

Asymptote node.asy Examples

Tao Wei
taowei@buffalo.edu

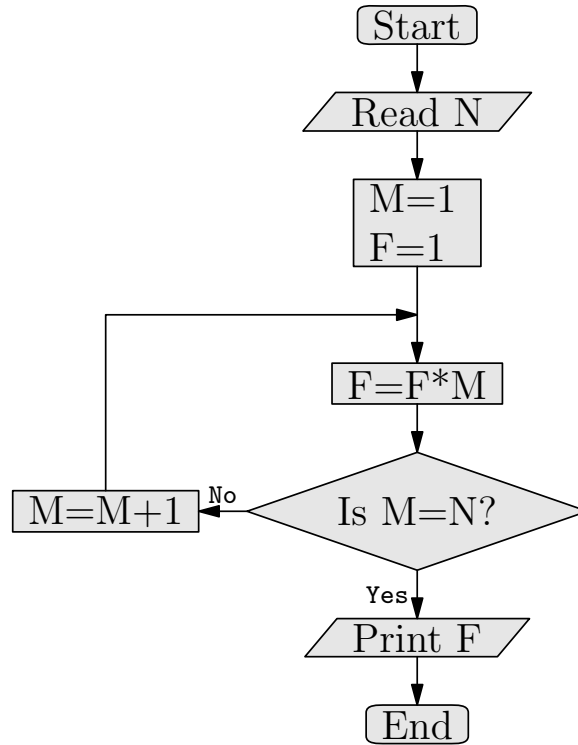
Contents

1	Syntax	1
1.1	Basic	1
1.2	Automatically Calculating Nodes Position and Style Drawing	3
1.3	Dock Syntax	4
1.4	Edge Struct	5
2	Edge Length and Length Between Nodes	5
2.1	Edge Length	5
2.2	Length Between Nodes	7
3	Functionality	7
3.1	Graph Illustration	7
3.2	Flowchart	8
3.3	Graph Theory	9
3.4	Automata	10
3.5	Boxes	12
3.5.1	Box Illustration	12
3.5.2	Box Example	14
3.6	Circuit	15
3.7	SML	17
3.7.1	Hello SML	17
3.7.2	Component	18
3.7.3	SML Lead	18
3.7.4	SML Class	20

1 Syntax

1.1 Basic

```
1 import node;  
2  
3 pair u=(3.5cm,0);
```



```

4 pair v=(0,1.2cm);
5 real dadjust=1.5;      // diamond vertical adjustment ratio
6 real sadjust=1.2;      // special adjustment ratio
7
8 Arrow=Arrow(6);
9 pen edgepen=fontsize(8pt)+fontcommand("\tttfamily");
10 draw_t style1=FillDrawer(lightgray,black);
11
12 real xmargin=3pt;
13 real ymargin=3pt;
14 real mag=1;
15
16 node start=roundbox("Start",(0,0),xmargin,style1,mag);
17 node read=parallelogram("Read_␣N", start.pos-v,xmargin,style1,
    mag);
18 node b1=box(minipage("M=1\\_␣F=1",1cm), read.pos-v, xmargin,
    style1,mag);
19 node b2=box("F=F*M",b1.pos-sadjust*v,xmargin,style1,mag);
20 node d1=diamond("Is_␣M=N?",b2.pos-dadjust*v,0,ymargin,style1,
    mag);
21 node b3=box("M=M+1",d1.pos-u,xmargin,style1,mag);
22 node print=parallelogram("Print_␣F",d1.pos-dadjust*v,xmargin,
    style1,mag);

```

```

23 node end=roundbox("End",print.pos-v,xmargin,style1,mag);
24
25 draw(start,read,print,end, b1,b2,b3,d1);
26
27 draw(start--read,edgepen,Arrow);
28 draw(read--b1,edgepen,Arrow);
29 draw(b1--b2,edgepen,Arrow);
30 draw(b2--d1,edgepen,Arrow);
31 draw("Yes",d1--print,edgepen,Arrow);
32 draw("No",d1--b3,edgepen,Arrow);
33 draw(print--end,edgepen,Arrow);
34 draw(b3--VH--middle(b1,b2),edgepen,Arrow);

```

1.2 Automatically Calculating Nodes Position and Style Drawing

```

1 import node;
2
3 // define style
4 defaultnodestyle=nodestyle(xmargin=3pt, ymargin=0, drawfn=
  FillDrawer(lightgray,black));
5 nodestyle ns2=nodestyle(xmargin=0, ymargin=3pt, drawfn=
  FillDrawer(lightgray,black));
6 defaultdrawstyle=drawstyle(p=fontsize(8pt)+fontcommand("\
  ttfamily"));
7
8 // define node
9 node start=sroundbox("Start");
10 node read=sparallelogram("Read_□N");
11 node b1=sbox(minipage("M=1\\□F=1",1cm));
12 node b2=sbox("F=F*M");
13 node d1=sdiamond("Is_□M=N?",0,ns2);
14 node b3=sbox("M=M+1");
15 node print=sparallelogram("Print_□F");
16 node end=sroundbox("End");
17
18 // calc node position
19 real u=0.5cm;
20 real v=0.5cm;
21 start<<reldown(v)<<read<<reldown(v)<<b1<<reldown(1.5*v)<<
22   b2<<reldown(v)<<d1<<reldown(v)<<print<<reldown(v)<<end
23   ;
24 d1<<relleft(u)<<b3;
25 // draw node

