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`.

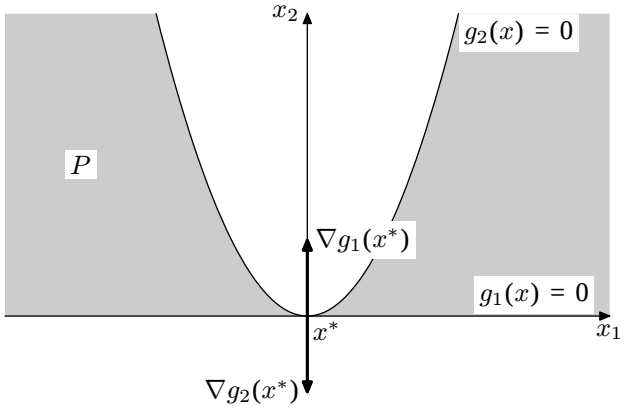
Including MetaPost graphics in LATEX documents is straightforward with the `graphicx` bundle and the latest `pdftex.def` by David Carlisle, Sebastian Rahtz, and Hans Hagen.

```
beginfig(1); % Start the first graphic.

% Declare a bunch of variables; points ...
pair   cw, nghwtcpig, tki  vtcpig, d ww o tcpig, v rtcpig;
pair v rnghw, v rtki  v, d ww o nghw, d ww o tki  v;

path tgwatkevk p_vy  , kptcpig; % paths ...

picture ewvrcdgn; % and an image.

% Set the coordinates;  tkikp equals (0, 0).
 cw =  tkikp; nghwtcpig = (−4eo , 0); tki  vtcpig = (4eo , 0);
d ww o tcpig = (0, −1eo ); v rtcpig = (0, 4eo );
v rnghw = (−4eo , 4eo ); d ww o nghw = (−4eo , 0);
v rtki  v = (4eo , 4eo ); d ww o tki  v = (4eo , 0);

% MetaPost does not (yet) provide the possibility for plotting functions,
% so we have to use intermediate points for the parabola.
tgwatkevk p_vy  = (−2eo , 4eo )
    for i = −8 upto 8: .. (i/4, (i/4) ∗ (i/4)) ∗ 1eo  endfor;

% buildcycle creates a closed path from several (sub)paths.
kptcpig = buildcycle(d ww o nghw -- v rnghw -- (−2eo , 4eo ),
    tgwatkevk p_vy  , (2eo , 4eo ) -- v rtki  v -- d ww o tki  v -- d ww o nghw);

% 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 kptcpig withcolor .8y  kxg;

% Introduce the x and y axis.
drawarrow nghwtcpig -- tki  vtcpig;
drawarrow d ww o tcpig -- v rtcpig;

draw tgwatkevk p_vy  ; % Draw the parabola.

% Mark various points of interest. This is done by TEX itself.
label ntv(btex ... etex,  cw);      % TEX label: "$x^% \ast$"
label d v(btex ... etex, tki  vtcpig);      % TEX label: "$x_1$"
label nw(btex ... etex, v rtcpig);      % TEX label: "$x_2$"

ewvrcdgn:= thelabel tv(
    btex ... etex,  cw + (0, 1eo ));      % TEX label: "$\nabla g_1(x^\ast)$"
unfill bbox ewvrcdgn; draw ewvrcdgn;

label nw(btex ... etex,  cw − (0, 1eo ));      % TEX label: "$\nabla g_2(x^\ast)$"
```

```
ewrcdgn:= thelabel vr(btex ... etex, (3eo , 0));      % TeX label: "$g_1(x)=0$"
unfill bbox ewrcdgn; draw ewrcdgn;

ewrcdgn:= thelabel rtv(btex ... etex, (2eo , 4eo ));      % TeX label: "$g_2(x)=0$"
unfill bbox ewrcdgn; draw ewrcdgn;

ewrcdgn:= thelabel(btex ... etex, (−3eo , 2eo ));      % TeX label: "$P$"
unfill bbox ewrcdgn; draw ewrcdgn;

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

drawarrow  cw -- ( cw + (0, 1eo ));
drawarrow  cw -- ( cw − (0, 1eo ));

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}`.

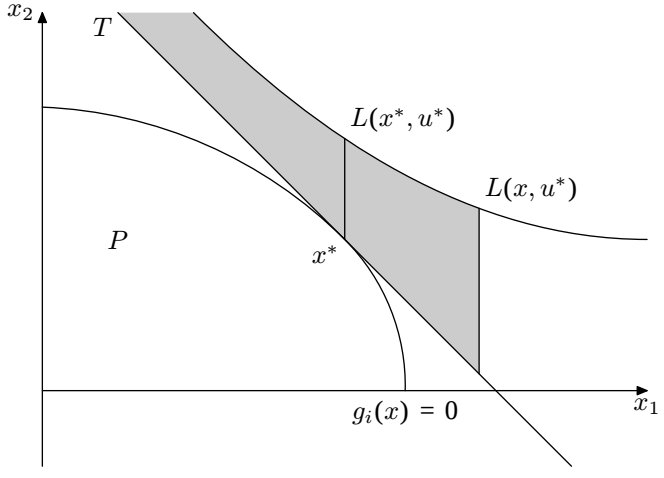
LaTeX users should be familiar with what they have to do. Note that you *o ww* rename the files written by MetaPost to something like `nlpgraph1.mps` for `pdftex.def`'s sake.



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 stu that should be visible.

