

logicDG Package

David Gómez

Índice

1. Introducción	1
2. Comandos	1
3. Entornos	2

1. Introducción

Esta librería tien como objetivo facilitar la escritura de proposiciones, demostraciones y procesos los cuales involucren proposiciones lógicas. Se encuentran en este entornos para escribir demostraciones a dos columnas y derivaciones, junto con comandos para la lógica de primer orden.

2. Comandos

Los comandos incluidos son los siguientes:

1. `\To` y `\Gets`: Haciendo referencia a los comandos `\to` (\rightarrow) y `\gets` (\leftarrow), los comandos presentados producen las versiones “robustas” de estas flechas.

sintáxis	salida
<code>\$p \to q\$</code>	$p \Rightarrow q$
<code>\$p \Gets q\$</code>	$p \Leftarrow q$.

2. `\Exists` y `\Forall`: Haciendo referencia a los comandos `\exists` (\exists) y `\forall` (\forall), los comandos presentados dan una forma sencilla de escribir una proposición de la lógica de primer orden. Ambos comandos tratan con tres argumentos, dos mandatorios y uno opcional de la siguiente forma

sintáxis	salida
<code>\$\Exists{a}{b}\$</code>	$(\exists a : b)$
<code>\$\Exists{a}[b]{c}\$</code>	$(\exists a b : c)$
<code>\$\Forall{a}{b}\$</code>	$(\forall a : b)$
<code>\$\Forall{a}[b]{c}\$</code>	$(\forall a b : c)$

Los delimitadores en estos comandos van a corresponder al tamaño de sus argumentos. Esto es más facil de ver con un ejemplo.

`\[\Exists{x}[x \in \mathbb{R}]{\int_0^x t \, dt = \frac{1}{2}} \]`

$$\left(\exists x \mid x \in \mathbb{R} : \int_0^x t \, dt = \frac{1}{2} \right)$$

3. `\theo`: Este comando simplifica la escritura para expresar que una proposición es teorema o cierta, bajo cierto lenguaje y suposiciones:

sintaxis	salida
<code>\$\$\theo{a}{b}\$\$</code>	$\vdash_a b$
<code>\$\$\theo[a]{b}{c}\$\$</code>	$a \vdash_b c$

3. Entornos

Los entornos incluidos son los siguientes:

1. `twocol`: Este entorno es usado para demostraciones a dos columnas. Cuenta con tres argumentos opcionales delimitados por `<>`, `()` y `[]` en ese mismo orden. La sintaxis de inicio es la siguiente: `\begin{twocol}<a>(b)[c]` donde **a** corresponde a un multiplicador, el cual cambia la distancia entre lineas durante el uso del entorno. Su valor predeterminado es 1.2. **b** corresponde a la distancia que se quiere entre las dos columnas (proposición y justificación). Su valor predeterminado es 3pt. **c** corresponde al número del que se quiere comenzar a contar los pasos. Su valor predeterminado es 0.

Por ejemplo

```
\begin{twocol}
  p \To q & Hipótesis\\
  p      & Hipótesis\\
  q      & (MPP 1,2)
\end{twocol}
```

resulta en

0. $p \Rightarrow q$ Hipótesis
1. p Hipótesis
2. q (MPP 1,2)

Usando los argumentos opcionales

```
\begin{twocol}<0.9>(5pt)[2]
  p \To q & Hipótesis\\
  p      & Hipótesis\\
  q      & (MPP 1,2)
\end{twocol}
```

El espaciado entre lineas es menor, el espaciado entre columnas es mayor y los pasos empiezan desde 2.

2. $p \Rightarrow q$ Hipótesis
3. p Hipótesis
4. q (MPP 1,2)

2. **derivation**: Este entorno es usado para demostraciones a modo de derivación, sin embargo, también se presta para cadenas de igualdades o procesos con pasos secuenciales similares. Este entorno viene de la mano con dos comandos, `\wff` (well formed formula) y `why`. El entorno es una matriz con columnas alineadas a la izquierda. El nombre se da simplemente para hacer claridad en el código y diferenciarlo de una matriz usual. Los comandos mencionados juegan con la columna sobre la que se escribe para dar el resultado buscado. El comando `why` cuenta con un argumento mandatorio, el cual es la justificación y uno opcional el cual hace referencia a la conexión entre los pasos (equivalencia, implicación, igualdad); el comando `wff` cuenta con un argumento mandatorio el cual hace referencia a la fórmula o paso que se realiza. El entorno cuenta con dos argumentos opcionales delimitados por (`<>`) y (`[]`) respectivamente. La sintaxis de inicio es `\begin{derivation}<a>[b]` donde `a` corresponde al multiplicador de la separación entre filas y `b` a la separación entre columnas.

```
\[
\begin{derivation}
  \wff{ p \land (p \To q) }\\
  \why{ usando $(p \To q)\equiv(\lnot p \lor q)$ }\\
  \wff{ p \land (\lnot p \lor q) }\\
  \equiv\\
  \wff{ (p \land \lnot p) \lor (p \land q) }
  \equiv\\
  \wff{ p \land q }\\
  \why[\To]{ usando $p \land q \To q$ }\\
  \wff{ q }
\end{derivation}
\]
```

$$\begin{aligned}
 & p \wedge (p \Rightarrow q) \\
 \equiv & \quad \langle \text{ usando } (p \Rightarrow q) \equiv (\neg p \vee q) \rangle \\
 & p \wedge (\neg p \vee q) \\
 \equiv & \\
 & (p \wedge \neg p) \vee (p \wedge q) \equiv \\
 & p \wedge q \\
 \Rightarrow & \quad \langle \text{ usando } p \wedge q \Rightarrow q \rangle \\
 & q
 \end{aligned}$$

Para ver los argumentos opcionales y un uso fuera de las derivaciones lógicas...

```
\[
\begin{derivation}<1.5>[5pt]
  \wff{ \int \frac{1}{1-x^2} \text{d}x }\\
  =\\
\end{derivation}
\]
```

```

\wff{ \int \frac{1}{(1-x)(1+x)} \text{dx} }\\
\why[=]{ $A(1+x) + B(1-x) = 1$ luego $A = B = 1/2$ }\\
\wff{ \frac{1}{2} \left( \int \frac{1}{1+x} + \frac{1}{1-x} \text{dx} \right) }\\
\why[=]{ Como $1=(1+x)' = -(1-x)'$ }\\
\wff{ \frac{1}{2} [\ln|1+x| - \ln|1-x|] + C }
\end{derivation}
\]

```

$$\begin{aligned}
& \int \frac{1}{1-x^2} dx \\
&= \\
& \int \frac{1}{(1-x)(1+x)} dx \\
&= \langle A(1+x) + B(1-x) = 1 \text{ luego } A = B = 1/2 \rangle \\
& \frac{1}{2} \left(\int \frac{1}{1+x} + \frac{1}{1-x} dx \right) \\
&= \langle \text{Como } 1 = (1+x)' = -(1-x)' \rangle \\
& \frac{1}{2} [\ln|1+x| - \ln|1-x|] + C
\end{aligned}$$

3. **longderivation**: Este entorno es una extensión del anterior, permitiendo cadenas mucho más largas, pues está basado en **longtable**, un entorno hecho para hacer tablas las cuales puedan sobrepasar el largo de una página. Esto es, no se generará error al hacer una cadena de pasos tal que exceda el largo de una página. También es útil cuando no se quiere iniciar la cadena en la siguiente página. Cuenta con los mismos argumentos que **derivation**.