Lenguajes y Compiladores

Miguel Pagano

10 de mayo de 2024

Cálculo Lambda

El Cálculo Lambda

- Church (1930), continuado por Kleene, Rosser (1930s)
  Motivación original: problemas de fundamentación de la matemática
- **Como notación**: se usa para generar una expresión que denote una función, sin necesidad de dar nombre
  Por ejemplo: $f(x) = x + y$ se puede escribir

  $$f = \lambda x.\, x + y,$$

  con lo cual uno se puede referir a la función f mediante la expresión $\lambda x.x + y$, sin necesidad de ponerle nombre.
- Todas las funciones recursivas son definibles en el Cálculo Lambda.

Sintaxis abstracta del CL

$\langle \mathit{expr} \rangle$	$::=$		expresiones o términos
		$\langle \mathit{var} \rangle$	variables
		$\mid \langle \mathit{expr} \rangle \langle \mathit{expr} \rangle$	aplicación
		$\mid \lambda \langle \mathit{var} \rangle .\ \langle \mathit{expr} \rangle$	abstracción o expresión lambda

Convención: La aplicación asocia a izquierda. Por ejemplo,

$\lambda x.(\lambda y.xyx)x$ es lo mismo que

$$\lambda x.(\lambda y.(xy)x)x$$

Variables libres

En la abstracción

$$\lambda x.x + 2$$

se determina que estamos construyendo una función que varía de acuerdo a qué valor toma x. Es decir, la abstracción es un ***ligador***.

$$\begin{array}{lcl} FV(v) & = & \{v\} \\ FV(ee') & = & FV(e) \cup FV(e') \\ FV(\lambda v.e) & = & FV(e) \; - \; \{v\} \end{array}$$

Operador de sustitución

Conjunto de sustituciones: $\Delta = \langle \mathit{var} \rangle \to \langle \mathit{expr} \rangle$

Operador de sustitución: $\_/\_ \in \langle \mathit{expr} \rangle \times \Delta \to \langle \mathit{expr} \rangle$

$$\begin{array}{rcl} v/\delta & = & \delta v \\ (ee')/\delta & = & (e/\delta)(e'/\delta) \\ (\lambda v.e)/\delta & = & \lambda v_{new}.\, e/[\delta | v : v_{new}] \\ & & \text{donde } v_{new} \notin \bigcup_{w \in FV(e)-\{v\}} FV(\delta\, w) \end{array}$$

Notaciones

$I \colon \Delta$	Sustitución identidad
$I(x) = x$	
$v \mapsto e \colon \Delta$	Sustitución en un punto
$v \mapsto e = [I \mid v : e]$	

Conversión α

Renombre: Cambio en $\lambda v.e$ de la variable ligada v (y todas sus ocurrencias) por una variable v' que no ocurra libre en e:

$$\lambda v'.\, e/v \mapsto v'$$

donde $v' \notin FV(e)$.

α-**conversión**: Si e_1 se obtiene a partir de e_0 por cero o más renombres de ocurrencias de subfrases. También se dice que e_0 α-convierte a e_1.

Notación para expresiones α- convertibles: $e_0 \equiv e_1$

Podemos definir $\equiv$ mediante un conjunto de reglas.

Ejecución: Contracción β

Redex: Es una expresión de la forma $(\lambda v.e)e'$

Contracción β: Reemplaza en e_0 una ocurrencia de un redex $(\lambda v.e)e'$ por su contracción $(e/v \mapsto e')$, y luego efectúa cero o más renombres de cualquier subexpresión.

Notación: Si e_1 es el resultado de una contracción β de e_0, entonces escribimos

$$e_0 \to e_1$$

Forma normal: expresión sin redices.

Las formas normales representan configuraciones terminales.

Por eso la semántica operacional del cálculo lambda consiste en efectuar contracciones β hasta obtener formas normales.

$\rightarrow^*$ denota la clausura transitiva y refexiva de $\rightarrow$

(o sea, aplicar $\rightarrow$ cero o más veces)

Formalmente:

$e \rightarrow^* e'$ si y sólo si existen $e_0, \ldots, e_n$ (con $n \geq 0$) tales que

$$e = e_0 \rightarrow e_1 \rightarrow \ldots \rightarrow e_n = e'$$

Notar que si $n = 0$ entonces $e = e'$

1. ¿Cómo damos un modelo semántico del cálculo lambda?
2. ¿Qué propiedades queremos de un “modelo” del cálculo lambda?

