CSCI 1010
Models of Computation

Lecture 24

P-Complete Problems
Overview

• The class $\mathbf{P}$.
• CIRCUIT VALUE
• Log-space computations and $\mathbf{P}$-complete langs.
• CIRCUIT VALUE is $\mathbf{P}$-complete.
• The composition of two log-space computations is log-space computable.
• MONOTONE CIRCUIT VALUE is $\mathbf{P}$-complete.
• LINEAR INEQUALITIES is $\mathbf{P}$-complete.
• Parallelizability of problems in $\mathbf{P}$. 
The Class \( \mathbf{P} \)

- A language \( L \subseteq \Sigma^* \) is in \( \mathbf{P} \) if there is a DTM \( M_L \) and polynomial \( p(n) \) such that for every \( x \in \Sigma^* \), \( M_L \) halts in \( p(|x|) \) steps and accepts \( x \) if it is in \( L \) and rejects it otherwise.
CIRCUIT VALUE (CV)

CIRCUIT VALUE (CV)

• *Instance*: A description of a circuit *over a complete basis* (AND, OR, NOT, e.g.) with fixed values for input variables and a designated output.

• *Answer*: “Yes” if the output has value 1.
Log-space Computations

• Logspace computations are defined for TMs in which the input tape is read-only.

• **Definition** A log-space program runs on a DTM and uses space on its work tape(s) that is logarithmic in the length n of its inputs.
Reducing \( L \in \mathcal{P} \) to CV

• Let \( M_L \) recognize \( L \in \mathcal{P} \).

• Write a program that describes a circuit to simulate \( M_L \) on \( x \) in \( T = p(n) \) steps, \( n = |x| \).
Deterministic Circuit Simulating DTM

- This circuit used to simulate NDTM except that it does not have choice inputs.
Program to Describe Circuit for $L \in \mathcal{P}$

\begin{verbatim}
for i = 0 to n-1
    write_cell_circuit_input(i, x_i)
for i = n to p(n)
    write_cell_circuit_input(i, \beta)
write_control_unit(0)
for t = 1 to p(n)
    write_control_unit(t)
    for j = 1 to p(n)
        write_cell_circuit(j, t)
write_circuit_for_accept_state
\end{verbatim}
Program to Describe Circuit for $L \in \mathbf{P}$

- $O(\log p(n)) = O(\log n)$ bits used to represent indices $i$, $t$, $j$.
- $O(\log n)$ bits used to represent
  - $\text{write}_\text{cell}_\text{circuit}_\text{input}(i, a)$
  - $\text{write}_\text{control}_\text{unit}(t)$
  - $\text{write}_\text{cell}_\text{circuit}(j, t)$
  - $\text{write}_\text{circuit}_\text{for}_\text{accept}_\text{state}$
- Can write previous program in $O(\log n)$ space.
**P-Complete Languages**

**Definition** A language \( L \) is \( P \)-complete if

a) \( L \) is in \( P \) and

b) Every language in \( P \) is reducible to \( L \) by a log-space DTM.

**Theorem** CIRCUIT VALUE is \( P \)-complete

**Proof** a) \( CV \) is in \( P \) because the value of a circuit can be computed from inputs in \( P \)-time. b) The above simulation demonstrates that a log-space DTM can translate any language \( L \in P \) to \( CV \).
Log-space Computations

• **Theorem** Log-space computations run in polynomial time.

• **Proof** Space $\leq a(\log n)$. $X$ = number configurations. $X \leq$ (no. input head positions) (no. strings of length $a(\log n)$ over alphabet of size $k$) (no. work tape head positions).

  $$X \leq O(n \ a(\log n) \ k^{a \ \log n}) = O(n^p)$$

  where $p = 1 + a(\log k) + \log(a(\log n))$. $X$ is polynomial in $n$.

• If a configuration repeats, the computation is in an infinite loop. **Thus, at most $X$ steps taken.**
Composition of Log-space Reductions
Composition of Log-space Reductions

• **Theorem** The composition of two log-space translators is a log-space translator.

