Autograph version 0.4.1

Paul Gastin
LITP
Université Paris 7
2, place Jussieu
F-75251 Paris Cedex 05
`Paul.Gastin@litp.ibp.fr`

15 novembre 1996

Abstract

`Autograph` est un ensemble de macros pour dessiner des automates et des graphes sous l'environnement picture de LaTeX 2.09 ou LaTeX 2_ε.

Contents

1 Usage

Mettre le fichier `autograph.sty` dans un répertoire accessible par LaTeX, i.e., quelque part dans le chemin `TEXINPUTS`.

Les macros de `autograph.sty` utilisent des macros définies dans `epic.sty`. Il est donc aussi nécessaire d'avoir le fichier `epic.sty` dans un répertoire accessible par LaTeX.

Pour utiliser les macros définies dans `autograph.sty`, il suffit de faire

```
\documentclass{article}                    % Pour LaTeX 2e
\usepackage{epic,autograph}
```

ou

```
\documentstyle[epic,autograph]{article}    % Pour LaTeX 2.09
```

2 Description des macros pour les automates

2.1 Configurations

unitlength

Voir manuel LaTeX. Valeur par défaut 4pt (environ 1.4mm). Ce choix est motivé par le fait que, en standard, LaTeX ne sait dessiner que des cercles dont le diamètre est multiple de 4pt. Pour les graphes, on peut choisir ce qu'on veut.

```
ex: \unitlength=1mm          % Pour les graphes
    \unitlength=4pt          % Pour les automates (~ 1,4mm)
```

setmaxbezier

Pour définir le nombre maximal de points des courbes de Bézier utilisées pour dessiner les boucles et les courbes. Dans la limite de ce maximum, le nombre de points utilisés est déterminé automatiquement pour assurer une courbe continue. Nombre entier. Valeur par défaut: 500.

Pour accélérer la compilation et l'impression on a intérêt à réduire grandement ce nombre dans les versions non définitives.

```
ex: \setnbptbezier{50}     % accélère la compilation et l'impression
```

2.2 États

setstatediam

Fixe la valeur courante en `\unitlength` du diamètre des états. Valeur par défaut: 6.

En standard, LaTeX ne sait dessiner que des cercles dont le diamètre est multiple de 4pt et au maximum 40pt. Donc, la valeur maximale est 10 pour `\unitlength`=4pt. Si une valeur non multiple de 4pt est choisie comme diamètre des états (combinaison entre `\setstatediam` et `\unitlength`), les transitions n'arriveront plus précisément sur le cercle des états.

```
ex: \setstatediam{10} % bien pour les slides
```

setrepeatedstatediam

Fixe la valeur courante en `\unitlength` du diamètre du cercle intérieur des états répétés. Valeur par défaut: 5.

```
ex: \setrepeatedstatediam{8} % bien pour les slides
```

letstate nom=(x,y)

Définition d'un état. Le nom ne contient que des lettres. Attention à ne pas mettre d'espace dans la définition. Un état est défini par les coordonnées (x,y) en `\unitlength` de son centre.

`\letstate` admet comme arguments optionnels le diamètre en `\unitlength` de l'état et le diamètre en `\unitlength` du cercle intérieur de l'état s'il est répété. La deuxième valeur n'est utilisée que par `\drawrepeatedstate`. On ne peut cependant pas donner une seule de ces deux valeurs. Par défaut, les valeurs courantes fixées par `\setstatediam` et `\setrepeatedstatediam` sont utilisées.

On peut redéfinir un état. Les nouvelles données seront utilisées à partir de la redéfinition.

```
ex: \letstate A=(0,0)
    \letstate[10,8] C=(-15,5)
```

Dessin d'un état

On fournit le nom de l'état et son étiquette. Pour `\drawinitialstate` et `\drawfinalstate`, le premier argument est une lettre parmi t(op), b(ottom), l(eft), r(ight) qui spécifie où se trouve la marque sur l'état. Il est optionnel avec par défaut l pour `\drawinitialstate` et r pour `\drawfinalstate`.

```
ex: \drawstate(A){3}
    \drawinitialstate(B){0}
    \drawfinalstate(C){1}
    \drawrepeatedstate(D){2}
    \drawstate(E){}                 % état sans nom
    \drawinitialstate[b](F){4}      % initial et final
    \drawfinalstate[t](F){}
```