Si existe forma normal, es única

Es consecuencia inmediata de:

Teorema de Church-Rosser Si $e \to^* e_0$ y $e \to^* e_1$, entonces existe e' tal que $e_0 \to^* e'$ y $e_1 \to^* e'$.

$$
\begin{tikzcd}
 & e \arrow[dl, "*"'] \arrow[dr, "*"] & \\
e_0 \arrow[dr, dashed, "*"'] & & e_1 \arrow[dl, dashed, "*"] \\
 & e' &
\end{tikzcd}
$$

Regla η

Un η-redex es una expresión de la forma $\lambda v.ev$, donde $v \notin FV\ e$

$$\frac{}{\lambda v.e\,v \to e}\ \text{si}\ v \notin FV\ e \qquad (\eta)$$

Evaluación Normal

Reglas para $\Rightarrow_N$

Regla para las formas canónicas

$$\frac{}{\lambda v.e \ \Rightarrow_N \ \lambda v.e}$$

Regla para la aplicación

$$\frac{e \ \Rightarrow_N \ \lambda v.e_0 \qquad (e_0/v \mapsto e') \ \Rightarrow_N \ z}{ee' \ \Rightarrow_N \ z}$$

Reglas para $\Rightarrow_E$

Regla para las formas canónicas

$$\frac{}{\lambda v.e \ \Rightarrow_E \ \lambda v.e}$$

Regla para la aplicación

$$\frac{e \ \Rightarrow_E \ \lambda v.e_0 \qquad e' \ \Rightarrow_E \ z' \qquad (e_0/v \mapsto z') \ \Rightarrow_E \ z}{ee' \ \Rightarrow_E \ z}$$

Sintaxis y semántica

1. ¿Cómo damos un modelo semántico del cálculo lambda?
2. ¿Qué propiedades queremos de un “modelo” del cálculo lambda?

Semántica Denotacional del Cálculo Lambda

Necesitamos poder modelar la aplicación y la abstracción:

$$\mathsf{abs}\colon (C \to C) \to C$$
$$\mathsf{app}\colon (C \to C) \times C \to C$$

tales que $\mathsf{app}(\langle \mathsf{abs}(f), x \rangle) = f(x)$.

Pero además permitimos la auto-aplicación, así que queremos:

$$C \underset{\overleftarrow{\psi}}{\overset{\xrightarrow{\phi}}{\cong}} C \to C$$

Sea $f : C \to C$ cualquier función, y definamos $p_f(x) = f(x(x))$.

Entonces, paradoja, toda función tiene punto fijo.

La función que mapea toda función a su punto fijo:

$$Y : (C \to C) \to C$$
$$Y(f) = p_f(p_f)$$

$$Y(f) = p_f(p_f) = f(p_f(p_f)) = p_f$$

¿Lo podemos hacer sintácticamente? Fijemos f:

$$p_f \doteq \lambda x.f\,(x\,x)$$

Entonces

$$p_f\,p_f \doteq (\lambda x.f\,(x\,x))(\lambda x.f\,(x\,x))$$

Ahora abstraigamos f:

$$Y \doteq \lambda f.(\lambda x.f\,(x\,x))(\lambda x.f\,(x\,x))$$

Dana Scott en 1969 mostró que podemos construir un dominio D_∞:

$$\phi\colon D_\infty \to [D_\infty \to D_\infty]$$
$$\psi\colon [D_\infty \to D_\infty] \to D_\infty$$

tales que

$$\phi \circ \psi = Id_{[D_\infty \to D_\infty]}$$
$$\psi \circ \phi = Id_{D_\infty}$$

Semántica Denotacional del Cálculo Lambda

Ambientes (Entornos): $Env = \langle var \rangle \to D_\infty$

Función semántica: $[\![\_]\!] \in \ \langle exp \rangle \to Env \to D_\infty$

Ecuaciones semánticas:

$$
\begin{aligned}
[\![v]\!]\eta &= \eta v \\
[\![e_0 e_1]\!]\eta &= \phi([\![e_0]\!]\eta)\ [\![e_1]\!]\eta \\
[\![\lambda v.e]\!]\eta &= \psi(\lambda d \in D_\infty.[\![e]\!][\eta|v:d])
\end{aligned}
$$

Propiedades de la semántica del CL

Teorema de Coincidencia:
Si $\eta w = \eta' w$ para todo $w \in FV\ e$, entonces $[\![e]\!]\eta = [\![e]\!]\eta'$.

Teorema de Renombre
Si $v_{new} \notin FV\ e - \{v\}$, entonces $[\![\lambda v_{new}.(e/v \mapsto v_{new})]\!] = [\![\lambda v.e]\!]$.

Sustituciones: $\Delta \ = \ \langle \mathit{var} \rangle \to \langle exp \rangle$

Teorema de Sustitución
Si $[\![\delta w]\!]\eta = \eta' w$ para todo $w \in FV\ e$, entonces $[\![e/\delta]\!]\eta = [\![e]\!]\eta'$.

Asumimos sin demostrar $[\![\Delta\Delta]\!]\eta = \bot$.

Corrección de reglas β y η

$$[\![(\lambda v.e)e']\!]\eta = [\![e/v \mapsto e']\!]\eta$$

$$[\![\lambda v.ev]\!]\eta = [\![e]\!]\eta, \text{ si } v \notin FV e$$

Es D_∞ modelo de la evaluación?

Cualquier abstracción $\lambda x.e$ es un valor (forma canónica) en ambos órdenes de evaluación.

En particular $\lambda x.\Delta\Delta$ es un valor, pero:

$$[\![\lambda x.\Delta\Delta]\!] = \perp$$

Necesitamos distinguir la denotación de valores de la denotación de términos que no tienen forma canónica.

