(1) Consider the Boolean variables $x_{ij\gamma}$, $y_{ij}$, and $z_{iq}$ to describe the configuration of a non-deterministic Turing machine at step $i$ as in class to prove that SAT and 3SAT are NP-complete (see notes of November 1, 4, and 6).\(^1\)

(a) What is the meaning of the formula

$$(x_{i,j,\sigma} = \text{true}) \land (y_{i,j} = \text{true}) \land (z_{i,q} = \text{true})$$

($\land$ means AND) in terms of the Turing machine configuration at step $i$?

(b) Explain why the formula in part (a) has the same true/false values as

$$x_{i,j,\sigma} \land y_{i,j} \land z_{i,q}$$

(c) Explain the meaning of the formula

$$x_{i,j,\sigma} \land y_{i,j} \land z_{i,q} \Rightarrow (x_{i,j',\sigma'} = x_{i+1,j',\sigma'})$$

and why part of our proof of the Cook-Levin theorem (i.e., that SAT and 3SAT are NP-complete) involves formulas like this.

(d) Show that the Boolean formula $u_4 = u_5$ is equivalent to the formula

$$(-u_4 \lor u_5) \land (u_4 \lor -u_5)$$

---

\(^1\) So $x_{i,j\gamma}$ is true iff at step $i$ and cell $j$ the symbol $\gamma$ appears there; $y_{ij}$ is true iff at step $i$ the tape head is over cell $j$; and $z_{iq}$ is true iff at step $i$ the configuration is in state $q$. 

Research supported in part by an NSERC grant.
(e) Use the fact that \(p \implies q\) is equivalent to \(\neg p \lor q\) (\(\lor\) is OR) to write
\[u_1 \land u_2 \land u_3 \implies (u_4 = u_5),\]
using just \(\neg, \lor, \land\) and the \(u_1, \ldots, u_5\).

(f) Write a 3CNF formula that is satisfiable iff the variables \(u_1, \ldots, u_5\) satisfy
\[u_1 \land u_2 \land u_3 \implies (u_4 = u_5).\]

(g) How are parts (c) and (f) involved in our proof (in class) that 3SAT is NP-complete?

(2) In view of Problem (1), explain the meaning of the formula
\[x_{i,j,\sigma} \land y_{i,j} \land z_{i,q} \implies \left( (x_{i+1,j,\sigma'} \land y_{i+1,j+1} \land z_{i+1,q'}) \lor (x_{i+1,j,\sigma''} \land y_{i+1,j-1} \land z_{i+1,q''}) \right).\]
For which value of \(\delta(q, \sigma)\) would you use a formula like this (where \(\delta\) is the non-deterministic Turing machine transition function)?

(3) Show that if SAT can be decided in polynomial time, then there is a Turing machine that in polynomial time takes as input a Boolean formula \(f(x_1, \ldots, x_n)\) and gives you true/false values for \(x_1, \ldots, x_n\) for which \(f(x_1, \ldots, x_n)\) is true, provided that \(f\) is satisfiable. [Hint: as a first step, notice that if \(f(x_1, \ldots, x_n)\) is satisfiable, then either \(f(\text{true}, x_2, \ldots, x_n)\) is satisfiable or \(f(\text{false}, x_2, \ldots, x_n)\) is satisfiable.]

(4) Using your answer to Question 2, show that if SAT can be decided in polynomial time, then there is a Turing machine that in polynomial time takes as input a graph and returns a 3-colouring of the graph if the graph can be 3-coloured.

(5) Using your answer to Question 2, show that if SAT can be decided in polynomial time, then there is a Turing machine that in polynomial time takes as input a positive integer and returns a non-trivial factorization of the integer (i.e., as the product of two smaller positive integers) if one exists.