Si on souhaite une autre direction pour la marque de l'état initial (final) on peut dessiner un état normal avec `\drawstate` puis dessiner avec `\drawtrans` une transition sans étiquette partant (arrivant) sur un état factice de diamètre 0.

```
ex: \letstate A=(0,0) \drawstate(A){1}
    \letstate[0,0] AIa=(-3.00,-5.20) % état initial à -120 degrés
    \letstate[0,0] AIb=(-4.24,-4.24) % état initial à -135 degrés
    \letstate[0,0] AIc=(-5.20,-3.00) % état initial à -150 degrés
    \letstate[0,0] AFa=( 5.20,-3.00) % état final à -30 degrés
    \letstate[0,0] AFb=( 4.24,-4.24) % état final à -45 degrés
    \letstate[0,0] AFc=( 3.00,-5.20) % état final à -60 degrés
    \drawtrans(AIa,A){} \drawtrans(AIb,A){} \drawtrans(AIc,A){}
    \drawtrans(A,AFa){} \drawtrans(A,AFb){} \drawtrans(A,AFc){}
```

2.3 Transitions

settranslabelskip

Pour définir l'espace en `\unitlength` entre la transition et son étiquette. Valeur par défaut: 1

```
ex: \settranslabelskip{2}
```

settransdecal

Pour définir le décalage en `\unitlength` entre une transition droite (drawtrans) et la droite joignant les deux états. Valeur par défaut: 0.

Pour éviter leur chevauchement, on utilise une valeur non nulle lorsqu'il y a une transition dans chaque sens entre les deux états. Une valeur positive (resp. négative) décale la transition vers sa gauche (resp. droite) en regardant dans le sens de la transition. Si la valeur est trop importante, la transition risque de ne pas toucher l'état.

```
ex: \settransdecal{1}      % bien pour éviter le chevauchement.
```

setprofcurve

Distance en `\unitlength` entre le milieu de la transition courbe et la droite joignant les deux états. Valeur par défaut: 3.

Une valeur positive (resp. négative) déforme la transition vers sa gauche (resp. droite) en regardant dans le sens de la transition.

```
ex: \setprofcurve{15}
    \setprofcurve{-7}
    \setprofcurve{0}        % trace une droite
```

Dessin d'une transition

On fournit les noms des états joints par la transition et l'étiquette de la transition. Le premier argument est une lettre parmi l(eft), r(ight) qui spécifie où se trouve l'étiquette par rapport à la transition lorsqu'on regarde dans le sens de la transition. Il est optionnel avec par défaut l.

drawtrans

Trace une transition droite.

```
ex: \drawtrans(A,B){$G,g,\gamma$}
    \drawtrans(D,A){}                  % transition sans étiquette
    \drawtrans[l](B,A){$h$}
```

drawcurvedtrans

Trace une transition courbe définie par une Bézier quadratique. Cela nécessite beaucoup plus de calculs que `\drawtrans`. La concavité de la courbe est fixée par `\setprofcurve`. Si cette distance est nulle, la courbe est en fait droite. Dans certains cas, le résultat est meilleur qu'avec `\drawtrans`.

```
ex: \drawcurvedtrans[r](A,C){$E,\varepsilon$}
```

drawqbeziertrans

Trace une transition courbe définie par une Bézier quadratique. En plus des données précédentes, on fournit les coordonnées du point qui définit les tangentes de la courbe de Bézier. Cela permet plus de souplesse dans la forme de la transition que `\drawcurvedtrans`.

```
ex: \drawqbeziertrans[r](A)(50,-30)(B){$C,c,\gamma$}
```

drawcbeziertrans

Trace une transition courbe définie par une Bézier cubique. En plus des données précédentes, on fournit les coordonnées des deux points qui définissent les tangentes de la courbe de Bézier. Permet un changement de concavité dans la courbe de la transition. Permet de dessiner des boucles dans une direction quelconque.

```
ex: \drawcbeziertrans(A)(-10,20)(10,-20)(B){$B,b,\beta$}
```

2.4 Boucles

setloopdiam

Définition du diamètre des boucles en `\unitlength`. Valeur par défaut: 6.

```
ex: \setloopdiam{10} % bien pour les slides
```

drawloop

On fournit le nom de l'état sur lequel porte la boucle et l'étiquette de la boucle. Le premier argument est une lettre parmi t(op), b(ottom), l(eft), r(ight) qui spécifie où se trouve la boucle. Il est optionnel avec par défaut t.

```
ex: \drawloop[b](A){$A,a,\alpha$}
    \drawloop[r](B){$A,a,\alpha$}
    \drawloop(C){}                          % boucle sans étiquette
```

Pour dessiner une boucle dans une autre direction, il faut utiliser `\drawcbeziertrans`.

```
ex: % Boucle à 45 degrés.
    % Remarque: 15.87 = sqrt(7)*6 où 6 est le diamètre commun
    % de la boucle et de l'état.
    \drawcbeziertrans(A)(0,15.87)(15.87,0)(A){$C,c,\gamma$}
```

3 Description des macros pour les graphes

setvertexdiam

Fixe le diamètre par défaut en `\unitlength` d'un sommet. Valeur par défaut: 6.

letvertex A=(x,y)

Définition d'un sommet. Le nom ne contient que des lettres. Attention à ne pas mettre d'espace dans la définition. Un sommet est défini par les coordonnées (x,y) en `\unitlength` de son centre.