```
beginfig(2);

pair tki_utcpig, nghtutcpig, v_rtcpig, d_w_o_tcpig;

tki_utcpig = (8eo, 0); nghtutcpig = tki_kp;
v_rtcpig = (0, 5eo); d_w_o_tcpig = (0, −1eo);

label d_v(btex ... etex, tki_utcpig);      % TeX label: "$x_1$"
label nhv(btex ... etex, v_rtcpig);        % TeX label: "$x_2$"

pair zcw; zcw = (4eo, 2eo);
label nhv(btex ... etex, zcw);             % TeX label: "$x^% \ast$"

path T; T = (1eo, 5eo) -- (7eo, −1eo);
label nhv(btex ... etex, point 0 of T);    % TeX label: "$T$"

label(btex ... etex, (1eo, 2eo));          % TeX label: "$P$"

path i_k; i_k = .75v_rtcpig .. {(1, −1)}zcw .. .6tki_utcpig;

label d_v(btex ... etex, .6tki_utcpig);    % TeX label: "$g_i(x)=0$"

path L; L = (2eo, 5eo){(1, −1)} .. {tki_v}(8eo, 2eo);

pair R, R, R;

R = L intersectionpoint ((xpart(zcw), −kpfipkv)
    -- (xpart(zcw), kpfipkv));

R = point .666667 of L;

label utv(btex ... etex, R);               % TeX label: "$L(x^% \ast,u^\ast)$"
label utv(btex ... etex, R);               % TeX label: "$L(x,u^\ast)$"

R = T intersectionpoint (R -- (xpart(R), −kpfipkv));

path M_pxgz;
M_pxgz = buildcycle(L, R -- R -- point 0 of T -- point 0 of L);

fill M_pxgz withcolor .8y_kng;

for i = T, i_k, L: draw i; endfor;

draw zcw -- R; draw R -- R;

drawarrow nghtutcpig -- tki_utcpig;
drawarrow d_w_o_tcpig -- v_rtcpig;

