Let me start with a few notes about how this document with the pretty-printed MetaPost source code was created. Since John Hobby's MetaPost is almost (but not quite) similar to Don Knuth's METAFONT, it's easy to use the `MFT` utility which converts METAFONT source to TEX source.

Prior to this conversion it is necessary to apply a few modifications. This is best done by Ulrik Vieths's `SED` script `mp2mft.sed` from `CTAN:graphics/metapost/contrib/misc/`. `MFT` should be invoked with the `-s mp2` option, also supplied by Ulrik's bundle. The resulting file can be processed by either plain TEX or (as for this application) by PDFTEX.

In the first case it is necessary to include Tom Rokicki's `epsf.tex` macros, in the latter you'll need Hans Hagen's `supp-pdf.tex` and the usual `\pdfoutput=1` and `\pdfcompresslevel=9`.

Including MetaPost graphics in LaTEX documents is straightforward with the `graphicx` bundle and the latest `pdftex.def` by David Carlisle, Sebastian Rahtz, and Hans Hagen.

beginfig(1); % Start the first graphic.

% Declare a bunch of variables; points . . .
pair *zast*, *leftrange*, *rightrange*, *bottomrange*, *toprange*;
pair *topleft*, *topright*, *bottomleft*, *bottomright*;

path *restriction_two*, *inrange*; % paths . . .

picture *cutlabel*; % and an image.

% Set the coordinates; *origin* equals (0, 0).
zast = *origin*; *leftrange* = $(-4cm, 0)$; *rightrange* = $(4cm, 0)$;
bottomrange = $(0, -1cm)$; *toprange* = $(0, 4cm)$;
topleft = $(-4cm, 4cm)$; *bottomleft* = $(-4cm, 0)$;
topright = $(4cm, 4cm)$; *bottomright* = $(4cm, 0)$;

% MetaPost does not (yet) provide the possibility for plotting functions,
% so we have to use intermediate points for the parabola.
restriction_two = $(-2cm, 4cm)$
  for $i = -8$ **upto** 8: .. $(i/4, (i/4) * (i/4)) * 1cm$ **endfor**;

% buildcycle creates a closed path from several (sub)paths.
inrange = buildcycle(*bottomleft* -- *topleft* -- $(-2cm, 4cm)$,
  restriction_two, $(2cm, 4cm)$ -- *topright* -- *bottomright* -- *bottomleft*);

% Note that PostScript works in an "additive" way, so we have to start
% with the background. Instead of 'white' you can use brilliant
% TEXnicolor.
fill *inrange* **withcolor** .8*white*;

% Introduce the x and y axis.
drawarrow *leftrange* -- *rightrange*;
drawarrow *bottomrange* -- *toprange*;

draw *restriction_two*; % Draw the parabola.

% Mark various points of interest. This is done by TEX itself.
label *lrt*(**btex** ... **etex**, *zast*); % TEX label: `"$x^% \ast$"`
label *bot*(**btex** ... **etex**, *rightrange*); % TEX label: `"$x_1$"`
label *lft*(**btex** ... **etex**, *toprange*); % TEX label: `"$x_2$"`

cutlabel := **thelabel** *rt*(
  btex ... **etex**, *zast* + $(0, 1cm)$); % TEX label: `"$\nabla g_1(x^\ast)$"`
unfill bbox *cutlabel*; **draw** *cutlabel*;

label *lft*(**btex** ... **etex**, *zast* − $(0, 1cm)$); % TEX label: `"$\nabla g_2(x^\ast)$"`

cutlabel := **thelabel** *top*(**btex** ... **etex**, (3*cm*, 0)); % TEX label: `"$g_1(x)=0$"`
unfill bbox *cutlabel*; **draw** *cutlabel*;

cutlabel := **thelabel** *lrt*(**btex** ... **etex**, (2*cm*, 4*cm*)); % TEX label: `"$g_2(x)=0$"`
unfill bbox *cutlabel*; **draw** *cutlabel*;

cutlabel := **thelabel**(**btex** ... **etex**, (−3*cm*, 2*cm*)); % TEX label: `"$P$"`
unfill bbox *cutlabel*; **draw** *cutlabel*;

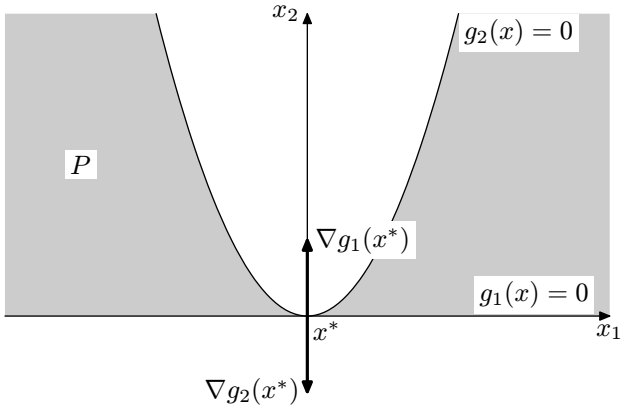
% Draw the two gradients with a somewhat thicker line.
pickup pencircle scaled 1.2*pt*;

drawarrow *zast* -- (*zast* + (0, 1*cm*));
drawarrow *zast* -- (*zast* − (0, 1*cm*));

endfig; % The first graphic is finished.

