ECE1733 Topics in Switching Theory Assignment #3

DPLL SAT Solver

Name: Shaoyu Chen

ID: 1002831091

Name: Zhehui Zhou

ID: 997716761

**Introduction**

The program reads a BLIF file that contains a CNF function,

**Execution Instruction**

1. Unzip the **1773\_A3.zip** file, you can find a folder called **1773\_A3**. In the folder there is a sub-folder called **Release** which contains the executable file called **DPLL.exe**.
2. Open the windows command shell **cmd.exe**, and cd to the folder contains the **DPLL.exe** executable file described in step 1.
3. Type: “**DPLL**.exe <path to the BLIF test file>”, you can find some test files in the **test\_nodes** folder within the **1773\_A3** folder.
4. The program should start, and shows the major steps it used to find the satisfiability of the CNF function.

**Execution Report**

Please see Appendix for the detailed information printed on the screen when the program is executed.

**Discussion**

This program prints every major step it took during the execution to the screen, which makes the debugging process easier.

**Appendix**

Detailed Execution Report, I used an Git BASH Shell to execute the program.

node1.blif:

$ ./assign1.exe ../test\_nodes/node1.blif

start!!!

Quine-McCluskey 2-level logic minimization program.

Reading file ../test\_nodes/node1.blif...

Pre-Execution Report:

Function 1: #inputs = 3; #cubes = 5; cost = 26

The cubical function contains:

cube#0: 001

cube#1: 100

cube#2: 111

cube#3: 010

cube#4: 101

Minimizing logic functions

Arrange the cubes by the cardinality of 1's:

001

100

010

101

111

Combined Cubes:

x01

10x

1x1

Combined Cubes:

Initial Table:

0: 001 1: 100 2: 111 3: 010 4: 101

0 1 2 3 4

010 1

x01 1 1

10x 1 1

1x1 1 1

After removing essential PI from table:

New cubical function:

cube#0: 010

cube#1: x01

cube#2: 10x

cube#3: 1x1

Post-Execution Report:

Function 1: #inputs = 3; #cubes = 4; cost = 18

Done.

node2.blif

$ ./assign1.exe ../test\_nodes/node2.blif

start!!!

Quine-McCluskey 2-level logic minimization program.

Reading file ../test\_nodes/node2.blif...

Pre-Execution Report:

Function 1: #inputs = 4; #cubes = 6; cost = 33

The cubical function contains:

cube#0: 00x0

cube#1: 100x

cube#2: 1010

cube#3: 1111

cube#4: 00x1

cube#5: 011x

Minimizing logic functions

Arrange the cubes by the cardinality of 1's:

00x0

100x

00x1

1010

011x

1111

Combined Cubes:

00xx

Combined Cubes:

Initial Table:

0: 00x0 1: 100x 2: 1010 3: 1111 4: 00x1 5: 011x

0 1 2 3 4 5

100x 1

1010 1

011x 1

1111 1

00xx 1 1

After removing essential PI from table:

New cubical function:

cube#0: 100x

cube#1: 1010

cube#2: 011x

cube#3: 1111

cube#4: 00xx

Post-Execution Report:

Function 1: #inputs = 4; #cubes = 5; cost = 27

Done.

node3.blif

$ ./assign1.exe ../test\_nodes/node3.blif

start!!!

Quine-McCluskey 2-level logic minimization program.

Reading file ../test\_nodes/node3.blif...

Pre-Execution Report:

Function 1: #inputs = 5; #cubes = 22; cost = 155

The cubical function contains:

cube#0: 00001

cube#1: 00100

cube#2: 00110

cube#3: 00111

cube#4: 01001

cube#5: 01010

cube#6: 01100

cube#7: 01111

cube#8: 10001

cube#9: 10011

cube#10: 10100

cube#11: 10111

cube#12: 11001

cube#13: 11010

cube#14: 11011

cube#15: 11100

cube#16: 11110

cube#17: 11111

cube#18: 01000

cube#19: 10000

cube#20: 10101

cube#21: 10110

Minimizing logic functions

Arrange the cubes by the cardinality of 1's:

00001

00100

01000

10000

00110

01001

01010

01100

10001

10100

00111

10011

11001

11010

11100

10101

10110

01111

10111

11011

11110

11111

Combined Cubes:

0x001

x0001

001x0

0x100

x0100

0100x

010x0

01x00

1000x

10x00

0011x

x0110

x1001

x1010

x1100

100x1

1x001

10x01

1x100

1010x

101x0

0x111

x0111

10x11

1x011

110x1

1101x

11x10

111x0

101x1

1011x

1x110

x1111

1x111

11x11

1111x

Combined Cubes:

xx001

xx001

x01x0

xx100

x01x0

xx100

10x0x

10x0x

x011x

x011x

1x0x1

10xx1

1x0x1

10xx1

1x1x0

101xx

1x1x0

101xx

xx111

xx111

1xx11

1xx11

11x1x

11x1x

1x11x

1x11x

Combined Cubes:

Initial Table:

0: 00001 1: 00100 2: 00110 3: 00111 4: 01001 5: 01010 6: 01100 7: 01111 8: 10001 9: 10011 a: 10100 b: 10111 c: 11001 d: 11010 e: 11011 f: 11100 10: 11110 11: 11111 12: 01000 13: 10000 14: 10101 15: 10110

