Álgebra Superior: Una perspectiva típica

Nicky García Fierros

10 de diciembre de 2024

Índice

1. Introducción

1.1. ¿Teoría de tipos? ¿Y la teoría de conjuntos?

2. Algunos conceptos categóricos

2.1. Introducción

La idea de esta sección es presentar simplemente algunos conceptos de la teoría de categorías a los cuales se harán referencia a lo largo del texto. Aunque es posible introducir el contenido principal del texto sin hacer mención explícita a las categorías, el autor encuentra fascinante, bella y esclarecedora la conexión que existe entre la lógica matemática, la teoría de tipos y la teoría de categorías y por lo tanto se ha decidido incluir esta sección así como las referencias explícitas a las categorías a lo largo del texto. Además, otra motivación para incluir esta sección en la tesis es que lamentablemente en la facultad de ciencias no es común que se impartan

cursos de forma obligatoria de teoría de categorías por lo que el autor considera que es irrazonable asumir que la lectora o el lector esté familiarizado con las categorías.

Dado que el principal contenido de este texto no son las categorías sino la teoría homotópica de tipos y su aplicación a la formalización de matemáticas, no se ahondará en las categorías más allá de lo necesario para exhibir la conexión entre las categorías y la teoría de tipos y la riqueza que esta introduce a la teoría; sin embargo, en tanto que el propósito también es aquel de motivar al lector o a la lectora a explorar estas conexiones, se incluirán referencias a textos donde se puede profundizar en el tema.

2.2. Primeros conceptos

Definición 2.1 (Categoría)**.** Una categoría $\mathscr{C}$ consiste de la siguiente información:

1. Una colección de objetos $\mathrm{Obj}(\mathscr{C})$.
2. Para cada par de objetos $A, B \in \mathrm{Obj}(\mathscr{C})$ una colección de morfismos $\mathscr{C}(A, B)$.
3. Una noción de composición entre morfismos de tal modo que si $f \in \mathscr{C}(A, B)$ y $g \in \mathscr{C}(B, C)$ entonces existe un morfismo $gf \in \mathscr{C}(A, C)$.

$$A \xrightarrow{f} B \xrightarrow{g} C \qquad A \xrightarrow[gf]{} C$$

Definición 2.2 (Categoría cartesiana cerrada (CCC) [2])**.** Una categoría $\mathscr{C}$ es cartesiana cerrada o *CCC* si

- Existe un objeto terminal $\mathbb{1}$.
- Existen operaciones $(\_ \times \_)$ y $(\_)^{(\_)}$ tales que:
  - Para toda $A \in \mathrm{Obj}(\mathscr{C})$ existe una único morfismo $A \xrightarrow{!A} \mathbb{1}$.
  - $\mathscr{C}(C, A \times B) \cong \mathscr{C}(C, A) \times \mathscr{C}(C, B)$.
  - $\mathscr{C}(A, C^B) \cong \mathscr{C}(A \times B, C)$.

Definir B para argu mentar el punto 3 d la definici de CCC y mostrar q Sets es C

Definición 2.3 (Topos)**.** Una categoría $\mathscr{E}$ es un topos si tiene la siguiente estructura:

- Para cada diagrama $X \to B \leftarrow Y$ existe un producto fibrado.
- Tiene un objeto terminal $\mathbb{1}$.
- Existe un objeto Ω y una flecha $\top : \mathbb{1} \to \Omega$ tal que para cualquier monomofismo $m : S \rightarrowtail B$ existe una única flecha $\chi_B : B \to \Omega \in \mathscr{E}$ tal que el siguiente diagrama es un producto fibrado:

$$\begin{array}{ccc} S & \longrightarrow & \mathbb{1} \\ {\scriptstyle m}\downarrow & \lrcorner & \downarrow{\scriptstyle \top} \\ B & \xrightarrow[\chi_B]{} & \Omega \end{array}$$

- Para cualquier objeto B existen un objeto $P\ B$ y una flecha $\in_B$: $B \times P\ B \to \Omega$ tal que para cualquier flecha $f : B \times A \to \Omega$ existe una única flecha $g : A \to P\ B$ tal que el siguiente diagrama conmuta:

$$\begin{array}{ccc} A & & B \times A & \xrightarrow{\forall f} & \Omega \\ {\scriptstyle \exists! g}\downarrow & & {\scriptstyle Id \times g}\downarrow & & \| \\ P\ B & & B \times P\ B & \xrightarrow[\in_B]{} & \Omega \end{array}$$

Observación 1. Un topos en particular es una categoría cartesiana cerrada.

Definición 2.4 (Limite)**.**

Definición 2.5 (Transformación natural)**.**

Definición 2.6 (Adjunción)**.**

3. Teoría de tipos dependientes y la formalización de matemáticas en Agda

3.1. Introducción

Meterle m
paja a est

La teoría homotópica de tipos es un área de estudio de las matemáticas relativamente nueva. Esta área de estudio contempla herramientas de la teoría de los lenguajes de programación, el álgebra, la teoría de categorías, la lógica matemática y la topología. El poder expresivo del lenguaje formal empleado por la teoría de tipos homotópica así como su fundamento teórico es tan expresivo y general que permite ofrecer una teoría alternativa a la teoría de conjuntos para fundamentar las matemáticas. Dentro de las ventajas que brinda emplear este lenguaje está la posibilidad de utilizar computadoras para verificar la correctud de demostraciones matemáticas.

Es importante notar que al ser esta una teoría constructiva desde su concepción, técnicas propias que dependen de axiomas o teoremas no constructivos como lo son la ley del tercer excluido, o el teorema de elección generalizado, no se encuentran disponibles en todos los contextos a diferencia de las "matemáticas clásicas".

En esta segunda parte del trabajo se explorarán de forma breve y concisa temas de la teoría homotópica de tipos con el objetivo de proponer y dar una base teórica para una formalización del temario de álgebra superior.

3.2. Teoría de tipos dependientes

3.2.1. Juicios, contextos y derivaciones

En la teoría de tipos se emplea un lenguaje formal que está basado en la deducción natural pues es un sistema en el que se cuenta con reglas de inferencia que se pueden combinar para formar derivaciones. Las derivaciones nos importan porque son el principal mecanismo para producir *términos* de un tipo determinado.

Como es de esperarse del título que carga la teoría de tipos, un **tipo** es un objeto primitivo de la teoría de tipos de la misma forma que un conjunto es un objeto primitivo de la teoría de conjuntos.

Este texto contempla que los objetos matemáticos tienen tipos, y un objeto matemático siempre está dado junto con su tipo, de modo que un objeto matemático no es solamente un objeto, sino un objeto con un determinado tipo. Como veremos en esta sección, un tipo está dado junto con una "receta" que describe cómo construir **elementos** de este determinado tipo.

Como se mencionó antes, un término es el resultado de la aplicación de reglas de inferencia y, como el autor no desea arruinar el placentero proceso de entender a un nuevo objeto matemático, conforme avancemos en este trabajo florecerán distintas formas útiles de pensar a los tipos y sus términos.

Entenderemos por una **derivación** a una sucesión de aplicaciones de **reglas de inferencia**.

Comenzamos por definir precisamente qué es un juicio en este lenguaje.

Definición 3.1 (Juicios, contextos)**.** Un **juicio** es alguna expresión de la forma:

1. $\Gamma \vdash A$ type (Desde Γ se deduce que A es un tipo)
2. $\Gamma \vdash a : A$ type (Desde Γ se deduce que a es un término de tipo A)
3. $\Gamma \vdash A \equiv B$ type (Desde Γ se deduce que A es un tipo juiciosamente equivalente al tipo B)
4. $\Gamma \vdash a \equiv b \ : A$ (Desde Γ se deduce que los términos a y b de tipo A son juiciosamente equivalentes)

donde Γ es una lista finita de declaraciones de variables tales que para cada $1 \leq k \leq n$ se puede derivar el juicio

$$x_1 : A_1, x_2 : A_2(x_1), \ldots, x_k : A_k(x_1, x_2, \ldots, x_{k-1}) \vdash A_{k+1}(x_1, x_2, \ldots, x_{k-1}, x_k) \text{ type}$$

y recibe el nombre de **contexto**; y lo que se encuentra a la derecha del símbolo $\vdash$ (léase "desde _ se deduce _") recibe el nombre de **tesis de juicio**.

Los contextos, de forma análoga a su rol en el cálculo de secuentes, denotan los supuestos que se están considerando para obtener la tesis de juicio. En tanto que los elementos potencialmente pueden ser suposiciones que carecen de fundamento previamente derivado se les suelen llamar *variables*. Los juicios los pensamos como hechos, a diferencia de las proposiciones; las cuales potencialmente son verdaderas o falsas. Alternativamente llamaremos **elementos** a los términos de un tipo dado, de modo que un juicio $a : A$ se puede leer como a es un elemento de tipo A.

El orden en los contextos nos importa porque nos interesa mantener la noción de dependencia de una expresión con respecto a otra, de modo que contextos como

$$\Gamma := \{f : A \supset B, A \text{ type}, B \text{ type}\}$$

resultan particularmente peligrosos, pues darían a entender que $f : A \supset B$ es una prueba de $A \supset B$ sin contar antes del conocimiento de que A y B son tipos y entonces se presentaría una situación semejante a la de suponer lo que se desea demostrar. El motivo por el cual la noción de dependencia es importante es porque en la teoría de tipos de Per Martin-Löf se hace una distinción importante entre **juicios** y **proposiciones**[1]. En los cursos de lógica (y álgebra superior) se enseña que una proposición es una oración para la cual es posible asignar un valor de verdad (*verdadero* o *falso* sea lo que signifique eso). Para los matemáticos y lógicos intuicionistas esta noción es incompleta en virtud de las dificultades que presenta el justificar las reglas para la formación de proposiciones mediante la cuantificación sobre dominios infinitos [2], por lo que se ofrece una noción alternativa para proposicoón:

Encontrar
entre las
donde se
afirma es-
to (adema
del libro
Martin L

una proposición se define al exhibir una demostración para lo que se propone

y

una proposición es verdadera si tiene una demostración.

[7]

De este modo, cuando escribimos algún juicio, como por ejemplo

$$\vdash a : A$$

es porque **a : A está demostrado**, y no estamos hipotetizando. Por otro lado, lo que escribimos como combinación de símbolos lógicos (por supuesto siguiendo las reglas de formación de su gramática) y está por verse su veracidad denominamos proposiciones. Así, cuando en un contexto escribimos

$$A \text{ type}, x : A, B(x) \text{ type}, y : B(x) \text{ type}, \ldots$$

estamos declarando que nuestras suposiciones son coherentes con el resto de nuestras reglas de formación en tanto que son resultado de juicios anteriores.