`\letstate` admet comme argument optionnel le diamètre en `\unitlength` du sommet. Par défaut, la valeur fixée par `\setvertexdiam` est utilisée. Ce diamètre est utilisé par `\drawcircledvertex` et pour déterminer l'origine et l'extrémité des arêtes et des boucles même si le sommet n'est pas encerclé.

On peut redéfinir un sommet. Les nouvelles données seront utilisées à partir de la redéfinition.

```
ex: \letvertex A=(0,0)
    \letvertex[10] C=(-15,5)
```

drawcircledvertex et drawvertex

Dessine un sommet avec ou sans cercle autour. Voir `\drawstate` page 4.

setedgelabelskip

Espace entre l'arête et son étiquette. Voir `\settranslabelskip` page 4.

setedgedecal

Décalage entre l'arête et l'axe des états. Voir `\settransdecal` page 4.

drawedge et drawundirectededge

Dessine une arête droite dirigée ou non. Voir `\drawtrans` page 5.

setprofcurve

Distance en `\unitlength` entre le milieu de l'arête courbe et la droite joignant les deux états. Voir `\setprofcurve` page 5.

drawcurvededge et drawundirectedcurvededge

Dessine une arête courbe dirigée ou non. Voir `\drawcurvedtrans` page 5.

drawqbezieredge et drawundirectedqbezieredge

Dessine une arête dirigée ou non suivant une courbe de Bézier quadratique. Voir `\drawqbeziertrans` page 5.

drawcbezieredge et drawundirectedcbezieredge

Dessine une arête dirigée ou non suivant une courbe de Bézier cubique. Voir `\drawcbeziertrans` page 5.

setloopdiam

Définition du diamètre des boucles. Voir `\setloopdiam` page 6.

drawloop et drawundirectedloop

Dessine une boucle dirigée ou non. Voir `\drawloop` page 6.

4 Autres macros

drawvector

Dessine un vecteur entre les points spécifiés. Aucune limitation sur la pente.

```
ex: \drawvector(10,70)(70,20)
```

qbeziervector

Courbe de Bézier quadratique avec une flèche au bout. Le nombre maximum de points de la courbe est commandé par `\setmaxbezier`.

```
ex: \qbeziervector(2,0)(16,15)(8,0)
```

cbezier et cbeziervector

Courbe de Bézier cubique avec ou sans flèche au bout. Permet un changement de concavité. Permet de dessiner une approximation d'un cercle de rayon arbitraire. Beaucoup plus coûteux que le "circle" de latex. Le nombre maximum de points de la courbe est commandé par `\setmaxbezier`.

```
ex: \cbeziervector(-4,0)(3,-15)(-17,-15)(-10,0)
    % Approximation d'un cercle de rayon 18 par 2 courbes.
    % Rem: 24 = 18*4/3
    \cbezier(-18,0)(-18,24)(18,24)(18,0)
    \cbezier(-18,0)(-18,-24)(18,-24)(18,0)
    % Triple changement de concavité par raccord de 2 courbes.
    \cbezier(-18,20)(-15,-20)(5,20)(0,0)
    \cbeziervector(0,0)(-5,-20)(15,20)(18,-20)
```

5 Problèmes connus et solutions

- Si on dessine trop de points dans une même page, par exemple avec de nombreuses courbes de Bézier, on peut dépasser la capacité mémoire de TeX.

  ```
  ! TeX capacity exceeded, sorry [main memory size=196608].
  \mon@bezier ...vance \t@ \@ne
  If you really absolutely need more capacity,
  you can ask a wizard to enlarge me.
  ```

  Solutions:

  – Augmenter la mémoire allouée à TeX.

  – Limiter le nombre de points tracés par les courbes de Bézier avec `\setmaxbezier`.

- Si on dessine des objets vraiment trop grands on peut avoir un dépassement de capacité arithmétique. C'est dû à la façon de calculer une racine carrée par epic.sty.

  ```
  ! Arithmetic overflow.
  \sqrtandstuff ...s \multiply \num@segments by 457
                                    \divide \num@segments \@x@...
  ```

  Solution: Augmenter la valeur de `\d@my@unit` dans le fichier autograph.sty.

