CSCI 1010
Models of Computation

Lecture 20
Limits to Language Recognition II
Overview

• Resource bounded computations
• Time and space complexity classes
• A simple speedup theorem
• Polynomial time reductions
• \textbf{NP}-complete languages again
Fun Fact

• The **Busy Beaver Function** $b(n)$ is a famous non-computable function. It is the number of characters that an $n$-state TM can write before halting when started on a blank tape.

• For TMs that write only blanks and 1s (no 0s) the function $b(6) > 95.5$ million. A machine was found that executes $> 8 \times 10^{15}$ steps!!!
Decision Problems Again

• A decision problem over the alphabet $\Sigma$ is defined by set $I \subseteq \Sigma^*$ instances and a function $\phi_p : I \rightarrow \{0,1\}$ where $I_{\text{yes}} = \{ w | \phi_p(w) = 1 \}$ and $I_{\text{no}} = I - I_{\text{yes}}$. 
Satisfiability

SATISFIABILITY

*Instances*: Literals $X = \{x_1, \overline{x}_1, x_2, \overline{x}_2, \ldots, x_n, \overline{x}_n\}$, and clauses $C = (c_1, c_2, \ldots, c_m)$, $c_i$ is a subset of $X$.

*Answer*: “Yes” if for assignment of 0,1 to $x_1$, $x_2$, ... $x_n$ at least one literal in each clause has value 1.

*Example*: $C_1 = (\{x_1, x_2, x_3\}, \{x_1, \overline{x}_2\}, \{x_2, \overline{x}_3\})$ is a “Yes” instance.

$C_2 = (\{x_1, x_2, x_3\}, \{x_1, \overline{x}_2\}, \{x_2, \overline{x}_3\}, \{x_3, \overline{x}_1\}, \{\overline{x}_1, \overline{x}_2, \overline{x}_3\})$ is a “No” instance.
TM Time and Space Bounds

• A DTM M computes a function $f$ of input length $n$ in **time** $T(n)$ if M computes $f$ in at most $T(n)$ time steps.

• A DTM M computes a function $f$ of input length $n$ in **space** $S(n)$ if M computes $f$ using at most $S(n)$ tape cells.
  – When $S(n)$ grows at least linearly with $n$, the DTM can be a one-tape TM.
  – When it grows more slowly, the DTM has a read-only input tape and a worktape using at most $S(n)$ cells.
Resource Bounded Computation

• Complexity classes are defined by “proper” space and time bounds.

• **Definition** A function \( r : \mathbb{N} \rightarrow \mathbb{N} \) (\( \mathbb{N} = \{0,1,2,\ldots\} \)) is **proper** if it is non-decreasing and for some tape symbol \( a \), there is a DTM \( M \) that on inputs of length \( n \), in time \( O(n+r(n)) \) and temporary space \( O(r(n)) \), \( M \) writes \( a^{r(n)} \) on one of its tapes and halts.
Time and Space Complexity Classes

• **Definition:** If \( r(n) \) is proper, \( \text{TIME}(r(n)) \) (\( \text{SPACE}(r(n)) \)) is the TM time (space) class containing languages that can be recognized in time (space) \( r(n) \) in the length \( n \) of inputs by a DTM.

• \( \text{NTIME}(r(n)) \) (\( \text{NSPACE}(r(n)) \)) is the nondeterministic TM time (space) complexity class similarly defined by NTM’s instead of DTM’s.

• The **union** of two complexity classes is another complexity class.
**Speedup Theorem**

- **Theorem**: (Linear speedup theorem) Given a TM that uses $kn$ steps ($kn$ tape cells) on inputs of length $n$, it can be simulated by a TM that uses $n$ steps ($n$ cells).

- **Proof** To show this, encode $k$ tape cells into one cell by expanding the size of the tape alphabet.
The Class NP

• **Definition:** If \( r(n) \) is proper, \( \text{TIME}(r(n)) \) \( \text{[NTIME}(r(n))] \) \( \text{(SPACE}(r(n)) \text{[NSPACE}(r(n))] \) is the time (space) class of languages recognizable in time (space) \( r(n) \) in the length \( n \) of inputs by a DTM \([NTM]\).

• **Definition** The class \( \text{NP} \) contains languages recognizable in polynomial time on a NDTM.
  \[
  \text{NP} = \bigcup_{k \geq 0} \text{NTIME}(n^k)
  \]
Polynomial-Time Reductions

**Definition:** Let $L_1$ and $L_2$ be languages and let $h : B^* \rightarrow B^*$ be a polynomial-time function. If $h$ is a polynomial-time reduction of $L_1$ to $L_2$ then $x \in L_1$ if and only if $h(x) \in L_2$. (We say that $L_1$ is polynomial-time reducible to $L_2$.)

