The BCG PostScript Format

Louis-Pascal Tock
INRIA Rhône-Alpes / Verimag

October 1995 (updated May 2000)

Abstract

This document describes the format of the PostScript files generated by `bcg_draw` and used in `bcg_edit`. Such files specify at once the structure of a BCG graph (states and transitions) and its graphical representation printable on any postscript printer. Any graph coded using this format can be modified interactively using the WYSIWYG (What You See Is What You Get) BCG graph editor called `bcg_edit`.

1 Features of a BCG graph

The BCG PostScript Format (BCG PSF) provides users with a powerful way for building any kind of graph. This format replaces the PostScript format generated by the early version of the `bcg_draw` tool described in [Ruf94]. Transitions can be represented either as straight edges, symmetrical curved edges, asymmetrical curved edges or loops (see Figure 1). Moreover, the arrow's shape and the vertex radius can easily be modified by the user.

2 Organization of the PostScript file

The BCG PSF is divided into five sections occurring in the following order:

1. Preamble section;
2. Constant section;
3. Header section;
4. Graph section;
5. Footer section.

Preamble section, header section and footer section are the same for all graphs. Only constant section and graph section have to be updated when modifying an already existing graph.

3 The preamble section

The preamble section should follow the PostScript file conventions. It must contain the graph's name, the creator's name, the coordinates of the bounding box, etc. For instance:

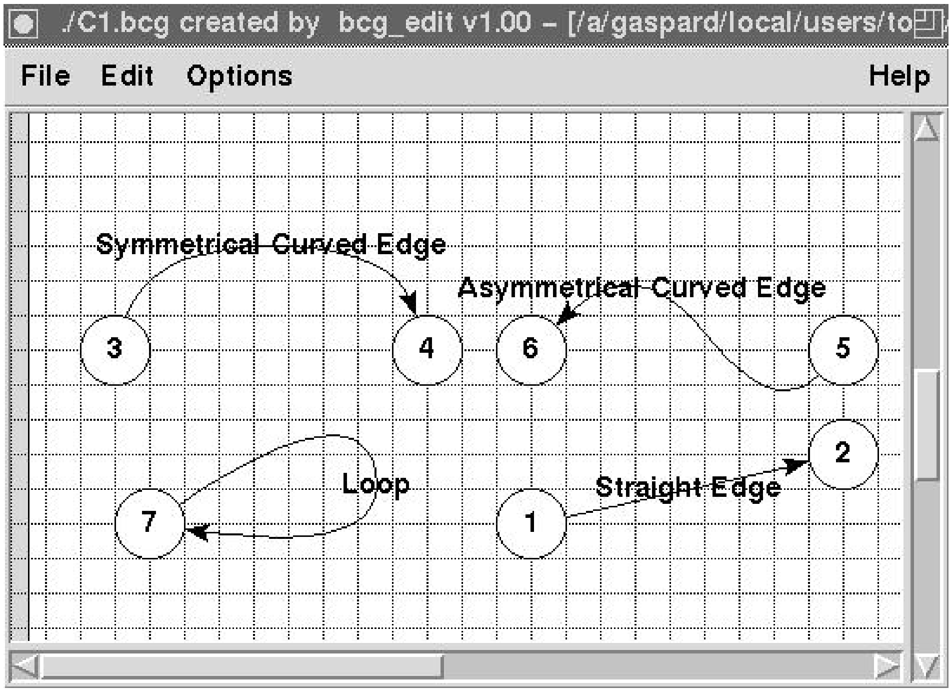

Figure 1: Items available for drawing your graph.

```
%!PS-Adobe-2.0
%%Title: <the name of your graph>
%%Creator: <you or the name of the tool you used>
%%CreationDate: <the current date>
%%Pages: 1
%%BoundingBox: <x1> <y1> <x2> <y2>
%%EndComments
```

4 The constant section

This section is delimited by two special lines, respectively:

```
% BCG_BEGIN_SECTION_1
...
% BCG_END_SECTION_1
```

which are recognized by the `bcg_edit` tool. Its purpose is to define constants used for drawing the graph as the vertex radius, the arrow's shape and the font. This section must contain the following constant declarations.

```
% BCG_BEGIN_SECTION_1
/vertex_radius { <a float number> } def
/arrow_shape_1 { <a float number> } def
/arrow_shape_2 { <a float number> } def
/arrow_shape_3 { <a float number> } def
/Times-Roman findfont 10.000000 scalefont setfont
/text_y_shift { -3 } def
```

```
% BCG_END_SECTION_1
```

vertex_radius is the radius of the vertices (unit: the PostScript point). Figure 2 explains the meaning of the constants arrow_shape_1, arrow_shape_2 and arrow_shape_3 (unit: the PostScript point) which define the shape of the arrows. text_y_shift is used for placing the edge's labels and should not be changed while you use the Times-Roman font given in the example. The current versions of bcg_draw and bcg_edit only use this Times-Roman font. Other fonts are not recognized by bcg_edit.

