Asymptote node.asy Example

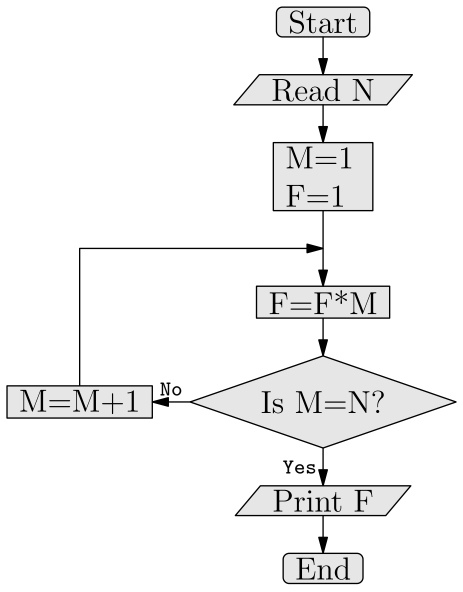
Tao Wei
taowei@buffalo.edu

Contents

1 Syntax

1.1 Basic

```
import node;

pair u=(3.5cm,0);
```



```
pair v=(0,1.2cm);
real dvadjust=1.5;      // diamond vertical adjustment ratio
real sadjust=1.2;                // special adjustment ratio

Arrow=Arrow(6);
pen edgepen=fontsize(8pt)+fontcommand("\ttfamily");
draw_t style1=FillDrawer(lightgray,black);

real xmargin=3pt;
real ymargin=3pt;
real mag=1;

node start=roundbox("Start",(0,0),xmargin,style1,mag);
node read=parallelogram("Read␣N", start.pos-v,xmargin,style1,
   mag);
node b1=box(minipage("M=1\\␣F=1",1cm), read.pos-v, xmargin,
   style1,mag);
node b2=box("F=F*M",b1.pos-sadjust*v,xmargin,style1,mag);
node d1=diamond("Is␣M=N?",b2.pos-dvadjust*v,0,ymargin,style1,
   mag);
node b3=box("M=M+1",d1.pos-u,xmargin,style1,mag);
node print=parallelogram("Print␣F",d1.pos-dvadjust*v,xmargin,
   style1,mag);
node end=roundbox("End",print.pos-v,xmargin,style1,mag);

draw(start,read,print,end, b1,b2,b3,d1);

draw(start--read,edgepen,Arrow);
draw(read--b1,edgepen,Arrow);
draw(b1--b2,edgepen,Arrow);
draw(b2--d1,edgepen,Arrow);
draw("Yes",d1--print,edgepen,Arrow);
draw("No",d1--b3,edgepen,Arrow);
draw(print--end,edgepen,Arrow);
draw(b3--VH--middle(b1,b2),edgepen,Arrow);
```

1.2 Automatically Calcuating Nodes Position and Style Drawing

```
import node;

// define style
defaultnodestyle=nodestyle(xmargin=3pt, ymargin=0, drawfn=
   FillDrawer(lightgray,black));
nodestyle ns2=nodestyle(xmargin=0, ymargin=3pt, drawfn=
   FillDrawer(lightgray,black));
defaultdrawstyle=drawstyle(p=fontsize(8pt)+fontcommand("\
   ttfamily"));

// define node
node start=sroundbox("Start");
node read=sparallelogram("Read␣N");
node b1=sbox(minipage("M=1\\␣F=1",1cm));
node b2=sbox("F=F*M");
node d1=sdiamond("Is␣M=N?",0,ns2);
node b3=sbox("M=M+1");
node print=sparallelogram("Print␣F");
node end=sroundbox("End");

// calc node position
real u=0.5cm;
real v=0.5cm;
start<<reldown(v)<<read<<reldown(v)<<b1<<reldown(1.5*v)<<
        b2<<reldown(v)<<d1<<reldown(v)<<print<<reldown(v)<<end
            ;
d1<<relleft(u)<<b3;

// draw node
draw(start,read,print,end, b1,b2,b3,d1);

// draw edge
sdraw(start--read);
sdraw(read--b1);
sdraw(b1--b2);
sdraw(b2--d1);
sdraw("Yes",d1--print);
sdraw("No",d1--b3);
sdraw(print--end);
sdraw(b3--VH--middle(b1,b2));
```

1.3 Dock Syntax

```
import node;

// define style
defaultnodestyle=nodestyle(xmargin=3pt, ymargin=0, drawfn=
   FillDrawer(lightgray,black));
nodestyle ns2=nodestyle(xmargin=0, ymargin=3pt, drawfn=
   FillDrawer(lightgray,black));
defaultdrawstyle=drawstyle(p=fontsize(8pt)+fontcommand("\
   ttfamily"), arrow=Arrow(6));

// define node
node start=sroundbox("Start");
node read=sparallelogram("Read␣N");
node b1=sbox(minipage("M=1\\␣F=1",1cm));
node b2=sbox("F=F*M");
node d1=sdiamond("Is␣M=N?",0,ns2);
node b3=sbox("M=M+1");
node print=sparallelogram("Print␣F");
node end=sroundbox("End");

// calc node position
real u=0.5cm;
real v=0.5cm;
node c1=vdock(v, centerat=-3,
    start,read,b1,new node,b2,d1,print,end);
hdock(u, centerat=1,
    b3, c1) @ refresh @ deepdraw;

// draw edge
sdraw(start--read);
sdraw(read--b1);
sdraw(b1--b2);
sdraw(b2--d1);
sdraw("Yes",d1--print);
sdraw("No",d1--b3);
sdraw(print--end);
sdraw(b3--VH--middle(b1,b2));
```

