COT 5405

Final Exam – Take Home

Submitted by

Rajib Dey

The Problem

1. We are given a mapping of an electrical circuit where we only know which resistor is connected between which two nodes. We have to find out the truth table based on few assumption

2. We are given a truth table. We have to find out the circuit that it belongs to.

Pseudocode for the First Problem

- We sort the resistors which are connected to 0 and 1. Then we check, the other end of these resistors. If they match, that means they can form a direct path from 0 to 1 in series.

- This way we find out all the possible paths from 0 to 1, where only 2 resistors are used.

- To find out longer paths where more than 2 resistors are used, we used Breadth-First-Search or BFS.

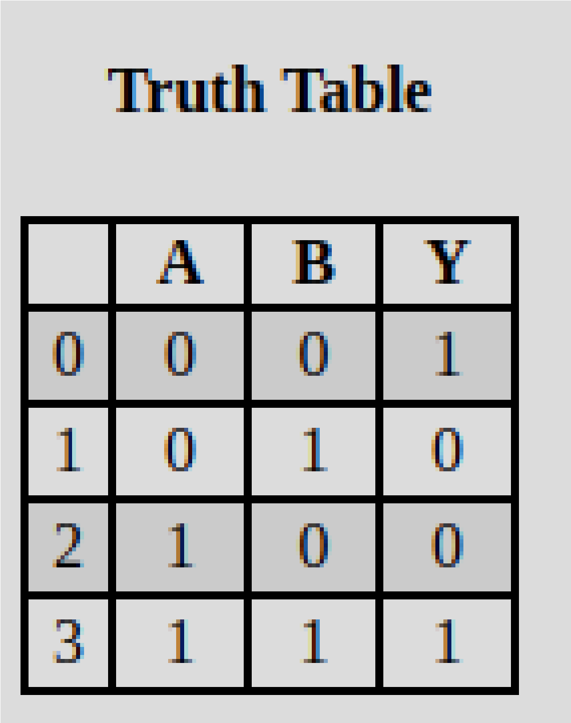
- Then while assigning the nodes those resistors will be connected to, we get the combinations of its neigbours and check if they are connected to other resistors or not. If they are not, then we connect that resistor with those nodes.

Pseudocode for the First Problem

- We also build up the equation for this path in the following manner: R1+R2+R3+…
- Then using the path and their corresponding resistor values, we build up the equation in Product of Sum (SOP) form.
- Our SOP building looks like this: path1 & path2 & path3 …

Pseudocode for the First Problem

- Then we use the SOP that we got in the previous step to get the truth table.
- As we are using maxterm to get the truth table, each of the path in our SOP would result in a 0 entry in the truth table.
- If our SOP is (A + B') (A' + B), then we know our truth table will have two zero entries at (0,1) and (1,0). Where (0,1) represents (A + B') and (1,0) represents (A' + B).
- In this case our Truth Table will look like this:
- We only need to consider where the result is 0. Then we look at how the values of A and B looks like in those cases. For 0 values, the corresponding variable will be not inverted and for 1 values, the corresponding variable will be inverted.



Truth Table

	A	B	Y
0	0	0	1
1	0	1	0
2	1	0	0
3	1	1	1

Pseudocode for the 2nd Problem

- As we already have the truth table, we do the opposite of what we did in the previous step to get the Product of Sum (POS).

- Everytime there is a 0 in the truth table, that will lead to the corresponding value to be not inverted. For each entry of 1 in the truth table the corresponding value will become inverted. And vice versa.

- After we get the POS, then we check if any of the Sum values has only two resistors. For these sum value entries, we know we can reach from 0 to 1 by these two resistors.

Pseudocode for the 2nd Problem

- For other sum value entries where we have more than 2 resistors, we assign the nodes randomly in between. This will build up a random (and probably incorrect) electrical circuit.

- To check the authenticity of the random circuit that we created, we use our first program to check if this random circuit is giving us the Truth Table that we already have.

- If the truth table from the random circuit does not match with the given truth table then we know it is a wrong circuit. Then we check for the next random circuit. This goes on till we find the circuit that gives us the provided truth table.