```

```

26 draw(start,read,print,end, b1,b2,b3,d1);
27
28 // draw edge
29 sdraw(start--read);
30 sdraw(read--b1);
31 sdraw(b1--b2);
32 sdraw(b2--d1);
33 sdraw("Yes",d1--print);
34 sdraw("No",d1--b3);
35 sdraw(print--end);
36 sdraw(b3--VH--middle(b1,b2));

```

1.3 Dock Syntax

```

1 import node;
2
3 // define style
4 defaultnodestyle=nodestyle(xmargin=3pt, ymargin=0, drawfn=
  FillDrawer(lightgray,black));
5 nodestyle ns2=nodestyle(xmargin=0, ymargin=3pt, drawfn=
  FillDrawer(lightgray,black));
6 defaultdrawstyle=drawstyle(p=fontsize(8pt)+fontcommand("\
  ttfamily"), arrow=Arrow(6));
7
8 // define node
9 node start=sroundbox("Start");
10 node read=sparallelogram("ReadN");
11 node b1=sbox(minipage("M=1\\NF=1",1cm));
12 node b2=sbox("F=F*M");
13 node d1=sdiamond("IsNM=N?",0,ns2);
14 node b3=sbox("M=M+1");
15 node print=sparallelogram("PrintF");
16 node end=sroundbox("End");
17
18 // calc node position
19 real u=0.5cm;
20 real v=0.5cm;
21 node c1=vdock(v, centerat=-3,
22   start,read,b1,new node,b2,d1,print,end);
23 hdock(u, centerat=1,
24   b3, c1) @ refresh @ deepdraw;
25
26 // draw edge
27 sdraw(start--read);
28 sdraw(read--b1);

```

```

29 sdraw(b1--b2);
30 sdraw(b2--d1);
31 sdraw("Yes",d1--print);
32 sdraw("No",d1--b3);
33 sdraw(print--end);
34 sdraw(b3--VH--middle(b1,b2));

```

1.4 Edge Struct

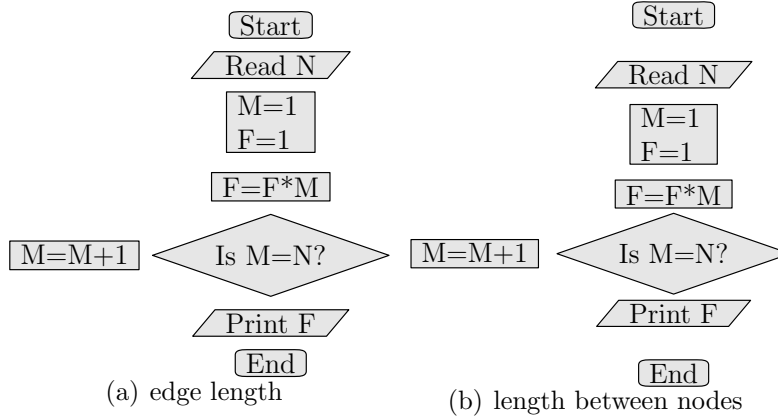
```

1 import node;
2
3 // define style
4 defaultnodestyle=nodestyle(xmargin=3pt, ymargin=0, drawfn=
  FillDrawer(lightgray,black));
5 nodestyle ns2=nodestyle(xmargin=0, ymargin=3pt, drawfn=
  FillDrawer(lightgray,black));
6 defaultdrawstyle=drawstyle(p=fontsize(8pt)+fontcommand("\
  ttfamily"), arrow=Arrow(6));
7
8 // define node
9 node start=sroundbox("Start");
10 node read=sparallelogram("Read_□N");
11 node b1=sbox(minipage("M=1\\□F=1",1cm));
12 node b2=sbox("F=F*M");
13 node d1=sdiamond("Is_□M=N?",0,ns2);
14 node b3=sbox("M=M+1");
15 node print=sparallelogram("Print_□F");
16 node end=sroundbox("End");
17
18 // dock position
19 real u=0.5cm;
20 real v=0.5cm;
21 node c1=vdock(v, centerat=-3,
22   start,read,b1,new node,b2,d1,print,end);
23 hdock(u, centerat=1,
24   b3, c1) @ refresh @ deepdraw;
25
26 // draw edge
27 draw(start--read, read--b1, b1--b2, b2--d1, (d1--print).l("Yes
  "),
28   (d1--b3).l("No"), print--end, b3--VH--middle(b1,b2));

```

2 Edge Length and Length Between Nodes

2.1 Edge Length



```

1 import node;
2
3 // define style
4 defaultnodestyle=nodestyle(xmargin=3pt, ymargin=0, drawfn=
  FillDrawer(lightgray,black));
5 nodestyle ns2=nodestyle(xmargin=0, ymargin=3pt, drawfn=
  FillDrawer(lightgray,black));
6 defaultdrawstyle=drawstyle(p=fontsize(8pt)+fontcommand("\
  ttfamily"));
7
8 // define node
9 node start=sroundbox("Start");
10 node read=sparallelogram("Read_N");
11 node b1=sbox(minipage("M=1\\_F=1",1cm));
12 node b2=sbox("F=F*M");
13 node d1=sdiamond("Is_M=N?",0,ns2);
14 node b3=sbox("M=M+1");
15 node print=sparallelogram("Print_F");
16 node end=sroundbox("End");
17
18 // calc node position
19 real u=0.2cm;
20 real v=0.2cm;
21 start<<reldown(v)<<read<<reldown(v)<<b1<<reldown(1.5*v)<<
22   b2<<reldown(v)<<d1<<reldown(v)<<print<<reldown(v)<<end
23   ;
24 d1<<relleft(u)<<b3;
25
26 // draw node
27 draw(start,read,print,end, b1,b2,b3,d1);