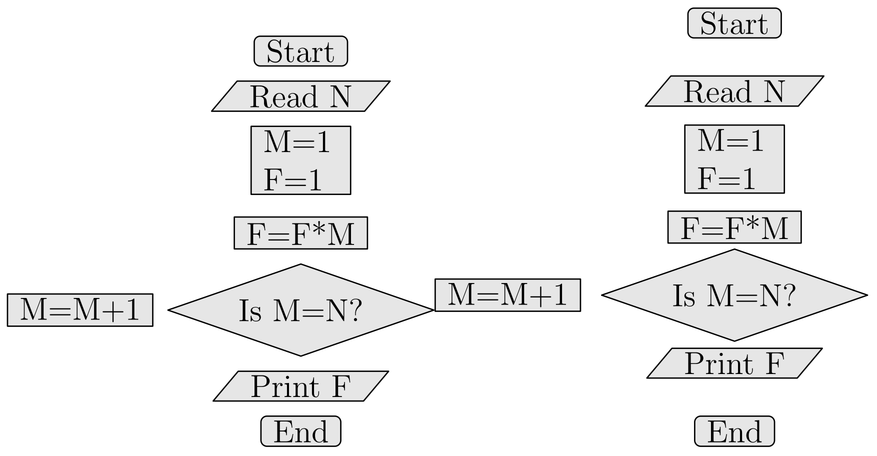
1.4 Edge Struct

```
import node;

// define style
defaultnodestyle=nodestyle(xmargin=3pt, ymargin=0, drawfn=
   FillDrawer(lightgray,black));
nodestyle ns2=nodestyle(xmargin=0, ymargin=3pt, drawfn=
   FillDrawer(lightgray,black));
defaultdrawstyle=drawstyle(p=fontsize(8pt)+fontcommand("\
   ttfamily"), arrow=Arrow(6));

// define node
node start=sroundbox("Start");
node read=sparallelogram("Read␣N");
node b1=sbox(minipage("M=1\\␣F=1",1cm));
node b2=sbox("F=F*M");
node d1=sdiamond("Is␣M=N?",0,ns2);
node b3=sbox("M=M+1");
node print=sparallelogram("Print␣F");
node end=sroundbox("End");

// dock position
real u=0.5cm;
real v=0.5cm;
node c1=vdock(v, centerat=-3,
    start,read,b1,new node,b2,d1,print,end);
hdock(u, centerat=1,
    b3, c1) @ refresh @ deepdraw;

// draw edge
draw(start--read, read--b1, b1--b2, b2--d1, (d1--print).l("Yes
   "),
    (d1--b3).l("No"), print--end, b3--VH--middle(b1,b2));
```

2 Edge Length and Length Between Nodes

2.1 Edge Length

```
import node;

// define style
defaultnodestyle=nodestyle(xmargin=3pt, ymargin=0, drawfn=
   FillDrawer(lightgray,black));
```



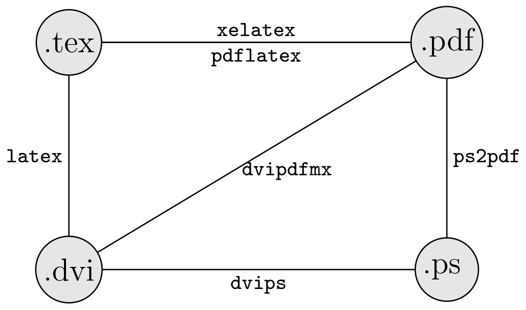
```
nodestyle ns2=nodestyle(xmargin=0, ymargin=3pt, drawfn=
    FillDrawer(lightgray,black));
defaultdrawstyle=drawstyle(p=fontsize(8pt)+fontcommand("\
    ttfamily"));

// define node
node start=sroundbox("Start");
node read=sparallelogram("Read␣N");
node b1=sbox(minipage("M=1\\␣F=1",1cm));
node b2=sbox("F=F*M");
node d1=sdiamond("Is␣M=N?",0,ns2);
node b3=sbox("M=M+1");
node print=sparallelogram("Print␣F");
node end=sroundbox("End");

// calc node position
real u=0.2cm;
real v=0.2cm;
start<<reldown(v)<<read<<reldown(v)<<b1<<reldown(1.5*v)<<
        b2<<reldown(v)<<d1<<reldown(v)<<print<<reldown(v)<<end
            ;
d1<<relleft(u)<<b3;

// draw node
draw(start,read,print,end, b1,b2,b3,d1);
```

2.2 Length Between Nodes

```
import node;

// define style
defaultnodestyle=nodestyle(xmargin=3pt, ymargin=0, drawfn=
    FillDrawer(lightgray,black));
nodestyle ns2=nodestyle(xmargin=0, ymargin=3pt, drawfn=
    FillDrawer(lightgray,black));
defaultdrawstyle=drawstyle(p=fontsize(8pt)+fontcommand("\
    ttfamily"));

// define node
node start=sroundbox("Start");
node read=sparallelogram("Read␣N");
node b1=sbox(minipage("M=1\\␣F=1",1cm));
node b2=sbox("F=F*M");
node d1=sdiamond("Is␣M=N?",0,ns2);
node b3=sbox("M=M+1");
node print=sparallelogram("Print␣F");
node end=sroundbox("End");

// calc node position
real u=3.0cm;
real v=0.9cm;
start<<edown(v)<<read<<edown(v)<<b1<<edown(v)<<
        b2<<edown(v)<<d1<<edown(v)<<print<<edown(v)<<end;
d1<<eleft(u)<<b3;

// draw node
draw(start,read,print,end, b1,b2,b3,d1);
```

