Cálculo ($\lambda x.$L$x$mbd$x$) a

Paradigmas de Lenguajes de Programación

Departamento de Ciencias de la Computación
Universidad de Buenos Aires

26 de septiembre de 2025

Objetivo de la clase

Dada la siguiente expresión:

$(\lambda x\colon \mathsf{Bool}.\ \lambda y : \mathsf{Bool} \to \mathsf{Bool}.\ y\ (y\ x))\ ((\lambda z\colon \mathsf{Bool}.\ \mathsf{true})\ \mathsf{false})\ (\lambda w\colon \mathsf{Bool}.\ w)$

¿Qué significa esto? ¿Significa algo? ¿Es válido? ¿Es un valor? ¿Cómo nos damos cuenta?

Vamos a ver

- ✨ Sintaxis del cálculo lambda
- ✨ Tipado
- ✨ Semántica operacional, estrategias de reducción
- ✨ Extensión de números naturales

Sintaxis

Los tipos del cálculo lambda simplemente tipado con booleanos se definen mediante la siguiente gramática:

$$\sigma ::= \mathsf{Bool} \mid \sigma \to \sigma$$

Sintaxis

Los tipos del cálculo lambda simplemente tipado con booleanos se definen mediante la siguiente gramática:

$$\sigma ::= \mathsf{Bool} \mid \sigma \to \sigma$$

y sus términos son los siguientes:

$$M ::= x \mid \lambda x : \sigma . M \mid M M \mid \mathsf{true} \mid \mathsf{false} \mid \mathsf{if}\ M\ \mathsf{then}\ M\ \mathsf{else}\ M$$

donde $x \in \mathcal{X}$, el conjunto de todas las variables. Llamamos $\mathcal{T}$ al conjunto de todos los términos.

Sintaxis

Los tipos del cálculo lambda simplemente tipado con booleanos se definen mediante la siguiente gramática:

$$\sigma ::= \mathsf{Bool} \mid \sigma \to \sigma$$

y sus términos son los siguientes:

$$M ::= x \mid \lambda x : \sigma . M \mid M M \mid \mathsf{true} \mid \mathsf{false} \mid \mathsf{if}\ M\ \mathsf{then}\ M\ \mathsf{else}\ M$$

donde $x \in \mathcal{X}$, el conjunto de todas las variables. Llamamos $\mathcal{T}$ al conjunto de todos los términos.

Variables libres y ligadas

Las variables libres son todas aquellas fuera del alcance de las λs. Se define la función $\mathsf{fv} : \mathcal{T} \to \mathcal{P}(\mathcal{X})$, que dado un término devuelve un conjunto de las variables libres en él.

Sintaxis

Los tipos del cálculo lambda simplemente tipado con booleanos se definen mediante la siguiente gramática:

$$\sigma ::= \mathsf{Bool} \mid \sigma \to \sigma$$

y sus términos son los siguientes:

$$M ::= x \mid \lambda x : \sigma.M \mid M M \mid \mathsf{true} \mid \mathsf{false} \mid \mathsf{if}\ M\ \mathsf{then}\ M\ \mathsf{else}\ M$$

donde $x \in \mathcal{X}$, el conjunto de todas las variables. Llamamos $\mathcal{T}$ al conjunto de todos los términos.

Variables libres y ligadas

Las variables libres son todas aquellas fuera del alcance de las λs. Se define la función $\mathsf{fv} : \mathcal{T} \to \mathcal{P}(\mathcal{X})$, que dado un término devuelve un conjunto de las variables libres en él.

$$\begin{aligned} \mathsf{fv}(x) &= \{x\} & \mathsf{fv}(\mathsf{true}) &= \emptyset \\ \mathsf{fv}(\lambda x : \sigma.M) &= \mathsf{fv}(M) \backslash \{x\} & \mathsf{fv}(\mathsf{false}) &= \emptyset \\ \mathsf{fv}(M N) &= \mathsf{fv}(M) \cup \mathsf{fv}(N) & \mathsf{fv}(\mathsf{if}\ M\ \mathsf{then}\ N\ \mathsf{else}\ O) &= \mathsf{fv}(M) \cup \mathsf{fv}(N) \cup \mathsf{fv}(O) \end{aligned}$$

Sintaxis

Los tipos del cálculo lambda simplemente tipado con booleanos se definen mediante la siguiente gramática:

$$\sigma ::= \mathsf{Bool} \mid \sigma \to \sigma$$

y sus términos son los siguientes:

$$M ::= x \mid \lambda x : \sigma.M \mid M M \mid \mathsf{true} \mid \mathsf{false} \mid \mathsf{if}\ M\ \mathsf{then}\ M\ \mathsf{else}\ M$$

donde $x \in \mathcal{X}$, el conjunto de todas las variables. Llamamos $\mathcal{T}$ al conjunto de todos los términos.

Variables libres y ligadas

Las variables libres son todas aquellas fuera del alcance de las λs. Se define la función $\mathsf{fv} : \mathcal{T} \to \mathcal{P}(\mathcal{X})$, que dado un término devuelve un conjunto de las variables libres en él.

$$\begin{aligned} \mathsf{fv}(x) &= \{x\} & \mathsf{fv}(\mathsf{true}) &= \emptyset \\ \mathsf{fv}(\lambda x : \sigma.M) &= \mathsf{fv}(M) \backslash \{x\} & \mathsf{fv}(\mathsf{false}) &= \emptyset \\ \mathsf{fv}(MN) &= \mathsf{fv}(M) \cup \mathsf{fv}(N) & \mathsf{fv}(\mathsf{if}\ M\ \mathsf{then}\ N\ \mathsf{else}\ O) &= \mathsf{fv}(M) \cup \mathsf{fv}(N) \cup \mathsf{fv}(O) \end{aligned}$$

Un término se llama cerrado si no tiene variables libres, es decir, M es cerrado si y sólo si $\mathsf{fv}(M) = \emptyset$.

