

# Generating Control-Flow Graph using Open-Source S/W

- GCC: version 4.8+ (recommend 5.x+)
- Graphviz (for processing .dot file)

# Get the target .c file

- Sample .c file that will be used: sample.c

```
1 #include <stdio.h>
2
3 int foo(int a) {
4     if (a > 0) return -a;
5     return a;
6 }
7
8 int main(int argc, char *argv[]) {
9     int b = 3;
10    if (foo(b) > 3) {
11        printf("Large\n");
12    } else {
13        printf("Small\n");
14    }
15    return 0;
16 }
```

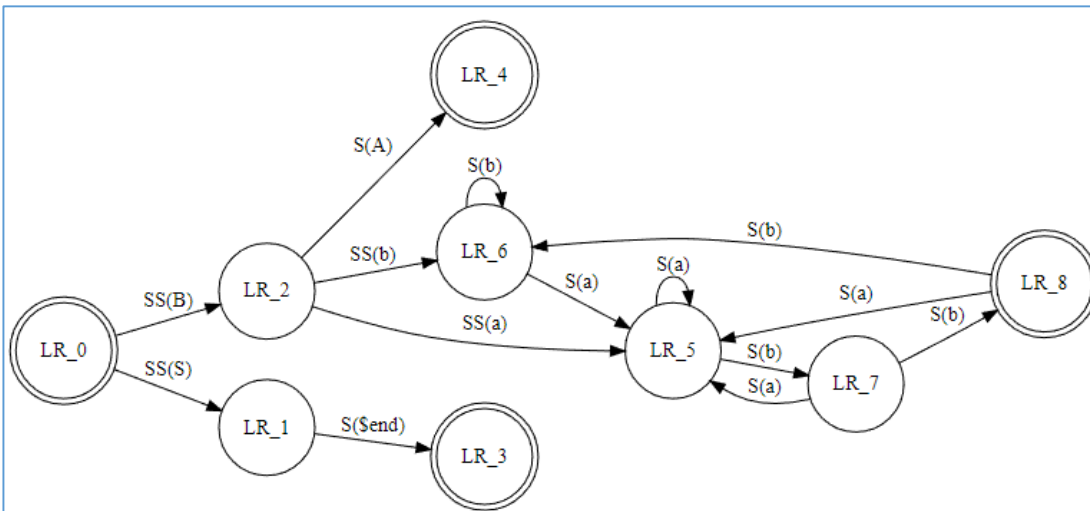
# Generate .dot file using gcc

- \$ gcc -fdump-tree-all-graph <target.c>
  - Or \$ gcc -fdump-tree-all-graph-lineno <target.c>
  - For more detailed options,  
<https://gcc.gnu.org/onlinedocs/gcc/Developer-Options.html>

```
lim@ubuntu:~/sample$ gcc -fdump-tree-all-graph sample.c
lim@ubuntu:~/sample$ l
a.out*
sample.c
sample.c.001t.tu
sample.c.002t.class
sample.c.003t.original
sample.c.004t.gimple
sample.c.006t.omplower
sample.c.006t.omplower.dot
sample.c.007t.lower
sample.c.007t.lower.dot
sample.c.010t.eh
sample.c.010t.eh.dot
sample.c.011t.cfg
sample.c.011t.cfg.dot
sample.c.012t.ompexp
sample.c.012t.ompexp.dot
sample.c.017t.fixup_cfg1
sample.c.017t.fixup_cfg1.dot
sample.c.018t.ssa
sample.c.018t.ssa.dot
sample.c.025t.fixup_cfg3
sample.c.025t.fixup_cfg3.dot
sample.c.026t.inline_param1
sample.c.026t.inline_param1.dot
sample.c.027t.einline
sample.c.027t.einline.dot
sample.c.042t.profile_estimate
sample.c.042t.profile_estimate.dot
sample.c.045t.release_ssa
sample.c.045t.release_ssa.dot
sample.c.046t.inline_param2
sample.c.046t.inline_param2.dot
sample.c.068t.fixup_cfg4
sample.c.068t.fixup_cfg4.dot
sample.c.183t.veclower
sample.c.183t.veclower.dot
sample.c.184t.cplxlower0
sample.c.184t.cplxlower0.dot
sample.c.191t.optimized
sample.c.191t.optimized.dot
sample.c.271t.statistics
sample.c.271t.statistics.dot
```

