DCpic v5.0.0
2013/05/01 (v15)

Pedro Quaresma
CISUC/Mathematics Department, University of Coimbra
3001-454 COIMBRA, PORTUGAL
pedro@mat.uc.pt  phone: +351-239 791 170  fax: +351-239 832 568
2013/05/01

This work may be distributed and/or modified under the conditions of the LaTeX Project Public License, either version 1.3 of this license or (at your option) any later version. The latest version of this license is in http://www.latex-project.org/lppl.txt and version 1.3 or later is part of all distributions of LaTeX version 2005/12/01 or later. This work has the LPPL maintenance status ‘maintained’.

The Current Maintainer of this work is Pedro Quaresma (pedro@mat.uc.pt).

This work consists of the files dcpic.sty, examples.tex, manDCPiC.tex, manDCPiCpt.tex
1 History

11/1990 - Version 1.0

10/1991 - Version 1.1

9/1993 - Version 1.2: the distance from the arrow end points and the objects became a fixed number (10 units), the distance arguments became optional. A new option for the arrows was introduced.

3/1995 - Version 1.3: the distance from the arrow to its label became fixed in 10 units. A new option for the arrows was introduced.

7/1996 - Version 2.1: a new syntax for the “mor” command was introduced.

5/2001 - Version 3.0: implementation of the command cmor for curved arrows.

11/2001 - Version 3.1: the arrow’s tips became more TeX-like.

1/2002 - Version 3.2: introduction of the “Logical specification of diagrams”, the commands “obj” and “mor” were modified accordingly.

5/2002 - Version 4.0: new syntax, non-compatible with past versions. The commands begindc, obj, and mor were modified.

3/2003 - Version 4.1: new kind of arrows, “surjective arrows”, was added. The drawing of the double tip only works well in the vertical, or horizontal case.

12/2004 - Version 4.1.1: new version for the drawing of surjective arrows, it solves the previous problem with it.

3/2007 - version 4.2: adds the “providespackage” directive that was missing. Adds dashed lines, and dotted lines.

5/2008 - version 4.2.1: deleting some counters, trying to avoid the problem “running out of counters”, that occurs because of the use of PiCTeX and DC PIC (only two...)

8/2008 - version 4.3: thanks to Ruben Debeerst (debeerst@mathematik.uni-kassel.de), he added a new arrow “equalline”. After that I decided to add: the doublearrow; the doublearrow with opposite directions; the null arrow. This last can be used as a simple form of adding new labels.

12/2008 - version 4.3.1: the command “id” is internalised (“!id”), it should be that way from the beginning because it is not to be used from the outside. The command “dasharrow” was changed to “dashArrow” to avoid a clash with the AMS command with the same name.

12/2009 - version 4.3.2: there was a conflict between dcpic.sty and hyperref in current texlive-2009 due to the one letter macro “\d” (thanks to Thorsten S <thorsten.schwander@gmail.com>). The “d” changed to “deuc” (Euclidian Distance). The “x” and “y” changed to “xO” “yO”
2 Introduction

DCpic is a package of \TeX \[3\] macros for graph modelling in a (La)\TeX \[4\] or Con\TeX t \[5\] document. Its distinguishing features are: the use of PICT\TeX \[7\] a powerful graphical engine, and a simple specification syntax. A graph is described in terms of its objects and its edges. The objects are textual elements and the edges can have various straight or curved forms.

A graph \[2\] in DCpic is a “picture” in PICT\TeX, in which we place our objects and morphisms (edges). The user’s commands in DCpic are: \texttt{begindc} and \texttt{enddc} which establish the coordinate system where the objects will by placed; \texttt{obj}, the command which defines the place and the contents of each object; \texttt{mor}, and \texttt{cmor}, the commands which define the morphisms, linear and curved edges, and its labels.

Until version 3.2 DCpic “Commutative Diagrams in PICT\TeX” was dedicated to commutative diagrams \[6\] ("Diagramas Comutativos” in Portuguese) drawing, that is the reason for the name “obj” (objects) and “mor” (morphism) for the nodes and edges.

Example:

\begin{verbatim}
\begindc{\commdiag}
obj(10,10){$A$
\obj(30,10){$B$
\obj(30,30){$C$
\mor{$A$}{$B$}{$f$}[atright,solidarrow]
\mor{$B$}{$C$}{$g$}[atright,solidarrow]
\mor{$A$}{$C$}{$h$}
\enddc
\end{verbatim}

Note: all the numeric values should be integer values.

3 Description

3.1 Available Commands

The environment:

\begin{verbatim}
\begindc[#1][#2]
\end{verbatim}

#1 - Graph type
0 = "commdiag", commutative diagram
1 = "digraph", direct graph
2 = "undigraph", undirect graph
3 = "cdigraph", direct graph with incircled objects
4 = "cundigraph", undirect graph with incircled objects

(optional) #2 - magnification factor (default value, 300)
\enddc

Objects:

\obj(#1,#2)[#3][#4][#5]
#1 and #2 - coordinates
(optional) #3 - Label, to be used in the morphisms command, if not present the #4 will be used for that purpose
#4 - Object contents
(optional) #5 - placement of the object (default value \north)
  0="\pcent", center
  1="\north", north
  2="\northeast", northeast
  3="\east", east
  4="\southeast", southeast
  5="\south", south
  6="\southwest", southwest
  7="\west", west
  8="\northwest", northwest

Note: if you omit the #3 argument (label), and the #4 argument is a complex (and long) \LaTeX command, then this command can crash. In this case you must specify a label (the empty label [], if you do not intend to use the label).