The final output can be included into your TEX document in the usual fashion. If you want to create standard output you can use `\epsfbox{nlpgraph.1}`. For PDF output use `\convertMPtoPDF{nlpgraph.1}{1}{1}`.

LaTEX users should be familiar with what they have to do. Note that you *must* rename the files written by MetaPost to something like `nlpgraph1.mps` for `pdftex.def`'s sake.



OK, just as I wrote in my announcement in the `pdftex` list, the code for these graphics is rather quick and dirty. Without comments even I don't quite understand what the following does. However, I just don't have the time to fill the gaps now. Also the code could be improved by using more parameters and letting MetaPost do the calculations.

On the other hand, the main idea should be clear: define some fixed points and their relative positions, draw lines and fill areas, finally put in labels and special markers. And remember not to overwrite earlier stuff that should be visible.

beginfig(2);

pair *rightrange*, *leftrange*, *toprange*, *bottomrange*;

rightrange = (8*cm*, 0); *leftrange* = *origin*;
toprange = (0, 5*cm*); *bottomrange* = (0, −1*cm*);

label *bot*(**btex** ... **etex**, *rightrange*); % TEX label: `"$x_1$"`
label *lft*(**btex** ... **etex**, *toprange*); % TEX label: `"$x_2$"`

pair *xast*; *xast* = (4*cm*, 2*cm*);
label *llft*(**btex** ... **etex**, *xast*); % TEX label: `"$x^% \ast$"`

path *T*; *T* = (1*cm*, 5*cm*) -- (7*cm*, −1*cm*);
label *llft*(**btex** ... **etex**, **point** 0 **of** *T*); % TEX label: `"$T$"`

label(**btex** ... **etex**, (1*cm*, 2*cm*)); % TEX label: `"$P$"`

path *g_i*; *g_i* = .75*toprange* .. {(1, −1)}*xast* .. .6*rightrange*;

label *bot*(**btex** ... **etex**, .6*rightrange*); % TEX label: `"$g_i(x)=0$"`

path *L*; *L* = (2*cm*, 5*cm*){(1, −1)} .. {*right*}(8*cm*, 2*cm*);

pair *SP*, *SPT*, *TSPT*;

SP = *L* intersectionpoint ((xpart(*xast*), −*infinity*)
-- (xpart(*xast*), *infinity*));

SPT = **point** .666667 **of** *L*;

label *urt*(**btex** ... **etex**, *SP*); % TEX label: `"$L(x^% \ast,u^\ast)$"`
label *urt*(**btex** ... **etex**, *SPT*); % TEX label: `"$L(x,u^\ast)$"`

TSPT = *T* intersectionpoint (*SPT* -- (xpart(*SPT*), −*infinity*));

path *Konvex*;
Konvex = buildcycle(*L*, *SPT* -- *TSPT* -- **point** 0 **of** *T* -- **point** 0 **of** *L*);

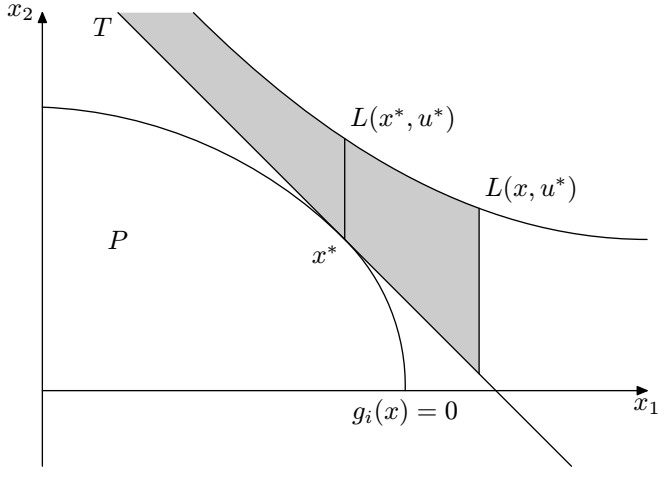
fill *Konvex* **withcolor** .8*white*;

for *i* = *T*, *g_i*, *L*: **draw** *i*; **endfor**;

draw *xast* -- *SP*; **draw** *SPT* -- *TSPT*;

drawarrow *leftrange* -- *rightrange*;
drawarrow *bottomrange* -- *toprange*;

