Strong normalization y medidas decrecientes: demostraciones sintácticas de terminación en λ-cálculo tipado

Pablo Barenbaum
Universidad de Buenos Aires
CONICET
Universidad Nacional de Quilmes

Cristian Sottile
Universidad de Buenos Aires
CONICET
Universidad Nacional de Quilmes

54 JAIIO

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

4 de agosto de 2025

λ-cálculo

Estructura inductiva de los programas

$$t \ ::= \ x \mid \lambda x.t \mid tt$$

Reglas de cómputo

$$(\lambda x.t)s \to_\beta t[s/x]$$

λ-cálculo

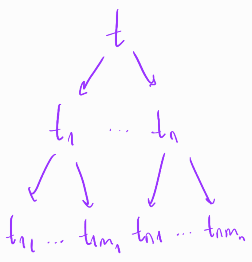
Estructura inductiva de los programas

$$t \ ::= \ x \mid \lambda x.t \mid tt$$

Reglas de cómputo

$$(\lambda x.t)s \to_\beta t[s/x]$$

Cadenas de reducción



λ-cálculo

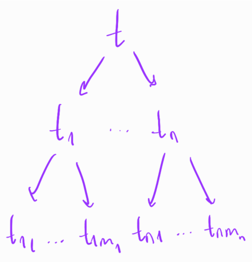
Estructura inductiva de los programas

$$t \ ::= \ x \mid \lambda x.t \mid tt$$

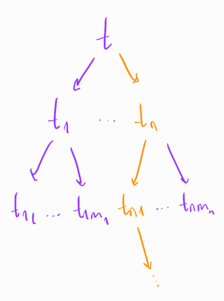
Reglas de cómputo

$$(\lambda x.t)s \to_\beta t[s/x]$$

Cadenas de reducción



Pueden ser infinitas



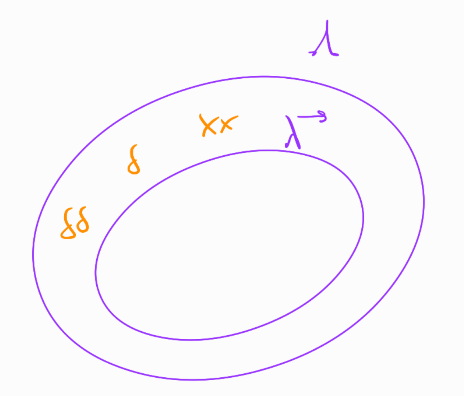
λ-cálculo tipado

Motivación Lenguaje más seguro

Admitimos solo términos que tengan "sentido" *e.g.* $f\ x$

Tipos

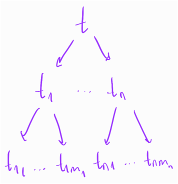
$$A \ ::= \ \tau \mid A \to A$$

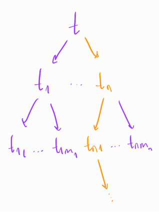


La propiedad de terminación: *Strong normalization*

Definición

$$\nexists t \to_\beta t_1 \to_\beta t_2 \to_\beta \cdots$$

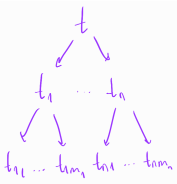


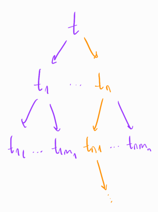


La propiedad de terminación: *Strong normalization*

Definición

$$\nexists t \to_\beta t_1 \to_\beta t_2 \to_\beta \cdots$$





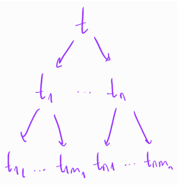
Motivación

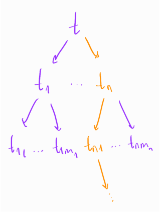
- Obtener un resultado del cómputo
- Equivale a la simplificación de pruebas (vía Curry-Howard)
- Desarrollo de técnicas (*e.g.* tipos intersección, logical relations)
- Es interesante

La propiedad de terminación: *Strong normalization*

Definición

$$\nexists t \to_\beta t_1 \to_\beta t_2 \to_\beta \cdots$$





Motivación

- Obtener un resultado del cómputo
- Equivale a la simplificación de pruebas (vía Curry-Howard)
- Desarrollo de técnicas (*e.g.* tipos intersección, logical relations)
- Es interesante

