \simeq \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}t')$. Finalmente, se apela al Lem. 4.8.1 y se concluye $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o) = \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}t\,\{\!\{\alpha \setminus^{\alpha'} s\}\!\}u) = \mathfrak{C}(\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}t)\,\{\!\{\alpha \setminus^{\alpha'} s\}\!\}u) \simeq \mathfrak{C}(\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}t')\,\{\!\{\alpha \setminus^{\alpha'} s\}\!\}u) = \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o')$.
- $\square = t\,\mathtt{T}$. Luego, $o = t\,u$ y $o' = t\,u'$ con $u \simeq u'$. Por *h.i.* se tiene $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}u) \simeq \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}u')$. Nuevamente, se apela al Lem. 4.8.1 y se concluye $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o) = \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}t\,\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}u)) \simeq \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}t\,\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}u')) = \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o')$.
- $\square = \lambda x.\mathtt{T}$. Luego, $o = \lambda x.t$ y $o' = \lambda x.t'$ con $t \simeq t'$. Por *h.i.* se tiene $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}t) \simeq \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}t')$ y se concluye $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o) = \lambda x.\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}t) \simeq \lambda x.\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}t') = \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o')$.
- $\square = \mu\gamma.\mathtt{C}$. Luego, $o = \mu\gamma.c$ y $o' = \mu\gamma.c'$ con $c \simeq c'$. Por *h.i.* se tiene $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}c) \simeq \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}c')$ y se concluye $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o) = \mu\gamma.\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}c) \simeq \mu\gamma.\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}c') = \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o')$.
- $\square = \mathtt{T}[x \setminus u]$. Luego, $o = t[x \setminus u]$ y $o' = t'[x \setminus u]$ con $t \simeq t'$. Por *h.i.* se tiene $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}t) \simeq \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}t')$ y se concluye pues $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o) = \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}t)[x \setminus \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}u)] \simeq \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}t')[x \setminus \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}u)] = \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o')$.
- $\square = t[x \setminus \mathtt{T}]$. Luego, $o = t[x \setminus u]$ y $o' = t[x \setminus u']$ con $u \simeq u'$. Por *h.i.* se tiene $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}u) \simeq \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}u')$ y se concluye pues $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o) = \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}t)[x \setminus \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}u)] \simeq \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}t)[x \setminus \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}u')] = \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o')$.
- $\square = [\alpha]\,\mathtt{T}$. Luego, $o = [\alpha]\,t$ y $o' = [\alpha]\,r'$ con $t \simeq t'$. Por *h.i.* se tiene $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}t) \simeq \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}t')$. Se concluye por Lem. 4.8.1, $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o) = \mathfrak{C}([\alpha']\,\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}t) :: s) \simeq \mathfrak{C}([\alpha']\,\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}t') :: s) = \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o')$.
- $\square = [\gamma]\,\mathtt{T}$ con $\gamma \neq \alpha$. Luego, $o = [\gamma]\,t$ y $o' = [\gamma]\,r'$ con $t \simeq t'$. Por *h.i.* se tiene $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}t) \simeq \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}t')$ y se concluye pues $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o) = [\gamma]\,\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}t) \simeq \mu\gamma.\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}t') = \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o')$.
- $\square = \mathtt{C}[\![\gamma \setminus^{\alpha} s']\!]$. Luego, $o = c[\![\gamma \setminus^{\alpha} s']\!]$ y $o' = c'[\![\gamma \setminus^{\alpha} s']\!]$ con $c \simeq c'$. Por *h.i.* se tiene $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}c) \simeq \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}c')$. Más aún, la operación de (meta-)replacement no genera nuevos NPW-redexes con γ. Luego, se tiene $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o) = \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}c)[\![\gamma \setminus^{\alpha'} \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}s') \cdot \mathfrak{C}(s)]\!] \simeq \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}c')[\![\gamma \setminus^{\alpha'} \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}s') \cdot \mathfrak{C}(s)]\!] = \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o')$ por lo que se concluye.
- $\square = \mathtt{C}[\![\gamma \setminus^{\delta} s']\!]$ con $\delta \neq \alpha$. Luego, $o = c[\![\gamma \setminus^{\delta} s']\!]$ y $o' = c'[\![\gamma \setminus^{\delta} s']\!]$ con $c \simeq c'$. Por *h.i.* se tiene $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}c) \simeq \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}c')$. Dado que la operación de (meta-)replacement no genera nuevos NPW-redexes con γ, se concluye $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o) = \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}c)[\![\gamma \setminus^{\delta} \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}s')]\!] \simeq \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}c')[\![\gamma \setminus^{\delta} \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}s')]\!] = \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o')$.
- $\square = c[\![\gamma \setminus^{\alpha} \mathtt{S}]\!]$. Luego, $o = c[\![\gamma \setminus^{\alpha} s']\!]$ y $o' = c[\![\gamma \setminus^{\alpha} s'']\!]$ con $s' \simeq s''$. Por *h.i.* se tiene $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}s') \simeq \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}s'')$. Más aún, la operación de (meta-)replacement no genera nuevos NPW-redexes con γ. Luego, se tiene $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o) = \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}c)[\![\gamma \setminus^{\alpha'} \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}s') \cdot \mathfrak{C}(s)]\!] \simeq \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}c)[\![\gamma \setminus^{\alpha'} \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}s'') \cdot \mathfrak{C}(s)]\!] = \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o')$ por lo que se concluye.

- $\mathtt{O} = c[\![\gamma \setminus^{\delta} \mathtt{S}]\!]$ con $\delta \neq \alpha$. Luego, $o = c[\![\gamma \setminus^{\delta} s']\!]$ y $o' = c[\![\gamma \setminus^{\delta} s'']\!]$ con $s' \simeq s''$. Por *h.i.* se tiene $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} s') \simeq \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} s'')$. Dado que la operación de (meta-)replacement no genera nuevos NPW-redexes con γ, se concluye $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o) = \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} c)[\![\gamma \setminus^{\delta} \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} s'')]\!] \simeq \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} c)[\![\gamma \setminus^{\delta} \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} s'')]\!] = \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o')$.
- $\mathtt{O} = \mathtt{C}[\gamma \setminus \alpha]$. Luego, $o = c[\gamma \setminus \alpha]$ y $o' = c'[\gamma \setminus \alpha]$ con $c \simeq c'$. Por *h.i.* se tiene $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} c) \simeq \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} c')$. Se concluye por Lem. 4.8.1, dado que $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o) = \mathfrak{C}(\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} c)[\![\gamma \setminus^{\gamma'} s]\!][\gamma' \setminus \alpha']) \simeq \mathfrak{C}(\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} c')[\![\gamma \setminus^{\gamma'} s]\!][\gamma' \setminus \alpha']) = \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o')$ con γ' un nombre fresco.
- $\mathtt{O} = \mathtt{C}[\gamma \setminus \delta]$ con $\delta \neq \alpha$. Luego, $o = c[\gamma \setminus \delta]$ y $o' = c'[\gamma \setminus \delta]$ con $c \simeq c'$. Por *h.i.* se tiene $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} c) \simeq \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} c')$ y se concluye $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o) = \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} c)[\gamma \setminus \delta] \simeq \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} c')[\gamma \setminus \delta] = \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o')$.
- $\mathtt{O} = \mathtt{T} \cdot s'$. Luego, $o = t \cdot s'$ y $o' = t' \cdot s'$ con $t \simeq t'$. Por *h.i.* se tiene $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} t) \simeq \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} t')$, por lo que se concluye $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o) = \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} t) \cdot \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} s') \simeq \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} t') \cdot \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} s') = \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o')$.
- $\mathtt{O} = t \cdot \mathtt{S}$. Luego, $o = t \cdot s'$ y $o' = t \cdot s'$ con $s' \simeq s''$. Por *h.i.* se tiene $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} s') \simeq \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} s'')$, por lo que se concluye $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o) = \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} t) \cdot \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} s') \simeq \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} t) \cdot \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} s'') = \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o')$.

2. Por inducción en o.

- $o = x$. Luego, $\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o = x$ y $\{\!\{\alpha \setminus^{\alpha'} s'\}\!\} o = y$. Se concluye por reflexividad de $\simeq$.
- $o = t\,u$. Luego, $\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o = \{\!\{\alpha \setminus^{\alpha'} s\}\!\} t\, \{\!\{\alpha \setminus^{\alpha'} s\}\!\} u$ y $\{\!\{\alpha \setminus^{\alpha'} s'\}\!\} o = \{\!\{\alpha \setminus^{\alpha'} s'\}\!\} t\, \{\!\{\alpha \setminus^{\alpha'} s'\}\!\} u$. Por *h.i.* se tienen $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} t) \simeq \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s'\}\!\} t)$ y $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} u) \simeq \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s'\}\!\} u)$. Finalmente, se concluye por Lem. 4.8.1, pues $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o) = \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} t\, \{\!\{\alpha \setminus^{\alpha'} s\}\!\} u) = \mathfrak{C}(\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} t)\, \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} u)) \simeq \mathfrak{C}(\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s'\}\!\} t)\, \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s'\}\!\} u)) = \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s'\}\!\} o)$.
- $o = \lambda x.t$. Luego, $\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o = \lambda x.\{\!\{\alpha \setminus^{\alpha'} s\}\!\} t$ y $\{\!\{\alpha \setminus^{\alpha'} s'\}\!\} o = \lambda x.\{\!\{\alpha \setminus^{\alpha'} s'\}\!\} t$. Por *h.i.* se tiene $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} t) \simeq \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s'\}\!\} t)$, por lo que se concluye $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o) = \lambda x.\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} t) \simeq \lambda x.\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s'\}\!\} t) = \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s'\}\!\} o)$.
- $o = \mu\gamma.c$. Luego, $\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o = \mu\gamma.\{\!\{\alpha \setminus^{\alpha'} s\}\!\} c$ y $\{\!\{\alpha \setminus^{\alpha'} s'\}\!\} o = \mu\gamma.\{\!\{\alpha \setminus^{\alpha'} s'\}\!\} c$. Por *h.i.* se tiene $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} c) \simeq \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s'\}\!\} c)$ y se concluye $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o) = \mu\gamma.\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} c) \simeq \mu\gamma.\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s'\}\!\} c) = \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s'\}\!\} o)$.
- $o = t[x \setminus u]$. Luego, se tienen $\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o = \{\!\{\alpha \setminus^{\alpha'} s\}\!\} t[x \setminus \{\!\{\alpha \setminus^{\alpha'} s\}\!\} u]$ y $\{\!\{\alpha \setminus^{\alpha'} s'\}\!\} o = \{\!\{\alpha \setminus^{\alpha'} s'\}\!\} t[x \setminus \{\!\{\alpha \setminus^{\alpha'} s'\}\!\} u]$. Por *h.i.* con ambos términos vale $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} t) \simeq \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s'\}\!\} t)$ y $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} u) \simeq \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s'\}\!\} u)$. Finalmente, se concluye dado que $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o) = \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} t)[x \setminus \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} u)] \simeq \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s'\}\!\} t)[x \setminus \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s'\}\!\} u)] = \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s'\}\!\} o)$.
- $o = [\alpha]\, t$. Luego, $\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o = [\alpha']\, (\{\!\{\alpha \setminus^{\alpha'} s\}\!\} t) :: s$ y $\{\!\{\alpha \setminus^{\alpha'} s'\}\!\} o = [\alpha']\, (\{\!\{\alpha \setminus^{\alpha'} s'\}\!\} t) :: s'$. Más aún, por Lem. 4.8.1 con $s \simeq s'$, se tiene $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o) \simeq \mathfrak{C}([\alpha']\, (\{\!\{\alpha \setminus^{\alpha'} s\}\!\} t) :: s')$. Por *h.i.* se tiene $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} t) \simeq \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s'\}\!\} t)$. Se concluye por Lem. 4.8.1 nuevamente, dado que $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o) \simeq \mathfrak{C}([\alpha']\, (\{\!\{\alpha \setminus^{\alpha'} s\}\!\} t) :: s') \simeq \mathfrak{C}([\alpha']\, (\{\!\{\alpha \setminus^{\alpha'} s'\}\!\} t) :: s') = \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s'\}\!\} o)$.
- $o = [\gamma]\, t$ con $\gamma \neq \alpha$. Luego, $\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o = [\alpha]\, \{\!\{\alpha \setminus^{\alpha'} s\}\!\} t$ y $\{\!\{\alpha \setminus^{\alpha'} s'\}\!\} o = [\alpha]\, \{\!\{\alpha \setminus^{\alpha'} s'\}\!\} t$. Por *h.i.* se tiene $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} t) \simeq \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s'\}\!\} t)$, por lo que se concluye $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o) = [\alpha]\, \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} t) \simeq [\alpha]\, \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s'\}\!\} t) = \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s'\}\!\} o)$.
- $o = c[\![\gamma \setminus^{\alpha} s'']\!]$. Luego, $\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o = (\{\!\{\alpha \setminus^{\alpha'} s\}\!\} c)[\![\gamma \setminus^{\alpha} \{\!\{\alpha \setminus^{\alpha'} s\}\!\} s'' \cdot s]\!]$ y $\{\!\{\alpha \setminus^{\alpha'} s'\}\!\} o = (\{\!\{\alpha \setminus^{\alpha'} s'\}\!\} c)[\![\gamma \setminus^{\alpha} \{\!\{\alpha \setminus^{\alpha'} s'\}\!\} s'' \cdot s']\!]$. Por *h.i.* se tienen $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} c) \simeq \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s'\}\!\} c)$ y $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} s'') \simeq \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s'\}\!\} s'')$. Más aún, la operación de (meta-)replacement no genera nuevos NPW-redexes con γ. Luego, $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o) = \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} c)[\![\gamma \setminus^{\alpha} \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} s'') \cdot s]\!] \simeq \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s'\}\!\} c)[\![\gamma \setminus^{\alpha} \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s'\}\!\} s'') \cdot s']\!] = \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\} o')$ por lo que se concluye.