endfig; % 2



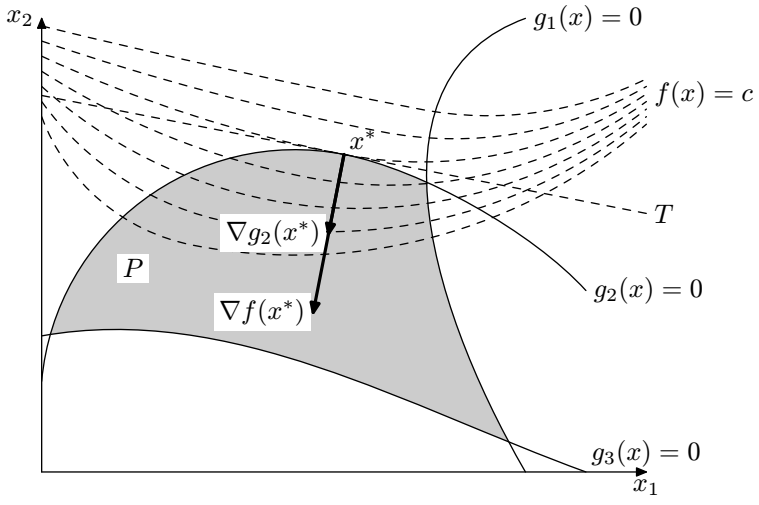
```
beginfig(3);

pair rightrange, toprange;
rightrange = (8cm, 0);  toprange = (0, 6cm);

for i = rightrange, toprange: drawarrow origin -- i;  endfor;

label bot(btex ... etex, rightrange);       % TeX label: "$x_1$"
label lft(btex ... etex, toprange);         % TeX label: "$x_2$"

pair xast;  xast = (.5 xpart(rightrange), .7 ypart(toprange));
label urt(btex ... etex, xast);         % TeX label: "$x^% \ast$"

path g[];

g[1] = (.8 xpart(rightrange), ypart(toprange)){dir 200}
    .. {dir 300}.8rightrange;
g[2] = (.9 xpart(rightrange), .4 ypart(toprange))
    .. tension 1.2 .. xast .. (0, .2 ypart(toprange));
g[3] = .9rightrange{dir 160} .. {dir 190}.3toprange;

path P;  P = buildcycle(g[1], g[2], g[3]);
fill P withcolor .8white;

for i = 1 upto 3: draw g[i];  endfor;

label rt(btex ... etex, point 0 of g[1]);       % TeX label: "$g_1(x)=0$"
label rt(btex ... etex, point 0 of g[2]);       % TeX label: "$g_2(x)=0$"
label urt(btex ... etex, point 0 of g[3]);      % TeX label: "$g_3(x)=0$"

numeric Txast;

(Txast, whatever) = g[2]intersectiontimes
    ((xpart(xast), -infinity) -- (xpart(xast), infinity));

pair A, B, C;

A = - direction Txast of g[2];
B = xast + whatever * A;  xpart(B) = xpart(rightrange);
C = xast + whatever * A;  xpart(C) = 0;

draw C -- B dashed evenly;  label rt(btex ... etex, B);       % TeX label: "$T$"

pair Left[], Middle[], Right[];

for i = -4 upto 2:
    Left[i] = (0cm, 5.5cm + .2i * cm);
    Middle[i] = xast + i * (.5cm, .3cm);
    Right[i] = (xpart(rightrange), 5cm + .1i * cm);
    draw Left[i] .. Middle[i]{A} .. Right[i] dashed evenly;
endfor;

label rt(btex ... etex, Right[0]);      % TeX label: "$f(x)=c$"

pickup pencircle scaled 1.2pt;

pair nablaf, nablag;

nablaf = xast + (.7A rotated -90);
nablag = xast + (.35A rotated -90);

picture cutlabel;

cutlabel := thelabel lft(btex ... etex, nablaf);        % TeX label: "$\nabla f(x^\ast)$"
unfill bbox cutlabel;  draw cutlabel;
```

cutlabel := **thelabel** *lft*(**btex** ... **etex**, *nablag*); % TEX label: `"$\nabla g_2(x^\ast)$"`
unfill bbox *cutlabel*; **draw** *cutlabel*;

drawarrow *xast* -- *nablaf*; **drawarrow** *xast* -- *nablag*;

cutlabel := **thelabel** *urt*(**btex** ... **etex**, (1*cm*, 2.5*cm*)); % TEX label: `"$P$"`
unfill bbox *cutlabel*; **draw** *cutlabel*;

endfig; % 3



beginfig(4);

pair *rightrange*, *toprange*;
rightrange = (8*cm*, 0); *toprange* = (0, 5*cm*);

label *bot*(**btex** ... **etex**, *rightrange*); % TEX label: `"$x_1$"`
label *lft*(**btex** ... **etex**, *toprange*); % TEX label: `"$x_2$"`

path *g*[], *f*;

g[1] = .3*toprange*{dir 25} ..
{*up*}(.5 xpart(*rightrange*), ypart(*toprange*));
g[2] = .7*toprange*{*right*} .. {dir 250}.9*rightrange*;

label *rt*(**btex** ... **etex**, **point** 1 **of** *g*[1]); % TEX label: `"$g_1(x)=0$"`
label *urt*(**btex** ... **etex**, **point** 0 **of** *g*[2]); % TEX label: `"$g_2(x)=0$"`

pair *xast*; *xast* = *g*[1] intersectionpoint *g*[2];

label *top*(**btex** ... **etex**, *xast*); % TEX label: `"$x^% \ast$"`

f = .5*toprange* .. *xast* .. (.8 xpart(*rightrange*), .9 ypart(*toprange*));

label *rt*(**btex** ... **etex**, **point** 2 **of** *f*); % TEX label: `"$f(x)$"`

numeric *Txast*[];
pair *A*[], *B*[], *C*[];

