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

Contenido

Alonzo Church

El cálculo λ

- fue creado por Alonzo Church, presentado en el artículo "An Unsolvable Problem of Elementary Number Theory" que fue publicado en abril de 1936, fue parte de sus investigaciones sobre los Fundamentos de la Matemáticas o Metamatemática;

Alonzo Church

El cálculo λ

- es un modelo universal de cálculo;

Alonzo Church

El cálculo λ

- es un modelo universal de cálculo;
- se basa en la abstracción y aplicación de funciones;

Alonzo Church

El cálculo λ

- utiliza los conceptos de variable acotada y sustitución;

El cálculo λ

- es equivalente a las Máquinas de Turing, como lo demostró Turing es su famoso artículo "On Computable Numbers, with an Application to the Entscheidungsproblem" publicado en 1937.

Definición 1

El alfabeto del cálculo λ consta de los siguientes símbolos $\{\lambda, ., (,), x, y, z, x_1, y_1, z_1, \cdots\}$.

Alfabeto y sintaxis

Definición 1

El alfabeto del cálculo λ consta de los siguientes símbolos $\{\lambda, ., (,), x, y, z, x_1, y_1, z_1, \cdots\}$.

Definición 2

El conjunto de términos-λ, denotado por Λ, se define inductivamente mediante las siguientes cláusulas.

- una variable x es un término-λ,
- si M es un término-λ, entonces $(\lambda x.M)$ es un término-λ llamado **abstracción**-λ,
- si M y N son términos-λ, entonces (MN) es un término-λ llamado **aplicación**.

Paréntesis

Note que de acuerdo a la definición (2) abstracción y aplicación introducen paréntesis, lo que puede afectar la legibilidad de los términos-λ. Por ello, adoptaremos algunas convenciones para omitir algunos paréntesis, pero sólo lo haremos si su omisión no causa ambigüedad. Las convenciones que asumiremos son las siguientes:

1. Omitimos los paréntesis más externos, y escribimos MN en lugar de (MN).

Paréntesis

Note que de acuerdo a la definición (2) abstracción y aplicación introducen paréntesis, lo que puede afectar la legibilidad de los términos-λ. Por ello, adoptaremos algunas convenciones para omitir algunos paréntesis, pero sólo lo haremos si su omisión no causa ambigüedad. Las convenciones que asumiremos son las siguientes:

1. Omitimos los paréntesis más externos, y escribimos MN en lugar de (MN).
2. Se asume que la aplicación es asociativa a la izquierda. Esto es, escribimos (MNP) en lugar de $((MN)P)$. Note que los paréntesis in $M(NP)$ son necesarios.

Paréntesis

Note que de acuerdo a la definición (2) abstracción y aplicación introducen paréntesis, lo que puede afectar la legibilidad de los términos-λ. Por ello, adoptaremos algunas convenciones para omitir algunos paréntesis, pero sólo lo haremos si su omisión no causa ambigüedad. Las convenciones que asumiremos son las siguientes:

1. Omitimos los paréntesis más externos, y escribimos MN en lugar de (MN).
2. Se asume que la aplicación es asociativa a la izquierda. Esto es, escribimos (MNP) en lugar de $((MN)P)$. Note que los paréntesis in $M(NP)$ son necesarios.
3. Escribimos $(\lambda x.\lambda y.M)$ en lugar de $(\lambda x.(\lambda y.M))$.

Paréntesis

Note que de acuerdo a la definición (2) abstracción y aplicación introducen paréntesis, lo que puede afectar la legibilidad de los términos-λ. Por ello, adoptaremos algunas convenciones para omitir algunos paréntesis, pero sólo lo haremos si su omisión no causa ambigüedad. Las convenciones que asumiremos son las siguientes:

1. Omitimos los paréntesis más externos, y escribimos MN en lugar de (MN).
2. Se asume que la aplicación es asociativa a la izquierda. Esto es, escribimos (MNP) en lugar de $((MN)P)$. Note que los paréntesis in $M(NP)$ son necesarios.
3. Escribimos $(\lambda x.\lambda y.M)$ en lugar de $(\lambda x.(\lambda y.M))$.
4. Algunas veces se combinan λs consecutivos y escribimos $\lambda xy.M$ en lugar de $\lambda x.\lambda y.M$, o en general $\lambda x_1 \cdots x_n.M$ en lugar de $\lambda x_1. \cdots .\lambda x_n.M$.

Paréntesis

Note que de acuerdo a la definición (2) abstracción y aplicación introducen paréntesis, lo que puede afectar la legibilidad de los términos-λ. Por ello, adoptaremos algunas convenciones para omitir algunos paréntesis, pero sólo lo haremos si su omisión no causa ambigüedad. Las convenciones que asumiremos son las siguientes:

1. Omitimos los paréntesis más externos, y escribimos MN en lugar de (MN).
2. Se asume que la aplicación es asociativa a la izquierda. Esto es, escribimos (MNP) en lugar de $((MN)P)$. Note que los paréntesis in $M(NP)$ son necesarios.
3. Escribimos $(\lambda x.\lambda y.M)$ en lugar de $(\lambda x.(\lambda y.M))$.
4. Algunas veces se combinan λs consecutivos y escribimos $\lambda xy.M$ en lugar de $\lambda x.\lambda y.M$, o en general $\lambda x_1 \cdots x_n.M$ en lugar de $\lambda x_1. \cdots .\lambda x_n.M$.
5. La abstracción-λ se extiende a la derecha tanto como sea posible. Por ejemplo, escribimos $\lambda x.xx$ en lugar de $\lambda x.(xx)$ o en general $(\lambda x.MN)$ en lugar de $(\lambda x.(MN))$.