- $o = c[\![\gamma \setminus^{\delta} s'']\!]$ con $\delta \neq \alpha$. Luego, $\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o = (\{\!\{\alpha \setminus^{\alpha'} s\}\!\}c)[\![\gamma \setminus^{\delta} \{\!\{\alpha \setminus^{\alpha'} s\}\!\}s'']\!]$ y $\{\!\{\alpha \setminus^{\alpha'} s'\}\!\}o = (\{\!\{\alpha \setminus^{\alpha'} s'\}\!\}c)[\![\gamma \setminus^{\delta} \{\!\{\alpha \setminus^{\alpha'} s'\}\!\}s'']\!]$. Por *h.i.* con ambos objectos se tiene tanto $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}c) \simeq \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s'\}\!\}c)$ como $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}s'') \simeq \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s'\}\!\}s'')$. Dado que la operación de (meta-)replacement no genera nuevos PW-redexes, se concluye $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o) = \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}c)[\![\gamma \setminus^{\delta} \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}s'')]\!] \simeq \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s'\}\!\}c)[\![\gamma \setminus^{\delta} \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s'\}\!\}s'')]\!] = \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o')$.
- $o = c[\gamma \setminus \alpha]$. Luego, se tienen $\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o = (\{\!\{\alpha \setminus^{\alpha'} s\}\!\}c)[\![\gamma \setminus^{\gamma'} s]\!][\gamma' \setminus \alpha']$ y $\{\!\{\alpha \setminus^{\alpha'} s'\}\!\}o = (\{\!\{\alpha \setminus^{\alpha'} s'\}\!\}c)[\![\gamma \setminus^{\gamma'} s']\!][\gamma' \setminus \alpha']$ con γ' un nombre fresco. Más aún, por Lem. 4.8.1 con $s \simeq s'$, se tiene $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o) \simeq \mathfrak{C}((\{\!\{\alpha \setminus^{\alpha'} s\}\!\}c)[\![\gamma \setminus^{\gamma'} s']\!][\gamma' \setminus \alpha'])$. Por *h.i.* se tiene $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}c) \simeq \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s'\}\!\}c)$. Finalmente, se concluye por Lem. 4.8.1 nuevamente, pues $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o) \simeq \mathfrak{C}((\{\!\{\alpha \setminus^{\alpha'} s\}\!\}c)[\![\gamma \setminus^{\gamma'} s']\!][\gamma' \setminus \alpha']) \simeq \mathfrak{C}((\{\!\{\alpha \setminus^{\alpha'} s'\}\!\}c)[\![\gamma \setminus^{\gamma'} s']\!][\gamma' \setminus \alpha']) = \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s'\}\!\}o)$.
- $o = c[\gamma \setminus \delta]$ con $\delta \neq \alpha$. Luego, se tienen $\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o = (\{\!\{\alpha \setminus^{\alpha'} s\}\!\}c)[\gamma \setminus \delta]$ y $\{\!\{\alpha \setminus^{\alpha'} s'\}\!\}o = (\{\!\{\alpha \setminus^{\alpha'} s'\}\!\}c)[\gamma \setminus \delta]$. Por *h.i.* se tiene $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}c) \simeq \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s'\}\!\}c)$. Más aún, la operación de (meta-)replacement no genera nuevos NPW-redexes con γ. Luego, $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o) = \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}c)[\gamma \setminus \delta] \simeq \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s'\}\!\}c)[\gamma \setminus \delta] = \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o')$ por lo que se concluye.
- $o = t \cdot s$. Luego, $\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o = \{\!\{\alpha \setminus^{\alpha'} s\}\!\}t \cdot \{\!\{\alpha \setminus^{\alpha'} s\}\!\}s$ y $\{\!\{\alpha \setminus^{\alpha'} s'\}\!\}o = \{\!\{\alpha \setminus^{\alpha'} s'\}\!\}t \cdot \{\!\{\alpha \setminus^{\alpha'} s'\}\!\}s$. Por *h.i.* se tienen $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}t) \simeq \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s'\}\!\}t)$ y $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}s) \simeq \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s'\}\!\}s)$. Finalmente, se concluye pues $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o) = \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}t) \cdot \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}s) \simeq \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s'\}\!\}t) \cdot \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s'\}\!\}s) = \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s'\}\!\}o)$.

□

Lema 4.8.2. *Sean $u, s, o \in \mathbb{O}_{\Lambda M}$ en $\mathfrak{C}$-forma normal tales que $o \simeq o'$, $u \simeq u'$ y $s \simeq s'$ con u, u' términos y s, s' stacks. Luego,*

1. $\mathfrak{C}(\{x \setminus u\}o) \simeq \mathfrak{C}(\{x \setminus u\}o')$ *y* $\mathfrak{C}(\{x \setminus u\}o) \simeq \mathfrak{C}(\{x \setminus u'\}o)$.
2. $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o) \simeq \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o')$ *y* $\mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}o) \simeq \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s'\}\!\}o)$.

Demostración. Por Lem. C.0.53 y C.0.54 respectivamente. □

Teorema 4.8.3. *Sea $o \in \mathbb{O}_{\Lambda M}$. Si $o \simeq p$ y $o \rightsquigarrow o'$, luego existe p' tal que $p \rightsquigarrow p'$ y $o' \simeq p'$.*

Demostración. La prueba es por inducción en $o \simeq p$. Por definición esto implica verificar cuatro condiciones: reflexividad, transitividad, simetría y congruencia (*i.e.* clausura por contextos):

Reflexividad Luego, $p = o$ por lo que basta tomar $p' = o'$ para concluir por reflexividad de $\simeq$.

Transitividad Luego, existe $q \in \mathbb{O}_{\Lambda M}$ tal que $o \simeq q$ y $q \simeq p$. Por *h.i.* con $o \simeq q$ y $o \rightsquigarrow o'$, existe $q' \in \mathbb{O}_{\Lambda M}$ tal que $q \rightsquigarrow q'$ y $o' \simeq q'$. Por *h.i.* nuevamente, ahora sobre $q \simeq p$ y $q \rightsquigarrow q'$, existe $p' \in \mathbb{O}_{\Lambda M}$ tal que $p \rightsquigarrow p'$ y $q' \simeq p'$. Se concluye con $o' \simeq p'$ por transitividad de $\simeq$.

Simetría/Congruencia La simetría y congruencia se demuestran simultáneamente por inducción en el contexto de clausura $\mathtt{Q}$ tal que $o = \mathtt{Q}\langle q \rangle$ y $p = \mathtt{Q}\langle q' \rangle$ con $q \simeq_* q'$, donde $\simeq_*$ es una regla de la Fig. 4.13. Más aún, $o \rightsquigarrow o'$ implica $o = \mathtt{O}\langle l \rangle$ y $o' = \mathtt{O}\langle r \rangle$ con $l \mapsto_* r$, $* \in \{\mathtt{S}, \mathtt{R}^{\bullet}\}$. Se consideran todas las posibles formas de $\mathtt{Q}$ y $\mathtt{O}$:

- $\mathtt{Q} = \Box$. Luego, $o = q \simeq_* q' = p$. Más aún, es el caso de $o, p \in \mathbb{T}_{\Lambda M}$. Se tienen solo dos reglas aplicables a términos:
  - $\simeq_{\mathtt{exsubs}}$. Luego, $o = \mathtt{LTT}\langle t \rangle[x \setminus u]$ y $p = \mathtt{LTT}\langle t[x \setminus u] \rangle$ con $x \notin \mathtt{LTT}$ y $\mathsf{fc}(u, \mathtt{LTT})$. Hay tres casos posibles:

1. S-redex en la raíz.

$$\begin{array}{ccc} o = \mathtt{LTT}\langle t\rangle[x \setminus u] & \simeq_{\mathtt{exsubs}} & \mathtt{LTT}\langle t[x \setminus u]\rangle = p \\ \mathtt{s}\downarrow & & \mathtt{s}\downarrow \\ o' = \mathfrak{C}(\{x \setminus u\}\mathtt{LTT}\langle t\rangle) & = & \mathfrak{C}(\mathtt{LTT}\langle\{x \setminus u\}t\rangle) = p' \end{array}$$