```

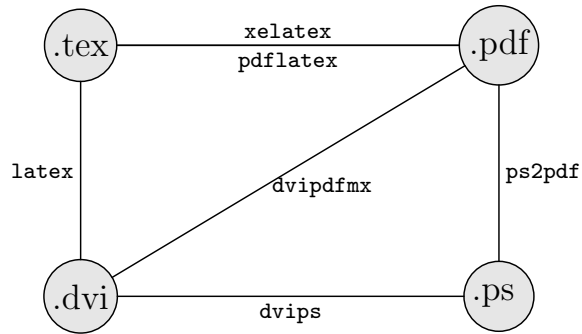
2.2 Length Between Nodes

```
1 import node;
2
3 // define style
4 defaultnodestyle=nodestyle(xmargin=3pt, ymargin=0, drawfn=
    FillDrawer(lightgray,black));
5 nodestyle ns2=nodestyle(xmargin=0, ymargin=3pt, drawfn=
    FillDrawer(lightgray,black));
6 defaultdrawstyle=drawstyle(p=fontsize(8pt)+fontcommand("\
    ttfamily"));
7
8 // define node
9 node start=sroundbox("Start");
10 node read=sparallelogram("Read_N");
11 node b1=sbox(minipage("M=1\\_F=1",1cm));
12 node b2=sbox("F=F*M");
13 node d1=sdiamond("Is_M=N?",0,ns2);
14 node b3=sbox("M=M+1");
15 node print=sparallelogram("Print_F");
16 node end=sroundbox("End");
17
18 // calc node position
19 real u=3.0cm;
20 real v=0.9cm;
21 start<<edown(v)<<read<<edown(v)<<b1<<edown(v)<<
22     b2<<edown(v)<<d1<<edown(v)<<print<<edown(v)<<end;
23 d1<<eleft(u)<<b3;
24
25 // draw node
26 draw(start,read,print,end, b1,b2,b3,d1);
```

3 Functionality

3.1 Graph Illustration

```
1 import node;
2
3 // define style
4 defaultnodestyle=nodestyle(drawfn=FillDrawer(lightgray,black))
    ;
5 defaultdrawstyle=drawstyle(p=fontsize(8pt)+fontcommand("\
    ttfamily"));
6
7 // define nodes
```



```

8 node q0=scircle(".tex");
9 node q1=scircle(".dvi");
10 node q2=scircle(".pdf");
11 node q3=scircle(".ps");
12
13 // calc position
14 real u=5cm;
15 real v=3cm;
16 q0<<edown(v)<<q1<<eright(u)<<q3<<eup(v)<<q2;
17
18 // draw nodes
19 draw(q0,q1,q2,q3);
20
21 // draw edges
22 draw((q0--q1).l("latex"),
23      (q0--q2).l("pdflatex"),
24      (q0--q2).l("xelatex").style("leftside"),
25      (q1--q3).l("dvips"),
26      (q3--q2).l("ps2pdf"),
27      (q1--q2).l("dvipdfmx"));

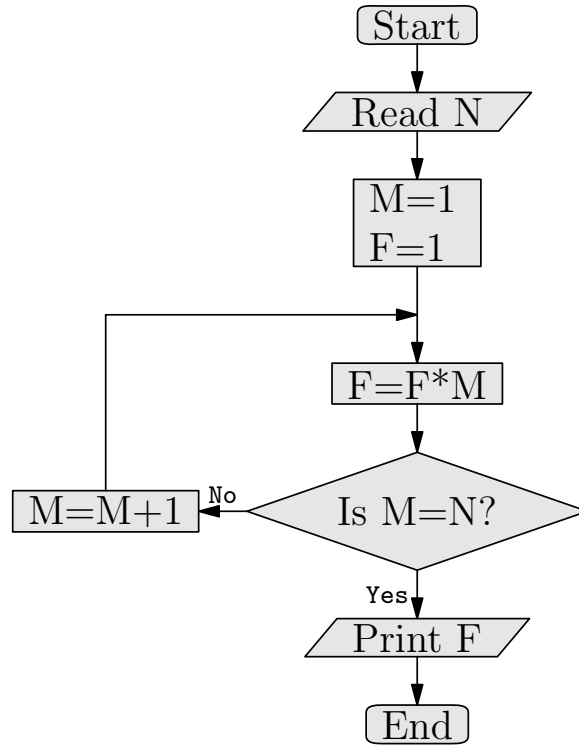
```

3.2 Flowchart

```

1 import node;
2
3 // define style
4 defaultnodestyle=nodestyle(xmargin=3pt, ymargin=0, drawfn=
5   FillDrawer(lightgray,black));
6 nodestyle ns2=nodestyle(xmargin=0, ymargin=3pt, drawfn=
7   FillDrawer(lightgray,black));
8 defaultdrawstyle=drawstyle(p=fontsize(8pt)+fontcommand("\
9   ttfamily"), arrow=Arrow(6));
10
11 // define node

```

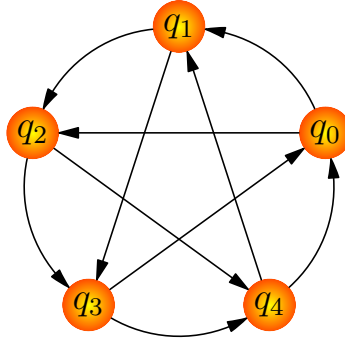



```

9 node start=sroundbox("Start");
10 node read=sparallelogram("Read_N");
11 node b1=sbox(minipage("M=1\\_F=1",1cm));
12 node b2=sbox("F=F*M");
13 node d1=sdiamond("Is_M=N?",0,ns2);
14 node b3=sbox("M=M+1");
15 node print=sparallelogram("Print_F");
16 node end=sroundbox("End");
17
18 // dock position
19 real u=0.5cm;
20 real v=0.5cm;
21 node c1=vdock(v, centerat=-3,
22     start,read,b1,new node,b2,d1,print,end);
23 hdock(u, centerat=1,
24     b3, c1) @ refresh @ deepdraw;
25
26 // draw edge
27 draw(start--read, read--b1, b1--b2, b2--d1, (d1--print).l("Yes
28     "),
29     (d1--b3).l("No"), print--end, b3--VH--middle(b1,b2));

