Cook's Theorem

Jason Belt
CIS 890

Lecture material adapted from Dr. Howell's CIS 770 lecture notes

October 26, 2010

Computational Complexity

Defn: Let $T : \mathbb{N} \to \mathbb{N}$. A TM M is said to have time complexity $T(n)$ if on every input string w, M takes no more than $T(|w|)$ transitions.

Defn: $\mathcal{P}$ is the set of all languages $L \subseteq \{0, 1\}^*$ such that there is a polynomial $p(n)$ and a TM M with time complexity $p(n)$ such that $L(M) = L$.

Defn: $\mathcal{NP}$ is the set of all languages $L \subseteq \{0, 1\}^*$ such that there is a polynomial $p(n)$ and a nondeterministic TM M with the time complexity $p(n)$ such that $L(M) = L$.

$\mathcal{NP}$ Classes

Defn: A language L is said to be $\mathcal{NP}$-hard if for every $L' \in \mathcal{NP}$, $L' \leq_m^p L$.

$\mathcal{NP}$ Classes

Defn: A language L is said to be $\mathcal{NP}$-hard if for every $L' \in \mathcal{NP}$, $L' \leq_m^p L$.

$L_1 \leq_m^p L_2$

Let $L_1 \subseteq \Sigma^*$, $L_2 \subseteq \Delta^*$. We say $L_1 \leq_m^p L_2$ if there exists a TM $M = (Q, \Sigma, \Gamma, \delta, q_0, B, \{q\})$ with polynomial running time complexity such that

- $\Delta \subseteq \Gamma$;
- on every input, M halts on an ID qy for some $y \in \Delta^*$; and
- if $q_0 x \vdash_M^* qy$, then $x \in L_1$ iff $y \in L_2$

$\mathcal{NP}$ Classes

Defn: A language L is said to be $\mathcal{NP}$-hard if for every $L' \in \mathcal{NP}$, $L' \leq_m^p L$.

$L_1 \leq_m^p L_2$

Let $L_1 \subseteq \Sigma^*$, $L_2 \subseteq \Delta^*$. We say $L_1 \leq_m^p L_2$ if there exists a TM $M = (Q, \Sigma, \Gamma, \delta, q_0, B, \{q\})$ with polynomial running time complexity such that

- $\Delta \subseteq \Gamma$;
- on every input, M halts on an ID qy for some $y \in \Delta^*$; and
- if $q_0 x \vdash_M^* qy$, then $x \in L_1$ iff $y \in L_2$

Defn: If $L \in \mathcal{NP}$-hard and $L \in \mathcal{NP}$ then L is said to be $\mathcal{NP}$-complete.

Boolean Satisfiability (SAT)

Input: A boolean formulat $\mathcal{F}$ consisting of boolean variables and the operators $\wedge$, $\vee$, $\neg$

Question: Is there an assignment of boolean values to the variables in $\mathcal{F}$ that causes $\mathcal{F}$ to evaluate to **true**

Claim: $L_{SAT} \in \mathcal{NP}$-complete, where L_{SAT} denotes the language of satisfiable formulas encoded over $\{0, 1\}$

Cook's Theorem: $SAT \in \mathcal{NP}$-complete

Proof:

1. $L_{SAT} \in \mathcal{NP}$.
    - Use a NTM to guess a truth assignment T for a given expression E. If $|E| = n$ then $O(n)$ time suffices on a multitape NTM. Note that there may be as many as 2^n unique truth assignments.
    - Evaluate E for the truth assignment T. Can be done in $O(n^2)$ time on a multitape NTM
2. $L_{SAT} \in \mathcal{NP}$-hard

    Proof idea:

    - For each language L in $\mathcal{NP}$, there is a polynomial $p(n)$ and a nondeterministic TM M with time complexity $p(n)$ such that $L(M) = L$
    - From $w \in \{0, 1\}^*$, we construct a formula $\mathcal{F}$ that is satisfiable iff there is an accepting computation of M on w
    - The time for the construction will be polynomial in $p(n)$