endfig; % 2
```



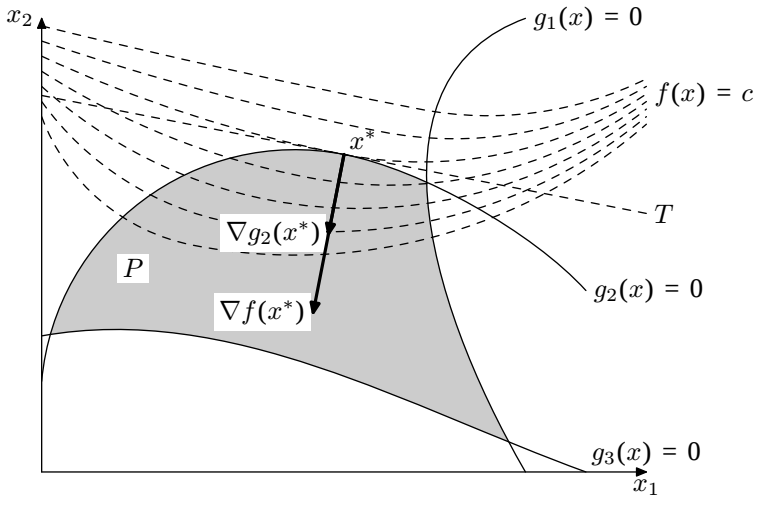
```
beginfig(3);

pair tki utcpig, v rtcpig;
tki utcpig = (8eo , 0); v rtcpig = (0, 6eo );

for i = tki utcpig, v rtcpig: drawarrow tkikp -- i; endfor;

label d v(btex ... etex, tki utcpig);      % TEX label: "$x_1$"
label nw(btex ... etex, v rtcpig);      % TEX label: "$x_2$"

pair zcww; zcw = (.5 xpart(tki utcpig), .7 ypart(v rtcpig));
label utv(btex ... etex, zcw);      % TEX label: "$x^% \ast$"

path g[];

g[1] = (.8 xpart(tki utcpig), ypart(v rtcpig)){dir 200}
    .. {dir 300}.8tki utcpig;
g[2] = (.9 xpart(tki utcpig), .4 ypart(v rtcpig))
    .. tension 1.2 .. zcw .. (0, .2 ypart(v rtcpig));
g[3] = .9tki utcpig{dir 160} .. {dir 190}.3v rtcpig;

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

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

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

numeric zcww;

( zcww, y cvgxgt) = g[2]intersectiontimes
    ((xpart(zcww), −kpfipkv ) -- (xpart(zcww), kpfipkv ));

pair A, B, C;

A = − direction zcww of g[2];
B = zcww + y cvgxgt ∗ A; xpart(B) = xpart(tki utcpig);
C = zcww + y cvgxgt ∗ A; xpart(C) = 0;

draw C -- B dashed gxgpn ; label tv(btex ... etex, B);      % TEX label: "$T$"

pair ghw[], kffrg[], Tki v[];

for i = −4 upto 2:
      ghw[i] = (0eo , 5.5eo + .2i ∗ eo );
      kffrg[i] = zcww + i ∗ (.5eo , .3eo );
    Tki v[i] = (xpart(tki utcpig), 5eo + .1i ∗ eo );
    draw ghw[i] .. kffrg[i]{A} .. Tki v[i] dashed gxgpn ;
endfor;

label tv(btex ... etex, Tki v[0]);      % TEX label: "$f(x)=c$"

pickup pencircle scaled 1.2rv;

pair pcdrch, pcdrci ;

pcdrch = zcww + (.7A rotated −90);
pcdrci = zcww + (.35A rotated −90);

picture ewwrcdgn;