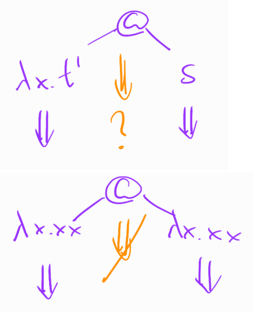
Reducibilidad [Tait'67, Girard'72]: la técnica más usada

- Concisa
- Extensible a sistemas más complejos (*e.g.* System F, CoC)

Construyendo reducibilidad por prueba y error

Primer intento

Inducción en t



Caso aplicación ts

- HI: t y s SN
- **no alcanza** para ver SN si $t = \lambda x.t'$

Solución

- observar qué deben cumplir los términos para ser SN
- ¿qué necesito de la HI?

Dado un término t, debe cumplir:

1. ser SN "solo"
2. ser SN al "combinarse" (aplicarse)

Candidatos de reducibilidad

- Los tipos indican cómo puede combinarse un término
- Por inducción en el tipo definimos los conjuntos de términos que cumplen: los **reducibles**

¿Por qué buscar alternativas?

Gallier (en *Proving properties of typed λ-terms using realizability, covers, and sheaves*)

> This paper provides some answers to the above questions. But before explaining our results, we would like to explain our motivations and our point of view a little more. Reducibility proofs are seductive and thrilling, but also elusive. Following these proofs step-by-step, we see that they "work" (when they are not wrong!), but I claim that most of us would still admit that they are not sure *why* these proofs work! The situation is somewhat comparable to driving a Ferrari (I suppose): the feeling of power is tremendous, but what exactly is under the hood? What kind of carburator, what kind of valve mechanism, gives such power and flexibility?

van de Pol (en *Two different strong normalization proofs?*)

> In the literature, these two methods are often put in contrast ([Gan80, § 6.3] and [Gir87, annex 2.C.1]). The proof using functionals seems to be more transparent and economizes on proof theoretical complexity. On the other hand, seeing the two proofs one gets the feeling that "somehow, the same thing is going on". Indeed De Vrijer [dV87, § 0.1] remarks that a proof using strong computability can be seen as abstracting from concrete information in the functionals that is not strictly needed in a termination proof, but which provides for an estimate of reduction lengths.

Medidas decrecientes

Definición

A mapping

$$\# : \Lambda \to \mathit{WFO}$$

satisfying

$$M \to_\beta N \implies \#(M) > \#(N)$$

Corolario

$$\nexists \quad M_1 \to_\beta M_2 \to_\beta \cdots$$

$$\#(M_1) > \#(M_2) > \cdots$$

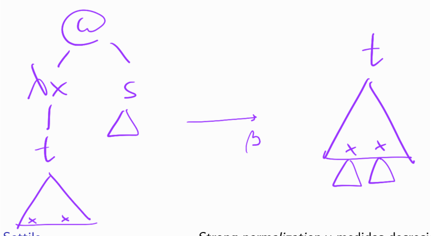
Motivación

- insight
- intuition
- metrics

El Koan #26

- Posed by Gödel
- Submitted by Barendregt
- Find an "easy" mapping from $\lambda^{\to}$ to ordinals

¿Por qué no es trivial?



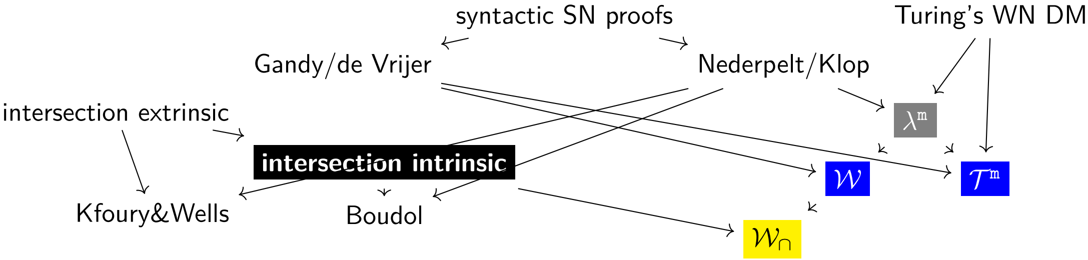
Resultados

[Barenbaum & Sottile FSCD'23]