Morphism (linear edges). This commando has two major variants:

1. Starting and Ending objects specification

\mor[#1]{#2}[#5,#6][#7][#8,#9]
As you can see this first form is (intentionally) badly formed, the arguments #3 and #4 are missing (the actual command is correctly formed).
#1 - The starting object reference
#2 - The ending object reference
from this two we will obtain the objects coordinates, and also the dimensions of the enclosing box.
The objects box dimensions are used to do an automatic adjustment of the edge width.
from #1 we obtain \((x,y)\), (#1,#2) in the second form
from #2 we obtain \((x',y')\), (#3,#4) in the second form
this values will be passed to the command second form
2. Two points coordinates specification

\mor(#1,#2)(#3,#4)[#5,#6]{#7}[#8,#9]

Now we can describe all the arguments

#1 and #2 - coordinates (beginning)
#3 and #4 - coordinates (ending)
(optional)#5,#6 - correction factors (default values, 10 and 10)
#5 - actual beginning of the edge
#6 - actual ending of the edge
#7 - text (morphism label)
(optional)#8,#9
#8 - label placement
  1 = \atright, at right, default value
  -1 = \atleft, at left
#9 - edge type
  0 = "\solidarrow", default value
  1 = "\dashArrow"
  2 = "\dotArrow"
  3 = "\solidline"
  4 = "\dashline"
  5 = "\dotline"
  6 = "\injectionarrow"
  7 = "\aplicationarrow"
  8 = "\surjectivearrow"
  9 = "\equalline"
  10 = "\doublearrow"
  11 = "\doubleopposite"
  12 = "\nullarrow" (to allow adding labels to existing arrows)

Note: insert a space after the command.

Curved Morphisms (quadratic edges):

\cmor(#1) #2(#3,#4){#5}[#6]
#1 - list of points (odd number)
#2 - tip direction
  0 = "\pup", pointing up
  1 = "\pdown", pointing down
  2 = "\pright", pointing right
  3 = "\pleft", pointing left
#3 and #4 - coordinates of the label
#5 - morphism label
(optional) #6 - edge type
  0 = "\solidarrow", default value
  1 = "\dashArrow"
  2 = "\solidline"
4 Examples

4.1 Different Types of Lines/Arrows

\begin{commdiag}{250}
\obj{10,10}{A}{\$A_{08}}
\obj{14,11}{A1}{\$A_{15}}
\obj{13,12}{A2}{\$A_{28}}
\obj{12,13}{A3}{\$A_{38}}
\obj{10,14}{A4}{\$A_{48}}
\obj{9,13}{A5}{\$A_{58}}
\obj{8,12}{A6}{\$A_{68}}
\obj{7,11}{A7}{\$A_{75}}
\obj{6,10}{A8}{\$A_{88}}
\obj{7,9}{A9}{\$A_{98}}
\obj{9,8}{A10}{\$A_{1010}}
\obj{12,8}{A11}{\$A_{118}}
\mor{A}{A0}{\$A_{08}}{\text{atright, solidarrow}}
\mor{A}{A1}{\$A_{15}}{\text{atright, dasharrow}}
\mor{A}{A2}{\$A_{28}}{\text{atright, dotArrow}}
\mor{A}{A3}{\$A_{38}}{\text{atright, solidline}}
\mor{A}{A4}{\$A_{48}}{\text{atright, dashline}}
\mor{A}{A5}{\$A_{58}}{\text{atleft, dotline}}
\mor{A}{A6}{\$A_{68}}{\text{atleft, injectionarrow}}
\mor{A}{A7}{\$A_{75}}{\text{atleft, applicationarrow}}
\mor{A}{A8}{\$A_{88}}{\text{atleft, surjectivearrow}}
\mor{A}{A9}{\$A_{98}}{\text{atleft, equalline}}
\mor{A}{A10}{\$A_{1010}}{\text{atleft, doublearrow}}
\mor{A}{A11}{\$A_{118}}{\text{atright, doubleopposite}}
\end{commdiag}

4.2 Diagrams with Curved Edges, Double Arrows, ...

Curved edges

\begin{commdiag}{0}
\obj{14,11}{A8}
\obj{39,11}{A8}

Notes: Insert a space after the command. The space after the list of points is mandatory.
Double Arrows and Natural Transformations

4.3 A Complex Diagram

The following diagram is one of the test-diagrams used by Feruglio [1], as we can see DCpic performs very well, drawing the complete diagram based on a very simple specification.
4.4 “Around the Word” [2]
4.5 Incircled Objects

\begin{cdigraph}[250]\end{cdigraph}

\begin{verbatim}
\obj{6,15}{9} \obj{9,16}{8} \obj{11,14}{7} \obj{13,14}{3} \obj{15,16}{2} \obj{18,15}{15} \obj{12,19}{1} \obj{12,22}{20} \mor{18}{17}{1} \mor{18}{11}{1} \mor{18}{19}{1} \mor{11}{12}{1} \mor{11}{10}{1} \mor{12}{13}{1} \mor{12}{5}{1} \mor{10}{6}{1} \mor{10}{9}{1} \mor{5}{6}{1} \mor{5}{4}{1} \mor{13}{17}{1} \mor{13}{14}{1} \mor{9}{19}{1} \mor{9}{8}{1} \mor{6}{7}{1} \mor{4}{3}{1} \mor{4}{14}{1} \mor{19}{20}{1} \mor{8}{11}{1} \mor{8}{7}{1} \mor{7}{3}{1} \mor{3}{2}{1} \mor{2}{1}{1} \mor{14}{15}{1} \mor{17}{16}{1} \mor{16}{20}{1} \mor{1}{20}{1} \mor{15}{16}{1}
\end{verbatim}
References