2. S-redex superpuesto con LTT. Luego, se tiene $\mathtt{LTT} = \mathtt{LTT}_1\langle\mathtt{LTT}_2[y \setminus v]\rangle$ con $o' = \mathtt{LTT}_1\langle\{y \setminus v\}\mathtt{LTT}_2\langle t\rangle\rangle[x \setminus u] = \mathtt{LTT}_1\langle\mathtt{LTT}'_2\langle t'\rangle\rangle[x \setminus u]$ y $p' = \mathtt{LTT}_1\langle\mathtt{LTT}'_2\langle t'[x \setminus u]\rangle\rangle$ dado que $\mathsf{fc}(u, \mathtt{LTT})$ implica $y \notin u$. Se concluye por Lem. 4.6.2 (1):

$$\begin{array}{ccc} o = \mathtt{LTT}_1\langle\mathtt{LTT}_2\langle t\rangle[y \setminus v]\rangle[x \setminus u] & \simeq_{\mathtt{exsubs}} & \mathtt{LTT}_1\langle\mathtt{LTT}_2\langle t[x \setminus u]\rangle[y \setminus v]\rangle = p \\ \mathtt{s}\downarrow & & \mathtt{s}\downarrow \\ o' = \mathfrak{C}(\mathtt{LTT}_1\langle\{y \setminus v\}\mathtt{LTT}_2\langle t\rangle\rangle[x \setminus u]) & \simeq & \mathfrak{C}(\mathtt{LTT}_1\langle\{y \setminus v\}\mathtt{LTT}_2\langle t[x \setminus u]\rangle\rangle) = p' \end{array}$$

3. $\mathtt{R}^\bullet$-redex superpuesto con LTT. Luego, se tiene $\mathtt{LTT} = \mathtt{LTC}\langle\mathtt{LCT}[\![\alpha \setminus^{\alpha'} s]\!]\rangle$ con $o' = \mathtt{LTC}\langle\{\!\{\alpha \setminus^{\alpha'} s\}\!\}\mathtt{LCT}\langle t\rangle\rangle[x \setminus u] = \mathtt{LTC}\langle\mathtt{LTC}'\langle t'\rangle\rangle[x \setminus u]$ y $p' = \mathtt{LTC}\langle\mathtt{LTC}'\langle t'[x \setminus u]\rangle\rangle$ dado que $\mathsf{fc}(u, \mathtt{LTT})$ implica $\alpha \notin u$. Se concluye por Lem. 4.6.2 (1):

$$\begin{array}{ccc} o = \mathtt{LTC}\langle\mathtt{LCT}\langle t\rangle[\![\alpha \setminus^{\alpha'} s]\!]\rangle[x \setminus u] & \simeq_{\mathtt{exsubs}} & \mathtt{LTC}\langle\mathtt{LCT}\langle t[x \setminus u]\rangle[\![\alpha \setminus^{\alpha'} s]\!]\rangle = p \\ \mathtt{R}^\bullet\downarrow & & \mathtt{R}^\bullet\downarrow \\ o' = \mathfrak{C}(\mathtt{LTC}\langle\{\!\{\alpha \setminus^{\alpha'} s\}\!\}\mathtt{LCT}\langle t\rangle\rangle[x \setminus u]) & \simeq & \mathfrak{C}(\mathtt{LTC}\langle\{\!\{\alpha \setminus^{\alpha'} s\}\!\}\mathtt{LCT}\langle t[x \setminus u]\rangle\rangle) = p' \end{array}$$

- $\simeq_\theta$. Luego, $o = \mu\alpha.[\alpha]\,t$ y $p = t$ con $\alpha \notin t$. Este caso es inmediato dado que todas las posible reducciones son internas.

- $\mathtt{Q} = \boxdot$. Luego, $o = q \simeq_* q' = p$. Más aún, es el caso de $o, p \in \mathbb{C}_{\Lambda M}$. Se tienen cinco reglas aplicables a comandos:

  - $\simeq_{\mathtt{exrepl}}$. Luego, $o = \mathtt{LCC}\langle c\rangle[\![\alpha \setminus^{\alpha'} s]\!]$ y $p = \mathtt{LCC}\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle$ con $\alpha \notin \mathtt{LCC}$, $\mathsf{fc}(\alpha', \mathtt{LCC})$ y $\mathsf{fc}(s, \mathtt{LCC})$. Hay tres casos posibles:

    1. $\mathtt{R}^\bullet$-redex en la raíz.

$$\begin{array}{ccc} o = \mathtt{LCC}\langle c\rangle[\![\alpha \setminus^{\alpha'} s]\!] & \simeq_{\mathtt{exrepl}} & \mathtt{LCC}\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle = p \\ \mathtt{R}^\bullet\downarrow & & \mathtt{R}^\bullet\downarrow \\ o' = \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}\mathtt{LCC}\langle c\rangle) & = & \mathfrak{C}(\mathtt{LCC}\langle\{\!\{\alpha \setminus^{\alpha'} s\}\!\}c\rangle) = p' \end{array}$$

    2. S-redex superpuesto con LCC. Luego, se tiene $\mathtt{LCC} = \mathtt{LCT}\langle\mathtt{LTC}[x \setminus u]\rangle$ con $o' = \mathtt{LCT}\langle\{x \setminus u\}\mathtt{LTC}\langle c\rangle\rangle[\![\alpha \setminus^{\alpha'} s]\!] = \mathtt{LCT}\langle\mathtt{LTC}'\langle c'\rangle\rangle[\![\alpha \setminus^{\alpha'} s]\!]$ y $p' = \mathtt{LCT}\langle\mathtt{LTC}'\langle c'[\![\alpha \setminus^{\alpha'} s]\!]\rangle\rangle$ dado que $\mathsf{fc}(s, \mathtt{LCC})$ implica $x \notin s$. Se concluye por Lem. 4.6.2 (2):

$$\begin{array}{ccc} o = \mathtt{LCT}\langle\mathtt{LTC}\langle c\rangle[x \setminus u]\rangle[\![\alpha \setminus^{\alpha'} s]\!] & \simeq_{\mathtt{exrepl}} & \mathtt{LCT}\langle\mathtt{LTC}\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle[x \setminus u]\rangle = p \\ \mathtt{s}\downarrow & & \mathtt{s}\downarrow \\ o' = \mathfrak{C}(\mathtt{LCT}\langle\{x \setminus u\}\mathtt{LTC}\langle c\rangle\rangle[\![\alpha \setminus^{\alpha'} s]\!]) & \simeq & \mathfrak{C}(\mathtt{LCT}\langle\{x \setminus u\}\mathtt{LTC}\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle\rangle) = p' \end{array}$$

    3. $\mathtt{R}^\bullet$-redex superpuesto con LCC. Luego, se tiene $\mathtt{LCC} = \mathtt{LCC}_1\langle\mathtt{LCC}_2[\![\gamma \setminus^{\delta} s']\!]\rangle$ con $o' = \mathtt{LCC}_1\langle\{\!\{\gamma \setminus^{\delta} s'\}\!\}\mathtt{LCC}_2\langle c\rangle\rangle[\![\alpha\setminus^{\alpha'} s]\!] = \mathtt{LCC}_1\langle\mathtt{LCC}'_2\langle c'\rangle\rangle[\![\alpha\setminus^{\alpha'} s]\!]$ y $p' = \mathtt{LCC}_1\langle\mathtt{LCC}'_2\langle c'[\![\alpha \setminus^{\alpha'} s]\!]\rangle\rangle$

dado que $\mathsf{fc}(\alpha, \mathtt{LCC})$ implica $\gamma \neq \alpha$, y $\mathsf{fc}(s, \mathtt{LCC})$ implica $\alpha \notin s$. Se concluye por Lem. 4.6.2 (2):

$$\begin{array}{ccc} o = \mathtt{LCC}_1\langle \mathtt{LCC}_2\langle c\rangle [\![\gamma \setminus^{\delta} s']\!]\rangle [\![\alpha \setminus^{\alpha'} s]\!] & \simeq_{\mathtt{exrepl}} & \mathtt{LCC}_1\langle \mathtt{LCC}_2\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle [\![\gamma \setminus^{\delta} s']\!]\rangle = p \\ \downarrow{\scriptstyle \mathtt{R}^\bullet} & & \downarrow{\scriptstyle \mathtt{R}^\bullet} \\ o' = \mathfrak{C}(\mathtt{LCC}_1\langle \{\!\{\gamma \setminus^{\delta} s'\}\!\} \mathtt{LCC}_2\langle c\rangle\rangle [\![\alpha \setminus^{\alpha'} s]\!]) & \simeq & \mathfrak{C}(\mathtt{LCC}_1\langle \{\!\{\gamma \setminus^{\delta} s'\}\!\} \mathtt{LCC}_2\langle c[\![\alpha \setminus^{\alpha'} s]\!]\rangle\rangle) = p' \end{array}$$

- $\simeq_{\mathtt{exren}}$. Luego, $o = \mathtt{LCC}\langle c\rangle[\alpha \setminus \beta]$ y $p = \mathtt{LCC}\langle c[\alpha \setminus \beta]\rangle$ con $\alpha \notin \mathtt{LCC}$ y $\mathsf{fc}(\beta, \mathtt{LCC})$. Hay dos casos posibles:
  1. S-redex superpuesto con $\mathtt{LCC}$. Luego, se tiene $\mathtt{LCC} = \mathtt{LCT}\langle \mathtt{LTC}[x \setminus u]\rangle$ con $o' = \mathtt{LCT}\langle \{x \setminus u\}\mathtt{LTC}\langle c\rangle\rangle[\alpha \setminus \beta] = \mathtt{LCT}\langle \mathtt{LTC}'\langle c'\rangle\rangle[\alpha \setminus \beta]$ y $p' = \mathtt{LCT}\langle \mathtt{LTC}'\langle c'[\alpha \setminus \beta]\rangle\rangle$. Se concluye por Lem. 4.6.2 (2):

$$\begin{array}{ccc} o = \mathtt{LCT}\langle \mathtt{LTC}\langle c\rangle[x \setminus u]\rangle[\alpha \setminus \beta] & \simeq_{\mathtt{exrepl}} & \mathtt{LCT}\langle \mathtt{LTC}\langle c[\alpha \setminus \beta]\rangle[x \setminus u]\rangle = p \\ \downarrow{\scriptstyle \mathtt{S}} & & \downarrow{\scriptstyle \mathtt{S}} \\ o' = \mathfrak{C}(\mathtt{LCT}\langle \{x \setminus u\}\mathtt{LTC}\langle c\rangle\rangle[\alpha \setminus \beta]) & \simeq & \mathfrak{C}(\mathtt{LCT}\langle \{x \setminus u\}\mathtt{LTC}\langle c[\alpha \setminus \beta]\rangle\rangle) = p' \end{array}$$

