EuroTeX 2001

DCpic, Commutative Diagrams in a (La)TeX Document

Pedro Quaresma

Centro de Informática e Sistemas da Universidade de Coimbra

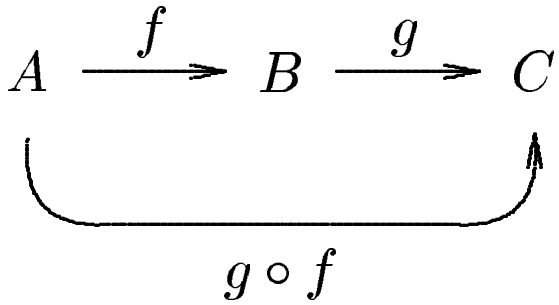
Departamento de Matemática, FCTUC

pedro@mat.uc.pt

DCpic is a package of PiCTeX macros for graphing (Commutative) Diagrams in a (La)TeX or ConTeXt document.

Commutative Diagrams are a kind of graphs which are widely used in Category Theory.

For example, the diagram:



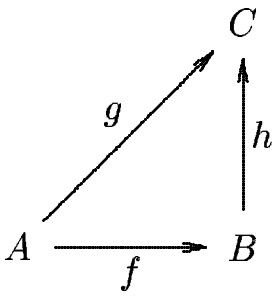
express the fact that we have arrow composition.

In a commutative diagrams package a user seeks the simplest notation, a logical notation, with the most powerful graphical engine possible, the visual part.

	DCpic	amscd	arrow	barr	borceaux	diagmac*	dratex	kuvio	pstricks	taylor	xypic
Specification matrix notation		√	√	√	√		√	√	√	√	√
absolute coordinates	√					√					√
Drawing H/V rules		√									
LaTeX slopes			√	√		√					
finite set of slopes					√			√			
finite set of slopes (XY-pic)											√
arbitrary slopes (PiCTeX)	√										
arbitrary slopes (PStricks)									√		
arbitrary slopes (DraTeX)							√				
arbitrary slopes (Postscript)									√	√	

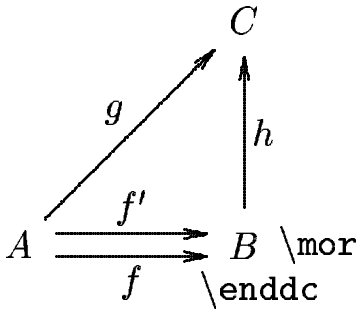
* not in CTAN
Source: Feruglio, Gabriel Valiente,*Typesetting Commutative Diagrams*,TUGboat, Volume 15 (1994), No. 4

A set of Objects, and a set of Arrows (morphisms), laid down in a (course) coordinate system (default value=30)



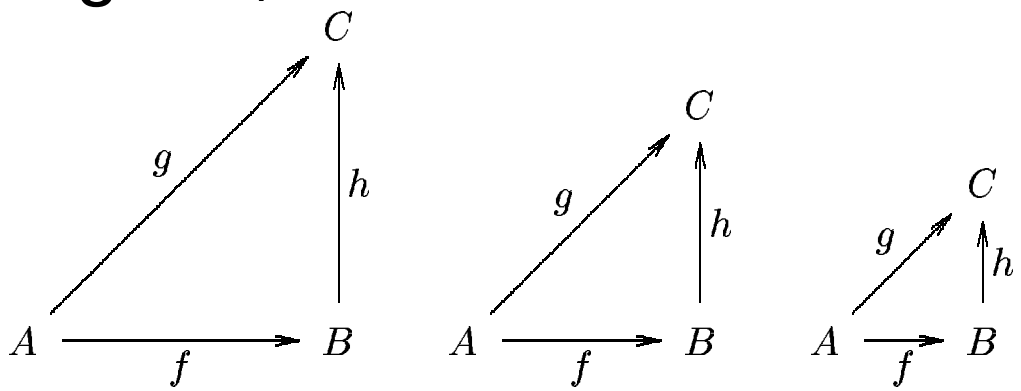

```
\begindc
\obj(1,1){$A$}
\obj(3,1){$B$}
\obj(3,3){$C$}
\mor(1,1)(3,1){$f$}[\atright,\solidarrow]
\mor(1,1)(3,3){$g$}
\mor(3,1)(3,3){$h$}[\atright,\solidarrow]
\enddc
```