3 Functionality

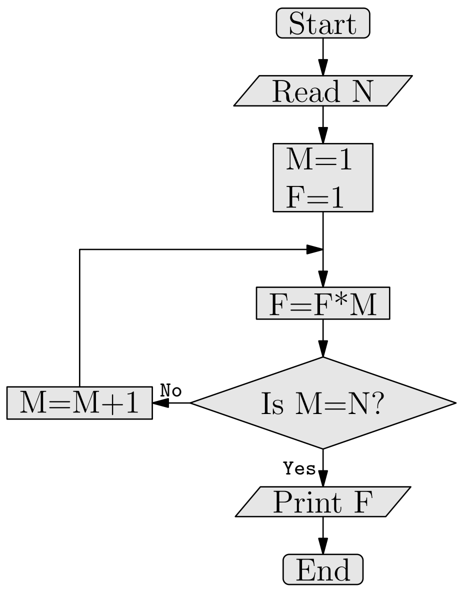
3.1 Graph Illustration



```
import node;

// define style
defaultnodestyle=nodestyle(drawfn=FillDrawer(lightgray,black))
    ;
```

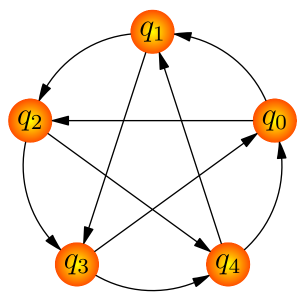
```
defaultdrawstyle=drawstyle(p=fontsize(8pt)+fontcommand("\
   ttfamily"));

// define nodes
node q0=scircle(".tex");
node q1=scircle(".dvi");
node q2=scircle(".pdf");
node q3=scircle(".ps␣");

// calc position
real u=5cm;
real v=3cm;
q0<<edown(v)<<q1<<eright(u)<<q3<<eup(v)<<q2;

// draw nodes
draw(q0,q1,q2,q3);

// draw edges
draw((q0--q1).l("latex"),
    (q0--q2).l("pdflatex"),
    (q0--q2).l("xelatex").style("leftside"),
    (q1--q3).l("dvips"),
    (q3--q2).l("ps2pdf"),
    (q1--q2).l("dvipdfmx"));
```

3.2 Flowchart

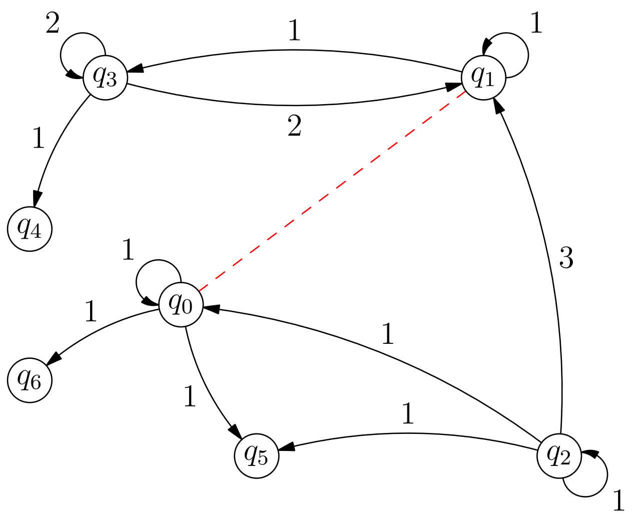


```
import node;

// define style
defaultnodestyle=nodestyle(xmargin=3pt, ymargin=0, drawfn=
    FillDrawer(lightgray,black));
nodestyle ns2=nodestyle(xmargin=0, ymargin=3pt, drawfn=
    FillDrawer(lightgray,black));
defaultdrawstyle=drawstyle(p=fontsize(8pt)+fontcommand("\
    ttfamily"), arrow=Arrow(6));

// define node
node start=sroundbox("Start");
node read=sparallelogram("Read␣N");
node b1=sbox(minipage("M=1\\␣F=1",1cm));
node b2=sbox("F=F*M");
node d1=sdiamond("Is␣M=N?",0,ns2);
node b3=sbox("M=M+1");
node print=sparallelogram("Print␣F");
node end=sroundbox("End");

// dock position
real u=0.5cm;
real v=0.5cm;
node c1=vdock(v, centerat=-3,
     start,read,b1,new node,b2,d1,print,end);
hdock(u, centerat=1,
     b3, c1) @ refresh @ deepdraw;

// draw edge
draw(start--read, read--b1, b1--b2, b2--d1, (d1--print).l("Yes
    "),
     (d1--b3).l("No"), print--end, b3--VH--middle(b1,b2));
```

