
Garantías de terminación en λ -cálculos polimórficos

Cristian Sottile

Universidad de Buenos Aires & CONICET & Universidad Nacional de Quilmes

7mo Día del ICC

DC, Exactas, UBA

14 de agosto de 2025



DEPARTAMENTO
DE COMPUTACION
Facultad de Ciencias Exactas y Naturales - UBA



ICC | Instituto de Ciencias
de la Computación



</CPI>
01010101.01001110.01010001

Background

Formalismo para representar a todas las funciones computables

Variables x

Funciones λ

Aplicaciones @

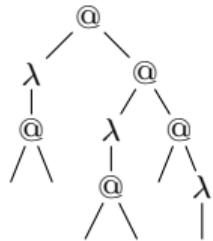
Formalismo para representar a todas las funciones computables

Variables x

Funciones λ

Aplicaciones @

Intuición



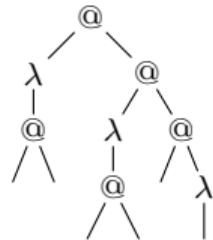
Formalismo para representar a todas las funciones computables

Variables x

Funciones λ

Aplicaciones @

Intuición



Cómputo

Transformaciones sucesivas en el árbol

(al juntar @ y λ)

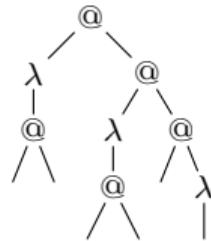
Formalismo para representar a todas las funciones computables

Variables x

Funciones λ

Aplicaciones @

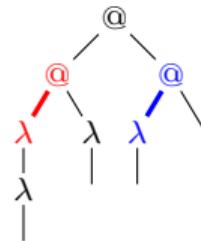
Intuición



Cómputo

Transformaciones sucesivas en el árbol

(al juntar @ y λ)



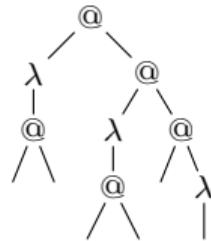
Formalismo para representar a todas las funciones computables

Variables x

Funciones λ

Aplicaciones @

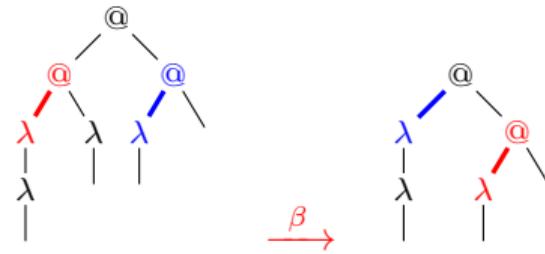
Intuición



Cómputo

Transformaciones sucesivas en el árbol

(al juntar @ y λ)



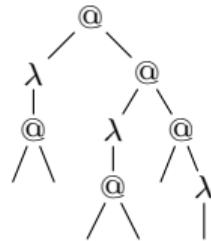
Formalismo para representar a todas las funciones computables

Variables x

Funciones λ

Aplicaciones @

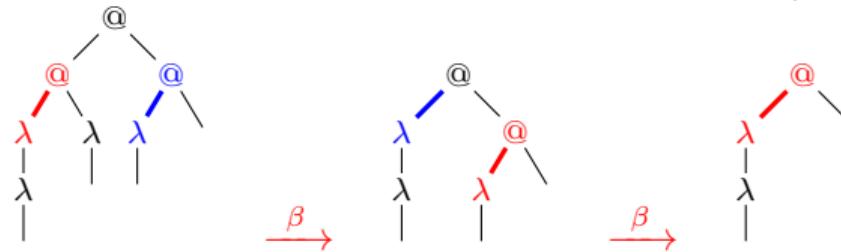
Intuición



Cómputo

Transformaciones sucesivas en el árbol

(al juntar @ y λ)



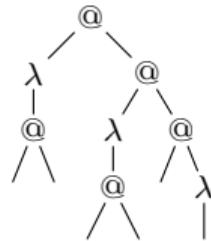
Formalismo para representar a todas las funciones computables

Variables x

Funciones λ

Aplicaciones @

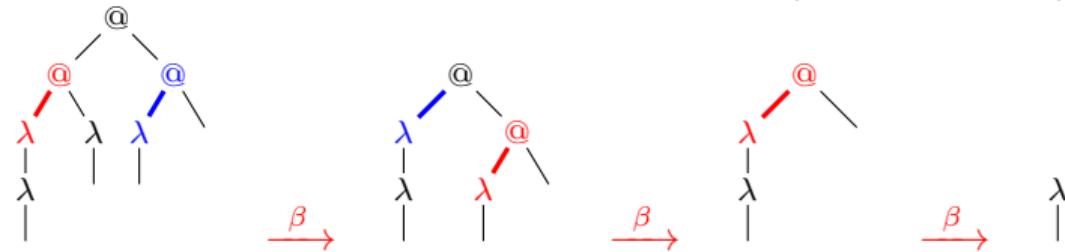
Intuición



Cómputo

Transformaciones sucesivas en el árbol

(al juntar @ y λ)



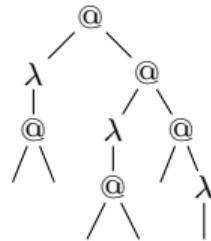
Formalismo para representar a todas las funciones computables

Variables x

Funciones λ

Aplicaciones @

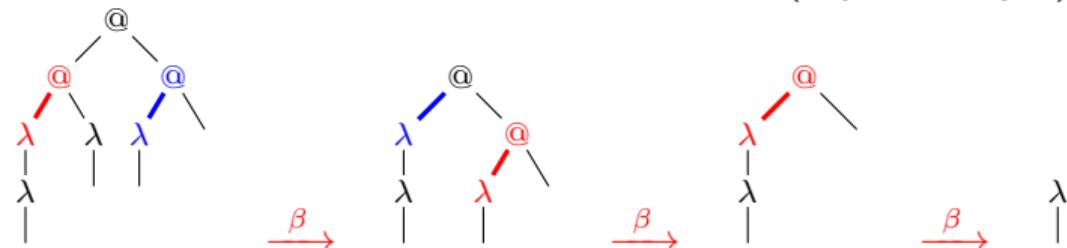
Intuición



Cómputo

Transformaciones sucesivas en el árbol

(al juntar @ y λ)

