Análisis de Lenguajes de Programación

Lambda cálculo

22 de Septiembre de 2025

Historia

1932 Church construye un sistema formal para modelar la matemática, el cual consiste en un cálculo de funciones puras.

1935 Kleene y Rosser demuestran que el sistema formal definido por Church es inconsistente.

1936 Church usa el lambda cálculo para estudiar la computabilidad y resolver el problema de desición propuesto por Hilbert, probando que no existe un algoritmo que permita decidir si una fórmula de la lógica de primer orden es un teorema.

Historia

En paralelo, **Turing** presenta su máquina y la utiliza para demostrar que existían problemas que una máquina no podía resolver.

Church y Turing escriben la **tesis de Church-Turing** donde formulan hipotéticamente la equivalencia entre los conceptos de **función computable** y **máquina de Turing**.

1940 Church introdujo el **cálculo lambda simplemente tipado** que es computacionalmente menos poderoso que el **cálculo lambda**, pero lógicamente consistente.

¿Qué es?

- Es un sistema formal que permite representar todas las funciones computables.
- Provee un mecanismo para:
  - Construir funciones,
  - aplicarlas y
  - evaluarlas
- Muchos problemas de diseño e implementación de lenguajes son estudiados en λ-cálculo por la sencillez de este cálculo.
- El primer problema no computable fue descubierto con lambda cálculo.

Sintaxis

Se denota con Λ el conjunto λ-términos, definido inductivamente por las siguientes reglas:

$$\frac{x \in X}{x \in \Lambda} \qquad \frac{t \in \Lambda \quad u \in \Lambda}{(t\,u) \in \Lambda} \qquad \frac{x \in X \quad t \in \Lambda}{(\lambda\, x\,.\, t) \in \Lambda}$$

donde X es un conjunto infinito de identificadores.

Ejemplos: x , (xy), $(\lambda x.x)$, $(\lambda x.(\lambda y.x))$, $(\lambda x.(\lambda y.((zx)y)))$

Convenciones

- Se utilizarán letras mayúsculas para denotar λ-términos.
- La aplicación asocia a izquierda.

  Escribimos $M\,N\,T$ en lugar de $((MN)\,T)$
- La abstracción se extiende tanto como sea posible.

  Escribimos $\lambda\,x\,.\,M\;N$ en lugar de $(\lambda\,x\,.\,M\;N)$
- Se pueden juntar varios λ en uno solo.

  Escribimos $\lambda\,x_1\,x_2\,\ldots\,x_n\,.\,M$ en lugar de $(\lambda\,x_1\,(\lambda\,x_2\,(\ldots\,\lambda\,x_n\,.\,M)\ldots)$

Variables libres y ligadas

- $FV(e)$ es el conjunto de ocurrencias de **variables libres** en e, definido como:

$$\begin{array}{lcl} FV(x) & = & \{x\} \\ FV(\lambda x\,.\,M) & = & FV(M) - \{x\} \\ FV(M\,N) & = & FV(M) \cup FV(N) \end{array}$$

- $BV(e)$ es el conjunto de ocurrencias de **variables ligadas** en e, definido como:

$$\begin{array}{lcl} BV(x) & = & \emptyset \\ BV(\lambda x\,.\,M) & = & BV(M) \cup \{x\} \\ BV(M\,N) & = & BV(M) \cup BV(N) \end{array}$$

Variables libres y ligadas

- Llamamos **ocurrencia de ligadura** a la variable x que aparece en $\lambda x \ldots$

- Un **término cerrado** es un término que no contiene variables libres.

- **Ejercicio:** Dar las ocurrencias de variables libres, ligadas y de ligadura del término:

$$(\lambda y \,.\, y\, x\, (\lambda x \,.\, y\, (\lambda y \,.\, z)\, x))\, v\, w$$

Equivalencia sintáctica

- Denotamos la equivalencia sintáctica con $\equiv$, donde $M \equiv N$ sii M es exactamente el mismo término que N

- Sin embargo muy pocas veces queremos distinguir términos que difieren sólo en el nombre de variables ligadas.
(Ej: $\lambda x \,.\, x \;\;,\;\; \lambda y \,.\, y$)

Substitución

Sean M y N términos, se define $M[N/x]$ como el resultado de sustituir N por toda **ocurrencia libre** de x en M, cambiando el nombre de ocurrencias ligadas, si fuese necesario.