```

3.3 Graph Theory



```

1 import node;
2
3 // define style
4 defaultnodestyle=nodestyle(drawfn=RadialShader(yellow,red));
5 defaultdrawstyle=drawstyle(p=fontsize(8pt)+fontcommand("\
   ttfamily"),
6   arrow=Arrow(6));
7
8 // define nodes
9 node q0=scircle("$q_0$");
10 node q1=scircle("$q_1$");
11 node q2=scircle("$q_2$");
12 node q3=scircle("$q_3$");
13 node q4=scircle("$q_4$");
14
15 // calc node postion and draw
16 real u=2cm;
17 real ang0=360/5, ang1=(180-ang0)/2, ang2=90-ang1, ang=180-ang2
   ;
18 real anginc=360/5;
19 fancydock(dir(ang), u, anginc, (0.5, 0.5),
20   q0, q1, q2, q3, q4) @ refresh @ deepdraw;
21
22 // draw edges
23 draw(q0--q2, q2--q4, q4--q1, q1--q3, q3--q0);
24 draw(q0..bend..q1, q1..bend..q2, q2..bend..q3,
25   q3..bend..q4, q4..bend..q0);

```

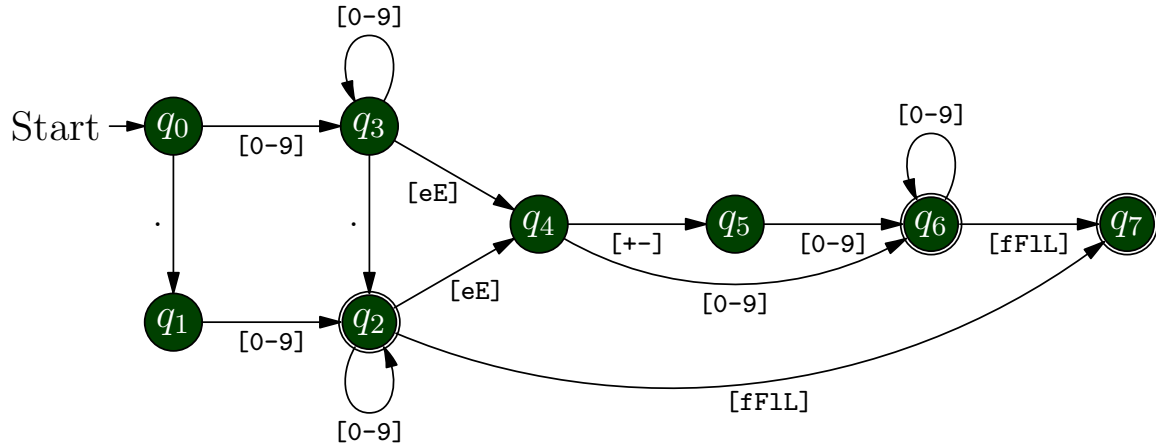
3.4 Automata¹

```

1
2 import node;
3

```

¹This is from AsymptoteByExample by Leoliu



```

4 // define node style
5 defaultnodestyle=nodestyle(textpen=white,
6     drawfn=FillDrawer(darkgreen,black));
7 nodestyle ns2=nodestyle(textpen=white,
8     drawfn=Filler(darkgreen)+DoubleDrawer(black));
9 nodestyle ns3=nodestyle(drawfn=None);
10 // define edge style
11 defaultdrawstyle=drawstyle(p=fontsize(8pt)+fontcommand("\
12     ttfamily"),
13     arrow=Arrow(6));
14 // define nodes
15 node q0=scircle("$q_0$");
16 node q1=scircle("$q_1$");
17 node q2=scircle("$q_2$",ns2);
18 node q3=scircle("$q_3$");
19 node q4=scircle("$q_4$");
20 node q5=scircle("$q_5$");
21 node q6=scircle("$q_6$",ns2);
22 node q7=scircle("$q_7$",ns2);
23 node start=scircle("Start",ns3);
24
25 // calculate nodes position
26 real u=2cm;
27 start<<eright(0.6u)<<q0<<edown(u)<<q1<<eright(u)<<
28 q2<<eup(u)<<q3<<edir(-30,u)<<q4<<eright(u)<<
29 q5<<eright(u)<<q6<<eright(u)<<q7;
30
31 // draw nodes
32 draw(start,q0,q1,q2,q3,q4,q5,q6,q7);
33

```

```

34 // draw edges
35 draw(
36     (q0--q1).l("."),
37     (q1--q2).l("[0-9]"),
38     (q3--q2).l("."),
39     (q2--q4).l("[eE]"),
40     (q0--q3).l("[0-9]").style("leftside"),
41     (q3--q4).l("[eE]").style("leftside"),
42     (q4--q5).l("[+-]").style("leftside"),
43     (q5--q6).l("[0-9]").style("leftside"),
44     (q6--q7).l("[fFlL]").style("leftside"),
45     (q3..loop(N)).l("[0-9]"),
46     (q2..loop(S)).l("[0-9]"),
47     (q6..loop(N)).l("[0-9]"),
48     (q4..bend..q6).l("[0-9]"),
49     (q2..bend..q7).l("[fFlL]"),
50     (start--q0));