5 The Code

%% DC-PiCTeX
%% Copyright (c) 1990-2013 Pedro Quaresma, University of Coimbra, Portugal
%% 11/1990 (version 1.0);
%% 10/1991 (version 1.1);
%% 9/1993 (version 1.2);
%% 3/1995 (version 1.3);
%% 7/1996 (version 2.1);
%% 5/2001 (version 3.0);
%% 11/2001 (version 3.1);
%% 1/2002 (version 3.2)
%% 5/2002 (version 4.0);
%% 3/2003 (version 4.1);
%% 12/2004 (version 4.1.1)
%% 3/2007 (version 4.2)
%% 5/2008 (version 4.2.1)
%% 8/2008 (version 4.3)
%% 12/2008 (version 4.3.1)
%% 12/2009 (version 4.3.2)
%% 4/2013 (version 4.4.0)
%% 5/2013 (version 5.0)
DCpic is a package of \TeX\ macros for graph modelling in a \LaTeX\ or Con\TeX\ t document. Its distinguishing features are:

- the use of \PiCTeX\, a powerful graphical engine, and a simple specification syntax. A graph is described in terms of its objects and its edges. The objects are textual elements and the edges can have various straight or curved forms.

- A graph in DCpic is a "picture" in \PiCTeX, in which we place our (em objects) and (em morphisms) (edges). The user's commands in DCpic are: \tt{begindc} and \tt{enddc} which establishes the coordinate system where the objects will be placed; \tt{obj}, the command which defines the place and the contents of each object; \tt{mor}, \tt{cmor}, the commands which define the morphisms, linear and curved edges, and their labels.

Example:

\begindc{\commdiag}[3]
\obj(10,15){$A$}
\obj(25,15){$B$}
\obj(40,15){$C$}
\mor{$A$}{$B$}{$f$}
\mor{$B$}{$C$}{$g$}
\cmor((10,11)(11,7)(15,6)(25,6)(35,6)(39,7)(40,11))
\pup(25,3){$g\circ f$}
\enddc

NOTES:
all the numeric values should be integer values.

Available commands:

The environment:
\begindc{#1}[#2]
#1 = Graph type
0 = "commdiag" (commutative diagram)
1 = "digraph" (direct graph)
2 = "undigraph" (undirected graph)
Objects:
\obj(#1,#2)[#3]{#4}[#5]

#1 and #2 - coordinates
( optional ) #3 - Label, to be used in the morphims command, if not present the #4 will be used to that purpose
#4 - Object contents
( optional ) #5 - placement of the object ( default value \north )
0=\percent, center
1=\north, north
2=\northeast, northeast
3=\east, east
4=\southeast, southeast
5=\south, south
6=\southwest, southwest
7=\west, west
8=\northwest, northwest

!!! Note !!!
if you omit the #3 argument ( label ) and the #4 argument is a complex LaTeX command this can cause this command to crash. In this case you must specify a label ( the empty label [], if you do not need it it for nothing ).

Morphims ( linear edges ). This command has two major variants
i) Starting and Ending objects specification
\mor(#1)[#2][#5][#6](#7)[#8][#9]

As you can see this first form is ( intentionally ) badly formed, the arguments #3 and #4 are missing ( the actual command is correctly formed ).

#1 - The starting object reference
#2 - The ending object reference

from this two we will obtain the objects coordinates, and also the dimensions of the enclosing box.

The objects box dimensions are used to do an automatic adjustment of the edge width.

from #1 we obtain ( x,y ), (#1,#2) in the second form
from #2 we obtain ( x',y' ), (#3,#4) in the second form

this values will be passed to the command second form

(ii) Two points coordinates specification
\mor(#1,#2)(#3,#4)[#5][#6](#7)[#8][#9]

Now we can describe all the arguments

#1 and #2 - coordinates ( beginning )
#3 and #4 - coordinates ( ending )
( optional ) #5,#6 - correction factors ( default values, 100 and 100 (10pt) )
#5 - actual beginning of the edge
#6 - actual ending of the edge
#7 - text ( morphism label )
( optional ) #8,#9
#8 - label placement
1 = \atright, at right, default value
-1 = \atleft, at left
#9 - edge type
0 = \"\solidarrow\", default edge

1 = \"\dashArrow\"

2 = \"\dotArrow\" (thanks to Xingliang Liang <jkl9543@gmail.com>)

3 = \"\solidline\"

4 = \"\dashline\"

5 = \"\dotline\"

6 = \"\injectionarrow\"

7 = \"\aplicationarrow\"

8 = \"\surjectivearrow\"

9 = \"\equalline\" (thanks to Ruben Debeerst <debeerst@mathematik.uni-kassel.de>)

10 = \"\doublearrow\"

11 = \"\doubleopposite\"

12 = \"\nullarrow\" (to allow adding labels to existing arrows)

Notes: the equalline "arrow" does not provide a second label.