Definimos $M[N/x]$ por inducción sobre M como:

$$
\begin{array}{lcll}
x[N/x] & \equiv & N & \\
y[N/x] & \equiv & y & \\
(P\,Q)[N/x] & \equiv & (P[N/x]\,Q[N/x]) & \\
(\lambda x\,.\,P)[N/x] & \equiv & (\lambda x\,.\,P) & \\
(\lambda y\,.\,P)[N/x] & \equiv & (\lambda y\,.\,P) & x \notin FV(P) \\
(\lambda y\,.\,P)[N/x] & \equiv & (\lambda y\,.\,P[N/x]) & x \in FV(P) \ \wedge \ y \notin FV(N) \\
(\lambda y\,.\,P)[N/x] & \equiv & \lambda z\,.\,(P[z/y])[N/x] & x \in FV(P) \ \wedge \ y \in FV(N)
\end{array}
$$

donde $z \notin FV(N\,P)$ y $x \not\equiv y$

Ejercicios

Aplicar las siguientes sustituciones:

1. $(\lambda\, y\, .\, x\, (\lambda\, w\, .\, v\, w\, x))[(uv)/x]$
2. $(\lambda\, y\, .\, x\, (\lambda\, x\, .\, x))[(\lambda\, y\, .\, x\, y)/x]$
3. $(y\, (\lambda\, v\, .\, x\, v))[(\lambda\, y\, .\, v\, y)/x]$
4. $(\lambda\, x\, .\, x\, y)[(u\, v)/x]$

α equivalencia

- La relación $\equiv_\alpha$, expresa que los nombres de las variables ligadas no son relevantes para la equivalencia de términos.

- Sea P un término que contiene a $\lambda x.M$, donde $y \notin M$, la operación que consiste en cambiar $\lambda x.M$ por:

$$\lambda y.(M[y/x])$$

se llama α-**conversión** o **cambio de variable ligada**.

- Si P puede convertirse en Q por una serie finita de α-**conversiones** decimos que $P \equiv_\alpha Q$.

Ejemplo

$$
\begin{aligned}
\lambda x\, y\, .\, x\,(x\, y) \quad &\equiv \quad \lambda x\, .\, (\lambda y\, .\, x\,(x\, y)) \\
&\equiv_{\alpha} \quad \lambda x\, .\, (\lambda v\, .\, x\,(x\, v)) \\
&\equiv_{\alpha} \quad \lambda u\, .\, (\lambda v\, .\, u\,(u\, v)) \\
&\equiv \quad \lambda u\, v\, .\, u\,(u\, v)
\end{aligned}
$$

Propiedades de $\equiv_\alpha$

1. Si $P \equiv_\alpha Q$, entonces $FV(P) = FV(Q)$.

2. $\equiv_\alpha$ es una relación de equivalencia, es decir, para cualesquiera P, Q y R:

   $$P \equiv_\alpha P$$
   $$P \equiv_\alpha Q \;\Rightarrow\; Q \equiv_\alpha P$$
   $$P \equiv_\alpha Q \;\wedge\; Q \equiv_\alpha R \;\Rightarrow\; P \equiv_\alpha R$$

3. La relación $\equiv_\alpha$ es preservada por la sustitución:

   $$P \equiv_\alpha P' \;\wedge Q \equiv_\alpha Q' \;\Rightarrow\; P[Q/x] \equiv_\alpha P'[Q'/x]$$

Semántica

- ¿Cómo evaluar términos en λ-cálculo?
- Un paso de reducción será la aplicación de una abstracción.
- Una β-reducción establece que un término formado por la aplicación de una abstracción a un argumento (de la forma $(\lambda x.\, P)Q$), reduce al cuerpo de la abstracción donde la variable ligada se substituyó por el argumento (es decir, $P[Q/x]$).

β-reducción

Definiciones

1. Llamaremos β-**redex** a un término de la forma $(\lambda\, x.\, P)Q$ y al término $P[Q/x]$ su **contracción**.

2. Si al reemplazar un β-redex en un término P por su contracción obtenemos un término P' , decimos que P se β-**contrae** a P' y escribimos $P \to_\beta P'$.

3. Decimos que P β-**reduce** a Q si $P \to_\beta^* Q$, donde $\to_\beta^*$ es la clausura, reflexiva, transitiva de $\to_\beta$.