Observación 2. Observe que nuestra definición de contexto permite la existencia de un contexto vacío pues por un argumento de vacuidad se verifica la satisfacibilidad de la propiedad de un contexto.

Observación 3. Obsérvese que la condición impuesta sobre un contexto se puede verificar de forma recursiva o inductiva:

- El caso base es mostrar que $x_1 : A_1$ se deduce desde el contexto vacío. Para afirmar que $x_1 : A_1$ es un juicio válido se debe haber deducido (o supuesto) que A_1 es un tipo en el contexto vacío.
- La clausula inductiva es codificada por la propiedad que define a un contexto.

Para verificar de forma recursiva que una lista de declaraciones de la forma

$$x_1 : A_1, x_2 : A_2(x_1), \ldots, x_k : A_k(x_1, x_2, \ldots, x_{k-1}) \vdash x_{k+1} : A_{k+1}(x_1, \ldots, x_{k-1}, x_k) \text{ type}$$

es un contexto basta probar que una lista de declaraciones de la forma

$$x_1 : A_1, x_2 : A_2(x_1), \ldots, x_k : A_{k-2}(x_1, x_2, \ldots, x_{k-2}) \vdash x_k : A_k(x_1, \ldots, x_{k-1}) \text{ type}$$

y así de forma sucesiva hasta dar con el caso base.

Definición 3.2 (Derivación)**.** Una **derivación** es un árbol finito con raíz en el que cada vértice es una regla de inferencia válida. A la raíz del árbol se le llama **conclusión** y a las hojas **hipótesis**.

Nos reservamos el derecho de poder definir nuevas reglas de inferencia a partir de otras, y diremos que estas nuevas reglas son **derivables**.

3.2.2. Familias de tipos

Poner el
digo en a
de esto

Una idea universal bastante útil es la de un "agrupamiento de agrupamientos"; ejemplos clásicos de este patrón de pensamiento son las familias de conjuntos; en teoría de conjuntos; y los enunciados; en lógica de primer órden. En la teoría de tipos dependientes de Per Martin-Löf contamos con un marco de trabajo que engloba esta idea, llamada *familia de tipos*.

[1] En [7] se pueden leer las ideas originales que concibieron a esta teoría.

[2] Necesito citar esto gg

Definición 3.3 (Familia de tipos)**.** Si A es un tipo en un contexto Γ, una **familia de tipos** $B(x)$ es un tipo en el contexto $\Gamma, x : A$ (o también diremos que $B(x)$ es un **tipo indizado sobre** A en el contexto Γ) y escribimos formalmente este hecho como

$$\Gamma, x : A \vdash B(x) \text{ type}$$

y en su forma de regla de inferencia podemos **introducirla** como

$$\frac{\Gamma \vdash x : A \qquad \varnothing \vdash A \text{ type}}{B(x) \text{ type}}$$

Por comodidad se suele omitir el contexto vacío y solamente se escribe la tesis de juicio, de modo que escribimos:

$$\frac{\Gamma \vdash x : A \qquad A \text{ type}}{B(x) \text{ type}}$$

o si damos por obvio que A tiene que ser un tipo para que el juicio $\Gamma \vdash x : A$ sea válido podemos solamente convenir escribir

$$\frac{\Gamma \vdash x : A}{\Gamma \vdash B(x) \text{ type}}$$

Por conveniencia y claridad, a partir de este punto emplearemos las convenciones de escritura que nos permiten obviar cosas a menos de que sea necesario para esclarecer.

Observación 4. Resulta bastante útil pensar a una familia de tipos como un tipo que varía según los términos de otro tipo. Es decir, si abusamos de notación, podemos pensar a una familia de tipos como una función

$$\begin{aligned} \text{Term(A)} &\to \text{Types} \\ x : A &\mapsto B(x) \text{ type} \end{aligned}$$

Un un futuro no muy lejano se exhibirá cómo expresar este hecho de manera formal dentro del lenguaje de la teoría de tipos dependiente.

Como es de esperarse que de una colecciones de colecciones podamos tomar *una parte*, análogamente de una familia de tipos podemos considerar lo que llamaremos una **sección**.

Definición 3.4 (Sección de una familia de tipos)**.** Si B es una familia de tipos sobre A en el contexto Γ, diremos que una **sección** de B es un término $b(x) : B(x)$ en un contexto $\Gamma, x : A$. En símbolos:

$$\Gamma, x : A \vdash b(x) : B(x)$$

La **regla de introducción** asociada entonces es:

$$\frac{\Gamma \vdash x : A \qquad \Gamma, x : A \vdash B(x) \text{ type}}{\Gamma \vdash b(x) : B(x)}$$

Y podemos entenderla como: *Si podemos deducir del contexto* Γ *que* x *es un término de tipo* A *y que* B *es una familia de tipos sobre* A*, entonces podemos deducir desde* Γ *que* $b(x) : B(x)$ *es una sección de* B*.*

Observación 5. Nótese que tanto el término como el tipo dependen del término $x : A$, de modo que abusando de la notación podemos pensar a este proceso como una función

$$\begin{aligned} \text{Term(A)} \times (\text{Term(A)} \to \text{Types}) &\to \text{Term}(B(x)) \\ \langle x : A \ , \ x : A \mapsto B(x) \text{ type} \rangle &\mapsto b(x) : B(x) \end{aligned}$$

3.2.3. Clases de reglas de inferencia

Las siguientes reglas de inferencia describen de forma explicita las suposiciones de que hicimos en la definición 3.3. Es de esperarse que, si tenemos en un contexto las variables A type, $x : A$, entonces desde ese mismo contexto podamos deducir A type y $x : A$ por separado.

$$\frac{\Gamma, x : A \vdash B(x) \text{ type}}{\Gamma \vdash A \text{ type}} \qquad \frac{\Gamma \vdash A \equiv B \text{ type}}{\Gamma \vdash A \text{ type}} \qquad \frac{\Gamma \vdash A \equiv B \text{ type}}{\Gamma \vdash B \text{ type}}$$

$$\frac{\Gamma \vdash a \equiv b : A}{\Gamma \vdash a : A} \qquad \frac{\Gamma \vdash a \equiv b : A}{\Gamma \vdash b : A} \qquad \frac{\Gamma \vdash a : A}{\Gamma \vdash A \text{ type}}$$

En tanto que es de interés que la noción de *ser juiciosamente iguales* sea una buena noción de equivalencia, es de esperarse que se postulen reglas que testifican que esta noción satisface los axiomas de una relación de equivalencia. [3]

$$\frac{\Gamma \vdash A \text{ type}}{\Gamma \vdash A \equiv A \text{ type}} \qquad \frac{\Gamma \vdash A \equiv B \text{ type}}{\Gamma \vdash B \equiv A \text{ type}}$$

$$\frac{\Gamma \vdash A \equiv B \text{ type} \qquad \Gamma \vdash B \equiv C \text{ type}}{\Gamma \vdash A \equiv C \text{ type}}$$

$$\frac{\Gamma \vdash a : A}{\Gamma \vdash a \equiv a : A} \qquad \frac{\Gamma \vdash a \equiv b : A}{\Gamma \vdash b \equiv a : A} \qquad \frac{\Gamma \vdash a \equiv b : A \qquad \Gamma \vdash b \equiv c : A}{\Gamma \vdash a \equiv c : A}$$

También es de esperarse que, si se tienen que dos tipos son juiciosamente equivalentes, y puedes deducir una tesis de juicio $\mathfrak{T}$ a partir de una variable, entonces al intercambiar el tipo sobre el que tomas la variable por su equivalente la misma tesis de juicio debería poder deducirse.

$$\frac{\Gamma \vdash A \equiv B \text{ type} \qquad \Gamma, x : A, \Theta \vdash \mathfrak{T}}{\Gamma, x : B, \Theta \vdash \mathfrak{T}} \text{ ConvVar}$$

Donde Θ es una extensión cualquiera del contexto $\Gamma, x : A$. Por ejemplo en el caso en que $\mathfrak{T}$ es $C(x)$ type tenemos

$$\frac{\Gamma \vdash A \equiv B \text{ type} \qquad \Gamma, x : A, \Theta \vdash C(x) \text{ type}}{\Gamma, x : B, \Theta \vdash C(x) \text{ type}}$$

En general, el concepto de sustituir es uno muy importante en las estructuras de pensamiento humanas, por lo que es de esperarse que dicho concepto también esté persente en esta teoría.

Al tener ya una noción de igualdad, podemos comenzar a hacernos preguntas sobre sustituciones de elementos en otros elementos.

Consideremos una sección $f(x)$ de una familia de tipos $B(x)$ indizado por $x : A$ en un contexto Γ. Al ser que $f(x)$ como expresión contiene al menos una referencia a x, y $f(x) : B(x)$ entonces al sustituir cada referencia de x por algún $a : A$ de forma simultánea sobre $f(x) : B(x)$ debemos esperar que $f[a/x]$ sea un elemento de $B[a/x]$.

En general, **la regla de sustitución**

$$\frac{\Gamma \vdash a : A \qquad \Gamma, x : A, \Theta \vdash \mathfrak{T}}{\Gamma, \Theta[a/x] \vdash \mathfrak{T}[a/x]} \, a/x$$

también nos permite sustituir de forma simultánea sobre un contexto dado. Es importante mencionar que el orden de los elementos en un contexto es escencial, pues $\Gamma, x : A, \Theta$ no es lo mismo que $\Gamma, \Theta, x : A$. El orden es indicativo de cierta potencial dependencia entre elementos del contexto.

Por ejemplo, si

$$\Gamma, x : A, s : S, m : M \vdash C(x) \; type$$

entonces por la regla de sustitución, dado $\Gamma \vdash a : A$ tendríamos

$$\Gamma, s : S[a/x], m : M[a/x] \vdash C[a/x] \; type$$

donde potencialmente los tipos de s, m sean distintos. Por otro lado si

$$\Gamma, s : S, m : M, x : A \vdash C(x) \; type$$

entonces la regla de sustitución sólo nos permite asegurar que

$$\Gamma, s : S, m : M \vdash C[a/x] \; type$$

y los tipos de s y m no sufren cambios.