  2. $\mathtt{R}^\bullet$-redex superpuesto con $\mathtt{LCC}$. Luego, se tiene $\mathtt{LCC} = \mathtt{LCC}_1\langle \mathtt{LCC}_2[\![\gamma \setminus^{\delta} s]\!]\rangle$ con $o' = \mathtt{LCC}_1\langle \{\!\{\gamma \setminus^{\delta} s\}\!\} \mathtt{LCC}_2\langle c\rangle\rangle[\alpha \setminus \beta] = \mathtt{LCC}_1\langle \mathtt{LCC}_2'\langle c'\rangle\rangle[\alpha \setminus \beta]$ y $p' = \mathtt{LCC}_1\langle \mathtt{LCC}_2'\langle c'[\alpha \setminus \beta]\rangle\rangle$ dado que $\mathsf{fc}(\alpha, \mathtt{LCC})$ implica $\gamma \neq \alpha$. Se concluye por Lem. 4.6.2 (2):

$$\begin{array}{ccc} o = \mathtt{LCC}_1\langle \mathtt{LCC}_2\langle c\rangle [\![\gamma \setminus^{\delta} s]\!]\rangle[\alpha \setminus \beta] & \simeq_{\mathtt{exrepl}} & \mathtt{LCC}_1\langle \mathtt{LCC}_2\langle c[\alpha \setminus \beta]\rangle [\![\gamma \setminus^{\delta} s]\!]\rangle = p \\ \downarrow{\scriptstyle \mathtt{R}^\bullet} & & \downarrow{\scriptstyle \mathtt{R}^\bullet} \\ o' = \mathfrak{C}(\mathtt{LCC}_1\langle \{\!\{\gamma \setminus^{\delta} s\}\!\} \mathtt{LCC}_2\langle c\rangle\rangle[\alpha \setminus \beta]) & \simeq & \mathfrak{C}(\mathtt{LCC}_1\langle \{\!\{\gamma \setminus^{\delta} s\}\!\} \mathtt{LCC}_2\langle c[\alpha \setminus \beta]\rangle\rangle) = p' \end{array}$$

- $\simeq_{\mathtt{pp}}$. Luego, $o = [\alpha']\,\lambda z.\mu\alpha.[\beta']\,\lambda y.\mu\beta.c$ y $p = [\beta']\,\lambda y.\mu\beta.[\alpha']\,\lambda z.\mu\alpha.c$ con $\beta \neq \alpha'$ y $\alpha \neq \beta'$. Este caso es inmediato dado que todas las posible reducciones son internas.
- $\simeq_{\rho}$. Luego, $o = [\beta]\,\mu\alpha.c$ y $p = c[\alpha \setminus \beta]$.Este caso es inmediato dado que todas las posible reducciones son internas.

- $\mathtt{Q} = \mathtt{T}\,u$. Luego, $o = \mathtt{T}\langle q\rangle\,u$ y $p = \mathtt{T}\langle q'\rangle\,u$. Hay dos casos de reducción posibles:
  1. $\mathtt{T}\langle q\rangle \rightsquigarrow t$. Luego, $o \rightsquigarrow \mathfrak{C}(t\,u) = o'$. Por *h.i.* existe t' tal que $\mathtt{T}\langle q'\rangle \rightsquigarrow t'$ y $t \simeq t'$. Más aún, se tiene $p \rightsquigarrow \mathfrak{C}(t'\,u) = p'$. Por Lem. 4.8.1 con $t \simeq t'$ y el contexto $\Box\,u$ se tiene $o = \mathfrak{C}(t\,u) \simeq \mathfrak{C}(t'\,u) = p'$, por lo que se concluye.
  2. $u \rightsquigarrow u'$. Luego, $o \rightsquigarrow \mathfrak{C}(\mathtt{T}\langle q\rangle\,u') = o'$ y $p \rightsquigarrow \mathfrak{C}(\mathtt{T}\langle q'\rangle\,u') = p'$. Se concluye por Lem. 4.8.1 con $\mathtt{T}\langle q\rangle \simeq \mathtt{T}\langle q'\rangle$ y el contexto $\Box\,u'$, $o' \simeq p'$.
- $\mathtt{Q} = t\,\mathtt{T}$. Luego, $o = t\,\mathtt{T}\langle q\rangle$ y $p = t\,\mathtt{T}\langle q'\rangle$. Hay dos casos de reducción posibles:
  1. $\mathtt{T}\langle q\rangle \rightsquigarrow u$. Luego, $o \rightsquigarrow \mathfrak{C}(t\,u) = o'$. Por *h.i.* existe u' tal que $\mathtt{T}\langle q'\rangle \rightsquigarrow u'$ y $u \simeq u'$. Más aún, se tiene $p \rightsquigarrow \mathfrak{C}(t\,u') = p'$. Por Lem. 4.8.1 con $u \simeq u'$ y el contexto $t\,\Box$ se tiene $o = \mathfrak{C}(t\,u) \simeq \mathfrak{C}(t\,u') = p'$, por lo que se concluye.
  2. $t \rightsquigarrow t'$. Luego, $o \rightsquigarrow \mathfrak{C}(t'\,\mathtt{T}\langle q\rangle) = o'$ y $p \rightsquigarrow \mathfrak{C}(t'\,\mathtt{T}\langle q'\rangle) = p'$. Se concluye por Lem. 4.8.1 con $\mathtt{T}\langle q\rangle \simeq \mathtt{T}\langle q'\rangle$ y el contexto $t'\,\Box$, $o' \simeq p'$.
- $\mathtt{Q} = \lambda x.\mathtt{T}$. Luego, $o = \lambda x.\mathtt{T}\langle q\rangle$ y $p = \lambda x.\mathtt{T}\langle q'\rangle$ con $\mathtt{T}\langle q\rangle \rightsquigarrow t$. Luego, $o \rightsquigarrow \lambda x.t = o'$. Por *h.i.* existe t' tal que $\mathtt{T}\langle q'\rangle \rightsquigarrow t'$ y $t \simeq t'$. Más aún, se tiene $p \rightsquigarrow \lambda x.t' = p'$. Se concluye dado que $o' = \lambda x.t \simeq \lambda x.t' = p'$.

