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

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

The original slides were made by Hyunsu Lim bookman01@kaist.ac.kr

Get the target .c file

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

```c
#include <stdio.h>

int foo(int a) {
  if (a > 0) return -a;
  return a;
}

int main(int argc, char *argv[]) {
  int b = 3;
  if (foo(b) > 3) {
    printf("Large\n");
  } else {
    printf("Small\n");
  }
  return 0;
}
```

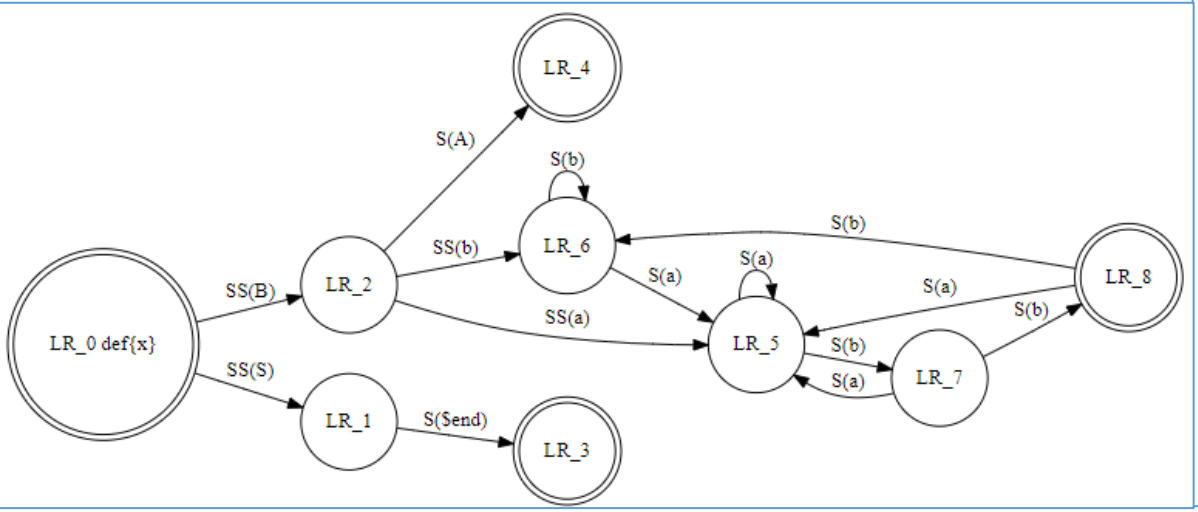
Generate .dot file using gcc

- $ 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

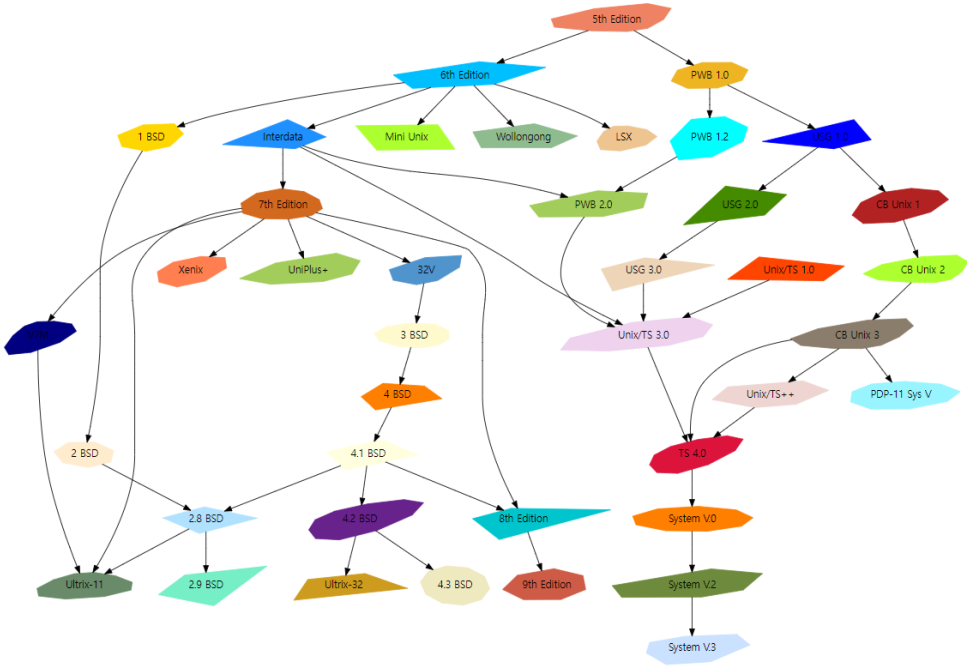
```
lim@ubuntulim:~/sample$ gcc -fdump-tree-all-graph sample.c
