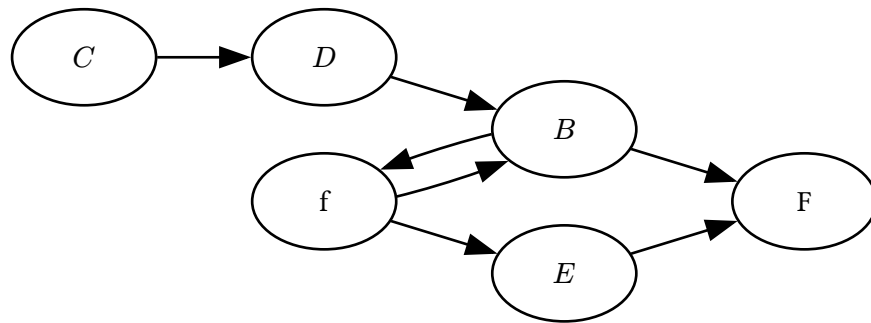


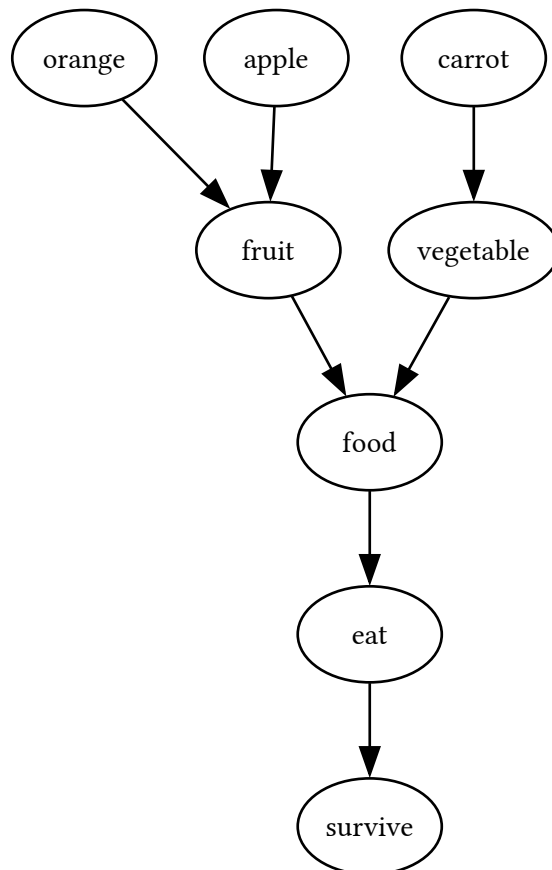
Graph 1: Test

```
digraph {  
  rankdir=LR;  
  f -> B  
  B -> f  
  C -> D  
  D -> B  
  E -> F  
  f -> E  
  B -> F  
}
```