- Si vous rencontrez d'autres problèmes, vous pouvez me les signaler par courrier électronique.

6 Description des évolutions

6.1 Projets

Je souhaite utiliser Postscript pour dessiner les flèches (droite, courbe, Bézier, ...) et les cercles des états. Les avantages attendus sont la levée de la limitation imposée par LaTeX sur le diamètre des cercles, une meilleure précision des dessins et une accélération radicale de la compilation en TeX des dessins contenant des courbes de Bézier. En particulier, le problème de dépassement de capacité mémoire devrait être définitivement résolu.

D'autre part, j'envisage de changer le nom *autograph* de cet ensemble de macros car celui-ci est déjà utilisé par un logiciel de traitement d'automates. Toute proposition est la bienvenue.

6.2 De 0.4 vers 0.4.1

Correction d'une erreur qui survenait lorsque l'étiquette d'une transition comportait une parenthèse fermante.

6.3 De 0.3 vers 0.4

- Introduction de courbes de Bézier cubiques.
- Ajout des macros `\drawqbeziertrans`, `\drawcbeziertrans`, `\drawqbezieredge`, `\drawundirectedqbezieredge` et `\drawundirectedloop`, `\qbeziervector`, `\cbezier` et `\cbeziervector`.
- Possibilité d'utiliser des valeurs non entières pour les coordonnées des états, des points, pour les diamètres, etc...

- Amélioration des algorithmes de dessin des transitions courbes et des boucles. Permet de supprimer les problèmes rencontrés avec la version précédente lorsque le diamètre d'une boucle était supérieur à 1.5 fois le diamètre de l'état ou lorsqu'une transition courbe était trop courte.

- Je n'utilise plus les courbes de Bézier quadratiques définies par latex. L'utilisation de "bezier.sty" avec LaTeX209 n'est donc plus nécessaire.

- Utilisation de `\setmaxbezier` à la place de `\setnbptbezier` qui définissait pour LaTeX 2.09 précisément le nombre de points tracés par les courbes de Bézier et pas le maximum comme pour LaTeX 2_ε. Afin d'assurer une certaine compatibilité, l'ancienne commande `\setnbptbezier` est toujours acceptée mais est maintenant synonyme de `\setmaxbezier`.

- La macro `\setprecision` ne sert plus a rien. Elle existe toujours pour assurer la compatibilité avec la version précédente mais n'a plus aucun effet. D'ailleurs, la version 0.3 comportait une erreur: la profondeur d'une courbe n'était pas remise à jour correctement lors d'un changement de précision.

6.4 De 0.2 vers 0.3

- utilise l(eft) et r(ight) à la place de t(op) et b(ottom) comme options pour le dessin des transitions et des arêtes. l(eft) se comporte exactement comme t(op). Par contre, r(ight) est différent de b(ottom). r(ight) a pour seul effet de placer l'étiquette à droite de la transition tandis que b(ottom) avait aussi pour effet d'inverser le sens du décalage fixé par `\settransdecal` ou `\setprofcurve`. Le résultat obtenu correspond plus à l'intuition, surtout si on fixe une valeur négative pour `\settransdecal` ou `\setprofcurve`. Afin d'assurer la compatibilité, les anciennes options t(op) et b(ottom) sont toujours acceptées, mais avec émission d'un warning.

- `\letstate` admet comme arguments optionnels le diamètre de l'état et le diamètre du cercle intérieur de l'état répété. On ne peut pas donner une seule de ces deux valeurs. Par défaut, les valeurs fixées par `\setstatediam` et `\setrepeatedstatediam` sont utilisées.

- `\letvertex` admet comme argument optionnel le diamètre du sommet. Par défaut, la valeur fixée par `\setvertexdiam` est utilisée.

- Le diamètre d'une boucle est maintenant indépendant de celui des états. Pour avoir un bon raccord, il faut toutefois que le rapport entre le diamètre de la boucle et celui de l'état n'excède pas 1,5.

- Il est maintenant possible de dessiner une transition (droite ou courbe) entre deux états de diamètres différents. Cette possibilité permet entre autres de dessiner une transition courbe avec changement de concavité. On utilise un état intermédiaire de diamètre 0 comme dans l'exemple suivant.

```
\letstate[4,3]  A=(-20,0) \drawstate(A){1}
\letstate[10,8] B=(20,0)  \drawstate(B){$x>2$}
\letstate[0,0]  C=(0,0)
\setprofcurve{8}  \drawundirectedcurvededge(A,C){}
\setprofcurve{-8} \drawcurvedtrans[r](C,B){$A,a,\alpha$}
```

6.5 De 0.1 vers 0.2

Modification de la valeur par défaut de `\setprecision`. L'ancienne valeur pouvait provoquer un "arithmetic overflow" lors du dessin de longues transitions.