(*Txast*[1], *whatever*) = *g*[1]intersectiontimes
((xpart(*xast*), −*infinity*) -- (xpart(*xast*), *infinity*));

A[1] = − **direction** *Txast*[1] **of** *g*[1];
B[1] = *xast* + (*A*[1] rotated 90);
C[1] = *xast* + *whatever* ∗ (*A*[1] rotated 90); ypart(*C*[1]) = 0;

(*Txast*[2], *whatever*) = *g*[2]intersectiontimes
((xpart(*xast*), −*infinity*) -- (xpart(*xast*), *infinity*));

A[2] = − **direction** *Txast*[2] **of** *g*[2];
B[2] = *xast* + .5(*A*[2] rotated 90);
C[2] = *xast* + *whatever* ∗ (*A*[2] rotated 90); ypart(*C*[2]) = 0;

(*Txast*[3], *whatever*) = *f*intersectiontimes
((xpart(*xast*), −*infinity*) -- (xpart(*xast*), *infinity*));

A[3] = − **direction** *Txast*[3] **of** *f*;
B[3] = *xast* + .8(*A*[3] rotated 90);

path *P*;

P = buildcycle(*origin* -- *C*[1] -- *xast*, *g*[1], *origin* -- *toprange*);
fill *P* **withcolor** .9*white*;

P := buildcycle(*C*[2] -- *rightrange*, *g*[2], *xast* -- *C*[2]);
fill *P* **withcolor** .9*white*;

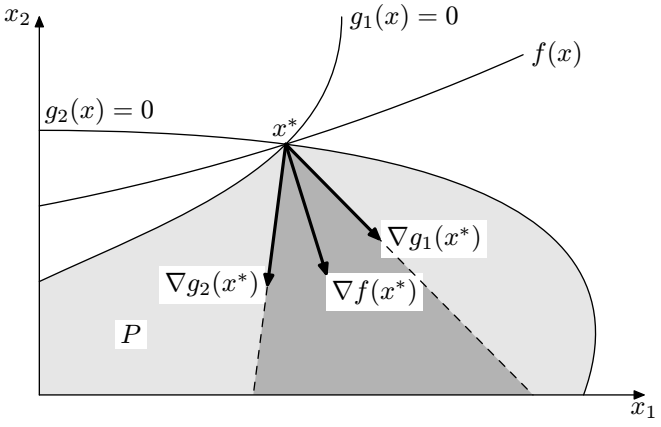
fill *xast* -- *C*[1] -- *C*[2] -- cycle **withcolor** .7*white*;

picture *cutlabel*;

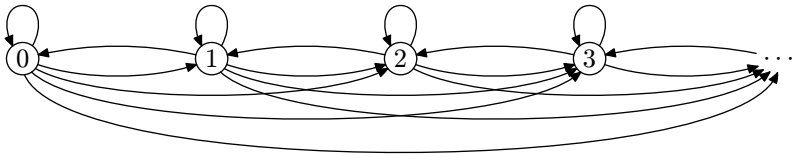
cutlabel = **thelabel** *lrt*(**btex** ... **etex**, *B*[3]); % TEX label: `"$\nabla f(x^\ast)$"`
unfill bbox *cutlabel*; **draw** *cutlabel*;

cutlabel := **thelabel** *rt*(**btex** ... **etex**, *B*[1]); % TEX label: `"$\nabla g_1(x^\ast)$"`
unfill bbox *cutlabel*; **draw** *cutlabel*;

```
cutlabel := thelabel lft(btex ... etex, B[2]);       % TeX label: "$\nabla g_2(x^\ast)$"
unfill bbox cutlabel; draw cutlabel;

cutlabel := thelabel lrt(btex ... etex, (1cm, 1cm));       % TeX label: "$P$"
unfill bbox cutlabel; draw cutlabel;

for i = 1 upto 2: draw xast -- C[i] dashed evenly; endfor;
for i = rightrange, toprange: drawarrow origin -- i; endfor;
for i = 1 upto 2: draw g[i]; endfor;

draw f;

pickup pencircle scaled 1.2pt;
for i = 1 upto 3: drawarrow xast -- B[i]; endfor;

endfig; % 4
```