Sintaxis

Asociatividad y precedencia

$\sigma \to \tau \to \rho = \sigma \to (\tau \to \rho) \neq (\sigma \to \tau) \to \rho$	Las flechas en los tipos asocian a derecha.
$MNO = (MN)O \neq M(NO)$	La aplicación asocia a izquierda.
$\lambda x : \sigma.MN = \lambda x : \sigma.(MN) \neq (\lambda x : \sigma.M)N$	El cuerpo de la lambda se extiende hasta el final del término, excepto que haya paréntesis.

Sintaxis

Asociatividad y precedencia

$\sigma \to \tau \to \rho = \sigma \to (\tau \to \rho) \neq (\sigma \to \tau) \to \rho$	Las flechas en los tipos asocian a derecha.
$MNO = (MN)O \neq M(NO)$	La aplicación asocia a izquierda.
$\lambda x : \sigma.MN = \lambda x : \sigma.(MN) \neq (\lambda x : \sigma.M)N$	El cuerpo de la lambda se extiende hasta el final del término, excepto que haya paréntesis.

Ejercicio: ¿Cuáles de las siguientes expresiones son términos del cálculo lambda? En los casos que sí lo sean, dibujar su árbol sintáctico y marcar las ocurrencias libres de las variables.

a) $\lambda x : \mathsf{Bool} \to \mathsf{Bool}.x$ true

b) $x\ y\ \lambda x : \mathsf{Bool} \to \mathsf{Bool}.x\ y$

c) $(\lambda x : \mathsf{Bool} \to \mathsf{Bool}.x\ y)(\lambda y : \mathsf{Bool}.x)$

d) $\lambda x : \mathsf{Bool}$

e) $\lambda x.x$

f) if x then y else $\lambda z : \mathsf{Bool}.z$

g) $\lambda y : \sigma.y$

h) true false

i) $x\ M$

j) if x then $\lambda x : \mathsf{Bool}.x$

Tipado

Tipos: La gramática que define los tipos del cálculo lambda simplemente tipado con booleanos es:

$$\sigma ::= \mathsf{Bool} \mid \sigma \to \sigma$$

Tipado

Tipos: La gramática que define los tipos del cálculo lambda simplemente tipado con booleanos es:

$$\sigma ::= \mathsf{Bool} \mid \sigma \to \sigma$$

Los contextos son conjuntos finitos de asociaciones entre tipos y variables. Por ejemplo:

$$\Gamma_1 = y{:}\mathsf{Bool} \to \mathsf{Bool} \qquad \Gamma_2 = y{:}\mathsf{Bool} \to \mathsf{Bool}, x{:}\mathsf{Bool}$$

son contextos válidos,

Tipado

Tipos: La gramática que define los tipos del cálculo lambda simplemente tipado con booleanos es:

$$\sigma ::= \mathsf{Bool} \mid \sigma \to \sigma$$

Los contextos son conjuntos finitos de asociaciones entre tipos y variables. Por ejemplo:

$$\Gamma_1 = y{:}\mathsf{Bool} \to \mathsf{Bool} \qquad \Gamma_2 = y{:}\mathsf{Bool} \to \mathsf{Bool}, x{:}\mathsf{Bool}$$

son contextos válidos, pero

$$\Gamma_3 = y{:}\mathsf{Bool} \to \mathsf{Bool}, y{:}\mathsf{Bool}$$

no lo es.

Tipado

Tipos: La gramática que define los tipos del cálculo lambda simplemente tipado con booleanos es:

$$\sigma ::= \mathsf{Bool} \mid \sigma \to \sigma$$

Los contextos son conjuntos finitos de asociaciones entre tipos y variables. Por ejemplo:

$$\Gamma_1 = y{:}\mathsf{Bool} \to \mathsf{Bool} \qquad \Gamma_2 = y{:}\mathsf{Bool} \to \mathsf{Bool}, x{:}\mathsf{Bool}$$

son contextos válidos, pero

$$\Gamma_3 = y{:}\mathsf{Bool} \to \mathsf{Bool}, y{:}\mathsf{Bool}$$

no lo es.

Juicios de tipado: Un juicio de tipado es la relación $\Gamma \vdash M : \tau$ y se lee "en el contexto Γ, M es de tipo τ ".

Tipado

Tipos: La gramática que define los tipos del cálculo lambda simplemente tipado con booleanos es:

$$\sigma ::= \mathsf{Bool} \mid \sigma \to \sigma$$

Los contextos son conjuntos finitos de asociaciones entre tipos y variables. Por ejemplo:

$$\Gamma_1 = y{:}\mathsf{Bool} \to \mathsf{Bool} \qquad \Gamma_2 = y{:}\mathsf{Bool} \to \mathsf{Bool}, x{:}\mathsf{Bool}$$

son contextos válidos, pero

$$\Gamma_3 = y{:}\mathsf{Bool} \to \mathsf{Bool}, y{:}\mathsf{Bool}$$

no lo es.