lim@ubuntulim:~/sample$ l
a.out*                          sample.c.025t.fixup_cfg3.dot
sample.c                        sample.c.026t.inline_param1
sample.c.001t.tu                sample.c.026t.inline_param1.dot
sample.c.002t.class       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

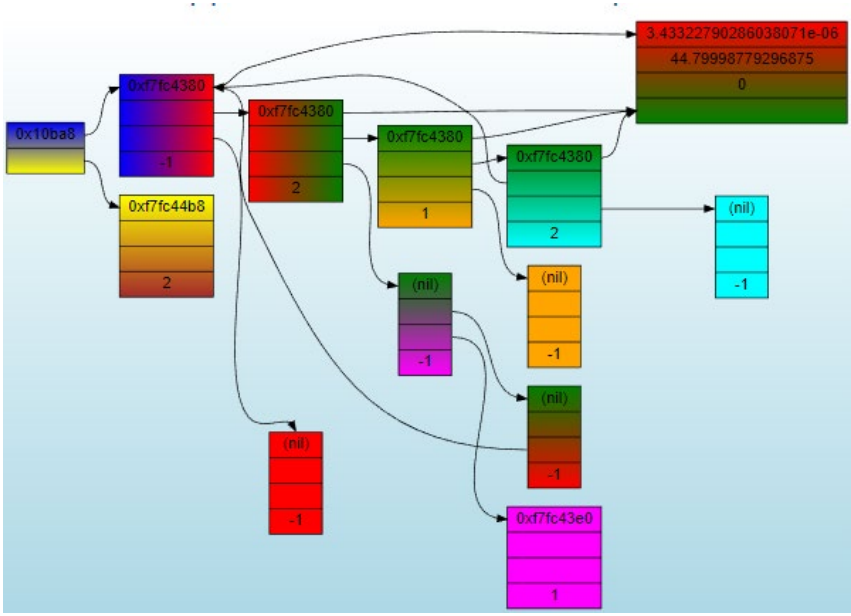
DOT (graph description language)

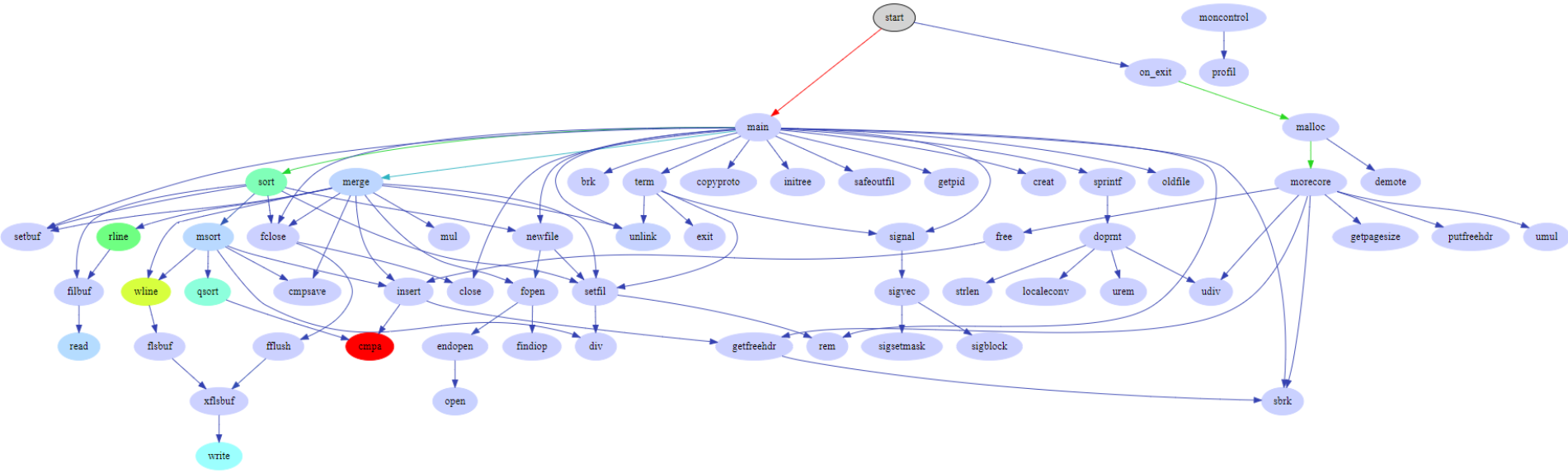
- DOT is a plain text graph description language (*.gv or * . dot)
- The DOT language defines a graph, but not layout of the graph.
  - Graphviz –libraries and utilities to manipulate graphs
  - http://www.webgraphviz.com/
  - https://commons.wikimedia.org/wiki/Category:Images_with_Dot_source_code