7 Autograph et Alpha

Alpha est un merveilleux éditeur shareware pour Macintosh créé par Pete Keleher. On peut se le procurer sur le web ou par ftp:

```
http://www.cs.umd.edu/~keleher/alpha.html
ftp://www.cs.umd.edu/pub/faculty/keleher/Alpha/
```

Tom Scavo (`http://web.syr.edu/~trscavo/`) a écrit un ensemble de macros pour Alpha permettant d'éditer un fichier LaTeX avec une facilité remarquable.

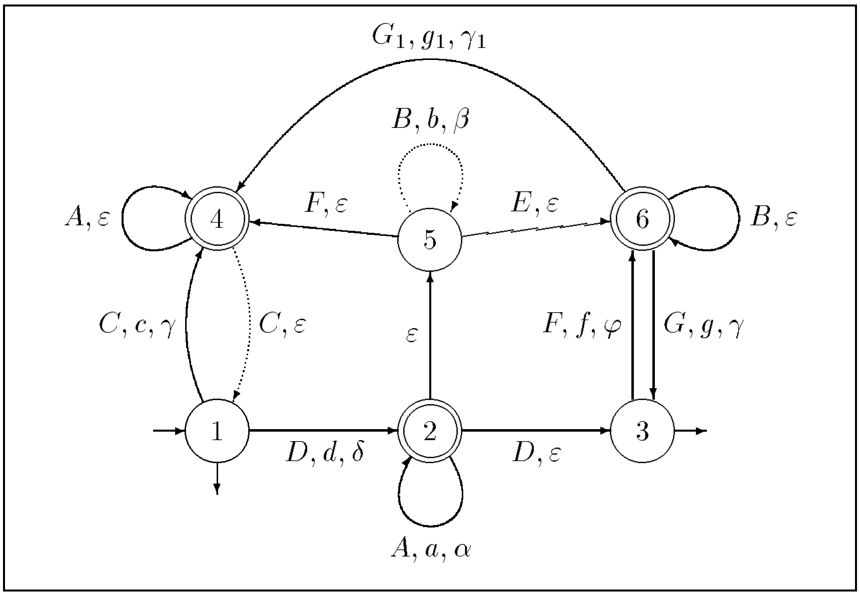
Vous pouvez configurer Alpha pour éditer avec la même facilité des figures utilisant **autograph**. Vous trouverez les fichiers nécessaires sur le web:

```
http://www-litp.ibp.fr:80/~gastin/
```

8 Exemples

8.1 Dessin d'un automate

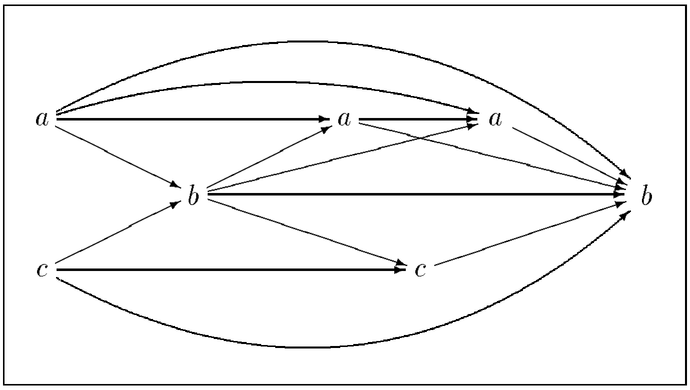
Exemple de dessin d'automate. On notera les transitions d'étiquettes E, ε et F, ε tracées respectivement par `\drawtrans` et `\drawcurvedtrans` avec une profondeur de courbe nulle. Le résultat est bien meilleur dans le second cas mais nécessite beaucoup plus de calculs.




```
\begin{center}
\begin{picture}(80,55)(-20,-15)
\thinlines
\put(-20,-15){\framebox(80,55){}}

\letstate A=(0,0)   \drawinitialstate(A){1}\drawfinalstate[b](A){}
\letstate B=(20,0)  \drawrepeatedstate(B){2}
\letstate C=(40,0)  \drawfinalstate(C){3}
\letstate D=(0,20)  \drawrepeatedstate(D){4}
\letstate E=(20,18) \drawstate(E){5}
\letstate F=(40,20) \drawrepeatedstate(F){6}

\drawloop[b](B){$A,a,\alpha$}    \drawloop[l](D){$A,\varepsilon$}
{ \setmaxbezier{50} \drawloop(E){$B,b,\beta$} }
\drawloop[r](F){$B,\varepsilon$}

\drawtrans[r](A,B){$D,d,\delta$} \drawtrans[r](B,C){$D,\varepsilon$}
{ \settransdecal{1}
  \drawtrans(C,F){$F,f,\varphi$} \drawtrans(F,C){$G,g,\gamma$}}
\drawtrans(B,E){$\varepsilon$}   \drawtrans(E,F){$E,\varepsilon$}
{ \setprofcurve{0} \drawcurvedtrans[r](E,D){$F,\varepsilon$} }

\drawcurvedtrans(A,D){$C,c,\gamma$}
{ \setmaxbezier{50} \drawcurvedtrans(D,A){$C,\varepsilon$} }
{ \setprofcurve{-15} \drawcurvedtrans[r](F,D){$G_1,g_1,\gamma_1$} }
\end{picture}
\end{center}
```