- **Cálculo con memoria $\lambda^{\mathtt{m}}$** para manipular el "descarte" de cómputo
- **Medida $\mathcal{W}$**: contar de memorias acumuladas
- **Medida $\mathcal{T}^{\mathtt{m}}$**: generalización de la medida parcial de Turing

[Barenbaum, Ronchi della Rocca, Sottile MFPS'25]

- Presentación **à la Church** de tipos intersección
- Adaptación de $\mathcal{W}$ a tipos intersección



El cálculo con memorias $\lambda^{\mathtt{m}}$

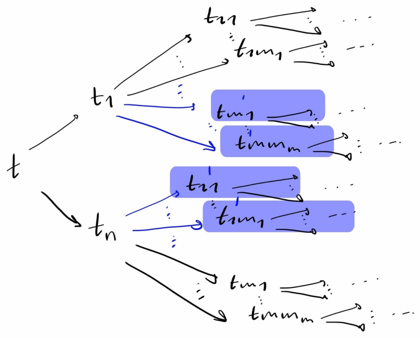
El cálculo $\lambda^{\mathtt{m}}$

Definición

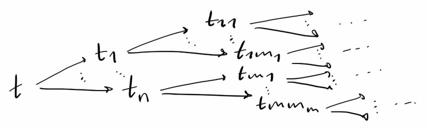
$$t \;::=\; x^A \mid \lambda x.t \mid t\,t \mid t\langle t\rangle \qquad (\lambda x^A.t)s \to_m t[s/x^A]\langle s\rangle$$

