

```

def mark_ang(expr a, b, c, rad) =
  begingroup
    path p;
    p = unitvector(a - b){(a - b) rotated 90} .. unitvector(c - b);
    draw p scaled rad shifted b;
  endgroup
enddef;

def mark_angtwice(expr a, b, c, rad) =
  begingroup
    save p;
    path p;
    p = unitvector(a - b){(a - b) rotated 90} .. unitvector(c - b);
    draw p scaled rad shifted b;
    draw p scaled (rad + .1u) shifted b;
  endgroup
enddef;

def mark_rt_angle(expr a, b, c) =
  draw((.5, 0) -- (.5, .5) -- (0, .5))
    zscaled (angle_radius * unitvector(a - b)) shifted b
enddef;

def ang_and_bis(expr a, b, c) =
  begingroup
    save d, co, si;
    pair d;
    c = a zscaled (co, si);
    d := a rotated (.5 * angle(co, si));
    draw b -- d dashed evenly;
    mark_ang(a, b, d, .5u);
    mark_ang(d, b, c, .7u);
  endgroup
enddef;

def ang_and_bistwice(expr a, b, c) =
  begingroup
    save d, co, si;
    pair d;
    c = a zscaled (co, si);
    d := a rotated (.5 * angle(co, si));
    draw b -- d dashed evenly;
    mark_angtwice(a, b, d, .5u);
    mark_angtwice(d, b, c, .6u);
  endgroup
enddef;

def draw_mark(expr p, a) =
  begingroup
    save t, dm; pair dm;
    t = arctime a of p;
    dm = marksize * unitvector direction t of p
      rotated 90;
    draw(-.5dm .. .5dm) shifted point t of p;
  endgroup

```

```

enddef;
def draw_marked(expr p, n) =
  begingroup
  save amid;
  amid = .5 * arclength p;
  for i = -(n - 1)/2 upto (n - 1)/2:
    draw_mark(p, amid + .6marksize * i);
  endfor
  draw p;
  endgroup
enddef;

def markcommon(expr a, b) =
  begingroup
  save s, p;
  pair s;
  path p;
  z1mrkc = .5[a, b];
  s = marksize * unitvector b - a;
  z0mrkc = z1mrkc - s;
  z2mrkc = z1mrkc + s;
  p = z0mrkc{up} .. {down}z1mrkc .. {up}z2mrkc;
  % .5marksize * unitvector direction 0 of p
  endgroup
enddef;

def markpoint(expr fp, n) =
  draw point n of fp -- unitvector direction n of fp rotated 90 scaled .2u shifted point n of fp;
enddef;

def tr(expr a, b) =
  begingroup
  save p;
  path p;
  z1trm = lrcorner a;
  y2trm = ypart urcorner a;
  x2trm = y2trm * cosd(b) / sind(b) + xpart lrcorner a;
  p = z1trm -- z2trm;
  i := 1;
  forever:
    dx := 0.3 * i * u;
    draw p shifted (-dx, 0) withpen pencircle scaled .15pt;
    i := i + 1;
  exitif xpart ulcorner a > x2trm - dx;
  endfor;
  clip currentpicture to a;
  endgroup
enddef;

def grid(expr a, b) =
  begingroup
  save p, q;
  path p, q;
  pickup pencircle scaled .15pt;
  z0 = (0, -.2u);

```

```

z100 = (0, b * u);
p = z0 -- z100 shifted (0, .2u);
for i = 0 upto a: draw p shifted ((u, 0) * i); endfor;
z200 = (-.2u, 0);
z300 = (a * u, 0);
q = z200 -- z300 shifted (.2u, 0);
draw q;
for j = 1 upto b: draw q shifted ((0, u) * j); endfor;
endgroup
enddef;

def parallelogram(suffix d, a, b)(expr e) =
pair c;
c = b shifted e;
d = a shifted e;
draw a -- b -- c -- d -- cycle;
enddef;

```