Lindsey Bieda and Joe Frambach
Dynamic Programming Problems
10.26.2011

6. Show that if one of the following three problems has a polynomial time algorithm then they all do.

- **BOOLCIRCUIT**: The problem is to determine whether a Boolean Circuit (with gates NOT, binary AND, and binary OR) has some input that causes all of the output lines to be 1. Assume that the fan-out (the number of gates that the output of a single gate can be fed into) of the gates in a circuit may be arbitrary.
- **BOOLCIRCUIT2**: The problem is to determine whether a Boolean Circuit (with gates NOT, binary AND, and binary OR), and fan-out at most 2, has some input that causes all of the output lines to be 1.
- **PLANAR**: The problem is to determine whether a planar Boolean Circuit (with gates NOT, binary AND, and binary OR) has some input that causes all of the output lines to be 1. A circuit is planar if it can be laid on on the 2D plane so that no pair of lines cross (if you like, you can assume that the layout is part of the input).

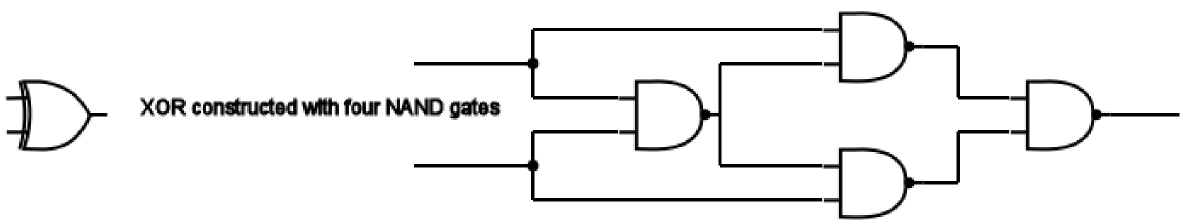
BOOLCIRCUIT2 $\leq$ BOOLCIRCUIT
Program BOOLCIRCUIT2:
    Read Input C.
    Return BOOLCIRCUIT(C)

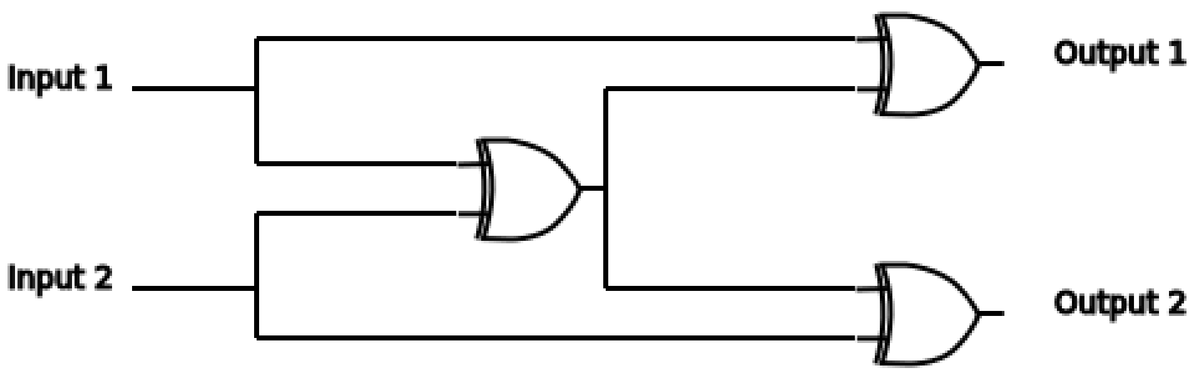
BOOLCIRCUIT $\leq$ PLANAR

Program PLANAR:

Read input C.

In $O(n^4)$ time, iterate through all 4-tuples of gates. We can determine if they have lines crossing by assuming the input also provides the layout (x-y coordinates) of each gate. Replace each crossing with the following structure:





Crossing two wires, without crossing.
Output 1 retains the value of Input 2.
Output 2 retains the value of Input 1.