Juicios de tipado: Un juicio de tipado es la relación $\Gamma \vdash M : \tau$ y se lee "en el contexto Γ, M es de tipo τ ". Por ejemplo:

$$\begin{aligned} \{x{:}\mathsf{Bool} \to \mathsf{Bool}\} &\vdash x : \mathsf{Bool} \to \mathsf{Bool} \\ &\vdash \mathsf{true} : \mathsf{Bool} \\ \{f{:}\mathsf{Bool} \to \mathsf{Bool}, x{:}\mathsf{Bool}\} &\vdash f\,x : \mathsf{Bool} \end{aligned}$$

son juicios de tipado válidos.

Tipado

Sistema de tipado

Los juicios de tipado $\Gamma \vdash M : \tau$ válidos se pueden derivar mediante el siguiente sistema de reglas de deducción:

$$\frac{}{\Gamma, x : \tau \vdash x : \tau} \; \textit{T-Var} \qquad \frac{\Gamma, x : \tau \vdash M : \sigma}{\Gamma \vdash \lambda x \colon \tau . M : \tau \to \sigma} \; \textit{T-Abs} \qquad \frac{\Gamma \vdash M : \tau \to \sigma \quad \Gamma \vdash N : \tau}{\Gamma \vdash M N : \sigma} \; \textit{T-App}$$

$$\frac{}{\Gamma \vdash \mathsf{true} : \mathsf{Bool}} \; \textit{T-True} \qquad \frac{}{\Gamma \vdash \mathsf{false} : \mathsf{Bool}} \; \textit{T-False} \qquad \frac{\Gamma \vdash M : \mathsf{Bool} \quad \Gamma \vdash N_1 : \tau \quad \Gamma \vdash N_2 : \tau}{\Gamma \vdash \mathsf{if}\ M\ \mathsf{then}\ N_1\ \mathsf{else}\ N_2 : \tau} \; \textit{T-If}$$

Tipado

Ejercicio: chequeo de tipos

Derivar los siguientes juicios de tipado, o explicar por qué no son válidos.

a) $\vdash (\lambda x : \mathsf{Bool}.\lambda y : \mathsf{Bool}.\mathsf{if}\ x\ \mathsf{then\ true\ else}\ y)\ \mathsf{false} : \mathsf{Bool} \to \mathsf{Bool}$

Tipado

Ejercicio: chequeo de tipos

Derivar los siguientes juicios de tipado, o explicar por qué no son válidos.

a) $\vdash (\lambda x : \mathsf{Bool}.\lambda y : \mathsf{Bool}.\mathsf{if}\ x\ \mathsf{then}\ \mathsf{true}\ \mathsf{else}\ y)\ \mathsf{false} : \mathsf{Bool} \to \mathsf{Bool}$

b) $\{x : \mathsf{Bool}\} \vdash \mathsf{true} : \mathsf{Bool}$

Tipado

Ejercicio: chequeo de tipos

Derivar los siguientes juicios de tipado, o explicar por qué no son válidos.

a) $\vdash (\lambda x : \mathsf{Bool}.\lambda y : \mathsf{Bool}.\mathsf{if}\ x\ \mathsf{then\ true\ else}\ y)\ \mathsf{false} : \mathsf{Bool} \to \mathsf{Bool}$

b) $\{x : \mathsf{Bool}\} \vdash \mathsf{true} : \mathsf{Bool}$

c) $\vdash \mathsf{if}\ x\ \mathsf{then}\ x\ \mathsf{else}\ z : \mathsf{Bool}$

Tipado

Ejercicio: chequeo de tipos

Derivar los siguientes juicios de tipado, o explicar por qué no son válidos.

a) $\vdash (\lambda x : \mathsf{Bool}.\lambda y : \mathsf{Bool}.\mathsf{if}\ x\ \mathsf{then\ true\ else}\ y)\ \mathsf{false} : \mathsf{Bool} \to \mathsf{Bool}$

b) $\{x : \mathsf{Bool}\} \vdash \mathsf{true} : \mathsf{Bool}$

c) $\vdash \mathsf{if}\ x\ \mathsf{then}\ x\ \mathsf{else}\ z : \mathsf{Bool}$

d) $\{x : \mathsf{Bool}\} \vdash \mathsf{if}\ x\ \mathsf{then}\ x\ \mathsf{else}\ (\lambda y : \mathsf{Bool}.y) : \mathsf{Bool} \to \mathsf{Bool}$

Tipado

Ejercicio: chequeo de tipos con incógnitas

Derivar un juicio de tipado para el siguiente término:

$\lambda x : \rho.\lambda y : \sigma.\lambda z : \tau.x(xyz)$ (identificando qué tipos pueden ser τ, σ y ρ)

Tipado

Ejercicio: chequeo de tipos con incógnitas

Derivar un juicio de tipado para el siguiente término:
$\lambda x : \rho.\lambda y : \sigma.\lambda z : \tau.x(xyz)$ (identificando qué tipos pueden ser τ, σ y ρ)

Ejercicio: tipos habitados

Decimos que un tipo τ está habitado si existe un término M tal que el juicio $\vdash M : \tau$ es derivable.