8.2 Dessin d'un graphe dirigé

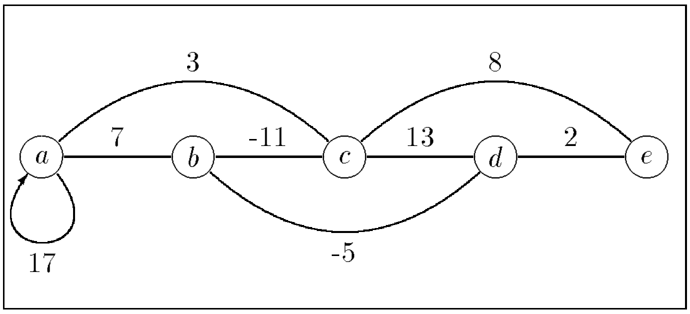
Exemple de dessin d'un graphe dirigé. Noter le choix du diamètre un peu plus grand pour les sommets AAA et BB pour éviter que les flèches ne se touchent.




```
\begin{center}
\unitlength=1mm
\begin{picture}(90,44)(-5,-22)
\thinlines
\put(-5,-22){\framebox(90,44){}}

\setvertexdiam{4}
\letvertex A=(0,10)        \drawvertex(A){$a$}
\letvertex C=(0,-10)       \drawvertex(C){$c$}
\letvertex B=(20,0)        \drawvertex(B){$b$}
\letvertex AA=(40,10)      \drawvertex(AA){$a$}
\letvertex[5] AAA=(60,10)  \drawvertex(AAA){$a$}
\letvertex CC=(50,-10)     \drawvertex(CC){$c$}
\letvertex[6] BB=(80,0)    \drawvertex(BB){$b$}

\drawedge(A,B){}       \drawedge(C,B){}
\drawedge(B,AA){}      \drawedge(AA,AAA){}
\drawedge(B,CC){}      \drawedge(AAA,BB){}
\drawedge(CC,BB){}     \drawedge(A,AA){}
\drawedge(B,BB){}      \drawedge(C,CC){}
\drawedge(B,AAA){}     \drawedge(AA,BB){}

\setprofcurve{5}       \drawcurvededge(A,AAA){}
\setprofcurve{15}      \drawcurvededge(A,BB){}
\setprofcurve{-15}     \drawcurvededge(C,BB){}
\end{picture}
\end{center}
```

8.3 Dessin d'un graphe non dirigé

Exemple de dessin d'un graphe non dirigé.