• **Proof** Let $T_1$ and $T_2$ be log-space translators where output of $T_1$ goes directly to input to $T_2$. When $T_2$ needs another input, $T_1$ produces a new output.

• Counter$\_1$ holds $h_1$, $T_1$’s position on (virtual) Output$_\_Tape_1$

• If $T_2$ needs to move back one step, $h_1$ is decremented and $T_1$ re-simulates on original input to produce $h_1 - 1$ outputs. Since $T_1$ executes at most poly many steps, $|Counter_1| = O(\log n)$ space.
MONOTONE CV is P-Complete

• A monotone circuit uses only AND and OR.

MONOTONE CIRCUIT VALUE

• *Instance*: A description of a **monotone** circuit with fixed values for inputs variables and a designated output.

• *Answer*: “Yes” if the output has value 1.
Monotone CV is \( P \)-Complete

• **Theorem** MONOTONE CIRCUIT VALUE is \( P \)-complete

• **Proof** Convert from circuits with AND, OR and NOT to dual rail circuits; represent \( x \) by \((x, \bar{x})\). Then \( x \) AND \( y \) is replaced by \((x \text{ AND } y, \bar{x} \text{ OR } \bar{y})\) and OR is replaced by \((x \text{ OR } y, \bar{x} \text{ AND } \bar{y})\). NOT \( x \) is replaced by \((\bar{x}, x)\), obtained from input \((x, x)\) by interchanging the two inputs.

• Given a circuit \( C \), a monotone circuit \( M \) is produced by replacing each AND, OR and NOT step in \( C \) by two steps in \( M \), which can be done in log-space.
LINEAR INEQUALITIES

- **Instance:** An integer-valued $m \times n$ matrix $A$ and a column $m$-vector $b$.
- **Answer:** “Yes” if there is a rational column $n$-vector $x > 0$ such that $Ax \leq b$. 

© John E. Savage

CSCI 1010 Lect 24
LINEAR INEQUALITIES P-Complete

• **Theorem** LINEAR INEQUALITIES is P-complete
• **Proof** Reduce CV to LINEAR INEQUALITIES (LI). Convert gates and inputs in a circuit into a set of inequalities. (Note: a=b equiv. to a≤b, b≤a)

<table>
<thead>
<tr>
<th>Type</th>
<th>TRUE</th>
<th>FALSE</th>
<th>NOT</th>
<th>AND</th>
<th>OR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>$x_i = 1$</td>
<td>$x_i = 0$</td>
<td>$w = \neg u$</td>
<td>$w = u \land v$</td>
<td>$w = u \lor v$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Inequalities</th>
<th>$x_i = 1$</th>
<th>$x_i = 0$</th>
<th>$0 \leq w \leq 1$</th>
<th>$0 \leq w \leq 1$</th>
<th>$0 \leq w \leq 1$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$w = 1 - u$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$w \leq u$</td>
<td>$u \leq w$</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$w \leq v$</td>
<td>$v \leq w$</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$u + v - 1 \leq w$</td>
<td>$w \leq u + v$</td>
<td></td>
</tr>
</tbody>
</table>

• Can convert instance of CV to LI in log-space.
• A “Yes” instance of CV iff a “Yes” instance of LI.
Parallelizability of Problems in $P$

**Theorem** If a $P$-complete problem can be solved in poly-logarithmic time with polynomially many processors on a Concurrent Read, Exclusive Write PRAM (parallelizable random access machine), then so can all problems in $P$, that is, they are all fully parallelizable.
Summary

• The class $\mathbf{P}$.

• CIRCUIT VALUE

• Log-space computations and $\mathbf{P}$-complete langs.

• CIRCUIT VALUE is $\mathbf{P}$-complete.

• The composition of two log-space computations is log-space computable.

• MONOTONE CIRCUIT VALUE is $\mathbf{P}$-complete.

• LINEAR INEQUALITIES is $\mathbf{P}$-complete.

• Parallelizability of problems in $\mathbf{P}$.