Intuiciones

árbol de ejecución en $\lambda^{\mathtt{m}}$



árbol de ejecución en cálculo λ simple



Redex degrees

Altura del tipo

Altura visto como árbol

Ejemplo: tipo identidad

$\tau \to \tau$

$\to$

$\tau \qquad \tau$

1

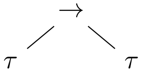
Redex degrees

Altura del tipo

Altura visto como árbol

Ejemplo: tipo identidad

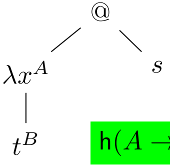
$\tau \to \tau$



1

Grado de un redex

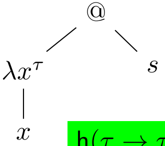
Altura del tipo de la función



$\mathsf{h}(A \to B)$

Ejemplo: función identidad

$(\lambda x^\tau.x)s$

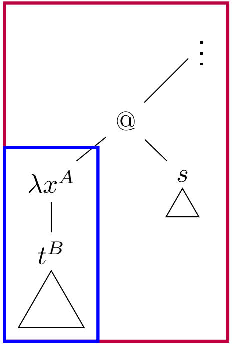


$\mathsf{h}(\tau \to \tau) = 1$

Turing: cota para redex degrees

Definición

grado del tipo de la función

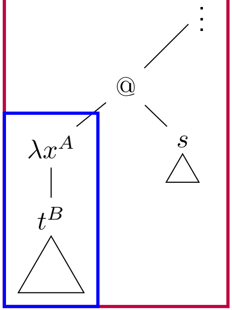


deg$(A \to B)$

Turing: cota para redex degrees

Definición

grado del tipo de la función

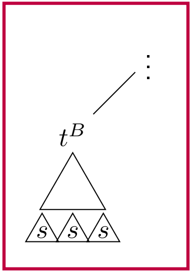


$\mathit{deg}(A \to B)$

$\to_\beta$

Observación de Turing

elegir un paso de reducción crea nuevos pasos **más pequeños**



$\mathit{deg}(R \text{ new}) < \mathit{deg}(A \to B)$

$\mathcal{W}$: counting memories

La medida $\mathcal{W}$: definición

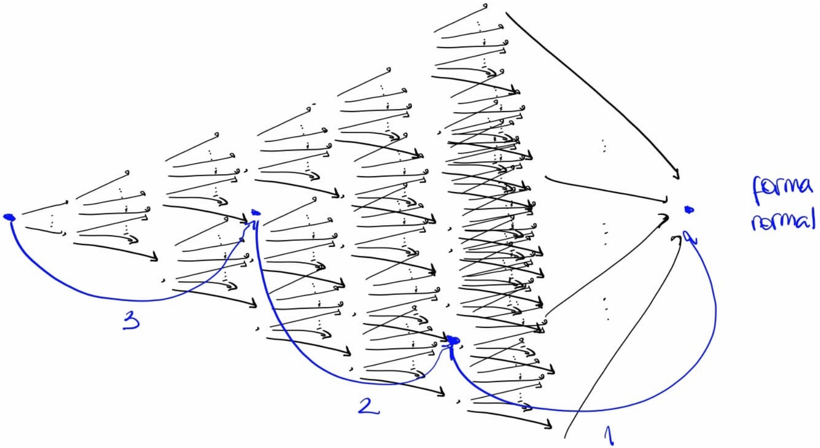
Operaciones

simplificación

de grado D
de un programa
(recursión estructural)

- Movilidad en el árbol
- Garantía de forma normal
- "Atajo de ejecución" en el metalenguaje

$\mathcal{W}(t) \;=\; \mathsf{w}(\mathsf{S}_*(t))$



La medida $\mathcal{W}$: prueba

Lemas

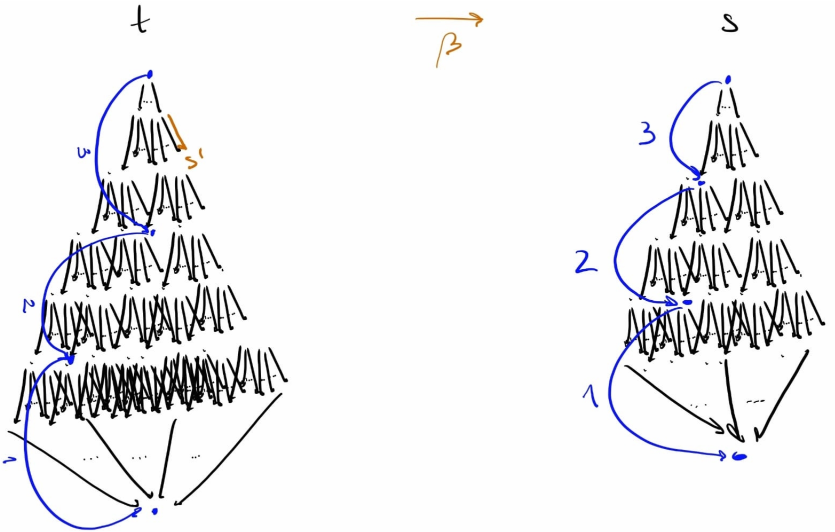
- $t \to_m^* \mathsf{S}_*(t)$
- $\mathsf{S}_*(t) = \mathtt{nf}(t)$
- $D_{max}(t) > D_{max}(\mathsf{S}_{D_{max}}(t))$

Teorema

$\mathcal{W}$ decrece con la ejecución $t \to_{\mathsf{Ch}} s \implies \mathcal{W}(t) > \mathcal{W}(s)$

Intuición

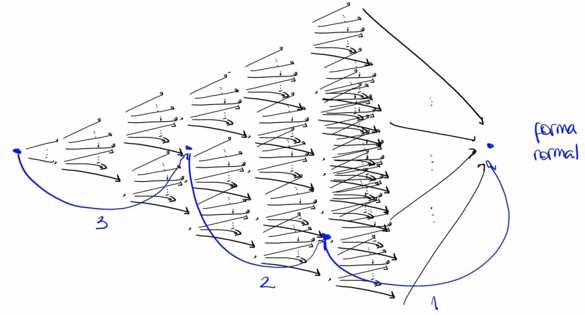
- Si $t \to s$ borra
- Si no $t \to s$ "aparece" en $\mathcal{W}(t)$ pero no en $\mathcal{W}(s)$



Otros resultados

Otra medida: generalización de una medida de Turing **[Barenbaum, S.]**

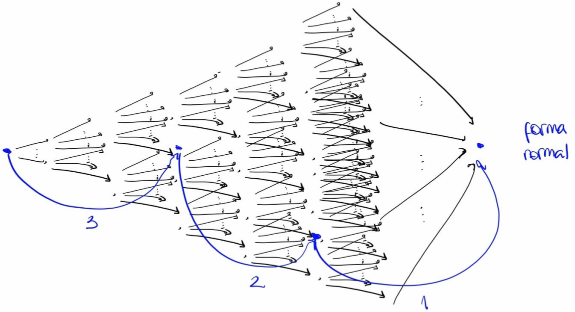
- Turing dio una medida parcial basada en multisets
- Solo funciona para una rama de ejecución
- Medida:
  - multisets anidados [(4,m1), (3,m2), (3,m3), ...]
  - elementos obtenidos de distintas proyecciones del árbol



Otros resultados

Otra medida: generalización de una medida de Turing [Barenbaum, S.]