- $\mathtt{Q} = \mu\alpha.\mathtt{C}$. Luego, $o = \mu\alpha.\mathtt{C}\langle q\rangle$ y $p = \mu\alpha.\mathtt{C}\langle q'\rangle$ con $\mathtt{C}\langle q\rangle \rightsquigarrow c$. Luego, $o \rightsquigarrow \mu\alpha.c = o'$. Por *h.i.* existe c' tal que $\mathtt{C}\langle q'\rangle \rightsquigarrow c'$ y $c \simeq c'$. Más aún, se tiene $p \rightsquigarrow \mu\alpha.c' = c'$. Se concluye dado que $o' = \mu\alpha.c \simeq \mu\alpha.c' = p'$.
- $\mathtt{Q} = \mathtt{T}[x \setminus u]$. Luego, $o = \mathtt{T}\langle q\rangle[x \setminus u]$ y $p = \mathtt{T}\langle q'\rangle[x \setminus u]$. Hay tres casos de reducción posibles:
  1. $o \rightsquigarrow_{\mathtt{S}} \mathfrak{C}(\{x \setminus u\}\mathtt{T}\langle q\rangle)$. Luego, se tiene $p \rightsquigarrow_{\mathtt{S}} \mathfrak{C}(\{x \setminus u\}\mathtt{T}\langle q'\rangle)$. Por Lem. C.0.53 (1) con $\mathtt{T}\langle q\rangle \simeq \mathtt{T}\langle q'\rangle$ se tiene $o' = \mathfrak{C}(\{x \setminus u\}\mathtt{T}\langle q\rangle) \simeq \mathfrak{C}(\{x \setminus u\}\mathtt{T}\langle q'\rangle) = p'$ por lo que se concluye.
  2. $\mathtt{T}\langle q\rangle \rightsquigarrow t$. Luego, $o \rightsquigarrow \mathfrak{C}(t[x \setminus u]) = o'$. Por *h.i.* existe t' tal que $\mathtt{T}\langle q'\rangle \rightsquigarrow t'$ y $t \simeq t'$. Más aún, se tiene $p \rightsquigarrow \mathfrak{C}(t'[x \setminus u]) = p'$. Por Lem. 4.8.1 con $t \simeq t'$ y el contexto $\Box[x \setminus u]$ se tiene $o = \mathfrak{C}(t[x \setminus u]) \simeq \mathfrak{C}(t'[x \setminus u]) = p'$, por lo que se concluye.
  3. $u \rightsquigarrow u'$. Luego, $o \rightsquigarrow \mathfrak{C}(\mathtt{T}\langle q\rangle[x \setminus u']) = o'$ y $p \rightsquigarrow \mathfrak{C}(\mathtt{T}\langle q'\rangle[x \setminus u']) = p'$. Se concluye por Lem. 4.8.1 con $\mathtt{T}\langle q\rangle \simeq \mathtt{T}\langle q'\rangle$ y el contexto $\Box[x \setminus u']$, $o' \simeq p'$.
- $\mathtt{Q} = t[x \setminus \mathtt{T}]$. Luego, $o = t[x \setminus \mathtt{T}\langle q\rangle]$ y $p = t[x \setminus \mathtt{T}\langle q'\rangle]$. Hay tres casos de reducción posibles:
  1. $o \rightsquigarrow_{\mathtt{S}} \mathfrak{C}(\{x \setminus \mathtt{T}\langle q\rangle\}t)$. Luego, se tiene $p \rightsquigarrow_{\mathtt{S}} \mathfrak{C}(\{x \setminus \mathtt{T}\langle q'\rangle\}t)$. Por Lem. C.0.53 (2) con $\mathtt{T}\langle q\rangle \simeq \mathtt{T}\langle q'\rangle$ se tiene $o' = \mathfrak{C}(\{x \setminus \mathtt{T}\langle q\rangle\}t) \simeq \mathfrak{C}(\{x \setminus \mathtt{T}\langle q'\rangle\}t) = p'$ por lo que se concluye.
  2. $\mathtt{T}\langle q\rangle \rightsquigarrow u$. Luego, $o \rightsquigarrow \mathfrak{C}(t[x \setminus u]) = o'$. Por *h.i.* existe u' tal que $\mathtt{T}\langle q'\rangle \rightsquigarrow u'$ y $u \simeq u'$. Más aún, se tiene $p \rightsquigarrow \mathfrak{C}(t[x \setminus u']) = p'$. Por Lem. 4.8.1 con $u \simeq u'$ y el contexto $t[x \setminus \Box]$ se tiene $o = \mathfrak{C}(t[x \setminus u]) \simeq \mathfrak{C}(t[x \setminus u']) = p'$, por lo que se concluye.
  3. $t \rightsquigarrow t'$. Luego, $o \rightsquigarrow \mathfrak{C}(\mathtt{T}\langle q\rangle[x \setminus t']) = o'$ y $p \rightsquigarrow \mathfrak{C}(\mathtt{T}\langle q'\rangle[x \setminus t']) = p'$. Se concluye por Lem. 4.8.1 con $\mathtt{T}\langle q\rangle \simeq \mathtt{T}\langle q'\rangle$ y el contexto $t'[x \setminus \Box]$, $o' \simeq p'$.
- $\mathtt{Q} = [\alpha]\,\mathtt{T}$. Luego, $o = [\alpha]\,\mathtt{T}\langle q\rangle$ y $p = [\alpha]\,\mathtt{T}\langle q'\rangle$ con $\mathtt{T}\langle q\rangle \rightsquigarrow t$. Luego, $o \rightsquigarrow [\alpha]\,t = o'$. Por *h.i.* existe t' tal que $\mathtt{T}\langle q'\rangle \rightsquigarrow t'$ y $t \simeq t'$. Más aún, se tiene $p \rightsquigarrow [\alpha]\,t' = p'$. Se concluye dado que $o' = [\alpha]\,t \simeq [\alpha]\,t' = p'$.
- $\mathtt{Q} = \mathtt{C}[\![\alpha \setminus^{\alpha'} s]\!]$. Luego, $o = \mathtt{C}\langle q\rangle[\![\alpha \setminus^{\alpha'} s]\!]$ y $p = \mathtt{C}\langle q'\rangle[\![\alpha \setminus^{\alpha'} s]\!]$. Hay tres casos de reducción posibles:
  1. $o \rightsquigarrow_{\mathtt{R}\bullet} \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}\mathtt{C}\langle q\rangle)$. Luego, se tiene $p \rightsquigarrow_{\mathtt{R}\bullet} \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}\mathtt{C}\langle q'\rangle)$. Por Lem. C.0.54 (1) con $\mathtt{C}\langle q\rangle \simeq \mathtt{C}\langle q'\rangle$ se tiene $o' = \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}\mathtt{C}\langle q\rangle) \simeq \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} s\}\!\}\mathtt{C}\langle q'\rangle) = p'$ por lo que se concluye.
  2. $\mathtt{C}\langle q\rangle \rightsquigarrow c$. Luego, $o \rightsquigarrow \mathfrak{C}(c[\![\alpha \setminus^{\alpha'} s]\!]) = o'$. Por *h.i.* existe c' tal que $\mathtt{C}\langle q'\rangle \rightsquigarrow c'$ y $c \simeq c'$. Más aún, se tiene $p \rightsquigarrow \mathfrak{C}(c'[\![\alpha \setminus^{\alpha'} s]\!]) = p'$. Por Lem. 4.8.1 con $c \simeq c'$ y el contexto $\boxdot[\![\alpha \setminus^{\alpha'} s]\!]$ se tiene $o = \mathfrak{C}(c[\![\alpha \setminus^{\alpha'} s]\!]) \simeq \mathfrak{C}(c'[\![\alpha \setminus^{\alpha'} s]\!]) = p'$, por lo que se concluye.
  3. $s \rightsquigarrow s'$. Luego, $o \rightsquigarrow \mathfrak{C}(\mathtt{C}\langle q\rangle[\![\alpha \setminus^{\alpha'} s']\!]) = o'$ y $p \rightsquigarrow \mathfrak{C}(\mathtt{C}\langle q'\rangle[\![\alpha \setminus^{\alpha'} s']\!]) = p'$. Se concluye por Lem. 4.8.1 con $\mathtt{C}\langle q\rangle \simeq \mathtt{C}\langle q'\rangle$ y el contexto $\boxdot[\![\alpha \setminus^{\alpha'} s']\!]$, $o' \simeq p'$.
- $\mathtt{Q} = s[\![\alpha \setminus^{\alpha'} \mathtt{S}]\!]$. Luego, $o = c[\![\alpha \setminus^{\alpha'} \mathtt{C}\langle q\rangle]\!]$ y $p = c[\![\alpha \setminus^{\alpha'} \mathtt{C}\langle q'\rangle]\!]$. Hay tres casos de reducción posibles:
  1. $o \rightsquigarrow_{\mathtt{R}\bullet} \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} \mathtt{C}\langle q\rangle\}\!\}c)$. Luego, se tiene $p \rightsquigarrow_{\mathtt{R}\bullet} \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} \mathtt{C}\langle q'\rangle\}\!\}c)$. Por Lem. C.0.54 (2) con $\mathtt{C}\langle q\rangle \simeq \mathtt{C}\langle q'\rangle$ se tiene $o' = \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} \mathtt{C}\langle q\rangle\}\!\}c) \simeq \mathfrak{C}(\{\!\{\alpha \setminus^{\alpha'} \mathtt{C}\langle q'\rangle\}\!\}c) = p'$ por lo que se concluye.
  2. $\mathtt{C}\langle q\rangle \rightsquigarrow s$. Luego, $o \rightsquigarrow \mathfrak{C}(c[\![\alpha \setminus^{\alpha'} s]\!]) = o'$. Por *h.i.* existe s' tal que $\mathtt{C}\langle q'\rangle \rightsquigarrow s'$ y $s \simeq s'$. Más aún, se tiene $p \rightsquigarrow \mathfrak{C}(c'[\![\alpha \setminus^{\alpha'} s]\!]) = p'$. Por Lem. 4.8.1 con $s \simeq s'$ y el contexto $c[\![\alpha \setminus^{\alpha'} \Box]\!]$ se tiene $o = \mathfrak{C}(c[\![\alpha \setminus^{\alpha'} s]\!]) \simeq \mathfrak{C}(c'[\![\alpha \setminus^{\alpha'} s]\!]) = p'$, por lo que se concluye.
  3. $c \rightsquigarrow c'$. Luego, $o \rightsquigarrow \mathfrak{C}(c'[\![\alpha \setminus^{\alpha'} \mathtt{C}\langle q\rangle]\!]) = o'$ y $p \rightsquigarrow \mathfrak{C}(c'[\![\alpha \setminus^{\alpha'} \mathtt{C}\langle q'\rangle]\!]) = p'$. Se concluye por Lem. 4.8.1 con $\mathtt{C}\langle q\rangle \simeq \mathtt{C}\langle q'\rangle$ y el contexto $c'[\![\alpha \setminus^{\alpha'} \Box]\!]$, $o' \simeq p'$.
- $\mathtt{Q} = \mathtt{C}[\alpha \setminus \beta]$. Luego, $o = \mathtt{C}\langle q\rangle[\alpha \setminus \beta]$ y $p = \mathtt{C}\langle q'\rangle[\alpha \setminus \beta]$ con $\mathtt{C}\langle q\rangle \rightsquigarrow c$. Luego, $o \rightsquigarrow c[\alpha \setminus \beta] = o'$. Por *h.i.* existe c' tal que $\mathtt{C}\langle q'\rangle \rightsquigarrow c'$ y $c \simeq c'$. Más aún, se tiene $p \rightsquigarrow c'[\alpha \setminus \beta] = c'$. Se concluye dado que $o' = c[\alpha \setminus \beta] \simeq c'[\alpha \setminus \beta] = p'$.

- $\mathtt{Q} = \mathtt{T} \cdot s$. Luego, $o = \mathtt{T}\langle q \rangle \cdot s$ y $p = \mathtt{T}\langle q' \rangle \cdot s$. Hay dos casos de reducción posibles:
  1. $\mathtt{T}\langle q \rangle \rightsquigarrow t$. Luego, $o \rightsquigarrow t \cdot s = o'$. Por *h.i.* existe t' tal que $\mathtt{T}\langle q' \rangle \rightsquigarrow t'$ y $t \simeq t'$. Más aún, se tiene $p \rightsquigarrow t' \cdot s = p'$. Se concluye dado que $o' = t \cdot s \simeq t' \cdot s = p'$.
  2. $s \rightsquigarrow s'$. Luego, $o \rightsquigarrow \mathtt{T}\langle q \rangle \cdot s' = o'$ y $p \rightsquigarrow \mathtt{T}\langle q' \rangle \cdot s' = p'$. Se concluye dado que $\mathtt{T}\langle q \rangle \simeq \mathtt{T}\langle q' \rangle$ implica $o' \simeq p'$.
- $\mathtt{Q} = t \cdot \mathtt{S}$. Luego, $o = t \cdot \mathtt{S}\langle q \rangle$ y $p = t \cdot \mathtt{S}\langle q' \rangle$. Hay dos casos de reducción posibles:
  1. $\mathtt{S}\langle q \rangle \rightsquigarrow s$. Luego, $o \rightsquigarrow t \cdot s = o'$. Por *h.i.* existe s' tal que $\mathtt{S}\langle q' \rangle \rightsquigarrow s'$ y $s \simeq s'$. Más aún, se tiene $p \rightsquigarrow t \cdot s' = p'$. Se concluye dado que $o' = t \cdot s \simeq t \cdot s' = p'$.
  2. $t \rightsquigarrow t'$. Luego, $o \rightsquigarrow t' \cdot \mathtt{S}\langle q \rangle = o'$ y $p \rightsquigarrow t' \cdot \mathtt{S}\langle q' \rangle = p'$. Se concluye dado que $\mathtt{S}\langle q \rangle \simeq \mathtt{S}\langle q' \rangle$ implica $o' \simeq p'$.

□

Bibliografía

[ABEV19] Mauricio Ayala-Rincón, Eduardo Bonelli, Juan Edi, and Andrés Viso. Typed path polymorphism. *Theor. Comput. Sci.*, 781:111–130, 2019.

[ABKL14] Beniamino Accattoli, Eduardo Bonelli, Delia Kesner, and Carlos Lombardi. A nonstandard standardization theorem. In Suresh Jagannathan and Peter Sewell, editors, *The 41st Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL '14, San Diego, CA, USA, January 20-21, 2014*, pages 659–670. ACM, 2014.

[ABM14] Beniamino Accattoli, Pablo Barenbaum, and Damiano Mazza. Distilling abstract machines. In Johan Jeuring and Manuel M. T. Chakravarty, editors, *Proceedings of the 19th ACM SIGPLAN international conference on Functional programming, Gothenburg, Sweden, September 1-3, 2014*, pages 363–376. ACM, 2014.

[AC93] Roberto M. Amadio and Luca Cardelli. Subtyping recursive types. *ACM Trans. Program. Lang. Syst.*, 15(4):575–631, 1993.

[Acc13] Beniamino Accattoli. Compressing polarized boxes. In Kupferman [Kup13], pages 428–437.

[Acc18] Beniamino Accattoli. Proof nets and the linear substitution calculus. In Bernd Fischer and Tarmo Uustalu, editors, *Theoretical Aspects of Computing - ICTAC 2018 - 15th International Colloquium, Stellenbosch, South Africa, October 16-19, 2018, Proceedings*, volume 11187 of *Lecture Notes in Computer Science*, pages 37–61. Springer, 2018.

[ADH+12] Zena M. Ariola, Paul Downen, Hugo Herbelin, Keiko Nakata, and Alexis Saurin. Classical call-by-need sequent calculi: The unity of semantic artifacts. In Tom Schrijvers and Peter Thiemann, editors, *Functional and Logic Programming - 11th International Symposium, FLOPS 2012, Kobe, Japan, May 23-25, 2012. Proceedings*, volume 7294 of *Lecture Notes in Computer Science*, pages 32–46. Springer, 2012.

