

Sustitución en el Cálculo- λ Puro

José de Jesús Lavalle Martínez

Benemérita Universidad Autónoma de Puebla
Facultad de Ciencias de la Computación
Fundamentos de Lenguajes de Programación CCOS 255

Primavera 2020

- 1 Introducción
- 2 Variables acotadas y libres
- 3 Sustitución
- 4 Ejercicios

Observación 1

- Intuitivamente la abstracción lambda $(\lambda x.M)$ nos permite definir una función sin nombre con parámetro formal x y cuerpo M .
- Intuitivamente la aplicación nos permite aplicar una función a su argumento.
- Uniendo estas intuiciones, si tenemos $(\lambda x.M)N$, necesitamos saber como **sustituir** el argumento N por el parámetro formal x en el cuerpo de la función M , lo cual denotaremos como $M[x := N]$.

- Sabemos, por nuestra experiencia programando, que no importa el nombre que usemos para los parámetros formales de un procedimiento, método o función de cualquier lenguaje de programación.

- Sabemos, por nuestra experiencia programando, que no importa el nombre que usemos para los parámetros formales de un procedimiento, método o función de cualquier lenguaje de programación.
- En matemáticas, $f(x) = x + y$ y $g(z) = z + y$ si bien no son sintácticamente iguales, sí son equivalentes porque calculan la misma función o tienen el mismo significado, ambas regresan la suma de su parámetro real y y .

- Sabemos, por nuestra experiencia programando, que no importa el nombre que usemos para los parámetros formales de un procedimiento, método o función de cualquier lenguaje de programación.
- En matemáticas, $f(x) = x + y$ y $g(z) = z + y$ si bien no son sintácticamente iguales, sí son equivalentes porque calculan la misma función o tienen el mismo significado, ambas regresan la suma de su parámetro real y y .
- Para evaluar una función sustituimos el parámetro real por el parámetro formal en el cuerpo de la función, así tenemos:
 $f(2) \mapsto 2 + y$ y $g(2) \mapsto 2 + y$. Por lo tanto, da lo mismo $x \mapsto x + y$ que $z \mapsto z + y$.

Motivación II

- Pero debemos tener cuidado, no podemos renombrar el parámetro formal por cualquier variable, si cambiamos el parámetro formal de nuestro ejemplo por y obtenemos $y \mapsto y + y = 2y$, es decir, cambiamos el significado de la función.

- Pero debemos tener cuidado, no podemos renombrar el parámetro formal por cualquier variable, si cambiamos el parámetro formal de nuestro ejemplo por y obtenemos $y \mapsto y + y = 2y$, es decir, cambiamos el significado de la función.
- Así, en el cuerpo de una función debemos distinguir entre las variables que son iguales al parámetro formal y aquellas que son distintas al parámetro formal, a las primeras les llamamos variables acotadas y a las segundas variable libres.

- Pero debemos tener cuidado, no podemos renombrar el parámetro formal por cualquier variable, si cambiamos el parámetro formal de nuestro ejemplo por y obtenemos $y \mapsto y + y = 2y$, es decir, cambiamos el significado de la función.
- Así, en el cuerpo de una función debemos distinguir entre las variables que son iguales al parámetro formal y aquellas que son distintas al parámetro formal, a las primeras les llamamos variables acotadas y a las segundas variable libres.
- Lo que cambió el significado en nuestro ejemplo es que la variable libre y en el cuerpo de $x \mapsto x + y$ fue capturada al cambiar el nombre del parámetro formal x por y para obtener $y \mapsto y + y = 2y$, pasó de ser libre a ser acotada.

Definición 2

Sea $(\lambda x.M)$ una abstracción- λ , cualquier aparición de la variable x en M se dice que está **acotada** y que el **ámbito** de λx es M , o que M está en el ámbito de λx . Si una variable no está acotada se dice que está **libre**.

Definición 3

Dos términos- λ M y N se dice que son equivalentes- α si sólo varían en el nombre de sus variables acotadas. Se le llama **conversión- α** al proceso de renombrar las variables acotadas.

Observación 4

Note que lo que hace acotada a una variable x es que esté en el ámbito de una abstracción- λ , es decir, para realizar de manera correcta la conversión- α , debemos cambiar el nombre de la variable que aparece después de λ y todas sus ocurrencias en el cuerpo M de la abstracción- λ por una variable que no aparezca en el término- λ en cuestión.