Source Code description

- The source code for Problem #1 resides in the folder submission_files_for_1

- To solve the first problem the file 1_circuit_To_Truth_table.py needs to be run with the input file in the specified format. The run command would be like:

```
python 1_circuit_To_Truth_table.py EXTRA1.cir
```

- This file creates a file called path.txt to enumurate all the path that can exist from 0 to 1. It uses this file to process the solution further and deletes it when this file is no longer needed.

Source Code description

- For small problems, creating this file will not take much time. But for problems where the number of resistors is high it can take hours to find all the possible paths.
- For 100 resistors it took my PC almost 5 hours to find all the possible paths. The resulting text file was almost 1 MB in size.

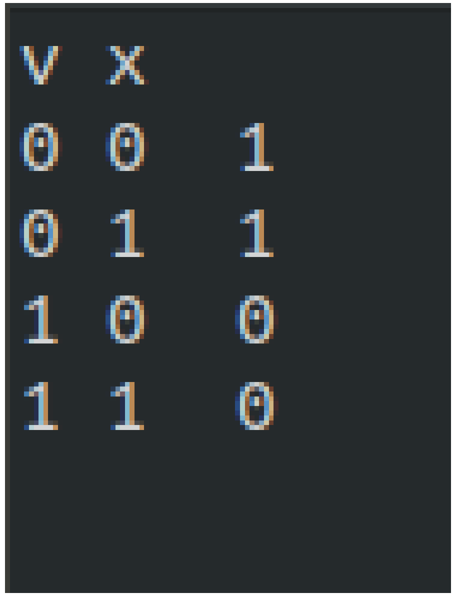
- After the program is done calculating the Truth Table, it saves it in a text file named truth-table.txt.
- Everytime the program is run, it will delete previously existing truth-table.txt in the folder.

Source Code description

- The source code for Problem #2 resides in the folder submission_files_for_2
- To solve the second problem the file 2_parse_truth.py needs to be run with the input file in the specified format. The run command would be like:
  - python 2_parse_truth.py test1.bool
- This file creates a file called circuit.txt to which is the ultimate solution of the problem.
- This file is in a format that can be used in as the input for Problem #1.
- Everytime the aforementioned python script is run, it checks for a previously generated circuit.txt file. If there is any, it deletes it to avoid any comlications.

Results For Specified Inputs

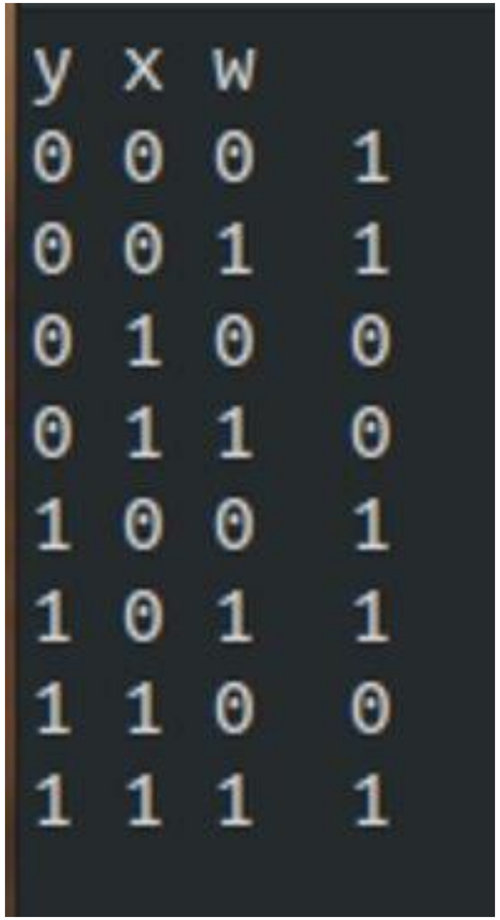
- Result for compute1:



```
v x
0 0  1
0 1  1
1 0  0
1 1  0
```