Es de esperarse que, si se tiene que dos elementos son juiciosamente iguales, entonces la sustitución respeta esta igualdad juiciosa.

$$\frac{\Gamma \vdash a \equiv a' : A \qquad \Gamma, x : A, \Delta \vdash B \text{ type}}{\Gamma, \Delta[a/x] \vdash B[a/x] \equiv B[a'/x]}$$

[3] Esta noción de equivalencia fue concebida en términos de una equivalencia en cuanto a reducción, es decir, si tras aplicar reglas de inferencia a dos expresiones sintacticamente distintas se concluye que ambas expresiones comparten una misma forma (normal) tras simplificar dichas expresiones lo más posible, entonces ambas expresiones son juiciosamente equivalentes [6] [5].

$$\frac{\Gamma \vdash a \equiv a' : A \qquad \Gamma, x : A, \Delta \vdash b : B \text{ type}}{\Gamma, \Delta[a/x] \vdash b[a/x] \equiv b[a'/x] : B[a/x]}\ [a/x]\text{-cong}$$

Notación.

- A partir de este momento acordamos en denotar a $b[x/a]$ por simplemente $b(a)$, y a $B[x/a]$ por $B(a)$ a menos que sea necesario emplear la notación usual de sustitución.
- La regla también se denotará por *S-cong* o simplemente *cong* para aligerar la notación cuando sea necesario.

Consideremos la siguente regla

$$\frac{\Gamma \vdash A \text{ type} \qquad \Gamma, \Theta \vdash \mathfrak{T}}{\Gamma, x : A, \Theta \vdash \mathfrak{T}}\ \text{W}$$

A primer vistazo parece ser que la regla nos quiere decir que si podemos derivar una tesis de juicio $\mathfrak{T}$ desde un contexto Γ, entonces al introducir una variable no presente en $\mathfrak{T}$ ni en Γ (una variable libre) podamos deducir exactamente lo mismo.

Sin embargo esta no es toda la historia. Una pregunta natural que surge es *¿qué ocurre con la nueva dependencia agregada sobre la variable $x : A$?*. Por ejemplo, consideremos que desde un contexto Γ podemos deducir que A type y B type. ¡Entonces la regla anterior nos dice que podemos deducir que B es una familia sobre A!

$$\frac{\Gamma \vdash A \text{ type} \qquad \Gamma \vdash B \text{ type}}{\Gamma, x : A \vdash B \text{ type}}$$

En tanto que estamos agregando hipótesis adicionales a una derivación, decimos que estamos *debilitando* la conclusión. De ahí que el nombre de la regla sea ***weakening*** o ***regla de debilitamiento***. Identificamos esta regla en un árbol de derivación por la letra W.

La **regla de introducción de variables**, o también conocida como **regla del elemento genérico**, es un caso particular de la regla de debilitamiento, en tanto que si la tesis de juicio es A type y Θ es vacío, entonces podemos derivar

$$\frac{\Gamma \vdash A \text{ type}}{\Gamma, x : A \vdash x : A}\ \text{VAR}$$

Como ejemplos de derivaciones se presentan a continuación algunas reglas derivables útiles desde las reglas discutidas anteriormente:

Teorema 3.1 (sustitución de variables por otras)**.**
Sean Γ y Θ contextos y $\mathfrak{T}$ una tesis de juicio tales que

$$\Gamma, x : A, \Theta \vdash \mathfrak{T}$$

Entonces se puede deducir que

$$\Gamma, x' : A, \Theta[x'/x] \vdash \mathfrak{T}[x/x']$$

Demostración.

$$\frac{\dfrac{\Gamma \vdash A \text{ type} \qquad \Gamma, x : A, \Theta \vdash \mathfrak{T}}{\Gamma, x' : A, x : A, \Theta \vdash \mathfrak{T}}\ \text{W} \qquad \dfrac{\Gamma \vdash A \text{ type}}{\Gamma, x' : A \vdash x' : A}\ \text{VAR}}{\Gamma, x' : A, \Theta[x'/x] \vdash \mathfrak{T}[x/x']}$$

□

Teorema 3.2 (regla de intercambio del orden de variables)**.**

$$\frac{\Gamma, x : A, y : B, \Theta \vdash \mathfrak{T}}{\Gamma, y : B, x : A, \Theta \vdash \mathfrak{T}}$$

Demostración.
Notemos que por el teorema 3.1 tenemos que de $\Gamma, x : A, y : B, \Theta \vdash \mathfrak{T}$ podemos deducir

$$\Gamma[y'/y], x : A, y' : B, \Theta[y'/y] \vdash \mathfrak{T}[y/y']$$

La idea ahora es agregar $y : B$ de vuelta al árbol de derivación y sustituir adecuadamente para obtener la tesis de juicio deseada.

$$\mathrm{W}\,\frac{\dfrac{\Gamma, x : A, y : B, \Theta \vdash \mathfrak{T}}{\Gamma[y'/y], x : A, y' : B, \Theta[y'/y] \vdash \mathfrak{T}[y/y']} \qquad \Gamma \vdash B \text{ type}}{\Gamma[y'/y], y : B, x : A, y' : B, \Theta[y'/y] \vdash \mathfrak{T}[y/y']} \qquad \frac{\dfrac{\Gamma \vdash B \text{ type}}{\Gamma, y : B \vdash y : B}\,\mathrm{VAR} \qquad \Gamma \vdash A \text{ type}}{\Gamma, y : B, x : A \vdash y : B}\,\mathrm{W}}{\Gamma, y : B, x : A, \Theta \vdash \mathfrak{T}}\,[y/y']$$

□

Teorema 3.3 (relga de conversión de elementos)**.**

$$\frac{\Gamma \vdash A \equiv A' \text{ type} \qquad \Gamma \vdash a : A}{\Gamma \vdash a : A'}$$

Demostración.

$$\mathrm{ConvVar}\,\frac{\mathrm{VAR}\,\dfrac{\dfrac{\Gamma \vdash A \equiv A' \text{ type}}{\Gamma \vdash A' \text{ type}}}{\Gamma, x' : A \vdash x : A'} \qquad \Gamma \vdash A \equiv A' \text{ type}}{\Gamma, x : A \vdash x : A'} \qquad \frac{\Gamma, x : A \vdash a : A}{\Gamma \vdash a : A'}\,[a/x]$$

□

Teorema 3.4 (regla de congruencia para la conversión de elementos)**.**

$$\frac{\Gamma \vdash A \equiv A' \text{ type} \qquad \Gamma \vdash a \equiv b : A}{\Gamma \vdash a \equiv b : A'}$$

Demostración.

Aqui no p
de simple-
mente hal
aplicado e
teorema a
terior???
Son la mi
ma prueb

$$\frac{\mathrm{VAR}\,\dfrac{\dfrac{\Gamma \vdash A \equiv A' \text{ type}}{\Gamma \vdash A' \text{ type}}}{\Gamma, x : A' \vdash x : A'} \qquad \Gamma \vdash A \equiv A'}{\Gamma, x : A \vdash x : A'}\,\mathrm{ConvVar} \qquad \frac{\Gamma \vdash a \equiv b : A}{\Gamma \vdash a \equiv b : A'}\,[a \equiv b/x]$$

Lo otro que se me ocurre es probar que por separado $a : A'$ y $b : A'$ y entonces como $a \equiv b : A$ y por separado son también de tipo A' entonces $a \equiv b : A'$. Sin embargo, desconozco qué regla podría usar para unir a estas dos letras en una equivalencia :(

$$\frac{\dfrac{\Gamma \vdash a \equiv b : A}{\Gamma \vdash a : A} \qquad \Gamma \vdash A \equiv A' \text{ type}}{\Gamma \vdash a : A'}\,\text{teo 5.2} \qquad \frac{\dfrac{\Gamma \vdash a \equiv b : A}{\Gamma \vdash b : A} \qquad \Gamma \vdash A \equiv A' \text{ type}}{\Gamma \vdash b : A'}\,\text{teo 5.2}}{\Gamma \vdash a \equiv b : A'}\,????$$

□

3.3. Tipos primitivos

Ya que contamos con un minimo fundamento sobre el cual poder construir tipos, procedemos a discutir sobre aquellos tipos que el sistema permite construir desde un contexto vacío. Estos tipos formarán los bloques básicos sobre los que haremos las construcciones de nuevos tipos y más aún, serán de gran utilidad para comenzar a darnos una idea de cómo codificar objetos matemáticos en este lenguaje.

3.3.1. Funciones dependientes

Poner cod
en agda d
esto

Una función dependiente podemos pensarla como aquella tal que permite que el codominio varíe en función de un elemento del dominio. En la teoría de conjuntos se presenta una construcción semejante, y es la del producto generalizado. Recordando, el producto generalizado de una familia indizada es

$$\prod_{i \in \Gamma} X_i := \{f : \Gamma \to \bigcup_{i \in \Gamma} X_i \mid \forall i \in \Gamma \; f(i) \in X_i\}$$

de modo que un elemento f del producto cartesiano es una función que dibuja una serie de posibilidades para el valor que puede tomar $f(i) \in X_i$. Esta misma situación se nos presentó al introducir las familias de tipos,

$$\mathrm{Ctx}, i : \Gamma \vdash X(i) \text{ type}$$
$$\mathrm{Ctx}, i : \Gamma \vdash f(i) : X(i)$$

de modo que $f : X$ es una función dependiente. Este tipo tiene distintos nombres en la literatura: **tipo** Π, **el producto cartesiano de una familia de tipos** y **el tipo de funciones dependientes**.

Definición 3.5 (tipo de funciones dependientes)**.** La **regla de formación** del tipo de funciones dependientes establece que la existencia de una familia de tipos es suficiente para obtener un tipo de funciones dependientes:
Regla de formación

$$\frac{\Gamma, x : A \vdash B(x) \text{ type}}{\Gamma \vdash \prod_{(x:A)} B(x) \text{ type}} \Pi$$

Además, la formación del tipo producto es congruente con la igualdad juiciosa, esto es,

$$\frac{\Gamma \vdash A \equiv A' \text{ type} \qquad \Gamma x : A \vdash B(x) \equiv B'(x) \text{ type}}{\Gamma \vdash \prod_{(x:A)} B(x) \text{ type} \equiv \prod_{(x:A')} B(x) \text{ type}} \Pi\text{-eq}$$

La **regla de introducción** establece que los elementos del tipo de funciones dependientes son exactamente las funciones que asignan a un término "índice" de la familia a una función.
Regla de introducción

$$\frac{\Gamma, x : A \vdash b(x) : B(x)}{\Gamma \vdash \lambda x \,.\, b(x) : B(x)} \lambda$$

Más aún, postulamos la congruencia de esta regla ante la igualdad juiciosa:

$$\frac{\Gamma, x : A \vdash b(x) \equiv b'(x) : B(x)}{\Gamma \vdash \lambda x \,.\, b(x) \equiv \lambda x \,.\, b'(x) : \prod_{(x:A)} B(x)} \lambda\text{-eq}$$

La **regla de eliminación** del tipo de funciones dependientes, como es de esperarse, nos permite eliminar de un árbol de deducción un término del tipo de funciones dependientes siempre y cuando podamos evaluarlo para obtener un término del tipo resultante:
Regla de eliminación

$$\frac{\Gamma \vdash f : \prod_{(x:A)} B(x)}{\Gamma, x : A \vdash f(x) : B(x)} \text{ev}$$

La **regla de cómputo** del tipo de funciones dependientes postula que la evaluación de un término de $\prod_{(x:A)} B(x)$ es simplemente evaluar el término dado en A en $b(x) : B(x)$, semejante a la reducción β del cálculo lambda:
Regla de cómputo

$$\frac{\Gamma, x : A \vdash b(x) : B(x)}{\Gamma, x : A \vdash (\lambda y \,.\, b(y))(x) \equiv b(x) : B(x)} \beta$$

Por otro lado, **la regla** η o también conocida como **regla/postulado de unicidad** nos asegura que los elementos de un tipo de funciones dependientes son exactamente funciones.

$$\frac{\Gamma \vdash b : \prod_{(x:A)} B(x)}{\Gamma \vdash \lambda x \,.\, b(x) \equiv b : \prod_{(x:A)} B(x)} \eta$$

Observación 6. Análogamente a su simíl en conjuntos, una familia de tipos involucra una elección, en este caso de un $b(x) : B(x)$ dado un $x : A$.