BOOLCIRCUIT $\leq$ BOOLCIRCUIT2

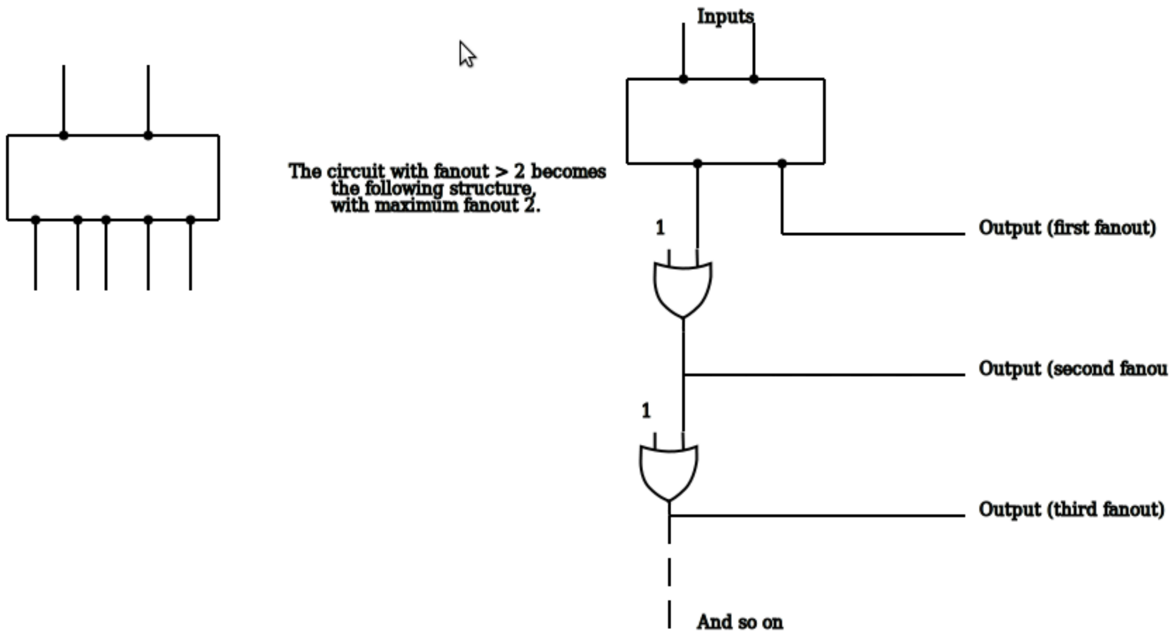
Program BOOLCIRCUIT:

Read input C.

For each gate g with fanout > 2: *This is done in O(n) time*

Replace g with the circuitry described in the following figure.

Output BOOLCIRCUIT2(C).



PLANAR $\leq$ BOOLCIRCUIT2

Program PLANAR:

Read input C.

For each gate g with fanout > 2: *This is done in O(n) time*

Replace g with the circuitry described in the previous figure.

Output BOOLCIRCUIT2(C).

10. Show that the clique problem is self-reducible. The decision problem is to take a graph G and an integer k and decide if G has a clique of size k or not. The optimization problem takes a graph G, and returns a largest clique in G. So you must show that if the decision problem has a polynomial time algorithm then the optimization problem also has a polynomial time algorithm. Recall that a clique is a collection of mutually adjacent vertices.

CLIQUE-OPT $\leq$ CLIQUE-DEC

Program CLIQUE-OPT:
 Read graph G, integer k.
 First, find the maximum size of the maximum clique.
 For $i = n$ to 1:
  If CLIQUE-DEC(G,i) returns true:
   size $s = i$.
   End for-loop.
 Next, find the actual clique by removing the vertices NOT in the clique
 For each vertex v in G:
 $G' = G$ - v
  If CLIQUE-DEC(G', s) returns true:
   $G = G'$.
 G is the graph containing only the maximum-size clique.
 Return G.

15. Consider the following variant of the MST problem. The input consists of an undirected graph G and an integer k. The problem is to find a spanning tree T of G such that the degree of each node in T is at most k, or report that no such tree exists. Show by reduction that if this problem has a polynomial time algorithm then the Hamiltonian path problem has a polynomial time algorithm. The Hamiltonian path problem asks you to determine whether a graph has a simple path that spans the vertices.

HAMPATH $\leq$ MST

Program HAMPATH:
    Read G.
    T = MST(G,2).
    if T contains all vertices of G:
        Return T.
    Return FAILURE.

If there exists an MST solution of at most 2 then that tree is a Hamiltonian path.