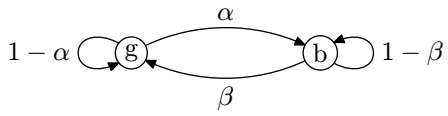


```
input boxes; % TEX labels are best handled as bordered objects

% Define shortcuts for drawing arrows from and to boxes.
vardef cuta(suffix a, b) expr p =
    drawarrow p cutbefore bpath a cutafter bpath b;
    point .5 * length p of p enddef;

vardef cutb(suffix a, b) expr p =
    drawarrow p cutbefore bpath a cutafter bpath b;
    point .8 * length p of p enddef;

vardef cutc(suffix a, b) expr p =
    drawarrow p cutbefore bpath a cutafter bpath b;
    point .2 * length p of p enddef;

% Define self-referential relations.
vardef loop(suffix a) expr p =
    cuta(a, a)a_c{curl 0} .. a_c + p .. {curl 0}a_c enddef;

beginfig(5);

% Define some circular objects with TEX labels.
circleit Knoten[0](btex ... etex);      % TEX label: "$0$"
circleit Knoten[1](btex ... etex);      % TEX label: "$1$"
circleit Knoten[2](btex ... etex);      % TEX label: "$2$"
circleit Knoten[3](btex ... etex);      % TEX label: "$3$"
circleit Knoten[4](btex ... etex);      % TEX label: "$% \cdots$"

% Declare the relative position for the labels and draw them.
% Also, apply self-refence.
for i = 0 upto 3:
    Knoten[i]c = origin + i * (2.5cm, 0); drawboxed(Knoten[i]);
    label(btex ... etex, loop(Knoten[i])(0, 20pt));      % TEX label: "\relax"
endfor;

Knoten[4]c = origin + 4 * (2.5cm, 0);
Knoten[4]dx = Knoten[4]dy; drawunboxed(Knoten[4]);

% Draw the lower arrows pointing right.
% Note the use of tension with varying parameters.
for i = 0 upto 3:
    for j = i + 1 upto 4:
        label(btex ... etex, cuta(Knoten[i], Knoten[j])      % TEX label: "\relax"
            Knoten[i]c{dir((i - j) * 25)}
                .. tension((j - i) * 1.2) .. Knoten[j]c);
    endfor;
endfor;

% Draw the upper arrows pointing left.
for i = 1 upto 4:
    label(btex ... etex, cuta(Knoten[i], Knoten[i - 1])      % TEX label: "\relax"
        Knoten[i]c{dir 165} .. Knoten[i - 1]c);
endfor;

endfig; % Here's the result:
```

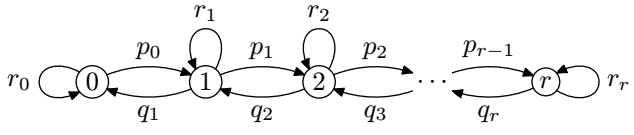


The material in this MetaPost source file is quite elementary. First a set of nodes (Knoten) are defined and placed, then the arrows are drawn and labelled.

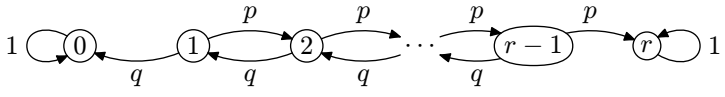
```
beginfig(6);

circleit Knoten[0](btex ... etex);  Knoten[0]c = origin;        % TeX label: "g"
circleit Knoten[1](btex ... etex);  Knoten[1]c = origin + (2.5cm, 0);        % TeX label: "b"

drawboxed(Knoten[0], Knoten[1]);

label lft(btex ... etex, loop(Knoten[0])(-20pt, 0));        % TeX label: "$1-\alpha$"
label rt(btex ... etex, loop(Knoten[1])(20pt, 0));        % TeX label: "$1-\beta$"

label top(btex ... etex, cuta(Knoten[0], Knoten[1])        % TeX label: "$\alpha$"
    Knoten[0]c{dir 30} .. Knoten[1]c);
label bot(btex ... etex, cuta(Knoten[1], Knoten[0])        % TeX label: "$\beta$"
    Knoten[1]c{dir 210} .. Knoten[0]c);

endfig;
```



```
beginfig(7);

circleit Knoten[0](btex ... etex);      % TeX label: "$0$"
circleit Knoten[1](btex ... etex);      % TeX label: "$1$"
circleit Knoten[2](btex ... etex);      % TeX label: "$2$"
circleit Knoten[3](btex ... etex);      % TeX label: "$% \cdots$"
circleit Knoten[4](btex ... etex);      % TeX label: "$r$"

for i = 0 upto 4: Knoten[i]c = origin + i * (1.5cm, 0); endfor;

drawboxed(Knoten[0], Knoten[1], Knoten[2], Knoten[4]);
drawunboxed(Knoten[3]);

label lft(btex ... etex, loop(Knoten[0])(-20pt, 0));      % TeX label: "$r_0$"
label rt(btex ... etex, loop(Knoten[4])(20pt, 0));        % TeX label: "$r_r$"
label top(btex ... etex, loop(Knoten[1])(0, 20pt));       % TeX label: "$r_1$"
label top(btex ... etex, loop(Knoten[2])(0, 20pt));       % TeX label: "$r_2$"

label top(btex ... etex, cuta(Knoten[0], Knoten[1])       % TeX label: "$p_0$"
    Knoten[0]c{dir 30} .. Knoten[1]c);
label top(btex ... etex, cuta(Knoten[1], Knoten[2])       % TeX label: "$p_1$"
    Knoten[1]c{dir 30} .. Knoten[2]c);
label top(btex ... etex, cuta(Knoten[2], Knoten[3])       % TeX label: "$p_2$"
    Knoten[2]c{dir 30} .. Knoten[3]c);
label top(btex ... etex, cuta(Knoten[3], Knoten[4])       % TeX label: "$p_{r-1}$"
    Knoten[3]c{dir 30} .. Knoten[4]c);

label bot(btex ... etex, cuta(Knoten[1], Knoten[0])       % TeX label: "$q_1$"
    Knoten[1]c{dir 210} .. Knoten[0]c);
label bot(btex ... etex, cuta(Knoten[2], Knoten[1])       % TeX label: "$q_2$"
    Knoten[2]c{dir 210} .. Knoten[1]c);
label bot(btex ... etex, cuta(Knoten[3], Knoten[2])       % TeX label: "$q_3$"
    Knoten[3]c{dir 210} .. Knoten[2]c);
label bot(btex ... etex, cuta(Knoten[4], Knoten[3])       % TeX label: "$q_r$"
    Knoten[4]c{dir 210} .. Knoten[3]c);

endfig;
```



```
beginfig(8);

circleit Knoten[0](btex ... etex);      % TeX label: "$0$"
circleit Knoten[1](btex ... etex);      % TeX label: "$1$"
circleit Knoten[2](btex ... etex);      % TeX label: "$2$"
circleit Knoten[3](btex ... etex);      % TeX label: "$% \cdots$"
circleit Knoten[4](btex ... etex);  Knoten[4]dx = Knoten[4]dy;      % TeX label: "$r-1$"
circleit Knoten[5](btex ... etex);      % TeX label: "$r$"

for i = 0 upto 5: Knoten[i]c = origin + i * (1.5cm, 0);  endfor;

drawboxed(Knoten[0], Knoten[1], Knoten[2], Knoten[4], Knoten[5]);
drawunboxed(Knoten[3]);

label lft(btex ... etex, loop(Knoten[0])(-20pt, 0));      % TeX label: "$1$"
label rt(btex ... etex, loop(Knoten[5])(20pt, 0));      % TeX label: "$1$"

for i = 1 upto 4:
    label top(btex ... etex, cuta(Knoten[i], Knoten[i + 1])      % TeX label: "$p$"
        Knoten[i]c{dir 30} .. Knoten[i + 1]c);
    label bot(btex ... etex, cuta(Knoten[i], Knoten[i - 1])      % TeX label: "$q$"
        Knoten[i]c{dir 210} .. Knoten[i - 1]c);
endfor;

endfig;
```