Curved Morphisms (quadratic edges):
\cmor(#1) #2(#3,#4){#5}[#6]
#1 - list of points (odd number)
#2 - tip direction
0 = \"\pup\", pointing up
1 = \"\pdown\", pointing down
2 = \"\pright\", pointing right
3 = \"\pleft\", pointing left
#3 and #4 - coordinates of the label
#5 - morphism label
#6 - edge type
0 = \"\solidarrow\", default value
1 = \"\dashArrow\"
2 = \"\solidline\"

Notes: insert a space after the command.
the space after the list of points is mandatory

Examples:
\documentclass[a4paper,11pt]{article}
\usepackage{dcpic,pictexwd}

\begin{document}
\begindc[3]
\obj(14,11){$A$}
\obj(39,11){$B$}
\mor(14,12)(39,12){$f$}[]\atright,\solidarrow
\mor(39,10)(14,10){$g$}[]\atright,\solidarrow
\cmor((10,11)(11,7)(15,6)(25,6)(35,6)(39,7)(40,11))
\pup(25,3){$id_A$}
\cmor((40,7)(41,3)(45,2)(49,3)(50,7)(49,11)(45,12))
\pleft(54,3){$id_B$}
\enddc

\end{document}

The command "begindc" now have an obligatory argument, this argument allows the specification of the graph type "commdiag" (0), commutative diagrams "digraph" (1), directed graphs "undigraph" (2), undirected graphs

The command "obj" has a new syntax: after the coordinates specification, an optional argument specifying a label, an obligatory argument given the "value" of the object and the final optional argument used in the graphs to set the relative position of the "value" to the "dot" defining the objects position, the default value is "north".

3/2003 (version 4.1)
Responding to a request of Jon Barker <jeb1@soton.ac.uk> I create a new type of arrow, the surjective arrow.
For now only horizontal and vertical versions, other angles are poorly rendered.
12/2004 (version 4.1.1)
New version for the surjective arrows, solve the problems with the first implementation of this option.
3/2007 (version 4.2)
Adds the "providespackage" directive that was missing.
Adds dashed lines, and dotted lines.
5/2008 (version 4.2.1)
Deleting some counters, trying to avoid the problem "running out of counters", that occurs because of the use of PiCTeX and DCpic (only two...)
8/2008 (version 4.3)
Thanks to Ruben Debeerst (debeerst@mathematik.uni-kassel.de), he added a new arrow "equalline". After that I decided to add: the doublearrow; the doublearrow with opposite directions; the null arrow. This last can be used as a simple form of adding new labels.
12/2008 (version 4.3.1)
The comand \id is internalised (\!id), it should be that way from the begining because it is not to be used from the outside.
The comand \dasharrow was changed to \dashArrow to avoid a clash with the AMS command with the same name.
12/2009 (version 4.3.2)
There is a conflict between dcpic.sty and hyperref in current texlive-2009 due to the one letter macro \d (thanks Thorsten S <thorsten.schwander@gmail.com>).
The \d changed to \deuc (Euclidian Distance). The \x and \y changed to \xO \yO
4/2013 (version 4.4.0)
Thanks to Xingliang Liang <jkl9543@gmail.com>. He added a new arrow "dotarrow".
5/2013 (version 5.0)
The base scale of the graph has changed from lpt to .lpt to solve a problem with the implementation of the oblique equalline (Thanks to Antonio de Nicola).
The LaTeX circle and oval commands where replaced by the PiCTeX circulararc and ellipticalarc commands to avoid differences in scales.
\newcount\yl%
\newcount\deuc%
\newcount\dm%
\newcount\xa%
\newcount\xb%
\newcount\xmed%
\newcount\xc%
\newcount\xd%
\newcount\xe
\newcount\xf
\newcount\ya%
\newcount\yb%
\newcount\xmed%
\newcount\yc%
\newcount\yd
\newcount\ye
\newcount\yf

%% "global variables"
\newcount\expansao%
\newcount\tipografo% version 4.0
\newcount\distanciaobjet% version 4.0
\newif\ifpara%

%% version 3.2
\newbox\caixa%
\newbox\caixaaux%
\newif\ifvazia%
\newif\ifcompara%
\newif\ifdiferentes%
\newcount\xaux%
\newcount\yaux%
\newcount\guardaauxa%
\newcount\alt%
\newcount\larg%
\newcount\prof%

%% for the trimming
\newcount\auxx
\newcount\auxy
\newif\ifajustas%
\newif\ifajustadist
\def\objPartida{}%
\def\objChegada{}%
\def\objNulo{}%

%%
%% Stack specification
%%

%%
%% Empty stack
%%
\def\vazia:{}

%%
%% Is Empty? : Stack -> Bool
%%
\def\pilhanvazia#1{|\let\arg=#1%
\if:\arg\vaziatrue\vaziafalse\else \vaziatrue\vaziafalse\fi}

%%
%% Push : Elems x Stack -> Stack
%%
\def\coloca#1#2{|edef\pilha(#1.#2)}
\def\atleft{1}
\def\pup{0}
\def\pdown{1}
\def\pright{2}
\def\pleft{3}
\def\commdiag{0}
\def\digraph{1}
\def\undigraph{2}
\def\cdigraph{3}
\def\cundigraph{4}
\def\pcent{0}
\def\north{1}
\def\northeast{2}
\def\east{3}
\def\southeast{4}
\def\south{5}
\def\southwest{6}
\def\west{7}
\def\northwest{8}