Observación 7. Observe que la regla de cómputo y la regla η son inversas mutuas.

Observe que, si tratamos con una familia de tipos constante; esto es que el tipo codominio no varia según el término índice; tenemos una función. Las reglas que definen al tipo de funciones dependientes se simplifican entonces:

$$\frac{\dfrac{\Gamma \vdash A \text{ type} \qquad \Gamma \vdash B \text{ type}}{\Gamma, x : A \vdash B \text{ type}}}{\Gamma \vdash \prod_{(x:A)} B \text{ type}} \text{ (weakening)}$$

De modo que, ante una situación como la anterior la regla de introducción nos diría que las funciones son exactamente las abstracciones lambdas sobre un tipo en función de un término:

$$\frac{\Gamma, x : A \vdash b(x) : B}{\Gamma \vdash \lambda x \,.\, b(x) : \prod_{(x:A)} B}$$

La regla de eliminación nos dice exactamente lo que esperaríamos de un tipo que codifica una función:

$$\frac{\Gamma \vdash f : \prod_{(x:A)} B}{\Gamma, x : A \vdash f(x) : B}$$

Si evaluamos una función f con dominio en A y codominio en B en un elemento $x : A$ del dominio, entonces $f(x) : B$.

Así, mediante una regla de derivación consolidamos nuestra definición del tipo de funciones o equivalentemente llamado tipo flecha:

Definición 3.6 (tipo de funciones)**.** El tipo de funciones de un tipo A en un tipo B se define como a continuación:

$$\frac{\dfrac{\dfrac{\Gamma \vdash A \text{ type} \qquad \Gamma \vdash B \text{ type}}{\Gamma, x : A \vdash B \text{ type}}}{\Gamma \vdash \prod_{(x:A)} B \text{ type}}}{\Gamma \vdash A \to B :\equiv \prod_{(x:A)} B \text{ type}}$$

En algunas ocasiones emplearemos la notación B^A para denotar $A \to B$. Esto es, B^A dentoa el tipo de funciones de A en B.

En general, dada una construcción podemos crear una definición con base en el resultado final. Para ello, conveniremos en el símbolo $:\equiv$ para denotar que se está realizando una definición. Como es de esperarse, las mismas reglas que aplicaban para el tipo de funciones dependientes aplican para nuestra definición del tipo de funciones:

$$\frac{\Gamma \vdash A \text{ type} \qquad \Gamma \vdash B \text{ type}}{\Gamma \vdash A \to B \text{ type}} \to \qquad \frac{\Gamma, x : A \vdash f(x) : B}{\Gamma \vdash \lambda \, x \,.\, f(x) : A \to B} \lambda$$

$$\frac{\Gamma \vdash A \equiv A' \text{ type} \qquad \Gamma \vdash B \equiv B' \text{ type}}{\Gamma \vdash A \to B \equiv A' \to B' \text{ type}} \to\text{-eq}$$

$$\frac{\Gamma, x : A \vdash b(x) \equiv b'(x) : B}{\Gamma \vdash \lambda x \,.\, b(x) \equiv \lambda x \,.\, b'(x) : A \to B} \lambda\text{-eq}$$

$$\frac{\Gamma \vdash f : A \to B}{\Gamma, x : A \vdash f(x) : B} \text{ ev} \qquad \frac{\Gamma \vdash f \equiv g : A \to B}{\Gamma, x : A \vdash f(x) \equiv g(x) : B} \text{ ev-eq}$$

$$\frac{\Gamma \vdash f : A \to B}{\Gamma \vdash \lambda x \,.\, f(x) \equiv f : A \to B} \eta \qquad \frac{\Gamma \vdash B \text{ type} \qquad \Gamma, y : A \vdash f(a) : B}{\Gamma, y : A \vdash (\lambda \, x \,.\, f(x))(y) \equiv f(y) : B} \beta$$

Poner la i
plementa-
ción en ag
de esto

Observación 8. Observe que dados dos tipos A y B podemos obtener una función genérica de A en B y evaluarla.

$$\frac{\dfrac{\dfrac{\Gamma \vdash A \text{ type} \qquad \Gamma \vdash B \text{ type}}{\Gamma \vdash A \to B \text{ type}} \to}{\Gamma, f : A \to B \vdash f : A \to B} \text{VAR}}{\Gamma, f : A \to B, x : A \vdash f(x) : B} \text{ ev}$$

Poner la
prueba en
agda de e
tambien

El siguiente lema nos permitirá simplificar las demostraciones al abstraer el proceso de evaluación de un término de un tipo flecha en un elemento de su dominio.

Lema 3.5.

$$\frac{\Gamma \vdash f : \prod_{(x:A)} B(x) \qquad \Gamma \vdash a : A}{\Gamma \vdash f(a) : B(a)}$$

Demostración.

$$\dfrac{\dfrac{\Gamma \vdash f : \prod_{(x:A)} B(x)}{\Gamma, x : A \vdash f(x) : B(x)}\,\text{ev} \qquad \dfrac{\Gamma \vdash a : A}{\Gamma, x : A \vdash a : A}\,\text{W}}{\Gamma, x : A \vdash f(a) : B(a)}\,a/x$$

□

Corolario 3.5.1.

$$\dfrac{\Gamma \vdash f : A \to B \qquad \Gamma \vdash a : A}{\Gamma \vdash f(a) : B}$$

Teorema 3.6 (Extensionalidad bajo la igualdad de juicio)**.**

$$\dfrac{\Gamma \vdash f : \prod_{(x:A)} B(x) \qquad \Gamma \vdash g : \prod_{(x:A)} B(x) \qquad \Gamma, x : A \vdash f(x) \equiv g(x) : B(x)}{\Gamma \vdash f \equiv g : \prod_{(x:A)} B(x)}$$

Demostración.

$$\dfrac{\dfrac{\lambda\text{-eq}\ \dfrac{\Gamma, x : A \vdash f(x) \equiv g(x) : B(x)}{\Gamma \vdash \lambda x\ .\ f(x) \equiv \lambda x\ .\ g(x) : \prod_{(x:A)} B(x)} \qquad \dfrac{\Gamma \vdash f : \prod_{(x:A)} B(x)}{\Gamma \vdash \lambda x\ .\ f(x) \equiv f : \prod_{(x:A)} B(x)}\,\eta}{\equiv\text{-trans}\ \Gamma \vdash \lambda x\ .\ f(x) \equiv g : \prod_{(x:A)} B(x)} \qquad \dfrac{\Gamma \vdash g : \prod_{(x:A)} B(x)}{\Gamma \vdash \lambda x\ .\ g(x) \equiv g : \prod_{(x:A)} B(x)}\,\eta}{\Gamma \vdash f \equiv g : \prod_{(x:A)} B(x)}$$

□

Corolario 3.6.1 (Extensionalidad bajo la igualdad de juicio)**.**

$$\dfrac{\Gamma \vdash f : B^A \qquad \Gamma \vdash g : B^A \qquad \Gamma, x : A \vdash f(x) \equiv g(x) : B(x)}{\Gamma \vdash f \equiv g : B^A}$$

Para terminar con esta subsección se presentan a continuación algunas construcciones útiles con tipos flecha.

Algunas construcciones útiles con el tipo flecha

La flecha identidad.

Deseamos definir un objeto que codifique a la flecha identidad. Sabemos que la flecha identidad es tal que para todo objeto perteneciente al dominio se corresponde a si mismo bajo esta flecha. En general, algo a observar de lo anterior es que en principio el dominio y contradominio de la flecha identidad puede ser el que sea mientras exista. De esta forma, comenzamos nuestra construcción postulando que de tener un tipo en algún contexto podemos entonces construir este objeto.

$$\Gamma \vdash A\ \text{type}$$

Luego, aplicando la regla de introducción de variables podemos obtener de lo anterior lo siguiente:

$$\dfrac{\Gamma \vdash A\ \text{type}}{\Gamma, x : A \vdash x : A}$$

Por la conclusión anterior, podemos entonces aplicar la regla de introducción del tipo flecha:

$$\dfrac{\dfrac{\Gamma \vdash A\ \text{type}}{\Gamma, x : A \vdash x : A}\,\text{VAR}}{\Gamma \vdash \lambda\ x\ .\ x : A \to A}\,\lambda$$

Para así poder concluir nuestra definición:

$$\dfrac{\dfrac{\dfrac{\Gamma \vdash A\ \text{type}}{\Gamma, x : A \vdash x : A}\,\text{VAR}}{\Gamma \vdash \lambda\ x\ .\ x : A \to A}\,\lambda}{\Gamma \vdash \text{Id}_A := \lambda\ x\ .\ x : A \to A}$$

Claramente nuestra flecha identidad debe satisfacer que todo elemento evaluado en dicha flecha es (juiciosamente) equivalente a si mismo.