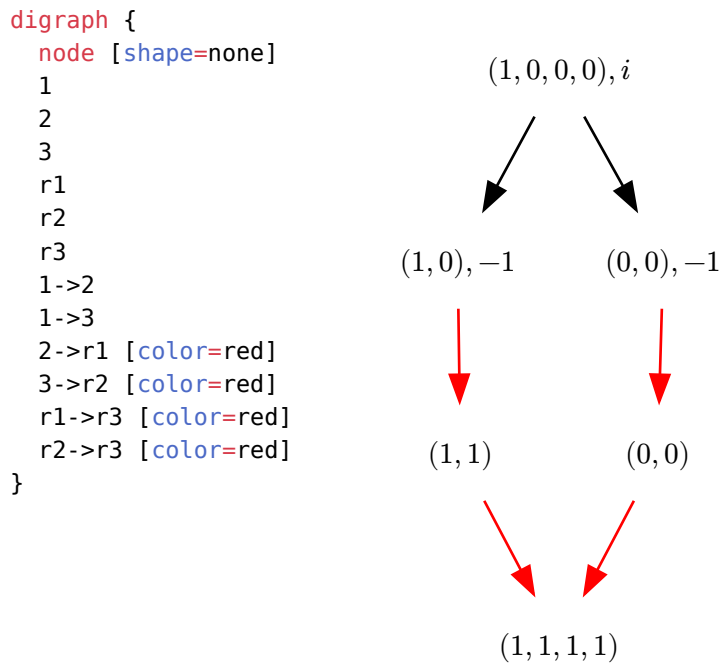
Graph 2: Eating

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

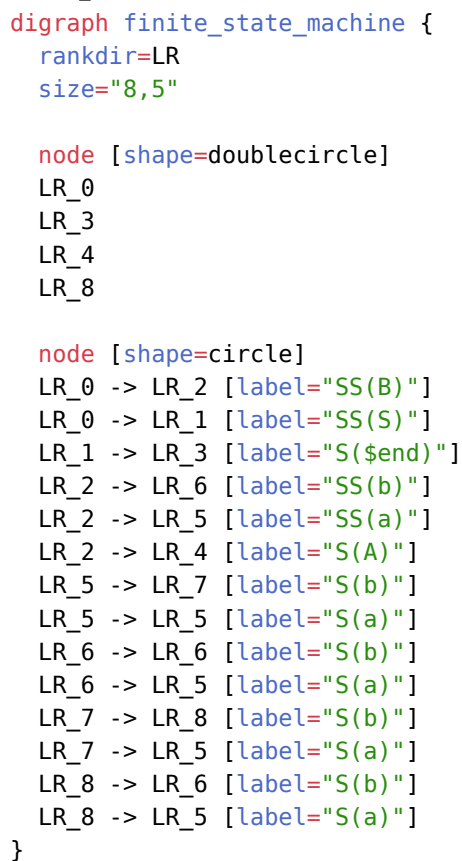


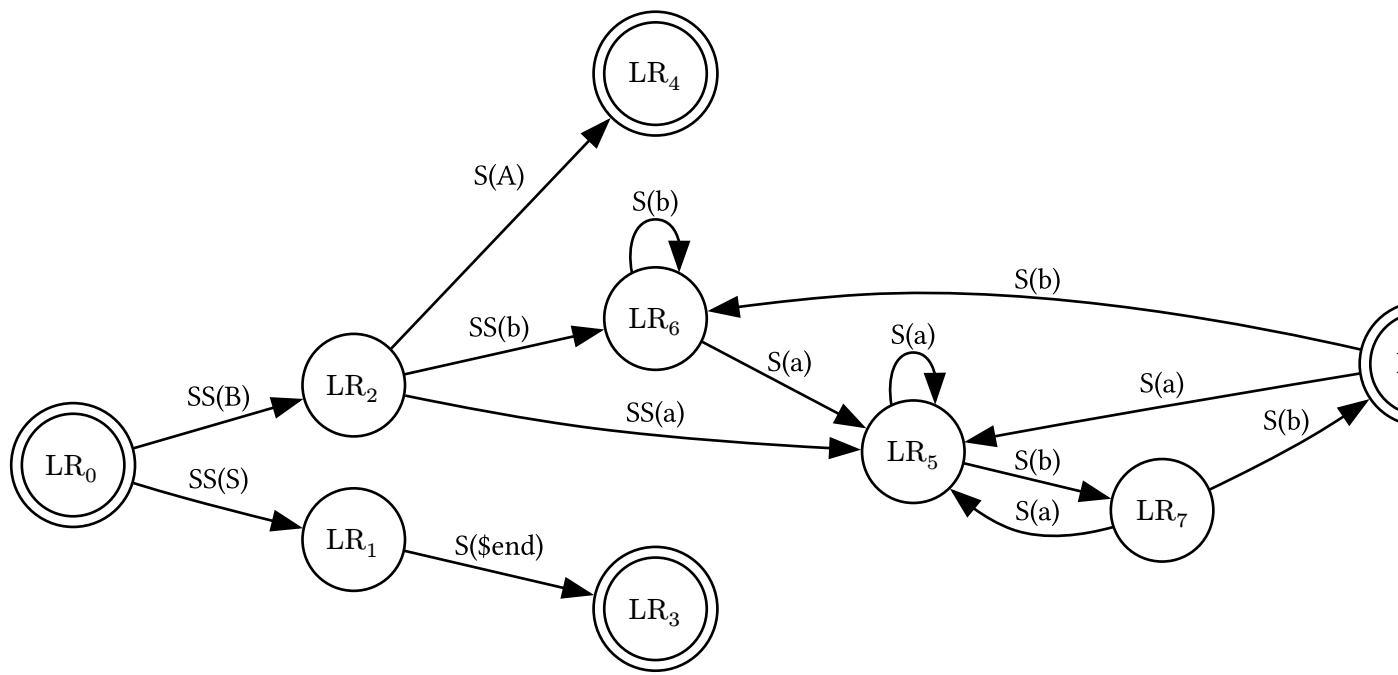
Graph 3: FFT

Labels are overridden manually.



