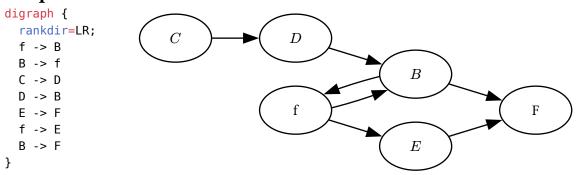
Graph 1: Test



Graph 2: Eating

```
digraph {
  orange -> fruit
                               orange
                                              apple
                                                             carrot
  apple -> fruit
  fruit -> food
  carrot -> vegetable
  vegetable -> food
  food -> eat
  eat -> survive
                                                            vegetable
                                              fruit
}
                                                      food \\
                                                       eat
                                                     survive
```

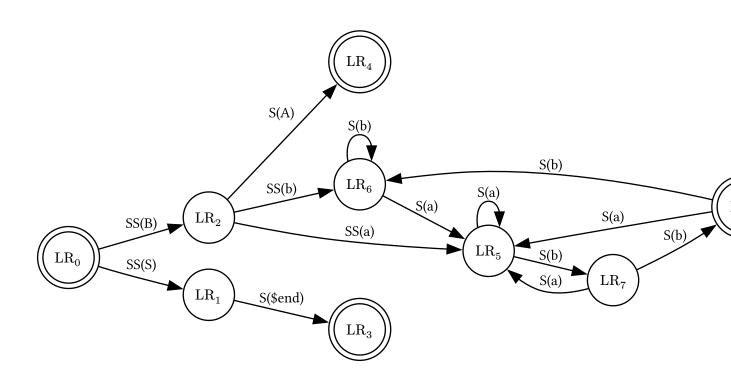
Graph 3: FFT

Labels are overridden manually.

```
digraph {
  node [shape=none]
                                     (1,0,0,0),i
  2
  3
  r1
  r2
  r3
                              (1,0),-1
                                              (0,0),-1
  1->2
  1->3
  2->r1 [color=red]
  3->r2 [color=red]
  r1->r3 [color=red]
  r2->r3 [color=red]
                                (1, 1)
                                                (0,0)
}
                                      (1, 1, 1, 1)
```

Graph 4: State Machine

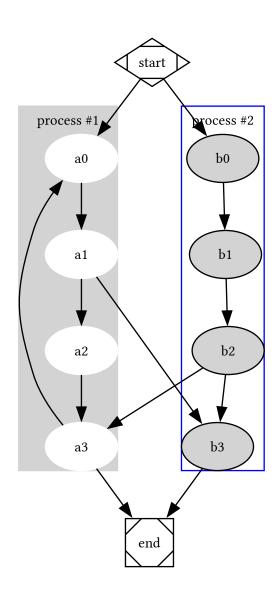
```
digraph finite_state_machine {
  rankdir=LR
  size="8,5"
  node [shape=doublecircle]
  LR 0
  LR_3
  LR_4
  LR_8
  node [shape=circle]
  LR_0 -> LR_2 [label="SS(B)"]
  LR 0 -> LR 1 [label="SS(S)"]
  LR_1 \rightarrow LR_3 [label="S(\$end)"]
  LR_2 -> LR_6 [label="SS(b)"]
  LR_2 -> LR_5 [label="SS(a)"]
  LR_2 -> LR_4 [label="S(A)"]
  LR_5 -> LR_7 [label="S(b)"]
  LR_5 -> LR_5 [label="S(a)"]
  LR 6 -> LR 6 [label="S(b)"]
 LR_6 -> LR_5 [label="S(a)"]
 LR_7 -> LR_8 [label="S(b)"]
  LR_7 -> LR_5 [label="S(a)"]
 LR_8 -> LR_6 [label="S(b)"]
 LR_8 -> LR_5 [label="S(a)"]
```



Graph 5: Clustering

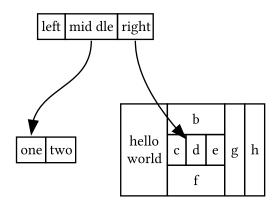
See http://www.graphviz.org/content/cluster.

```
digraph G {
  subgraph cluster 0 {
    style=filled;
    color=lightgrey;
    node [style=filled,color=white];
    a0 -> a1 -> a2 -> a3;
    label = "process #1";
  }
  subgraph cluster 1 {
    node [style=filled];
    b0 -> b1 -> b2 -> b3;
    label = "process #2";
    color=blue
  }
  start -> a0;
  start -> b0;
  a1 -> b3;
 b2 -> a3;
  a3 -> a0;
  a3 -> end;
  b3 -> end;
  start [shape=Mdiamond];
  end [shape=Msquare];
}
```



Graph 6: HTML

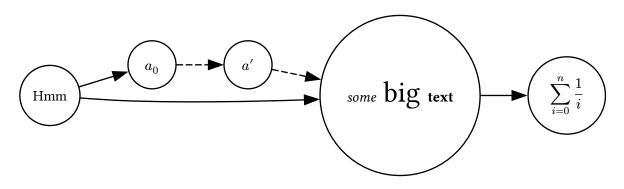
```
digraph structs {
    node [shape=plaintext]
    struct1 [label=<
<TABLE BORDER="0" CELLBORDER="1" CELLSPACING="0">
  <TR><TD>left</TD><TD PORT="f1">mid dle</TD><TD PORT="f2">right</TD></TR>
</TABLE>>];
    struct2 [label=<
<TABLE BORDER="0" CELLBORDER="1" CELLSPACING="0">
  <TR><TD PORT="f0">one</TD><tD>two</TD></TR>
</TABLE>>];
    struct3 [label=<
<TABLE BORDER="0" CELLBORDER="1" CELLSPACING="0" CELLPADDING="4">
    <TD ROWSPAN="3">hello<BR/>world</TD>
    <TD COLSPAN="3">b</TD>
    <TD ROWSPAN="3">q</TD>
    <TD ROWSPAN="3">h</TD>
  </TR>
```



Graph 7: Overridden labels

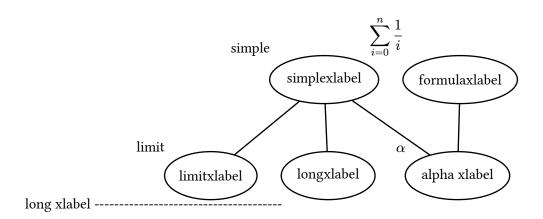
Labels for nodes big and sum are overridden.

```
digraph {
  rankdir=LR
  node[shape=circle]
  Hmm -> a_0
  Hmm -> big
  a_0 -> "a'" -> big [style="dashed"]
  big -> sum
}
```



```
graph {
   simplexlabel[xlabel="simple"]
   simplexlabel -- limitxlabel
   simplexlabel -- longxlabel
   longxlabel[xlabel="long xlabel ------"
   "alpha xlabel"[xlabel="alpha"]
```

```
simplexlabel -- "alpha xlabel"
limitxlabel[xlabel="limit"]
formulaxlabel -- "alpha xlabel"
```



Graph 8: Automatic math labels