# DOT (graph description language)

- DOT is a plain text graph description language ( \*.gv or \*.dot)
- The DOT language defines a graph, but not layout of the graph.
  - Graphviz –libraries and utilities to manipulate graphs
  - <http://www.webgraphviz.com/>
  - [https://commons.wikimedia.org/wiki/Category:Images\\_with\\_Dot\\_source\\_code](https://commons.wikimedia.org/wiki/Category:Images_with_Dot_source_code)



```
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)" ];
}
```

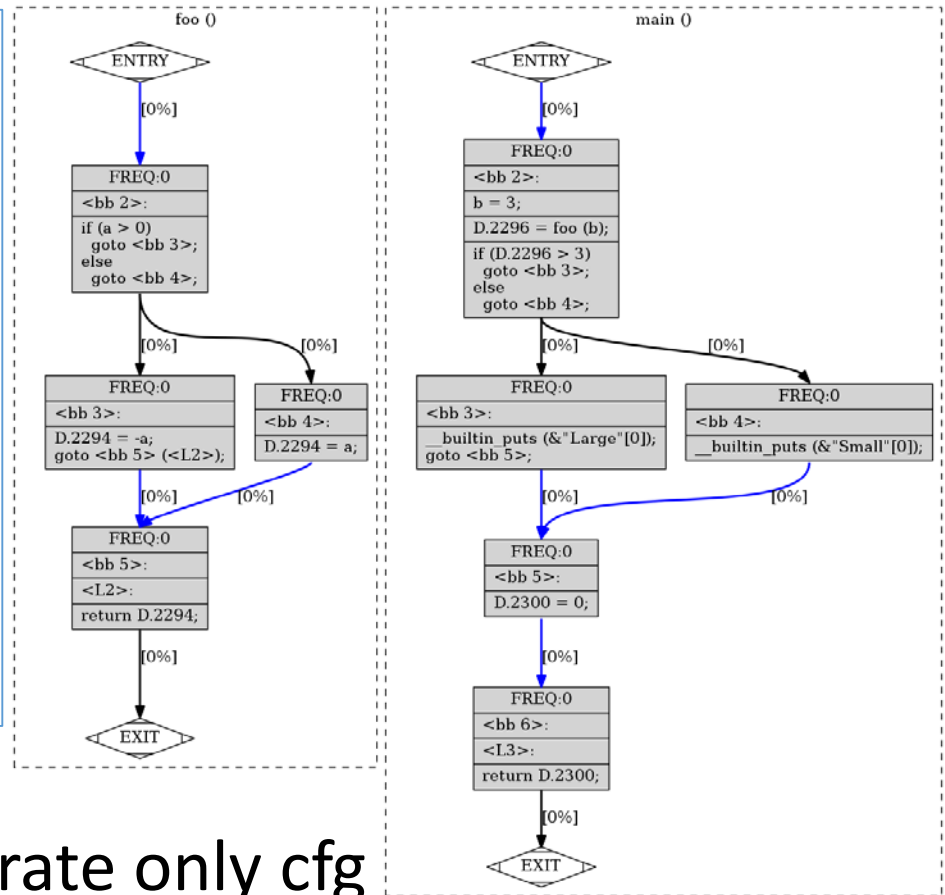
# Process .dot file using Graphviz

- `$ dot -Tpng <target.dot> -o <output.png>`
- `<target.dot>` file is the file that ends with `.cfg.dot`
  - In sample case, it is ``sample.c.011t.cfg.dot``

```
lim@ubuntu:~/sample$ dot -Tpng sample.c.011t.cfg.dot -o cfg.png
lim@ubuntu:~/sample$ l
a.out*                sample.c.025t.fixup_cfg3.dot
cfg.png               sample.c.026t.inline_param1
sample.c              sample.c.026t.inline_param1.dot
sample.c.001t.tu      sample.c.027t.einline
sample.c.002t.class   sample.c.027t.einline.dot
```

# Result graph image

```
1 #include <stdio.h>
2
3 int foo(int a) {
4     if (a > 0) return -a;
5     return a;
6 }
7
8 int main(int argc, char *argv[]) {
9     int b = 3;
10    if (foo(b) > 3) {
11        printf("Large\n");
12    } else {
13        printf("Small\n");
14    }
15    return 0;
16 }
```



- Warning: Gcc 4.x may generate only cfg of the last function in a taraget C file