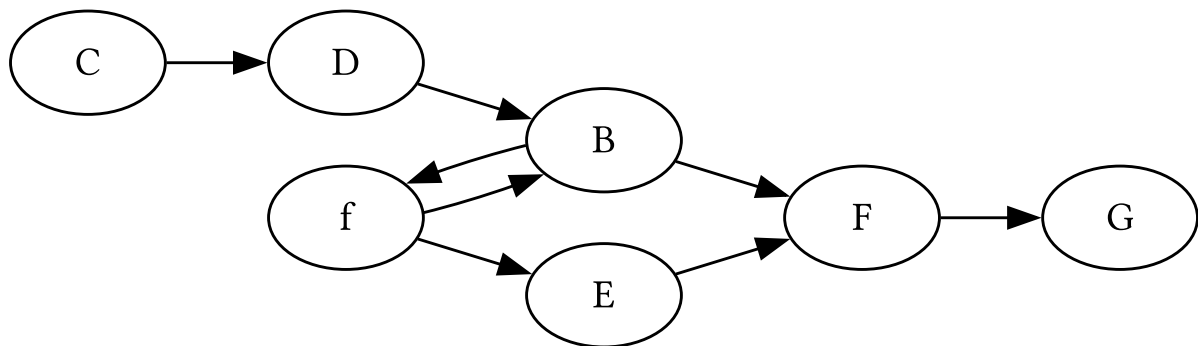


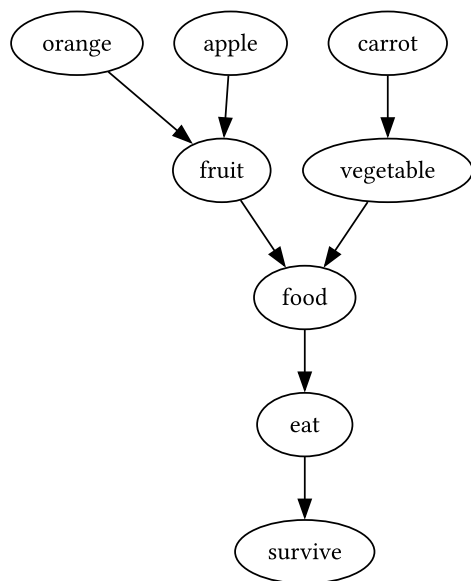
Graph 1: Test

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



Graph 2: Eating

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

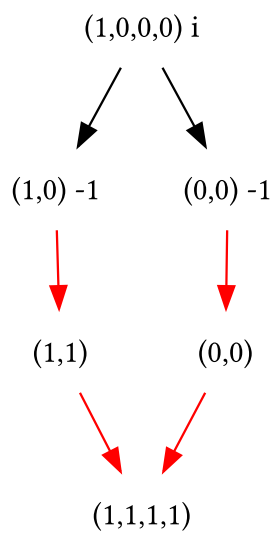


Graph 3: FFT

```

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

```



Graph 4: State Machine

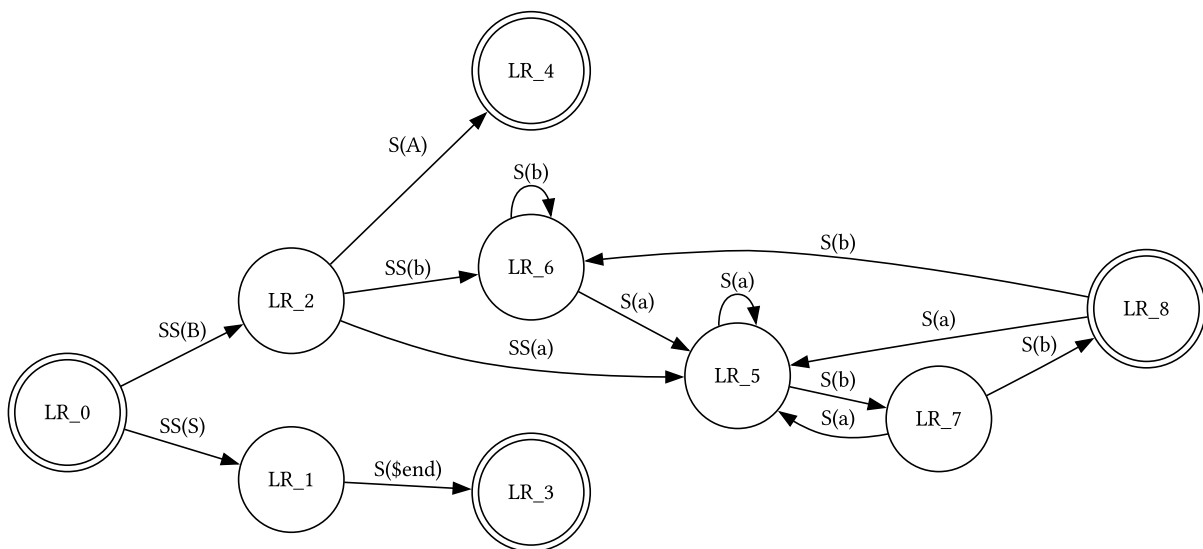
```

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 -> 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

<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];
    }
}

```

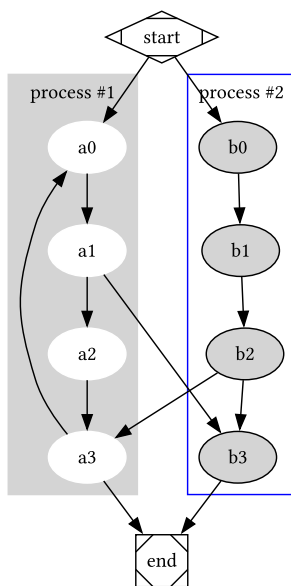
```

    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" 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;
}

```

