6.892: Algorithmic Lower Bounds, Spring 2019

Prof. Erik Demaine, Jeffrey Bosboom, Jayson Lynch

Problem Set 4

Due: Tuesday, March 5, 2019 at noon

Problem 4.1 [Circuit SAT to (formula) SAT].

Lecture 4 introduced SAT and Circuit SAT and stated that both are NP-hard. It is straightforward to reduce SAT to Circuit SAT by giving an algorithm to convert a formula into an equivalent circuit. This problem is to reduce **Circuit SAT to SAT** by giving a polynomial-time algorithm to convert a circuit into an equivalent formula.

The input to your reduction is a circuit represented by a **directed acyclic graph** with n **source vertices** (in-degree 0, representing the circuit inputs $x_1, x_2, \ldots, x_n$) and **one sink** vertex (out-degree 0, representing the circuit output). The remaining vertices each represent and are labeled by a logic gate — **AND**, **OR**, or **NOT** — with incoming edges from the vertices providing the input to that gate and outgoing edges to the vertices consuming the output of the gate. NOT gates have exactly one input, while AND and OR gates can have any positive number of inputs. Note that the output of a gate can be connected to the input of any number of gates (according to its out-degree). The decision question is whether there exists an assignment $[x_1, x_2, \ldots, x_n] = \{0, 1\}^n$ such that the circuit output is 1.

Your reduction's **size blowup** must be polynomial to be a correct reduction: the formula's size must be polynomial in the circuit's size. Ideally, your reduction's blowup will be only **linear**.