3.3 Graph Theory



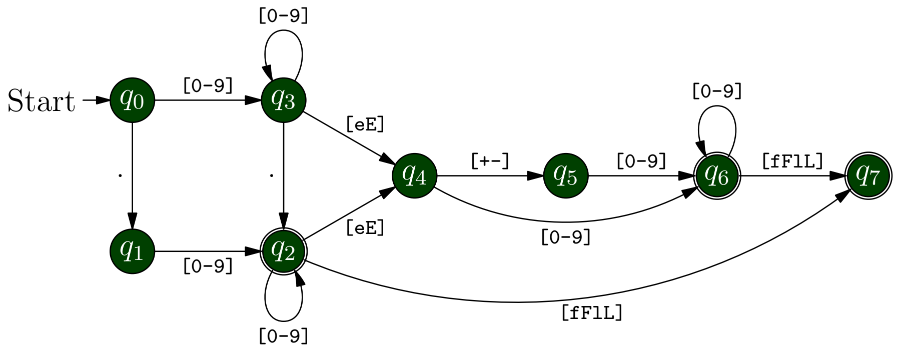
```
import node;

// define style
defaultnodestyle=nodestyle(drawfn=RadialShader(yellow,red));
defaultdrawstyle=drawstyle(p=fontsize(8pt)+fontcommand("\
   ttfamily"),
    arrow=Arrow(6));

// define nodes
node q0=scircle("$q_0$");
node q1=scircle("$q_1$");
node q2=scircle("$q_2$");
node q3=scircle("$q_3$");
node q4=scircle("$q_4$");

// calc node postion and draw
real u=2cm;
real ang0=360/5, ang1=(180-ang0)/2, ang2=90-ang1, ang=180-ang2
   ;
real anginc=360/5;
fancydock(dir(ang), u, anginc, (0.5, 0.5),
    q0, q1, q2, q3, q4) @ refresh @ deepdraw;

// draw edges
draw(q0--q2, q2--q4, q4--q1, q1--q3, q3--q0);
draw(q0..bend..q1, q1..bend..q2, q2..bend..q3,
    q3..bend..q4, q4..bend..q0);
```



```
import nodegraph;

```

```
node[] ver=circles("$q_0$", "$q_1$", "$q_2$", "$q_3$", "$q_4$"
    , "$q_5$", "$q_6$");
pair[] pos={(0,0), (4,3), (5,-2), (-1,3), (-2,1), (1,-2),
    (-2,-1)};
real[][] matadj={{1,1,0,0,0,1,1},
  {0,1,0,1,0,0},
  {1,3,1,0,0,1},
  {0,2,0,2,1,0}};

setpos(ver, pos*1cm);

edge[] edge=genedge(ver, matadj);
edge[edgeind(0,0,matadj)].g=(ver[0]..loop(NW,90,1)).g;
edge[edgeind(0,1,matadj)]=(ver[0]--ver[1]).style(drawstyle(p=
    red+dashed));

draw(edge);
draw(ver);
```

3.4 Automata[1]



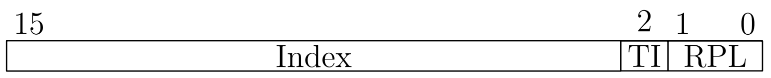
```

import node;

// define node style
defaultnodestyle=nodestyle(textpen=white,
    drawfn=FillDrawer(darkgreen,black));
nodestyle ns2=nodestyle(textpen=white,
    drawfn=Filler(darkgreen)+DoubleDrawer(black));
nodestyle ns3=nodestyle(drawfn=None);
// define edge style
```

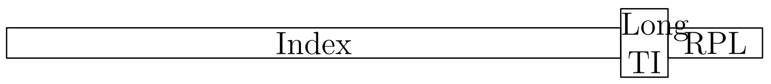
[1]This is from AsymptoteByExample by Leoliu

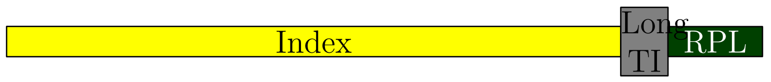
```
defaultdrawstyle=drawstyle(p=fontsize(8pt)+fontcommand("\
   ttfamily"),
    arrow=Arrow(6));

// define nodes
node q0=scircle("$q_0$");
node q1=scircle("$q_1$");
node q2=scircle("$q_2$",ns2);
node q3=scircle("$q_3$");
node q4=scircle("$q_4$");
node q5=scircle("$q_5$");
node q6=scircle("$q_6$",ns2);
node q7=scircle("$q_7$",ns2);
node start=scircle("Start",ns3);

// calculate nodes position
real u=2cm;
start<<eright(0.6u)<<q0<<edown(u)<<q1<<eright(u)<<
q2<<eup(u)<<q3<<edir(-30,u)<<q4<<eright(u)<<
q5<<eright(u)<<q6<<eright(u)<<q7;

// draw nodes
draw(start,q0,q1,q2,q3,q4,q5,q6,q7);

// draw edges
draw(
    (q0--q1).l("."),
    (q1--q2).l("[0-9]"),
    (q3--q2).l("."),
    (q2--q4).l("[eE]"),
    (q0--q3).l("[0-9]").style("leftside"),
    (q3--q4).l("[eE]").style("leftside"),
    (q4--q5).l("[+-]").style("leftside"),
    (q5--q6).l("[0-9]").style("leftside"),
    (q6--q7).l("[fFlL]").style("leftside"),
    (q3..loop(N)).l("[0-9]"),
    (q2..loop(S)).l("[0-9]"),
    (q6..loop(N)).l("[0-9]"),
    (q4..bend..q6).l("[0-9]"),
    (q2..bend..q7).l("[fFlL]"),
    (start--q0));
```

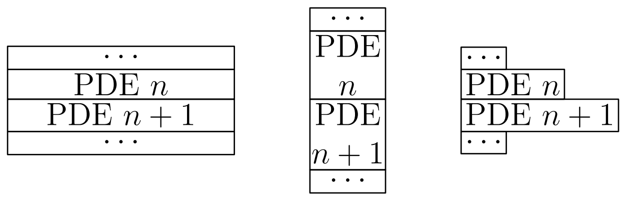
3.5 Boxes[2]

