R13943151 黃鈺翔

Part1

(a) The structure is written in comp.blif.

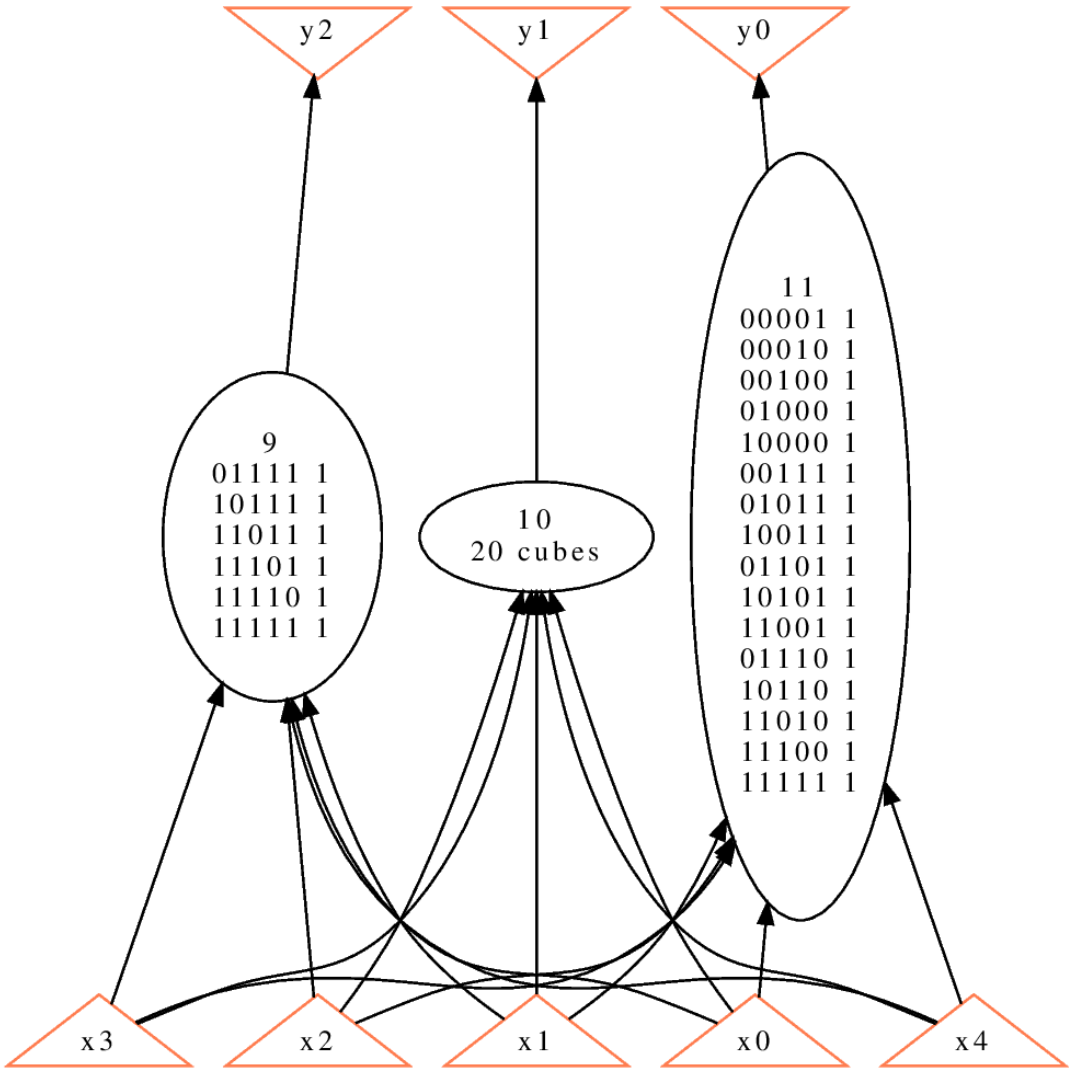
(b) Execution result :

```
abc 10> read ./comp.blif
abc 11> print_stats
comp                       : i/o =    5/    3  lat =    0  nd =     3  edge =     15  cube =    42  lev = 1
abc 11> show
```