Cook's Theorem: $SAT \in \mathcal{NP}$-complete

Construction overview:

- We will view a computation as a sequence of IDs $\alpha_0, \ldots, \alpha_{p(n)}$ such that either $\alpha_i \vdash \alpha_{i+1}$ or $\alpha_i = \alpha_{i+1}$.
- Each α_i will be of the form $X_{-p_n} \cdots X_0 \cdots X_{p(n)+1}$ where X_j is either a tape symbol or a state.
- We use boolean variable y_{ijA} to denote whether X_j of α_j is A.
- $\mathcal{F}$ will constrain the sequence of IDs to be an accepting computation of w.

Cook's Theorem: $SAT \in \mathcal{NP}$-complete

We will describe a set of formulas, each enforcing certain constraints on the variables y_{ijA}, for $0 \leq i \leq p(n)$, $-p(n) \leq j \leq p(n)+1$, $A \in Q \cup \Gamma$.

$\mathcal{F}$ will be the conjuction of these formulas.

α_0 is the initial ID:

- y_{00q_0}
- y_{0ja_j} for $1 \leq j \leq n$, where $a_1 \cdots a_n = w$.
- y_{0jB} for $-p(n) \leq j < 0$, $n < j \leq p(n)+1$.

$\alpha_{p(n)}$ contains a final state

$$\bigvee_{j=-p(n)}^{p(n)+1} \bigvee_{q \in F} y_{p(n)jq}$$

Cook's Theorem: $SAT \in \mathcal{NP}$-complete

- We still need to enforce that $\alpha_i \vdash \alpha_{i+1}$ or $\alpha_i = \alpha_{i+1}$ for $0 \leq i \leq p(n)$.
- For $0 \leq i \leq p(n)$, $-p(n) \leq j \leq p(n) + 1$, we construct a formula enforcing one of the following
    1. X_{ij} is a state and $X_{i+1,j-1}X_{i+1,j}X_{i+1,j+1}$ results from doing nothing or taking a transition of M from $X_{i,j-1}X_{ij}X_{i,j+1}$ (if $j = -p(n)$ or $j = p(n) + 1$, this is omitted); or
    2. $X_{i,j-1}$, X_{ij}, and $X_{i,j+1}$ are not states, and $X_{i+1,j} = X_{ij}$

Cook's Theorem: $SAT \in \mathcal{NP}$-complete

Constraint 1 is enforced by the disjunction of the following formulas:

- For each $q \in Q$, $X, Y \in \Gamma$, and $(q', Z, R) \in \delta(q, Y)$:
  $y_{i,j-1,X} \wedge y_{i+1,j-1,X} \wedge y_{ijq} \wedge y_{i+1,j,Z} \wedge y_{i,j+1,Y} \wedge y_{i+1,j+1,q'}$.
- For each $q \in Q$, $X, Y \in \Gamma$, and $(q', Z, L) \in \delta(q, Y)$:
  $y_{i,j-1,X} \wedge y_{i+1,j-1,q'} \wedge y_{ijq} \wedge y_{i+1,j,X} \wedge y_{i,j+1,Y} \wedge y_{i+1,j+1,Z}$.
- For each $q \in Q$, $X, Y \in \Gamma$:
  $y_{i,j-1,X} \wedge y_{i+1,j-1,X} \wedge y_{ijq} \wedge y_{i+1,j,q} \wedge y_{i,j+1,Y} \wedge y_{i+1,j+1,Y}$.

Cook's Theorem: $SAT \in \mathcal{NP}$-complete

Constraint 2 is enforced by the conjunction of:

- $\bigvee_{X \in \Gamma} y_{i,j-1,X}$;
- $\bigvee_{X \in \Gamma} (y_{ijX} \wedge y_{i+1,j,X})$; and
- $\bigvee_{X \in \Gamma} y_{i,j+1,X}$.

Conjuncts containing out-of-bounds subscripts are omitted.

- The formula can be constructed in polynomial time.
- The formula is satisfiable iff M has an accepting computation on w
- Therefore, SAT is $\mathcal{NP}$-hard.