Tipado

Ejercicio: chequeo de tipos con incógnitas

Derivar un juicio de tipado para el siguiente término:
$\lambda x : \rho.\lambda y : \sigma.\lambda z : \tau.x(xyz)$ (identificando qué tipos pueden ser τ, σ y ρ)

Ejercicio: tipos habitados

Decimos que un tipo τ está habitado si existe un término M tal que el juicio $\vdash M : \tau$ es derivable.
Demostrar si los siguientes tipos están habitados (para cualquier σ, τ, ρ):

a) $\sigma \to \tau \to \sigma$

Tipado

Ejercicio: chequeo de tipos con incógnitas

Derivar un juicio de tipado para el siguiente término:
$\lambda x : \rho.\lambda y : \sigma.\lambda z : \tau.x(xyz)$ (identificando qué tipos pueden ser τ, σ y ρ)

Ejercicio: tipos habitados

Decimos que un tipo τ está habitado si existe un término M tal que el juicio $\vdash M : \tau$ es derivable.
Demostrar si los siguientes tipos están habitados (para cualquier σ, τ, ρ):

a) $\sigma \to \tau \to \sigma$
b) $(\tau \to \rho) \to (\sigma \to \tau) \to (\sigma \to \rho)$

Tipado

Ejercicio: chequeo de tipos con incógnitas

Derivar un juicio de tipado para el siguiente término:
$\lambda x : \rho.\lambda y : \sigma.\lambda z : \tau.x(xyz)$ (identificando qué tipos pueden ser τ, σ y ρ)

Ejercicio: tipos habitados

Decimos que un tipo τ está habitado si existe un término M tal que el juicio $\vdash M : \tau$ es derivable.
Demostrar si los siguientes tipos están habitados (para cualquier σ, τ, ρ):

a) $\sigma \to \tau \to \sigma$

b) $(\tau \to \rho) \to (\sigma \to \tau) \to (\sigma \to \rho)$

c) $\sigma \to \tau$

Semántica operacional

Consiste en un conjunto de reglas que definen la relación de reducción $\to$ entre términos.
Cuando $M \to N$, decimos que M reduce o reescribe a N.

Semántica operacional

Consiste en un conjunto de reglas que definen la relación de reducción $\to$ entre términos. Cuando $M \to N$, decimos que M reduce o reescribe a N.

Formas normales

Un término es o está en forma normal cuando no existe ninguna regla que lo reduzca a otro.

Semántica operacional

Consiste en un conjunto de reglas que definen la relación de reducción $\to$ entre términos.
Cuando $M \to N$, decimos que M reduce o reescribe a N.

Formas normales

Un término es o está en forma normal cuando no existe ninguna regla que lo reduzca a otro.

Determinismo

Decimos que la reducción está determinada (hay determinismo) cuando cada término que no está en forma normal tiene una única forma de reducir.

Semántica operacional

Consiste en un conjunto de reglas que definen la relación de reducción $\to$ entre términos. Cuando $M \to N$, decimos que M reduce o reescribe a N.

Formas normales

Un término es o está en forma normal cuando no existe ninguna regla que lo reduzca a otro.

Determinismo

Decimos que la reducción está determinada (hay determinismo) cuando cada término que no está en forma normal tiene una única forma de reducir.

Estrategias de reducción

Para implementar un lenguaje, necesitamos una relación de reducción que esté determinada. Existen estrategias call-by-name y call-by-value. En la parte práctica de la materia vamos a usar la estrategia **call-by-value**, y en particular nos va a interesar mantener el determinismo de las reglas de reducción.

Semántica operacional

La siguiente gramática de valores y las reglas de reducción definen la estrategia call-by-value.

$$V ::= \mathsf{true} \mid \mathsf{false} \mid \lambda x : \sigma . M$$

Semántica operacional

La siguiente gramática de valores y las reglas de reducción definen la estrategia call-by-value.

$$V ::= \mathsf{true} \mid \mathsf{false} \mid \lambda x : \sigma.M$$

$$
\begin{aligned}
(\lambda x : \sigma.M)V &\to M\{x := V\} && (\textit{E-AppAbs o } \beta) \\
\mathsf{if\ true\ then\ } M \mathsf{\ else\ } N &\to M && (\textit{E-IfTrue}) \\
\mathsf{if\ false\ then\ } M \mathsf{\ else\ } N &\to N && (\textit{E-IfFalse})
\end{aligned}
$$

Semántica operacional

La siguiente gramática de valores y las reglas de reducción definen la estrategia call-by-value.

$$V ::= \mathsf{true} \mid \mathsf{false} \mid \lambda x : \sigma . M$$

$$
\begin{aligned}
(\lambda x : \sigma . M) V &\to M\{x := V\} && (\textit{E-AppAbs} \text{ o } \beta)\\
\mathsf{if\ true\ then}\ M\ \mathsf{else}\ N &\to M && (\textit{E-IfTrue})\\
\mathsf{if\ false\ then}\ M\ \mathsf{else}\ N &\to N && (\textit{E-IfFalse})
\end{aligned}
$$

Si $M \to N$, entonces:

$$
\begin{aligned}
MO &\to NO && (\textit{E-App}_1 \text{ o } \mu)\\
VM &\to VN && (\textit{E-App}_2 \text{ o } \nu)\\
\mathsf{if}\ M\ \mathsf{then}\ O\ \mathsf{else}\ P &\to \mathsf{if}\ N\ \mathsf{then}\ O\ \mathsf{else}\ P && (\textit{E-If})
\end{aligned}
$$

Semántica operacional

Valores

Los valores son los resultados esperados de los programas. Se definen como los términos cerrados y bien tipados V producidos por la gramática de valores.

Semántica operacional

Valores

Los valores son los resultados esperados de los programas. Se definen como los términos cerrados y bien tipados V producidos por la gramática de valores.

Ejercicio: ¿Cuáles de los siguientes términos son valores?

a) if true then $(\lambda x : \mathsf{Bool}.x)$ else $(\lambda x : \mathsf{Bool}.\mathsf{false})$

Semántica operacional

Valores

Los valores son los resultados esperados de los programas. Se definen como los términos cerrados y bien tipados V producidos por la gramática de valores.