[AF97] Zena M. Ariola and Matthias Felleisen. The call-by-need lambda calculus. *J. Funct. Program.*, 7(3):265–301, 1997.

[AFM+95] Zena M. Ariola, Matthias Felleisen, John Maraist, Martin Odersky, and Philip Wadler. The call-by-need lambda calculus. In Ron K. Cytron and Peter Lee, editors, *Conference Record of POPL'95: 22nd ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, San Francisco, California, USA, January 23-25, 1995*, pages 233–246. ACM Press, 1995.

[AGK18] Beniamino Accattoli, Stéphane Graham-Lengrand, and Delia Kesner. Tight typings and split bounds. *PACMPL*, 2(ICFP):94:1–94:30, 2018.

[AHS11] Zena M. Ariola, Hugo Herbelin, and Alexis Saurin. Classical call-by-need and duality. In C.-H. Luke Ong, editor, *Typed Lambda Calculi and Applications - 10th International Conference, TLCA 2011, Novi Sad, Serbia, June 1-3, 2011. Proceedings*, volume 6690 of *Lecture Notes in Computer Science*, pages 27–44. Springer, 2011.

[AK96] Zena M. Ariola and Jan Willem Klop. Equational term graph rewriting. *Fundam. Inform.*, 26(3/4):207–240, 1996.

[AK10] Beniamino Accattoli and Delia Kesner. The structural *lambda*-calculus. In Anuj Dawar and Helmut Veith, editors, *Computer Science Logic, 24th International Workshop, CSL 2010, 19th Annual Conference of the EACSL, Brno, Czech Republic, August 23-27, 2010. Proceedings*, volume 6247 of *Lecture Notes in Computer Science*, pages 381–395. Springer, 2010.

[AK12] Beniamino Accattoli and Delia Kesner. The permutative λ-calculus. In Nikolaj Bjørner and Andrei Voronkov, editors, *Logic for Programming, Artificial Intelligence, and Reasoning - 18th International Conference, LPAR-18, Mérida, Venezuela, March 11-15, 2012. Proceedings*, volume 7180 of *Lecture Notes in Computer Science*, pages 23–36. Springer, 2012.

[AL95] Andrea Asperti and Cosimo Laneve. Paths, computations and labels in the lambda-calculus. *Theor. Comput. Sci.*, 142(2):277–297, 1995.

[All90] Frances E. Allen, editor. *Conference Record of the Seventeenth Annual ACM Symposium on Principles of Programming Languages, San Francisco, California, USA, January 1990*. ACM Press, 1990.

[AM01] Andrea Asperti and Harry G. Mairson. Parallel beta reduction is not elementary recursive. *Inf. Comput.*, 170(1):49–80, 2001.

[AMR06] Ariel Arbiser, Alexandre Miquel, and Alejandro Ríos. A lambda-calculus with constructors. In Frank Pfenning, editor, *Term Rewriting and Applications, 17th International Conference, RTA 2006, Seattle, WA, USA, August 12-14, 2006, Proceedings*, volume 4098 of *Lecture Notes in Computer Science*, pages 181–196. Springer, 2006.

[AMR09] Ariel Arbiser, Alexandre Miquel, and Alejandro Ríos. The lambda-calculus with constructors: Syntax, confluence and separation. *J. Funct. Program.*, 19(5):581–631, 2009.

[Aud94] Philippe Audebaud. Explicit substitutions for the lambda-mu-calculus. Technical Report RR-1994-26, LIP, Ecole Normale Supérieure de Lyon, Lyon, 1994.

[Bal10] Thibaut Balabonski. Optimality for dynamic patterns. In Temur Kutsia, Wolfgang Schreiner, and Maribel Fernández, editors, *Proceedings of the 12th International ACM SIGPLAN Conference on Principles and Practice of Declarative Programming, July 26-28, 2010, Hagenberg, Austria*, pages 231–242. ACM, 2010.

[Bal12] Thibaut Balabonski. *La pleine paresse, une certaine optimalité*. PhD thesis, Université Paris-Diderot, 2012.

[Bal13] Thibaut Balabonski. Weak optimality, and the meaning of sharing. In Greg Morrisett and Tarmo Uustalu, editors, *ACM SIGPLAN International Conference on Functional Programming, ICFP'13, Boston, MA, USA - September 25 - 27, 2013*, pages 263–274. ACM, 2013.

[Bar85] Hendrik Pieter Barendregt. *The lambda calculus - its syntax and semantics*, volume 103 of *Studies in logic and the foundations of mathematics*. North-Holland, 1985.

[Bar92] Hendrik Pieter Barendregt. Lambda calculi with types. In S. Abramsky, H. P. Barendregt, Dov M. Gabbay, and S. E. Maibaum, editors, *Handbook of Logic in Computer Science*, volume 2, pages 117–309. Oxford University Press, Inc., New York, NY, USA, 1992.

[BBBK17] Thibaut Balabonski, Pablo Barenbaum, Eduardo Bonelli, and Delia Kesner. Foundations of strong call by need. *PACMPL*, 1(ICFP):20:1–20:29, 2017.

[BCKL03] Gilles Barthe, Horatiu Cirstea, Claude Kirchner, and Luigi Liquori. Pure patterns type systems. In Alex Aiken and Greg Morrisett, editors, *Conference Record of POPL 2003: The 30th SIGPLAN-SIGACT Symposium on Principles of Programming Languages, New Orleans, Louisisana, USA, January 15-17, 2003*, pages 250–261. ACM, 2003.

[BDS13] Hendrik Pieter Barendregt, Wil Dekkers, and Richard Statman. *Lambda Calculus with Types*. Perspectives in logic. Cambridge University Press, 2013.

[BH98] Michael Brandt and Fritz Henglein. Coinductive axiomatization of recursive type equality and subtyping. *Fundam. Inform.*, 33(4):309–338, 1998.

[BKKS87] Hendrik P. Barendregt, Richard Kennaway, Jan Willem Klop, and M. Ronan Sleep. Needed reduction and spine strategies for the lambda calculus. *Inf. Comput.*, 75(3):191–231, 1987.

[BKLR12] Eduardo Bonelli, Delia Kesner, Carlos Lombardi, and Alejandro Ríos. Normalisation for dynamic pattern calculi. In Ashish Tiwari, editor, *23rd International Conference on Rewriting Techniques and Applications (RTA'12) , RTA 2012, May 28 - June 2, 2012, Nagoya, Japan*, volume 15 of *LIPIcs*, pages 117–132. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2012.

[BKLR17] Eduardo Bonelli, Delia Kesner, Carlos Lombardi, and Alejandro Ríos. On abstract normalisation beyond neededness. *Theor. Comput. Sci.*, 672:36–63, 2017.

[BKR18] Antonio Bucciarelli, Delia Kesner, and Simona Ronchi Della Rocca. Inhabitation for non-idempotent intersection types. *Logical Methods in Computer Science*, 14(3), 2018.

[BKV17] Antonio Bucciarelli, Delia Kesner, and Daniel Ventura. Non-idempotent intersection types for the lambda-calculus. *Logic Journal of the IGPL*, 25(4):431–464, 2017.

[BKvO03] Marc Bezem, Jan Willem Klop, and Vincent van Oostrom. *Term Rewriting Systems (TeReSe)*. Cambridge University Press, 2003.

[BL13] Alexis Bernadet and Stéphane Lengrand. Non-idempotent intersection types and strong normalisation. *Logical Methods in Computer Science*, 9(4), 2013.

[BN98] Franz Baader and Tobias Nipkow. *Term rewriting and all that*. Cambridge University Press, 1998.

[BR12] Erika De Benedetti and Simona Ronchi Della Rocca. Bounding normalization time through intersection types. In Stéphane Graham-Lengrand and Luca Paolini, editors, *Proceedings Sixth Workshop on Intersection Types and Related Systems, ITRS 2012, Dubrovnik, Croatia, 29th June 2012*, volume 121 of *EPTCS*, pages 48–57, 2012.

[BV17] Eduardo Bonelli and Andrés Viso. Strong normalization for recursive types with strong equality. Technical report, Universidad de Buenos Aires, Buenos Aires, Argentina, 2017.

[Car92] Felice Cardone. An algebraic approach to the interpretation of recursive types. In Jean-Claude Raoult, editor, *CAAP*, volume 581 of *Lecture Notes in Computer Science*, pages 66–85. Springer, 1992.

[CD78] Mario Coppo and Mariangiola Dezani-Ciancaglini. A new type assignment for λ-terms. *Arch. Math. Log.*, 19(1):139–156, 1978.

[CD80] Mario Coppo and Mariangiola Dezani-Ciancaglini. An extension of the basic functionality theory for the λ-calculus. *Notre Dame Journal of Formal Logic*, 21(4):685–693, 1980.

[CF12] Stephen Chang and Matthias Felleisen. The call-by-need lambda calculus, revisited. In Helmut Seidl, editor, *Programming Languages and Systems - 21st European Symposium on Programming, ESOP 2012, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2012, Tallinn, Estonia, March 24 - April 1, 2012. Proceedings*, volume 7211 of *Lecture Notes in Computer Science*, pages 128–147. Springer, 2012.

[CG05] Dario Colazzo and Giorgio Ghelli. Subtyping recursion and parametric polymorphism in kernel fun. *Inf. Comput.*, 198(2):71–147, 2005.

[CH00] Pierre-Louis Curien and Hugo Herbelin. The duality of computation. In Martin Odersky and Philip Wadler, editors, *Proceedings of the Fifth ACM SIGPLAN International Conference on Functional Programming (ICFP '00), Montreal, Canada, September 18-21, 2000.*, pages 233–243. ACM, 2000.

[CHL96] Pierre-Louis Curien, Thérèse Hardin, and Jean-Jacques Lévy. Confluence properties of weak and strong calculi of explicit substitutions. *J. ACM*, 43(2):362–397, 1996.

[Chu32] Alonzo Church. A set of postulates for the foundation of logic. *Annals of mathematics*, 33(2):346–366, 1932.

[CK98] Horatiu Cirstea and Claude Kirchner. ρ-calculus. Its Syntax and Basic Properties. In *CCL*, pages 66–85, 1998.

[CK01] Horatiu Cirstea and Claude Kirchner. The rewriting calculus - part I. *Logic Journal of the IGPL*, 9(3):339–375, 2001.

[CK04] Serenella Cerrito and Delia Kesner. Pattern matching as cut elimination. *Theor. Comput. Sci.*, 323(1-3):71–127, 2004.

[CKL01a] Horatiu Cirstea, Claude Kirchner, and Luigi Liquori. Matching power. In Aart Middeldorp, editor, *Rewriting Techniques and Applications, 12th International Conference, RTA 2001, Utrecht, The Netherlands, May 22-24, 2001, Proceedings*, volume 2051 of *LNCS*, pages 77–92. Springer, 2001.