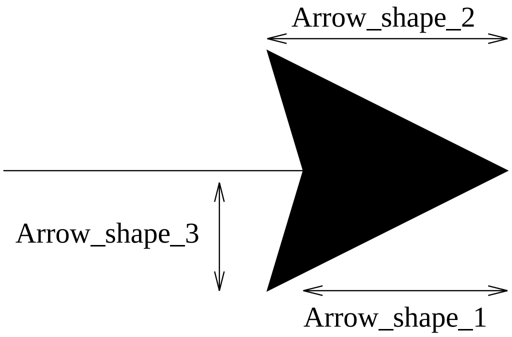

Figure 2: Shape of the arrows.

5 The header section

This section should reproduce the contents of the file $CADP/incl/bcg_draw_header.ps distributed with the CADP toolbox. It defines a collection of PostScript functions to be used in the next section. Examples of these functions are vertex, edge, spline.

6 The graph section

This section is delimited by two lines, respectively:

```
% BCG_BEGIN_SECTION_2
...
% BCG_END_SECTION_2
```

It specifies the graph's structure in using the Bcg Psf functions defined in the header section. These functions occurred in any order and draw an element of the graph: a vertex, a straight edge, a symmetrical curved edge, an asymmetrical curved edge or a loop. For instance, the following lines will produce the graph shown in Figure 3.

```
% BCG_BEGIN_SECTION_2
(4) 244.71 265.15 vertex
(1) 129.552 308.334 vertex
(2) 316.683 236.361 vertex
(20) 388.657 308.334 vertex
(3) 475.025 308.334 vertex
(DIS REQ) 316.717 194.423 323.225 226.301 343.942 194.444 \
          289.492 194.401 306.351 220.416 1.0 \
          spline % 2 2 50.0 -1.57  1.0
```

```
(CON REQ) 170.461 232.039 130.877 296.408 143.947 178.782 \
          201.526 293.94 228.901 275.69 1.0 spline % 1 4  0.96
(DIS IND) 191.773 368.303 133.901 319.519 158.541 382.898 \
          446.036 382.898 468.14 326.043 1.0 \
          spline % 1 3 80.0 1.2  0.52
(CON IND) 270.759 254.731 305.542 240.818 262.351 258.094 \
          1.0 edge % 2 4 0.7
(DATE REQ) 266.296 306.529 379.545 316.143 350.691 340.871 \
           258.496 313.212 249.949 283.414 1.0 \
           spline % 20 4 50.0 1.0  1.52
(CON RESP) 439.038 250.756 465.655 300.838 403.052 250.756 \
           475.025 250.756 404.466 297.795 1.0 \
           spline % 3 20  1.0
(DATE IND) 336.169 255.846 325.169 244.846 375.222 294.899 \
           1.0 edge % 2 20 0.2
% BCG_END_SECTION_2
```