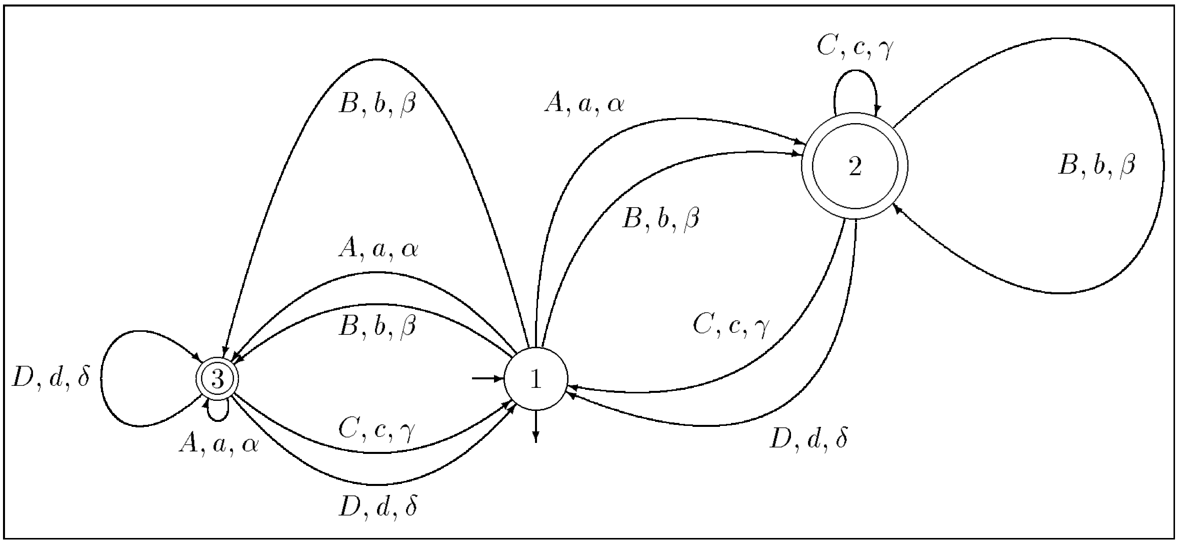



```
\begin{center}
\unitlength=1mm
\begin{picture}(90,30)(-5,-15)
\thinlines
\put(-5,-15){\framebox(90,30){}}

\setvertexdiam{6}
\letvertex A=(0,0)      \drawcircledvertex(A){$a$}
\letvertex B=(20,0)     \drawcircledvertex(B){$b$}
\letvertex C=(40,0)     \drawcircledvertex(C){$c$}
\letvertex D=(60,0)     \drawcircledvertex(D){$d$}
\letvertex E=(80,0)     \drawcircledvertex(E){$e$}

\drawundirectededge(A,B){7}
\drawundirectededge(B,C){-11}
\drawundirectededge(C,D){13}
\drawundirectededge(D,E){2}

\setprofcurve{10}
\drawundirectedcurvededge(A,C){3}
\drawundirectedcurvededge(C,E){8}
\setprofcurve{-10}
\drawundirectedcurvededge[r](B,D){-5}

\drawundirectedloop[b](A){17}
\end{picture}
\end{center}
```

8.4 Astuces

On peut utiliser des transitions courbes de différentes profondeurs pour tracer plusieurs transitions entre deux états.

On peut mettre l'étiquette d'une boucle à l'intérieur de la boucle en donnant une valeur négative à `\settranslabelskip`.

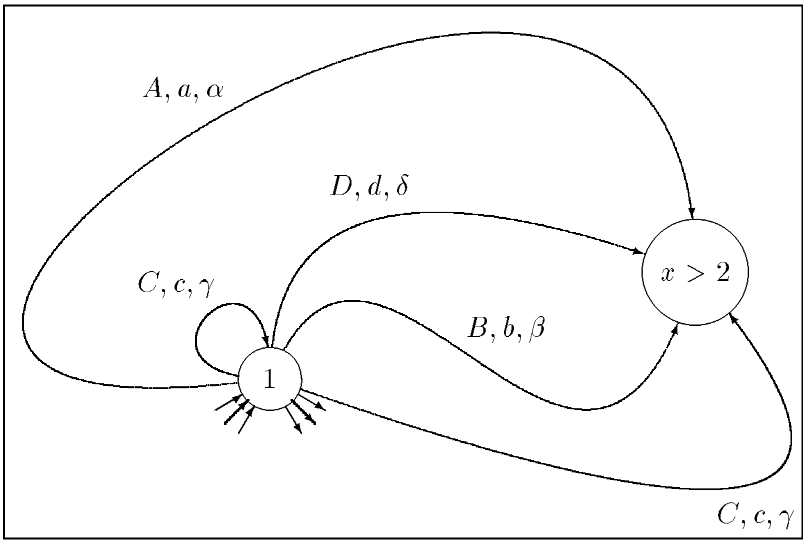



```
\begin{center}
\begin{picture}(110,50)(-50,-15)
\thinlines
\put(-50,-15){\framebox(110,50){}}

\letstate A=(0,0)   \drawinitialstate(A){1}\drawfinalstate[b](A){}
\letstate[10,8] B=(30,20) \drawrepeatedstate(B){2}
\letstate[4,3]  C=(-30,0) \drawrepeatedstate(C){3}

\setprofcurve{15}
\drawcurvedtrans(A,B){$A,a,\alpha$}    \drawcurvedtrans(B,A){$D,d,\delta$}
\setprofcurve{10}
\drawcurvedtrans[r](A,B){$B,b,\beta$}  \drawcurvedtrans[r](B,A){$C,c,\gamma$}
\setprofcurve{-10}
\drawcurvedtrans[r](A,C){$A,a,\alpha$} \drawcurvedtrans[r](C,A){$D,d,\delta$}
\setprofcurve{-7}
\drawcurvedtrans(A,C){$B,b,\beta$}     \drawcurvedtrans(C,A){$C,c,\gamma$}
{ \setprofcurve{-30} \settranslabelskip{3} \setmaxbezier{1000}
  \drawcurvedtrans(A,C){$B,b,\beta$} }

\setloopdiam{2}   \drawloop[b](C){$A,a,\alpha$}
\setloopdiam{9}   \drawloop[l](C){$D,d,\delta$}
\setloopdiam{4}   \drawloop[t](B){$C,c,\gamma$}
\setloopdiam{24}  \setmaxbezier{1000}  \settranslabelskip{-10}
\drawloop[r](B){$B,b,\beta$}
\end{picture}
\end{center}
```