Modelos para evaluación

Consideremos las reglas de evaluación de la aplicación:

$$\frac{e \Rightarrow \lambda v.e'' \qquad e' \Rightarrow z' \qquad (e''/v \rightarrow \mathbf{z'}) \Rightarrow z}{ee' \Rightarrow z} \text{(Eager)}$$

$$\frac{e \Rightarrow \lambda v.e'' \qquad (e''/v \rightarrow \mathbf{e'}) \Rightarrow z}{ee' \Rightarrow z} \text{(Normal)}$$

Notar que en un caso reemplazamos valores y en el otro expresiones (eventualmente divergentes).

Modelos para evaluación

En ambos casos queremos un dominio para dar sentido a las expresiones que tienen forma canónica y otro para las expresiones divergentes.

	Valores (V)	Términos (D)
Normal	$[D \to D]$	$V_\perp$
Eager	$[V \to D]$	$V_\perp$

Notemos que distinguimos la función constantemente $\perp$ de $\perp_D$.

Semántica Denotacional Normal

Queremos las siguientes ecuaciones de dominios:

$$D^N = V_\perp^N$$
$$V^N \cong [D^N \to D^N]$$

usaremos el siguiente isomorfismos y funciones:

$$\phi\colon V^N \to [D^N \to D^N]$$
$$\psi\colon [D^N \to D^N] \to V^N$$

con

$$\phi \circ \psi = Id_{[D^N \to D^N]}$$
$$\psi \circ \phi = Id_{V^N}$$
$$\iota_\perp\colon V^N \to D^N$$

Semántica Denotacional del Cálculo Lambda Normal

$$
\begin{aligned}
D^N &= V^N_\perp \\
V^N &\cong [D^N \to D^N]
\end{aligned}
$$

Ambientes: $Env = (\langle var \rangle \to D^N)$

Función semántica:

$$
\begin{aligned}
[\![\_]\!] &\in \langle exp \rangle \to Env \to D^N \\
[\![v]\!]\eta &= \eta v \\
[\![e_0 e_1]\!]\eta &= \phi_{\perp\!\!\perp}([\![e_0]\!]\eta)([\![e_1]\!]\eta) \\
[\![\lambda v.e]\!]\eta &= \iota_\perp \circ \psi\,(\lambda d \in D.[\![e]\!][\eta|v : d])
\end{aligned}
$$

Pregunta

1. ¿Si $[\![e_1]\!]\eta = \bot$, sucede que $[\![e_0\, e_1]\!]\eta = \bot$?
2. ¿Valen los teoremas de coincidencia, renombre y sustitución?

Corrección de evaluación
Si $e \Rightarrow^N z$, entonces $[\![e]\!]\eta = \iota_\bot v$ y además $[\![e]\!]\eta = [\![z]\!]\eta$.

Resulta crucial la validez de la regla β.

Invalidez de la regla η
¿Por qué?

Consideremos $(\lambda x.\lambda y.y)(\Delta\Delta)$.

Mientras que en la evaluación normal tiene un valor, no lo tiene en la evaluación eager.

Consecuentemente necesitamos otro modelo para esta última.

Semántica Denotacional Eager

Queremos las siguientes ecuaciones de dominios:

$$D^E = V^E_\perp$$
$$V^E \cong [V^E \to D^E]$$

usaremos el siguiente isomorfismos y funciones:

$$\phi\colon V^E \to [V^E \to D^E]$$
$$\psi\colon [V^E \to D^E] \to V^E$$

con

$$\phi \circ \psi = Id_{[V^E \to D^E]}$$
$$\psi \circ \phi = Id_{V^E}$$
$$\iota_\perp\colon V^E \to D^E$$

Semántica Denotacional del Cálculo Lambda Eager

$$D^E = V^E_\perp$$
$$V^E \cong [V^E \to D^E]$$

Ambientes: $Env = \langle var \rangle \to V^E$

Función semántica:

$$[\![\_]\!] \in \langle exp \rangle \to Env \to D^E$$
$$[\![v]\!]\eta = \iota_\perp(\eta\, v)$$
$$[\![e_0 e_1]\!]\eta = (\phi_{\perp\!\!\!\perp}([\![e_0]\!]\eta))_{\perp\!\!\!\perp}([\![e_1]\!]\eta)$$
$$[\![\lambda x.e]\!]\eta = \iota_\perp \circ \psi\,(\lambda v \in V^E.[\![e]\!][\eta|x:v])$$

Preguntas

- ¿Si $[\![e_1]\!]\eta = \perp$, sucede que $[\![e_0\, e_1]\!]\eta = \perp$?
- ¿Valen los teoremas de coincidencia, renombre y sustitución?

Corrección de evaluación
Si $e \Rightarrow^E z$, entonces $[\![e]\!]\eta = \iota_\perp v$ y además $[\![e]\!]\eta = [\![z]\!]\eta$.

Invalidez de la regla β
¿Por qué?

Invalidez de la regla η
¿Por qué?