ewwrcdgn := thelabel nw(btex ... etex, pcdrch);      % TEX label: "$\nabla f(x^\ast)$"
unfill bbox ewwrcdgn; draw ewwrcdgn;
```

ewnrcdgn := **thelabel** *rhv*(**btex** ... **etex**, *pcdrci*); % TEX label: "$\nabla g_2(x^\ast)$"
unfill bbox *ewnrcdgn*; **draw** *ewnrcdgn*;

drawarrow *zcw* -- *pcdrch*; **drawarrow** *zcw* -- *pcdrci*;

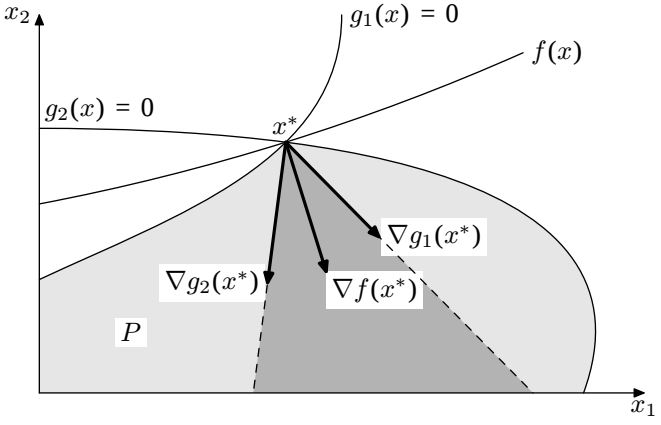
ewnrcdgn := **thelabel** *wtv*(**btex** ... **etex**, (1*eo* , 2.5*eo*)); % TEX label: "P"
unfill bbox *ewnrcdgn*; **draw** *ewnrcdgn*;

endfig; % 3

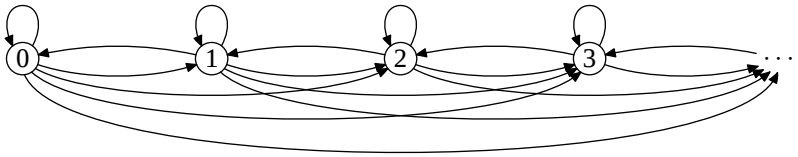


```
beginfig(4);

pair tkiutcpig, v rtcpig;
tkiutcpig = (8eo , 0); v rtcpig = (0, 5eo );

label d v(btex ... etex, tkiutcpig);      % TEX label: "$x_1$"
label rtv(btex ... etex, v rtcpig);       % TEX label: "$x_2$"

path g[], f;

g[1] = .3v rtcpig{dir 25} ..
    {ur}(.5 xpart(tkiutcpig), ypart(v rtcpig));
g[2] = .7v rtcpig{tki v} .. {dir 250}.9tkiutcpig;

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

pair zcw; zcw = g[1] intersectionpoint g[2];

label v r(btex ... etex, zcw);      % TEX label: "$x^% \ast$"

f = .5v rtcpig .. zcw .. (.8 xpart(tkiutcpig), .9 ypart(v rtcpig));

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

numeric zcw[];
pair A[], B[], C[];

( zcw[1], y cvgxgt) = g[1]intersectiontimes
    ((xpart(zcw), −kpfipkv ) -- (xpart(zcw), kpfipkv ));

A[1] = − direction zcw[1] of g[1];
B[1] = zcw + (A[1] rotated 90);
C[1] = zcw + y cvgxgt ∗ (A[1] rotated 90); ypart(C[1]) = 0;

( zcw[2], y cvgxgt) = g[2]intersectiontimes
    ((xpart(zcw), −kpfipkv ) -- (xpart(zcw), kpfipkv ));

A[2] = − direction zcw[2] of g[2];
B[2] = zcw + .5(A[2] rotated 90);
C[2] = zcw + y cvgxgt ∗ (A[2] rotated 90); ypart(C[2]) = 0;

( zcw[3], y cvgxgt) = fintersectiontimes
    ((xpart(zcw), −kpfipkv ) -- (xpart(zcw), kpfipkv ));

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

path P;

P = buildcycle( tkikp -- C[1] -- zcw, g[1], tkikp -- v rtcpig);
fill P withcolor .9y kvg;

P := buildcycle(C[2] -- tkiutcpig, g[2], zcw -- C[2]);
fill P withcolor .9y kvg;

fill zcw -- C[1] -- C[2] -- cycle withcolor .7y kvg;

picture ewvrcdgn;