**Definition:** Decision problems (languages) $P_1$ and $P_2$ are in the relation $P_1 \leq_p P_2$ if $P_1$ is polynomial-time reducible to $P_2$. 
Relationships Between Reductions

• **Theorem**: Let \( P_1 \) and \( P_2 \) be decision problems. If \( P_1 \leq_p P_2 \) and \( P_1 \) is not in \( P \), \( P_2 \) is not in \( P \).

• This is a direct analog of the theorem about unrestricted reducibility concerning decidable and undecidable languages established in the last lecture and found in Chapter 5. However, in this case the reduction is P-time.
Re-Definition of \textbf{NP}-Complete

\textbf{Definition}: A decision problem $P_2$ is \textbf{NP}-complete if $P_2$ is in \textbf{NP} and for every problem $P_1$ in \textbf{NP}, $P_1 \leq_p P_2$.

\textbf{SATISFIABILITY}

\textit{Instances}: Literals $X = \{x_1, \overline{x_1}, x_2, \overline{x_2}, ..., x_n, \overline{x_n}\}$, and clauses $C = (c_1, c_2, ..., c_m)$, $c_i$ is a subset of $X$.

\textit{Answer}: “Yes” if for assignment of 0, 1 to $x_1$, $x_2$, ..., $x_n$ at least one literal in each clause has value 1.
NP-Complete Problems

3-COLORING

*Instance:* Description of a graph $G = (V, E)$.

*Answer:* “Yes” if exists assignment of three colors to vertices $V$ so that adjacent vertices have different colors.

EXACT COVER

*Instance:* A set $S = \{u_1, u_2, \ldots, u_p\}$ and family $\{S_1, \ldots, S_n\}$ of subsets of $S$.

*Answer:* “Yes” if there exist disjoint $S_{j_1}, S_{j_2}, \ldots, S_{j_t}$ in family whose union is $S$.

($S_{j_1}, S_{j_2}, \ldots, S_{j_t}$ exactly cover $S$.)
NP-Complete Problem Reductions

- **Theorem:** EXACT COVER is NP-complete.
EXACT COVER is **NP-Complete**

**Proof** Clearly EXACT COVER in **NP**. Now reduce 3-COLORING to it.

- Given $G = (V,E)$, form instance of EXACT COVER with $S = V \cup \{<e,i> | e \in E, 0 \leq i \leq 2\}$ and subsets $S_{v,i} = \{v\} \cup \{<e,i> | e \text{ incident on } v\}$ and $R_{e,i} = \{<e,i>\}$ for $v \in V, e \in E, 0 \leq i \leq 2$.

- Thus, $S = \{v_1, v_2, v_3, v_4, v_5, v_6, ..., <e_{1,0}>, <e_{1,1}>, <e_{1,2}>, <e_{2,0}>, ... \}$
EXACT COVER

• $S = \{v_1, v_2, v_3, v_4, v_5, v_6, <e_{1,0}>, <e_{1,1}>, <e_{1,2}>, <e_{2,0}>, \ldots \}$

• $S_{v_1,0} = \{v_1\} \cup \{<e_{1,0}>, <e_{2,0}>, <e_{4,0}>\}$,
• $S_{v_1,1} = \{v_1\} \cup \{<e_{1,1}>, <e_{2,1}>, <e_{4,1}>\}$, etc.
• $R_{e_{1,0}} = \{<e_{1,0}>\}$, $R_{e_{1,1}} = \{<e_{1,1}>\}$, etc.
• Assume $G$ 3-colorable. Let $c(v) \in \{0,1,2\}$ be color of $v$. Sets $\{S_{v,c(v)} \mid v \in V\}$ and $\{R_{e,i} \mid <e,i> \notin S_{v,c(v)}\}$ for $i = 0, 1, 2$ form an exact cover 'cause if $e = (v,w)$, $c(v) \neq c(w)$ and $S_{v,c(v)}$ and $S_{w,c(w)}$, which can only have elements $<e,i>$ in common, are disjoint. Furthermore, all elements of $S$ are in some set.
• Assume exact cover. For each $v \in V$ there is a unique $c(v)$ such that $v \in S_{v,c(v)}$. To show $G$ is 3-colorable assume not and show a contradiction. If not, there is edge $e = (v,w)$ with $c(v) = c(w)$ such that $<e,c(v)>$ is in both $S_{v,c(v)}$ and $S_{w,c(w)}$, contradicting an exact cover.

• We conclude that $G$ is 3-colorable iff $S$ has exact cover.
Review

• Resource bounded computations
• Time and space complexity classes
• A simple speedup theorem
• Polynomial time reductions
• $\textbf{NP}$-complete languages again