- Turing dio una medida parcial basada en multisets
- Solo funciona para una rama de ejecución
- Medida:
  - multisets anidados [(4,m1), (3,m2), (3,m3), ...]
  - elementos obtenidos de distintas proyecciones del árbol



Adaptación de $\mathcal{W}$ a tipos intersección [Barenbaum, Ronchi della Rocca, S.]

- Tipos intersección: las variables tienen muchos tipos
- Definimos variante intrínseca
- Introducimos memorias y definimos la medida

Conclusiones y trabajo futuro

Conclusiones

- Repasamos técnicas de demostración de strong normalization
- Definimos el cálculo con memorias λ^m, con el que:
  - definimos $\mathcal{W}$: la medida de las memorias acumuladas en el resultado
  - adaptamos $\mathcal{W}$ a $\Lambda_\cap$, obteniendo una medida más sencilla que las existentes
  - generalizamos una medida de Turing desde WN a SN

Conclusiones y trabajo futuro

Conclusiones

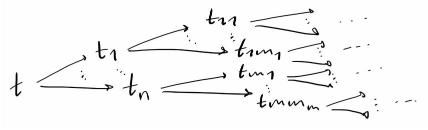
- Repasamos técnicas de demostración de strong normalization
- Definimos el cálculo con memorias λ^m, con el que:
  - definimos $\mathcal{W}$: la medida de las memorias acumuladas en el resultado
  - adaptamos $\mathcal{W}$ a $\Lambda_\cap$, obteniendo una medida más sencilla que las existentes
  - generalizamos una medida de Turing desde WN a SN

Trabajo futuro

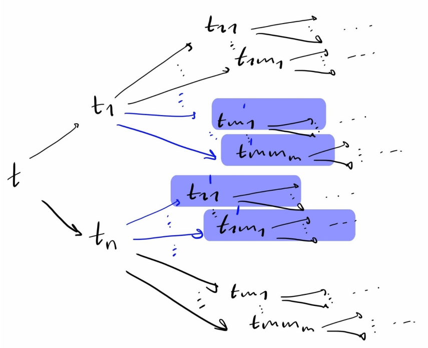
- **Refinado** de $\mathcal{W}$ a **exactitud**: $\mathcal{W}(t)$ es exactamente el "costo" de ejecución de t
- **Adaptación** de $\mathcal{W}$ a otro sistema de tipos intersección
- **Formalización** en asistentes de pruebas (Agda, Coq, Lean)
- **Comparación** de $\mathcal{W}$ con las medidas de Gandy y de Vrijer
- **Explorar** la posibilidad de extender $\mathcal{W}$ a System F

Refinado de $\mathcal{W}$ a exactitud

- árbol de ejecución en cálculo λ simple



- árbol de ejecución en $\lambda^{\mathtt{m}}$



$\mathcal{T}^{\mathtt{m}}$: generalizing Turing's WN measure

La medida $\mathcal{T}^{\mathtt{m}}$

Adaptación a **SN** de una medida dada por Turing **para WN**

$$[3, 3, 3, 2, 1, 1, 1]$$

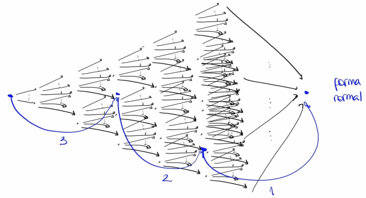
Agregamos información

$$[\ (2, \boxed{?}),\ \ (1, \boxed{?})\ ]$$

Idea

Development of degree D reduction involving only redexes D

All developments of degree D paths of the complete D-reduction graph from t



Mediante una técnica similar a la simplificación, **seleccionamos partes de este árbol**, y la medida es una anidación sobre esas partes