```

3.5 Boxes²

3.5.1 Box Illustration

```

1 import nodebox;
2
3 // define styles
4 nodestyle ns0=defaultnodestyle;
5 nodestyle ns1=nodestyle(FillDrawer(yellow));
6 nodestyle ns2=nodestyle(FillDrawer(gray));
7 nodestyle ns3=nodestyle(white, FillDrawer(darkgreen));
8 nodestyle ns4=nodestyle(FillDrawer(red));
9 nodestyle ns5=nodestyle(FillDrawer(blue));
10
11 // define nodes
12 node[] a=boxes(10cm, new real[]{13/16, 1/16, 2/16}, "Index", "
    TI", "RPL");
13 labelin("15", a[0], NW, NE);
14 labelin("2", a[1], N);
15 labelin("1", a[2], NW, NE);
16 labelin("0", a[2], NE, NW);
17 node c1=hpack(a[0], a[1], a[2]);
18 node c2=hbox(10cm, new real[]{13/16, 1/16, 2/16}, "Index", "
    Long␣TI", "RPL");
19 node c3=hbox(10cm, new real[]{13/16, 1/16, 2/16}, new
    nodestyle[]{ns1, ns2, ns3}, "Index", "Long␣TI", "RPL");

```

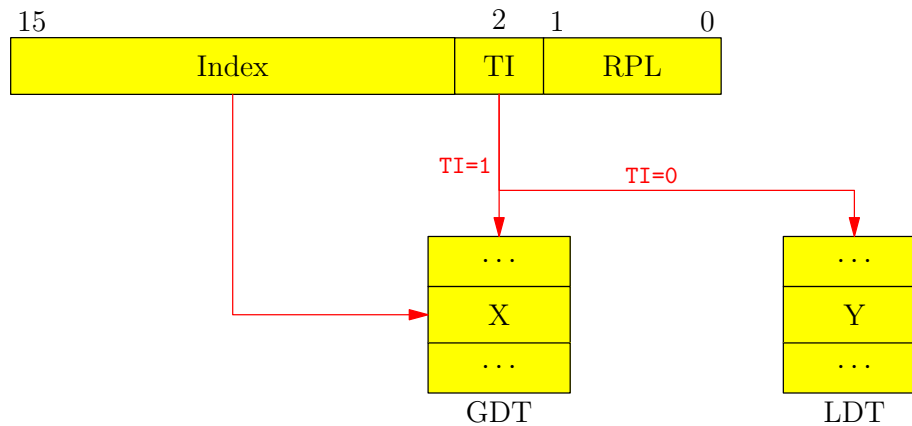
²This is from BoxesForAsymptote by Addylee2004@163.com


```

20 node c4=vbox(3cm, "$\cdots$", "PDE_{$n}$", "PDE_{$n+1}$", "$\cdots$");
21 node c5=vbox(1cm, "$\cdots$", "PDE_{$n}$", "PDE_{$n+1}$", "$\cdots$");
22 node c6=vbox(flush=W, "$\cdots$", "PDE_{$n}$", "PDE_{$n+1}$", "$\cdots$");
23 node c7=hbox(flush=N, "one_line", minipage2("two\par_lines"),
    "flush_up");
24 node c8=vbox("$\cdots$", "PDE_{$n}$", "PDE_{$n+1}$", "$\cdots$");
25 node c9=vpack(
26     hbox("G","D","0","0","0","0","Limit"),
27     hbox("Base_Address"),
28     hbox("1","1","0","1"));
29 node c10=vpack(
30     hbox(new nodestyle[] {ns2, ns2, ns4, ns4, ns4, ns4, ns3}, "
        G","D","0","0","0","0","Limit"),
31     hbox("Base_Address",ns1),
32     hbox(new nodestyle[] {ns5}, "1","1","0","1"));
33
34 // dock
35 vdock(1cm,
36     c1, c2, c3, hdock(1cm, c4, c5, c6), hdock(3cm, c7, c8), c9
    , c10) @ refresh @ deepdraw;

```

3.5.2 Box Example



```

1 import nodebox;
2
3 // define style
4 defaultnodestyle=nodestyle(ymargin=0.2cm, drawfn=FillDrawer(
    yellow));