Observación 5

En términos de la representación arborea de un término- λ , debemos cambiar el nodo etiquetado λx , por ejemplo, por λz y los descendientes de dicho nodo que tengan etiqueta x la debemos cambiar por z .

Observación 6

- 1 Recuerde que a $(\lambda x.M)$ le corresponde la siguiente representación arborea:

$$\begin{array}{c} \lambda x \\ | \\ M \end{array}$$

Observación 6

- 1 Recuerde que a $(\lambda x.M)$ le corresponde la siguiente representación arborea:

$$\begin{array}{c} \lambda x \\ | \\ M \end{array}$$

- 2 Recuerde que en la representación arborea de un término- λ las hojas necesariamente se corresponden con variables.

Observación 6

- 1 Recuerde que a $(\lambda x.M)$ le corresponde la siguiente representación arborea:



- 2 Recuerde que en la representación arborea de un término- λ las hojas necesariamente se corresponden con variables.
- 3 Por lo tanto, una variable x está acotada si al ascender desde ella hasta la raíz existe un nodo etiquetado λx .

Ejemplo 7

En el término- λ $(\lambda x.x)$, la aparición de x está acotada por λx .

Ejemplo 7

En el término- λ $(\lambda x.x)$, la aparición de x está acotada por λx .

Note que en su representación arborea, en el camino de x hacia la raíz hay un nodo etiquetado λx .

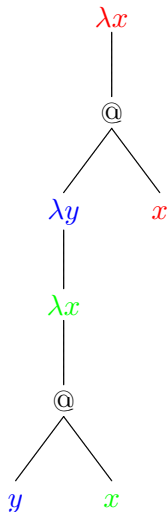


Ejemplo variables acotadas II

En el término- λ $(\lambda x. (\lambda y. \lambda x. yx)x)$

Ejemplo variables acotadas II

En el término- λ $(\lambda x.(\lambda y.\lambda x.yx)x)$

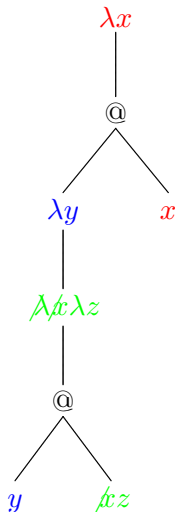


Ejemplo variables acotadas III

En el término- λ $(\lambda x. (\lambda y. \lambda x. yx)x) = (\lambda x. (\lambda y. \lambda z. yz)x)$

Ejemplo variables acotadas III

En el término- λ $(\lambda x.(\lambda y.\lambda x.yx)x) = (\lambda x.(\lambda y.\lambda z.yz)x)$

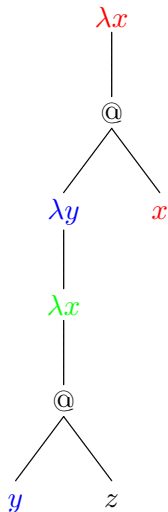


Ejemplo variables acotadas IV

En el término- λ $(\lambda x. (\lambda y. \lambda x. yz)x)$

Ejemplo variables acotadas IV

En el término- λ $(\lambda x.(\lambda y.\lambda x.yz)x)$



Definición 8

El conjunto de variables libres de un término- λ M , denotado por $FV(M)$, está definido inductivamente de la siguiente manera:

- 1 $FV(x) = \{x\}$,
- 2 $FV(MN) = FV(M) \cup FV(N)$,
- 3 $FV(\lambda x.M) = FV(M) \setminus \{x\}$.

Definición 8

El conjunto de variables libres de un término- λ M , denotado por $FV(M)$, está definido inductivamente de la siguiente manera:

- 1 $FV(x) = \{x\}$,
- 2 $FV(MN) = FV(M) \cup FV(N)$,
- 3 $FV(\lambda x.M) = FV(M) \setminus \{x\}$.

Un término- λ es **cerrado** si su conjunto de variables libres es vacío.

Ejemplos variables libres y acotadas

- La variable x subrayada está acotada en $(\lambda x.\underline{x})$.
- Ambas ocurrencias subrayadas de la variable x están acotadas en $(\lambda x.\underline{xx})$.
- La variable x subrayada está acotada en $(\lambda x.\underline{xy})$. La variable y subrayada está libre en $\lambda x.\underline{xy}$.
- La variable x subrayada está acotada en $x(\lambda x.\underline{x})$. La variable x subrayada está libre en $\underline{x}(\lambda x.x)$.
- La variable x subrayada está acotada en $(\lambda x.\underline{x})yx$. La variable x subrayada está libre en $(\lambda x.x)\underline{yx}$.