Lema 3.7.

$$\dfrac{\Gamma \vdash A\ \text{type}}{\Gamma, x : A \vdash \text{Id}_A(x) \equiv x}$$

Demostración.

$$\dfrac{\dfrac{\dfrac{\dfrac{\Gamma \vdash A \text{ type}}{\Gamma \vdash \mathrm{Id}_A : A \to A}}{\Gamma \vdash \mathrm{Id}_A \equiv \lambda\, y.\, y : A \to A}}{\Gamma, x : A \vdash \mathrm{Id}_A(x) \equiv (\lambda\, y\, .\, y)(x) : A}\,\to\text{-ev}}{\Gamma, x : A \vdash \mathrm{Id}_A(x) \equiv x : A}\,\beta$$

□

Tomando múltiples argumentos

Ciertamente pareciera que nuestro tratamiento sobre el tipo flecha tiene la limitante sobre el número de argumentos que puede tomar una función. En las matemáticas que conocemos es común observar funciones que requiere de más de una entrada, como por ejemplo la suma aritmética entre dos números naturales. Sin embargo, el tratamiento dado sobre los elementos del tipo Π nos permite expresar esta clase de funciones. Para demostrarlo, consideremos el caso de la suma de dos números naturales:

$$+_{\mathbb{N}} : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$$

Notemos que si proporcionamos un número natural $n \in \mathbb{N}$ y lo aplicamos como entrada a la función suma, estamos ante la situación en donde cualquier otro número natural $m \in \mathbb{N}$ al ser aplicado como entrada a la función (manteniendo fijo a n) nos da como resultado la suma de la nueva entrada m con n. Es decir, podemos pensar que al aplicar un solo argumento a la función, restringimos las entradas a un grado de libertad menor, que es lo mismo que decir que estamos ante una nueva función que, para este caso particular, toma un argumento menos.

$$\begin{aligned} +_n : \mathbb{N} &\to \mathbb{N} \\ m &\mapsto m +_{\mathbb{N}} n \end{aligned}$$

Esta perspectiva entonces pareciera sugerir que las funciones que toman más de dos parámetros son simplemente funciones que toman un argumento y regresan una función que toma el siguiente argumento y así de forma sucesiva hasta llegar al resultado final.

$$\begin{aligned} +_{\mathbb{N}} : \mathbb{N} &\to (\mathbb{N} \to \mathbb{N}) \\ n &\mapsto +_n : \mathbb{N} \to \mathbb{N} \end{aligned}$$

Este proceso recibe el nombre de "currying" en honor al matemático Haskell Curry, a pesar de que fueron Frege y Schönfinkel quienes originalmente concibieron la idea.[4]

De manera semejante, si $C(x, y)$ es una familia de tipos indizada por dos elementos $x : A$ y $y : B$, entonces podemos formar el tipo

$$\dfrac{\dfrac{\Gamma, y : B \vdash \prod_{(y:B)} C(x, y) \text{ type}}{\Gamma, x : A \vdash \prod_{(y:B)} C(x, y) \text{ type}}\,\Pi}{\prod_{(x:A)} \prod_{(y:B)} C(x, y) \text{ type}}\,\Pi$$

Queda pendiente demostrar que efectivamente esta perspectiva es correcta.

Teorema 3.8 (Currying)**.**

La composición de flechas.
Consideremos tres tipos en un contexto Γ, A, B y C. Gracias a la regla de debilitamiento de los tres tipos podemos obtener al menos dos funciones genéricas: $A \to B$ y $B \to C$, un elemento de B y un elemento de C.

$$\dfrac{\dfrac{\dfrac{\Gamma \vdash A \text{ type} \qquad \Gamma \vdash B \text{ type}}{\Gamma \vdash A \to B \text{ type}}\,\to}{\Gamma, f : A \to B \vdash f : A \to B}\,\text{VAR}}{\Gamma, f : A \to B, x : A \vdash f(x) : B}\,\text{ev} \qquad \dfrac{\dfrac{\dfrac{\Gamma \vdash B \text{ type} \qquad \Gamma \vdash C \text{ type}}{\Gamma \vdash B \to C \text{ type}}\,\to}{\Gamma, g : B \to C \vdash g : B \to C}\,\text{VAR}}{\Gamma, g : B \to C, y : B \vdash g(y) : C}\,\text{ev}$$

En virtud de lo anterior, omitimos en el árbol de deducción estos pasos. Luego, como podemos evaluar funciones genéricas en elementos y contamos con uno, podemos obtener un elemento de C.

[4] Ver [1] y [10] para más información sobre sus origenes.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Gamma \vdash A \text{ type} \qquad \Gamma \vdash B \text{ type}}{\Gamma \vdash B^A \text{ type}}}{\Gamma, f : B^A \vdash f : B^A}}{\Gamma, f : B^A, x : A \vdash f(x) : B}}{\Gamma, f : B^A, x : A, g : B^A \vdash f(x) : B} \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Gamma \vdash B \text{ type} \qquad \Gamma \vdash C \text{ type}}{\Gamma \vdash B \to C \text{ type}}}{\Gamma, g : B \to C \vdash g : B \to C}}{\Gamma, g : B \to C, y : B \vdash g(y) : C}}{\Gamma, g : C^B, f : B^A, y : B \vdash g(y) : C}}{\Gamma, g : C^B, f : B^A, y : B, x : A \vdash g(y) : C}$$

Por sustitución de $f(x)$ sobre y obtenemos $g(f(x)) : C$. Luego, aplicando nuestro lema 3.5 podemos deducir

$$\Gamma, g : C^B, f : B^A, x : A \vdash g(f(x)) : C$$

$$\dfrac{\dfrac{\dfrac{\Gamma \vdash A \text{ type} \qquad \Gamma \vdash B \text{ type}}{\Gamma, f : B^A, x : A \vdash f(x) : B}}{\Gamma, f : B^A, x : A, g : C^B \vdash f(x) : B} \qquad \dfrac{\dfrac{\dfrac{\Gamma \vdash B \text{ type} \qquad \Gamma \vdash C \text{ type}}{\Gamma, g : B \to C, y : B \vdash g(y) : C}}{\Gamma, g : C^B, f : B^A, y : B \vdash g(y) : C}}{\Gamma, g : C^B, f : B^A, y : B, x : A \vdash g(y) : C}}{\Gamma, g : C^B, f : B^A, x : A \vdash g(f(x)) : C}$$

Abstrayendo sobre x obtenemos una función que nos recuerda a la composición de funciones.

$$\dfrac{\dfrac{\dfrac{\dfrac{\Gamma \vdash A \text{ type} \qquad \Gamma \vdash B \text{ type}}{\Gamma, f : B^A, x : A \vdash f(x) : B}}{\Gamma, f : B^A, x : A, g : C^B \vdash f(x) : B} \qquad \dfrac{\dfrac{\dfrac{\Gamma \vdash B \text{ type} \qquad \Gamma \vdash C \text{ type}}{\Gamma, g : B \to C, y : B \vdash g(y) : C}}{\Gamma, g : C^B, f : B^A, y : B \vdash g(y) : C}}{\Gamma, g : C^B, f : B^A, y : B, x : A \vdash g(y) : C}}{\Gamma, g : C^B, f : B^A, x : A \vdash g(f(x)) : C}}{\Gamma, g : C^B, f : B^A \vdash \lambda x \,.\, g(f(x)) : C^A}\,\lambda$$

Finalmente, abstraemos sobre f y g para dar con un término que es testigo de la existencia de la composición de funciones.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Gamma \vdash A \text{ type} \qquad \Gamma \vdash B \text{ type}}{\Gamma, f : B^A, x : A \vdash f(x) : B}}{\Gamma, f : B^A, x : A, g : C^B \vdash f(x) : B} \qquad \dfrac{\dfrac{\dfrac{\Gamma \vdash B \text{ type} \qquad \Gamma \vdash C \text{ type}}{\Gamma, g : B \to C, y : B \vdash g(y) : C}}{\Gamma, g : C^B, f : B^A, y : B \vdash g(y) : C}}{\Gamma, g : C^B, f : B^A, y : B, x : A \vdash g(y) : C}}{\Gamma, g : C^B, f : B^A, x : A \vdash g(f(x)) : C}}{\Gamma, g : C^B, f : B^A \vdash \lambda x \,.\, g(f(x)) : C^A}\,\lambda}{\Gamma, g : C^B \vdash \lambda f \,.\, \lambda x \,.\, g(f(x)) : B^A \to C^A}\,\lambda}{\Gamma \vdash \lambda g \,.\, (\lambda f \,.\, \lambda x \,.\, g(f(x))) : C^B \to (B^A \to C^A)}\,\lambda}{\Gamma \vdash \_ \circ \_ := \lambda g \,.\, (\lambda f \,.\, \lambda x \,.\, g(f(x))) : C^B \to (B^A \to C^A)}\,\lambda$$

Es decir, dados $f : B^A, g : C^B$

$$g \circ f \equiv \lambda x \,.\, g(f(x)) : C^A$$

Ya que tenemos una noción de composición de funciones y una identidad para cada tipo, estaría muy bien mostrar que nuestra noción de composición es asociativa.

Teorema 3.9 (Asociatividad de la composición)**.** Si de un contexto Γ se tienen flechas $f : B^A$, $g : C^B$ y $h : D^C$, entonces $h \circ (g \circ f) \equiv (h \circ g) \circ f : A \to D$.

Demostración.

$$\dfrac{\dfrac{\Gamma \vdash f : B^A}{\Gamma, x : A \vdash f(x) : B}\,\text{ev} \qquad \dfrac{\dfrac{\Gamma \vdash g : C^B}{\Gamma, y : B \vdash g(y) : C}\,\text{ev}}{\Gamma, y : B, x : A \vdash g(y) : C}\,\text{W}}{\Gamma, x : A \vdash g(f(x)) : C} \qquad \dfrac{\dfrac{\Gamma \vdash h : D^C}{\Gamma, z : C \vdash h(z) : D}\,\text{ev}}{\Gamma, z : C, x : A \vdash h(z) : D}\,\text{W}$$
$$\dfrac{\Gamma, x : A \vdash g(f(x)) : C \qquad \Gamma, z : C, x : A \vdash h(z) : D}{\Gamma, x : A \vdash h(g(f(x))) : D}\,g(f(x))/z$$

Observemos que de $\Gamma, x : A \vdash h(g(f(x))) : D$ podemos obtener las siguientes derivaciones tras aplicaciones sucesivas de la definición de la composición:

$$\dfrac{\dfrac{\dfrac{\dfrac{\Gamma, x : A \vdash h(g(f(x))) : D}{\Gamma, x : A \vdash h(g(f(x))) \equiv h(g(f(x))) : D}\equiv\text{-refl}}{\Gamma \vdash \lambda x\ .\ h(g(f(x))) \equiv \lambda x\ .\ h(g(f(x))) : D^A}\to\text{-eq}}{\Gamma \vdash \lambda x\ .\ h(g(f(x))) \equiv \lambda x\ .\ h((g \circ f)(x)) : D^A}}{\Gamma \vdash \lambda x\ .\ h(g(f(x))) \equiv \lambda x\ .\ h \circ (g \circ f)(x) : D^A}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\Gamma, x : A \vdash h(g(f(x))) : D}{\Gamma, x : A \vdash h(g(f(x))) \equiv h(g(f(x))) : D}\equiv\text{-refl}}{\Gamma \vdash \lambda x\ .\ h(g(f(x))) \equiv \lambda x\ .\ h(g(f(x))) : D^A}\to\text{-eq}}{\Gamma \vdash \lambda x\ .\ (h \circ g)(f(x)) \equiv \lambda x\ .\ h(g(f(x))) : D^A}}{\Gamma \vdash ((h \circ g) \circ f)(x) \equiv \lambda x\ .\ h(g(f(x))) : D^A}$$