```

```

5 defaultdrawstyle=drawstyle(p=red+fontsize(10pt)+fontcommand("\
   ttfamily"), arrow=Arrow);
6
7 // define nodes
8 node[] b=boxes(10cm, new real[]{5/8, 1/8, 2/8}, "Index", "TI",
   "RPL");
9 labelin("15", b[0], NW, NE);
10 labelin("2", b[1], N);
11 labelin("1", b[2], NW, NE);
12 labelin("0", b[2], NE, NW);
13 // node b=hpack(centerat=1, b1, b2, b3);
14
15 node m=vbox(2cm, "$\cdots$", "X", "$\cdots$");
16 labelin("GDT", m, S);
17 node n=vbox(2cm, "$\cdots$", "X", "$\cdots$");
18 labelin("LDT", n, S);
19
20 // dock and draw nodes
21 node bb=hdock(0cm, centerat=1, b[0], b[1], b[2]);
22 node c1=hdock(3cm, centerat=0, m, n);
23 vdock(2cm, bb, c1) @ refresh @ deepdraw;
24
25 // draw edges
26 (b[0]--VH--m).draw();
27 (b[1]--m).l("TI=1").draw();
28 (b[1]--VHV--n).l("TI=0").style("leftside").draw();
29 // (middle(b[1],m)--HV--n).l("TI=0").style("leftside").draw();
30
31 // Notes: center depend on edges
32 // using pack: only one center: m, n
33 // using dock: multiple centers: b

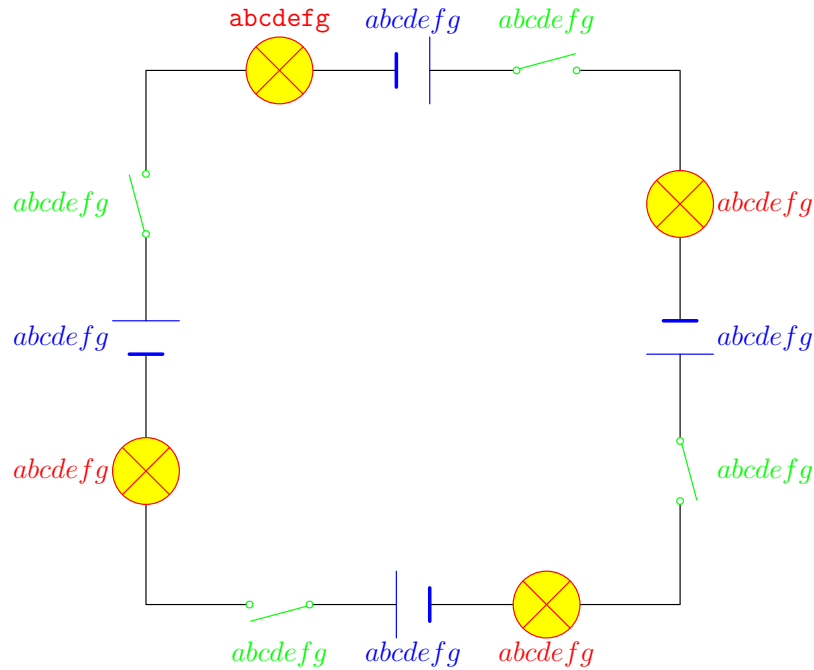
```

3.6 Circuit

```

1 // import node;
2 import nodecircuit;
3
4 defaultcircuitlightstyle=symbolstyle(textpen=red+fontcommand("\
   ttfamily"), filler=Filler(yellow),
5   indrawer=red, outdrawer=red);
6 defaultcircuitbatterystyle=symbolstyle(textpen=blue+
   fontcommand("\tttfamily"), indrawer=InDrawer(blue,1,3));
7 defaultcircuitswitchstyle=symbolstyle(textpen=green+
   fontcommand("\tttfamily"), indrawer=green);

```



```

8
9 node l1=circuit_light("abcdefg", N);
10 node l2=circuit_light("$abcdefg$", E);
11 node l3=circuit_light("$abcdefg$", S);
12 node l4=circuit_light("$abcdefg$", W);
13 node b1=circuit_battery(E, "$abcdefg$");
14 node b2=circuit_battery(S, "$abcdefg$");
15 node b3=circuit_battery(W, "$abcdefg$");
16 node b4=circuit_battery(N, "$abcdefg$");
17 node s1=circuit_switch(N, "$abcdefg$");
18 node s2=circuit_switch(E, "$abcdefg$");
19 node s3=circuit_switch(S, "$abcdefg$");
20 node s4=circuit_switch(W, "$abcdefg$");
21
22 real u=2cm;
23 real v=2cm;
24 l1<<eright(u)<<b1<<eright(u)<<s1<<eright(u)<<new node<<
25 edown(v)<<l2<<edown(v)<<b2<<edown(v)<<s2<<edown(v)<<new node<<
26 elleft(u)<<l3<<elleft(u)<<b3<<elleft(u)<<s3<<elleft(u)<<new node<<
27 eup(v)<<l4<<eup(v)<<b4<<eup(v)<<s4<<eup(u);
28
29 draw(l1--b1);
30 draw(b1--s1);
31 draw(s1--HV--l2);
32 draw(l2--b2);

```



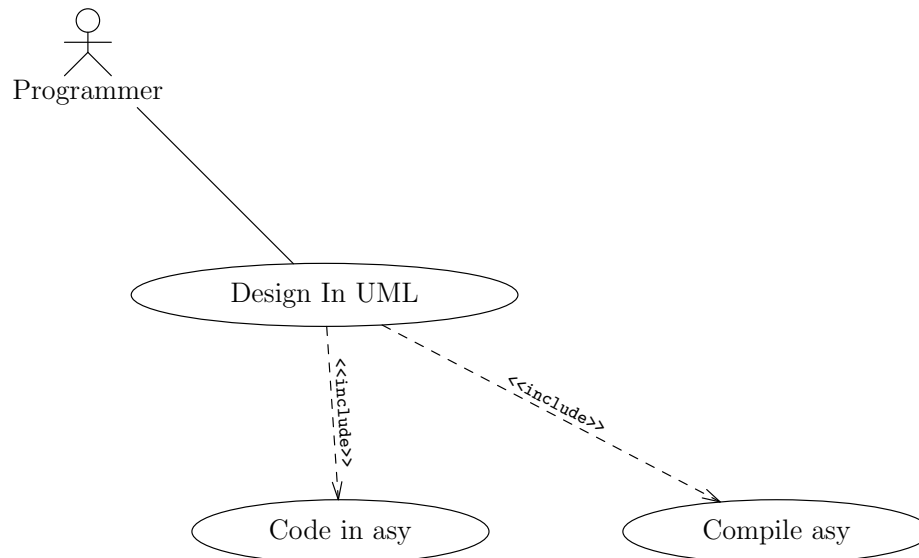
```

33 draw(b2--s2);
34 draw(s2--VH--l3);
35 draw(l3--b3);
36 draw(b3--s3);
37 draw(s3--HV--l4);
38 draw(l4--b4);
39 draw(b4--s4);
40 draw(s4--VH--l1);
41
42 draw(l1,l2,l3,l4,b1,b2,b3,b4,s1,s2,s3,s4);