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

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

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

ewvrcdgn := thelabel rtv(btex ... etex, (1eo , 1eo ));      % TEX label: "$P$"
unfill bbox ewvrcdgn; draw ewvrcdgn;

for i = 1 upto 2: draw zcw -- C[i] dashed gxgpn ; endfor;
for i = tki vtcpig, v rtcpig: drawarrow tkikp -- i; endfor;
for i = 1 upto 2: draw g[i]; endfor;

draw f;

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

endfig; % 4
```

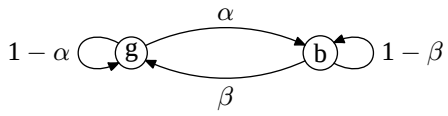


```
input boxes; % TEX labels are best handled as bordered objects

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

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

vardef ewve(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 n r(suffix a) expr p =
    ewvc(a, a)a_c{curl 0} .. a_c + p .. {curl 0}a_c enddef;

beginfig(5);

% Define some circular objects with TEX labels.
circleit Mp vgp[0](btex ... etex);      % TEX label: "$0$"
circleit Mp vgp[1](btex ... etex);      % TEX label: "$1$"
circleit Mp vgp[2](btex ... etex);      % TEX label: "$2$"
circleit Mp vgp[3](btex ... etex);      % TEX label: "$3$"
circleit Mp vgp[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:
    Mp vgp[i]c = tkikp + i*(2.5eo , 0); drawboxed(Mp vgp[i]);
    label(btex ... etex, n r(Mp vgp[i])(0, 20rv));      % TEX label: "\relax"
endfor;

Mp vgp[4]c = tkikp + 4*(2.5eo , 0);
Mp vgp[4]fz = Mp vgp[4]f ; drawunboxed(Mp vgp[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, ewvc(Mp vgp[i], Mp vgp[j])      % TEX label: "\relax"
            Mp vgp[i]c{dir((i - j)*25)}
                .. tension((j - i)*1.2) .. Mp vgp[j]c);
    endfor;
endfor;

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

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

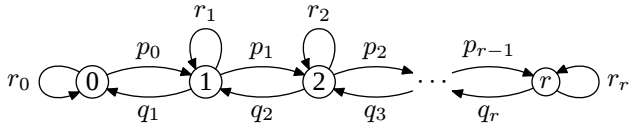


The material in this MetaPost source file is quite elementary. First a set of nodes (Knoten) are defined and placed, then the arrows are drawn and labelled.

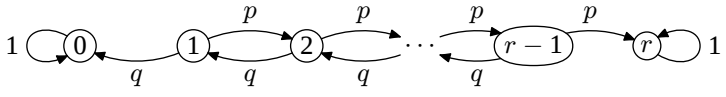
```
beginfig(6);

circleit Mp vgp[0](btex ... etex); Mp vgp[0]c = tkikp;       % TeX label: "g"
circleit Mp vgp[1](btex ... etex); Mp vgp[1]c = tkikp + (2.5eo , 0);       % TeX label: "b"

drawboxed(Mp vgp[0], Mp vgp[1]);

label lftw(btex ... etex, n r(Mp vgp[0])(−20rv, 0));       % TeX label: "$1-\alpha$"
label tv(btex ... etex, n r(Mp vgp[1])(20rv, 0));       % TeX label: "$1-\beta$"

label v r(btex ... etex, ewxc(Mp vgp[0], Mp vgp[1])       % TeX label: "$\alpha$"
    Mp vgp[0]c{dir 30} .. Mp vgp[1]c);
label d v(btex ... etex, ewxc(Mp vgp[1], Mp vgp[0])       % TeX label: "$\beta$"
    Mp vgp[1]c{dir 210} .. Mp vgp[0]c);

endfig;
```