Results For Specified Inputs

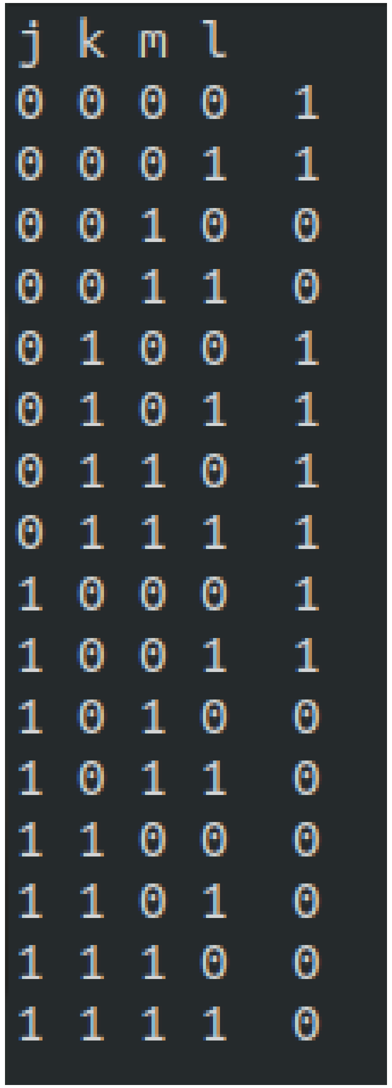
- Result for compute2:



```
y x w
0 0 0  1
0 0 1  1
0 1 0  0
0 1 1  0
1 0 0  1
1 0 1  1
1 1 0  0
1 1 1  1
```

Results For Specified Inputs

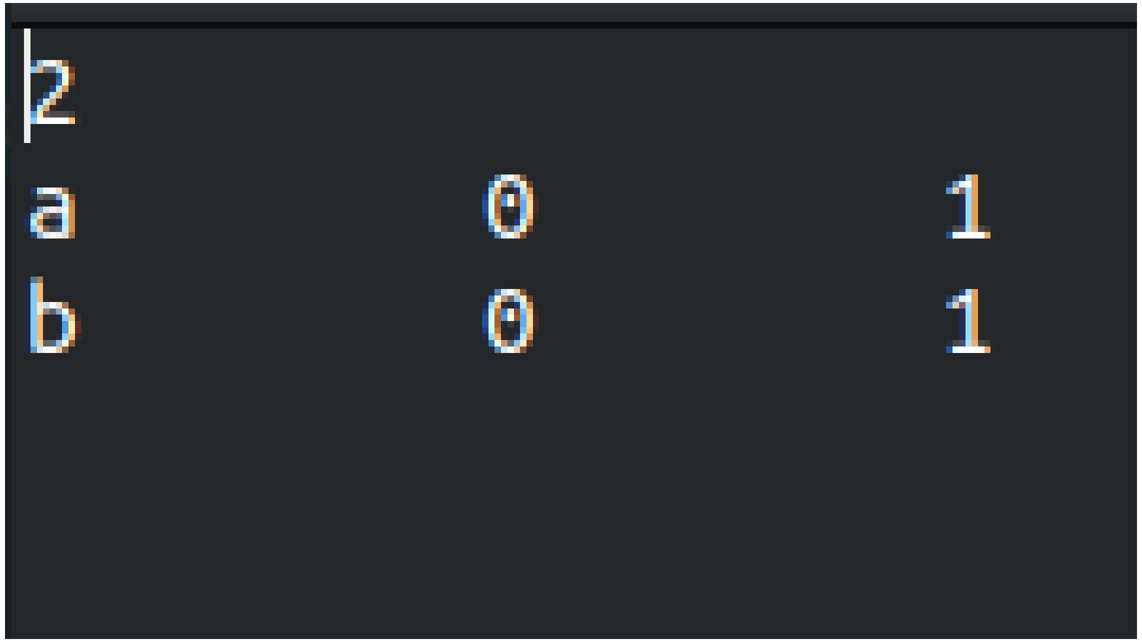
- Result for compute3:



```
j k m l
0 0 0 0  1
0 0 0 1  1
0 0 1 0  0
0 0 1 1  0
0 1 0 0  1
0 1 0 1  1
0 1 1 0  1
0 1 1 1  1
1 0 0 0  1
1 0 0 1  1
1 0 1 0  0
1 0 1 1  0
1 1 0 0  0
1 1 0 1  0
1 1 1 0  0
1 1 1 1  0
```