Ejercicio: ¿Cuáles de los siguientes términos son valores?

a) if true then $(\lambda x : \text{Bool}.x)$ else $(\lambda x : \text{Bool.false})$

b) $\lambda x : \text{Bool.false}$

Semántica operacional

Valores
Los valores son los resultados esperados de los programas. Se definen como los términos cerrados y bien tipados V producidos por la gramática de valores.

Ejercicio: ¿Cuáles de los siguientes términos son valores?

a) if true then $(\lambda x : \mathsf{Bool}.x)$ else $(\lambda x : \mathsf{Bool}.\mathsf{false})$
b) $\lambda x : \mathsf{Bool}.\mathsf{false}$
c) $(\lambda x : \mathsf{Bool}.x)$ false

Semántica operacional

Valores

Los valores son los resultados esperados de los programas. Se definen como los términos cerrados y bien tipados V producidos por la gramática de valores.

Ejercicio: ¿Cuáles de los siguientes términos son valores?

a) if true then $(\lambda x : \mathsf{Bool}.x)$ else $(\lambda x : \mathsf{Bool}.\mathsf{false})$

b) $\lambda x : \mathsf{Bool}.\mathsf{false}$

c) $(\lambda x : \mathsf{Bool}.x)$ false

d) true

Semántica operacional

Valores

Los valores son los resultados esperados de los programas. Se definen como los términos cerrados y bien tipados V producidos por la gramática de valores.

Ejercicio: ¿Cuáles de los siguientes términos son valores?

a) if true then $(\lambda x$: Bool.$x)$ else $(\lambda x$: Bool.false$)$

b) λx : Bool.false

c) $(\lambda x$: Bool.$x)$ false

d) true

e) if x then true else false

Semántica operacional

Valores

Los valores son los resultados esperados de los programas. Se definen como los términos cerrados y bien tipados V producidos por la gramática de valores.

Ejercicio: ¿Cuáles de los siguientes términos son valores?

a) if true then $(\lambda x : \mathsf{Bool}.x)$ else $(\lambda x : \mathsf{Bool}.\mathsf{false})$

b) $\lambda x : \mathsf{Bool}.\mathsf{false}$

c) $(\lambda x : \mathsf{Bool}.x)$ false

d) true

e) if x then true else false

f) $\lambda x : \mathsf{Bool}.(\lambda y : \mathsf{Bool}.x)$ false

Semántica operacional

Valores

Los valores son los resultados esperados de los programas. Se definen como los términos cerrados y bien tipados V producidos por la gramática de valores.

Ejercicio: ¿Cuáles de los siguientes términos son valores?

a) if true then $(\lambda x : \text{Bool}.x)$ else $(\lambda x : \text{Bool.false})$
b) $\lambda x : \text{Bool.false}$
c) $(\lambda x : \text{Bool}.x)$ false
d) true
e) if x then true else false
f) $\lambda x : \text{Bool}.(\lambda y : \text{Bool}.x)$ false
g) $\lambda x : \text{Bool} \to \text{Bool}.x$ true

Semántica operacional

Valores

Los valores son los resultados esperados de los programas. Se definen como los términos cerrados y bien tipados V producidos por la gramática de valores.

Ejercicio: ¿Cuáles de los siguientes términos son valores?

a) if true then $(\lambda x : \text{Bool}.x)$ else $(\lambda x : \text{Bool.false})$
b) $\lambda x : \text{Bool.false}$
c) $(\lambda x : \text{Bool}.x)$ false
d) true
e) if x then true else false
f) $\lambda x : \text{Bool}.(\lambda y : \text{Bool}.x)$ false
g) $\lambda x : \text{Bool} \to \text{Bool}.x$ true

Ejercicio: ¿Cuál es el resultado de evaluar las siguientes expresiones? ¿El resultado es siempre un valor? Escribir la reducción paso por paso.

a) $((\lambda x : \text{Bool}.\lambda y : \text{Bool}.\text{if } x \text{ then true else } y)$ false$)$ true

Semántica operacional

Valores

Los valores son los resultados esperados de los programas. Se definen como los términos cerrados y bien tipados V producidos por la gramática de valores.

Ejercicio: ¿Cuáles de los siguientes términos son valores?

a) if true then $(\lambda x : \text{Bool}.x)$ else $(\lambda x : \text{Bool.false})$
b) $\lambda x : \text{Bool.false}$
c) $(\lambda x : \text{Bool}.x)$ false
d) true
e) if x then true else false
f) $\lambda x : \text{Bool}.(\lambda y : \text{Bool}.x)$ false
g) $\lambda x : \text{Bool} \to \text{Bool}.x$ true

Ejercicio: ¿Cuál es el resultado de evaluar las siguientes expresiones? ¿El resultado es siempre un valor? Escribir la reducción paso por paso.

a) $((\lambda x : \text{Bool}.\lambda y : \text{Bool}.\text{if } x \text{ then true else } y)$ false$)$ true
b) $(\lambda x : \text{Bool}.\lambda y : \text{Bool} \to \text{Bool}.y(yx))((\lambda z : \text{Bool.true})$ false$)(\lambda w : \text{Bool}.w)$

Determinismo

Ejercicio: Probar que la semántica operacional de cálculo lambda con booleanos, con la estrategia call-by-value, está determinada.
Es decir, probar que si $M \to M_1$ y $M \to M_2$, entonces $M_1 = M_2$.