$\mathcal{W}_\cap$: extending $\mathcal{W}$ to Idempotent Intersection Types

Motivation

To find *simpler* proofs of *strong normalization* for idempotent intersection types

Existing decreasing measures

Motivation

To find *simpler* proofs of *strong normalization* for idempotent intersection types

Existing decreasing measures

[Kfoury & Wells'95]

- **Domain of DM:** multiset of numbers
- **Methodology:** WN $\implies$ SN + DM proving WN (indirect)
- **Auxiliary calculus:** a la Curry

Motivation

To find *simpler* proofs of *strong normalization* for idempotent intersection types

Existing decreasing measures

[Kfoury & Wells'95]

- **Domain of DM:** multiset of numbers
- **Methodology:** WN $\implies$ SN + DM proving WN (indirect)
- **Auxiliary calculus:** a la Curry

[Boudol'03]

- **Domain of DM:** pair of numbers
- **Methodology:** WN $\implies$ SN + DM proving WN (indirect)
- **Auxiliary calculus:** a la Church, ad hoc

Motivation

To find *simpler* proofs of *strong normalization* for idempotent intersection types

Existing decreasing measures

[Kfoury & Wells'95]

- **Domain of DM:** multiset of numbers
- **Methodology:** WN $\implies$ SN + DM proving WN (indirect)
- **Auxiliary calculus:** a la Curry

[Boudol'03]

- **Domain of DM:** pair of numbers
- **Methodology:** WN $\implies$ SN + DM proving WN (indirect)
- **Auxiliary calculus:** a la Church, ad hoc

Our proposal **Barenbaum, Ronchi della Rocca & Sottile (WIP)**

- **Domain of DM:** **number**
- **Methodology:** DM proving SN **(direct)**
- **Auxiliary calculus:** a la Church, **correspondent** of a la Curry calculus

Tipos intersección idempotente($\Lambda_{\cap}^{\mathsf{Cu}}$)

[Coppo-Dezzani'79]

Key idea Allowing variables to have multiple types

$$\Gamma, x : A \vdash x : A \qquad \rightsquigarrow \qquad \Gamma, \boxed{x : \{A_1, ..., A_n\}} \vdash x : A_i$$

Tipos intersección idempotente($\Lambda_{\cap}^{\mathsf{Cu}}$)

[Coppo-Dezzani'79]

Key idea Allowing variables to have multiple types

$$\Gamma, x : A \vdash x : A \qquad \rightsquigarrow \qquad \Gamma, x : \{A_1, ..., A_n\} \vdash x : A_i$$

Grammar of types

$$A ::= a \mid \{A_1, ..., A_n\} \to A \qquad (A_i \neq A_j \text{ if } i \neq j) \quad (n \in \mathbb{N})$$

Tipos intersección idempotente($\Lambda_{\cap}^{\mathsf{Cu}}$)

[Coppo-Dezzani'79]

Key idea Allowing variables to have multiple types

$$\Gamma, x : A \vdash x : A \qquad \rightsquigarrow \qquad \Gamma, \boxed{x : \{A_1, ..., A_n\}} \vdash x : A_i$$

Grammar of types

$$A ::= a \mid \{A_1, ..., A_n\} \to A \qquad (A_i \neq A_j \text{ if } i \neq j) \quad (n \in \mathbb{N})$$

Typing rules

$$\dfrac{\Gamma \vdash_{\mathsf{e}} M : \vec{A} \to B \qquad \boxed{\dfrac{(\Gamma \vdash_{\mathsf{e}} N : A_i)_{i \in I}}{\Gamma \Vdash_{\mathsf{e}} \boxed{N} : \{A_1, \ldots, A_n\}}\,\cap}}{\Gamma \vdash_{\mathsf{e}} M\, N : B}\,@$$

Tipos intersección idempotente($\Lambda_\cap^{\mathsf{Ch}}$)

Key idea Allowing variables to have multiple types

$$\Gamma, x : A \vdash x : A \qquad \rightsquigarrow \qquad \Gamma, x : \{A_1, ..., A_n\} \vdash x^{A_i} : A_i$$

Tipos intersección idempotente($\Lambda^{\mathsf{Ch}}_{\cap}$)