%% Private Macro
%% Adjust the distance between the arrows and the objects regarding
%% the dimensions of the objects.
%% \ajusta{x}{x1}{y}{y1}{d}{Object} (ajusta = adjust)
%% Input
%% (x,y) e (x1,y1) - start, end coordinates of arrow
%% d - distance specified by the user (default value, 100)
%% Objects - reference of the object pointed by the arrow
%% Output
%% d - adjusted distance
%% The adjusted distance is the greatest value between 100 and the
%% object's box dimensions. If the user specify a value this is not
%% altered.
%% If the arrow is horizontal the length is used.
%% If the arrow is vertical the height is used for arrows in the 1st
%% or 2nd quadrant, or the depth if the arrow is in the 3rd or 4th
%% quadrant. If the arrow is oblique the value is chosen accordingly:
%% from 315 to 45 degrees length is used
%% from 45 to 135 degrees height is used
%% from 135 to 225 degrees length is used
%% from 225 to 315 degrees depth is used
\def\ajusta#1#2#3#4#5#6{
  \aux=#5%
  \let\auxobj=#6%
  \ifcase \tipografo % commutative diagrams
    \ifnum\number\aux=100
      \ajustadisttrue % if needed, adjust
    \else
      \ajustadistfalse % if not, keeps unchanged
    \fi
  \else % graphs (directed, undirected, with frames)
    \ajustadistfalse
  \fi
  \ifajustadist
    \let\auxobjo=\auxobj%
    \ifcase \tipografo % commutative diagrams
      \ifnum\number\aux=100
        \ajustadisttrue % if needed, adjust
      \else
        \ajustadistfalse % if not, keeps unchanged
      \fi
    \else
      \ajustadistfalse
    \fi
    \let\pilhaaux=\pilha%
    \loop
      \!topo{\pilha}\
      \!retira{\pilha}\
      \!compara{\!id}{\auxobjo}\
      \ifcompara\vaziafalse \else\!pilhanvazia\pilha \fi
  \fi
}
\ifnvazia%
\repeat%
% push the values into the stack
\let\pilha=\pilhaaux%
\ifvazia%
\ifdiferentes%
%% It is not possible to make the adjustment given the fact that the
%% user did not provide a label for the object in question. We set a
%% value equal to the default value (100)
++
larg=131072% these values are for unit of .1pt
\prof=65536%
\alt=65536%
%\fi%
\divide\larg by 13107% these values are for unit of .1pt
\divide\prof by 6553%
\divide\alt by 6553%
\ifnum\number\yO=\number\yl1
% Case 1 -- horizontal arrow
% with the division by 13107 we get half the size of the box, for a
% centered text, the adding of 30 is an empirical adjustment.
++
\advance\larg by 30
\ifnum\number\larg>\aux
#5=\larg
\else
\ifnum\number\xO=\number\xl
\ifnum\number\yl>\number\yO
% Case 2.1 -- vertical arrow, down direction
++
\ifnum\number\alt>\aux
#5=\alt
\else
% Case 2.2 -- vertical arrow, up direction
++
\ifnum\number\alt<\aux
#5=\alt
\else
% Case 3 -- oblique arrow
% Case 3.1 --- from 315o to 45o; |x-xl|>|y-yl|
% Case 3.3 --- from 135o to 225o; |x-xl|>|y-yl|; Length
\auxqx=\xO
\advance\auxqx by -\xl
%!absoluto{\auxqx}{\auxqx}\%
\auxgy=\yO
\advance\auxgy by -\yl
%!absoluto{\auxgy}{\auxgy}\%
\ifnum\auxqx>\auxgy
\ifnum\larg<100
\larg=100
\fi
\advance\larg by 30
#5=\larg
\else
% Case 3.2 --- from 45o to 135o; |x-xl|<|y-yl| e y<0; Length
\ifnum\yl>\yO
\ifnum\larg<100
19
Case 3.4 -- from 225° to 315°; |x-xl|<|y-yl| e y<0; Depth

Private Macro
% Square root
% raiz(n){m} (raiz = root)
% ->
% n - natural number
% <-
% m - greatest natural number less then the square root of n

\def\raiz#1#2{\auxa=#1%
\auxb=1%
\loop
\aux=\auxb%
\advance \aux by 1%
\multiply \aux by \aux%
\ifnum \aux < \auxa%
\advance \auxb by 1%
\paratrue%
\else\ifnum \aux=\auxa%
\advance \auxb by 1%
\paratrue%
\else\parafalse%
\fi
\fi
\ifpara%
\repeat
#2=\auxb}

Private Macro
% Find the starting and ending points of the "arrow" and also the
% label position (one coordinate at a time)
% Input
% x1,x2,x3,x4,x5
% Output
% x6
% x2 - x1
% x6 = x3 +|--- ------- x4
% x5
\def\ucoord{(x1)(x2)(x3)(x4)(x5){+- 1}
\advance \aux by -#1%
\multiply \aux by #4%
\divide \aux by #6%
\ifnum #7 = -1 \multiply \aux by -1 \fi%
\advance \aux by #3%
#6=\aux}

Private Macro
% Euclidean distance between two points
% quadrado = square
\% quadrado{n}{m}{l}
\% Input
\% n - natural number
\% m - natural number
\% Output
\% l = (n-m)\times(n-m)
\def\!quadrado#1#2#3{\aux=#1\
\advance \aux by -#2\%
\multiply \aux by \aux\%
#3=\aux}