Results For Specified Inputs

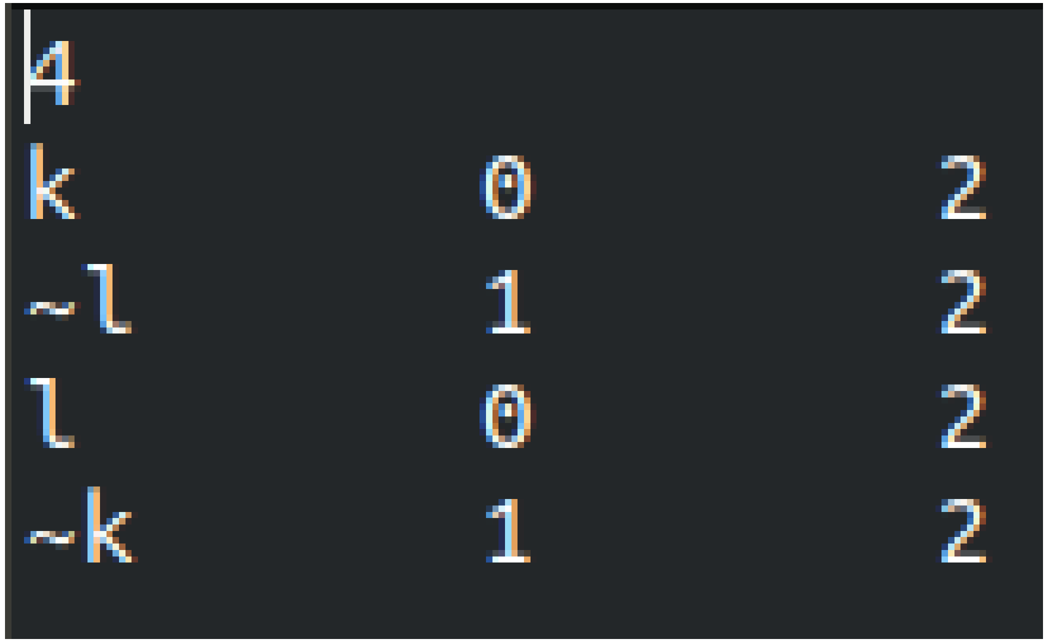
- For test1.bool



```
2
a        0        1
b        0        1
```