A set of Objects, and a set of Arrows (morphisms), laid down in a (tight) coordinate system




```
\begindc[3]
\obj(1,10){$A$}
\obj(3,10){$B$}
\obj(3,30){$C$}
\mor(1,9)(30,9){$f$}[\atright,\solidarrow]
\mor(1,11)(3,11){$f^\prime$}
\mor(1,10)(3,30){$g$}
\mor(3,10)(3,30){$h$}[\atright,\solidarrow]
\enddc
```

The magnification factor gives us the capability of adapting the size of the diagram to the available space, without having to redesign the diagram,

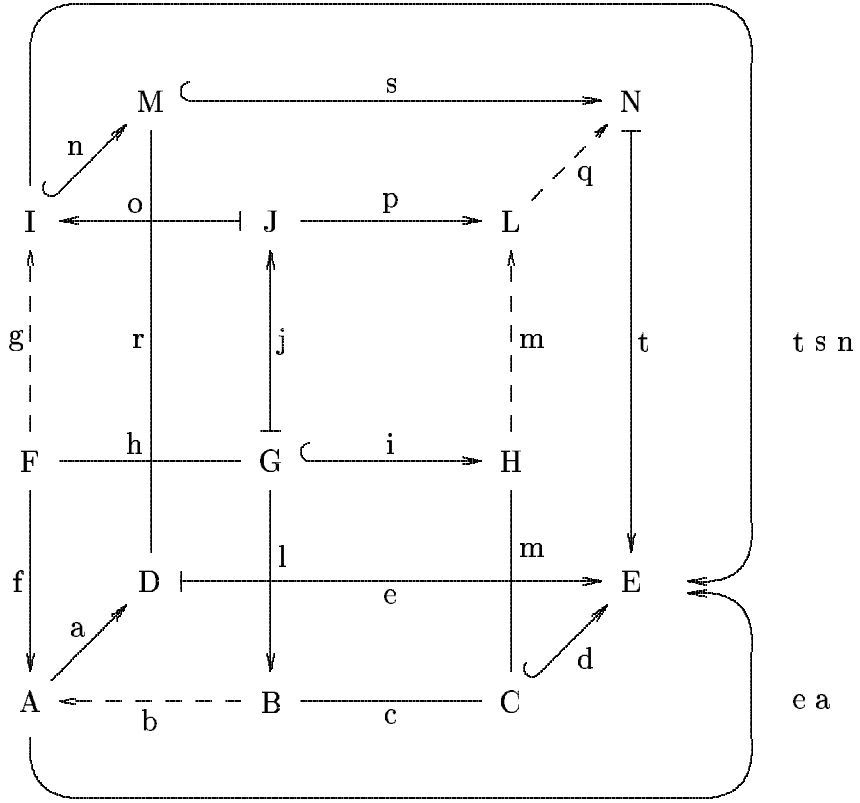


three instances of:

```
\begindc[X]
\obj(1,1){$A$}
\obj(3,1){$B$}
\obj(3,3){$C$}
\mor(1,1)(3,1){$f$}[\atright,\solidarrow]
\mor(1,1)(3,3){$g$}
\mor(3,1)(3,3){$h$}[\atright,\solidarrow]
\enddc
```