Key idea Allowing variables to have multiple types

$$\Gamma, x : A \vdash x : A \qquad \rightsquigarrow \qquad \Gamma, \boxed{x : \{A_1, ..., A_n\}} \vdash \boxed{x^{A_i}} : A_i$$

Typing rules

$$\frac{\Gamma \vdash_{\mathsf{e}} M : \{A_1, \ldots, A_n\} \to B \qquad \boxed{\dfrac{(\Gamma \vdash_{\mathsf{e}} s_i : A_i)_{i \in I}}{\Gamma \Vdash_{\mathsf{e}} \boxed{\{s_1, ..., s_n\}} : \{A_1, \ldots, A_n\}}\ \cap}}{\Gamma \vdash_{\mathsf{e}} M\,\{s_1, ..., s_n\} : B}\ @$$

Tipos intersección idempotente($\Lambda_\cap^{\mathsf{Ch}}$)

Key idea Allowing variables to have multiple types

$$\Gamma, x : A \vdash x : A \qquad \rightsquigarrow \qquad \Gamma, \boxed{x : \{A_1, ..., A_n\}} \vdash \boxed{x^{A_i}} : A_i$$

Typing rules

$$\dfrac{\Gamma \vdash_{\mathsf{e}} M : \{A_1, \ldots, A_n\} \to B \qquad \dfrac{(\Gamma \vdash_{\mathsf{e}} s_i : A_i)_{i \in I}}{\boxed{\Gamma \Vdash_{\mathsf{e}} \boxed{\{s_1, ..., s_n\}} : \{A_1, \ldots, A_n\}}}\cap}{\Gamma \vdash_{\mathsf{e}} M\,\{s_1, ..., s_n\} : B}@$$

Propiedades

- Correpondencia
- Simulación

$$\begin{array}{ccc} M & \xrightarrow[\beta]{} & N \\ \sqcup & & \sqcup \\ t & \overset{+}{\underset{\mathsf{Ch}}{\dashrightarrow}} & s \end{array} \qquad\qquad \begin{array}{ccccc} t & \xrightarrow[\mathsf{Ch}]{} & s & \overset{*}{\underset{\mathsf{Ch}}{\dashrightarrow}} & s \\ \sqcap & & & & \sqcap \\ M & & \underset{\beta}{\dashrightarrow} & & N \end{array}$$

La medida decreciente

Adaptación de la técnica a $\Lambda_{\cap}^{\mathsf{Ch}}$

- Adición de memorias al lenguaje con multitérminos-tipos
- Definición de medida $\mathsf{w}(\mathsf{S}_*(t))$
- Adaptación/adición de casos en las definiciones y pruebas

La medida decreciente

Adaptación de la técnica a $\Lambda_{\cap}^{\mathsf{Ch}}$

- Adición de memorias al lenguaje con multitérminos-tipos
- Definición de medida $\mathsf{w}(\mathsf{S}_*(t))$
- Adaptación/adición de casos en las definiciones y pruebas

Traslación del resultado: de Church a Curry

$$\begin{array}{ccc} \boldsymbol{M} & \to_{\boldsymbol{\beta}} & \boldsymbol{N} \\ \sqcup & & \sqcup \\ t & \to^{+}_{\mathsf{Ch}} & s \\ \Big| & & \Big| \\ \mathcal{W}_{\cap}(t) & & \mathcal{W}_{\cap}(s) \end{array}$$

La medida decreciente

Adaptación de la técnica a $\Lambda_{\cap}^{\mathsf{Ch}}$

- Adición de memorias al lenguaje con multitérminos-tipos
- Definición de medida $\mathsf{w}(\mathsf{S}_*(t))$
- Adaptación/adición de casos en las definiciones y pruebas

Traslación del resultado: de Church a Curry

$$\begin{tikzcd}
\boldsymbol{M} \arrow[r, "\beta"] \arrow[d, phantom, "\sqcup"] & \boldsymbol{N} \arrow[d, phantom, "\sqcup"] \\
t \arrow[r, "+"', "\mathsf{Ch}"] \arrow[d, dash] & s \arrow[d, dash] \\
\mathcal{W}_{\cap}(t) & \mathcal{W}_{\cap}(s)
\end{tikzcd}$$

$\Lambda_{\cap}^{\mathsf{Cu}}$ satisface SN $\Gamma \vdash_{\mathsf{Cu}} M : A \implies M \in SN$