1 Cook's Theorem

Theorem 1.1 (Cook's Theorem)**.** *The Satisfiability problem is $\mathcal{NP}$-complete.*

2 Outline of proof

The proof consists of two basic steps:

1. Convert the execution of a polynomial-time NDTM to a bunch of well formed Boolean formulae such that the formulae are satisfied if and only if the machine accepts the input.
2. Show the sum of the lengths of the formulae is polynomial in the size of the problem.

- $\mathcal{NP}$-Hard (L) - can polynomially reduce any $\mathcal{NP}$ problem to L.
- $\mathcal{NP}$-Complete - $L \in \mathcal{NP}$
- $L \in \mathcal{NP} \implies$ NDTM for L that runs in polynomial time.
- An NDTM is the only model we have for NP problems.
- SAT $\in \mathcal{NP}$ (can check solution in polynomial time).
- Therefore, if I can polynomially reduce an arbitrary polynomial NDTM to SAT, I've proven SAT is $\mathcal{NP}$-complete.

A NDTM has a finite set of states, a finite number of transitions and has a finite alphabet. We know the NDTM is polynomial, so it must complete in polynomial time. Let's call that polynomial $p(n)$.

Unfortunately, logical formula are clumsy at representing things that change over time, and that makes the proof a bit long. Nevertheless, the basic idea is to encode the constraints placed on a TM (e.g. only in one state at one time), plus the constraints of the change in state over time (changes from state x to state y) as a logical formula.

Let's be a little more formal. Let $L(M)$ be a language accepted by a non-deterministic Turing machine, M. For any input u, we need to transform the computations of M on u into a conjunctive normal form formula, $f(u)$, such that $f(u)$ is satisfiable if and only if $u \in L(M)$. Rather than treat an NDTM directly as is done in the textbook, we are going to consider a DTM with a certificate $C(u)$ as it makes the proof a little bit simpler.

- Let $L(M)$ be a language accepted by a non-deterministic Turing machine, M.
- For any input u, we need to transform the computations of M on u into a conjunctive normal form formula, $f(u)$, such that $f(u)$ is satisfiable if and only if $u \in L(M)$.
- consider a DTM with a certificate $C(u)$

Definition of M:

- States $Q = \{q_0, q_1, \ldots, q_m\}$
- Tape alphabet $\Gamma = \{B = a_0, a_1, \ldots, a_s, a_{s+1}, \ldots, a_v\}$
- Input alphabet $\Sigma = \{a_{s+1}, \ldots, a_v\}$
- Single accepting state $F = \{q_m\}$ (no loss in generality)

Since the computation terminates after $p(n)$ transitions the only tape squares that need to be considered are those up to tape square $p(n)$.

The trick is to define the complete operation of M on u as a well formed formula. We can do this by defining clauses that encode the constraints placed on M for it to behave as a Turing Machine. Informally, these constraints are:

One State At every time, M is in exactly one of a set $q_1, q_2, \ldots$ of states

One Position At every time, the head is in exactly one position of the tape

One Symbol At every time, each tape square has exactly one tape symbol

Transition At time $t+1$ the state, tape content, position of the head are related to their configuration at time t. This means

- Tape squares not under the head do not change
- The tape square under the head, the state, and the new head position change by a transition

Initial Configuration At time 0 the tape squares $0, 1, \ldots, n$ contain $B, u_1, u_2, \ldots, u_n$; and the tape squares $n+1, n+2, \ldots$ contain the symbols of $C(u)$.

Final Configuration At time $p(n)$ the machine is in the designated final state. We don't need to worry about termination before $p(n)$ transitions because we can define M to loop in the accepting state.

We are going to treat each of these constraints in turn. Each clause relating to a constraint will be a numbered equation in the following notes.

Proof. Now on to the proof proper.

3 One and only one state

First define Q_{it} as the variable that M is in state q_i at time t. It should be clear that at any time t, one of the Q_{it} must be true, and all others must be false. The clause for at least one state is:

$$Q_{0t} \vee Q_{1t} \vee \ldots \vee Q_{mt} \qquad 0 \leq t \leq p(n). \tag{1}$$

There are $p(n) + 1$ clauses, each with $m + 1$ terms.

To ensure only one state, we need the following clause:

$$\neg Q_{rt} \vee \neg Q_{st} \qquad t = 0, 1, \ldots, p(n); 0 \leq r \neq s \leq m. \tag{2}$$

There are $(p(n) + 1)\frac{m(m-1)}{2}$ clauses each with 2 terms[1].

4 One and only one head position

First define variable P_{it} to represent the statement "M's head is on position i of the tape at time t". Again, it should be clear that at any time t, one of the P_{it} must be true, and all others must be false. For at least one head position we have:

$$P_{0t} \vee P_{1t} \vee \ldots \vee P_{p(n),t} \qquad 0 \leq t \leq p(n). \tag{3}$$

There are $p(n) + 1$ clauses, each with $p(n) + 1$ terms.