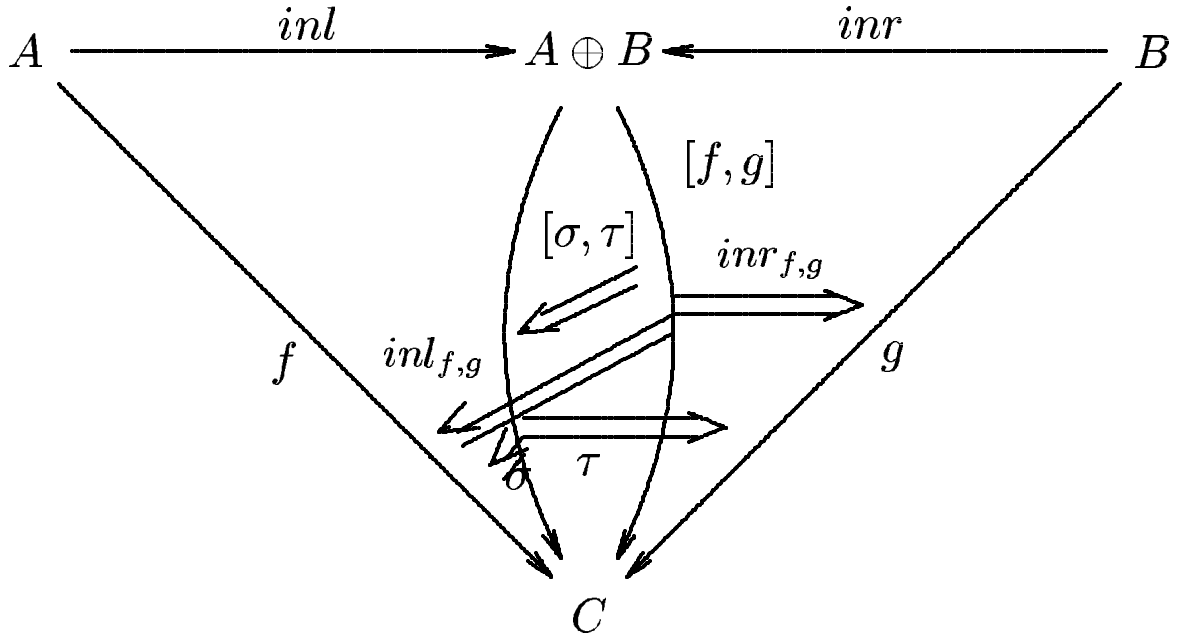
with X=40, X=30 (default value), and X=20

There are many kinds of arrows:



"solidarrow"; "dasharrow"; "solidline"; "injectionarrow"; "aplicationarrow"; and curved (quadratic) arrows.

Its always possible to embed PiCTeX commands in a DCpic diagram, allowing the user to produce all sorts of diagrams.



the $\Longrightarrow$ arrows were produced with PiCTeX commands embedded in a DCpic diagram.

The available commands: **`begindc, enddc, obj, mor, and cmor`**

The environment **`begindc ... enddc`**, establishes a Cartesian coordinate system with 1pt units.

```
\begindc[<magnification>]
...
\enddc
```

the optional argument **`<magnification>`** gives the user a multiplication factor **`1pt x <magnification>`** that allows us to magnify or shrink a diagram to suit the available space.

The translation to PiCTeX is:

```
\beginpicture
\setcoordinatesystem units <1pt,1pt>
\expansao=#1
```

and

```
\endpicture
```

respectively, **`expansao`** (expansion) is a global variable that will be used by the other commands

The command **obj** is the command that controls the placement of the objects.

```
\obj(<x>,<y>){<contents>}
```

"x" and "y" are the coordinates of the centre of a "hbox",the "contents" will be placed in that box expanding from the centre to both sides.

Its definition is

```
\def\obj(#1,#2)#3{\x=#1
  \y=#2
  \multiply \x by \expansao
  \multiply \y by \expansao
\put{#3} at {\x} {\y}}
```

The command **`mor`** is the command that controls the drawing of the linear arrows.