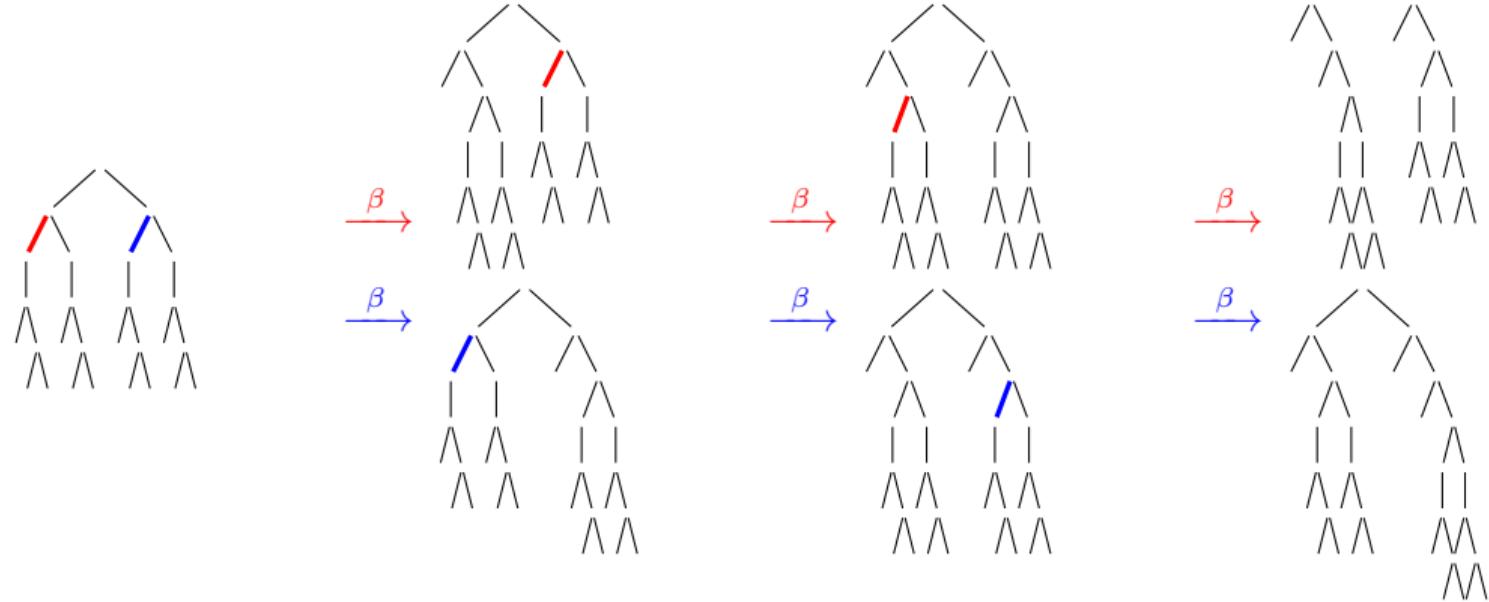


Terminación

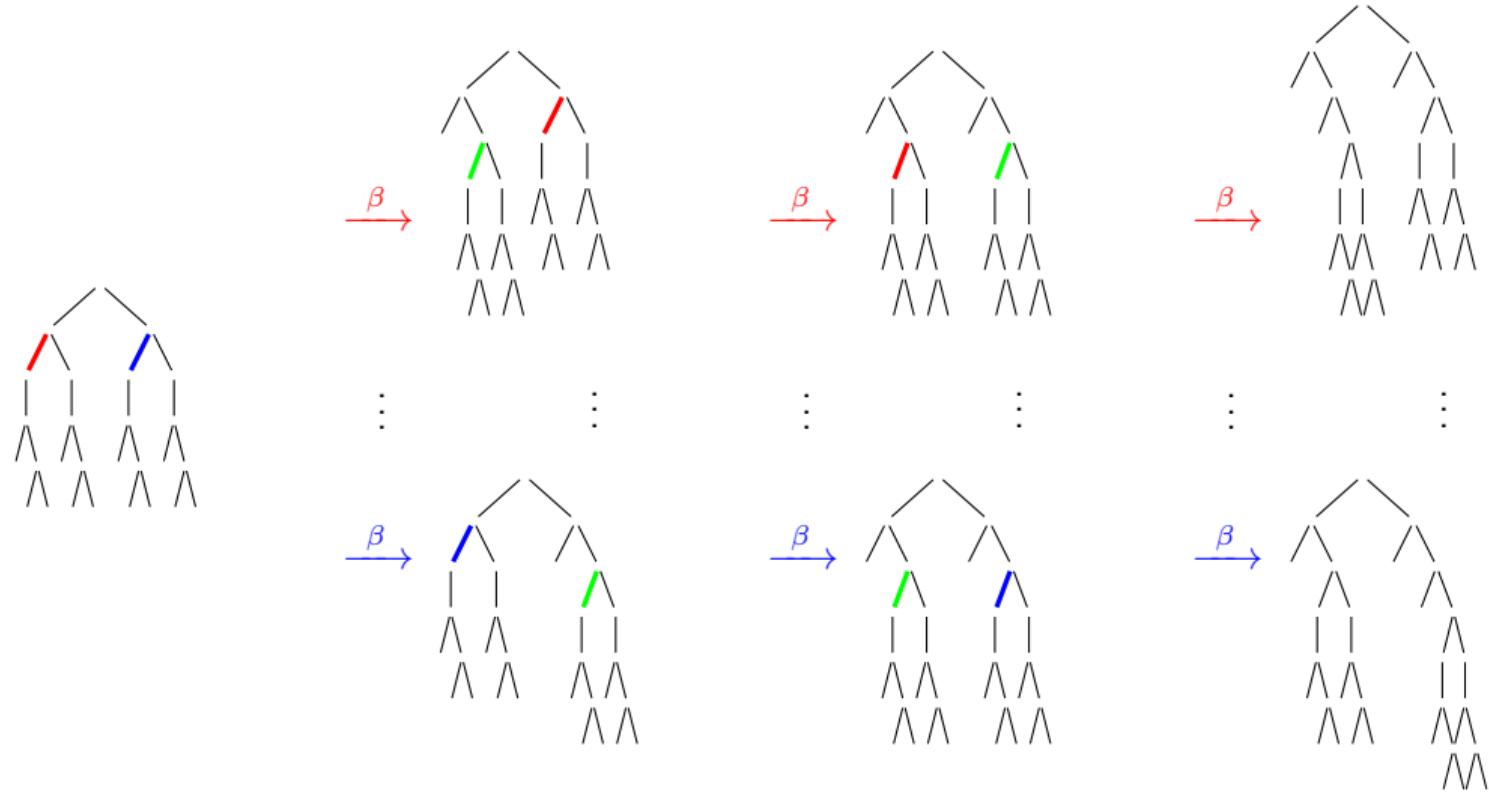
Árbol sin posibles transformaciones

(no hay @ y λ juntos)