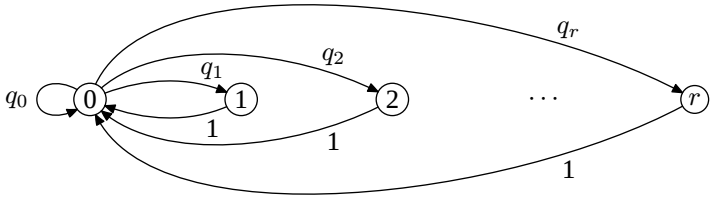
```
beginfig(7);

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

for i = 0 upto 4: Mp vgp[i]c = tkikp + i*(1.5eo , 0); endfor;

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

label mlv(btex ... etex, n r(Mp vgp[0])(-20rv, 0));      % TeX label: "$r_0$"
label tv(btex ... etex, n r(Mp vgp[4])(20rv, 0));        % TeX label: "$r_r$"
label v r(btex ... etex, n r(Mp vgp[1])(0, 20rv));       % TeX label: "$r_1$"
label v r(btex ... etex, n r(Mp vgp[2])(0, 20rv));       % TeX label: "$r_2$"

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

label d v(btex ... etex, ewvc(Mp vgp[1], Mp vgp[0])      % TeX label: "$q_1$"
    Mp vgp[1]c{dir 210} .. Mp vgp[0]c);
label d v(btex ... etex, ewvc(Mp vgp[2], Mp vgp[1])      % TeX label: "$q_2$"
    Mp vgp[2]c{dir 210} .. Mp vgp[1]c);
label d v(btex ... etex, ewvc(Mp vgp[3], Mp vgp[2])      % TeX label: "$q_3$"
    Mp vgp[3]c{dir 210} .. Mp vgp[2]c);
label d v(btex ... etex, ewvc(Mp vgp[4], Mp vgp[3])      % TeX label: "$q_r$"
    Mp vgp[4]c{dir 210} .. Mp vgp[3]c);

endfig;
```



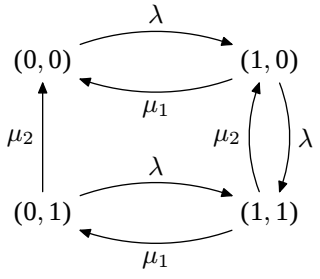
```
beginfig(8);

circleit Mp vgp[0](btex ... etex);       % TEX label: "$0$"
circleit Mp vgp[1](btex ... etex);       % TEX label: "$1$"
circleit Mp vgp[2](btex ... etex);       % TEX label: "$2$"
circleit Mp vgp[3](btex ... etex);       % TEX label: "$% \cdots$"
circleit Mp vgp[4](btex ... etex); Mp vgp[4]fz = Mp vgp[4]f ;       % TEX label: "$r-1$"
circleit Mp vgp[5](btex ... etex);       % TEX label: "$r$"

for i = 0 upto 5: Mp vgp[i]c = tkikp + i * (1.5eo , 0); endfor;

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

label mw(btex ... etex, n r(Mp vgp[0])(−20rv, 0));       % TEX label: "$1$"
label tv(btex ... etex, n r(Mp vgp[5])(20rv, 0));       % TEX label: "$1$"

for i = 1 upto 4:
    label v r(btex ... etex, ewvc(Mp vgp[i], Mp vgp[i + 1])       % TEX label: "$p$"
        Mp vgp[i]c{dir 30} .. Mp vgp[i + 1]c);
    label d v(btex ... etex, ewvc(Mp vgp[i], Mp vgp[i − 1])       % TEX label: "$q$"
        Mp vgp[i]c{dir 210} .. Mp vgp[i − 1]c);
endfor;

endfig;
```