Extensión con números naturales

Sintaxis y tipado

Se extienden las gramáticas de términos y tipos de la siguiente manera:

$$\sigma ::= \ldots \mid \mathsf{Nat}$$
$$M ::= \ldots \mid \mathsf{zero} \mid \mathsf{succ}(M) \mid \mathsf{pred}(M) \mid \mathsf{isZero}(M)$$

Extensión con números naturales

Sintaxis y tipado

Se extienden las gramáticas de términos y tipos de la siguiente manera:

$$\begin{aligned} \sigma &::= \ldots \mid \mathsf{Nat} \\ M &::= \ldots \mid \mathsf{zero} \mid \mathsf{succ}(M) \mid \mathsf{pred}(M) \mid \mathsf{isZero}(M) \end{aligned}$$

Se extiende el sistema de tipado con las siguientes reglas:

$$\frac{}{\Gamma \vdash \mathsf{zero} : \mathsf{Nat}}\ \textit{T-Zero} \qquad \frac{\Gamma \vdash M : \mathsf{Nat}}{\Gamma \vdash \mathsf{succ}(M) : \mathsf{Nat}}\ \textit{T-Succ}$$

$$\frac{\Gamma \vdash M : \mathsf{Nat}}{\Gamma \vdash \mathsf{pred}(M) : \mathsf{Nat}}\ \textit{T-Pred} \qquad \frac{\Gamma \vdash M : \mathsf{Nat}}{\Gamma \vdash \mathsf{isZero}(M) : \mathsf{Bool}}\ \textit{T-IsZero}$$

Extensión con números naturales

Semántica operacional

Se extienden los valores de la siguiente manera:

$$V ::= \ldots \mid \mathsf{zero} \mid \mathsf{succ}(V)$$

Además, usamos la notación $\underline{n}$ para $\mathsf{succ}^n(\mathsf{zero})$ con $n \geq 0$.

Extensión con números naturales

Semántica operacional

Se extienden los valores de la siguiente manera:

$$V ::= \ldots \mid \mathsf{zero} \mid \mathsf{succ}(V)$$

Además, usamos la notación $\underline{n}$ para $\mathsf{succ}^n(\mathsf{zero})$ con $n \geq 0$.
Se extiende la semántica operacional con las siguientes reglas:

$$\mathsf{pred}(\mathsf{succ}(V)) \to V \qquad (\textit{E-PredSucc})$$

$$\mathsf{isZero}(\mathsf{zero}) \to \mathsf{true} \qquad (\textit{E-IsZero}_0)$$

$$\mathsf{isZero}(\mathsf{succ}(V)) \to \mathsf{false} \qquad (\textit{E-IsZero}_n)$$

Extensión con números naturales

Semántica operacional

Se extienden los valores de la siguiente manera:

$$V ::= \ldots \mid \mathsf{zero} \mid \mathsf{succ}(V)$$

Además, usamos la notación $\underline{n}$ para $\mathsf{succ}^n(\mathsf{zero})$ con $n \geq 0$.
Se extiende la semántica operacional con las siguientes reglas:

$$\mathsf{pred}(\mathsf{succ}(V)) \to V \qquad (\textit{E-PredSucc})$$

$$\mathsf{isZero}(\mathsf{zero}) \to \mathsf{true} \qquad (\textit{E-IsZero}_0)$$

$$\mathsf{isZero}(\mathsf{succ}(V)) \to \mathsf{false} \qquad (\textit{E-IsZero}_n)$$

Si $M \to N$, entonces:

$$\mathsf{succ}(M) \to \mathsf{succ}(N) \qquad (\textit{E-Succ})$$

$$\mathsf{pred}(M) \to \mathsf{pred}(N) \qquad (\textit{E-Pred})$$

$$\mathsf{isZero}(M) \to \mathsf{isZero}(N) \qquad (\textit{E-IsZero})$$

Extensión con números naturales

Ejercicio

a) Esta extensión ¿mantiene las propiedades de determinismo, preservación de tipos y progreso?

Extensión con números naturales

Ejercicio

a) Esta extensión ¿mantiene las propiedades de determinismo, preservación de tipos y progreso?

b) ¿Qué términos representan las expresiones $\underline{0}$, $\underline{1}$ y $\underline{2}$? ¿Cómo reducen?

Extensión con números naturales

Ejercicio

a) Esta extensión ¿mantiene las propiedades de determinismo, preservación de tipos y progreso?

b) ¿Qué términos representan las expresiones $\underline{0}$, $\underline{1}$ y $\underline{2}$? ¿Cómo reducen?

c) Demostrar los siguientes juicios de tipado, o explicar por qué no son válidos:

   * $\vdash (\lambda x : \mathsf{Nat}.\mathsf{succ}(x))\ \mathsf{zero} : \mathsf{Nat}$

Extensión con números naturales

Ejercicio

a) Esta extensión ¿mantiene las propiedades de determinismo, preservación de tipos y progreso?

b) ¿Qué términos representan las expresiones $\underline{0}$, $\underline{1}$ y $\underline{2}$? ¿Cómo reducen?

c) Demostrar los siguientes juicios de tipado, o explicar por qué no son válidos:

- $\vdash (\lambda x : \mathsf{Nat}.\mathsf{succ}(x))\ \mathsf{zero} : \mathsf{Nat}$
- $x : \mathsf{Bool} \vdash \mathsf{succ}(\mathsf{zero}) : \mathsf{Nat}$

Extensión con números naturales

Ejercicio

a) Esta extensión ¿mantiene las propiedades de determinismo, preservación de tipos y progreso?