Result of “show” in step 3,

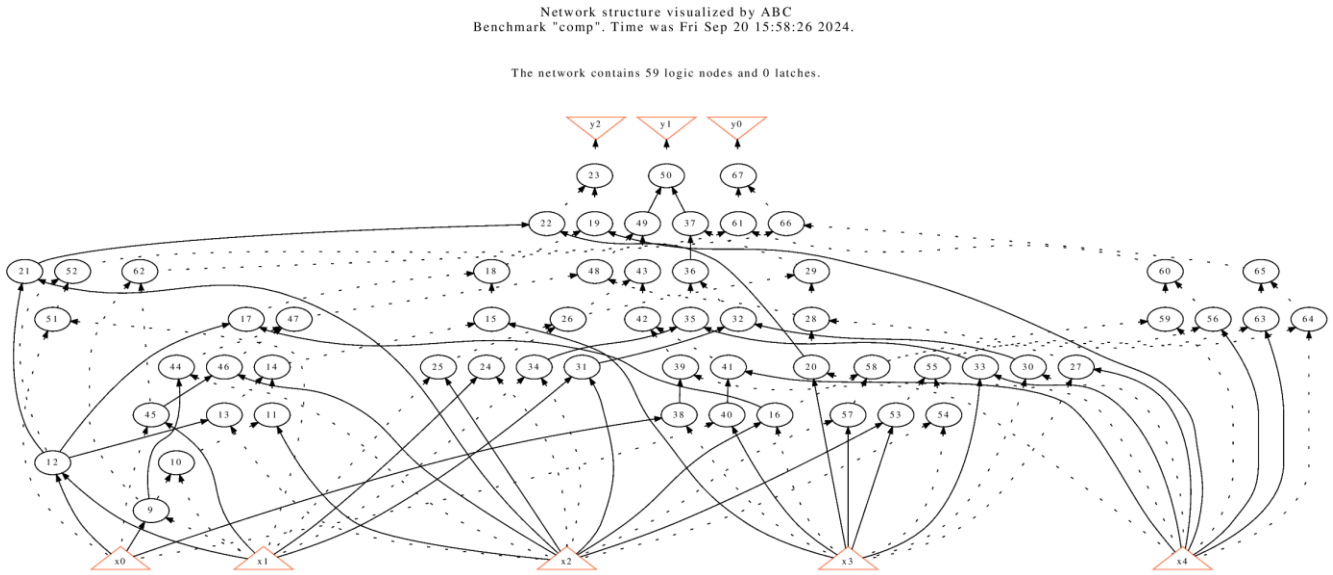
Network structure visualized by ABC
Benchmark "comp". Time was Fri Sep 20 15:54:21 2024.

The network contains 3 logic nodes and 0 latches.