beginfig(9);

circleit *Knoten*[0](**btex** ... **etex**); % TEX label: `"$0$"`
circleit *Knoten*[1](**btex** ... **etex**); % TEX label: `"$1$"`
circleit *Knoten*[2](**btex** ... **etex**); % TEX label: `"$2$"`
circleit *Knoten*[3](**btex** ... **etex**); % TEX label: `"$% \cdots$"`
circleit *Knoten*[4](**btex** ... **etex**); % TEX label: `"$r$"`

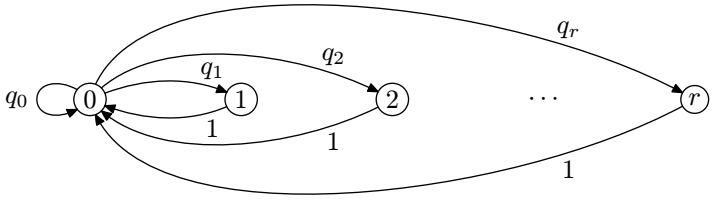
for $i = 0$ **upto** 4: *Knoten*[*i*]*c* = *origin* + *i* ∗ (2*cm*, 0); **endfor**;

drawboxed(*Knoten*[0], *Knoten*[1], *Knoten*[2], *Knoten*[4]);
drawunboxed(*Knoten*[3]);

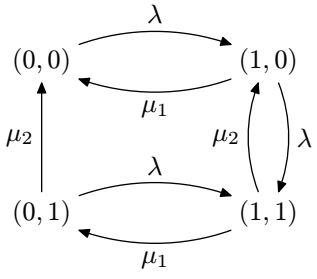
label *lft*(**btex** ... **etex**, *loop*(*Knoten*[0])(−20*pt*, 0)); % TEX label: `"$q_0$"`

label *top*(**btex** ... **etex**, *cutb*(*Knoten*[0], *Knoten*[1]) % TEX label: `"$q_1$"`
Knoten[0]*c*{dir 25} .. {dir −30}*Knoten*[1]*c*);
label *top*(**btex** ... **etex**, *cutb*(*Knoten*[0], *Knoten*[2]) % TEX label: `"$q_2$"`
Knoten[0]*c*{dir 50} .. tension 1.2 .. {dir −30}*Knoten*[2]*c*);
label *top*(**btex** ... **etex**, *cutb*(*Knoten*[0], *Knoten*[4]) % TEX label: `"$q_r$"`
Knoten[0]*c*{dir 75} .. tension 1.44 .. {dir −30}*Knoten*[4]*c*);

