Cook’s Theorem

This is the last theorem of the course, and one of the most important!

We will prove that SAT is NP-complete.

Thus there really are the “hardest” problems in NP.
What is SAT?

First we need to define boolean formulas.

A boolean formula consists of variables that take the values 0 and 1 (equivalently false and true) and logical connectives. Parentheses may be used for grouping.

The logical connectives are: NOT (¬), AND (∧), OR (∨), IMPLIES (⇒), and IFF (⇔).
Cook’s Theorem

Here’s an example of a boolean formula.

\[(x_1 \land \lnot x_2) \implies (x_3 \lor x_4).\]

A boolean formula is *satisfiable* if we can assign truth values to the variables and make the formula true.

For example, the above formula is satisfiable because we can take \((x_1, x_2, x_3, x_4) = (1, 0, 0, 1)\).

On the other hand, the formula \(x_1 \land \lnot x_1\) is clearly not satisfiable.
Then SAT, the language of satisfiable boolean formulas, is defined to be \( \{ e(\varphi) : \varphi \text{ is a satisfiable boolean formula} \} \).

We will show that SAT is NP-complete: this is called Cook’s theorem.

The easy part is to show that SAT \( \in \) NP.

A nondeterministic Turing machine can guess a satisfying assignment, substitute the values of the variables, and evaluate the logical expression, all in polynomial time. (Convince yourself.)
Stephen Cook (b. 1939) is an emeritus professor at the University of Toronto.

His 1971 paper, “The complexity of theorem proving procedures”, introduced the notion of NP-completeness and proved that SAT is NP-complete.

Around the same time, Leonid Levin had similar ideas in the Soviet Union, so Cook’s theorem is sometimes called the Cook-Levin theorem.

Stephen Cook won the Turing award in 1982 for his work.
Now the hard part: we need to show every language in NP reduces to SAT.

Let $L$ be a language in NP. We need to exhibit a polynomial-time computable function $f$ such that $x \in L \iff f(x) \in \text{SAT}$.

Here’s the 30-second version of the proof.

Since $L \in \text{NP}$, that means there is a nondeterministic TM $M$ running in polynomial time that decides $L$.

We will construct a logical formula $f(x)$ that encodes the possible computations of $M$ on the input $x$. This logical formula will be satisfiable iff $M$ running on $x$ can reach $q_{\text{acc}}$. 
What does the logical formula $f(x)$ have to say?

It has to say

(a) $M$ starts with $x$ on its tape
(b) Each configuration of $M$ follows from the previous configuration by a valid move of $M$
(c) The TM ends up in $q_{acc}$.

What will the variables of $f(x)$ be? They will represent aspects of each configuration.
Cook’s Theorem

We will think of a list of configurations of
\( M = (Q, \Sigma, \Gamma, \delta, q_0, q_{acc}, q_{rej}) \), indexed by time, namely
\( C_0, C_1, \ldots, C_{p(n)} \).

Remember that each configuration looks like a string \( x(p, a)y \),
where the current contents of the tape is \( xay \), the TM is in state
\( p \), and the TM is scanning the \( a \).

The \((p, a)\) is a “composite symbol” of state and tape alphabet
symbol.

Let \( \Delta = \Gamma \cup Q \times \Gamma \) be the set of all possible symbols occurring in
any configuration.
Cook’s Theorem

Since $M$ runs in polynomial time, this means that there is some polynomial $p$ such that each configuration can be bounded in length by $p(n)$ and there are at most $p(n) + 1$ of them, starting at configuration 0.

We will pad configurations on the right by blanks, if necessary, so they are all $p(n)$ symbols long.

We will also make the convention that if the TM ever halts, then all succeeding configurations are the same as the halting configuration.
Cook’s Theorem

We need to have a boolean variable that is true if position $i$ of $C_t$, the string representing the configuration at time $t$, has symbol $a$, and false otherwise.

Call this variable $c_{t,i,a}$.

Then we will construct seven clauses, each asserting that some aspect of the computation is valid.
Suppose \( x = a_1 \cdots a_n \).