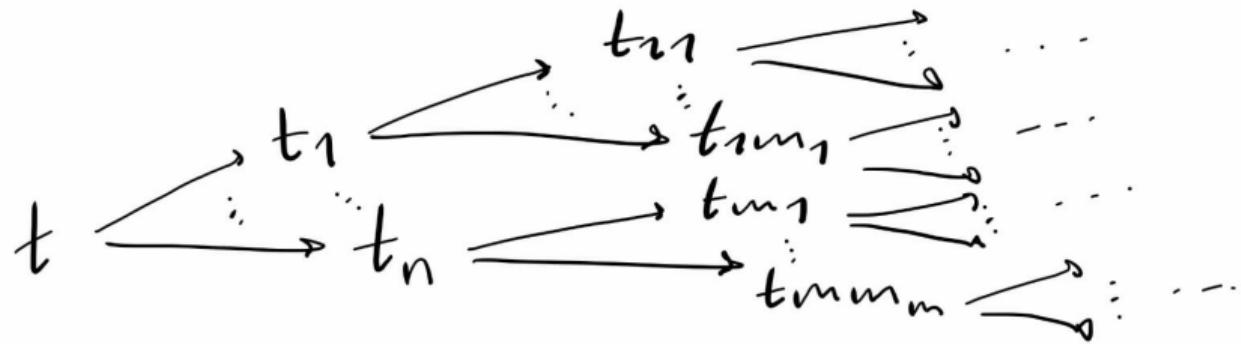
El árbol de ejecución



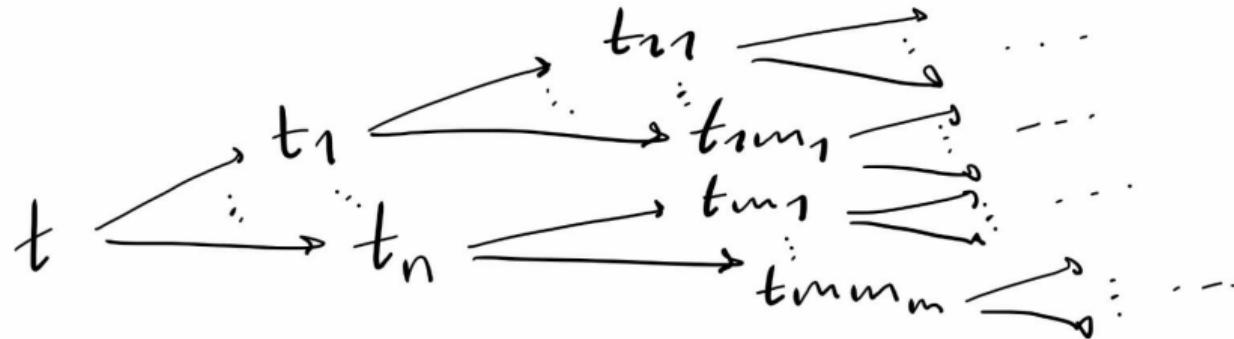
El árbol de ejecución



El árbol de ejecución

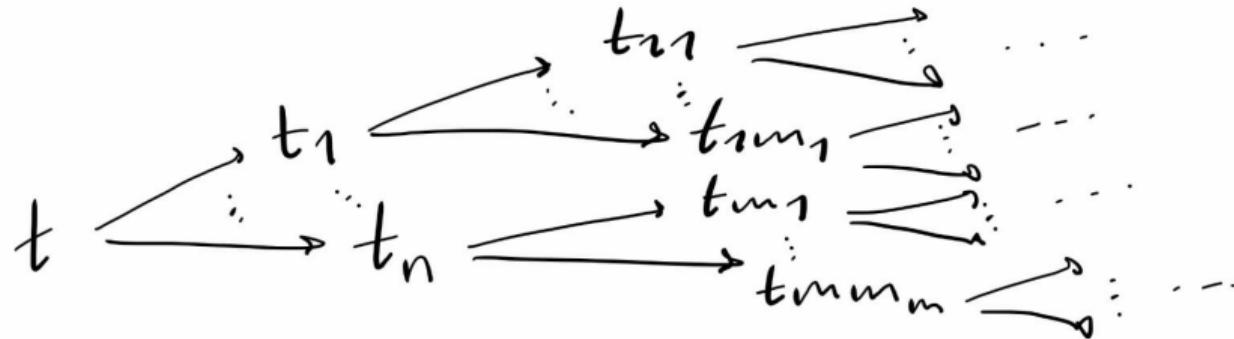


La propiedad de terminación: definición



Todas las ramas del árbol de ejecución son finitas

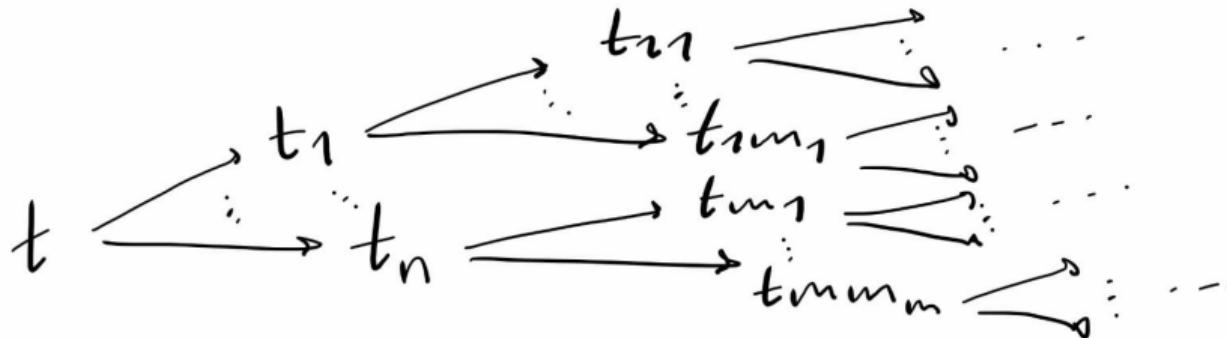
La propiedad de terminación: definición



Todas las ramas del árbol de ejecución son finitas

- ▶ Es conocida para sistemas clásicos

La propiedad de terminación: definición

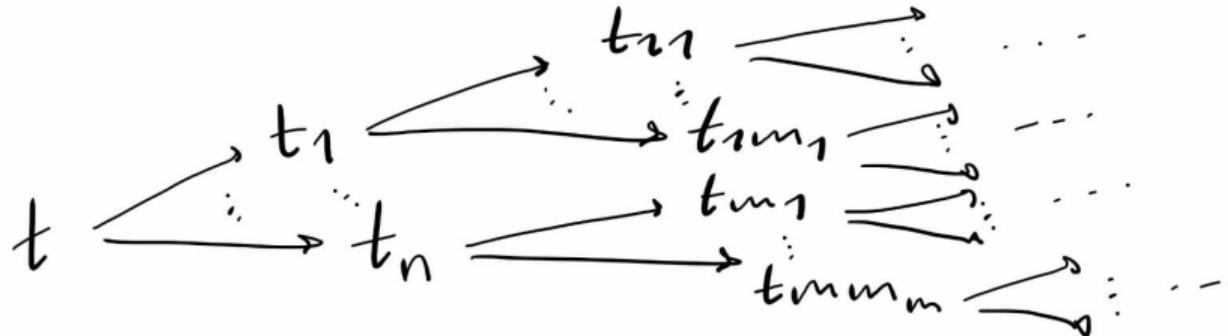


Todas las ramas del árbol de ejecución son finitas

- ▶ Es conocida para sistemas clásicos
- ▶ Es **importante** para λ -cálculos tipados

nos interesa demostrarla cuando definimos un nuevo cálculo

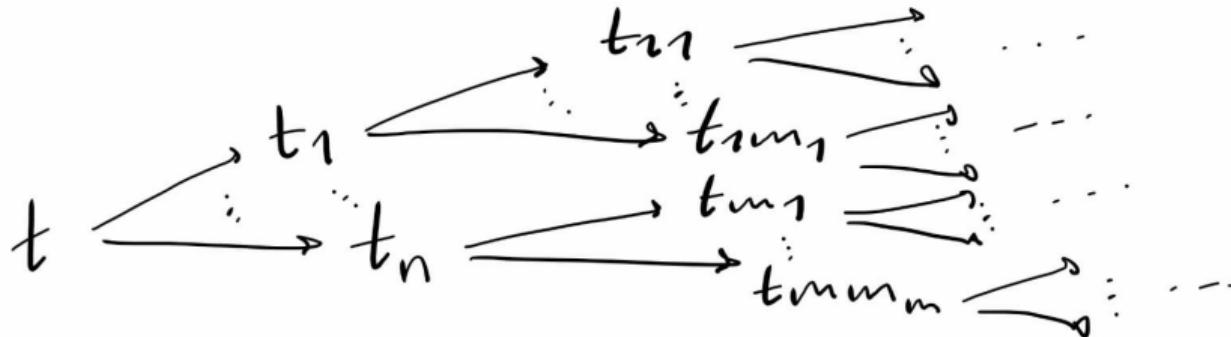
La propiedad de terminación: definición



Todas las ramas del árbol de ejecución son finitas

- ▶ Es conocida para sistemas clásicos
- ▶ Es **importante** para λ -cálculos tipados
 - nos interesa demostrarla cuando definimos un nuevo cálculo
- ▶ Es de las más **difíciles** de probar
 - nos interesa seguir encontrando formas de demostrarla