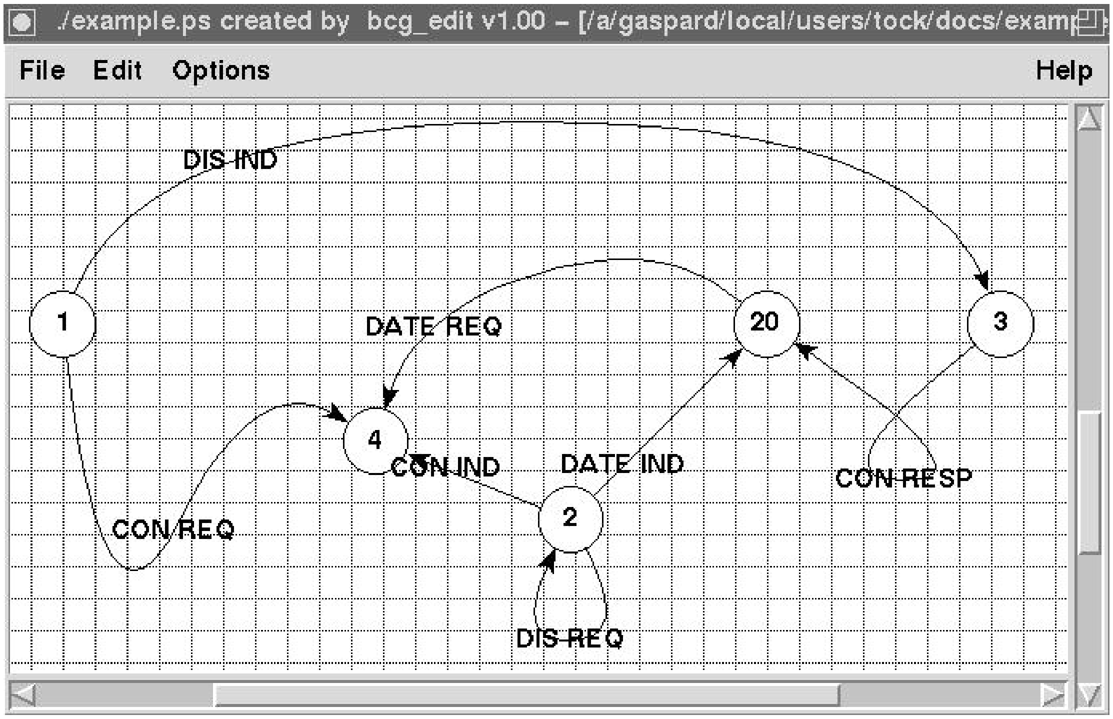
./example.ps created by bcg_edit v1.00 – [/a/gaspard/local/users/tock/docs/examp
File
Edit
Options
Help
DIS IND
1
DATE REQ
20
3
4
CON IND
DATE IND
CON RESP
2
CON REQ
DIS REQ

Figure 3: An example of graph.

There should be one line per vertex or transition. In the previous example, we inserted backslash characters at the end of the line which have to be wrapped to fit into the page width. Although the data occurring after the % character are considered as comments by PostScript interpreters, they should not be removed because they contain additional informations required by the `bcg_edit` tool.

In order to quickly print the graph, Bcg Psf performs few calculations. In the sequel, we give the syntax of the line used to define each item and algorithms allowing to compute values required for the printing (your program will have to carry out these calculations). We use the following notations:

- <`Label`> is a text string corresponding to the label of an edge. According to PostScript conventions, every occurrence of the characters “(”, “)”, or “\” in this string must be preceded by a backslash character, i.e., ‘\(”, “\)”, or “\\”;

- <Number> is an integer;
- Lx, Ly are the abscissa and ordinate of a label's center, given in PostScript point;
- Xn, Yn are the abscissa and ordinate of a control point, given in PostScript point;
- S is the arrow scale factor of an edge. It is contained between 0 and 1.0. When the distance between two vertices is less than the ARROW_SHAPE_1 constant, this factor allows to reduce the arrow's size. In that case, ARROW_SHAPE_1, ARROW_SHAPE_2 and ARROW_SHAPE_3 are multiplied by S before being used;
- orig, dest are the origin and destination state number of an edge. These numbers refer to vertices whose labels were <orig> and <dest>;
- t is the position of a label on an edge;
- X_{orig}, Y_{orig} are the center's abscissa and ordinate of the origin vertex of an edge, given in PostScript point;
- X_{dest}, Y_{dest} are the center's abscissa and ordinate of the destination vertex of an edge, given in PostScript point.

6.1 Vertex description

A line of the form [(<Number>) X Y vertex] creates a vertex centered in X, Y and labeled by Number. This label will be drawn in the center of the vertex. Number becomes the number of the vertex. See figure 4.

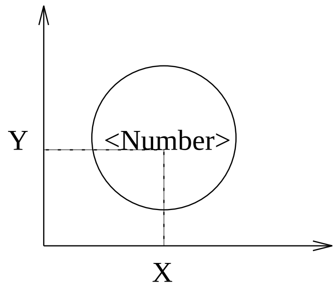

Figure 4: Vertex.

