

# TEX-LAMBDA

## Introducción

Este paquete permite escribir en el modo de matemáticas de L<sup>A</sup>T<sub>E</sub>X términos bien formados del cálculo lambda sin tipos.

Los comandos principales son `\lc{x}` y `\lc*{x}`. El primero traduce el término lambda  $x$  en código para el modo matemáticas de L<sup>A</sup>T<sub>E</sub>X sin abusos de notación, el segundo traduce a  $x$  de la manera mas concisa posible, utilizando los abusos de notación estándar en la literatura del cálculo lambda. En ambos casos,  $x$  puede ser escrito con o sin abuso de notación o una mescolanza de ambos estilos.

La sintáxis aceptada es:

- § Si  $x$  es una secuencia de caracteres sin espacios, entonces  $x$  es un término lambda aceptado (*átomo*).
- § Si  $x$  y  $y$  son términos lambda, entonces  $(x\ y)$  es un término lambda aceptado (*aplicación*).
- § Si  $x$  es una secuencia de caracteres sin espacios y  $y$  es un término lambda aceptado, entonces  $(\lambda x.y)$  es un término lambda aceptado (*abstracción*).

Los abusos de notación son:

- § Si  $x$  es una aplicación o una abstracción, se pueden ignorar los paréntesis.
- § Si  $x$  es una abstracción cuyo cuerpo es otra abstracción, se pueden agrupar los argumentos de ambas abstracciones, e.g  $(\lambda x.(\lambda y.M))$  es equivalente a  $(\lambda x\ y.M)$ .
- § Si  $x$  es una aplicación anidada con asociación a la izquierda, se pueden escribir los términos en las aplicaciones de manera consecutiva, e.g  $((a\ b)c)d$  es equivalente a  $(a\ b\ c\ d)$ .

## Ejemplos

### Átomos

Escribiendo `\lc{x}` se obtiene  $x$ .  
Escribiendo `\lc*{x}` se obtiene  $x$ .

### Abstracción lambda

Escribiendo `\lc{\lambda x.x}` se obtiene  $(\lambda x.x)$ .  
Escribiendo `\lc*{\lambda x.x}` se obtiene  $\lambda x.x$ .

### Aplicación lambda

Escribiendo `\lc{x\ y\ z}` se obtiene  $((x\ y)\ z)$ .  
Escribiendo `\lc*{x\ y\ z}` se obtiene  $x\ y\ z$ .

## Numerales de Church

Escribiendo `\lc{\f x.f(f(f(f(f x))))}` se obtiene  $(\lambda f.(\lambda x.(f (f (f (f (f x))))))$ .

Escribiendo `\lc*{\f x.f(f(f(f(f x))))}` se obtiene  $\lambda f x.f (f (f (f (f x))))$ .

## Términos lambda variados

Escribiendo `\lc{x y z (y x)}` se obtiene  $((x y) z) (y x)$ .

Escribiendo `\lc*{((x y) z) (y x)}` se obtiene  $x y z (y x)$ .

Escribiendo `\lc{\x.u x y}` se obtiene  $(\lambda x.((u x) y))$ .

Escribiendo `\lc*{(\x.((u x) y))}` se obtiene  $\lambda x.u x y$ .

Escribiendo `\lc{\u.u(\x.y)}` se obtiene  $(\lambda u.(u (\lambda x.y)))$ .

Escribiendo `\lc*{(\u.(u (\x.y)))}` se obtiene  $\lambda u.u (\lambda x.y)$ .

Escribiendo `\lc{(\u.v u u)z y}` se obtiene  $((\lambda u.((v u) u)) z) y$ .

Escribiendo `\lc*{(((\u.((v u) u)) z) y)}` se obtiene  $(\lambda u.v u u) z y$ .

Escribiendo `\lc{u x(y z)(\v.v y)}` se obtiene  $((u x) (y z)) (\lambda v.(v y))$ .

Escribiendo `\lc*{(((u x)(y z))(\v.(v y)))}` se obtiene  $u x (y z) (\lambda v.v y)$ .

Escribiendo `\lc{(\x y z.x z(y z))u v w}` se obtiene  $((((\lambda x.(\lambda y.(\lambda z.((x z) (y z)))) u) v) w)$ .

Escribiendo `\lc*{((((\x.(\y.(\z.((x z)(y z))))u)v)w)}` se obtiene  $(\lambda x y z.x z (y z)) u v w$ .

## Estilos

Para obtener diferentes estilos de términos, se puede utilizar el comando `\lca{x}{args}`, donde  $x$  es un término lambda como en los anteriores comandos y  $args$  son las banderas (o *flags*) que determinan el formato del término.

Las banderas admitidas son `s`, `v`, `l` y `d`. Si ejecutas el comando `./texlambda --help` obtendrás la siguiente descripción de las banderas:

TeX-LaMbDa [ <option> ... ] <str>

where <option> is one of

-s, --spaced : Spaced terms mode - Introduces spacing

-v, --bold-variables : Bold variables mode - Make variable names bold

-l, --bold-lambdas : Bold lambdas mode - Makes lambdas bold

-d, --bold-dots : Bold dots mode - Makes dots bold

-e, --explicit : Explicit mode - Removes abuse of notation

--help, -h : Show this help

-- : Do not treat any remaining argument as a switch (at this level)

Multiple single-letter switches can be combined after one '-'; for

example: '-h-' is the same as '-h --'

El modo explícito es controlado por el modificador estrella en los comandos `lc` y `lca`, así que no debes utilizar la bandera `e`.

## Ejemplos de modificación de estilos

Por ejemplo, para obtener “*negritas*” en las lambdas y puntos, se utiliza el comando `\lca{x}{ld}`, también sirve usar como banderas `dl` ya que el orden no importa:

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

Si queremos tener únicamente los átomos en “*negritas*” se escribe `\lca{x}{v}`:

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

La versión no explícita de este término sería `\lca*{x}{v}`:

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

Y si deseamos un término lambda mas espaciado se puede utilizar `\lca*{x}{vs}`:

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

Por el momento es lo único que puede estilizar el programa `texlambda`, sin embargo estoy trabajando en poder poner en negritas los paréntesis y poder realizar con comandos de latex y de manera declarativa  $\beta$ -reducción y  $\alpha$ -conversión.

## Operaciones y equivalencias

El paquete `TEX- $\mathcal{LM}^{\mathcal{B}}\mathcal{D}_{\lambda}$`  también tiene algunos comandos para escribir operaciones y equivalencias utilizadas frecuentemente en la literatura. Por el momento son:

§ `\betaredu`  $\rightarrow$   $\beta$ -reducción.

§ `\alphaconv`  $\rightarrow$   $\alpha$ -conversión.

§ `\synteq`  $\rightarrow$   $\equiv$ .

§ `\termilen{\lca{\x.x}}`  $\rightarrow$   $\|(\lambda x.x)\|$ .

§ `\alphacong`  $\rightarrow$   $\alpha$ -congruente.

Hacen falta muchas para completar las que utilizaré en mi tesis de licenciatura, sin embargo, aún no establezco una notación fija para las operaciones.