De modo que, por la transitividad de $\equiv$ podemos conlcuir la equivalencia que deseamos.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Gamma, x : A \vdash h(g(f(x))) : D}{\Gamma, x : A \vdash h(g(f(x))) \equiv h(g(f(x))) : D}}{\Gamma \vdash \lambda x\ .\ h(g(f(x))) \equiv \lambda x\ .\ h(g(f(x))) : D^A}}{\Gamma \vdash \lambda x\ .\ h(g(f(x))) \equiv \lambda x\ .\ h((g \circ f)(x)) : D^A}}{\Gamma \vdash \lambda x\ .\ h(g(f(x))) \equiv \lambda x\ .\ h \circ (g \circ f)(x) : D^A} \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Gamma, x : A \vdash h(g(f(x))) : D}{\Gamma, x : A \vdash h(g(f(x))) \equiv h(g(f(x))) : D}}{\Gamma \vdash \lambda x\ .\ h(g(f(x))) \equiv \lambda x\ .\ h(g(f(x))) : D^A}}{\Gamma \vdash \lambda x\ .\ (h \circ g)(f(x)) \equiv \lambda x\ .\ h(g(f(x))) : D^A}}{\Gamma \vdash \lambda x\ .\ ((h \circ g) \circ f)(x) \equiv \lambda x\ .\ h(g(f(x))) : D^A}}{\Gamma \vdash ((h \circ g) \circ f)(x) \equiv h \circ (g \circ f)(x) : D^A}$$

□

Teorema 3.10. Sea $f : A \to B$ en un contexto Γ. Entonces $\mathrm{Id}_B \circ f \equiv f$ y $f \circ \mathrm{Id}_A \equiv f$.

Demostración. Notemos que el hecho $f \in \Gamma$ implica que A, B type $\in \Gamma$. Así,

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Gamma \vdash f : B^A}{\Gamma, x : A \vdash f(x) : B}\text{ev} \qquad \dfrac{\dfrac{\Gamma \vdash B \text{ type}}{\Gamma, y : B \vdash \mathrm{Id}_B(y) \equiv y : B}\text{lema 5.7} \qquad \Gamma \vdash A \text{ type}}{\Gamma, x : A, y : B \vdash \mathrm{Id}_B(y) \equiv y : B}\text{W}}{\Gamma, x : A \vdash \mathrm{Id}_B(f(x)) \equiv f(x) : B}[f(x)/y]}{\Gamma \vdash \lambda x\ .\ \mathrm{Id}_B(f(x)) \equiv \lambda x\ .\ f(x) : B^A}\lambda\text{-eq}}{\Gamma \vdash \mathrm{Id}_B \circ f \equiv f : B^A}\circ\text{-def} \qquad \dfrac{\Gamma \vdash f : B^A}{\Gamma \vdash \lambda x\ .\ f(x) \equiv f : B^A}\eta}{\Gamma \vdash \mathrm{Id}_B \circ f \equiv f : B^A}\equiv\text{-trans}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Gamma \vdash A \text{ type}}{\Gamma \vdash \mathrm{Id}_A : A^A}\text{ev}}{\Gamma, x : A \vdash \mathrm{Id}_A(x) : A}\text{ev}}{\Gamma, x : A \vdash \mathrm{Id}_A(x) \equiv x : A}\text{lema 5.7} \qquad \dfrac{\dfrac{\Gamma \vdash f : AB^A}{\Gamma, y : A \vdash f(y) : B}\text{ev} \qquad \Gamma \vdash A \text{ type}}{\Gamma, y : A, x : A \vdash f(y) : B}\text{W}}{\Gamma, y : A, x : A \vdash \mathrm{Id}_A(x) \equiv x : A}\text{W}}{\Gamma, x : A \vdash f(\mathrm{Id}_A(x)) \equiv f(x) : B}[\mathrm{Id}_A(x)/y]\text{-cong}}{\Gamma, x : A \vdash \lambda x.f(\mathrm{Id}_A(x)) \equiv \lambda x.f(x) : B^A}\lambda\text{-equiv}}{\Gamma, x : A \vdash f \circ \mathrm{Id}_A \equiv \lambda x.f(x) : B^A}\circ\text{-def} \qquad \dfrac{\Gamma \vdash f : B^A}{\Gamma \vdash \lambda x.f(x) \equiv f : B^A}\eta}{\Gamma, x : A \vdash f \circ \mathrm{Id}_A \equiv f : B^A}\equiv\text{-trans}$$

□

Con ello, los tipos flecha satisfacen:

- Para cualquier tipo A existe $\mathrm{Id}_A : A \to A$.
- Para cualesquiera dos tipos A, B existe un término $A \to B$.
- Dadas dos flechas $A \to B$, $B \to C$ existe una tercera flecha $A \to C$.
- Para cualesquiera flecha $f : A \to B$ se satisface
  - $f \circ \mathrm{Id}_A \equiv f$
  - $\mathrm{Id}_B \circ f \equiv f$
- Para cualesquiera tres flechas $f : A \to B, g : B \to C, h : C \to D$ se tiene que $f \circ (g \circ h) \equiv (f \circ g) \circ h$

y así entonces tenemos

Teorema 3.11. La colección

$$\mathbf{Type} := \{A \mid A \text{ type}\}$$

junto con la familia de colecciones con $A, B \in \mathbf{Type}$

$$\mathbf{Type}(A, B) := \{f \mid f : A \to B\}$$

conforman una categoría con la noción de composición definida por $\_ \circ \_$.

Para concluir con esta sección se presentan las siguientes construcciones de funciones que serán de útilidad en secciones posteriores y algunas de sus propiedades.

La función constante

$$\frac{\Gamma \vdash A \text{ type}}{\Gamma, y : B \vdash \text{const}_y : A \to B}$$

Construcción

Una técnica común al momento de querer construir algún objeto es observar las dependencias que debería tener cierto objeto y claro, tener clara la idea del objeto que se quiere construir. En el caso de la función constante la idea es muy simple, por lo que resulta en un excelente ejemplo de la aplicación de esta técnica. Partimos de un valor existente arbitrario, y queremos que nuestra función constante sea tal que para cualquier otro valor que pueda tomar nuestra función, al valuar dicho valor en la función obtengamos el valor del que partimos.

$$\frac{?}{\Gamma, y : B \vdash \lambda x \,.\, y : A \to B}$$

Por nuestras reglas de formación de tipos, sabemos que para llegar al término $\lambda x \,.\, y : A \to B$ debió haber ocurrido antes la aplicación de la regla λ

$$\dfrac{\dfrac{?}{\Gamma, y : B, x : A \vdash y : B}}{\Gamma, y : B \vdash \lambda x \,.\, y : A \to B}$$

Y la presencia de $x : A$ en la tesis de juicio $\Gamma, y : B, x : A \vdash y : B$ nos recuerda a la aplicación de la regla de debilitamiento sobre la tesis de juicio $\Gamma, y : B \vdash y : B$. Como tenemos por hipótesis $\Gamma \vdash A$ type podemos aplicar dicha regla sin mayor obstáculo

$$\dfrac{\dfrac{\dfrac{?}{\Gamma, y : B \vdash y : B} \quad \Gamma \vdash A \text{ type}}{\Gamma, y : B, x : A \vdash y : B}\,\mathrm{W}}{\Gamma, y : B \vdash \lambda x \,.\, y : A \to B}\,\lambda$$

Finalmente, al partir de la existencia de algún elemento en B, podemos simplemente suponer que partimos de la existencia de B en el contexto y con ello mediante la regla de elemento genérico obtener el juicio deseado.

$$\dfrac{\dfrac{\dfrac{\dfrac{\Gamma \vdash B \text{ type}}{\Gamma, y : B \vdash y : B} \quad \Gamma \vdash A \text{ type}}{\Gamma, y : B, x : A \vdash y : B}\,\mathrm{W}}{\Gamma, y : B \vdash \lambda x \,.\, y : A \to B}\,\lambda}{\Gamma, y : B \vdash \text{const}_y \equiv \lambda x \,.\, y : A \to B}$$

Observación 9.
Podemos corraborar que nuestra construcción se comporta como esperamos al observar el resultado de evaluar $\text{const}_y : A \to B$ en algún elemento de A. Notemos que por definición si $\Gamma, y : B \vdash \text{const}_y : A \to B$ entonces $\Gamma, x : A \vdash \text{const}_y(x) \equiv y : B$.

En efecto,

$$\dfrac{\dfrac{\dfrac{\Gamma, y : B \vdash \text{const}_y : A \to B}{\Gamma, y : B, x : A \vdash \text{const}_y(x) : B}\,\text{ev}}{\Gamma, y : B, x : A \vdash \text{const}_y(x) \equiv (\lambda z \,.\, y)(x) : B}}{\Gamma, y : B, x : A \vdash \text{const}_y(x) \equiv y : B}\,\beta$$

Propiedades

Lema 3.12.

$$\frac{\Gamma \vdash f : A \to B}{\Gamma, c : C \vdash \mathrm{const}_c \circ f \equiv \mathrm{const}_c : A \to C}$$

Demostración.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Gamma \vdash B\ \mathrm{type}}{\Gamma, c : C \vdash \mathrm{const}_c : B \to C}\ \mathrm{ev}}{\Gamma, c : C, z : B \vdash \mathrm{const}_c(z) : C}\qquad \Gamma \vdash A\ \mathrm{type}}{\Gamma, c : C, z : B, x : A \vdash \mathrm{const}_c(z) : C}\ \mathrm{W}\qquad \dfrac{\Gamma \vdash f : A \to B}{\Gamma, x : A \vdash f(x) : B}\ \mathrm{ev}}{\Gamma, c : C, x : A \vdash \mathrm{const}_c(f(x)) : C}}{\Gamma, c : C, x : A \vdash \mathrm{const}_c(f(x)) \equiv c : C}\ \mathrm{obs\ 9}}{\Gamma, c : C \vdash \lambda x\ .\ \mathrm{const}_c(f(x)) \equiv \lambda x\ .\ c : A \to C}\ \lambda\text{-eq}}{\Gamma, c : C \vdash \lambda x\ .\ \mathrm{const}_c(f(x)) \equiv \mathrm{const}_c : A \to C}\ \mathrm{def}}{\Gamma, c : C \vdash \lambda x\ .\ (\mathrm{const}_c \circ f)(x) \equiv \mathrm{const}_c : A \to C}\ \mathrm{def}}{\Gamma, c : C \vdash \mathrm{const}_c \circ f \equiv \mathrm{const}_c : A \to C}\ \eta$$