6.2 Straight edge description

A line of the form [(<Label>) Lx Ly X1 Y1 X2 Y2 S edge % orig dest t] creates a straight edge starting at X1, Y1 (the border of the vertex origin), ending at X2, Y2 (the border of the vertex destination) and labeled by Label. The label's position is given by the ratio $t = \frac{d}{D}$ where d is the distance between the origin and the label's center, and D is the distance between the two vertices' borders. For example, a 0.5 ratio means that the label is centered on the edge. An arrow scaled using S is added at the end of the edge. See Figure 5.

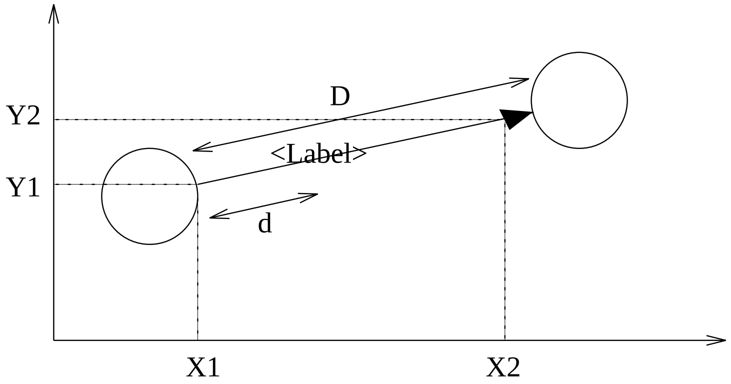

Figure 5: Straight edge.

The algorithm computing `Lx`, `Ly`, `X1`, `Y1`, `X2`, and `Y2` from `orig`, `dest`, `S` and `t` is given below.

```
**if** $X_{dest} = X_{orig}$ **then**
   $dx_1 = 0;\ dx_2 = 0$
   **if** $Y_{dest} > Y_{orig}$ **then**
      $dy_1 = \mathrm{VERTEX\_RADIUS}$
      $dy_2 = \mathrm{VERTEX\_RADIUS} + \mathrm{ARROW\_SHAPE\_1} * S$
   **else**
      $dy_1 = -\mathrm{VERTEX\_RADIUS}$
      $dy_2 = -(\mathrm{VERTEX\_RADIUS} + \mathrm{ARROW\_SHAPE\_1} * S)$
   **endif**
**else**
   $\alpha = \arctan(\frac{Y_{dest} - Y_{orig}}{X_{dest} - X_{orig}})$
   $dx_1 = \cos(\alpha) * \mathrm{VERTEX\_RADIUS}$
   $dy_1 = \sin(\alpha) * \mathrm{VERTEX\_RADIUS}$
   $dx_2 = \cos(\alpha) * (\mathrm{VERTEX\_RADIUS} + \mathrm{ARROW\_SHAPE\_1} * S)$
   $dy_2 = \sin(\alpha) * (\mathrm{VERTEX\_RADIUS} + \mathrm{ARROW\_SHAPE\_1} * S)$
**endif**
**if** $X_{dest} < X_{orig}$ **then**
   $dx_1 = -dx_1; \quad dy_1 = -dy_1$
   $dx_2 = -dx_2; \quad dy_2 = -dy_2$
**endif**
$X_1 = X_{orig} + dx_1; \quad Y_1 = Y_{orig} + dy_1$
$X_2 = X_{dest} - dx_2; \quad Y_2 = Y_{dest} - dy_2$
```

```
(* Label coordinate calculation *)
$d = \sqrt{(X_{dest} - X_{orig})^2 + (Y_{dest} - Y_{orig})^2} * t$
if  $X_{dest} = X_{orig}$  then
   $dx = 0$
   if  $Y_{dest} > Y_{orig}$  then $dy = d$
   else  $dy = -d$
else
   $\alpha = \arctan(\frac{Y_{dest}-Y_{orig}}{X_{dest}-X_{orig}})$
   $dx = d * \cos(\alpha)$
   $dy = d * \sin(\alpha)$
endif
if  $X_{dest} < X_{orig}$  then $dx = -dx$;  $dy = -dy$
$Lx = X_{orig} + dx$
$Ly = Y_{orig} + dy$
```

6.3 Symmetrical curved edge description

A line of the form [(<Label>) Lx Ly X1 Y1 X2 Y2 X3 Y3 X4 Y4 S spline % orig dest ρ θ t] creates a symmetrical curved edge labeled by Label. This curved line is made of two Bézier curves defined by the two control points C (X2, Y2) and C' (X3, Y3). It is symmetrical to the median line of the segment defined by the two vertices. Because C' is the symmetrical point of C with respect to the perpendicular line, it is sufficient to specify the curve's shape by ρ and θ as described in Figure 6. ρ and θ are the polar coordinates of C. An arrow scaled using S is added at the end of the curve.

