Ex. 2 Computational Complexity & Physics David Gross, Richard Kueng
`www.thp.uni-koeln.de/gross/teaching.html` Due: Nov 11th

[P3] [*The Cook-Levin Theorem*] The goal of this exercise is to prove that Sat is NP-complete. For our purposes, we define Sat to be the language of all Boolean formulas for which there exists a satisfying assignment. A Boolean formula consists of variables x_i taking values in $\{\text{true}, \text{false}\}$ (equivalently, $\{1,0\}$), the logical operations of "and", "or", and "not", and parantheses. A formula $\phi(x_1, \ldots, x_n)$ is satisfiable if there exists an assignment of $\text{true}, \text{false}$ to each x_i such that $\phi(x_1, \ldots, x_n)$ evaluates to true (c.f. lecture notes).

By definition of NP-hardness, we have to show that every language L in NP is poly-time reducible to Sat. Let L be such a language, let $x \in \{0,1\}^*$ bit an input bit string, and let $n = |x|$ be its length. By the definition of polynomial reduction, we have to build a Boolean formula that is satisfiable if and only if $x \in L$. Also, it must be possile to carry out this conversion to a Boolean formula in a number of steps polynomial in n.

Let V be the verifier for L, as in the definition of NP. The goal is to construct a boolean formula that evaluates to true if and only if there exsits a suitable u such that the Turing machine V, on input of $\langle x, u \rangle$ halts and outputs 1. This is very similar to the arithmetic statement we constructed in the proof of Gödel's Theorem.

(1) As was the case with Gödel, we need to find a formula that is satisfied if there exists a u and a valid history $n = \langle n_1, \ldots, n_t \rangle$ of states of the verifier $V(x, u)$, such that the end result is 1. A major difference is that in this case, we have information about the runtime. What is that piece of information? Use it to define an encoding of the state of the TM by a string of boolean variables (equivalent to a bit string) of the form $n_i = \langle \text{state of DFA}, \text{position of head}, \text{contents of entire tape that will be used} \rangle$. Give *expicit* formulas for the positions at which the various pieces of information are stored in the string. Why do we use a different encoding from the Gödel case? (2 P.)

(2) Find a boolean formula $\phi_{\text{init}}(u, n)$ that evaluates to true if and only if n_1 represents the verifier in starting state with $\langle x, u \rangle$ on the tape. Design formulas $\phi_{\text{halt,true}}(n), \phi_{\text{next}}(n_t, n_{t+1}, t)$ that check the obvious conditoins (refer back to Gödel if necessary). (5 P.)

(3) Combine these steps to show that Sat is NP-*complete*. In particular, argue that the reduction you have constructed is *polynomial*. (3 P.)