La propiedad de terminación: definición



Todas las ramas del árbol de ejecución son finitas

- ▶ Es conocida para sistemas clásicos
- ▶ Es **importante** para λ -cálculos tipados
 - nos interesa demostrarla cuando definimos un nuevo cálculo
- ▶ Es de las más **difíciles** de probar
 - nos interesa seguir encontrando formas de demostrarla
- ▶ Dos técnicas

semántica

reducibilidad

sintáctica

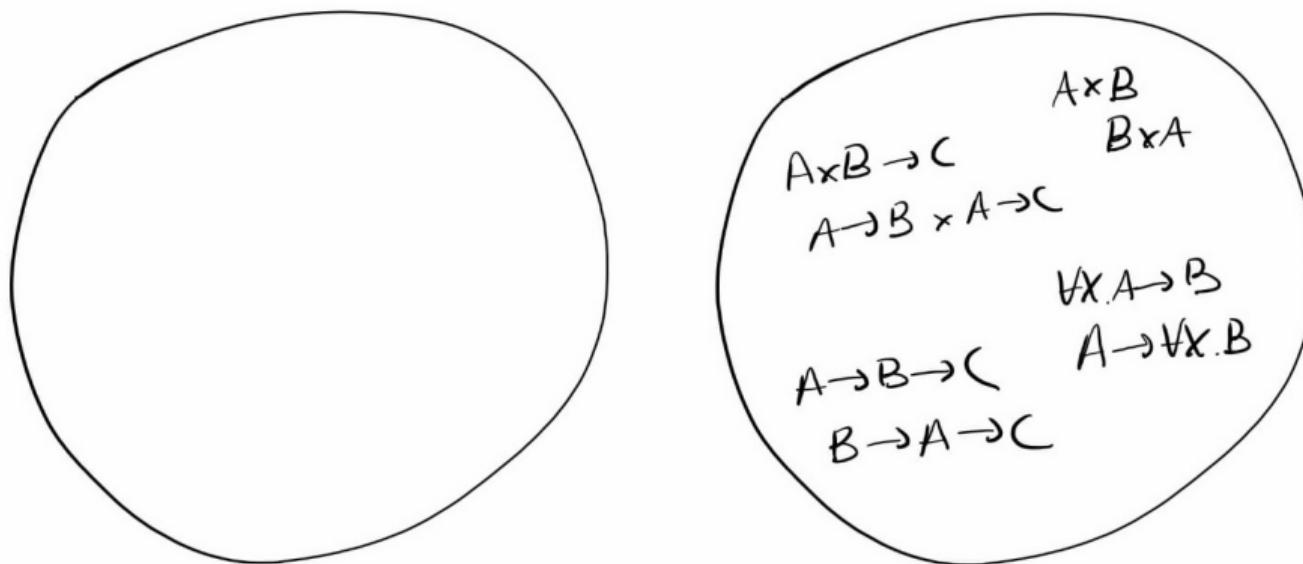
medidas decrecientes

Trabajo en progreso

con Alejandro Díaz-Caro

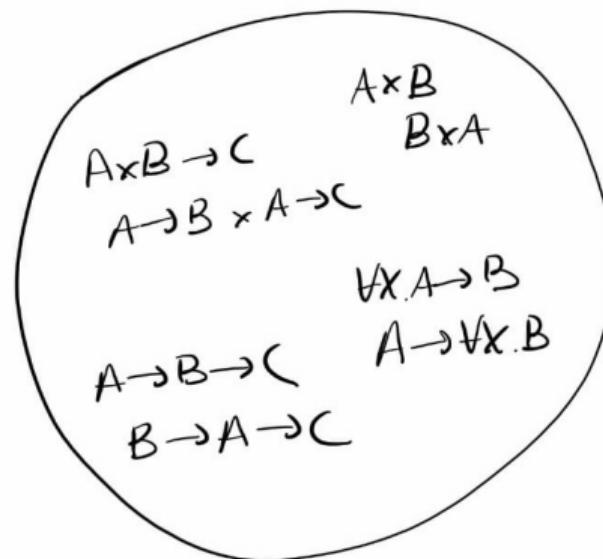
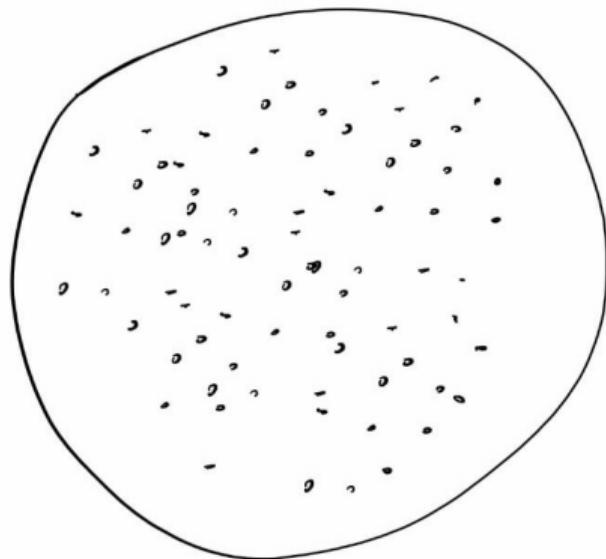
El método (semántico) de reducibilidad [Tait'67, Girard'72]