Ejemplos de reducción

$(\lambda x\,.\,(\lambda y\,.\,y\,x)\,z)\,v \to_\beta ((\lambda y\,.\,y\,x)\,z)[x/v] \equiv ((\lambda y\,.\,y\,v)\,z) \to_\beta$
$(y\,v)[z/y] \equiv z\,v$

$(\lambda x\,.\,(\lambda y\,.\,y\,x)\,z)\,v \to_\beta ((\lambda x\,.(\,y\,x)[z/y])\,v) \equiv ((\lambda x\,.\,z\,x)\,v) \to_\beta$
$(z\,x)[v/x] \equiv z\,v$

$(\lambda x\,.\,x\,x)(\lambda x\,.\,x\,x) \to_\beta (x\,x)[(\lambda x\,.\,x\,x)/x] \equiv (\lambda x\,.\,x\,x)(\lambda x\,.\,x\,x)$
$\to_\beta \ldots$

$(\lambda x\,.\,x\,x\,y)(\lambda x\,.\,x\,x\,y) \to_\beta (x\,x\,y)[(\lambda x\,.\,x\,x\,y)/x]$
$\equiv (\lambda x\,.\,x\,x\,y)(\lambda x\,.\,x\,x\,y)\,y$
$\to_\beta (\lambda x\,.\,x\,x\,y)(\lambda x\,.\,x\,x\,y)\,y\,y$
$\to_\beta \ldots$

Estretegias de reducción

- Existen diferentes **estrategias de reducción**, que básicamente describen un orden en que los β-redex son reducidos.
- Algunas de éstas estrategias tienen nombre:
  1. reducción normal
  2. *call-by-value (llamada por valor)*
  3. *call-by-name (llamada por nombre)*
- Definiremos una semántica operacional, donde las transiciones serán β-contracciones.

Semántica operacional

$$\frac{t_1 \to_\beta t_1'}{t_1\ t_2 \to_\beta t_1'\ t_2} \qquad \text{(E-App1)}$$

$$\frac{t_2 \to_\beta t_2'}{t_1\ t_2 \to_\beta t_1\ t_2'} \qquad \text{(E-App2)}$$

$$\frac{t \to_\beta t'}{\lambda x\,.\,t \to_\beta \lambda x\,.\,t'} \qquad \text{(E-Abs)}$$

$$(\lambda x\,.\,t_1)t_2 \to_\beta t_1[t_2/x] \qquad \text{(E-AppAbs)}$$

Forma normal β

Definición Una **forma normal** β o β-nf es un término que no contiene β-redex.

Si un término $P \to_\beta^* Q$, donde Q es una β-nf, decimos que Q es una forma normal β de P.

Por ejemplo, $z\,v$ es una β-nf de $(\lambda\, x\,.\,(\lambda\, y\,.\,y\,x)\,z)\,v$

Propiedades de $\rightarrow_\beta^*$

- Nada nuevo es introducido en una reducción:

$$P \rightarrow_\beta^* Q \;\Rightarrow\; FV(Q) \subseteq FV(P)$$

Observación: Pueden desaparecer variables libres durante la reducción. Por ejemplo: $(\lambda\, x\,.\,(\lambda\, y\,.\,y))\; z \rightarrow_\beta\; (\lambda\, y\,.\,y)$

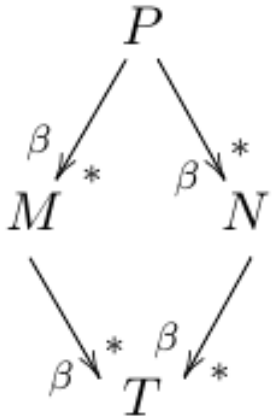
- La relación $\rightarrow_\beta^*$ es preservada por la sustitución:

$$P \rightarrow_\beta^* P' \;\wedge\; Q \rightarrow_\beta^* Q' \;\Rightarrow\; P[Q/x] \rightarrow_\beta^* P'[Q'/x]$$

Confluencia

Vimos que $\rightarrow_\beta$ es no-determinista, sin embargo si partimos de un término t y aplicamos β-contracciones que llevan a distintos términos, eventualmente se llegará a la misma forma normal.

Teorema (Church-Rossel) Si $P \rightarrow_\beta^* M$ y $P \rightarrow_\beta^* N$, entonces existe T tal que $M \rightarrow_\beta^* T$ y $N \rightarrow_\beta^* T$.



Colorario: Todo término tiene a lo sumo una forma normal (módulo $\equiv_\alpha$).

β-equivalencia

Definición: P es β-equivalente a Q (escribimos $P =_\beta Q$) sii Q puede ser obtenido partiendo de P y realizando una serie finita de β-contracciones, β-expansiones (β-contracciones inversas) y α-conversiones.

Obs: La relación $\rightarrow_\beta$ no es simétrica, pero $=_\beta$ sí.