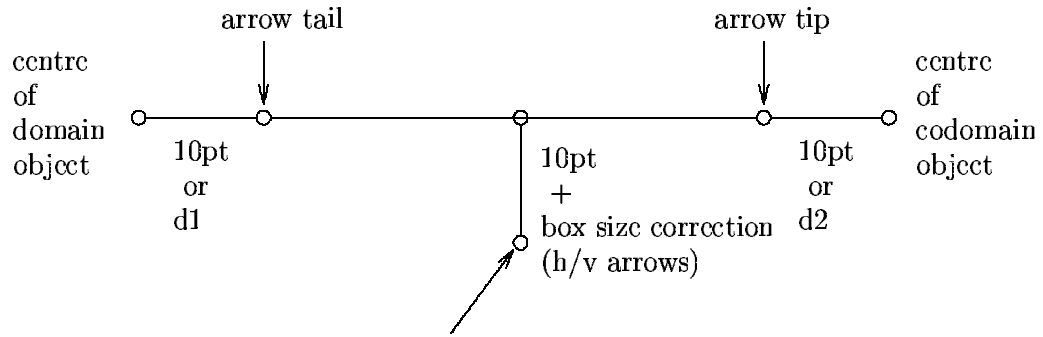
```
\mor(<x1>,<y1>)(<x2>,<y2>)[<d1>,<d2>]{<label>}[<label
 placement>,<arrow type>]
```

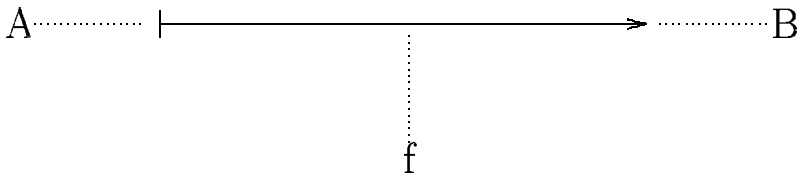
where:

- "x1" and "y1" are the coordinates of the domain object.
- "x2" and "y2" are the coordinates of the codomain object.
- "d1" is an optional argument that modifies the distance from the centre of the domain object to actual start of the arrow, default value 10.
- "d2" the same as "d1" but for the codomain object, default values 10.
- "label" its the arrow label, it will be putted in the centre of an "hbox"
- "label placement" (opcional argument)
  - 1 = \atleft, default value
  - -1 = \atright
- "arrow type" (opcional argument)
  - 0 = "\solidarrow", default arrow
  - 1 = "\dasharrow"
  - 2 = "\solidline"
  - 3 = "\injectionarrow"
  - 4 = "\aplicationarrow"

The **`mor`** command draw an arrow from an object to another object, a label is putted near the middle point of the arrow.

To do the actual drawing is necessary to calculate the arrow beginning, the arrow ending, and a reference point for the "hbox" containing the label.





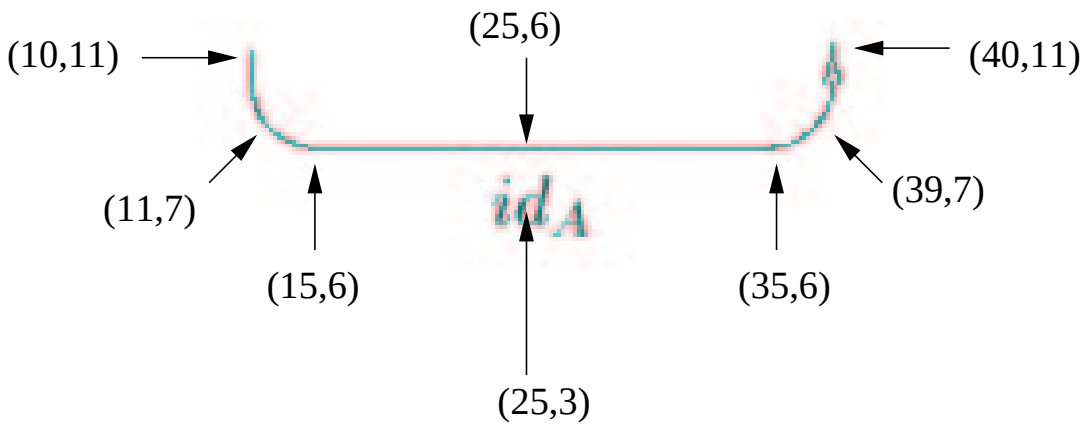
The command **`cmor`** is the command that controls the drawing of the curved (quadratic) arrows.

```
\cmor(<list_of_points>) {<arrow
 direction>}(<x>,<y>){<label>}[<arrow type>]
```

where:

- "list of points" is a list with an odd number of pairs (<x>,<y>)
- "arrow direction"
  - \pup, pointing up
  - \pdown, pointing down
  - \pleft, pointing left
  - \pright, pointing right
- "x" and "y" are the coordinates of the centre of a "hbox" containing the label
- "label", is the actual text
- "arrow type"
  - 0 = "\solidarrow", default arrow
  - 1 = "\dasharrow"
  - 2 = "\solidline"

```
\cmor((10,11)(11,7)(15,6)(25,6)(35,6)(39,7)(40,11)) \pup(25,3){$id_A$}
```

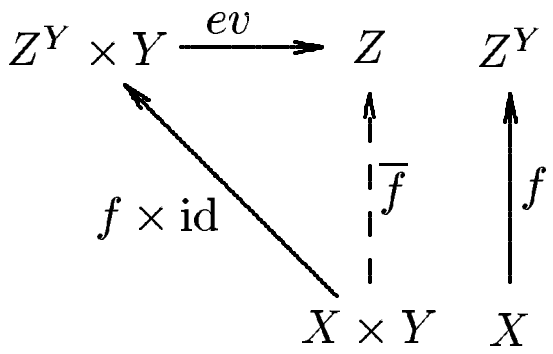


(10,11)(11,7)(15,6) (15,6)(25,6)(35,6) (35,6)(39,7)(40,11)

10 10+1 11+4
11 11−4 7−1

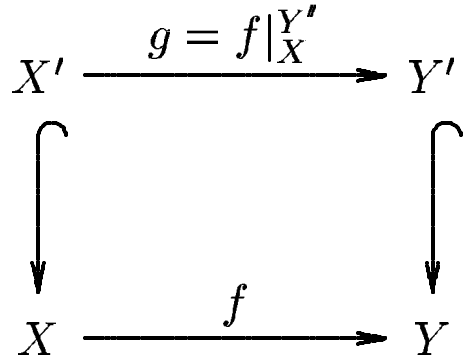
- Three points define a quadratic arc, a sequence of quadratic arcs define a quadratic curve.
- Expanded chess-horse movement, that is, (x,y), (x±4,y±1), ((x±4)±1,(y±1)±4), or (x,y), (x±1,y±4), ((x±1)±4,(y±4)±1).
- The label is placed in a "hbox" centred in (25,3).
- The arrow is pointing up.

Example: Exponentials



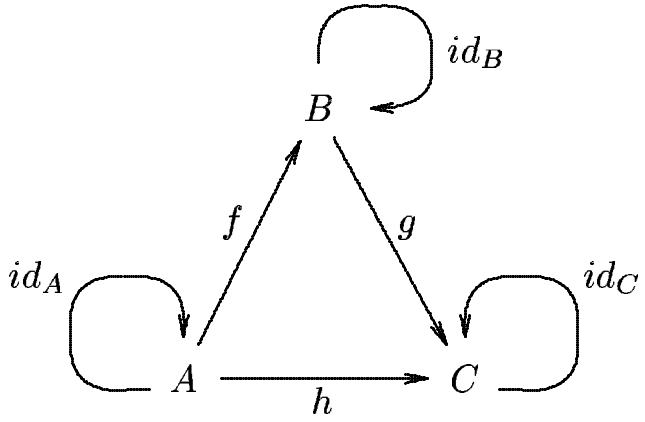

```
\begindc
\obj(1,3){$Z^Y\times Y$}
\obj(3,3){$Z$}
\obj(3,1){$X\times{}Y$}
\obj(4,1){$X$}
\obj(4,3){$Z^Y$}
\mor(1,3)(3,3)[20,10]{$ev$}
\mor(3,1)(1,3){$f\times{}\mathrm{id}$}
\mor(3,1)(3,3){$\overline{f}$}[\atright,\dasharrow]
\mor(4,1)(4,3){$f$}[\atright,\solidarrow]
\enddc
```

Example: Function Restriction.