```

3.7 SML: Simple Modeling Language³

3.7.1 Hello SML



```

1 import nodesml;
2
3 // define style
4 defaultnodestyle=nodestyle(mag=1.4);
5 defaultdrawstyle=drawstyle(align=LeftSide, p=fontsize(8pt)+
    fontcommand("\tttfamily")+dashed, Arrow(SimpleHead));
6 drawstyle es2=drawstyle(p=fontsize(8pt)+fontcommand("\tttfamily
    "));
7
8 // define nodes
9 real symsize=0.5cm;
10 node a=sml_actor(symsize, "Programmer");

```

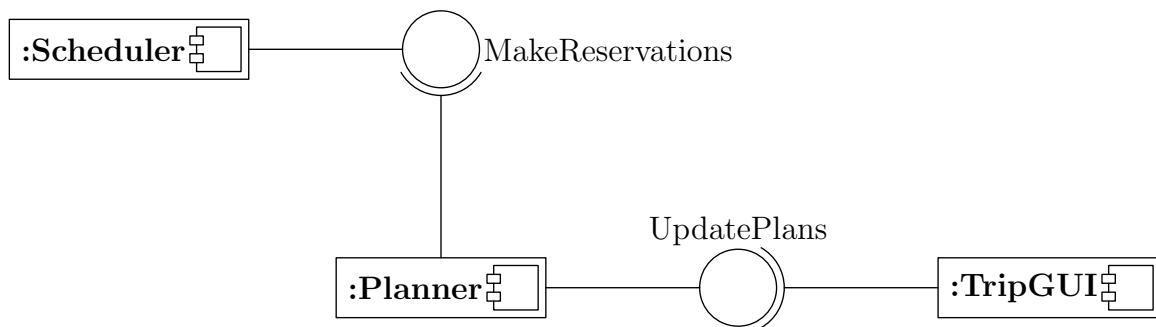
³This is from sml4asy by cuichaoy@gmail.com

```

11 node c=sellipse("Design_In_UML");
12 node c1=sellipse("Code_in_asy");
13 node c2=sellipse("Compile_asy");
14
15 // dock and draw nodes
16 real u=2cm, v=5cm;
17 dock(dir(-45), v, centerat=1, rel=false,
18     a, c, hdock(u, c1, c2)) @ refresh @ deepdraw;
19
20 // draw edges
21 (a--c).style(es2).draw();
22 (c--c1).l("<<include>>").style("autorot").draw();
23 (c--c2).l("<<include>>").style("autorot").draw();

```

3.7.2 Component

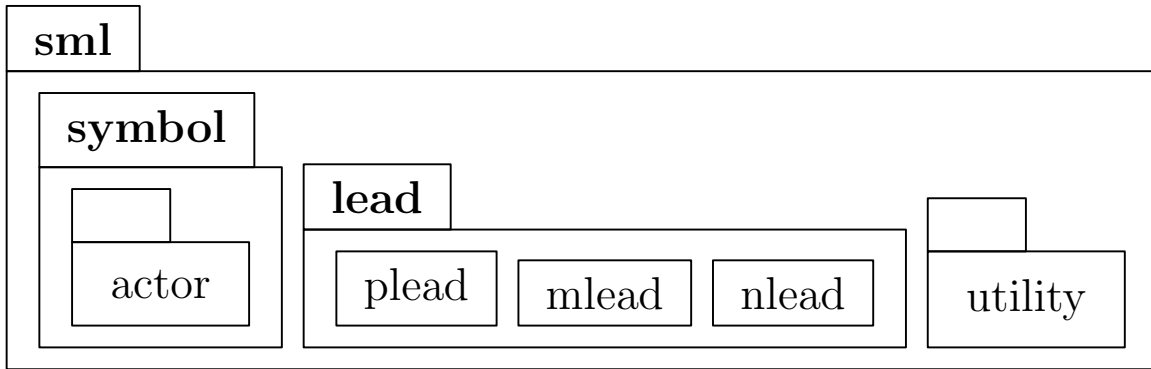


```

1 import nodesml;
2
3 node a=sml_com(":Scheduler");
4 node b=sml_iball(S, "MakeReservations");
5 node c=sml_com(":Planner");
6 node d=sml_iball(E, "UpdatePlans");
7 node e=sml_com(":TripGUI");
8
9 real u=2cm, v=2cm;
10 vdock(v,
11     hdock(u, centerat=-1, a, b),
12     hdock(u, centerat=0, c, d, e)) @ refresh @ deepdraw;
13
14 draw(a--b, b--c, c--d, d--e);

