

El CL puro sólo contiene variables, aplicaciones y la notación lambda (llamada abstracción).

0

$\langle exp \rangle ::=$	expresiones o términos
$\langle var \rangle$	variables
$ \langle exp \rangle \langle exp \rangle$	aplicación
$ \lambda \langle var \rangle . \langle exp \rangle$	abstracción o expresión lambda

Convención: La aplicación asocia a izquierda. En $\lambda v.e$, la primer ocurrencia de v es ligadora y su alcance es e . Por ejemplo,

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

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

Variables libres:

$$FV(v) = \{v\}$$

$$FV(ee') = FV(e) \cup FV(e')$$

$$FV(\lambda v.e) = FV(e) - \{v\}$$

Conjunto de sustituciones: $\Delta = \langle var \rangle \rightarrow \langle exp \rangle$

Operador sustitución: $-/_- \in \langle exp \rangle \times \Delta \rightarrow \langle exp \rangle$

0

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

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 0 o más renombres de ocurrencias de subfrases. También se dice que e_0 α -convierte a e_1 .

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

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

$$e_0 \rightarrow e_1$$

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 reflexiva de \rightarrow
(o sea, aplicar \rightarrow cero o más veces)

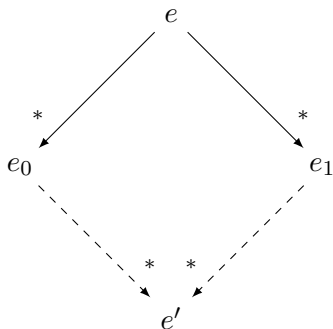
Formalmente:

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

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

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

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



Corolario 1. Salvo renombre, toda expresión tiene a lo sumo una forma normal.

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

$$\frac{}{\lambda v.ev \rightarrow e} \text{ si } v \notin FV\ e \quad (\eta)$$

La idea de ejecución (llamada evaluación) que se implementa habitualmente tiene las siguientes diferencias con la relación

- sólo se evalúan expresiones cerradas (es decir, sin variables libres)
- es determinística,
- no busca formas normales sino formas canónicas.

Evaluación (en orden) normal: lenguajes funcionales lazy (Haskell)

Evaluación eager o estricta: lenguajes estrictos (ML).

La noción de forma canónica depende de la definición de evaluación. Se define una noción de forma canónica para la evaluación normal, y otra para la evaluación eager. En el caso del cálculo lambda coinciden: **son las abstracciones**

Propiedad: Una aplicación cerrada no puede ser forma normal.

Corolario: Una expresión cerrada que es forma normal es también forma canónica.

Semántica natural o big-step: En este tipo de semántica, uno no describe un paso de ejecución, sino directamente una relación entre los términos y sus valores (que también son términos, son formas canónicas). Llamaremos \Rightarrow a esta relación.

Reglas para \Rightarrow_N

Regla para las formas canónicas

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

Regla para la aplicación

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

Reglas para \Rightarrow_E

Regla para las formas canónicas

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

Regla para la aplicación

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