interpretación de tipos en conjuntos con buenas propiedades



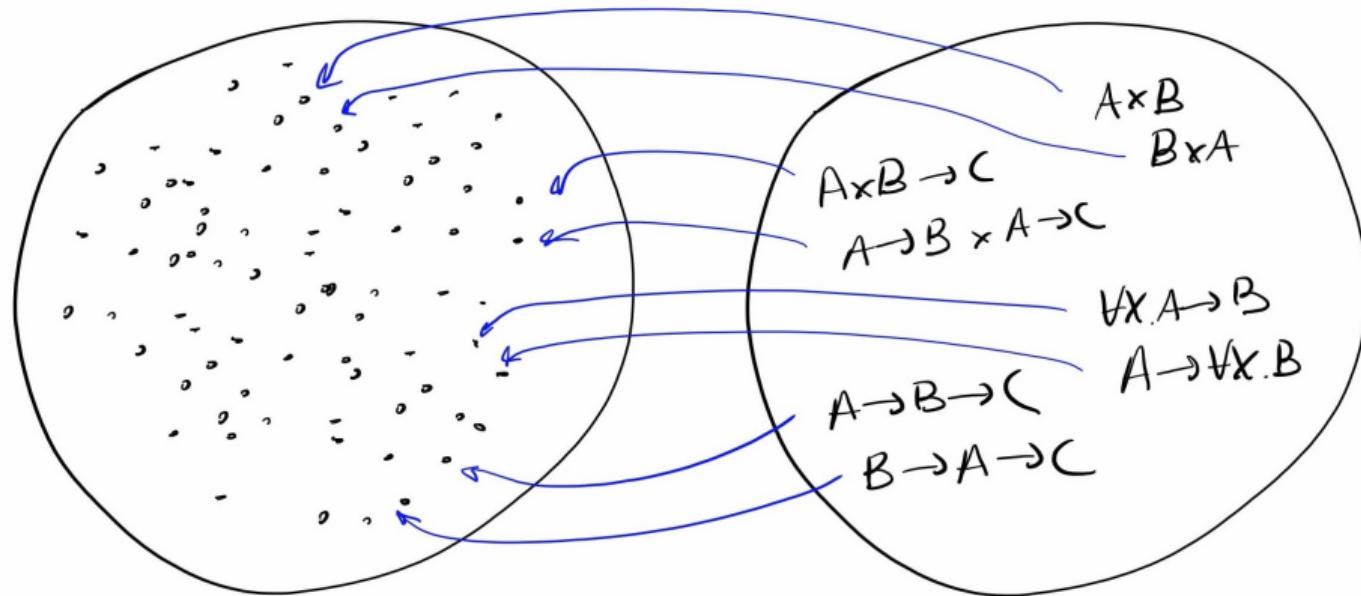
El método (semántico) de reducibilidad [Tait'67, Girard'72]

interpretación de tipos en conjuntos con buenas propiedades



El método (semántico) de reducibilidad [Tait'67, Girard'72]

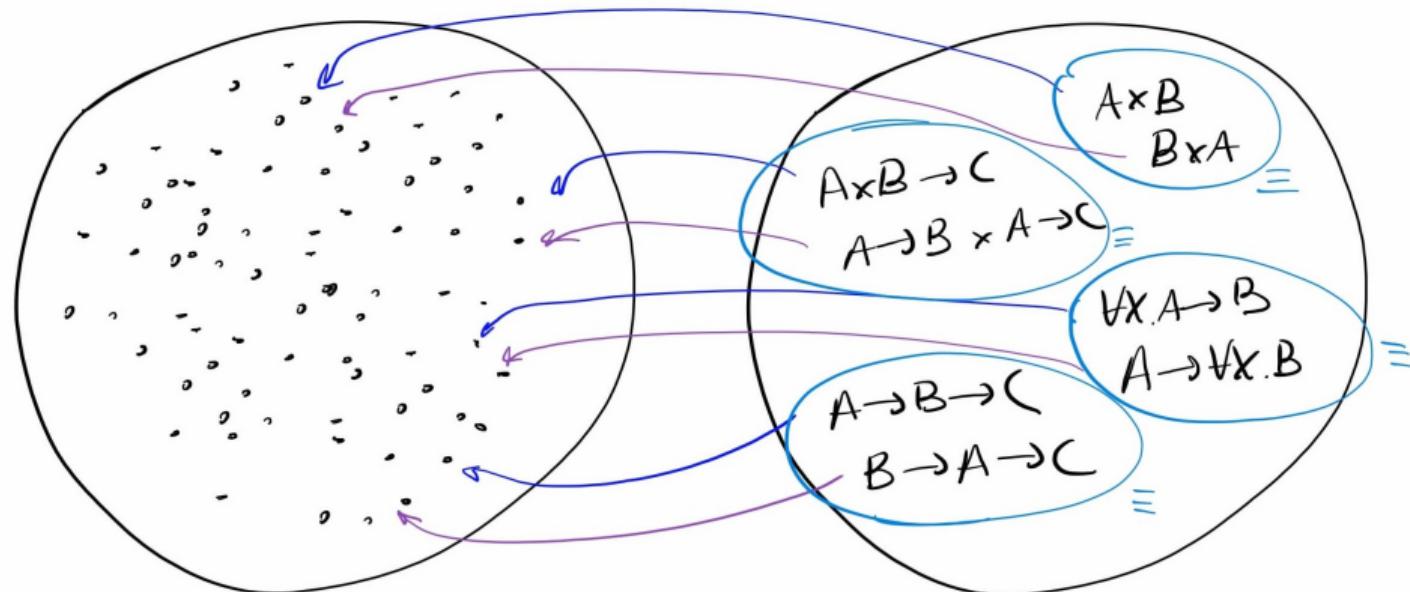
interpretación de tipos en conjuntos con buenas propiedades



Reducibilidad módulo isomorfismos [WIP con Alejandro Díaz-Caro]

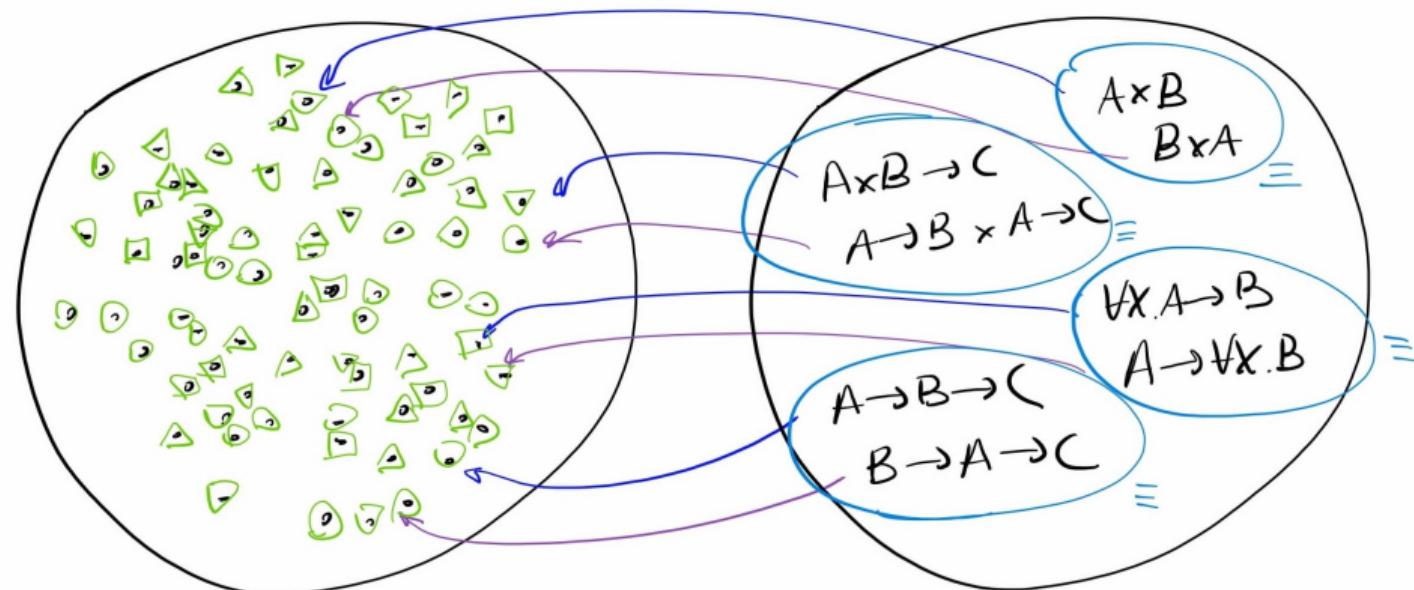
λ -cálculo polimórfico módulo isomorfismos: equivalencias en tipos y en programas