For only one head position we need:

$$\neg P_{rt} \vee \neg P_{st} \qquad 0 \leq t \leq p(n); 0 \leq r \neq s \leq p(n). \tag{4}$$

There are $(p(n) + 1)(p(n) + 1)p(n)/2$ clauses each with 2 terms.

[1] $\frac{m(m-1)}{2}$ is m choose 2.

5 One and only one tape symbol per square

First define variable S_{ijt} to represent the statement "Tape square i contains symbol a_j at time t". This one is a little more complicated, but again it should be clear that at time t and tape square i, one of the S_{ijt} must be true, and all others must be false. To ensure at least one is true we need:

$$S_{i0t} \vee S_{i1t} \vee \ldots \vee S_{ivt} \qquad 0 \leq i, t \leq p(n). \tag{5}$$

There are $(p(n)+1)(p(n)+1)$ clauses, each with $v+1$ terms.

To ensure that at most one is true we need:

$$\neg S_{irt} \vee \neg S_{ist} \qquad 0 \leq i, t \leq p(n); 0 \leq r \neq s \leq v. \tag{6}$$

There are $(p(n)+1)(p(n)+1)(v+1)v/2$ clauses, each with 2 terms.

6 Transition

6.1 The tape not under the head doesn't change

In this case we need not say anything about the tape under the head (because that may change). But we do need to say something about all the tape positions that are not under the head. We want to say that if the head is not over tape position i, then the symbol at i does not change from time t to $t+1$. Or in other words:

$$\neg P_{it} \implies (S_{irt} \implies S_{ir,t+1}),$$

for all tape positions i, for all times t, and for all tape symbols a_r.

Recall that $a \implies b$ is read as a implies b, and is equivalent to $\neg a \vee b$. So the above clause can be converted (in steps) to:

$$\begin{aligned} &\neg P_{it} \implies (\neg S_{irt} \vee S_{ir,t+1}) \\ &P_{it} \vee \neg S_{irt} \vee S_{ir,t+1} \qquad 0 \leq i, t \leq p(n); 0 \leq r \leq v. \end{aligned} \tag{7}$$

The number of clauses in this case is $(p(n)+1)p(n)(v+1)$, each with 3 terms.

6.2 The tape under the head changes by a transition

Let's assume that at time t, M is in state q_i scanning symbol a_r in tape position k. In this case, the variables Q_{it}, P_{kt}, S_{krt} are all true. Let $(q_i, a_r) \longrightarrow [q_j, a_s, d]$ be a transition rule ($d = \pm 1$) — that is, d shifts the head either left or right.

Then the following clause is satisfied only when the next state is q_j:

$$\neg Q_{it} \vee \neg P_{kt} \vee \neg S_{krt} \vee Q_{j,t+1} \qquad 0 \leq i, j \leq m; 0 \leq k, t \leq p(n); 0 \leq r \leq v. \tag{8}$$

There are $(m+1)(m+1)(p(n)+1)(p(n)+1)(v+1)$ such clauses.

We need a similar condition for the new tape symbol at $t+1$:

$$\neg Q_{it} \vee \neg P_{kt} \vee \neg S_{krt} \vee S_{ks,t+1} \qquad 0 \leq i \leq m; 0 \leq k, t \leq p(n); 0 \leq r, s \leq v. \tag{9}$$

There are $(m+1)(p(n)+1)(p(n)+1)(v+1)(v+1)$ such clauses.

And another set of clauses for the new head position:

$$\neg Q_{it} \vee \neg P_{kt} \vee \neg S_{krt} \vee P_{k+d,t+1} \qquad 0 \leq i \leq m; 0 \leq k, t \leq p(n); 0 \leq r \leq v; d = \pm 1 \tag{10}$$

There are $2(m+1)(p(n)+1)(p(n)+1)v$ of these clauses.

7 Initial and Final Configuration

Initially we have the input u on the tape in the first n positions followed by the certificate $C(u)$. Assume that $u = a_{r_1} a_{r_2} \ldots a_{r_n}$. Then we also need clauses describing the initial state, the initial position of the head, the initial set of symbols on the tape, and the final accepting state of the machine:

$$Q_{00} \tag{11}$$

$$P_{00} \tag{12}$$

$$S_{i,r_i,0} \qquad 0 \leq i \leq n \tag{13}$$

$$Q_{m,p(n)} \tag{14}$$

Note that we have dealt with the non-determinacy by leaving the variables $S_{ij,0}$ (where $i > n$) unspecified (this is the certificate).

8 Finally ...

It should be clear that all of these clauses define the working of the machine M on the input $uC(u)$. If at time $p(n)$, M is in the accepting state, then the set of clauses must be satisfied. If M is not in the accepting state at time $p(n)$, then the set of clauses is not satisfied. So we have shown that any NDTM reduces to SAT. Furthermore, each of the clauses involves polynomially-many terms and the number of clauses is polynomial in the size of the machine plus the size of the input, therefore we can construct the SAT problem in polynomial time and space. Therefore any NDTM is polynomially-reducible to SAT. Since SAT is in $\mathcal{NP}$ and all problems in $\mathcal{NP}$ can be reduced to SAT, it follows that SAT is $\mathcal{NP}$-complete.