- Recuerde que iniciamos diciendo que para poder evaluar $(\lambda x.M)N$ necesitábamos saber como sustituir el argumento N por el parámetro formal x en el cuerpo de la función M , lo cual denotamos mediante $M[x := N]$.
- Note que en $M[x := N]$, x es el parámetro formal en $(\lambda x.M)$, pero que M ya no está en el ámbito de λx , por lo tanto todas las apariciones de x en M son libres.

Definición 9

La sustitución de N por x en M , denotada por $M[x := N]$ se define por inducción sobre la definición de términos- λ como sigue:

- 1 $x[x := N] = N$,

Definición 9

La sustitución de N por x en M , denotada por $M[x := N]$ se define por inducción sobre la definición de términos- λ como sigue:

- 1 $x[x := N] = N$,
- 2 $y[x := N] = y$ para $y \neq x$,

Definición 9

La sustitución de N por x en M , denotada por $M[x := N]$ se define por inducción sobre la definición de términos- λ como sigue:

- 1 $x[x := N] = N$,
- 2 $y[x := N] = y$ para $y \neq x$,
- 3 $(PQ)[x := N] = (P[x := N]Q[x := N])$,

Definición 9

La sustitución de N por x en M , denotada por $M[x := N]$ se define por inducción sobre la definición de términos- λ como sigue:

- 1 $x[x := N] = N$,
- 2 $y[x := N] = y$ para $y \neq x$,
- 3 $(PQ)[x := N] = (P[x := N]Q[x := N])$,
- 4 $(\lambda x.P)[x := N] = (\lambda x.P)$,

Definición 9

La sustitución de N por x en M , denotada por $M[x := N]$ se define por inducción sobre la definición de términos- λ como sigue:

- 1 $x[x := N] = N$,
- 2 $y[x := N] = y$ para $y \neq x$,
- 3 $(PQ)[x := N] = (P[x := N]Q[x := N])$,
- 4 $(\lambda x.P)[x := N] = (\lambda x.P)$,
- 5 $(\lambda y.P)[x := N] = (\lambda y.(P[x := N]))$ si $x \neq y$ y $y \notin FV(N)$,

Definición 9

La sustitución de N por x en M , denotada por $M[x := N]$ se define por inducción sobre la definición de términos- λ como sigue:

- 1 $x[x := N] = N$,
- 2 $y[x := N] = y$ para $y \neq x$,
- 3 $(PQ)[x := N] = (P[x := N]Q[x := N])$,
- 4 $(\lambda x.P)[x := N] = (\lambda x.P)$,
- 5 $(\lambda y.P)[x := N] = (\lambda y.(P[x := N]))$ si $x \neq y$ y $y \notin FV(N)$,
- 6 $(\lambda y.P)[x := N] = (\lambda z.(P[y := z][x := N]))$ si $x \neq y, z \notin FV(N) \cup FV(P)$ y $y \in FV(N)$.

Definición de sustitución en palabras

- 1 Sólo puedo sustituir un término- λ N por una variable x si la variable x está libre.

Definición de sustitución en palabras

- 2 No puedo sustituir un término- λ N por una variable x si lo que tengo es una variable y , aunque ésta esté libre.

- 3 Para sustituir un término- λ N por una variable x en una aplicación, hago la sustitución en ambos miembros de la aplicación.

- ④ No puedo sustituir un término- λ N por una variable x si la variable x es el parámetro formal de la abstracción- λ , dado que todas las apariciones de la variable x en el cuerpo P están acotadas.

- 5 Puedo sustituir un término- λ N por una variable x si la variable y es el parámetro formal de la abstracción- λ y si N no tiene ocurrencias libres de la variable y , porque esas ocurrencias se volverían acotadas después de la sustitución en P .

- 6 En caso de que N tenga ocurrencias libres de la variable y , puedo cambiar y por una nueva variable z tanto en la abstracción como en el cuerpo P y luego aplicarle al resultado la sustitución de N por x .