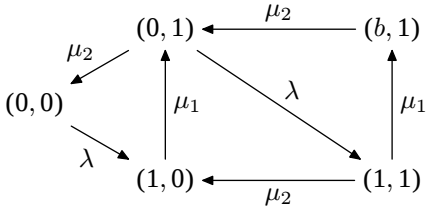
```
beginfig(9);

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

for i = 0 upto 4: Mp vgp[i]c = tkikp + i * (2eo , 0); endfor;

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

label mlw(btex ... etex, n r(Mp vgp[0])(−20rv, 0));       % TEX label: "$q_0$"

label v r(btex ... etex, ewwd(Mp vgp[0], Mp vgp[1])       % TEX label: "$q_1$"
    Mp vgp[0]c{dir 25} .. {dir −30}Mp vgp[1]c);
label v r(btex ... etex, ewwd(Mp vgp[0], Mp vgp[2])       % TEX label: "$q_2$"
    Mp vgp[0]c{dir 50} .. tension 1.2 .. {dir −30}Mp vgp[2]c);
label v r(btex ... etex, ewwd(Mp vgp[0], Mp vgp[4])       % TEX label: "$q_r$"
    Mp vgp[0]c{dir 75} .. tension 1.44 .. {dir −30}Mp vgp[4]c);

label d v(btex ... etex, ewwe(Mp vgp[4], Mp vgp[0])       % TEX label: "$1$"
    Mp vgp[4]c{dir 210} .. tension 1.44 .. {dir 105}Mp vgp[0]c);
label d v(btex ... etex, ewwe(Mp vgp[2], Mp vgp[0])       % TEX label: "$1$"
    Mp vgp[2]c{dir 210} .. tension 1.2 .. {dir 130}Mp vgp[0]c);
label d v(btex ... etex, ewwe(Mp vgp[1], Mp vgp[0])       % TEX label: "$1$"
    Mp vgp[1]c{dir 210} .. {dir 155}Mp vgp[0]c);
endfig;
```



```
beginfig(10);

boxit Mp vgp[0](btex ... etex);      % TEX label: "$(0,0)$"
boxit Mp vgp[1](btex ... etex);      % TEX label: "$(1,0)$"
boxit Mp vgp[2](btex ... etex);      % TEX label: "$(0,1)$"
boxit Mp vgp[3](btex ... etex);      % TEX label: "$(1,1)$"

Mp vgp[0]c = tkikp;
Mp vgp[1]c - Mp vgp[0]c =
    Mp vgp[3]c - Mp vgp[2]c = (3eo , 0);
Mp vgp[0]c - Mp vgp[2]c = (0, 2eo );

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

label v r(btex ... etex, ewxc(Mp vgp[0], Mp vgp[1])      % TEX label: "$\lambda$"
    Mp vgp[0]c{dir 30} .. Mp vgp[1]c);
label d v(btex ... etex, ewxc(Mp vgp[1], Mp vgp[0])      % TEX label: "$\mu_1$"
    Mp vgp[1]c{dir 210} .. Mp vgp[0]c);

label rhw(btex ... etex, ewxc(Mp vgp[2], Mp vgp[0])      % TEX label: "$\mu_2$"
    Mp vgp[2]c -- Mp vgp[0]c);

label rhw(btex ... etex, ewxc(Mp vgp[3], Mp vgp[1])      % TEX label: "$\mu_2$"
    Mp vgp[3]c{dir 120} .. Mp vgp[1]c);
label tv(btex ... etex, ewxc(Mp vgp[1], Mp vgp[3])       % TEX label: "$\lambda$"
    Mp vgp[1]c{dir 300} .. Mp vgp[3]c);

label d v(btex ... etex, ewxc(Mp vgp[3], Mp vgp[2])      % TEX label: "$\mu_1$"
    Mp vgp[3]c{dir 210} .. Mp vgp[2]c);
label v r(btex ... etex, ewxc(Mp vgp[2], Mp vgp[3])      % TEX label: "$\lambda$"
    Mp vgp[2]c{dir 30} .. Mp vgp[3]c);

endfig;
```



```
beginfig(11);

boxit.Mp_vgp[0](btex ... etex);      % TeX label: "$(0,0)$"
boxit.Mp_vgp[1](btex ... etex);      % TeX label: "$(0,1)$"
boxit.Mp_vgp[2](btex ... etex);      % TeX label: "$(b,1)$"
boxit.Mp_vgp[3](btex ... etex);      % TeX label: "$(1,0)$"
boxit.Mp_vgp[4](btex ... etex);      % TeX label: "$(1,1)$"

Mp_vgp[1]c = tkikp;
Mp_vgp[2]c - Mp_vgp[1]c =
    Mp_vgp[4]c - Mp_vgp[3]c = (3eo, 0);
Mp_vgp[1]c - Mp_vgp[3]c = (0, 2eo);
Mp_vgp[0]c - Mp_vgp[3]c =
    Mp_vgp[1]c - Mp_vgp[3]c rotated 60;

for i = 0 upto 4: drawunboxed(Mp_vgp[i]); endfor;

label.whv(btex ... etex, ewvc(Mp_vgp[1], Mp_vgp[0]))      % TeX label: "$\mu_2$"
    Mp_vgp[1]c -- Mp_vgp[0]c);

label.v r(btex ... etex, ewvc(Mp_vgp[2], Mp_vgp[1]))      % TeX label: "$\mu_2$"
    Mp_vgp[2]c -- Mp_vgp[1]c);

label.tv(btex ... etex, ewvc(Mp_vgp[3], Mp_vgp[1]))       % TeX label: "$\mu_1$"
    Mp_vgp[3]c -- Mp_vgp[1]c);

label.utv(btex ... etex, ewvc(Mp_vgp[1], Mp_vgp[4]))      % TeX label: "$\lambda$"
    Mp_vgp[1]c -- Mp_vgp[4]c);

label.tv(btex ... etex, ewvc(Mp_vgp[4], Mp_vgp[2]))       % TeX label: "$\mu_1$"
    Mp_vgp[4]c -- Mp_vgp[2]c);

label.nhv(btex ... etex, ewvc(Mp_vgp[0], Mp_vgp[3]))      % TeX label: "$\lambda$"
    Mp_vgp[0]c -- Mp_vgp[3]c);

label.d v(btex ... etex, ewvc(Mp_vgp[4], Mp_vgp[3]))      % TeX label: "$\mu_2$"
    Mp_vgp[4]c -- Mp_vgp[3]c);

endfig;
```



```
beginfig(12);

numeric wxgt_rgpiv ; wxgt_rgpiv = 2eo ;

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] * wxgt_rgpiv ;
endfor;

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

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

for i = 1 upto 20:
    eng[i]fz = eng[i]f ; eng[i]c = point[i];
endfor;

pickup pencircle scaled 1rv;

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