b) ¿Qué términos representan las expresiones $\underline{0}$, $\underline{1}$ y $\underline{2}$? ¿Cómo reducen?

c) Demostrar los siguientes juicios de tipado, o explicar por qué no son válidos:

   * $\vdash (\lambda x : \mathsf{Nat}.\mathsf{succ}(x))\ \mathsf{zero} : \mathsf{Nat}$
   * $x : \mathsf{Bool} \vdash \mathsf{succ}(\mathsf{zero}) : \mathsf{Nat}$
   * $x : \mathsf{Bool} \vdash \mathsf{if}\ x\ \mathsf{then}\ x\ \mathsf{else}\ \mathsf{zero} : \mathsf{Nat}$

Extensión con números naturales

Ejercicio

a) Esta extensión ¿mantiene las propiedades de determinismo, preservación de tipos y progreso?

b) ¿Qué términos representan las expresiones $\underline{0}$, $\underline{1}$ y $\underline{2}$? ¿Cómo reducen?

c) Demostrar los siguientes juicios de tipado, o explicar por qué no son válidos:

- $\vdash (\lambda x : \mathsf{Nat}.\mathsf{succ}(x))\ \mathsf{zero} : \mathsf{Nat}$
- $x : \mathsf{Bool} \vdash \mathsf{succ}(\mathsf{zero}) : \mathsf{Nat}$
- $x : \mathsf{Bool} \vdash \mathsf{if}\ x\ \mathsf{then}\ x\ \mathsf{else}\ \mathsf{zero} : \mathsf{Nat}$

d) Escribir la reducción paso por paso de los siguientes términos:

- $\mathsf{isZero}(\mathsf{succ}(\mathsf{pred}(\mathsf{succ}(\mathsf{zero}))))$

Extensión con números naturales

Ejercicio

a) Esta extensión ¿mantiene las propiedades de determinismo, preservación de tipos y progreso?

b) ¿Qué términos representan las expresiones $\underline{0}$, $\underline{1}$ y $\underline{2}$? ¿Cómo reducen?

c) Demostrar los siguientes juicios de tipado, o explicar por qué no son válidos:

- $\vdash (\lambda x : \mathsf{Nat}.\mathsf{succ}(x))\ \mathsf{zero} : \mathsf{Nat}$
- $x : \mathsf{Bool} \vdash \mathsf{succ}(\mathsf{zero}) : \mathsf{Nat}$
- $x : \mathsf{Bool} \vdash \mathsf{if}\ x\ \mathsf{then}\ x\ \mathsf{else}\ \mathsf{zero} : \mathsf{Nat}$

d) Escribir la reducción paso por paso de los siguientes términos:

- $\mathsf{isZero}(\mathsf{succ}(\mathsf{pred}(\mathsf{succ}(\mathsf{zero}))))$
- $\mathsf{isZero}(\mathsf{pred}(\mathsf{succ}(\mathsf{pred}(\mathsf{zero}))))$

Extensión con números naturales

Ejercicio

a) Esta extensión ¿mantiene las propiedades de determinismo, preservación de tipos y progreso?

b) ¿Qué términos representan las expresiones $\underline{0}$, $\underline{1}$ y $\underline{2}$? ¿Cómo reducen?

c) Demostrar los siguientes juicios de tipado, o explicar por qué no son válidos:

- $\vdash (\lambda x : \mathsf{Nat}.\mathsf{succ}(x))\ \mathsf{zero} : \mathsf{Nat}$
- $x : \mathsf{Bool} \vdash \mathsf{succ}(\mathsf{zero}) : \mathsf{Nat}$
- $x : \mathsf{Bool} \vdash \mathsf{if}\ x\ \mathsf{then}\ x\ \mathsf{else}\ \mathsf{zero} : \mathsf{Nat}$

d) Escribir la reducción paso por paso de los siguientes términos:

- $\mathsf{isZero}(\mathsf{succ}(\mathsf{pred}(\mathsf{succ}(\mathsf{zero}))))$
- $\mathsf{isZero}(\mathsf{pred}(\mathsf{succ}(\mathsf{pred}(\mathsf{zero}))))$
- $\mathsf{isZero}(\mathsf{pred}(\mathsf{succ}(\mathsf{pred}(x))))$

Extensión con números naturales

Ejercicio

a) Esta extensión ¿mantiene las propiedades de determinismo, preservación de tipos y progreso?

b) ¿Qué términos representan las expresiones $\underline{0}$, $\underline{1}$ y $\underline{2}$? ¿Cómo reducen?

c) Demostrar los siguientes juicios de tipado, o explicar por qué no son válidos:
   - $\vdash (\lambda x : \mathsf{Nat}.\mathsf{succ}(x))\ \mathsf{zero} : \mathsf{Nat}$
   - $x : \mathsf{Bool} \vdash \mathsf{succ}(\mathsf{zero}) : \mathsf{Nat}$
   - $x : \mathsf{Bool} \vdash \mathsf{if}\ x\ \mathsf{then}\ x\ \mathsf{else}\ \mathsf{zero} : \mathsf{Nat}$

d) Escribir la reducción paso por paso de los siguientes términos:
   - $\mathsf{isZero}(\mathsf{succ}(\mathsf{pred}(\mathsf{succ}(\mathsf{zero}))))$
   - $\mathsf{isZero}(\mathsf{pred}(\mathsf{succ}(\mathsf{pred}(\mathsf{zero}))))$
   - $\mathsf{isZero}(\mathsf{pred}(\mathsf{succ}(\mathsf{pred}(x))))$

Regla opcional si queremos recuperar la propiedad de Progreso:

$$\mathsf{pred}(\mathsf{zero}) \to \mathsf{zero} \qquad\qquad (\textit{E-Pred}_0)$$

Notar que esto cambia la semántica.