□

Lema 3.13.

$$\frac{\Gamma \vdash A\ \mathrm{type}\qquad \Gamma \vdash g : B \to C}{\Gamma, b : B \vdash g \circ \mathrm{const}_b \equiv \mathrm{const}_{g(b)} : A \to C}$$

Demostración.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Gamma \vdash g : B \to C}{\Gamma, z : B \vdash g(z) : C}\ \mathrm{ev}}{\Gamma, z : B, x : A \vdash g(z) : C}\qquad \Gamma \vdash A\ \mathrm{type}}{\Gamma, z : B, x : A \vdash g(z) : C}\ \mathrm{W}\qquad \dfrac{\dfrac{\Gamma \vdash A\ \mathrm{type}}{\Gamma, b : B \vdash \mathrm{const}_b : A \to B}\ \mathrm{ev}}{\Gamma, b : B, x : A \vdash \mathrm{const}_b(x) : B}\ \mathrm{ev}}{\Gamma, b : B, x : A \vdash \mathrm{const}_b(x) \equiv b : B}\ \mathrm{Obs\ 9}}{\Gamma, b : B, x : A \vdash g(\mathrm{const}_b(x)) \equiv g(b) : C}}{\Gamma, b : B \vdash \lambda x\ .\ g(\mathrm{const}_b(x)) \equiv \lambda x\ .\ g(b) : A \to C}\ \lambda}{\Gamma, b : B \vdash \lambda x\ .\ (g \circ \mathrm{const}_b)(x) \equiv \lambda x\ .\ g(b) : A \to C}\ \mathrm{def}}{\Gamma, b : B \vdash \lambda x\ .\ (g \circ \mathrm{const}_b)(x) \equiv \mathrm{const}_g(b) : A \to C}\ \mathrm{def}}{\Gamma, b : B \vdash g \circ \mathrm{const}_b \equiv \mathrm{const}_g(b) : A \to C}\ \eta$$

□

3.3.2. Tipos inductivos y coincidencia de patrones

Martin-Löf en [4] presenta un análisis desde el punto de vista de la teoría de la prueba sobre la teoría intuicionista de las definiciones inductivas generalizadas aplicadas de forma iterada un número finito de veces. En este análisis exhibe cómo una reformulación de estas reglas desemboca en una extensión del Haupstatz de Gentzen a la teoría de definiciones inductivas iteradas.[5] Justamente la teoría de definiciones inductivas iteradas es un predecesor de la teoría de tipos dependientes, tema que nos ocupa en este escrito.

La presentación moderna y relativamente madura de las ideas anteriormente descritas por Per Martin-Löf se presenta de forma intuitiva a continuación.

[5]El Haupstatz de Gentzen también es conocido como **el teorema de corte-eliminación**. Este teorema tiene implicaciones fuertes, como permitir mostrar la inconsistencia de un sistema, todo término de prueba se reduce a su forma normal en un número finito de pasos, o ser la justificación rigurosa de la correctud sobre introducir lemas auxiliares en la demostración de un teorema. En breve, la regla de corte-eliminación estipula:

$$\frac{\Gamma, A \vdash B\qquad \Delta \vdash A}{\Delta, \Gamma \vdash B}$$

Y el Haupstatz afirma que en el cálculo de secuentes toda prueba sin la regla de corte pero con una regla adicional (llamada mix) se puede reescribir como una prueba sin la regla adicional [3].

Podemos pensar a los tipos inductivos como aquellos que son libremente generados por sus términos canónicos. Supongamos que $\mathfrak{I}$ es un tipo dependiente con los siguientes términos canónicos:

$$\frac{}{x_0 : \mathfrak{I}} \qquad \frac{}{\kappa : \mathfrak{I} \times \mathfrak{I} \to \mathfrak{I}}$$

Siguiendo la intuición de que los tipos inductivos se pueden pensar como libremente generados por sus términos canónicos, entonces $x : \mathfrak{I}$ es de la forma $x \equiv x_0 : \mathfrak{I}$, o $x \equiv \kappa(s,t) : \mathfrak{I}$ con algunos otros $s, t : \mathfrak{I}$.

Supongamos ahora que tenemos el siguente juicio:

$$\Gamma, x : \mathfrak{I} \vdash \mathfrak{D}(x) \text{ type}$$

La pregunta es ahora ¿cómo producimos un elemento del tipo $\mathfrak{D}$ el cual depende de $\mathfrak{I}$? Por lo que se ha comentado ya en secciones anteriores, claramente la deducción deberá tener al menos una forma parecida a la siguiente:

$$\frac{\begin{array}{c}\Gamma, x : \mathfrak{I} \vdash \mathfrak{D}(x) \\ \vdots \end{array}}{\Gamma, ? \vdash ? : \mathfrak{D}(x)}$$

Retomando la interpretación lógica de dichas reglas, nuestro objetivo es producir una demostración de $\mathscr{D}(x)$ para cualesquiera $x : \mathfrak{I}$. Continuando operando bajo esta idea de ser $\mathfrak{I}$ un tipo libremente generado por sus términos canónicos, es claro entonces que al conocer explicitamente las formas que toma cualquier $x : \mathfrak{I}$ podemos aprovechar esta estructura para generar los términos que nos son de interés. Tomando un ejemplo menos abstracto, consideremos la siguiente definición del tipo de listas de un tipo A:

$$\frac{\Gamma \vdash A \text{ type}}{\Gamma \vdash \text{List}_A \text{ type}} \qquad \frac{\Gamma \vdash A \text{ type}}{\Gamma \vdash \varnothing : \text{List}_A} \qquad \frac{\Gamma \vdash x : A \qquad \Gamma \vdash xs : \text{List}_A}{\Gamma \vdash x.xs : \text{List}_A}$$

Si tenemos un tipo dependiente de List_X para cualquier x type, entonces nuestra regla de eliminación debería ser capáz de lidiar con cualesquiera forma que tome un elemento de List_X pues dicho tipo está indexado sobre todos estos elementos, de modo que las instancias del esquema de regla de eliminación presentada anteriormente deben ser consideradas:

$$\frac{\begin{array}{c}\Gamma, \varnothing : \text{List}_X \vdash \mathfrak{D}(\varnothing) \text{ type} \\ \vdots \end{array}}{\Gamma, \varnothing : \text{List}_X \vdash \rho(\varnothing) : \mathfrak{D}(\varnothing)} \qquad \frac{\begin{array}{c}\Gamma, xs : \text{List}_X \vdash \mathfrak{D}(xs) \text{ type} \\ \Gamma \vdash x : X \\ \vdots \end{array}}{\Gamma, x.xs : \text{List}_X \vdash \zeta(x.xs) : \mathfrak{D}(x.xs)}$$

con ρ y ζ términos capaces de transformar elementos de List_X en términos de $\mathfrak{D}$. De esta forma, si los constructores de $\mathfrak{D}$ type son satisfechos por los elementos canónicos de List_X entonces podemos asegurar que para cualquier forma $xs : \text{List}_X$ existirá un término $c(\rho, \zeta, x) : \mathfrak{D}(xs)$.

Finalmente, una regla de cómputo deberá regresar la demostración de un caso particular de la forma que tome el tipo inductivo dada la demostración del caso general. En nuestro ejemplo anterior esto se vería como la siguiente regla:

$$\frac{\begin{array}{c}\Gamma, xs : \text{List}_X \vdash \mathfrak{D}(xs) \text{ type} \\ \Gamma, \varnothing : \text{List}_X \vdash \rho(\varnothing) : \mathfrak{D}(\varnothing) \\ \Gamma, x.xs : \text{List}_X \vdash \zeta(x.xs) : \mathfrak{D}(x.xs) \end{array}}{\Gamma, xs : \text{List}_X \vdash c(\rho, \zeta, \varnothing) \equiv \rho(\varnothing) : \mathfrak{D}(\varnothing)} \qquad \frac{\begin{array}{c}\Gamma, xs : \text{List}_X \vdash \mathfrak{D}(xs) \text{ type} \\ \Gamma, \varnothing : \text{List}_X \vdash \rho(\varnothing) : \mathfrak{D}(\varnothing) \\ \Gamma, x.xs : \text{List}_X \vdash \zeta(x.xs) : \mathfrak{D}(x.xs) \end{array}}{\Gamma, xs : \text{List}_X \vdash c(\rho, \zeta, \varnothing) \equiv \zeta(x.xs) : \mathfrak{D}(x.xs)}$$

Tipo unitario

Tipo vacío

Tipo producto

Tipo co-producto

3.3.3. Pares dependientes

Los pares dependientes fueron concebidos originalmente por Per Martin-Löf como un análogo a la unión disjunta de una familia de conjuntos [7]. Por lo tanto, en la teoría de tipos dependientes el principio de considerar "pares" de elementos de una familia de conjuntos conservando un elemento que apunta de qué conjunto viene se preserva. La representación usual de la unión disjunta de conjuntos es

$$\coprod_{i\in I} A_i = \bigcup_{i\in I}\{(i,x) : x \in A_i\}$$

Notemos que si $w \in \coprod A_i$, entonces es porque **existe** un índice $i \in I$ para el cual existe un x tal que $x \in A_i$. Fundamentalmente w cuenta entonces con la siguiente información:

- un índice que evidencía de la presencia de un elemento de A_i en la suma disjunta y,
- un elemento concreto que pertenece a A_i y queda encapsulado por $w = \langle i, x\rangle$.

Así pues, es claro que tanto x como A_i dependen de i para exhibir su presencia dentro de la unión disjunta. Si sustituimos a la familia de conjuntos $\{A_i\}_{i\in I}$ por su homólogo en la teoría de tipos,

$$i : I \vdash x : A(i)$$

lo anterior se reflejaría en que tal estructura, de existir, sus elementos canónicos deberían tener la información análoga:

- un índice $i : I$ que evidencía de la presencia de un término de $A(i)$ en la estructura y,
- un elemento concreto x que pertenece a $A(i)$ y queda encapsulado por $\langle i : I, x : A(i)\rangle$.