[CKL01b] Horatiu Cirstea, Claude Kirchner, and Luigi Liquori. The rho cube. In Furio Honsell and Marino Miculan, editors, *Foundations of Software Science and Computation Structures, 4th International Conference, FOSSACS 2001 (ETAPS 2001), Genova, Italy, April 2-6, 2001, Proceedings*, volume 2030 of *LNCS*, pages 168–183. Springer, 2001.

[CKL02] Horatiu Cirstea, Claude Kirchner, and Luigi Liquori. Rewriting calculus with(out) types. *Electr. Notes Theor. Comput. Sci.*, 71:3–19, 2002.

[CKP00] Roberto Di Cosmo, Delia Kesner, and Emmanuel Polonowski. Proof-nets and explicit substitutions. In Jerzy Tiuryn, editor, *Foundations of Software Science and Computation Structures, Third International Conference, FOSSACS 2000, Held as Part of the Joint European Conferences on Theory and Practice of Software,ETAPS 2000, Berlin, Germany, March 25 - April 2, 2000, Proceedings*, volume 1784 of *Lecture Notes in Computer Science*, pages 63–81. Springer, 2000.

[CLW03] Horatiu Cirstea, Luigi Liquori, and Benjamin Wack. Rewriting calculus with fixpoints: Untyped and first-order systems. In Stefano Berardi, Mario Coppo, and Ferruccio Damiani, editors, *Types for Proofs and Programs, International Workshop, TYPES 2003, Torino, Italy, April 30 - May 4, 2003, Revised Selected Papers*, volume 3085 of *LNCS*, pages 147–161. Springer, 2003.

[CMMS91] Luca Cardelli, Simone Martini, John C. Mitchell, and Andre Scedrov. An extension of system F with subtyping. In Takayasu Ito and Albert R. Meyer, editors, *TACS '91, Sendai, Japan, September 24-27, 1991, Proceedings*, volume 526 of *Lecture Notes in Computer Science*, pages 750–770. Springer, 1991.

[Cop98] Mario Coppo. Recursive types: the syntactic and semantic approaches. In *Theories of Types and Proofs*, pages 16–41. Kyoto University, 1998.

[Cou83] Bruno Courcelle. Fundamental properties of infinite trees. *Theor. Comput. Sci.*, 25:95–169, 1983.

[dC07] Daniel de Carvalho. *Sémantiques de la logique linéaire et temps de calcul.* PhD thesis, Université Aix-Marseille II, 2007.

[dG94] Philippe de Groote. On the relation between the lambda-mu-calculus and the syntactic theory of sequential control. In Frank Pfenning, editor, *Logic Programming and Automated Reasoning, 5th International Conference, LPAR'94, Kiev, Ukraine, July 16-22, 1994, Proceedings*, volume 822 of *Lecture Notes in Computer Science*, pages 31–43. Springer, 1994.

[DM79] Nachum Dershowitz and Zohar Manna. Proving termination with multiset orderings. *Commun. ACM*, 22(8):465–476, 1979.

[DPR05] Roberto Di Cosmo, François Pottier, and Didier Rémy. Subtyping recursive types modulo associative commutative products. In Pawel Urzyczyn, editor, *TLCA 2005, Nara, Japan, April 21-23, 2005, Proceedings*, volume 3461 of *Lecture Notes in Computer Science*, pages 179–193. Springer, 2005.

[DR95] Vincent Danos and Laurent Regnier. Proof-nets and the Hilbert space. In *Workshop on Advances in Linear Logic, New York, NY, USA*, pages 307–328. Cambridge University Press, 1995.

[DST80] Peter J. Downey, Ravi Sethi, and Robert Endre Tarjan. Variations on the common subexpression problem. *J. ACM*, 27(4):758–771, 1980.

[Edi15] Juan Edi. Chequeo de tipos eficiente para path polymorphism. Master's thesis, Universidad de Buenos Aires, 2015.

[EVB15] Juan Edi, Andrés Viso, and Eduardo Bonelli. Efficient type checking for path polymorphism. In Tarmo Uustalu, editor, *21st International Conference on Types for Proofs and Programs, TYPES 2015, May 18-21, 2015, Tallinn, Estonia*, volume 69 of *LIPIcs*, pages 6:1–6:23. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2015.

[FF87] Matthias Felleisen and Daniel P. Friedman. Control operators, the secd-machine, and the λ-calculus. In Martin Wirsing, editor, *Formal Description of Programming Concepts - III: Proceedings of the IFIP TC 2/WG 2.2 Working Conference on Formal Description of Programming Concepts - III, Ebberup, Denmark, 25-28 August 1986*, pages 193–222. North-Holland, 1987.

[GAL92] Georges Gonthier, Martín Abadi, and Jean-Jacques Lévy. The geometry of optimal lambda reduction. In Ravi Sethi, editor, *Conference Record of the Nineteenth Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, Albuquerque, New Mexico, USA, January 19-22, 1992*, pages 15–26. ACM Press, 1992.

[Gar94] Philippa Gardner. Discovering needed reductions using type theory. In Masami Hagiya and John C. Mitchell, editors, *Theoretical Aspects of Computer Software, International Conference TACS '94, Sendai, Japan, April 19-22, 1994, Proceedings*, volume 789 of *Lecture Notes in Computer Science*, pages 555–574. Springer, 1994.

[Gen35a] Gerhard Gentzen. Untersuchungen über das logische schließen. I. *Mathematische Zeitschrift*, 39(1):176–210, 1935.

[Gen35b] Gerhard Gentzen. Untersuchungen über das logische schließen. II. *Mathematische Zeitschrift*, 39(1):405–431, 1935.

[GHP13] Tom Gundersen, Willem Heijltjes, and Michel Parigot. Atomic lambda calculus: A typed lambda-calculus with explicit sharing. In Kupferman [Kup13], pages 311–320.

[Gir87] Jean-Yves Girard. Linear logic. *Theoretical Computer Science*, 50:1–102, 1987.

[GK96] John R. W. Glauert and Zurab Khasidashvili. Relative normalization in deterministic residual structures. In Hélène Kirchner, editor, *Trees in Algebra and Programming - CAAP'96, 21st International Colloquium, Linköping, Sweden, April, 22-24, 1996, Proceedings*, volume 1059 of *Lecture Notes in Computer Science*, pages 180–195. Springer, 1996.

[Gri90] Timothy Griffin. A formulae-as-types notion of control. In Allen [All90], pages 47–58.

[GS17] Stefano Guerrini and Marco Solieri. Is the optimal implementation inefficient? elementarily not. In Miller [Mil17], pages 17:1–17:16.

[Gue96] Stefano Guerrini. *Theoretical and Practical Issues of Optimal Implementations of Functional Languages*. PhD thesis, Università di Pisa, 1996.

[Gue99] Stefano Guerrini. A general theory of sharing graphs. *Theor. Comput. Sci.*, 227(1-2):99–151, 1999.

[HL10] Kohei Honda and Olivier Laurent. An exact correspondence between a typed pi-calculus and polarised proof-nets. *Theor. Comput. Sci.*, 411(22-24):2223–2238, 2010.

[Jay04] C. Barry Jay. The pattern calculus. *ACM Trans. Program. Lang. Syst.*, 26(6):911–937, 2004.

[Jay09] Barry Jay. *Pattern Calculus - Computing with Functions and Structures*. Springer, 2009.

[JK06] Barry Jay and Delia Kesner. Pure pattern calculus. In Peter Sestoft, editor, *Programming Languages and Systems, 15th European Symposium on Programming, ESOP 2006, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2006, Vienna, Austria, March 27-28, 2006, Proceedings*, volume 3924 of *Lecture Notes in Computer Science*, pages 100–114. Springer, 2006.

[JK09] Barry Jay and Delia Kesner. First-class patterns. *J. Funct. Program.*, 19(2):191–225, 2009.

[JP97] Trevor Jim and Jens Palsberg. Type inference in systems of recursive types with subtyping, 1997.

[Kah03] Wolfram Kahl. Basic pattern matching calculi: Syntax, reduction, confluence, and normalisation, 2003.

[Kah04] Wolfram Kahl. Basic pattern matching calculi: a fresh view on matching failure. In Yukiyoshi Kameyama and Peter J. Stuckey, editors, *Functional and Logic Programming, 7th International Symposium, FLOPS 2004, Nara, Japan, April 7-9, 2004, Proceedings*, volume 2998 of *Lecture Notes in Computer Science*, pages 276–290. Springer, 2004.

[KBV20] Delia Kesner, Eduardo Bonelli, and Andrés Viso. Strong bisimulation for control operators (invited talk). In Maribel Fernández and Anca Muscholl, editors, *28th EACSL Annual Conference on Computer Science Logic, CSL 2020, January 13-16, 2020, Barcelona, Spain*, volume 152 of *LIPIcs*, pages 4:1–4:23. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

[Kes09] Delia Kesner. A theory of explicit substitutions with safe and full composition. *Logical Methods in Computer Science*, 5(3), 2009.

[Kes16] Delia Kesner. Reasoning about call-by-need by means of types. In Bart Jacobs and Christof Löding, editors, *Foundations of Software Science and Computation Structures - 19th International Conference, FOSSACS 2016, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2016, Eindhoven, The Netherlands, April 2-8, 2016, Proceedings*, volume 9634 of *Lecture Notes in Computer Science*, pages 424–441. Springer, 2016.

[Kfo00] A. J. Kfoury. A linearization of the lambda-calculus and consequences. *J. Log. Comput.*, 10(3):411–436, 2000.

[Kha93] Zurab Khasidashvili. Optimal normalization in orthogonal term rewriting systems. In Claude Kirchner, editor, *Rewriting Techniques and Applications, 5th International Conference, RTA-93, Montreal, Canada, June 16-18, 1993, Proceedings*, volume 690 of *Lecture Notes in Computer Science*, pages 243–258. Springer, 1993.

[KL05] Delia Kesner and Stéphane Lengrand. Extending the explicit substitution paradigm. In Jürgen Giesl, editor, *Term Rewriting and Applications, 16th International Conference, RTA 2005, Nara, Japan, April 19-21, 2005, Proceedings*, volume 3467 of *Lecture Notes in Computer Science*, pages 407–422. Springer, 2005.

[Klo80] Jan Willem Klop. *Combinatory reduction systems.* PhD thesis, Univ. Utrecht, 1980.

[KPS95] Dexter Kozen, Jens Palsberg, and Michael I. Schwartzbach. Efficient recursive subtyping. *Mathematical Structures in Computer Science*, 5(1):113–125, 1995.

[KRV18] Delia Kesner, Alejandro Ríos, and Andrés Viso. Call-by-need, neededness and all that. In Christel Baier and Ugo Dal Lago, editors, *Foundations of Software Science and Computation Structures - 21st International Conference, FOSSACS 2018, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2018, Thessaloniki, Greece, April 14-20, 2018, Proceedings*, volume 10803 of *Lecture Notes in Computer Science*, pages 241–257. Springer, 2018.

[Kup13] Orna Kupferman, editor. *28th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2013, New Orleans, LA, USA, June 25-28, 2013*. IEEE Computer Society, 2013.