Los desafíos de terminación bajo isomorfismos



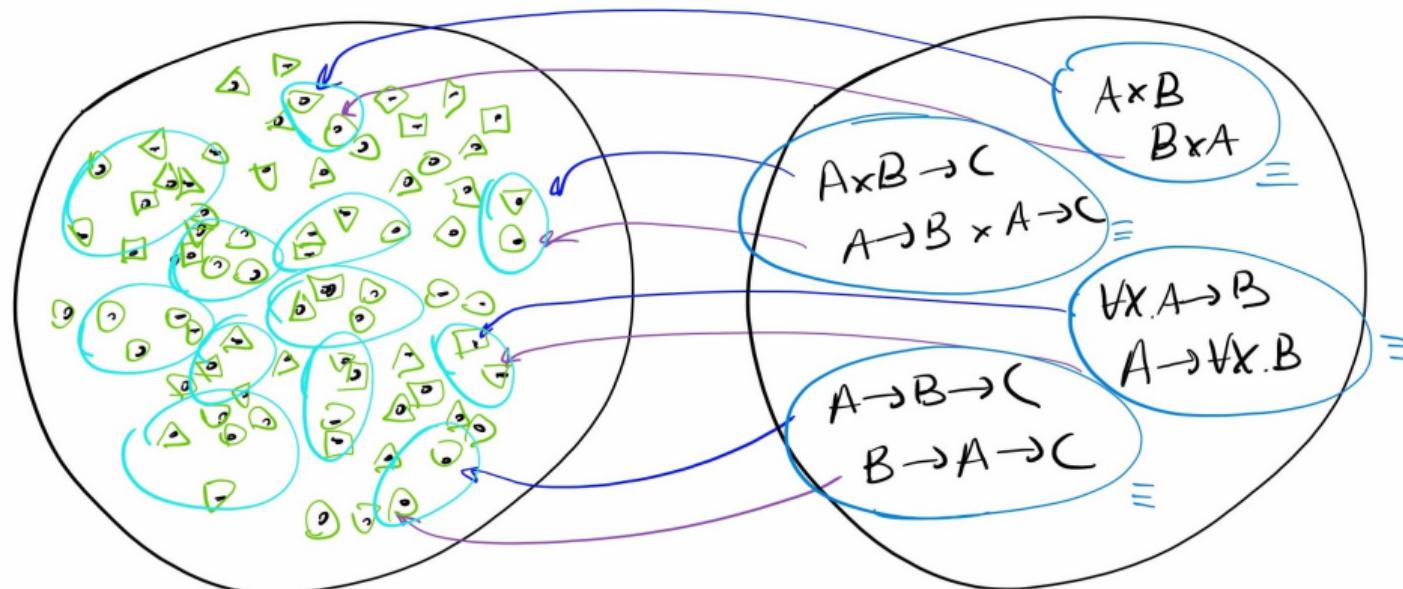
Reducibilidad módulo isomorfismos [WIP con Alejandro Díaz-Caro]

Reducibilidad à la Parigot: agregamos estructura a los candidatos



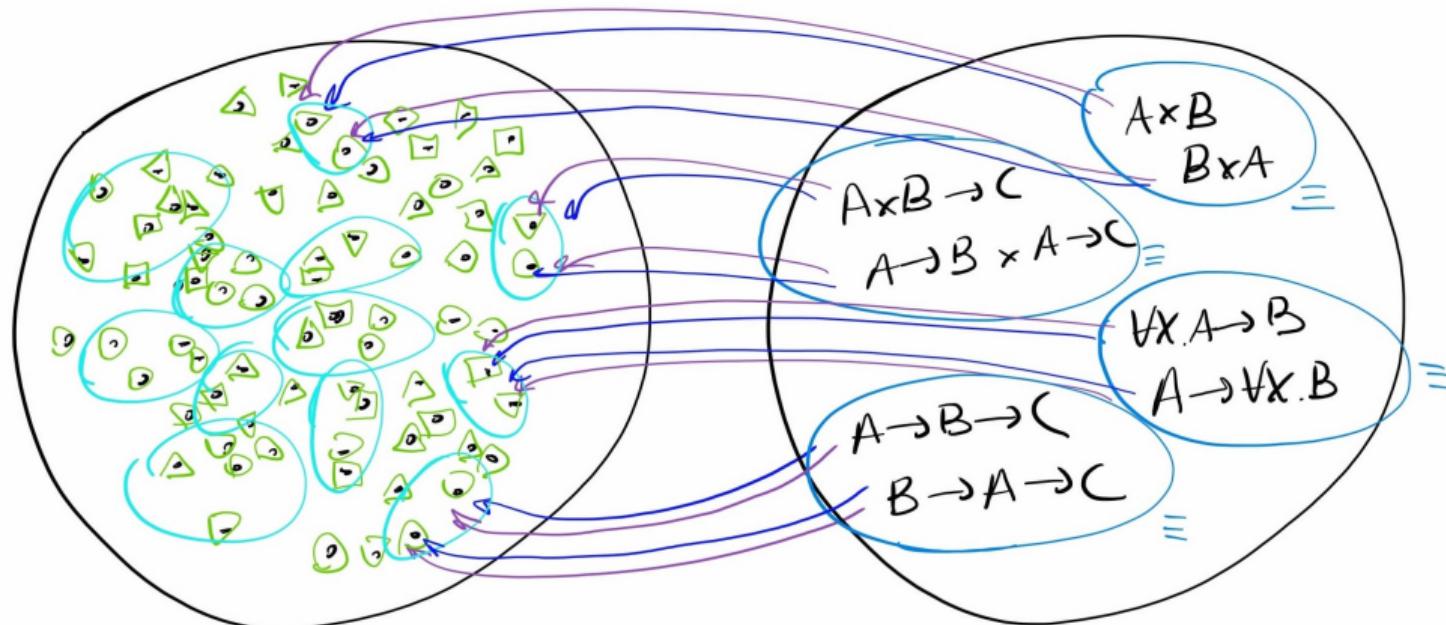
Reducibilidad módulo isomorfismos [WIP con Alejandro Díaz-Caro]