Ejercicio: Probar que
$(\lambda\, x\, y\, z\, .\, x\, z\, y)(\lambda\, x\, y\, .\, x) =_\beta (\lambda\, x\, y\, .\, x)(\lambda x\, .\, x)$

Propiedades de $=_\beta$

Lema (Sustitución y $=_\beta$)

$$M =_\beta M' \ \wedge \ N =_\beta N' \Rightarrow M[N/x] =_\beta M'[N'/x]$$

Teorema (Church-Rosser para $=_\beta$)
Si $P =_\beta Q$, entonces existe T tal que:

$$P \to_\beta^* T \quad Q \to_\beta^* T$$

Extensionalidad

- El principio de extensionalidad para funciones matemáticas:

$$\forall\, x\,.\, f(x) = g(x) \Rightarrow f = g$$

no vale en el cálculo presentado. Por ej, $(\lambda\, x.y\, x) \neq_\beta y$

- Definición: Llamamos η-redex a un término de la forma $\lambda\, x\,.\, P\, x$, donde $x \notin FV(P)$, y a M su η-contracción. Escribimos:

$$\lambda\, x\,.\, P\, x \to_\eta P$$

Relación $=_{\beta\eta}$

- Definición: Decimos que P $\beta\eta$-contrae a Q sii $P \to_\beta Q$ o $P \to_\eta Q$ y escribimos:

$$P \to_{\beta\eta} Q$$

  Además $\to^*_{\beta\eta}$ es la clausura, reflexiva, transitiva de $\to_{\beta\eta}$

- Definición: Un término que no contiene $\beta\eta$-redex es una $\beta\eta$ forma normal (escribimos $\beta\eta$-*nf*)

- Teorema (Church-Rosser para $=_{\beta\eta}$)
  Si $P =_{\beta\eta} Q$, entonces existe T tal que:

$$P \to^*_{\beta\eta} T \quad Q \to^*_{\beta\eta} T$$

Ejemplos de reducción

Analizamos la reducción de los siguientes términos:

1. $\Omega \equiv (\lambda\, x\, .\, x\, x)(\lambda\, x\, .\, x\, x)$
2. $T2 \equiv (\lambda\, x\, .\, y)\, \Omega$
3. $T3 \equiv (\lambda\, x\, .\, x\, y)\, (\lambda\, x\, .\, x)$

Normalizante

- Definición: Un término P es **fuertemente normalizante** (P es SN) si toda secuencia de reducción que comienza en P es finita y termina en una forma normal.

- Definición: Un término P es **débilmente normalizante** (P es WN) si tiene forma normal.

Ejemplos: Ω no es SN ni WN, T_2 es WN, T_3 es SN y WN,

Reducción normal

- Un redex es **maximal** si no está contenido en algún redex.
- Un redex es **maximal izquierdo** si es el rexed maximal de más a la izquierda.
- La estrategia de reducción normal consiste en reducir siempre el término asociado al redex maximal izquierdo.
- Teorema: Si la reducción normal de un término P es infinita, P no tiene forma normal.

Formas neutrales y normales

Las siguientes categorías sintácticas, capturan la sintaxis de términos que no son abstracciones (*na*), formas normales (*nf*) y términos en forma neutral (*neu*), los cuales no son valores y no pueden reducirse.

$$
\begin{array}{lcl}
na & ::= & x \mid t_1\ t_2 \\
nf & ::= & \lambda\, x\, .\, nf \mid neu \\
neu & ::= & x \mid neu\ nf
\end{array}
$$

Reglas de evaluación para la reducción normal

$$\frac{na \to_\beta t_1'}{na\ t_2 \to_\beta t_1'\ t_2} \qquad \text{(E-App1)}$$

$$\frac{t_2 \to_\beta t_2'}{neu\ t_2 \to_\beta neu\ t_2'} \qquad \text{(E-App2)}$$

$$\frac{t \to_\beta t'}{\lambda x\,.\,t \to_\beta \lambda x\,.\,t'} \qquad \text{(E-Abs)}$$

$$(\lambda x\,.\,t_1)t_2 \to_\beta t_1[t_2/x] \qquad \text{(E-AppAbs)}$$

Resumen

- Podemos expresar en λ-cálculo cualquier función computable, las que pueden ser calculadas por una máquina de Turing.
- Se utiliza como modelo de los lenguajes funcionales, es un modelo simple.
- La evaluación de un lambda término consiste en eliminar los β-redexs aplicando β-constracciones.
- Una lambda expresión que no contiene β-redexs está en su forma normal.
- No toda expresión lambda tiene forma normal, pero si existe, es única.
- La estrategia de reducción normal nos aseguraba hallar la forma normal de expresiones que pueden reducir a una forma normal.