0 1 2 3 4 5 6 7 8 9 A B C D E F 10 11 12 13 14 15

0100x 1 1

010x0 1 1

01x00 1 1

x1010 1 1

xx001 1 1 1 1

x01x0 1 1 1 1

xx100 1 1 1 1

10x0x 1 1 1 1

x011x 1 1 1 1

1x0x1 1 1 1 1

10xx1 1 1 1 1

1x1x0 1 1 1 1

101xx 1 1 1 1

xx111 1 1 1 1

1xx11 1 1 1 1

11x1x 1 1 1 1

1x11x 1 1 1 1

After removing essential PI from table:

0: 00100 1: 00110 2: 01010 3: 01100 4: 10011 5: 11010 6: 11011 7: 11100 8: 11110 9: 01000 a: 10110

0 1 2 3 4 5 6 7 8 9 A

0100x 1

010x0 1 1

01x00 1 1

x1010 1 1

x01x0 1 1 1

xx100 1 1 1

x011x 1 1

1x0x1 1 1

10xx1 1

1x1x0 1 1 1

101xx 1

1xx11 1 1

11x1x 1 1 1

1x11x 1 1

After removing dominated PIs from table:

0: 00100 1: 00110 2: 01010 3: 01100 4: 10011 5: 11010 6: 11011 7: 11100 8: 11110 9: 01000 a: 10110

0 1 2 3 4 5 6 7 8 9 A

010x0 1 1

01x00 1 1

x1010 1 1

x01x0 1 1 1

xx100 1 1 1

1x0x1 1 1

1x1x0 1 1 1

1xx11 1 1

11x1x 1 1 1

After removing essential PI from table:

0: 01010 1: 01100 2: 10011 3: 11010 4: 11011 5: 11100 6: 11110 7: 01000

0 1 2 3 4 5 6 7

010x0 1 1

01x00 1 1

x1010 1 1

xx100 1 1

1x0x1 1 1

1x1x0 1 1

1xx11 1 1

11x1x 1 1 1

After removing the least cost PIs from table:

0: 01010 1: 10011 2: 11010 3: 11011 4: 11110 5: 01000

0 1 2 3 4 5

010x0 1 1

01x00 1

x1010 1 1

1x0x1 1 1

1x1x0 1

1xx11 1 1

11x1x 1 1 1

After removing dominated PIs from table:

0: 01010 1: 10011 2: 11010 3: 11011 4: 11110 5: 01000

0 1 2 3 4 5

010x0 1 1

x1010 1 1

1x0x1 1 1

1xx11 1 1

11x1x 1 1 1

After removing essential PI from table:

0: 10011

0

x1010

1x0x1 1

1xx11 1

After removing dominated PIs from table:

0: 10011

0

1x0x1 1

1xx11 1

After removing the least cost PIs from table:

1xx11

New cubical function:

cube#0: xx001

cube#1: 10x0x

cube#2: xx111

cube#3: x01x0

cube#4: xx100

cube#5: 010x0

cube#6: 11x1x

cube#7: 1x0x1

Post-Execution Report:

Function 1: #inputs = 5; #cubes = 8; cost = 42

Done.

node4.blif

$ ./assign1.exe ../test\_nodes/node4.blif

start!!!

Quine-McCluskey 2-level logic minimization program.

Reading file ../test\_nodes/node4.blif...

Pre-Execution Report:

Function 1: #inputs = 4; #cubes = 6; cost = 32

The cubical function contains:

cube#0: 00x0

cube#1: 100x

cube#2: x010

cube#3: 1111

cube#4: 00x1

cube#5: 011x

Minimizing logic functions

Arrange the cubes by the cardinality of 1's:

00x0

100x

x010

00x1

011x

1111

Combined Cubes:

00xx

Combined Cubes:

Initial Table:

0: 00x0 1: 100x 2: x010 3: 1111 4: 00x1 5: 011x

0 1 2 3 4 5

100x 1

x010 1

011x 1

1111 1

00xx 1 1

After removing essential PI from table:

New cubical function:

cube#0: 100x

cube#1: x010

cube#2: 011x

cube#3: 1111

cube#4: 00xx

Post-Execution Report:

Function 1: #inputs = 4; #cubes = 5; cost = 26

Done.

node5.blif

$ ./assign1.exe ../test\_nodes/node5.blif

start!!!

Quine-McCluskey 2-level logic minimization program.

Reading file ../test\_nodes/node5.blif...

Pre-Execution Report:

Function 1: #inputs = 4; #cubes = 4; cost = 20

The cubical function contains:

cube#0: xx00

cube#1: 110x

cube#2: 1x11

cube#3: 10x0

Minimizing logic functions

Arrange the cubes by the cardinality of 1's:

xx00

10x0

110x

1x11

Combined Cubes:

Initial Table:

0: xx00 1: 110x 2: 1x11 3: 10x0

0 1 2 3

xx00 1

10x0 1

110x 1

1x11 1

After removing essential PI from table:

New cubical function:

cube#0: xx00

cube#1: 10x0

cube#2: 110x

cube#3: 1x11

Post-Execution Report:

Function 1: #inputs = 4; #cubes = 4; cost = 20

Done.