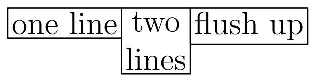
3.5.1 Box Illustration

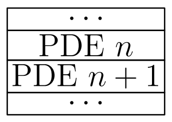


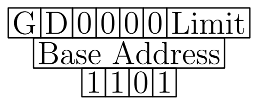


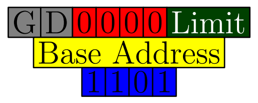












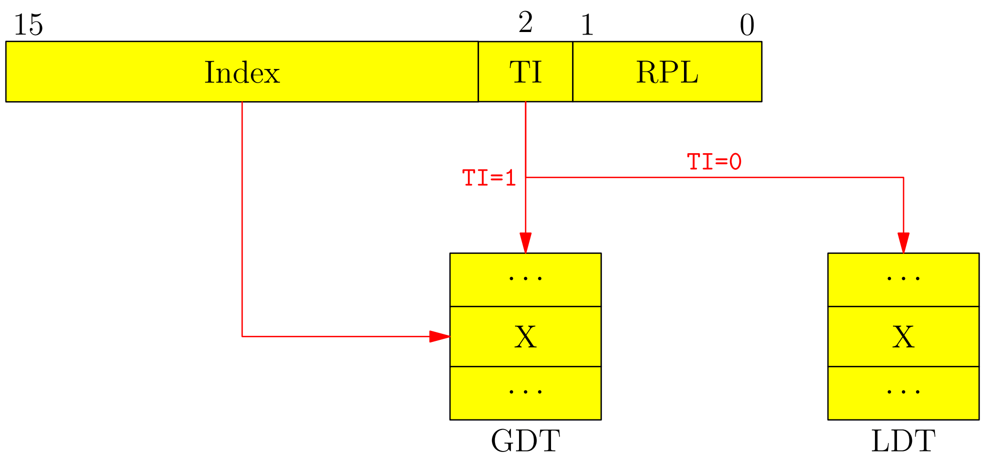
```
import nodebox;

// define styles
nodestyle ns0=defaultnodestyle;
nodestyle ns1=nodestyle(FillDrawer(yellow));
nodestyle ns2=nodestyle(FillDrawer(gray));
nodestyle ns3=nodestyle(white, FillDrawer(darkgreen));
nodestyle ns4=nodestyle(FillDrawer(red));
nodestyle ns5=nodestyle(FillDrawer(blue));
```

[2]This is from BoxesForAsymptote by Addylee2004@163.com

```

// define nodes
node[] a=boxes(10cm, new real[]{13/16, 1/16, 2/16}, "Index", "TI", "RPL");
labelin("15", a[0], NW, NE);
labelin("2", a[1], N);
labelin("1", a[2], NW, NE);
labelin("0", a[2], NE, NW);
node c1=hpack(a[0], a[1], a[2]);
node c2=hbox(10cm, new real[]{13/16, 1/16, 2/16}, "Index", "Long␣TI", "RPL");
node c3=hbox(10cm, new real[]{13/16, 1/16, 2/16}, new nodestyle[]{ns1, ns2, ns3}, "Index", "Long␣TI", "RPL");
node c4=vbox(3cm, "$\cdots$", "PDE␣$n$", "PDE␣$n+1$", "$\cdots$");
node c5=vbox(1cm, "$\cdots$", "PDE␣$n$", "PDE␣$n+1$", "$\cdots$");
node c6=vbox(flush=W, "$\cdots$", "PDE␣$n$", "PDE␣$n+1$", "$\cdots$");
node c7=hbox(flush=N, "one␣line", minipage2("two\par␣lines"), "flush␣up");
node c8=vbox("$\cdots$", "PDE␣$n$", "PDE␣$n+1$", "$\cdots$");
node c9=vpack(
    hbox("G","D","0","0","0","0","Limit"),
    sbox("Base␣Address"),
    hbox("1","1","0","1"));
node c10=vpack(
    hbox(new nodestyle[]{ns2, ns2, ns4, ns4, ns4, ns4, ns3}, "G","D","0","0","0","0","Limit"),
    sbox("Base␣Address",ns1),
    hbox(new nodestyle[]{ns5}, "1","1","0","1"));

// dock
vdock(1cm,
    c1, c2, c3, hdock(1cm, c4, c5, c6), hdock(3cm, c7, c8), c9, c10) @ refresh @ deepdraw;
```