[KV14] Delia Kesner and Daniel Ventura. Quantitative types for the linear substitution calculus. In Josep Díaz, Ivan Lanese, and Davide Sangiorgi, editors, *Theoretical Computer Science - 8th IFIP TC 1/WG 2.2 International Conference, TCS 2014, Rome, Italy, September 1-3, 2014. Proceedings*, volume 8705 of *Lecture Notes in Computer Science*, pages 296–310. Springer, 2014.

[KV17] Delia Kesner and Pierre Vial. Types as resources for classical natural deduction. In Miller [Mil17], pages 24:1–24:17.

[KV19a] Delia Kesner and Daniel Ventura. A resource aware semantics for a focused intuitionistic calculus. *Mathematical Structures in Computer Science*, 29(1):93–126, 2019.

[KV19b] Delia Kesner and Pierre Vial. Non-idempotent types for classical calculi in natural deduction style. *Logical Methods in Computer Science*, 16(1), 2019.

[KvOdV08] Jan Willem Klop, Vincent van Oostrom, and Roel C. de Vrijer. Lambda calculus with patterns. *Theor. Comput. Sci.*, 398(1-3):16–31, 2008.

[KvOvR93] Jan Willem Klop, Vincent van Oostrom, and Femke van Raamsdonk. Combinatory reduction systems: Introduction and survey. *Theor. Comput. Sci.*, 121(1&2):279–308, 1993.

[KW04] A. J. Kfoury and J. B. Wells. Principality and type inference for intersection types using expansion variables. *Theor. Comput. Sci.*, 311(1-3):1–70, 2004.

[Lam90] John Lamping. An algorithm for optimal lambda calculus reduction. In Allen [All90], pages 16–30.

[Lan65a] Peter J. Landin. Correspondence between ALGOL 60 and Church's lambda-notation: part I. *Commun. ACM*, 8(2):89–101, 1965.

[Lan65b] Peter J. Landin. A correspondence between ALGOL 60 and Church's lambda-notations: Part II. *Commun. ACM*, 8(3):158–167, 1965.

[Lan93] Cosimo Laneve. *Optimality and Concurrency in Interaction Systems*. PhD thesis, Università di Pisa, 1993.

[Lau02] Olivier Laurent. *A study of polarization in logic*. PhD thesis, Université de la Méditerranée - Aix-Marseille II, 2002.

[Lau03] Olivier Laurent. Polarized proof-nets and lambda-μ-calculus. *Theor. Comput. Sci.*, 290(1):161–188, 2003.

[Lév75] Jean-Jacques Lévy. *Réductions Correctes et Optimales dans le Lambda-Calcul*. PhD thesis, Université de Paris VII, 1975.

[LLD+04] Stéphane Lengrand, Pierre Lescanne, Daniel J. Dougherty, Mariangiola Dezani-Ciancaglini, and Steffen van Bakel. Intersection types for explicit substitutions. *Inf. Comput.*, 189(1):17–42, 2004.

[LR03] Olivier Laurent and Laurent Regnier. About translations of classical logic into polarized linear logic. In *18th IEEE Symposium on Logic in Computer Science (LICS 2003), 22-25 June 2003, Ottawa, Canada, Proceedings*, pages 11–20. IEEE Computer Society, 2003.

[LW05] Luigi Liquori and Benjamin Wack. The polymorphic rewriting-calculus: [type checking vs. type inference]. *Electr. Notes Theor. Comput. Sci.*, 117:89–111, 2005.

[Men87] P.F. Mendler. *Inductive definitions in type theory.* PhD thesis, Cornell University, NY, USA, 1987.

[Mil80] Robin Milner. *A Calculus of Communicating Systems*, volume 92 of *Lecture Notes in Computer Science*. Springer, 1980.

[Mil17] Dale Miller, editor. *2nd International Conference on Formal Structures for Computation and Deduction, FSCD 2017, September 3-9, 2017, Oxford, UK*, volume 84 of *LIPIcs*. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017.

[MOW98] John Maraist, Martin Odersky, and Philip Wadler. The call-by-need lambda calculus. *J. Funct. Program.*, 8(3):275–317, 1998.

[MRV20] Alexis Martín, Alejandro Ríos, and Andrés Viso. Pure Pattern Calculus *à la* de Bruijn. Technical report, Universidad de Buenos Aires, Buenos Aires, Argentina, 2020.

[Nip90] Tobias Nipkow. A critical pair lemma for higher-order rewrite systems and its applications to $\lambda^\star$. In Gérard P. Huet and Gordon D. Plotkin, editors, *Proceedings of the First Workshop on Logical Frameworks, Antibes, France, May 1990*, pages 361–376, 1990.

[NM04] Peter Møller Neergaard and Harry G. Mairson. Types, potency, and idempotency: why nonlinearity and amnesia make a type system work. In Chris Okasaki and Kathleen Fisher, editors, *Proceedings of the Ninth ACM SIGPLAN International Conference on Functional Programming, ICFP 2004, Snow Bird, UT, USA, September 19-21, 2004*, pages 138–149. ACM, 2004.

[Par92] Michel Parigot. Recursive programming with proofs. *Theor. Comput. Sci.*, 94(2):335–336, 1992.

[Pet09] Barbara Petit. A polymorphic type system for the lambda-calculus with constructors. In Pierre-Louis Curien, editor, *Typed Lambda Calculi and Applications, 9th International Conference, TLCA 2009, Brasilia, Brazil, July 1-3, 2009. Proceedings*, volume 5608 of *Lecture Notes in Computer Science*, pages 234–248. Springer, 2009.

[Pet11] Barbara Petit. Semantics of typed lambda-calculus with constructors. *Logical Methods in Computer Science*, 7(1), 2011.

[Pey87] Simon L. Peyton Jones. *The Implementation of Functional Programming Languages.* Prentice-Hall, 1987.

[Pie02] Benjamin C. Pierce. *Types and programming languages.* MIT Press, 2002.

[Pol04] Emmanuel Polonovski. *Subsitutions explicites, logique et normalisation. (Explicit substitutions, logic and normalization).* PhD thesis, Paris Diderot University, France, 2004.

[PR10a] Michele Pagani and Simona Ronchi Della Rocca. Linearity, non-determinism and solvability. *Fundam. Inform.*, 103(1-4):173–202, 2010.

[PR10b] Michele Pagani and Simona Ronchi Della Rocca. Solvability in resource lambda-calculus. In C.-H. Luke Ong, editor, *Foundations of Software Science and Computational Structures, 13th International Conference, FOSSACS 2010, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2010, Paphos, Cyprus, March*

20-28, 2010. Proceedings, volume 6014 of *Lecture Notes in Computer Science*, pages 358–373. Springer, 2010.

[PS16] Pierre-Marie Pédrot and Alexis Saurin. Classical by-need. In Peter Thiemann, editor, *Programming Languages and Systems - 25th European Symposium on Programming, ESOP 2016, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2016, Eindhoven, The Netherlands, April 2-8, 2016, Proceedings*, volume 9632 of *Lecture Notes in Computer Science*, pages 616–643. Springer, 2016.

[PZ01] Jens Palsberg and Tian Zhao. Efficient and flexible matching of recursive types. *Inf. Comput.*, 171(2):364–387, 2001.

[RC86] Jonathan Rees and William Clinger. Revised3 report on the algorithmic language scheme. *SIGPLAN Not.*, 21(12):37–79, 1986.

[Reg94] Laurent Regnier. Une équivalence sur les lambda-termes. *Theor. Comput. Sci.*, 126(2):281–292, 1994.

[Rey70] John C. Reynolds. GEDANKEN - a simple typeless language based on the principle of completeness and the reference concept. *Commun. ACM*, 13(5):308–319, 1970.

[Rey98] John C. Reynolds. Definitional interpreters for higher-order programming languages. *Higher-Order and Symbolic Computation*, 11(4):363–397, 1998.

[Roc88] Simona Ronchi Della Rocca. Principal type scheme and unification for intersection type discipline. *Theor. Comput. Sci.*, 59:181–209, 1988.

[SS98] Gerald Jay Sussman and Guy L. Steele. Scheme: A interpreter for extended lambda calculus. *Higher Order Symbol. Comput.*, 11(4):405–439, 1998.

[Ste84] Guy L. Steele. *Common LISP: The Language*. Digital Press, USA, 1984.

[SU06] Morten Heine Sørensen and Pawel Urzyczyn. *Lectures on the Curry-Howard Isomorphism, Volume 149 (Studies in Logic and the Foundations of Mathematics)*. Elsevier Science Inc., New York, NY, USA, 2006.

[Tak95] Masako Takahashi. Parallel reductions in lambda-calculus. *Inf. Comput.*, 118(1):120–127, 1995.

[Tar55] Alfred Tarski. A lattice-theoretical fixpoint theorem and its applications. *Pacific J. Math.*, 5(2):285–309, 1955.

[vB92] Steffen van Bakel. Complete restrictions of the intersection type discipline. *Theor. Comput. Sci.*, 102(1):135–163, 1992.

[VBA15] Andrés Viso, Eduardo Bonelli, and Mauricio Ayala-Rincón. Type soundness for path polymorphism. In Mario R. F. Benevides and René Thiemann, editors, *Proceedings of the Tenth Workshop on Logical and Semantic Frameworks, with Applications, LSFA 2015, Natal, Brazil, August 31 - September 1, 2015*, volume 323 of *Electronic Notes in Theoretical Computer Science*, pages 235–251. Elsevier, 2015.

[vBV14] Steffen van Bakel and Maria Grazia Vigliotti. A fully-abstract semantics of lambda-mu in the pi-calculus. In Paulo Oliva, editor, *Proceedings Fifth International Workshop on Classical Logic and Computation, CL&C 2014, Vienna, Austria, July 13, 2014*, volume 164 of *EPTCS*, pages 33–47, 2014.

[Via17] Pierre Vial. *Non-Idempotent Intersection Types, Beyond Lambda-Calculus.* PhD thesis, Université Paris-Diderot, 2017.

[vO90] Vincent van Oostrom. Lambda calculus with patterns. Technical Report IR-228, Vrije Universiteit, Amsterdam, 1990.

[Vou04] Jerome Vouillon. Subtyping union types. In Jerzy Marcinkowski and Andrzej Tarlecki, editors, *CSL*, volume 3210 of *Lecture Notes in Computer Science*, pages 415–429. Springer, 2004.

[Vui74] Jean Vuillemin. *Proof-techniques for Recursive Programs.* PhD thesis, Stanford University, 1974.

[Vui75] Jean Vuillemin. *Syntaxe, Sémantique et Axiomatique d'un Langage de Programmation Simple.* PhD thesis, Université de Paris VII, 1975.

[Wad71] Christopher P. Wadsworth. *Semantics and Pragmatics of the Lambda Calculus.* PhD thesis, Oxford University, 1971.

[Zha02] Tian Zhao. *Type Matching and Type Inference for Object-Oriented Systems.* PhD thesis, Computer Science, Purdue University, 2002.