Results For Specified Inputs

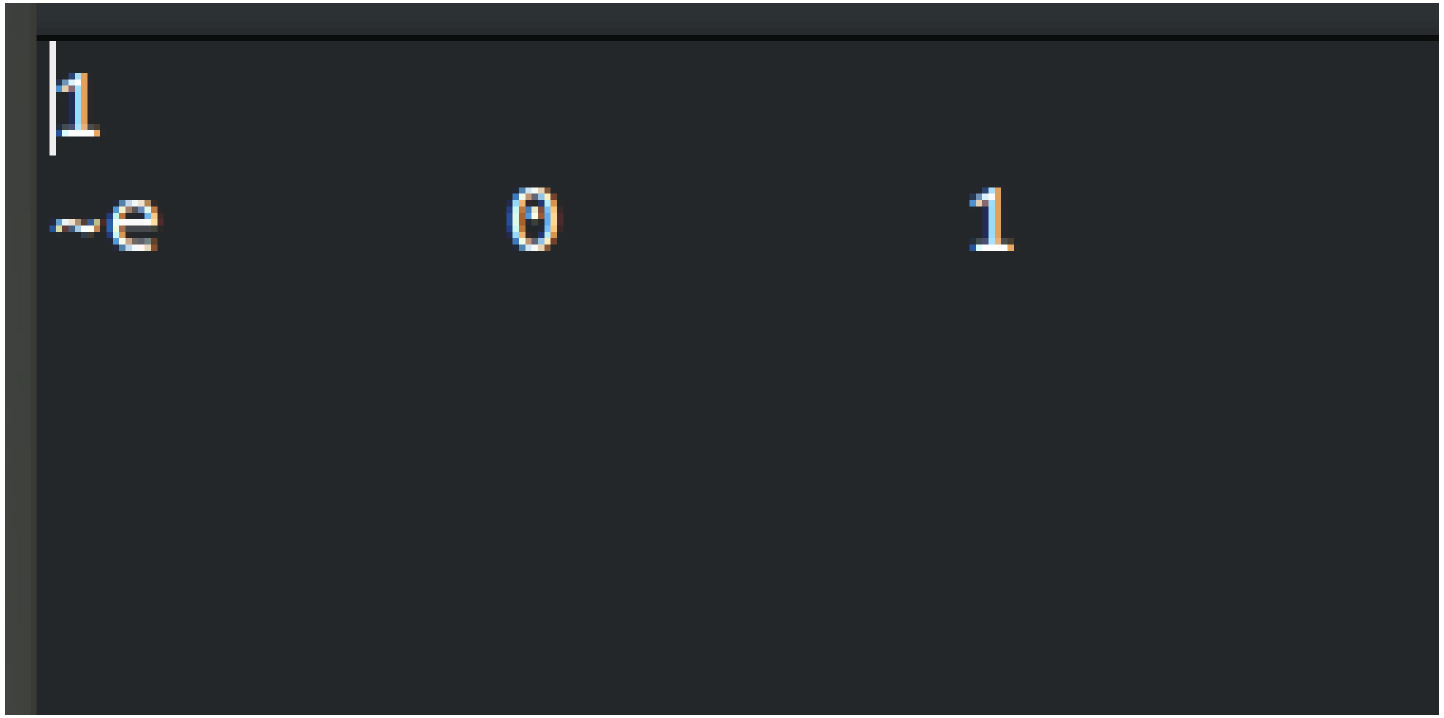
- For test2.bool



```
4
k          0          2
~l         1          2
l          0          2
~k         1          2
```

Results For Specified Inputs

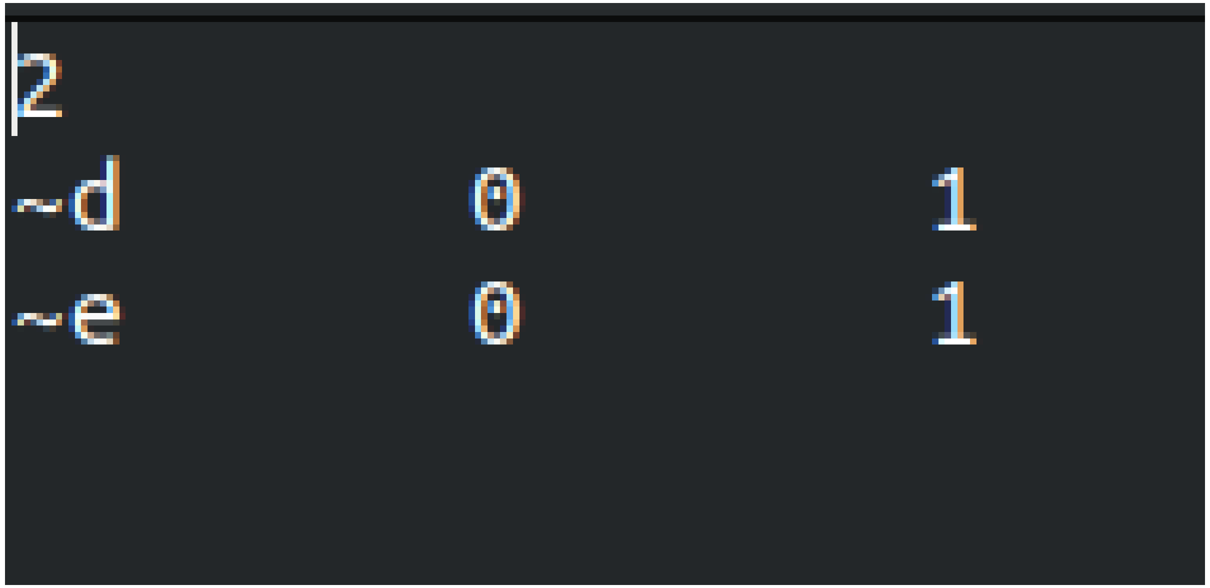
- For test3.bool



```
1
~e        0         1
```