Ejemplo 3

Elimine del siguiente término-λ la mayor cantidad posible de paréntesis, siguiendo las convenciones establecidas.

$$((\lambda x.(\lambda y.(\lambda z.(y((xy)z)))))(\lambda v.(\lambda w.(vw))))$$

Solución:

$$((\lambda x.(\lambda y.(\lambda z.(y((xy)z)))))(\lambda v.(\lambda w.(vw)))) \stackrel{1}{=}$$

$$(\lambda x.(\lambda y.(\lambda z.(y((xy)z)))))(\lambda v.(\lambda w.(vw))) \stackrel{2}{=}$$

$$(\lambda x.(\lambda y.(\lambda z.(y(xyz)))))(\lambda v.(\lambda w.(vw))) \stackrel{5}{=}$$

$$(\lambda x.(\lambda y.(\lambda z.y(xyz))))(\lambda v.(\lambda w.(vw))) \stackrel{3}{=}$$

Ejemplo de eliminación de paréntesis II

$$(\lambda x.(\lambda y.\lambda z.y(xyz)))(\lambda v.(\lambda w.(vw))) \overset{3}{=}$$

$$(\lambda x.\lambda y.\lambda z.y(xyz))(\lambda v.(\lambda w.(vw))) \overset{4}{=}$$

$$(\lambda xyz.y(xyz))(\lambda v.(\lambda w.(vw))) \overset{5}{=}$$

$$(\lambda xyz.y(xyz))(\lambda v.(\lambda w.vw)) \overset{3}{=}$$

$$(\lambda xyz.y(xyz))(\lambda v.\lambda w.vw) \overset{4}{=}$$

$$(\lambda xyz.y(xyz))(\lambda vw.vw)$$

Representación arbórea de términos-λ

1. Las variables son las hojas de un árbol.

$$x$$

Representación arbórea de términos-λ

1. Las variables son las hojas de un árbol.

$$x$$

2. Una abstracción $\lambda x.M$ se representa como un árbol con un solo hijo.

$$\lambda x$$

$$M$$

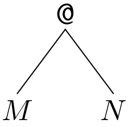
Representación arbórea de términos-λ

1. Las variables son las hojas de un árbol.

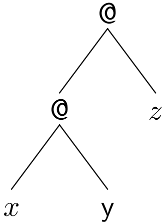
$$x$$

2. Una abstracción $\lambda x.M$ se representa como un árbol con un solo hijo.

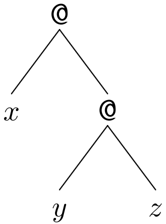
3. Una aplicación MN se representa como un árbol con dos hijos.



- xyz se representa mediante

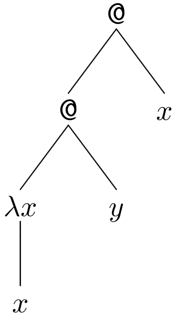


- $x(yz)$ se representa mediante



Ejemplos de representaciones arbóreas II

- A $(\lambda x.x)yx$ le corresponde el siguiente árbol



Un **subtérmino** es una parte de un término que corresponde a un subárbol del árbol sintáctico. Un término es un subtérmino de si mismo. Por ejemplo, M es un subtérmino de $\lambda x.M$, pero λx no es un subtérmino de $\lambda x.M$.

Ejercicios

1. Aplique, lo más que se pueda, las convenciones notacionales a los siguientes términos-λ:
    1. $(((\lambda x.(\lambda y.((xx)y)))(\lambda z.(\lambda u.(u(uz)))))(\lambda w.w))$
    2. $((\lambda x.(\lambda y.((((xy)z)u)(vw))))(\lambda x.(y(xx))))$
    3. $((\lambda x.x)((\lambda x.(\lambda y.((xy)y)))(\lambda x.(\lambda y.(x(xy))))))$
2. Escriba los siguientes términos-λ con todos los paréntesis y todas las λs:
    1. $(\lambda xy.x(yz))(\lambda x.yxx)$
    2. $(\lambda xy.xyyy)(\lambda x.zx)$
3. Represente con árboles los términos-λ de los dos ejercicios anteriores.
4. Construya un tipo en ML para los términos-λ, llámele `t` (te), sus constructores deben ser `v` (uve) para las variables, `l` (ele) para la abstracción-λ y `a` (a) para la aplicación. Las variables que usaremos serán $x_1, x_2, \cdots$, construya un tipo para ellas llamado `vars` con constructor `x` (equis).