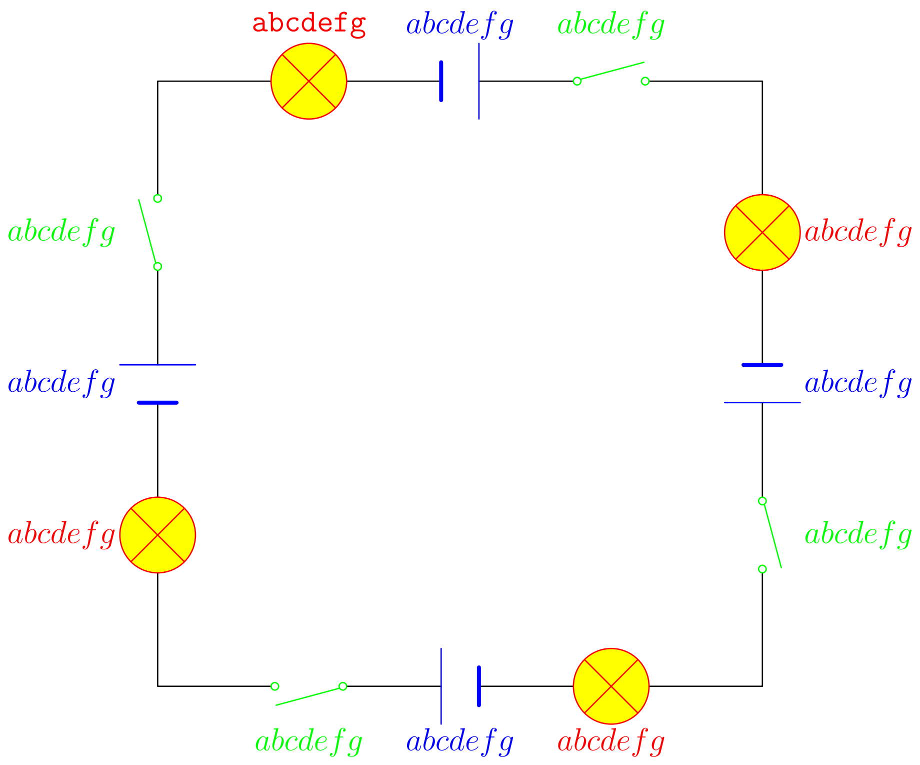
3.5.2 Box Example

```
import nodebox;

// define style
defaultnodestyle=nodestyle(ymargin=0.2cm, drawfn=FillDrawer(yellow));
```



```
defaultdrawstyle=drawstyle(p=red+fontsize(10pt)+fontcommand("\
    ttfamily"), arrow=Arrow);

// define nodes
node[] b=boxes(10cm, new real[]{5/8, 1/8, 2/8}, "Index", "TI",
     "RPL");
labelin("15", b[0], NW, NE);
labelin("2", b[1], N);
labelin("1", b[2], NW, NE);
labelin("0", b[2], NE, NW);
// node b=hpack(centerat=1, b1, b2, b3);

node m=vbox(2cm, "$\cdots$", "X", "$\cdots$");
labelin("GDT", m, S);
node n=vbox(2cm, "$\cdots$", "X", "$\cdots$");
labelin("LDT", n, S);

// dock and draw nodes
node bb=hdock(0cm, centerat=1, b[0], b[1], b[2]);
node c1=hdock(3cm, centerat=0, m, n);
vdock(2cm, bb, c1) @ refresh @ deepdraw;

// draw edges
(b[0]--VH--m).draw();
(b[1]--m).l("TI=1").draw();
(b[1]--VHV--n).l("TI=0").style("leftside").draw();
// (middle(b[1],m)--HV--n).l("TI=0").style("leftside").draw();

// Notes: center depend on edges
// using pack: only one center: m, n
// using dock: multiple centers: b
```

3.6 Circuit



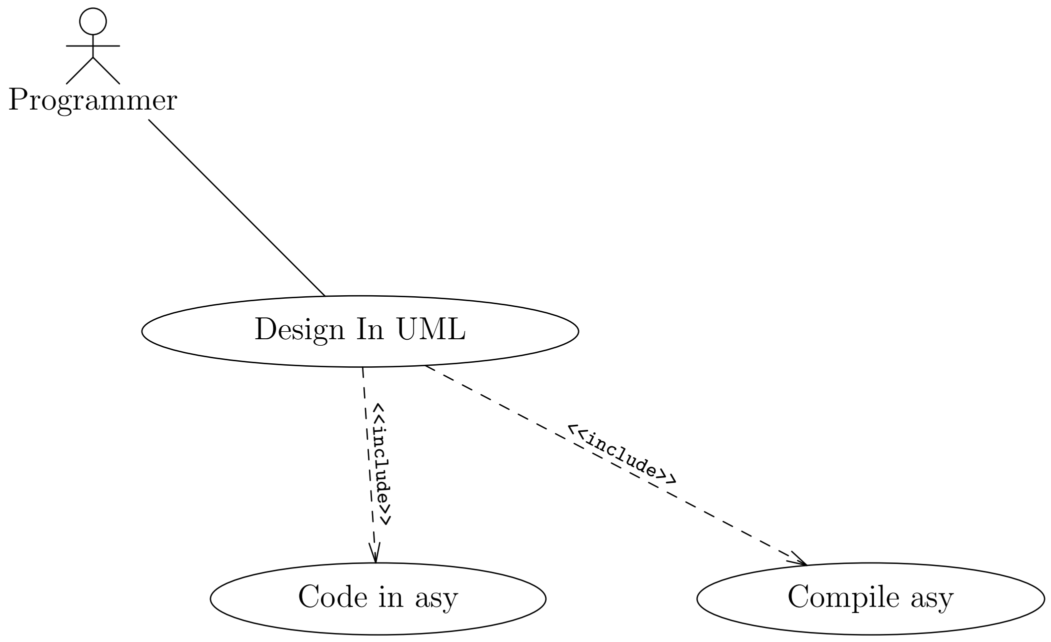
```
// import node;
import nodecircuit;

defaultcircuitlightstyle=symbolstyle(textpen=red+fontcommand("
   \ttfamily"), filler=Filler(yellow),
    indrawer=red, outdrawer=red);
defaultcircuitbatterystyle=symbolstyle(textpen=blue+
   fontcommand("\ttfamily"), indrawer=InDrawer(blue,1,3));
defaultcircuitswitchstyle=symbolstyle(textpen=green+
   fontcommand("\ttfamily"), indrawer=green);

node l1=circuit_light("abcdefg", N);
node l2=circuit_light("$abcdefg$", E);
node l3=circuit_light("$abcdefg$", S);
node l4=circuit_light("$abcdefg$", W);
node b1=circuit_battery(E, "$abcdefg$");
node b2=circuit_battery(S, "$abcdefg$");
node b3=circuit_battery(W, "$abcdefg$");
node b4=circuit_battery(N, "$abcdefg$");
node s1=circuit_switch(N, "$abcdefg$");
node s2=circuit_switch(E, "$abcdefg$");
node s3=circuit_switch(S, "$abcdefg$");
node s4=circuit_switch(W, "$abcdefg$");

real u=2cm;
real v=2cm;
l1<<eright(u)<<b1<<eright(u)<<s1<<eright(u)<<new node<<
edown(v)<<l2<<edown(v)<<b2<<edown(v)<<s2<<edown(v)<<new node<<
eleft(u)<<l3<<eleft(u)<<b3<<eleft(u)<<s3<<eleft(u)<<new node<<
eup(v)<<l4<<eup(v)<<b4<<eup(v)<<s4<<eup(u);

draw(l1--b1);
draw(b1--s1);
draw(s1--HV--l2);
draw(l2--b2);
draw(b2--s2);
draw(s2--VH--l3);
draw(l3--b3);
draw(b3--s3);
draw(s3--HV--l4);
draw(l4--b4);
draw(b4--s4);
draw(s4--VH--l1);

draw(l1,l2,l3,l4,b1,b2,b3,b4,s1,s2,s3,s4);
```