```
digraph finite_state_machine {
  rankdir=LR;
  size="8,5"
  node [shape = doublecircle];
  “LR_0 def{x}” LR_3 LR_4 LR_8;
  node [shape = circle];
 "LR_0 def{x}" -> LR_2 [ label ="SS(B)" ];
 "LR_0 def{x}" -> 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)" ];
}
```


5th Edition
6th Edition
PWB 1.0
1 BSD
Interdata
Mini Unix
Wollongong
LSX
PWB 1.2
USG 1.0
7th Edition
PWB 2.0
USG 2.0
CB Unix 1
Xenix
UniPlus+
32V
USG 3.0
Unix/TS 1.0
CB Unix 2
3 BSD
Unix/TS 3.0
CB Unix 3
4 BSD
Unix/TS++
PDP-11 Sys V
2 BSD
4.1 BSD
TS 4.0
2.8 BSD
4.2 BSD
8th Edition
System V.0
Ultrix-11
2.9 BSD
Ultrix-32
4.3 BSD
9th Edition
System V.2
System V.3


3.43322790286038071e-06
44.7999877929687
0
0x10ba8
0xf7fc4380
-1
0xf7fc44b8
2
0xf7fc4380
2
0xf7fc4380
1
0xf7fc4380
2
(nil)
-1
(nil)
-1
(nil)
-1
(nil)
-1
(nil)
-1
0xf7fc43e0
1


start
moncontrol
profil
on_exit
main
malloc
sort
merge
brk
term
copyproto
initree
safeoutfil
getpid
creat
sprintf
oldfile
morecore
demote
setbuf
rline
msort
fclose
mul
newfile
unlink
exit
signal
free
doprnt
getpagesize
putfreehdr
umul
filbuf
wline
qsort
cmpsave
insert
close
fopen
setfil
sigvec
strlen
localeconv
urem
udiv
read
flsbuf
fflush
cmpa
endopen
findiop
div
getfreehdr
rem
sigsetmask
sigblock
sbrk
xflsbuf
open
write

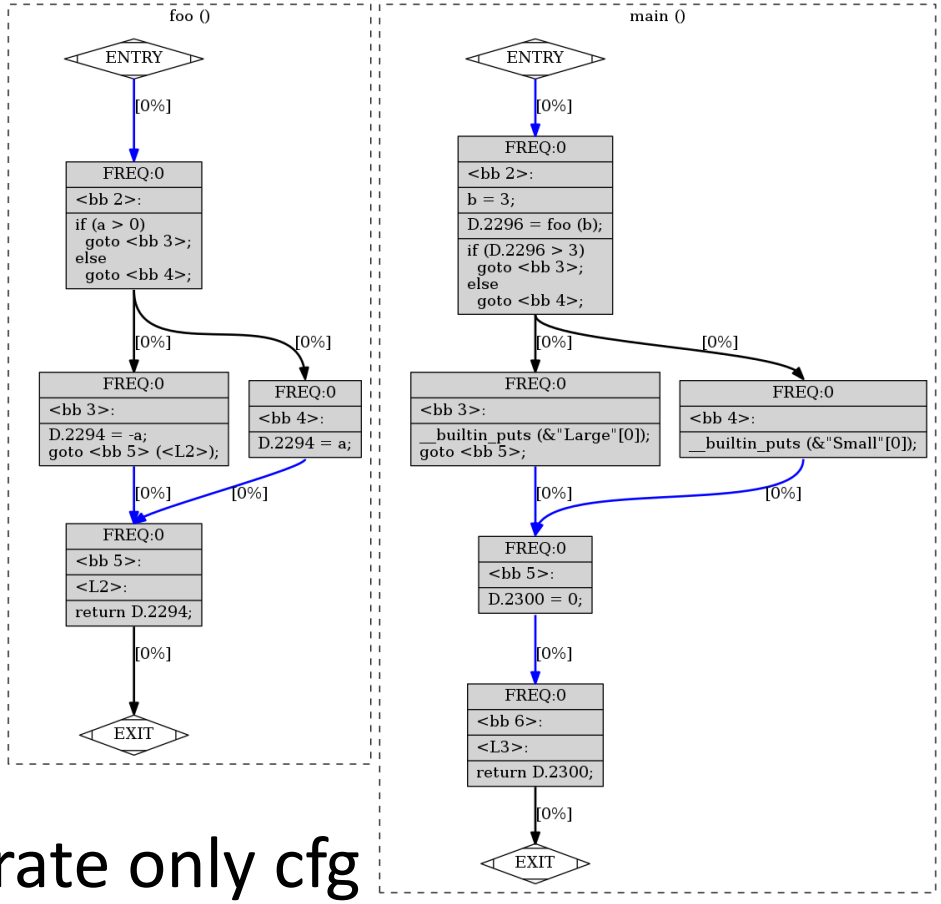
Process .dot file using Graphviz

- You can use http://www.webgraphviz.com/ to generate visual graph from .dot file.
- Or
  - $ 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@ubuntulim:~/sample$ dot -Tpng sample.c.011t.cfg.dot -o cfg.png
lim@ubuntulim:~/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 target C file