Simplificando la escritura

Podemos definir macros para expresiones que vayamos a utilizar con frecuencia. Por ejemplo:

✨ $Id_\tau \stackrel{def}{=}$

Simplificando la escritura

Podemos definir macros para expresiones que vayamos a utilizar con frecuencia. Por ejemplo:

- ✨ $Id_\tau \stackrel{def}{=} \lambda x : \tau . x$
- ✨ and $\stackrel{def}{=}$

Simplificando la escritura

Podemos definir macros para expresiones que vayamos a utilizar con frecuencia. Por ejemplo:

- ✨ $\mathit{Id}_\tau \stackrel{def}{=} \lambda x \colon \tau . x$
- ✨ and $\stackrel{def}{=} \lambda x \colon \mathsf{Bool}. \lambda y \colon \mathsf{Bool}. \mathsf{if}\ x\ \mathsf{then}\ y\ \mathsf{else\ false}$

Cambiando reglas semánticas

Supongamos que agregamos la siguiente regla para las abstracciones:

Si $M \to N$, entonces:

$$\lambda x\colon \tau.M \to \lambda x\colon \tau.N \qquad (\zeta)$$

Cambiando reglas semánticas

Supongamos que agregamos la siguiente regla para las abstracciones:

Si $M \to N$, entonces:

$$\lambda x : \tau.M \to \lambda x : \tau.N \qquad (\zeta)$$

Ejercicio

1. Repensar el conjunto de valores para respetar esta modificación, pensar por ejemplo si λx: Bool.Id_{Bool} true es o no un valor.

Cambiando reglas semánticas

Supongamos que agregamos la siguiente regla para las abstracciones:

Si $M \to N$, entonces:

$$\lambda x : \tau.M \to \lambda x : \tau.N \qquad (\zeta)$$

Ejercicio

1. Repensar el conjunto de valores para respetar esta modificación, pensar por ejemplo si $\lambda x :$ Bool.*Id*$_{\text{Bool}}$ true es o no un valor. ¿Y $\lambda x :$ Bool.x?

Cambiando reglas semánticas

Supongamos que agregamos la siguiente regla para las abstracciones:

Si $M \to N$, entonces:

$$\lambda x : \tau.M \to \lambda x : \tau.N \qquad (\zeta)$$

Ejercicio

1. Repensar el conjunto de valores para respetar esta modificación, pensar por ejemplo si $\lambda x :$ Bool.*Id*$_{\text{Bool}}$ true es o no un valor. ¿Y $\lambda x :$ Bool.x?

   V ::= true | false | $\lambda x : \sigma.F$ | zero | succ(V), donde F es una forma normal.

2. ¿Qué reglas deberían modificarse para no perder el determinismo?

Cambiando reglas semánticas

Supongamos que agregamos la siguiente regla para las abstracciones:
Si $M \to N$, entonces:

$$\lambda x\colon \tau.M \to \lambda x\colon \tau.N \qquad (\zeta)$$

Ejercicio

1. Repensar el conjunto de valores para respetar esta modificación, pensar por ejemplo si $\lambda x\colon \mathsf{Bool}.\mathit{Id}_{\mathsf{Bool}}$ true es o no un valor. ¿Y $\lambda x\colon \mathsf{Bool}.x$?

   V ::= true | false | $\lambda x\colon \sigma.F$ | zero | succ(V), donde F es una forma normal.

2. ¿Qué reglas deberían modificarse para no perder el determinismo?

   $$(\lambda x : \sigma.F)V \to F\{x := V\} \qquad (\beta)$$

3. Utilizando la nueva regla y los valores definidos, reducir la expresión:
   $\lambda z\colon \mathsf{Nat} \to \mathsf{Nat}.(\lambda x\colon \mathsf{Nat} \to \mathsf{Nat}.x\ \underline{23})\lambda z\colon \mathsf{Nat}.\underline{0}$

Cambiando reglas semánticas

Supongamos que agregamos la siguiente regla para las abstracciones:

Si $M \to N$, entonces:

$$\lambda x : \tau . M \to \lambda x : \tau . N \qquad (\zeta)$$

Ejercicio

3. Utilizando la nueva regla y los valores definidos, reducir la expresión:

$$\begin{aligned} & \lambda z : \mathsf{Nat} \to \mathsf{Nat}.(\lambda x : \mathsf{Nat} \to \mathsf{Nat}.x\ \underline{23})\lambda z : \mathsf{Nat}.\underline{0} \\ \to_{\zeta,\beta}\ & \lambda z : \mathsf{Nat} \to \mathsf{Nat}.(\lambda z : \mathsf{Nat}.\underline{0})\ \underline{23} \\ \to_{\zeta,\beta}\ & \lambda z : \mathsf{Nat} \to \mathsf{Nat}.\underline{0} \end{aligned}$$

¿Qué se puede concluir? ¿Tiene sentido o no agregar esta regla?

Para la próxima clase: extensiones

Intenten hacer los ejercicios 20 y 21 de la guía 4 (extensiones con pares y uniones disjuntas) para la próxima clase práctica.

Continuará...

¿? ¿? ¿? ¿? ¿? ¿? ¿? ¿? ¿? ¿? ¿? ¿? ¿?

$$(\lambda x\colon \textit{Clase.fin}\ x)\ (\textit{Cálculo Lambda}\ I)$$