Let’s make a formula asserting the 0’th configuration is correct:

\[
P_1 : c_{0,0,(q_0, \omega)} \land \left( \bigwedge_{1 \leq j \leq n} c_{0,j,a_j} \right) \land \left( \bigwedge_{n+1 \leq j \leq p(n)} c_{0,j, \omega} \right)
\]

This says first symbol of the 0’th configuration is \((q_0, \omega)\), the next \(n\) symbols are \(x\), and the following \(p(n) - n\) symbols are \(\omega\).
Proof of Cook’s Theorem

Next, assert that every cell contains \textit{at least one} symbol at any given moment:

\[
P_2 : \bigwedge_{0 \leq t \leq p(n)} \bigwedge_{0 \leq i \leq p(n)} \bigvee_{a \in \Delta} c_{t,i,a}.
\]

And every cell contains \textit{at most one} symbol at any given moment:

\[
P_3 : \bigwedge_{0 \leq t \leq p(n)} \bigwedge_{0 \leq i \leq p(n)} \neg \bigvee_{a, b \in \Delta, a \neq b} (c_{t,i,a} \land c_{t,i,b}).
\]
Proof of Cook’s Theorem

Assert that every configuration contains \textit{at least one} “composite symbol” specifying a state. Let us call $S = Q \times \Gamma$ the set of composite symbols.

\[ P_4 : \bigwedge_{0 \leq t \leq p(n)} \bigvee_{0 \leq i \leq p(n)} \bigvee_{a \in S} c_{t,i,a}. \]

And every configuration contains \textit{at most one} “composite symbol” specifying a state.

\[ P_5 : \bigwedge_{0 \leq t \leq p(n)} \bigwedge_{a,b \in S, a \neq b} \left( \bigvee_{0 \leq i \leq p(n)} c_{t,i,a} \right) \land \left( \bigvee_{0 \leq i \leq p(n)} c_{t,i,b} \right). \]
Proof of Cook’s Theorem

Now, assert that every configuration $C_{i+1}$ follows from the preceding one $C_i$ by a valid TM move.

This is the hardest one to make. The basic idea is that in each of the three cases of left move, right move, and stationary move, the symbol in position $i$ of a configuration at time $t+1$ depends only on the symbols in positions $i-1$, $i$, and $i+1$ at time $t$.

Namely, if it is a left move, say $\delta(p, a) = (q, b, \leftarrow)$ then $\cdots c(p, a)d \cdots$ becomes $\cdots (q, c)bd \cdots$.

If it is a right move, say $\delta(p, a) = (q, b, \rightarrow)$ then $\cdots c(p, a)d \cdots$ becomes $\cdots cb(q, d) \cdots$.

And if it is a stationary move, say $\delta(p, a) = (q, b, \downarrow)$, then $\cdots c(p, a)d \cdots$ becomes $\cdots c(q, b), d \cdots$.

In all other cases the symbol at time $t+1$ is the same as it was at time $t$. 
Proof of Cook’s Theorem

So using this idea, based on the transition function $\delta$ of $M$ we can construct predicate $f$ that given the $b_1, b_2, b_3, b_4$, evaluates to true if $b_4$ could be the position at time $t + 1$ at position $i$ if the symbols at positions $i - 1, i, i + 1$ at time $t$ were $b_1, b_2, b_3$.

We say “could be” because $M$ is nondeterministic and there could be several possible outcomes $b_4$ for each $b_1, b_2, b_3$, depending on what move is chosen.

So the formula is $P_6$:

$$P_6 : \bigwedge_{0 \leq t < p(n)} \bigwedge_{0 \leq i \leq p(n)} \bigvee_{f(b_1, b_2, b_3, b_4) \text{ is true}} \left( c_{t, i-1, b_1} \land c_{t, i, b_2} \land c_{t, i+1, b_3} \land c_{t+1, i, b_4} \right).$$

(Actually this is not quite right because of the symbols at the beginning and end of the tape, but you get the idea.)
Proof of Cook’s Theorem

The last configuration has a composite symbol with \( q_{\text{acc}} \) in it. Define \( A = \{ q_{\text{acc}} \} \times \Gamma \), the set of composite symbols corresponding to accepting.

\[
P_7 : \bigvee_{0 \leq i \leq p(n)} \bigvee_{a \in A} c_{p(n),i,a}.
\]

Then we form \( f(x) \) by taking the “and” of \( P_1, P_2, P_3, P_4, P_5, P_6, P_7. \)

The size of this formula is then \( O(p(n)^2) \), so given \( x \) we can construct it in polynomial time.

This completes the proof.
Now that we have one NP-complete problem, we can find many others as follows:

**Theorem.** If $A$ is NP-hard and $A \leq_p B$, then $B$ is NP-hard.

**Proof.** Let $C \in \text{NP}$. Since $A$ is NP-hard, we know that $C \leq_p A$. But then, by the transitivity of polynomial-time reductions, we know $C \leq_p B$. Since $C$ was arbitrary, this proves that $B$ is NP-hard.