\%Private Macro
\% Euclidean distance between arrows and its tags
\% Input
\% (x,y), (x',y') morphism's name (tag)
\% Output
\% dnm - distance between an arrow and its tags
\% (with a trim given by the tag’s size
\% Observations
\% The trimming is for horizontal and vertical arrows
\% only. Oblique arrows are dealt in a different way
\%
\% Algorithm
\% caixa0 <- morfism name
\% if x-xl = 0 then {vertical arrow}
\% aux <- caixa0 width
\% dnm <- conversion-sp-pt(aux)/2+3
\% else {non-vertical arrow}
\% if y-y1 = 0 then {horizontal arrow}
\% aux <- caixa0 height+depth
\% dnm <- conversion-sp-pt(aux)/2+3
\% else {oblique arrow}
\% dnm <- 3
\% endif
\% endif
\% endalgorithm
\def\!distnomemor#1#2#3#4#5#6{\setbox0=\hbox{#5}\
\aux=#1
\advance \aux by -#3
\ifnum \aux=0
\aux=\wd0 \divide \aux by 131072
\advance \aux by 30
#6=\aux
\else
\aux=#2
\advance \aux by -#4
\ifnum \aux=0
\aux=\ht0 \advance \aux by \dp0 \divide \aux by 131072
\advance \aux by 30
#6=\aux
\else
#6=30
\fi
\fi
}

\%
\% The environment "begindc...enddc"
\%
\def\begindc#1{\ifnextchar[{}{\begindc{#1}}{\begindc{#1}[30]}}
\def\begindc#1[#2]{\beginpicture
\let\pilha=\vazia
\setcoordinatesystem units <1pt,1pt>
\expandafter\endpicture
\begindc{#1}[#2]}}
\ifcase \distanciaobjetivos
  \tipoarco=0 \% arrow
  \tipografia=0 \% commutative diagram
\or
  \distanciaobjetivos=20
  \tipoarco=0 \% arrow
  \tipografia=1 \% directed graph
\or
  \distanciaobjetivos=10
  \tipoarco=3 \% line
  \tipografia=2 \% undirected graph
\or
  \distanciaobjetivos=80
  \tipoarco=0 \% arrow
  \tipografia=3 \% directed graph
\or
  \distanciaobjetivos=80
  \tipoarco=3 \% line
  \tipografia=4 \% undirected graph
\fi}

\def\enddc{\endpicture}
\def\drawarrowhead <#1> \[#2,#3\]{{
\ifnextchar<\drawarrowhead{#1}{#2}{#3}{#4}{#5}\drawarrowhead{#1}{#2}{#3}}

% Xingliang Liang <jkl9543@gmail.com>
% ** \ljoin (XCOORD,YCOORD)
% ** Draws a straight line starting at the last point specified
% ** by the most recent \start, \ljoin, or \qjoin, and
% ** ending at (XCOORD,YCOORD).
\def\ljoindummy (#1,#2){{
\advance\intervalno by 1
\xE=\m{#1}\xunit \yE=\m{#2}\yunit
\rotateaboutpivot\xE\yE
\xdiff=\xE \advance \xdiff by -\xS%** xdiff = xE - xS
\ydiff=\yE \advance \ydiff by -\yS%** ydiff = yE - yS
\Pythag\xdiff\ydiff\arclength% ** arclength = sqrt(xdiff**2+ydiff**2)
\global\advance \totalarclength by \arclength%** set by dashpat to \linesolid or \linedashed
\xE=\xS \yE=\yS \yE%** shift ending points to starting points
\ignorespaces}