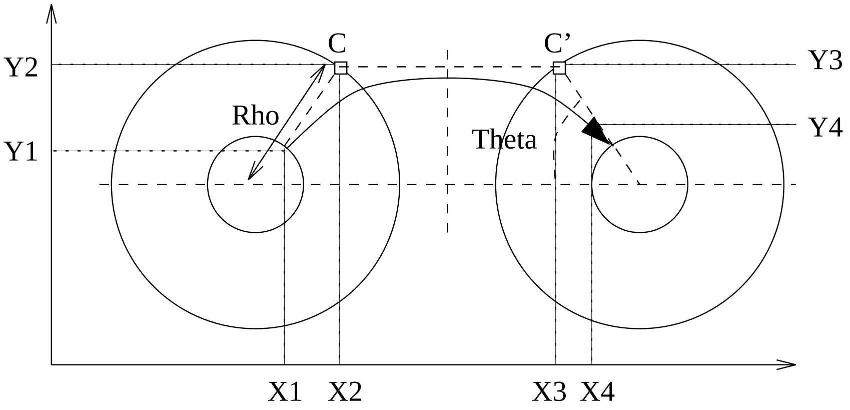

Figure 6: Symmetrical curved edge.

t, contained between 0 and 2, specifies the position of the label on either the first Bézier curve ($0 \leq t \leq 1$) or the second Bézier curve ($1 \leq t \leq 2$). For example, a 1.0 ratio means that the label is centered on the edge.

The algorithm computing Lx, Ly, X1, Y1, X2, Y2, X3, Y3, X4, and Y4 from orig, dest, ρ, θ, S and t is given below.

```
if  $X_{dest} = X_{orig}$  then
  if  $Y_{dest} < Y_{orig}$  then $\alpha = -\frac{\pi}{2}$
  else  $\alpha = \frac{\pi}{2}$
else
  $\alpha = \arctan(\frac{Y_{dest}-Y_{orig}}{X_{dest}-X_{orig}})$
  if  $X_{dest} < X_{orig}$  then $\alpha = -\alpha$
endif
$dx_1 = \cos(\theta + \alpha) * \mathrm{VERTEX\_RADIUS}$
$dy_1 = \sin(\theta + \alpha) * \mathrm{VERTEX\_RADIUS}$
$dx_2 = \cos(\theta + \alpha) * \rho$;   $dy_2 = \sin(\theta + \alpha) * \rho$
$dx_3 = \cos(\theta - \alpha) * \rho$;   $dy_3 = \sin(\theta - \alpha) * \rho$
$dx_4 = \cos(\theta - \alpha) * (\mathrm{VERTEX\_RADIUS} + \mathrm{ARROW\_SHAPE\_1} * S)$
$dy_4 = \sin(\theta - \alpha) * (\mathrm{VERTEX\_RADIUS} + \mathrm{ARROW\_SHAPE\_1} * S)$
$dx_{4L} = \cos(\theta - \alpha) * \mathrm{VERTEX\_RADIUS}$
$dy_{4L} = \sin(\theta - \alpha) * \mathrm{VERTEX\_RADIUS}$
if  $X_{dest} < X_{orig}$  then
  $dx_1 = -dx_1$;   $dx_2 = -dx_2$
  $dx_3 = -dx_3$;   $dx_4 = -dx_4$
endif
$X_1 = X_{orig} + dx_1$;   $Y1 = Y_{orig} + dy_1$
$X_2 = X_{orig} + dx_2$;   $Y2 = Y_{orig} + dy_2$
$X_3 = X_{dest} - dx_3$;   $Y3 = Y_{dest} + dy_3$
$X_4 = X_{dest} - dx_4$;   $Y4 = Y_{dest} + dy_4$
$X_{4L} = X_{dest} - dx_{4L}$;   $Y_{4L} = Y_{dest} + dy_{4L}$
(* Label coordinate calculation *)
if  $t < 1$  then
  $c_0 = X_1$;  $c_1 = Y_1$
  $c_2 = 0.333 * X_1 + 0.667 * X_2$;  $c_3 = 0.333 * Y_1 + 0.667 * Y_2$
  $c_4 = 0.833 * X_2 + 0.167 * X_3$;  $c_5 = 0.833 * Y_2 + 0.167 * Y_3$
  $c_6 = 0.5 * X_2 + 0.5 * X_3$;  $c_7 = 0.5 * Y_2 + 0.5 * Y_3$
else
  $c_0 = 0.5 * X_2 + 0.5 * X_3$;  $c_1 = 0.5 * Y_2 + 0.5 * Y_3$
  $c_2 = 0.167 * X_2 + 0.833 * X_3$;  $c_3 = 0.167 * Y_2 + 0.833 * Y_3$
  $c_4 = 0.667 * X_3 + 0.333 * X_{4L}$;  $c_5 = 0.667 * Y_3 + 0.333 * Y_{4L}$
  $c_6 = X_{4L}$;  $c_7 = Y_{4L}$
  $t = t - 1$
endif
$Lx = c_0 * (1-t)^3 + 3.0 * (c_2 * t * (1-t)^2 + c_4 * t^2 * (1-t)) + c_6 * t^3$
$Ly = c_1 * (1-t)^3 + 3.0 * (c_3 * t * (1-t)^2 + c_5 * t^2 * (1-t)) + c_7 * t^3$
```

