Programming Assignment 2

CMPE 250, Data Structures and Algorithms, Fall 2014

Instructor: A. T. Cemgil
TA's: Barış Kurt, Atakan Arıkan

Due: 08 November 2014, 23:59

Logic Circuit Evaluation

A logic circuit is a directed acyclic graph that is composed of input nodes, logic gates and output nodes. The edges of the graph are the wires that connect those components. In this project you are supposed to calculate the output of each node in a given logic circuit. If the circuit contains a cycle, you should return error. Figure 1 shows a circuit with 5 input nodes, and 3 output nodes and 8 logic gates. The input nodes emit either a 0 or 1 signal. The signals emitted by other nodes are calculated according to the logic rules. The signals emitted by all the nodes in Figure 1 is calculated as follows:

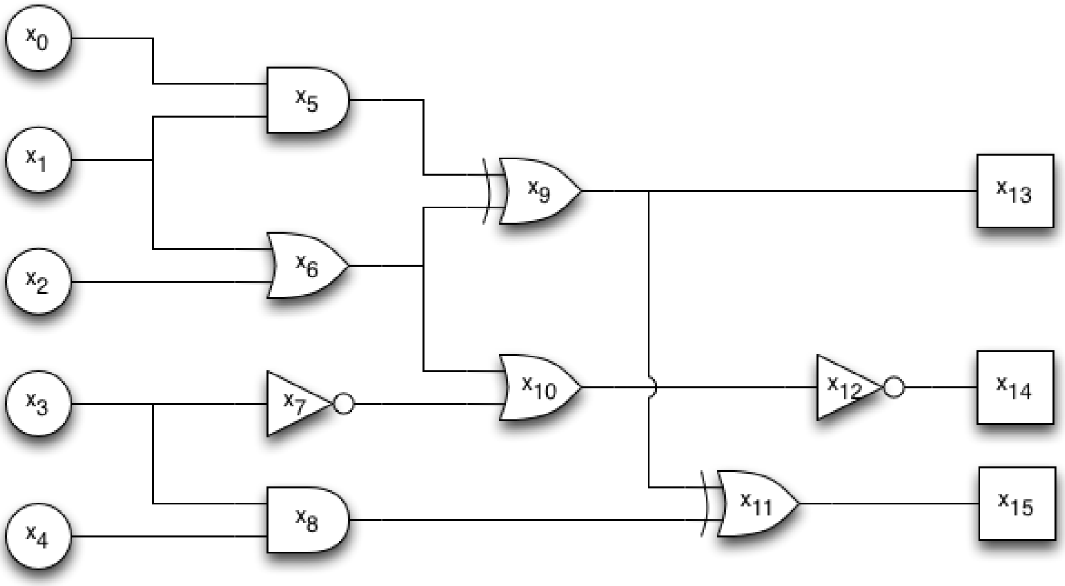

Figure 1: A logic circuit.

$$\begin{array}{lll}
x_0 = x_0 & x_5 = x_0 \wedge x_1 & x_{10} = x_6 \vee x_7 \\
x_1 = x_1 & x_6 = x_1 \vee x_2 & x_{11} = x_9 \oplus x_8 \\
x_2 = x_2 & x_7 = \neg x_3 & x_{12} = \neg x_{10} \\
x_3 = x_3 & x_8 = x_3 \wedge x_4 & x_{13} = x_9 \\
x_4 = x_4 & x_9 = x_5 \oplus x_6 & x_{14} = x_{12}, x_{15} = x_{11}
\end{array}$$

Graph Nodes

- **Input nodes** generate a signal of 1 (steady high voltage) or 0 (steady low voltage), and have at least one wire carrying their signal to some other component.
- **Output nodes** haveexactly one wire connecting to then, and they maintain the signal received from this wire.
- **Logic gates** can have one or two wires carrying signal to then, and they emit a new signal according to their functionality. There are 4 types of gates that is used in this peoject: AND($\wedge$), OR($\vee$), XOR($\oplus$) and NOT($\neg$). Their outputs are given in Table 1.

X	Y	$\neg$X	X $\wedge$ Y	X $\vee$ Y	X $\oplus$ Y
0	0	1	0	0	0
0	1	1	0	1	1
1	0	0	0	1	1
1	1	0	1	1	0

Table 1: Truth table for NOT, AND, OR and XOR logic gates.