%% \drawarrowhead(4pt)(DimC)(DimD) <xshift,yshift> from (\xa) (\ya) to (\xb) (\yb)
%% \def\drawarrowhead\#1\#2\#3<\#4,\#5> from \#6 \#7 to \#8 \#9 {
%%  ** convert to dimensions
\xloc=\m{\#6}\xunit
\yloc=\m{\#7}\yunit
\xdxpos=\xloc \ \dimena=\m{\#6}\xunit \advance \dxpos -\dimena
\dypos=\yloc \ \dimena=\m{\#7}\yunit \advance \dypos -\dimena
\let\Mx\Mx \Mx%** save current c/d mode
\setdimenmode%** go into dimension mode
%%  ** draw shaft of arrow
\xshift=\#4\relax \yshift=\#5\relax%** pick up shift
\reverserotatetoonly\xshift\yshift%** back rotate shift
\advance\xshift\xloc \advance\yshift\yloc
%%  ** draw shaft of arrow
\xshift=\#4\relax \yshift=\#5\relax%** pick up shift
\reverserotatetoonly\xshift\yshift%** back rotate shift
\advance\xshift\xloc \advance\yshift\yloc
\ljoindummy (\xshift,\yshift)
\% ** find 32*cosine and 32*sine of angle of rotation
\Pythag\dxpos\dypos\arclength
\divide\dxpos\arclength\dxpos
\dxpos=32\dxpos \!removept\dxpos\!cos
\divide\dypos\arclength\dypos
\dypos=32\dypos \!removept\dypos\!sin
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\%
\!retira\pilha\% \\compara{\id}{\objChegada}\% 
\ifcompara \nvaziafalse \else \pilhanvazia\pilha \fi 
\ifvazia 
\repeat 
\ifnvazia 
\fi 
\ifvazia 
\fi 
\let\pilha=\pilhaaux 
\ifnextchar[\mora{\number\xO}{\number\yO}{\number\xaux}{\number\yaux}\{\number\distanciaobjmor,\number\distanciaobjmor\}}\{\number\distanciaobjmor,\number\distanciaobjmor\}\{\number\distanciaobjmor,\number\distanciaobjmor\}\{1,\number\tipoarco\} \} 
\def\mora#1#2#3#4\{#5,#6\}{#7}\% 
\ifnextchar[\morb{#1}{#2}{#3}{#4}{#5}{#6}{#7}\{\number\distanciaobjmor,\number\distanciaobjmor\}}\{\number\distanciaobjmor,\number\distanciaobjmor\}\{\number\distanciaobjmor,\number\distanciaobjmor\}\{1,\number\tipoarco\} \} 
\def\morb#1#2#3#4#5#6#7\{#8,#9\}{#10} \{\number\distanciaobjmor,\number\distanciaobjmor\}\{\number\distanciaobjmor,\number\distanciaobjmor\}\{1,\number\tipoarco\} \} 
\def\quadrado\{\number\xO}\{\number\yO\}\{\number\auxa\} \{\number\auxb\} \{\number\deuc\} \{\number\deuc\} \{\number\deuc\} \{1\} 
\def\raiz\{\number\deuc\}\{\number\deuc\} \{\number\deuc\} \{\number\deuc\} \{1\} 
\auxa=#5 
\compara{\objNulo}{\objPartida}\% 
\ifdiferentes% adjusting only when needed 
\ajusta\{\xO\}{\yO\}{\yl\}{\xaux}\{\objPartida\} 
\ajustatrue 
\def\objPartida{}% reset the value of the starting object 
\fi 
\auxa=#5 
\compara{\objNulo}{\objChegada}\% 
\auxa=#6 
\compara{\objNulo}{\objChegada}\% 
\auxa=#6 
\auxb=#6 
\compara{\objNulo}{\objChegada}\% 
\auxa=#6 
\auxb=#6 
\compara{\objNulo}{\objChegada}\%
% adjusting only when needed
% adjustment
%!ajusta\{xO\}{xL}\{yl\}{yO}{auxb}{objChegada}%
def\objChegada{}% reset the value of the end object
!
!advance \auxa by -auxb%
!ucoord{number\{xO\}{xL}\{number\auxa\}{number\{deuc\}\{xb\}\{i\}}%
!ucoord{number\{yO\}{yL}\{number\auxa\}{number\{deuc\}\{yb\}\{i\}}%
xmed=xa%
!advance \xmed by \xb%
\divide \xmed by 2
\ymed=ya%
!advance \ymed by \yb%
\divide \ymed by 2
%
% find the coordinates of the label position: (xc,yc)
%
% after this the values of xmed and ymed are no longer important
%
!distnomemor{number\{xO\}{xL}\{number\{yO\}{yL}\{number\{xa\}{yl\}{number\{deuc\}\{dnm\}{\dnm}-%
!ucoord{number\{xO\}{xL}\{number\ymed\}{number\{deuc\}\{yc\}{\#8}%
%
% draw the "arrow"
%
\ifcase #9 % 0=solid arrow
\arrow <4pt> [.2,1.1] from \{xa\} \{ya\} to \{xb\} \{yb\}
or % 1=dashed arrow
\setdashes <2pt>
\plot \{xa\} \{ya\} \{xb\} \{yb\} /
\setsolid%
\drawarrowhead <4pt> [.2,1.1] from \{xa\} \{ya\} to \{xb\} \{yb\}
or % 3=solid line
\setlinear
\plot \{xa\} \{ya\} \{xb\} \{yb\} /
or % 3=dashed line
\setdashes <2pt>
\plot \{xa\} \{ya\} \{xb\} \{yb\} /
\setsolid
\drawarrowhead <4pt> [.2,1.1] from \{xa\} \{ya\} to \{xb\} \{yb\}
or % 5=dotted line
\setdots <2pt>
\plot \{xa\} \{ya\} \{xb\} \{yb\} /
\setsolid
\drawarrowhead <4pt> [.2,1.1] from \{xa\} \{ya\} to \{xb\} \{yb\}
or % 6=injective arrow
%
% 30 units, the radius for the tail of the arrow
%
% recover the value of auxa
\auxa=\guardauxa%
% makes an adjustment to cope with the tail of the arrow, giving
% space to the semi-circle
\advance \auxa by 30%
%
% Note: the values of (xa,ya) will be modified, they will be
% "pushed" further away from (x,y) in order to acomodate the tail
% of the "arrow"
%
% find the point (xd,yd), the center of a 2pt (20*0.1) circle