label *bot*(**btex** ... **etex**, *cutc*(*Knoten*[4], *Knoten*[0]) % TEX label: `"$1$"`
Knoten[4]*c*{dir 210} .. tension 1.44 .. {dir 105}*Knoten*[0]*c*);
label *bot*(**btex** ... **etex**, *cutc*(*Knoten*[2], *Knoten*[0]) % TEX label: `"$1$"`
Knoten[2]*c*{dir 210} .. tension 1.2 .. {dir 130}*Knoten*[0]*c*);
label *bot*(**btex** ... **etex**, *cutc*(*Knoten*[1], *Knoten*[0]) % TEX label: `"$1$"`
Knoten[1]*c*{dir 210} .. {dir 155}*Knoten*[0]*c*);
endfig;



```
beginfig(10);

boxit Knoten[0](btex ... etex);      % TeX label: "$(0,0)$"
boxit Knoten[1](btex ... etex);      % TeX label: "$(1,0)$"
boxit Knoten[2](btex ... etex);      % TeX label: "$(0,1)$"
boxit Knoten[3](btex ... etex);      % TeX label: "$(1,1)$"

Knoten[0]c = origin;
Knoten[1]c - Knoten[0]c =
    Knoten[3]c - Knoten[2]c = (3cm, 0);
Knoten[0]c - Knoten[2]c = (0, 2cm);

for i = 0 upto 3: drawunboxed(Knoten[i]); endfor;

label top(btex ... etex, cuta(Knoten[0], Knoten[1])      % TeX label: "$\lambda$"
    Knoten[0]c{dir 30} .. Knoten[1]c);
label bot(btex ... etex, cuta(Knoten[1], Knoten[0])      % TeX label: "$\mu_1$"
    Knoten[1]c{dir 210} .. Knoten[0]c);

label lft(btex ... etex, cuta(Knoten[2], Knoten[0])      % TeX label: "$\mu_2$"
    Knoten[2]c -- Knoten[0]c);

label lft(btex ... etex, cuta(Knoten[3], Knoten[1])      % TeX label: "$\mu_2$"
    Knoten[3]c{dir 120} .. Knoten[1]c);
label rt(btex ... etex, cuta(Knoten[1], Knoten[3])       % TeX label: "$\lambda$"
    Knoten[1]c{dir 300} .. Knoten[3]c);

label bot(btex ... etex, cuta(Knoten[3], Knoten[2])      % TeX label: "$\mu_1$"
    Knoten[3]c{dir 210} .. Knoten[2]c);
label top(btex ... etex, cuta(Knoten[2], Knoten[3])      % TeX label: "$\lambda$"
    Knoten[2]c{dir 30} .. Knoten[3]c);

endfig;
```



beginfig(11);

boxit *Knoten*[0](**btex** ... **etex**); % TEX label: `"$(0,0)$"`
boxit *Knoten*[1](**btex** ... **etex**); % TEX label: `"$(0,1)$"`
boxit *Knoten*[2](**btex** ... **etex**); % TEX label: `"$(b,1)$"`
boxit *Knoten*[3](**btex** ... **etex**); % TEX label: `"$(1,0)$"`
boxit *Knoten*[4](**btex** ... **etex**); % TEX label: `"$(1,1)$"`

Knoten[1]c = *origin*;
Knoten[2]c − *Knoten*[1]c =
    Knoten[4]c − *Knoten*[3]c = (3*cm*, 0);
Knoten[1]c − *Knoten*[3]c = (0, 2*cm*);
Knoten[0]c − *Knoten*[3]c =
    Knoten[1]c − *Knoten*[3]c rotated 60;

for $i = 0$ **upto** 4: **drawunboxed**(*Knoten*[i]); **endfor**;

label *ulft*(**btex** ... **etex**, *cuta*(*Knoten*[1], *Knoten*[0]) % TEX label: `"$\mu_2$"`
    Knoten[1]c -- *Knoten*[0]c);

label *top*(**btex** ... **etex**, *cuta*(*Knoten*[2], *Knoten*[1]) % TEX label: `"$\mu_2$"`
    Knoten[2]c -- *Knoten*[1]c);

label *rt*(**btex** ... **etex**, *cuta*(*Knoten*[3], *Knoten*[1]) % TEX label: `"$\mu_1$"`
    Knoten[3]c -- *Knoten*[1]c);

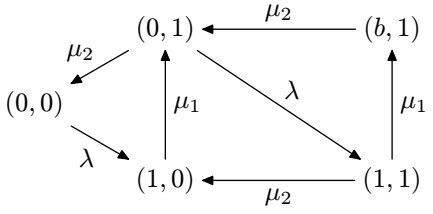
label *urt*(**btex** ... **etex**, *cuta*(*Knoten*[1], *Knoten*[4]) % TEX label: `"$\lambda$"`
    Knoten[1]c -- *Knoten*[4]c);

label *rt*(**btex** ... **etex**, *cuta*(*Knoten*[4], *Knoten*[2]) % TEX label: `"$\mu_1$"`
    Knoten[4]c -- *Knoten*[2]c);

label *llft*(**btex** ... **etex**, *cuta*(*Knoten*[0], *Knoten*[3]) % TEX label: `"$\lambda$"`
    Knoten[0]c -- *Knoten*[3]c);