Ejemplo 10

- Correctas:

- $(\lambda y.xy)[x := y] = (\lambda z.yz)$,
- $(\lambda y.xy)[x := y] = (\lambda u.yu)$,
- $(x(\lambda y.xy))[x := y] = y(\lambda z.yz)$,
- $(\lambda y.xy)[x := z] = (\lambda y.zy)$.

- Incorrectas:

- $(\lambda y.xy)[x := y] = (\lambda y.yy)$,
- $(x(\lambda y.xy))[x := y] = y(\lambda y.yy)$.

Un ejemplo paso a paso

$$\begin{aligned} & (\overbrace{x}^P \overbrace{(\lambda y. xy)}^Q) [x := \overbrace{y}^N] \stackrel{3}{=} \\ & (x [x := \overbrace{y}^N]) ((\lambda y. xy) [x := y]) \stackrel{1}{=} \\ & y ((\lambda y. \overbrace{xy}^P) [x := \overbrace{y}^N]) \stackrel{6}{=} \\ & y (\lambda z. (xy [y := z] [x := y])) \stackrel{3}{=} \\ & y (\lambda z. ((x [y := z]) (y [y := z])) [x := y]) \stackrel{2}{=} \\ & y (\lambda z. (x (y [y := z])) [x := y]) \stackrel{1}{=} \\ & y (\lambda z. (xz) [x := y]) \stackrel{3}{=} \\ & y (\lambda z. (x [x := y] z [x := y])) \stackrel{1}{=} \\ & y (\lambda z. (yz [x := y])) \stackrel{2}{=} \\ & y (\lambda z. yz) \end{aligned}$$

- 1 Defina en ML la función `subs = fn : t * vars * t -> t`, sus parámetros se corresponden a los elementos de una sustitución $M[x := N]$, es decir el primer argumento de `subs` es M , el segundo es x y el tercero es N .
- 2 A los siete casos de prueba que les pondré en la página deben de hacer su traducción a términos- λ y deben hacer a mano las sustituciones, con ello ganarán experiencia en el manejo de los términos- λ , en su codificación en ML y sabrán si las respuestas de su función son correctas o incorrectas.

Ejemplo 11

$\text{subs}(1(x\ 2, a(v(x\ 1), v(x\ 2))), x\ 1, v(x\ 2))$; corresponde a

$$(\lambda x_2. x_1 x_2)[x_1 := x_2] \stackrel{6}{=}$$

$$(\lambda x_3. (x_1 x_2)[x_2 := x_3][x_1 := x_2]) \stackrel{3}{=}$$

$$(\lambda x_3. (x_1[x_2 := x_3]x_2[x_2 := x_3])[x_1 := x_2]) \stackrel{2}{=}$$

$$(\lambda x_3. (x_1 x_2[x_2 := x_3])[x_1 := x_2]) \stackrel{1}{=}$$

$$(\lambda x_3. (x_1 x_3)[x_1 := x_2]) \stackrel{3}{=}$$

$$(\lambda x_3. (x_1[x_1 := x_2]x_3[x_1 := x_2])) \stackrel{1}{=}$$

$$(\lambda x_3. (x_2 x_3[x_1 := x_2])) \stackrel{2}{=}$$

$$(\lambda x_3. (x_2 x_3)) = (\lambda x_3. x_2 x_3)$$

que corresponde a $\text{val it} = 1(x\ 3, a(v(x\ 2), v(x\ 3))) : \tau$

- 1 Noten que la definición de sustitución tiene 6 casos, 2 para variables, 1 para aplicación y 3 para abstracción- λ y sólo tenemos definidos tres constructores, así que tendrán que resolver los dos casos de variables y los tres de abstracción- λ en el cuerpo de la función, porque no pueden hacer “pattern matching” para el mismo constructor más de una vez.

- 2 Deben implementar la función `subs` tal y como está en la definición de substitución. Hay diferentes maneras de definir la substitución para el cálculo- λ pero la que les presento es la que se implementa más fácilmente.

- 1 Para poder implementar los casos 5 y 6, como indica la definición necesitan calcular las variables libres de un término- λ para ello usen listas y la definición de variables libres. También necesitarán la función que elimina un elemento de una lista y que verifica si un elemento pertenece a una lista.

- 2 Para el caso 6, la forma más fácil de saber cuál es la variable “nueva” z , es aprovechando que nuestras variables están numeradas, así se aseguran de no atrapar una variable libre encontrando el número de variable mayor que hay en las variables libres de $N \cup P$, la variable nueva será igual al número mayor encontrado más uno.