```

3.7.3 SML Lead



```

1 import nodebox;
2
3 node sml_lead(string s ... node[] nds)
4 {
5   nodestyle boxstyle=nodestyle(xmargin=0.2cm, ymargin=0.1cm);
6   node body=hpack(flush=S, skip=0.2cm, xmargin=0.3cm, ymargin
7     =0.2cm, drawfn=Drawer ... nds);
8   s="\bfseries□"+s;
9   pair D=gettextsize(s);
10  real refh=gettextsize("e").y;
11  node lead=sbox((max(0.5cm,D.x), max(refh, D.y)), s, boxstyle
12    );
13  return vpack(flush=W, lead, body);
14 }
15
16 node sml_lead(string s ... string[] str)
17 {
18   nodestyle nonestyle;
19   nodestyle boxstyle=nodestyle(xmargin=0.2cm, ymargin=0.1cm);
20   node[] nds;
21   if (str.length==1)
22     nds.push(snone(str[0], nonestyle));
23   else
24   {
25     for (string str: str)
26       nds.push(sbox(str, boxstyle));
27   }
28   return sml_lead(s ... nds);
29 }
30
31 node c1=sml_lead("symbol", sml_lead("", "actor"));
32 node c2=sml_lead("lead", "plead", "mlead", "nlead");
33 node c3=sml_lead("", "utility");

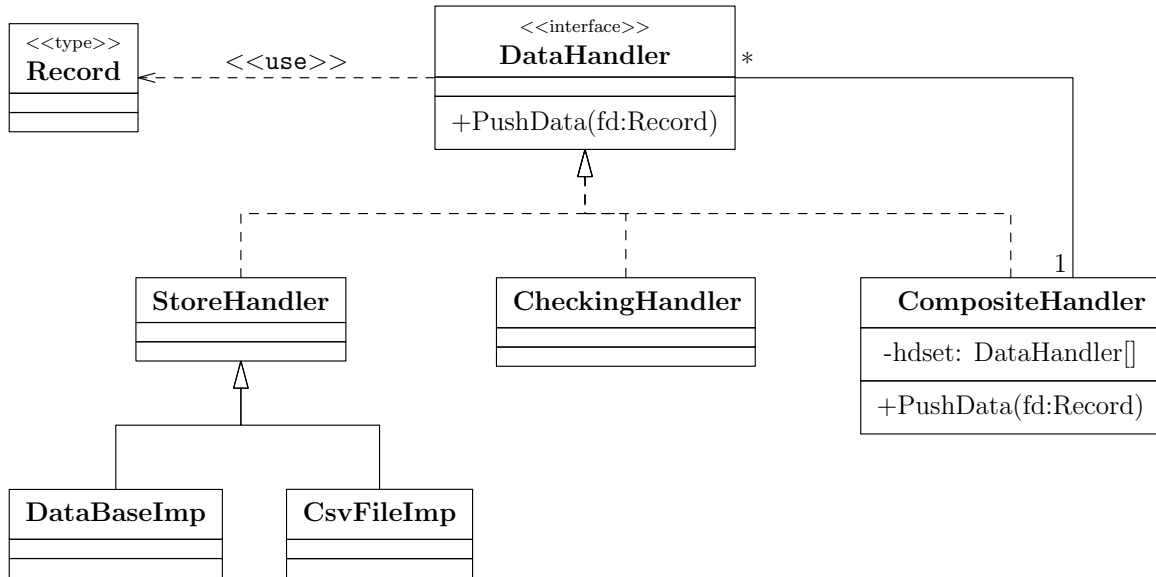
```

```

32 node cc=sml_lead("sml", c1, c2, c3);
33
34 draw(cc);

```

3.7.4 SML Class



```

1 import nodebox;
2
3 // define sml_class
4 node sml_class(string name="", string id="", string attribs=""
5 , string opers="")
6 {
7     if (id!="")
8         name="{\scriptsize_$<<$"+id+"$>>$}\par_{\bfseries_$"+name+"
9         }";
10    else
11        name="\bfseries_$"+name;
12    nodestyle boxstyle=nodestyle(xmargin=0.2cm, ymargin=0.15cm);
13    return vbox(new nodestyle[] {boxstyle}, minipage2(name),
14        attribs, opers);
15 }
16
17 // define style
18 drawstyle es2=drawstyle(p=dashed+fontcommand("\ttfamily"),
19     Arrow(SimpleHead));
20 drawstyle es3=drawstyle(p=dashed, BeginArrow(12,NoFill));
21 drawstyle es4=drawstyle(BeginArrow(12,NoFill));

```

```

19 // define nodes
20 node record=sml_class("Record", "type");
21 node datah=sml_class("DataHandler", "interface", "", "+
    PushData(fd:Record)");
22 node storeh=sml_class("StoreHandler");
23 node checkh=sml_class("CheckingHandler");
24 node comph=sml_class("CompositeHandler","", "-hdset:␣
    DataHandler[]", "+PushData(fd:Record)");
25 node dbi=sml_class("DataBaseImp");
26 node cfi=sml_class("CsvFileImp");
27
28 // dock, flush and draw nodes
29 node c1=hdock(1cm, dbi, cfi);
30 node c2=vdock(2cm, centerat=0, storeh, c1);
31 node c3=hdock(6cm, flush=N, rel=false, c2, checkh, comph);
32 node c4=hdock(4cm, centerat=1, record, datah);
33 node cc=vdock(2cm, c4, c3) @ refresh;
34 flush(W, dbi, record);
35 cc @ deepdraw;
36
37 // draw edges
38 (datah--record).l("$<<$use$>>$").style(es2).draw();
39 (datah--VHVd(1cm)--storeh).style(es3).draw();
40 (datah--VHVd(1cm)--checkh).style(es3).draw();
41 (datah--VHVd(1cm)--comph).style(es3).draw();
42 (storeh--VHVd(1cm)--dbi).style(es4).draw();
43 (storeh--VHVd(1cm)--cfi).style(es4).draw();
44 (datah--HV--node(pos=comph^NNE)).draw();
45
46 // label
47 label("*",datah^E,NE);
48 label("1",comph^NNE,NW);

```