Input / Output Format

Your program should take as input an input file name and an output file name. Your program **must** take the arguments from the command line in the following format:

```
./your_program [input_file] [output_file]
```

Input File Format

The input file contains the following:

1. The first line contains the number of nodes, N.
2. The second line contains N symbols, indicating the type of each node.
   - 0 indicates an input node emiting 0.
   - 1 indicates an input node emiting 1.
   - N indicates a NOT gate.
   - A indicates an AND gate.
   - R indicates an OR gate.
   - X indicates a XOR gate.
   - O indicates an output (the letter O, not zero).
3. The remainder of the input file contains an adjacency list for describing the graph. Each line in the adjacency list has the format "n_i n_j n_k ...", meaning that the node n_i is connected to nodes n_j, n_k, etc. The node numbers starts from zero.

Output File Format

Your program should output a text file. If the circuit contains a cycle, the text file should have one line containing only the integer -1.

Example 1:

For example, the input file for the logic circuit in Figure 1 is as follows:

```
5 8 3
1 0 1 1 A R N A X R X N O O O
0 5
1 5 6
2 6
3 7 8
4 8
5 9
6 9 10
7 10
8 11
9 11 13
10 12
11 15
12 14
```

The output file that should be generated for this example contains the following single file. The solution is also shown in Figure 2

```
1 0 1 1 0 0 1 0 0 1 1 1 0 1 0 1
```

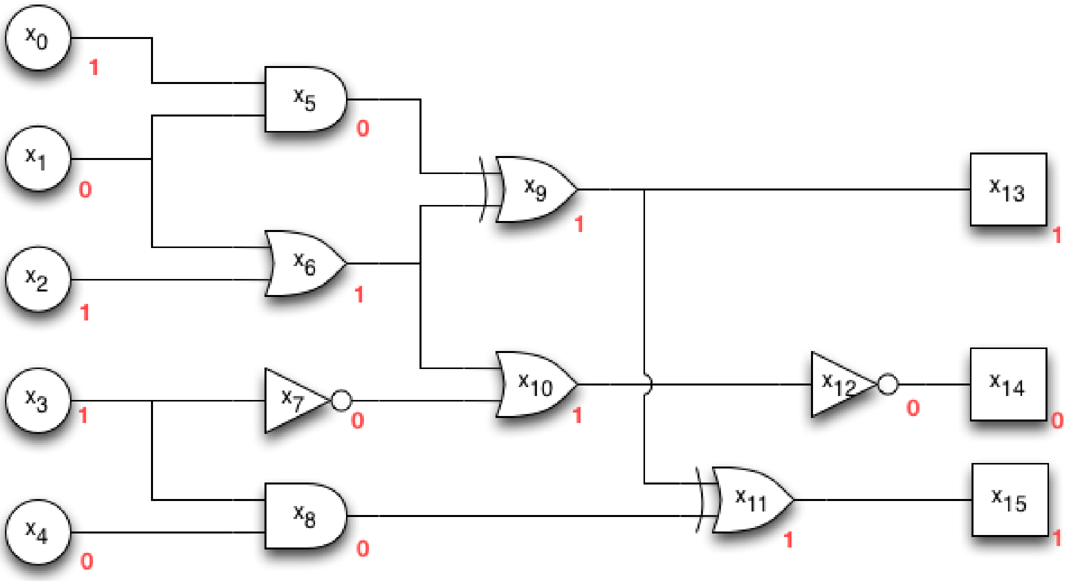

Figure 2: A logic circuit.

Example 2

This example shows an invalid circuit which does not produce an acyclic directed graph as shown in Figure 3. The input file for this example is:

```
2 2 1
1 0 A R 0
0 2
1 3
2 3 4
3 2
```

And the output file contains the following single line:

```
-1
```

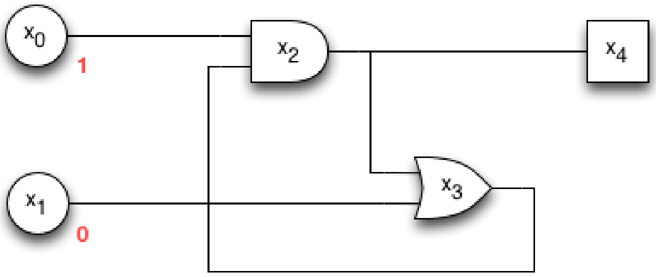

Figure 3: A logic circuit with cycle.

Submission Details

You are supposed to use the Git system provided to you for all projects. No other type of submission will be accepted. Also pay attention to the following points:

- All source codes are checked automatically for similarity with other submissions and exercises from previous years. Make sure you write and submit your own code.
- In our case, you are expected to use C++ as powerful, steady and flexible as possible. Use mechanisms that affects these issues positively.
- Make sure you document your code with necessary inline comments, and use meaningful variable names. Do not over-comment, or make your variable names unnecessarily long.