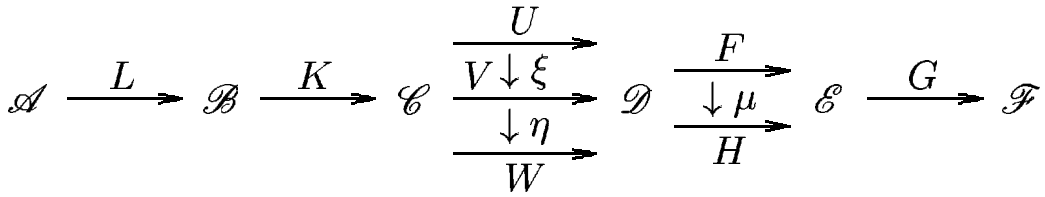

```
\begindc[28]
\obj(1,1){$X$}
\obj(1,3){$X^\prime$}
\obj(4,1){$Y$}
\obj(4,3){$Y^\prime$}
\mor(1,1)(4,1){$f$}
\mor(1,3)(1,1){}[\atright,\injectionarrow]
\mor(4,3)(4,1){}[\atright,\injectionarrow]
\mor(1,3)(4,3){$g=f|^{Y^\prime}_{X^\prime}$}
\enddc
```

Example: 3-Category




```
\begindc[3]
\obj(4,11){$A$}
\obj(39,11){$C$}
\obj(26,35){$B$}
\mor(4,11)(39,11){$h$}[\atright,\solidarrow]
\mor(4,11)(26,35){$f$}
\mor(26,35)(39,11){$g$}
\cmor((11,10)(10,10)(9,10)(5,11)(4,15)(5,19)(9,20)(13,19)(14,15))
\pdown(1,20){$d_A$}
\cmor((42,10)(43,10)(4,10)(48,11)(49,15)(48,19)(44,20)(40,19)(39,15))
\pdown(52,20){$d_C$}
\cmor((26,39)(27,43)(31,44)(35,43)(36,39)(35,36)(31,35))
\pleft(40,40){$d_B$}
\enddc
```

Example: Godement's "five" rules.




```
\begindc[7]
\obj(12,10){$\mathcal{A}$}
\obj(19,10){$\mathcal{B}$}
\obj(26,10){$\mathcal{C}$}
\obj(34,10){$\mathcal{D}$}
\obj(41,10){$\mathcal{E}$}
\obj(48,10){$\mathcal{F}$}
\mor(12,10)(19,10){$L$}
\mor(19,10)(26,10){$K$}
\mor(26,10)(34,10){$V\qquad\ $}
\mor(26,12)(34,12){$U$}
\mor(26,12)(34,12){$\downarrow\xi$}[\atright,\solidarrow]
\mor(26,8)(34,8){$\downarrow\eta$}
\mor(26,8)(34,8){$W$}[\atright,\solidarrow]
\mor(34,11)(41,11){$F$}
\mor(34,9)(41,9){$\downarrow\mu$}
\mor(34,9)(41,9){$H$}[\atright,\solidarrow]
\mor(41,10)(48,10){$G$}
\enddc
```

Conclusions: DCpic a package for drawing Commutative Diagrams, based in PiCTeX.

- Generic - TeX, LaTeX, ConTeXt, and all the formats that support PiCTeX.
- Specification - placement of "objects" in a coordinates system
  - simple specification
  - many types of arrows
  - options for fine tuning
- Drawing - PiCTeX power under the hood
  - arbitrary slopes
  - inclusion of PiCTeX commands if necessary

Work to the future:

- splines in PiCTeX
- arrows adjusting its size to the size of the objects
- more types of arrows (!?)
- ...

Thank you