La existencia de esta estructura se postula, y se denominan **pares dependientes**, **tipo** Σ, **co-producto dependiente** y **tipo suma dependiente** y usualmente se denota por la letra griega mayúscula Σ. Existen al menos dos formas de presentar al tipo de los pares dependientes: como fue concebido por Martin-Löf [6], y como un tipo inductivo. En este escrito se empleará la forma inductiva de presentar a los tipos, por lo que la exposición seguirá de forma cercana a la que hace Egbert Rijke en [9] y en mayor o menor medida a aquella seguida por el Programa para los Fundamentos Univalentes [8].

cuales son verdaderamente las diferencias entre ambas exposiciones?

Definición 3.7 (El tipo de los pares dependientes)**.** Dada una familia de tipos B sobre A, el tipo de pares dependientes se define como el tipo inductivo $\sum_{(x:A)} B(x)$

$$\frac{\Gamma, x : A \vdash B(x)\ \text{type}}{\Gamma \vdash \sum_{(x:A)} B(x)}\ \Sigma$$

acompañado de una función de emparejamiento:

$$\text{pair} : \prod_{(x:A)} \left(B(x) \to \sum_{(y:A)} B(y) \right)$$

o equivalentemente

$$\frac{\Gamma, x : A \vdash b : B(x)}{\text{pair}(x,b) : \sum_{(y:A)} B(y)}$$

y el principio de inducción:

$$\frac{\Gamma, p : \sum_{(x:A)} B(x) \vdash P(p)\ \text{type}}{\text{ind}_\Sigma : \left(\prod_{(x:A)} \prod_{(y:B(x))} P(\text{pair}(x,y))\right) \to \left(\prod_{(z:\sum_{(x:A)} B(x))} P(z)\right)}$$

tal que satisface la regla de cómputo

$$\text{ind}_\Sigma(g, \text{pair}(x,y)) \equiv g(x,y)$$

Escribiremos $\langle x, y\rangle$ en lugar de $\text{pair}(x,y)$.

Realmente dependen sobre el primer elemento en el pr dependien solo es sobre el tipo del segundo elemento no tambie puede ser sobre el mis segundo e mento? i.e $\langle x, y(x)\rangle$ y no $\langle x, y\rangle$ e general?

quien es g

Observación 10. Alternativamente, una definición de una función dependiente

$$f : \prod_{(z:\sum_{(x:A)} B(x))} P(z)$$

por inducción utilizando una función

$$g : \prod_{(x:A)} \prod_{y:B(y)} P((x,y))$$

puede ser presentada por coincidencia de patrones de la siguiente forma:

$$f(\text{pair}(x,y)) := g(x,y)$$

Observación 11. Si queremos definir una función

$$f : \prod_{(z:\sum_{(x:A)} B(x))} P(z)$$

por Σ-inducción, entonces debemos asumir un par $\langle x, y\rangle$ consistente de $x : A$ y $y : B(x)$ con la meta en mente de construir un elemento del tipo $P(x,y)$. El principio de inducción de los tipos Σ es por lo tanto el converso a la operación de currying, dado por la función

$$\text{ev-pair} : \left(\prod_{(z:\sum_{(x:A)} B(x))} P(z)\right) \to \left(\prod_{(x:A)} \prod_{(y:B(x))} P(x,y)\right)$$

dada por $f \mapsto \lambda x.\lambda y.f(x,y)$. El principio de inducción ind_Σ es por lo tanto también conocido como la operación de uncurrying.

Definición 3.8. Dados $\Gamma \vdash A$ type, $\Gamma x : A \vdash B(x)$ Definimos la primera y segunda proyección por el principio de inducción del tipo Σ como a continuación:
primera proyección

$$\text{pr}_1 : \left(\sum_{(x:A)} B(x)\right) \to A$$
$$\text{pr}_1\langle x, y\rangle \equiv x$$

segunda proyección

$$\text{pr}_2 : \prod_{(p:\sum_{(x:A)} B(x))} B(\text{pr}_1(p))$$
$$\text{pr}_2\langle x, y\rangle \equiv y$$

Un caso especial del tipo Σ ocurre cuando B es una familia constante sobre A, es decir, cuando B solo es un tipo debilitado por A. En este caso, el tipo $\sum_{(x:A)} B$ es el tipo de los pares ordenados *comunes y corrientes* donde $x : A$ y $y : B$ de modo que el tipo de y no depende de x.

Definición 3.9. Si A y B son tipos en un contexto Γ, definimos su producto (cartesiano) $A \times B$ como

$$\frac{\dfrac{\Gamma \vdash A \text{ type} \qquad \Gamma \vdash B \text{ type}}{x : A \vdash B \text{ type}}}{A \times B := \sum_{(x:A)} B(x)}\ \text{W}$$

En tanto que el producto se define mediante el tipo Σ, se tiene entonces que este tipo también satisface la regla de inducción de los tipos Σ. Para este caso particular la regla de inducción esatablece que para cualquier familia P sobre $A \times B$ existe una función

$$\text{ind}_\times : \left(\prod_{(x:A)} \prod_{(y:B)} P(x,y)\right) \to \left(\prod_{(z:A\times B)} P(z)\right)$$

tal que satisface la regla de cómputo

$$\text{ind}_\times(g, \langle x, y\rangle) \equiv g(x,y)$$

Así también las proyecciones se definen de manera análoga para el tipo producto.

[6] La presentación clásica del tipo Σ se puede encontrar en [8] y en [7]

3.3.4. Breve comentario sobre la correspondencia Curry-Howard-Lambek

decidir si esto sí va aquí o ant hmm

3.3.5. Tipos de identidad

3.3.6. Universos de tipos

acá iría la parte de lógica pro posicional en agda u (ejercicios la sección

La teoría de conjuntos permite tener conjuntos cuyos elementos son conjuntos. De manera similar, la teoría de tipos ofrece un mecanismo para definir tipos cuyos términos son también tipos. Los **universos** de tipos consisten de un tipo $\mathfrak{U}$ junto con una familia de tipos $\mathfrak{T}$ definida sobre $\mathfrak{U}$. La idea es pensar que dado $X : \mathfrak{U}$, $\mathfrak{T}(X)$ es una "*interpretación*" externa a $\mathfrak{U}$ de X.

Definición 3.10 (Universo)**.** Un universo es un tipo $\mathfrak{U}$ junto con una familia $\mathfrak{U}$ sobre $\mathfrak{U}$ llamada *familia universal* tales que satisfacen los siguientes axiomas:

- $\mathfrak{U}$ es tal que existe
$$\check{\Pi} : \prod_{X:\mathfrak{U}} (\mathfrak{T}(X) \to \mathfrak{U}) \to \mathfrak{U}$$
tal que satisface la siguiente igualdad de juicio:
$$\mathfrak{T}(\Pi(\check{X}, Y)) \equiv \prod_{x:\mathfrak{T}(X)} \mathfrak{T}(Y(x))$$
para cualesquiera $X : \mathfrak{U}$ y $Y : \mathfrak{T}(X) \to \mathfrak{U}$.

- $\mathfrak{U}$ es tal que existe
$$\check{\Sigma} : \prod_{X:\mathfrak{U}} (\mathfrak{T}(X) \to \mathfrak{U}) \to \mathfrak{U}$$
tal que satisface la siguiente igualdad de juicio:
$$\mathfrak{T}(\Sigma(X, Y)) \equiv \sum_{x:\mathfrak{T}(X)} \mathfrak{T}(Y(x))$$
para cualesquiera $X : \mathfrak{U}$ y $Y : \mathfrak{T}(X) \to \mathfrak{U}$.

- $\mathfrak{U}$ es tal que existe
$$\check{I} : \prod_{X:\mathfrak{U}} \mathfrak{T}(X) \to (\mathfrak{T}(X) \to \mathfrak{U})$$
tal que satisface la siguiente igualdad de juicio:
$$\mathfrak{T}(\check{I}(X, x, y)) \equiv (x = y)$$
para cualesquiera $X : \mathfrak{U}$ y $x, y : \mathfrak{T}(X)$.

- $\mathfrak{U}$ es tal que existe
$$\check{+} : \mathfrak{U} \to \mathfrak{U} \to \mathfrak{U}$$
tal que satisface la igualdad de juicio
$$\mathfrak{T}(X \check{+} Y) \equiv \mathfrak{T}(X) + \mathfrak{T}(Y)$$

- $\mathfrak{U}$ es tal que existen $\check{\mathbb{1}}, \check{\mathbb{0}}, \check{\mathbb{N}}$ tales que satisfacen las siguientes igualdades de juicio:
$$\mathfrak{T}(\check{\mathbb{1}}) \equiv \mathbb{1}$$
$$\mathfrak{T}(\check{\mathbb{0}}) \equiv \mathbb{0}$$
$$\mathfrak{T}(\check{\mathbb{N}}) \equiv \mathbb{N}$$

This catalog of 23 entries was generated by calibre on Wednesday, 21. December 2022 14:21

Referencias

[1] Haskell B. Curry. Some philosophical aspects of combinatory logic. In Jon Barwise, H. Jerome Keisler, and Kenneth Kunen, editors, *The Kleene Symposium*, volume 101 of *Studies in Logic and the Foundations of Mathematics*, pages 85–101. Elsevier, 1980.

[2] Joachim Lambek and Philip J. Scott. *Introduction to Higher Order Categorical Logic*. Cambridge University Press, 1986.

[3] Paolo Mancosu and Richard Zach. *An Introduction to Proof Theory: Normalization, Cut-Elimination, and Consistency Proofs*, volume 1. Oxford University Press, Oct 2021.

[4] Per Martin-Löf. *Hauptsatz for the Intuitionistic Theory of Iterated Inductive Definitions*, page 179–216. Elsevier, 1971.

[5] Per Martin-Löf. An intuitionistic theory of types: Predicative part. In H.E. Rose and J.C. Shepherdson, editors, *Logic Colloquium '73*, volume 80 of *Studies in Logic and the Foundations of Mathematics*, pages 73–118. Elsevier, 1975.

[6] Per Martin-Löf. 127An intuitionistic theory of types. In *Twenty Five Years of Constructive Type Theory*. Oxford University Press, 10 1998.

[7] Per Martin-Löf and Giovanni Sambin. *Intuitionistic Type Theory*, volume 1. Bibliopolis, Dec 1984.

[8] Univalent Foundations Program. Homotopy type theory: Univalent foundations of mathematics, Dec 100.

[9] Egbert Rijke. Introduction to homotopy type theory, Dec 100.

[10] Jean van Heijenoort. *From Frege to Gödel : a source book in mathematical logic, 1879-1931*, page 355. Source Books in the History of the Sciences. Harvard University Press, London, England, July 1990.