6.4 Asymmetrical curved edge description

A line of the form [(`<Label>`) `Lx Ly X1 Y1 X2 Y2 X3 Y3 X4 Y4 S` spline % `orig dest t`] creates an asymmetrical curved edge labeled by `Label`. This curved line is made of two Bézier curves. Its shape is defined by two control points C2 (X2, Y2) and C3 (X3, Y3) as described in Figure 7. An arrow scaled using `S` is added at the end of the curve. `t`, contained between 0 and 2, specifies the position

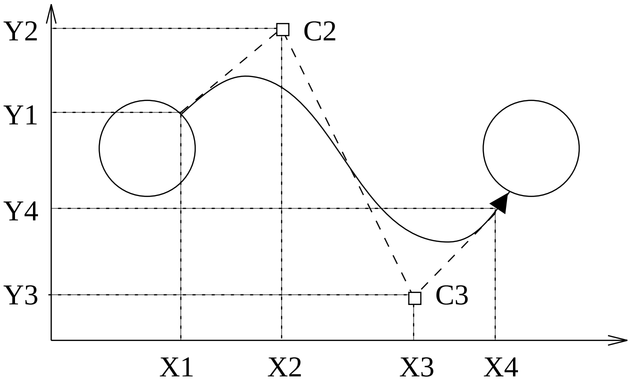

Figure 7: Asymmetrical curved edge.

of the label on either the first Bézier curve ($0 \leq t \leq 1$) or the second Bézier curve ($1 \leq t \leq 2$). For example, a 1.0 ratio means that the label is centered on the edge.