Candidatos nuevos uniendo existentes por isomorfismo



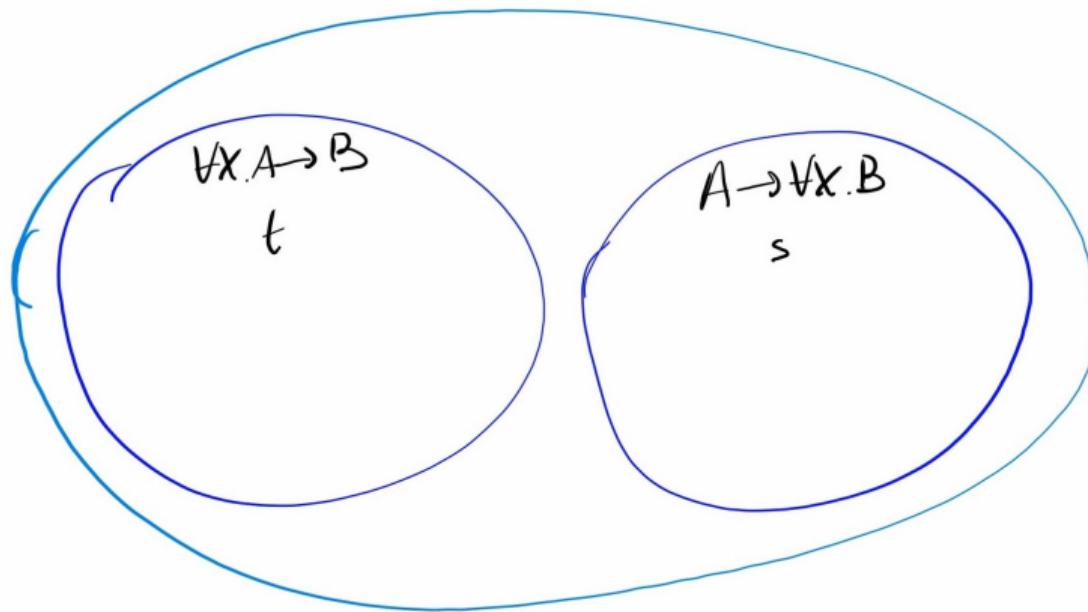
Reducibilidad módulo isomorfismos [WIP con Alejandro Díaz-Caro]

Asociación/interpretación compuesta: cómo asociar clases con clases



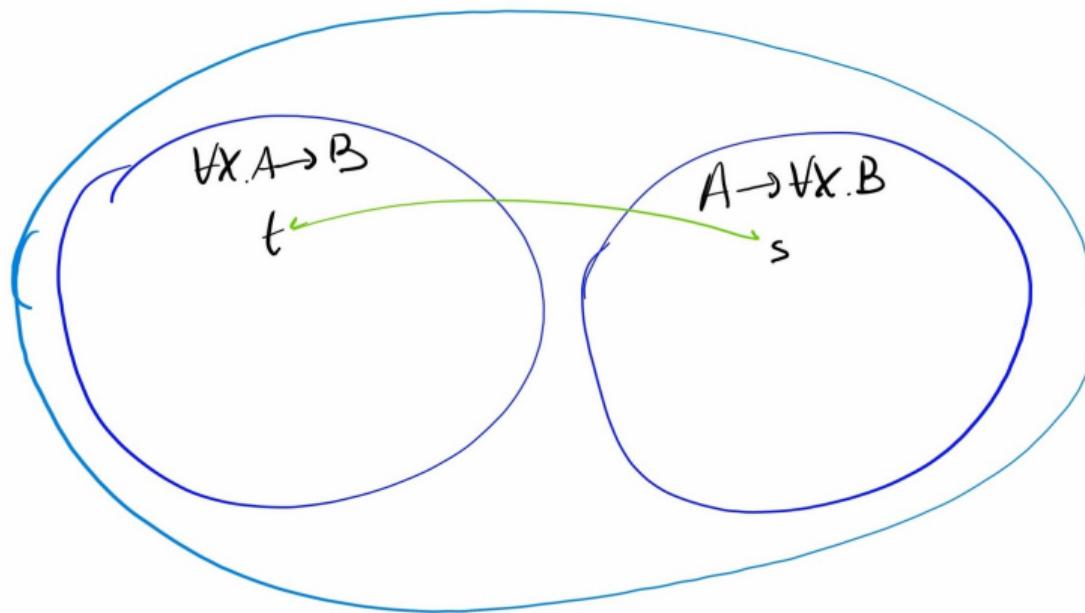
Reducibilidad módulo isomorfismos [WIP con Alejandro Díaz-Caro]

Metiéndonos en una clase de equivalencia de tipos



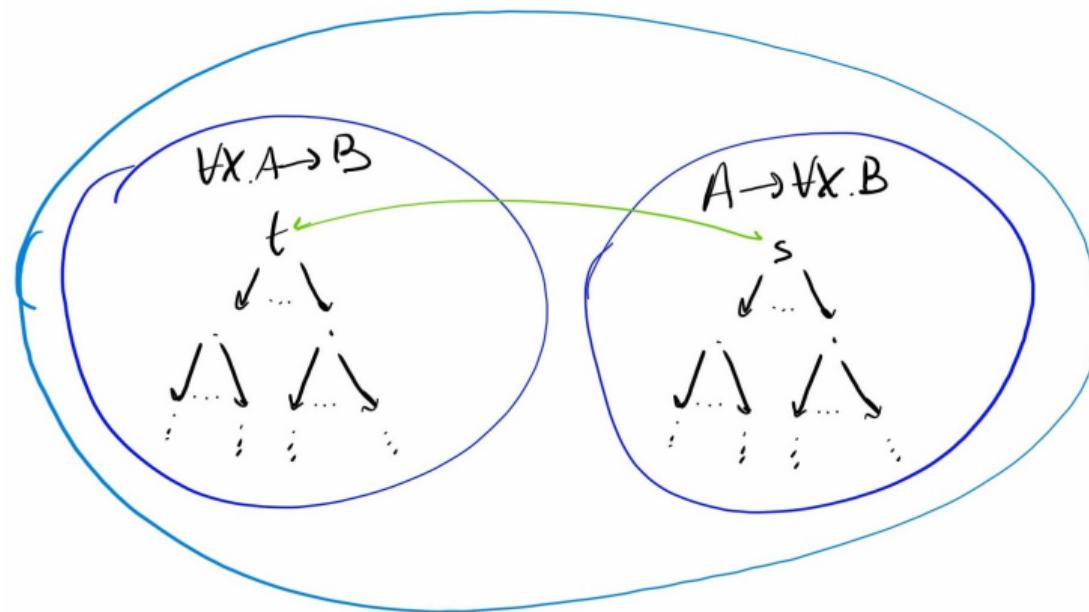
Reducibilidad módulo isomorfismos [WIP con Alejandro Díaz-Caro]