Graph 4: State Machine





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 -> end;
b3 -> end;

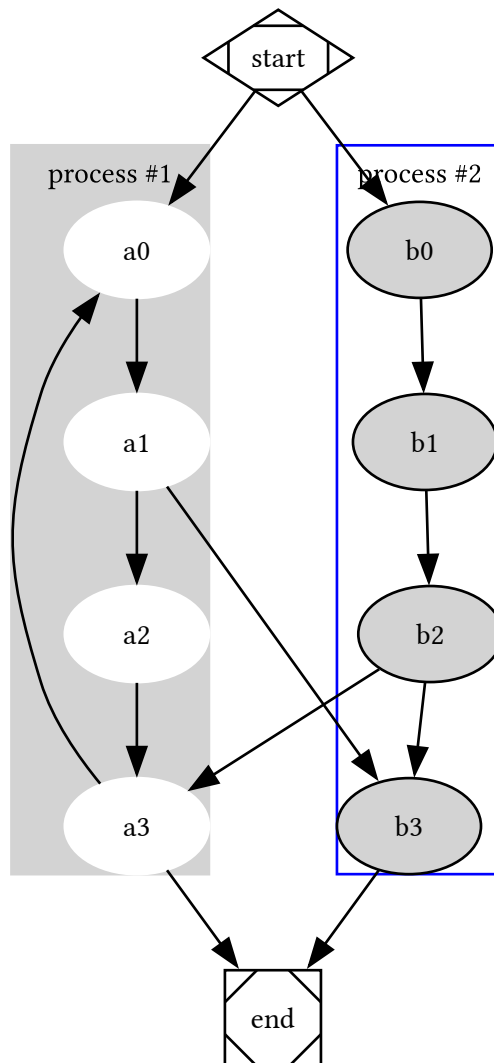
```

```

start [shape=Mdiamond];
end [shape=Msquare];

```

}



Graph 6: HTML

```

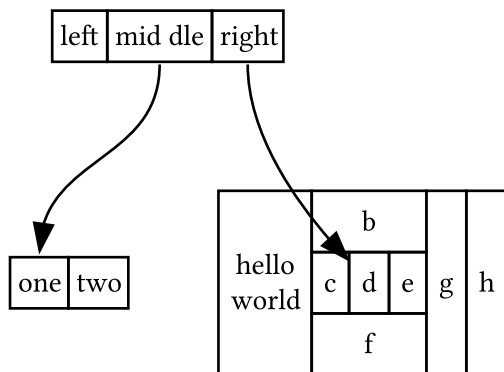
digraph structs {
    node [shape=plaintext]
    struct1 [label=<
<TABLE BORDER="0" CELLBORDER="1" CELSPACING="0">
  <TR><TD>left</TD><TD PORT="f1">mid dle</TD><TD PORT="f2">right</TD></TR>
</TABLE>>];
    struct2 [label=<
<TABLE BORDER="0" CELLBORDER="1" CELSPACING="0">
  <TR><TD PORT="f0">one</TD>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

```

```

<TR>
  <TD>c</TD><TD PORT="here">d</TD><TD>e</TD>
</TR>
<TR>
  <TD COLSPAN="3">f</TD>
</TR>
</TABLE>>];
  struct1:f1 -> struct2:f0;
  struct1:f2 -> struct3:here;
}

```



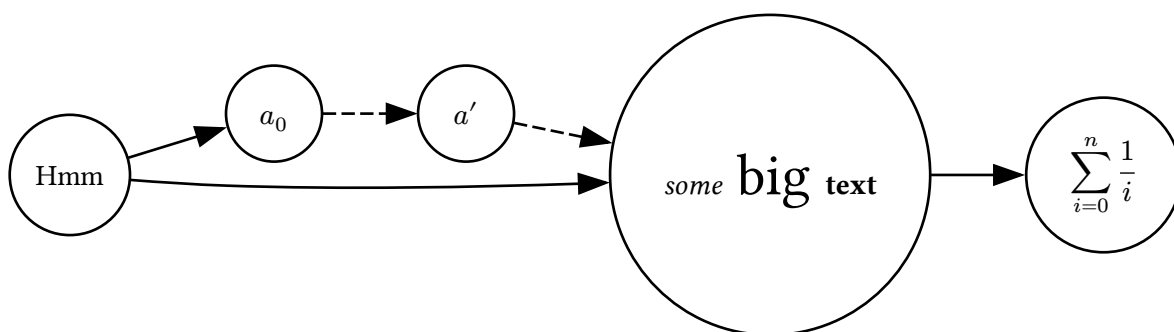
Graph 7: Overridden labels

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 8: Automatic math labels

```
digraph {  
  a -> alpha  
  phi -> rho  
  rho -> a  
  tau -> omega  
  phi -> a_8  
  a_8 -> alpha  
  a_8 -> omega  
  alpha_8 -> omega  
}
```