The algorithm computing `Lx`, `Ly`, `X1`, `Y1`, `X4`, and `Y4` from `orig`, `dest`, `X2`, `Y2`, `X3`, `Y3`, `S` and `t` is given below.

```
if $X_{orig} = X_2$ then
  if $Y_2 < Y_{orig}$ then $\theta = -\frac{\pi}{2}$
  else $\theta = \frac{\pi}{2}$
else
  if $X_{dest} \geq X_{orig}$ then
    if $X_2 < X_{orig}$ then $\theta = \pi + \arctan(\frac{Y_2 - Y_{orig}}{X_2 - X_{orig}})$
    else $\theta = \arctan(\frac{Y_2 - Y_{orig}}{X_2 - X_{orig}})$
  else
    if $X_2 < X_{orig}$ then $\theta = -\arctan(\frac{Y_2 - Y_{orig}}{X_2 - X_{orig}})$
    else $\theta = \pi - \arctan(\frac{Y_2 - Y_{orig}}{X_2 - X_{orig}})$
  endif
endif
$dx_1 = \cos(\theta) * \text{VERTEX\_RADIUS}$
$dy_1 = \sin(\theta) * \text{VERTEX\_RADIUS}$
if $X_{dest} = X_3$ then
    if $Y_{dest} < Y_3$ then $\theta = -\frac{\pi}{2}$
    else $\theta = \frac{\pi}{2}$
else
  if $X_{dest} \geq X_{orig}$ then
    if $X_3 < X_{dest}$ then $\theta = \pi + \arctan(\frac{Y_3 - Y_{dest}}{X_3 - X_{dest}})$
    else $\theta = \arctan(\frac{Y_3 - Y_{dest}}{X_3 - X_{dest}})$
  else
    if $X_3 < X_{orig}$ then $\theta = -\arctan(\frac{Y_3 - Y_{dest}}{X_3 - X_{dest}})$
    else $\theta = \pi - \arctan(\frac{Y_3 - Y_{dest}}{X_3 - X_{dest}})$
  endif
endif
```

```
$dx_4 = \cos(\theta) * (\text{VERTEX\_RADIUS} + \text{ARROW\_SHAPE\_1} * S)$
$dy_4 = \sin(\theta) * (\text{VERTEX\_RADIUS} + \text{ARROW\_SHAPE\_1} * S)$
$dx_{4L} = \cos(\theta) * \text{VERTEX\_RADIUS}$
$dy_{4L} = \sin(\theta) * \text{VERTEX\_RADIUS}$
if  $X_{dest} < X_{orig}$  then $dx_1 = -dx_1$;   $dx_4 = -dx_4$
$X_1 = X_{orig} + dx_1$;   $Y_1 = Y_{orig} + dy_1$
$X_4 = X_{dest} - dx_4$;   $Y_4 = Y_{dest} - dy_4$
$X_{4L} = X_{dest} - dx_{4L}$;   $Y_{4L} = Y_{dest} - dy_{4L}$
(* Label coordinate calculation *)
if  $t < 1$   then
   $c_0 = X_1$;  $c_1 = Y_1$
   $c_2 = 0.333 * X_1 + 0.667 * X_2$;  $c_3 = 0.333 * Y_1 + 0.667 * Y_2$
   $c_4 = 0.833 * X_2 + 0.167 * X_3$;  $c_5 = 0.833 * Y_2 + 0.167 * Y_3$
   $c_6 = 0.5 * X_2 + 0.5 * X_3$;  $c_7 = 0.5 * Y_2 + 0.5 * Y_3$
else
   $c_0 = 0.5 * X_2 + 0.5 * X_3$;  $c_1 = 0.5 * Y_2 + 0.5 * Y_3$
   $c_2 = 0.167 * X_2 + 0.833 * X_3$;  $c_3 = 0.167 * Y_2 + 0.833 * Y_3$
   $c_4 = 0.667 * X_3 + 0.333 * X_{4L}$;  $c_5 = 0.667 * Y_3 + 0.333 * Y_{4L}$
   $c_6 = X_{4L}$;  $c_7 = Y_{4L}$
   $t = t - 1$
endif
$Lx = c_0 * (1-t)^3 + 3.0 * (c_2 * t * (1-t)^2 + c_4 * t^2 * (1-t)) + c_6 * t^3$
$Ly = c_1 * (1-t)^3 + 3.0 * (c_3 * t * (1-t)^2 + c_5 * t^2 * (1-t)) + c_7 * t^3$
```

6.5 Loop description

A line of the form [(<Label>) `Lx Ly X1 Y1 X2 Y2 X3 Y3 X4 Y4 S` spline % `orig dest` ρ α `t`] creates a loop labeled by `Label`. This curved line is made of two Bézier curves. Its shape is defined by ρ and α as described in Figure 8. ρ specifies the length of the loop and α its angle around the vertex. θ is given by ρ*θ=*constant* where *constant* usually equals 40 (value used in `bcg_draw` and in `bcg_edit`).

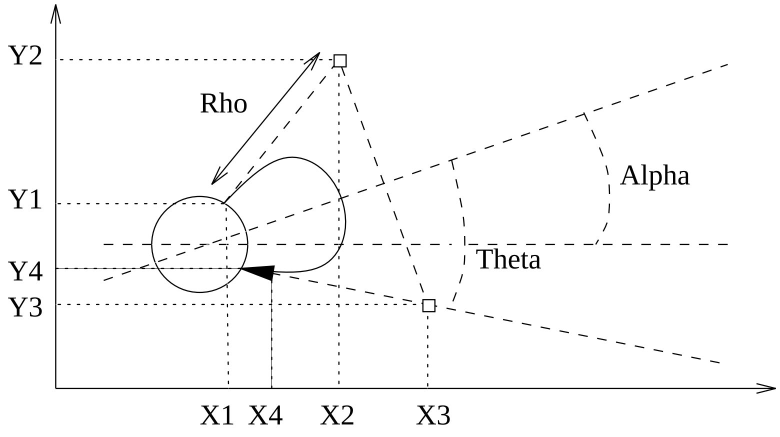

Figure 8: Loop.