Results For Specified Inputs

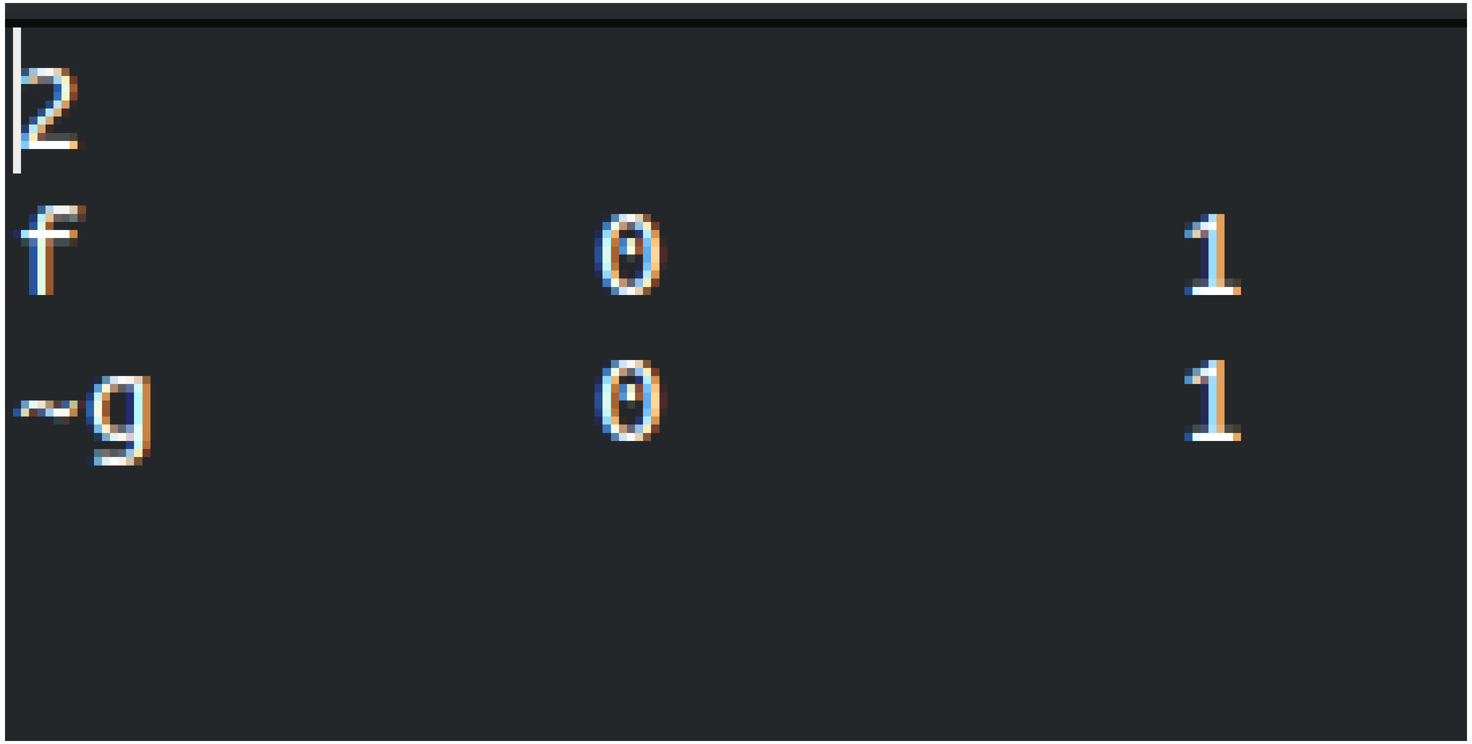
- For test4.bool



```
2
~d        0        1
~e        0        1
```

Results For Specified Inputs

- For test5.bool



```
2
f          0          1
~g         0          1
```

Results files

- All the result files are also included in both the folders