3.7 SML: Simple Modeling Language[3]

3.7.1 Hello SML

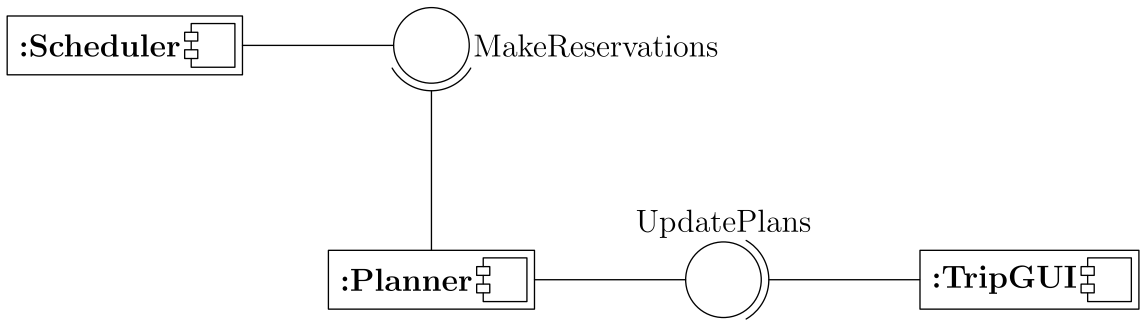
```
import nodesml;

// define style
defaultnodestyle=nodestyle(mag=1.4);
defaultdrawstyle=drawstyle(align=LeftSide, p=fontsize(8pt)+
   fontcommand("\ttfamily")+dashed, Arrow(SimpleHead));
drawstyle es2=drawstyle(p=fontsize(8pt)+fontcommand("\ttfamily
   "));

// define nodes
real symsize=0.5cm;
node a=sml_actor(symsize, "Programmer");
node c=sellipse("Design␣In␣UML");
node c1=sellipse("Code␣in␣asy");
node c2=sellipse("Compile␣asy");
```

[3]This is from sml4asy by cuichaox@gmail.com



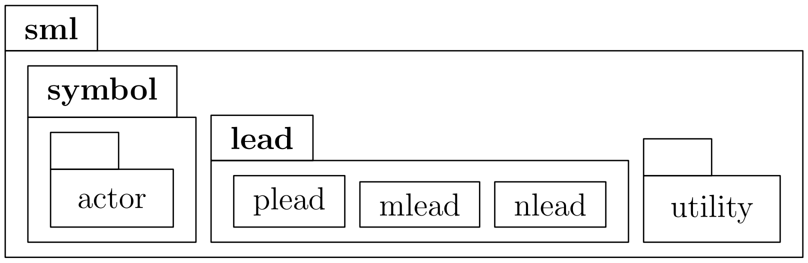
```

// dock and draw nodes
real u=2cm, v=5cm;
dock(dir(-45), v, centerat=1, rel=false,
    a, c, hdock(u, c1, c2)) @ refresh @ deepdraw;

// draw edges
(a--c).style(es2).draw();
(c--c1).l("<<include>>").style("autorot").draw();
(c--c2).l("<<include>>").style("autorot").draw();
```