8.5 Encore des astuces

Pour un état initial (ou final), on peut faire des flèches ayant un angle arbitraire, il faut définir un état fictif de diamètre 0 et tracer une transition de cet état fictif vers l'état initial.

Les courbes de Bézier quadratiques ou cubiques permettent de mieux contrôler la forme des transitions. On peut aussi faire une boucle dans une direction arbitraire en utilisant `\drawcbeziertrans`.

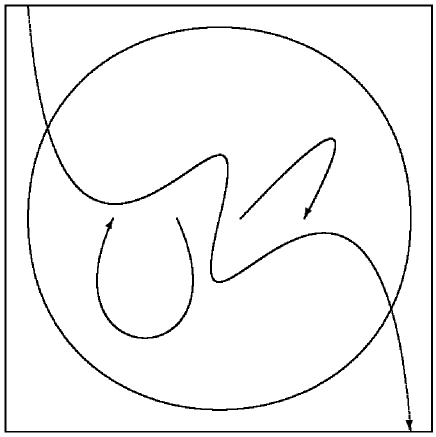



```
\begin{center}
\begin{picture}(75,50)(-25,-15)
\thinlines
\put(-25,-15){\framebox(75,50){}}

\letstate[10,8] B=(40,10) \drawstate(B){$x>2$}
\letstate A=(0,0) \drawstate(A){1}
\letstate[0,0] AIa=(-3.00,-5.20) % état initial à -120 degrés
\letstate[0,0] AIb=(-4.24,-4.24) % état initial à -135 degrés
\letstate[0,0] AIc=(-5.20,-3.00) % état initial à -150 degrés
\letstate[0,0] AFa=( 5.20,-3.00) % état final à -30 degrés
\letstate[0,0] AFb=( 4.24,-4.24) % état final à -45 degrés
\letstate[0,0] AFc=( 3.00,-5.20) % état final à -60 degrés
\drawtrans(AIa,A){} \drawtrans(AIb,A){} \drawtrans(AIc,A){}
\drawtrans(A,AFa){} \drawtrans(A,AFb){} \drawtrans(A,AFc){}

% Boucle à 135 degrés.
% Rem: 15.87 = sqrt(7)*6 ou 6 est le diamètre commun
% de la boucle et de l'état.
\drawcbeziertrans(A)(-15.87,0)(0,15.87)(A){$C,c,\gamma$}

\drawqbeziertrans(A)(0,25)(B){$D,d,\delta$}
{\setmaxbezier{1000} \drawqbeziertrans[r](A)(70,-25)(B){$C,c,\gamma$}}

{\setmaxbezier{1000} \drawcbeziertrans(A)(-70,-10)(40,70)(B){$A,a,\alpha$}}
\drawcbeziertrans(A)(10,25)(30,-25)(B){$B,b,\beta$}
\end{picture}
\end{center}
```

8.6 Toujours des astuces

LaTeX ne permet pas de tracer un cercle de grand diamètre. On peut tracer une approximation d'un cercle avec deux courbes de Bézier cubiques. L'erreur est inférieure à 2%.

On peut aussi faire des courbes avec un nombre arbitraire de changements de concavité en concaténant plusieurs courbes de Bézier cubiques.



```
\begin{center}
\setmaxbezier{800}
\begin{picture}(40,40)(-20,-20)
\thinlines
\put(-20,-20){\framebox(40,40){}}

% Cercle de rayon 18 formé de 2 courbes de Bézier. Erreur < 2%.
% Remarque: 24 = 18 * 4/3.
\cbezier(-18,0)(-18,24)(18,24)(18,0)
\cbezier(-18,0)(-18,-24)(18,-24)(18,0)

\qbeziervector(2,0)(16,15)(8,0)
\cbeziervector(-4,0)(3,-15)(-17,-15)(-10,0)

\cbezier(-18,20)(-15,-20)(5,20)(0,0)
\cbeziervector(0,0)(-5,-20)(15,20)(18,-20)

\end{picture}
\end{center}
```