label *bot*(**btex** ... **etex**, *cuta*(*Knoten*[4], *Knoten*[3]) % TEX label: `"$\mu_2$"`
    Knoten[4]c -- *Knoten*[3]c);

endfig;



```
beginfig(12);

numeric outer_length;  outer_length = 2cm;

pair point[];

point[1] = (0.0, 1.6);  point[2] = (0.6, 0.5);
point[3] = (1.7, 0.8);  point[4] = (1.7, 2.2);
point[5] = (0.5, 2.8);  point[6] = (0.5, 1.9);
point[7] = (1.5, 0.0);  point[8] = (2.9, 0.5);
point[9] = (3.0, 2.8);  point[10] = (1.5, 3.7);

for i = 1 upto 10:
    point[i] := point[i] * outer_length;
endfor;

point[12] = point[7] + (1.3, 0) * outer_length;

point[16] - point[11] = point[9] - point[4];
point[17] - point[12] = point[10] - point[5];
point[17] - point[16] = point[4] - point[5];
point[18] - point[17] = point[5] - point[1];
point[18] - point[13] = point[6] - point[1];
point[19] - point[18] = point[1] - point[2];
point[19] - point[14] = point[7] - point[2];
point[20] - point[15] = point[8] - point[3];
point[20] - point[19] = point[2] - point[3];

circleit Ecke[1](btex ... etex);      % TeX label: " $1$"
circleit Ecke[2](btex ... etex);      % TeX label: " $2$"
circleit Ecke[3](btex ... etex);      % TeX label: " $3$"
circleit Ecke[4](btex ... etex);      % TeX label: " $4$"
circleit Ecke[5](btex ... etex);      % TeX label: " $5$"
circleit Ecke[6](btex ... etex);      % TeX label: " $6$"
circleit Ecke[7](btex ... etex);      % TeX label: " $7$"
circleit Ecke[8](btex ... etex);      % TeX label: " $8$"
circleit Ecke[9](btex ... etex);      % TeX label: " $9$"
circleit Ecke[10](btex ... etex);     % TeX label: "$10$"
circleit Ecke[11](btex ... etex);     % TeX label: "$11$"
circleit Ecke[12](btex ... etex);     % TeX label: "$12$"
circleit Ecke[13](btex ... etex);     % TeX label: "$13$"
circleit Ecke[14](btex ... etex);     % TeX label: "$14$"
circleit Ecke[15](btex ... etex);     % TeX label: "$15$"
circleit Ecke[16](btex ... etex);     % TeX label: "$16$"
circleit Ecke[17](btex ... etex);     % TeX label: "$17$"
circleit Ecke[18](btex ... etex);     % TeX label: "$18$"
circleit Ecke[19](btex ... etex);     % TeX label: "$19$"
circleit Ecke[20](btex ... etex);     % TeX label: "$20$"

for i = 1 upto 20:
    Ecke[i]dx = Ecke[i]dy;  Ecke[i]c = point[i];
endfor;

pickup pencircle scaled 1pt;

draw Ecke[1]c -- Ecke[2]c;  draw Ecke[4]c -- Ecke[5]c;
draw Ecke[14]c -- Ecke[19]c;  draw Ecke[13]c -- Ecke[9]c;
draw Ecke[18]c -- Ecke[17]c;  draw Ecke[3]c -- Ecke[8]c;
draw Ecke[2]c -- Ecke[7]c -- Ecke[12]c;
```

```
draw Ecke[15]c -- Ecke[10]c dashed withdots;
draw Ecke[6]c -- Ecke[11]c dashed withdots;
draw Ecke[11]c -- Ecke[16]c -- Ecke[20]c dashed withdots;

pickup pencircle scaled 4pt;

draw Ecke[1]c -- Ecke[6]c dashed withdots;
draw Ecke[6]c -- Ecke[15]c dashed withdots;
draw Ecke[15]c -- Ecke[20]c dashed withdots;
draw Ecke[20]c -- Ecke[19]c dashed withdots;
draw Ecke[19]c -- Ecke[18]c;  draw Ecke[18]c -- Ecke[13]c;
draw Ecke[13]c -- Ecke[8]c;  draw Ecke[8]c -- Ecke[12]c;
draw Ecke[12]c -- Ecke[17]c;
draw Ecke[17]c -- Ecke[16]c dashed withdots;
draw Ecke[16]c -- Ecke[11]c dashed withdots;
draw Ecke[11]c -- Ecke[7]c dashed withdots;
draw Ecke[7]c -- Ecke[2]c;  draw Ecke[2]c -- Ecke[3]c;
draw Ecke[3]c -- Ecke[4]c;  draw Ecke[4]c -- Ecke[9]c;
draw Ecke[9]c -- Ecke[14]c;  draw Ecke[14]c -- Ecke[10]c;
draw Ecke[10]c -- Ecke[5]c;  draw Ecke[5]c -- Ecke[1]c;

pickup pencircle scaled .4pt;

for i = 20 downto 1:
    unfill bpath Ecke[i];  drawboxed(Ecke[i]);
endfor;

endfig;

end
```