3.7.2 Component



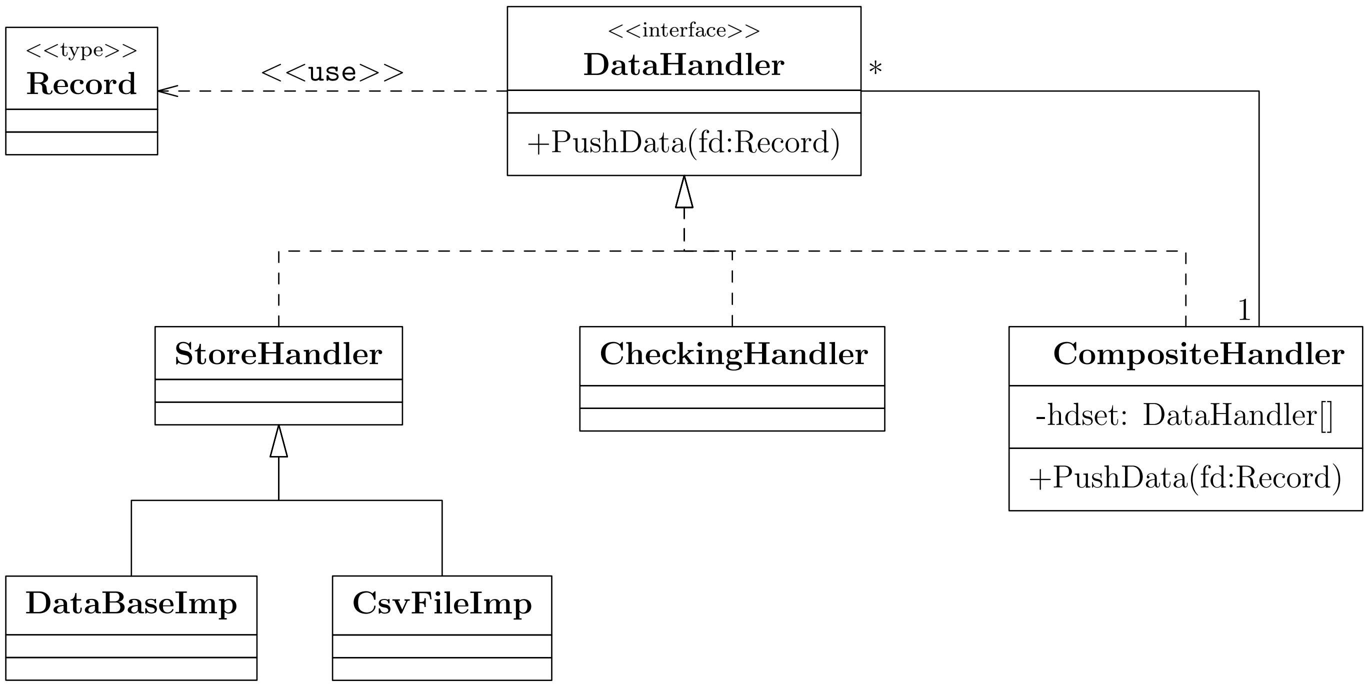
```
import nodesml;

node a=sml_com(":Scheduler");
node b=sml_iball(S, "MakeReservations");
node c=sml_com(":Planner");
node d=sml_iball(E, "UpdatePlans");
node e=sml_com(":TripGUI");

real u=2cm, v=2cm;
vdock(v,
    hdock(u, centerat=-1, a, b),
    hdock(u, centerat=0, c, d, e)) @ refresh @ deepdraw;

draw(a--b, b--c, c--d, d--e);
```

3.7.3 SML Lead



```
import nodebox;

node sml_lead(string s ... node[] nds)
{
  nodestyle boxstyle=nodestyle(xmargin=0.2cm, ymargin=0.1cm);
  node body=hpack(flush=S, skip=0.2cm, xmargin=0.3cm, ymargin
     =0.2cm, drawfn=Drawer ... nds);
  s="\bfseries␣"+s;
  pair D=gettextsize(s);
  real refh=gettextsize("e").y;
  node lead=sbox((max(0.5cm,D.x), max(refh, D.y)), s, boxstyle
     );
  return vpack(flush=W, lead, body);
}

node sml_lead(string s ... string[] strs)
{
  nodestyle nonestyle;
  nodestyle boxstyle=nodestyle(xmargin=0.2cm, ymargin=0.1cm);
  node[] nds;
  if (strs.length==1)
    nds.push(snone(strs[0], nonestyle));
  else
  {
    for (string str: strs)
      nds.push(sbox(str, boxstyle));
  }
  return sml_lead(s ... nds);
}

node c1=sml_lead("symbol", sml_lead("", "actor"));
node c2=sml_lead("lead", "plead", "mlead", "nlead");
node c3=sml_lead("", "utility");
node cc=sml_lead("sml", c1, c2, c3);

draw(cc);
```

3.7.4 SML Class



```
import nodebox;

// define sml_class
node sml_class(string name="", string id="", string attribs=""
    , string opers="")
{
  if (id!="")
    name="{\scriptsize␣$<<$"+id+"$>>$}\par␣{\bfseries␣"+name+"
       }";
  else
```

```
    name="\bfseries␣"+name;
  nodestyle boxstyle=nodestyle(xmargin=0.2cm, ymargin=0.15cm);
  return vbox(new nodestyle[]{boxstyle}, minipage2(name),
      attribs, opers);
}

// define style
drawstyle es2=drawstyle(p=dashed+fontcommand("\ttfamily"),
   Arrow(SimpleHead));
drawstyle es3=drawstyle(p=dashed, BeginArrow(12,NoFill));
drawstyle es4=drawstyle(BeginArrow(12,NoFill));

// define nodes
node record=sml_class("Record", "type");
node datah=sml_class("DataHandler", "interface", "", "+
   PushData(fd:Record)");
node storeh=sml_class("StoreHandler");
node checkh=sml_class("CheckingHandler");
node comph=sml_class("CompositeHandler","","-hdset:␣
   DataHandler[]", "+PushData(fd:Record)");
node dbi=sml_class("DataBaseImp");
node cfi=sml_class("CsvFileImp");

// dock, flush and draw nodes
node c1=hdock(1cm, dbi, cfi);
node c2=vdock(2cm, centerat=0, storeh, c1);
node c3=hdock(6cm, flush=N, rel=false, c2, checkh, comph);
node c4=hdock(4cm, centerat=1, record, datah);
node cc=vdock(2cm, c4, c3) @ refresh;
flush(W, dbi, record);
cc @ deepdraw;

// draw edges
(datah--record).l("$<<$use$>>$").style(es2).draw();
(datah--VHVd(1cm)--storeh).style(es3).draw();
(datah--VHVd(1cm)--checkh).style(es3).draw();
(datah--VHVd(1cm)--comph).style(es3).draw();
(storeh--VHVd(1cm)--dbi).style(es4).draw();
(storeh--VHVd(1cm)--cfi).style(es4).draw();
(datah--HV--node(pos=comph^NNE)).draw();

// label
label("*",datah^E,NE);
label("1",comph^NNE,NW);
```