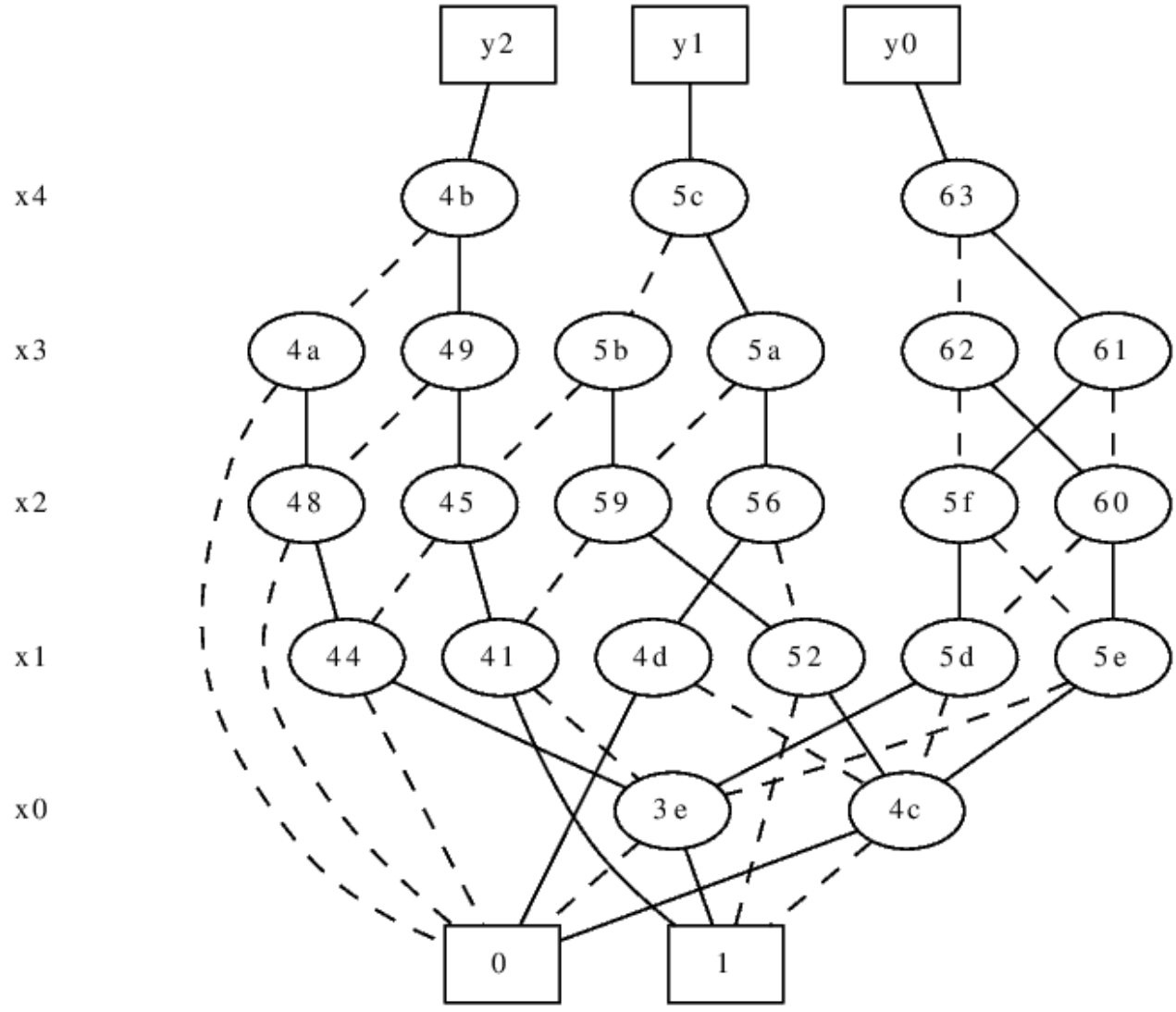
```
abc 03> strash
abc 04> show
```

Result of “show” in step 5,




```
abc 04> collapse
abc 05> show_bdd -g
```

Result of “show_bdd -g” in step 7,



Part2

(a)

1.

Both commands convert the current network into AIG. Command “aig” converts local functions of the nodes to AIGs. Its network from command “show” is still the single-output-cover form from the blif file. Command “strash” would transforms the current network into an AIG by one-level structural hashing; the resulting logic network is composed of two-input AND gates.

After command “aig,” the level of the network remains the same, whereas the network after command “strash” become 8.

```
abc 06> read ./comp.blif
abc 07> aig
abc 07> print_stats
comp                        : i/o =    5/    3  lat =    0  nd =     3  edge =     15  aig  =    65  lev = 1
abc 07> strash
abc 08> print_stats
comp                        : i/o =    5/    3  lat =    0  and =     59  lev =  8
```

2.

Command “bdd” Converts local functions of the nodes to BDDs. After command “aig,” the statistics shows that cubes are converts to bdds.

Command “collapse” recursively composes the fanin nodes into the fanout nodes resulting in a network, in which each CO is produced by a node, whose fanins are CIs. The command will eliminate the redundancy and simplify the bdd structure.

```
abc 08> read ./comp.blif
abc 09> aig
abc 09> bdd
abc 09> print_stats
comp                        : i/o =    5/    3  lat =    0  nd =     3  edge =     15  bdd  =    21  lev = 1
abc 09> collapse
abc 10> print_stats
comp                        : i/o =    5/    3  lat =    0  nd =     3  edge =     15  bdd  =    21  lev = 1
```

The print_stats of both structure are the same for comp.blif . However, the result might be different if using different *.blif (e.g. adder.blif).

```
abc 13> read ./lsv/pa1/benchmarks/adder.blif
abc 14> aig
abc 14> bdd
abc 14> print_stats
top                         : i/o =  256/    1  lat =    0  nd =  1020  edge =   2040  bdd  =  2040  lev = 255
abc 14> collapse
abc 15> print_stats
top                         : i/o =  256/    1  lat =    0  nd =     1  edge =    110  bdd  =   163  lev = 1
```

(b) Command “logic” can transform the AIG into a logic network with the SOP representation of the two-input AND-gates.

```
abc 03> read ./comp.blif
abc 04> strash
abc 05> logic
abc 06> print_stats
comp                        : i/o =    5/    3  lat =    0  nd =    59  edge =    118  cube =    59  lev = 8
abc 06> show
```

Network structure visualized by ABC
Benchmark "comp". Time was Fri Sep 20 16:01:59 2024.

The network contains 59 logic nodes and 0 latches.