

**Problem Set 4 Solution**

*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  $x_1, x_2, \dots, x_n$ ) 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  $[x_1, x_2, \dots, x_n] = \{0, 1\}^n$  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**.

**Solution:** This type of transformation is called a *Tseytin transformation*; see e.g. [https://en.wikipedia.org/wiki/Tseytin\\_transformation](https://en.wikipedia.org/wiki/Tseytin_transformation).

Given an instance of Circuit SAT, we construct an instance of SAT as follows: We define  $x \Leftrightarrow y$  as  $(x \wedge y) \vee (\neg x \wedge \neg y)$ , which is a formula saying  $x$  and  $y$  have the same truth value.

- We have a variable  $x$  for each vertex  $x$  of the circuit.
- The formula is an AND of the following clauses:
  - For each NOT vertex  $y$  with input  $x$ , the clause  $y \Leftrightarrow \neg x$ .
  - For each AND vertex  $y$  with inputs  $x_1, \dots, x_k$ , the clause  $y \Leftrightarrow (x_1 \wedge \dots \wedge x_k)$ .
  - For each OR vertex  $y$  with inputs  $x_1, \dots, x_k$ , the clause  $y \Leftrightarrow (x_1 \vee \dots \vee x_k)$ .
  - The clause  $x_s$ , where  $x_s$  is the sink vertex.

This construction clearly takes polynomial time. Each vertex and edge in the circuit leads to a constant amount of the formula, so the blowup is linear.

Suppose the original Circuit SAT instance was solvable, so there is an assignment of truth values to the source vertices which makes the sink vertex true. Consider the assignment to the formula which has the same value on source vertices, and at each gate vertex has the value taken by that gate when the inputs to the circuit are set as in the solution to the circuit. This assignment satisfies each clause representing a gate, and satisfies the clause  $x_s$  because the assignment makes the output of the circuit true. So there is a satisfying assignment for the formula.

Now suppose the formula has a satisfying assignment. We take the input to the circuit corresponding to the values of the variables  $x_1, x_n$  which represent the source vertices. With this input, the value at each vertex  $y$  in the circuit is the same as the value of  $y$  in the satisfying assignment, since the output of each gate must be correct for the corresponding clause to be satisfied. Since  $x_s$  must be true in the satisfying assignment, this input to the circuit makes the output true, so the circuit is satisfiable.