25
\ucoord{\number\xO}{\number\xa}{\number\deuc}{\number\auxa}{\number\yd}{1}
\ucoord{\number\yO}{\number\ya}{\number\deuc}{\number\auxa}{\number\xd}{1}
% building the "arrow"
\arrow <4pt> [.2,1.1] from \{\xa\} \{\ya\} to \{\xb\} \{\yb\}
% and its "tail"
\circulararc -180 degrees from \{\xa\} \{\ya\} center at \{\xd\} \{\yd\}
\or % 7=maps "arrow" ("|-->")
% 20
% Note: the values of xmed and ymed will be modified
% find the two points that defines the tail of the arrow (segment
% (xmed,ymed) and (xd,yd))
\ucoord{\number\yO}{\number\yl}{\number\xa}{\number\auxa}{\number\deuc}{\xmed}{-1}
\ucoord{\number\xO}{\number\xl}{\number\ya}{\number\auxa}{\number\deuc}{\ymed}{1}
\ucoord{\number\yO}{\number\yl}{\number\xa}{\number\auxa}{\number\deuc}{\xd}{1}
\ucoord{\number\xO}{\number\xl}{\number\ya}{\number\auxa}{\number\deuc}{\yd}{-1}
% building the "arrow"
\arrow <4pt> [.2,1.1] from \{\xa\} \{\ya\} to \{\xb\} \{\yb\}
% and its "tail"
\setlinear
\plot \{\xmed\} \{\ymed\} \{\xd\} \{\yd\} /
\or % $\hookrightarrow$=surjective arrow ("--->")
% building arrow with the first tip
\arrow <4pt> [.2,1.1] from \{\xa\} \{\ya\} to \{\xb\} \{\yb\}
% and the second tip
\setlinear
\arrow <6pt> [0,.72] from \{\xa\} \{\ya\} to \{\xb\} \{\yb\}
\or % 9=equalline
% by Ruben Debeerst: equal-line
% sets the separation (distance) between the two parallel lines, if
% horizontal or vertical 1pt (10*0.1) is enough, if not 1.1pt (11*0.1)
\auxa=11
\ifnum\number\yO=\number\yl
\auxa=10
\fi
\ifnum\number\xO=\number\xl
\auxa=10
\fi
% the two parallel lines will be given by \{xmed,ymed\}(\xd,\yd), and
% \{xe,ye\}(\xf,\yf)
\ucoord{\number\yO}{\number\yl}{\number\xa}{\number\auxa}{\number\deuc}{\xmed}{-1}
\ucoord{\number\xO}{\number\xl}{\number\ya}{\number\auxa}{\number\deuc}{\ymed}{1}
\ucoord{\number\yO}{\number\yl}{\number\xa}{\number\auxa}{\number\deuc}{\xd}{1}
\ucoord{\number\xO}{\number\xl}{\number\ya}{\number\auxa}{\number\deuc}{\yd}{-1}
\ucoord{\number\yO}{\number\yl}{\number\xb}{\number\auxa}{\number\deuc}{\xmed}{-1}
\ucoord{\number\xO}{\number\xl}{\number\yb}{\number\auxa}{\number\deuc}{\ymed}{1}
\ucoord{\number\yO}{\number\yl}{\number\xb}{\number\auxa}{\number\deuc}{\xd}{1}
\ucoord{\number\xO}{\number\xl}{\number\yb}{\number\auxa}{\number\deuc}{\yd}{-1}
\setlinear
\plot \{\xmed\} \{\ymed\} \{\xe\} \{\ye\} /
\plot \{\xd\} \{\yd\} \{\xf\} \{\yf\} /
\or % 10=double arrow
% sets the separation (distance) between the two parallel lines, if
% horizontal or vertical 2pt is enough, if not 2.5pt. The extra space
% is needed because of the arrow tip.
\auxa=25
\ifnum\number\yO=\number\yl
\auxa=20
\fi
\ifnum\number\xO=\number\xl
\auxa=20
\fi
The two parallel lines will be given by \((x_{med}, y_{med})(x_d, y_d)\), and \((x_e, y_e)(x_f, y_f)\).

\[ \begin{align*}
\text{if } &\text{ \(x_{med} \neq x_d\) and \(y_{med} \neq y_d\),} \\
\text{the area will be given by } &\text{\( (x_{med}, y_{med})(x_d, y_d)\), and} \\
\text{\((x_e, y_e)(x_f, y_f)\).}
\end{align*} \]

\[ \begin{align*}
\text{if } &\text{ \(x_{med} = x_d\) and \(y_{med} = y_d\),} \\
\text{the area will be given by } &\text{\( (x_{med}, y_{med})(x_d, y_d)\).}
\end{align*} \]
\multiply \xO by \expansao
\multiply \yO by \expansao
\xl=\xO\%
\yl=\yO\%
\aux=1\
\multiply \aux by \auxa\%
\advance\xl by \aux\%
\aux=1\
\multiply \aux by \auxb\%
\advance\yl by \aux\%
\arrow <4pt> [.2,1.1] from {\xO} {\yO} to {\xl} {\yl}\\
%%
%% The definition of the command "cmor"
%%
\def\cmor#1 #2(#3,#4)#5{%
  \ifnextchar[{{!cmora{#1}{#2}{#3}{#4}{#5}}}{!cmora{#1}{#2}{#3}{#4}{#5}[0] }\}
\def!cmora#1#2#3#4#5[#6]{%
  \ifcase #2% "pup" (pointing up)
    \auxa=0% x do not change
    \auxb=1% y "up"
  \or% "pdown" (pointing down)
    \auxa=0% x do not change
    \auxb=-1% y "down"
  \or% "pright" (pointing right)
    \auxa=1% x "right"
    \auxb=0% y do not change
  \or% "pleft" (pointing left)
    \auxa=-1% x "left"
    \auxb=0% y do not change
  \fi % the line
  \ifcase #6 % arrow solid
    \modifplot#1% draw the line
    \setaxy#1
  \or % arrow (with tip) dashed
    \setdashes
    \modifplot#1% draw the line
    \setaxy#1
    \setsolid
  \or % arrow (without tip)
    \modifplot#1% draw the line
  \fi % injection morphism
\}
\xO=#3\%
\yO=#4\%
\multiply \xO by \expansao\%
\multiply \yO by \expansao\%
\put {#5} at {\xO} {\yO}\\
%%
%% Command to build the objects
%%\obj(x,y){<text>}[<label>]
%%
\def\obj(#1,#2){%
  \ifnextchar[{{!obja{#1}{#2}}}{!obja{#1}{#2}[Nulo]}}
\def!obja#1#2[#3]#4{%
  \ifnextchar[{{!objb{#1}{#2}{#3}{#4}}}{!objb{#1}{#2}{#3}{#4}[1]}}
\def!objb#1#2#3#4[#5]{%
  \xO=#1\%
  \yO=#2\%
  \def\!pinta{\normalsize$\bullet$}% sets the normal size of the bullet
  \def\!nulo{Nulo}%
  \def\!arg{#3}%
  \!compara{\!arg}{\!nulo}%
}
\fi
}
\catcode\!'=12 % ***** THIS MUST NEVER BE OMITTED (see PiCTeX)