```
draw  eng[15]c --  eng[10]c dashed y kv f  vu;
draw  eng[6]c --  eng[11]c dashed y kv f  vu;
draw  eng[11]c --  eng[16]c --  eng[20]c dashed y kv f  vu;

pickup pencircle scaled 4rv;

draw  eng[1]c --  eng[6]c dashed y kv f  vu;
draw  eng[6]c --  eng[15]c dashed y kv f  vu;
draw  eng[15]c --  eng[20]c dashed y kv f  vu;
draw  eng[20]c --  eng[19]c dashed y kv f  vu;
draw  eng[19]c --  eng[18]c; draw  eng[18]c --  eng[13]c;
draw  eng[13]c --  eng[8]c; draw  eng[8]c --  eng[12]c;
draw  eng[12]c --  eng[17]c;
draw  eng[17]c --  eng[16]c dashed y kv f  vu;
draw  eng[16]c --  eng[11]c dashed y kv f  vu;
draw  eng[11]c --  eng[7]c dashed y kv f  vu;
draw  eng[7]c --  eng[2]c; draw  eng[2]c --  eng[3]c;
draw  eng[3]c --  eng[4]c; draw  eng[4]c --  eng[9]c;
draw  eng[9]c --  eng[14]c; draw  eng[14]c --  eng[10]c;
draw  eng[10]c --  eng[5]c; draw  eng[5]c --  eng[1]c;

pickup pencircle scaled .4rv;

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

endfig;

end
```