This constant binds θ and ρ in order to preserve the shape of the loop when ρ varies. An arrow scaled using `S` is added at the end of the curve. For the loop, we have `orig=dest`. `t`, contained between 0 and 2, specifies the position of the label on either the first Bézier curve ($0 \leq t \leq 1$) or the second Bézier curve ($1 \leq t \leq 2$). For example, a 1.0 ratio means that the label is centered on the edge.

The algorithm computing `Lx`, `Ly`, `X1`, `Y1`, `X4`, and `Y4` from `orig`, `dest`, ρ, α, `S` and `t` is given below.

```
$\theta = \frac{LOOP\_CONST}{\rho}$   (LOOP_CONST=40 for bcg_draw and bcg_edit)
$dx_1 = \cos(\alpha + \theta) * \text{VERTEX\_RADIUS}$
$dy_1 = \sin(\alpha + \theta) * \text{VERTEX\_RADIUS}$
$dx_2 = \cos(\alpha + \theta) * \rho$
$dy_2 = \sin(\alpha + \theta) * \rho$
$dx_3 = \cos(\alpha - \theta) * \rho$
$dy_3 = \sin(\alpha - \theta) * \rho$
$dx_4 = \cos(\alpha - \theta) * (\text{VERTEX\_RADIUS} + \text{ARROW\_SHAPE\_1} * S)$
$dy_4 = \sin(\alpha - \theta) * (\text{VERTEX\_RADIUS} + \text{ARROW\_SHAPE\_1} * S)$
$dx_{4L} = \cos(\alpha - \theta) * \text{VERTEX\_RADIUS}$
$dy_{4L} = \sin(\alpha - \theta) * \text{VERTEX\_RADIUS}$
$X_1 = X_{orig} + dx_1; \quad Y_1 = Y_{orig} + dy_1$
$X_2 = X_{orig} + dx_2; \quad Y_2 = Y_{orig} + dy_2$
$X_3 = X_{orig} + dx_3; \quad Y_3 = Y_{orig} + dy_3$
$X_4 = X_{orig} + dx_4; \quad Y_4 = Y_{orig} + dy_4$
$X_{4L} = X_{orig} + dx_{4L}; \quad Y_{4L} = Y_{orig} + dy_{4L}$
(* Label coordinate calculation *)
if  $t < 1$  then
   $c_0 = X_1;\ c_1 = Y_1$
   $c_2 = 0.333 * X_1 + 0.667 * X_2;\ c_3 = 0.333 * Y_1 + 0.667 * Y_2$
   $c_4 = 0.833 * X_2 + 0.167 * X_3;\ c_5 = 0.833 * Y_2 + 0.167 * Y_3$
   $c_6 = 0.5 * X_2 + 0.5 * X_3;\ c_7 = 0.5 * Y_2 + 0.5 * Y_3$
else
   $c_0 = 0.5 * X_2 + 0.5 * X_3;\ c_1 = 0.5 * Y_2 + 0.5 * Y_3$
   $c_2 = 0.167 * X_2 + 0.833 * X_3;\ c_3 = 0.167 * Y_2 + 0.833 * Y_3$
   $c_4 = 0.667 * X_3 + 0.333 * X_{4L};\ c_5 = 0.667 * Y_3 + 0.333 * Y_{4L}$
   $c_6 = X_{4L};\ c_7 = Y_{4L}$
   $t = t - 1$
endif
$Lx = c_0 * (1 - t)^3 + 3.0 * (c_2 * t * (1 - t)^2 + c_4 * t^2 * (1 - t)) + c_6 * t^3$
$Ly = c_1 * (1 - t)^3 + 3.0 * (c_3 * t * (1 - t)^2 + c_5 * t^2 * (1 - t)) + c_7 * t^3$
```

7 The footer section

This section should reproduce the contents of the file `$CADP/incl/bcg_footer.ps` distributed with the CADP toolbox.

References

[Ruf94] Renaud Ruffiot. Définition et réalisation d'un atelier logiciel pour l'étude des systèmes de transitions. Mémoire d'ingénieur CNAM, INRIA Rhône-Alpes, Grenoble, December 1994.

Contents

List of Figures