Hay programas relacionados por isomorfismo



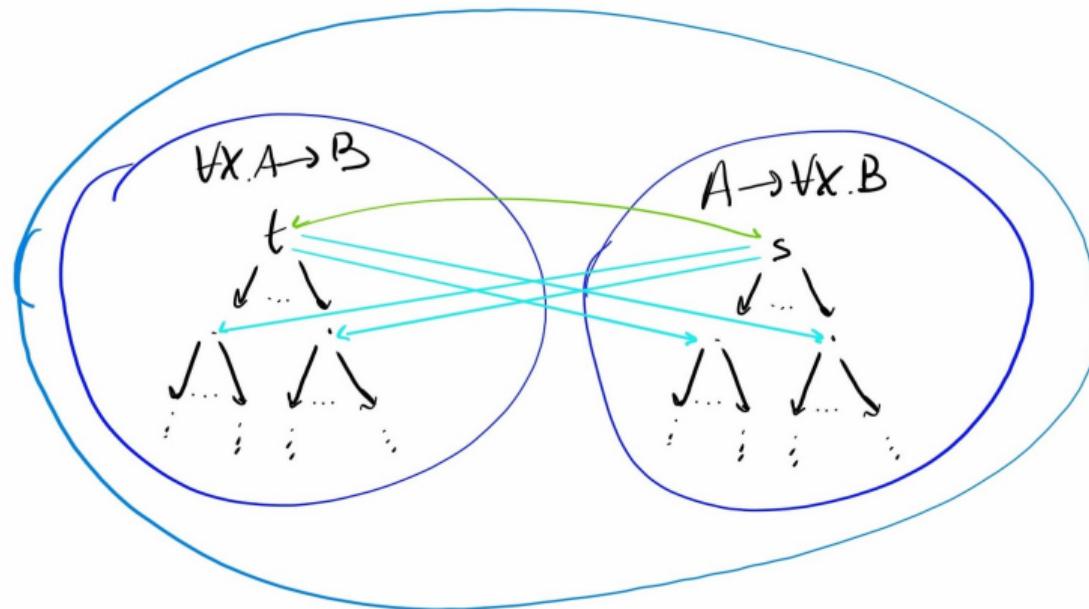
Reducibilidad módulo isomorfismos [WIP con Alejandro Díaz-Caro]

Considerando el árbol de ejecución



Reducibilidad módulo isomorfismos [WIP con Alejandro Díaz-Caro]

Ahora comparten pasos de computación



Reducibilidad módulo isomorfismos [WIP con Alejandro Díaz-Caro]

- ▶ equivalencia entre tipos ✓
- ▶ equivalencia entre términos WIP

Otro trabajo

con Pablo Barenbaum y Simona Ronchi della Rocca

El método (sintáctico) de medidas decrecientes [Gandy'80, de Vrijer'87]

función de programas en un orden bien fundado
tal que decrece con cada paso de computación

$$\# : \Lambda \rightarrow \text{Ord} \quad t \rightarrow s \implies \#(t) > \#(s)$$

Una medida decreciente para Λ_{\cap} (tipos intersección idempotente)

[Barenbaum, Ronchi della Rocca, S.'25]

- ▶ Existen medidas decrecientes
 - ▶ Kfoury & Wells'95
 - ▶ Boudol'03
- ▶ La nuestra introduce mejoras
 - ▶ dominio más simple: **números naturales**
 - ▶ es completa: **funciona para todas las ramas**

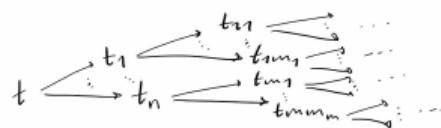
Una medida decreciente para Λ_{\cap} (tipos intersección idempotente)

[Barenbaum, Ronchi della Rocca, S.'25]

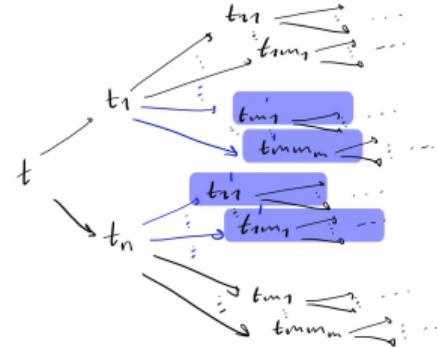
Una medida decreciente para Λ_{\cap} (tipos intersección idempotente)

[Barenbaum, Ronchi della Rocca, S.'25]

1. Enriquecemos el árbol



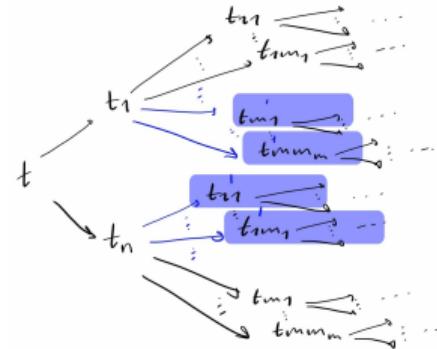
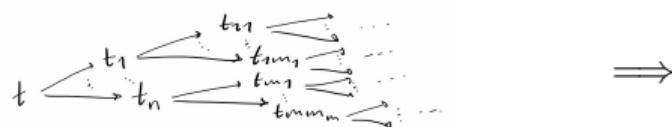
\Rightarrow



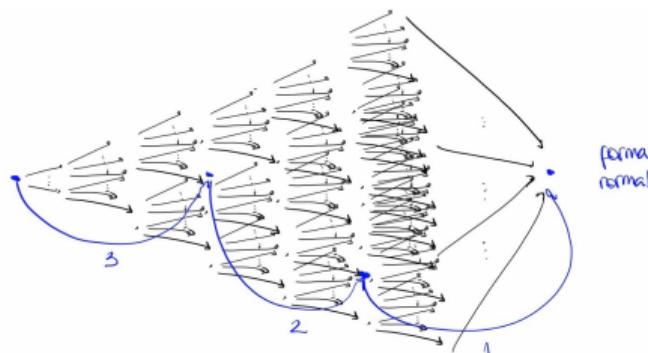
Una medida decreciente para Λ_{\cap} (tipos intersección idempotente)

[Barenbaum, Ronchi della Rocca, S.'25]

1. Enriquecemos el árbol



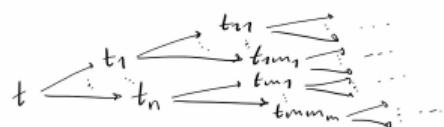
2. Reconstruimos parcialmente el árbol



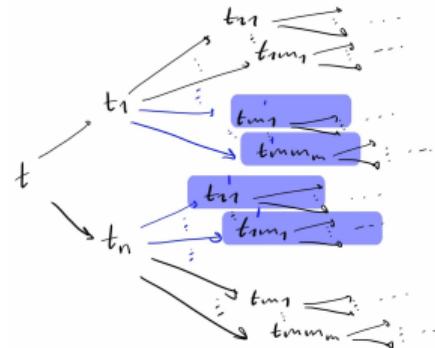
Una medida decreciente para Λ_{\cap} (tipos intersección idempotente)

[Barenbaum, Ronchi della Rocca, S.'25]

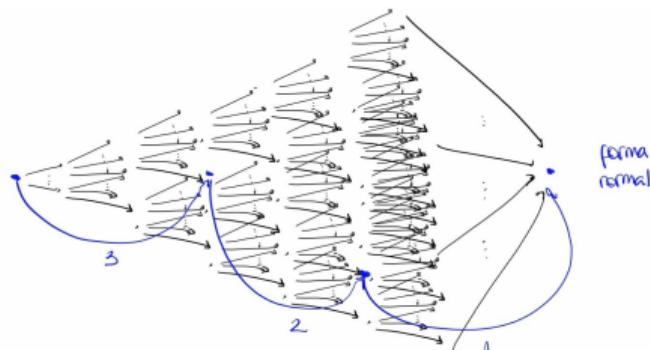
1. Enriquecemos el árbol



\Rightarrow



2. Reconstruimos parcialmente el árbol



3. Buscamos nodos irreproducibles post-ejecución

