logicDG Package

David Gómez

Índice

1. Introducción

Esta librería tiene 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

2.1. To y Gets

Haciendo referencia a los comandos `$\to$` ($\to$) y `$\gets$` ($\leftarrow$), los comandos presentados producen las versiones "robustas" de estas flechas.

sintáxis	salida
`$p \To q$`	$p \Rightarrow q$
`$p \Gets q$`	$p \Leftarrow q$.

2.2. 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
`$\Exists{a}{b}$`	$(\exists a \mid: b)$
`$\Exists{a}[b]{c}$`	$(\exists a \mid b : c)$
`$\Forall{a}{b}$`	$(\forall a \mid: b)$
`$\Forall{a}[b]{c}$`	$(\forall a \mid b : c)$

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

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

$$\left(\exists x \,\middle|\, x \in \mathbb{R} : \int_0^x t \,\mathrm{d}t = \frac{1}{2}\right)$$

2.3. theo

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

sintáxis	salida
`$\theo{a}{b}$`	$\vdash_{\mathrm{a}} b$
`$\theo[a]{b}{c}$`	$a \vdash_{\mathrm{b}} c$

3. Entornos

3.1. twocol

Este entorno es usado para demostraciones a dos columnas. Cuenta con tres argumentos opcionales delimitados por <>, () y [] en ese mismo orden. La sintáxis de inicio es `\begin{twocol}<a>(b)[c]`, donde `a` corresponde al multiplicador de 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; y `c` corresponde al número del que se quiere comenzar a contar los pasos. Su valor predeterminado es `0`.

sintáxis

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

salida

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

Usando los argumentos opcionales...

sintáxis

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

salida

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

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

3.2. 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 conección entre los pasos (equivalencia, implicación, igualdad), su sintáxis es `\why[a]{b}`, donde `a` corresponde al conector entre pasos, este tiene como valor predeterminado ($\equiv$). El argumento `b` es el texto que se desea escribir entre dos pasos conectados con `a`; 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 sintáxis de inicio es `\begin{derivation}<a>[b]` donde `a` corresponde al multiplicador de la distancia entre lineas. Su valor predeterminado es 1.2; y `b` a la separación entre columnas. Su valor predeterminado es 0pt.

sintáxis

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

salida

$$
\begin{derivation}
  p \land (p \Rightarrow q) \\
\equiv \quad \langle\ \text{usando } (p \Rightarrow q) \equiv (\neg p \lor q)\ \rangle \\
  p \land (\neg p \lor q) \\
\equiv \\
  (p \land \neg p) \lor (p \land q) \equiv \\
  p \land q \\
\Rightarrow \quad \langle\ \text{usando } p \land q \Rightarrow q\ \rangle \\
  q
\end{derivation}
$$

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

sintáxis

```
\[
\begin{derivation}<1.5>[5pt]
    \wff{(A \cup B) \cap (A \cup B^c)}\\
  =\\
    \wff{((A\cup B)\cap A)\cup((A\cup B)\cap B^c)}\\
  \why[=]{Como $A\subseteq(A\cup B)$}\\
    \wff{A \cup (A\cap B^c) \cup (B\cap B^c)}\\
  \why[=]{Como $A\cap B^c \subseteq A$}\\
    \wff{A}
\end{derivation}
\]
```

salida

$$
\begin{derivation}<1.5>[5pt]
    (A \cup B) \cap (A \cup B^c)\\
  =\\
    ((A\cup B)\cap A)\cup((A\cup B)\cap B^c)\\
  \why[=]{Como $A\subseteq(A\cup B)$}\\
    A \cup (A\cap B^c) \cup (B\cap B^c)\\
  \why[=]{Como $A\cap B^c \subseteq A$}\\
    A
\end{derivation}
$$

3.3. 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`.

4. Argumentos Opcionales

4.1. round

Este argumento cambia los delimitadores del comando why por paréntesis comunes.

```
\documentclass{article}

\usepackage[round]{logicDG}

\begin{document}
\[
\begin{derivation}
    \wff{a}\\
  \why{texto}\\
    \wff{b}
\end{derivation}
\]
\end{document}
```

$$
\begin{derivation}
    a\\
  \why{texto}\\
    b
\end{derivation}
$$

4.2. square

Este argumento cambia los delimitadores del comando why por paréntesis cuadrados.

```
\documentclass{article}

\usepackage[square]{logicDG}

\begin{document}
\[
\begin{derivation}
    \wff{a}\\
  \why{texto}\\
    \wff{b}
\end{derivation}
\]
\end{document}
```

$$\begin{array}{l} a \\ \equiv \quad [\,texto\,] \\ b \end{array}$$

4.3. arrows

Este argumento cambia el valor predeterminado del argumento opcional en why por ($\Leftrightarrow$).

```
\documentclass{article}

\usepackage[arrows]{logicDG}

\begin{document}
\[
\begin{derivation}
    \wff{a}\\
  \why{texto}\\
    \wff{b}
\end{derivation}
\]
\end{document}
```

$$\begin{array}{l} a \\ \Leftrightarrow \quad \langle\ \text{texto}\ \rangle \\ b \end{array}$$