CSCI 1010
Models of Computation

Lecture 10
First NP-Complete Language
Overview

• Definitions of **NP** and **NP**-complete restated
• CIRCUIT SAT revisited
• Simulating NTM by a circuit
• Simulating the tape and control units
• Reduction of recognition of language in **NP** to computation by a circuit.
• Proof that this reduction can be done polynomial time.
• CIRCUIT SAT is **NP**-complete.
Polynomial-Time Reduction

- **Definition:** A polynomial-time reduction (P-time) from language $L_1 \subseteq \Gamma^*$ to language $L_2 \subseteq \Sigma^*$ is a reduction $f : \Gamma^* \rightarrow \Sigma^*$ ($x \in L_1$ iff $f(x) \in L_2$) computable by a DTM in time polynomial in the length of its input. ($f$ P-time translates $L_1$ to $L_2$)

![Diagram](attachment:image.png)

The Recognizer for $L_1$ invokes the recognizer $R$ for $L_2$. 

© John E. Savage  
CSCI 1010 Lect 10
NP-Complete Language

• A language \( L \subseteq \Gamma^* \) is \textbf{NP-complete} if
  1. it is in \textbf{NP} and
  2. for every language \( L_0 \subseteq \Gamma^* \) in \textbf{NP}, there is a \textit{P}-time reduction \( f_0 \) from \( L_0 \) to \( L \).

• If only 2. holds, we say \( L \) is \textbf{NP-hard}.

Recognizer for \( L_0 \) in \textbf{NP} invokes recognizer \( R \) for \( L \).

Recognizer \( R \) for \( L \)
Is $P = NP$?

- Because an $NP$-complete language is in $NP$, it can be P-time reduced to another such language.
- Thus, an $NP$-complete language is a “hardest” language in $NP$ within polynomial bounds.

- If one $NP$-complete language recognizer requires exponential time, all do and $P \neq NP$.
  - Recall definition of a reduction.
- If one $NP$-complete language is in $P$, all are in $P$. That is, $P = NP$. 
Circuit Satisfiability

• A one-output circuit is **satisfiable** if its inputs can be chosen so that the output is 1.

• We generalize as follows:
  – A circuit is **satisfiable** if for fixed values of its *deterministic* inputs its *nondeterministic* inputs can be chosen so that the output is 1.
  – Previously all inputs were nondeterministic.

• CIRCUIT SAT is the set of satisfiable circuits.
A First \textbf{NP}-Complete Problem

• To show that language \( L \) is \textbf{NP}-complete we must show that it is in \textbf{NP} and every language in \textbf{NP} can be reduced to \( L \) in P-time.

• Our first \textbf{NP}-complete language is CIRCUIT SAT.
A First **NP**-Complete Problem

• A language $L$ in **NP** is specified by giving a polynomial $p(n)$ and an NTM $M_L$ such that $M_L$ non-deterministically recognizes $L$ in time $p(n)$ where $n$ is the length of the input string.

• We now give a P-time algorithm $A$ that given an $L_0$ in **NP** produces a circuit satisfiable on input $w$ iff $w \in L_0$. The “Yes” circuit instances of these circuits are in CIRCUIT SAT.
A First $\textbf{NP}$-Complete Problem

• To show that language $L$ is $\textbf{NP}$-complete we must show that it is in $\textbf{NP}$ and every language in $\textbf{NP}$ can be reduced to $L$ in P-time.

• Our first $\textbf{NP}$-complete language is CIRCUIT SAT.

• For arbitrary language $L_0$ in $\textbf{NP}$ recognized in P-time, we give an algorithm that prints an instance of a circuit that is satisfiable on NTM input $w$ iff $w \in L_0$. “Yes” circuit instances form CIRCUIT SAT.
Simulating NTM with a Circuit

• In Lecture 7 we simulated a DTM with a circuit!
• We now simulate NTM tape & control units with circuits.
• The combined circuit is satisfiable iff on input $w$ and with proper choice inputs the NTM can accept the input $w$. 

© John E. Savage
NTM Decomposition

- Control unit is NFSM
- Tape cells are FSMs
- To simulate by circuits unwind each cell loop.
- Assume T time steps and m = T memory cells.
Simulating an DFSM by a Circuit

- FSM computes $f^{(T)}$ in $T$ cycles. Unwind the loop!
Simulating an NFSM by a Circuit
Encoding of Tape Parameters

• Tape state at time t: \( A_t = (a_{0,t}, a_{1,t}, \ldots, a_{m-1,t}) \)
  – \( a_{i,t} \) is contents of ith cell at time t.
  – Each cell holds b bits, i.e. \( a_{i,t} = (a_{0,i,t}, a_{1,i,t}, \ldots, a_{b-1,i,t}) \)

• Head command: \( h = (h_{+1}, h_0, h_{-1}) \) – only one 1 in \( h \)

• Head pos.: \( s_t = (s_{0,t}, s_{1,t}, \ldots, s_{m-1,t}) \) – only one 1 in \( s_t \)

• Input to memory cell: \( w = (w_0, w_1, \ldots, w_{b-1}) \)

• Cell output: \( v_{j,t} = (v_{0,j,t}, v_{1,j,t}, \ldots, v_{b-1,j,t}) \) – 0 b-tuple if head not over jth cell at time t.
Tape Cell Circuit $C_{j,t}$

- Circuit $C_{j,t}$ receives input from adjacent cells $C_{j,t-1}, C_{j-1,t-1}, C_{j+1,t-1}$ as well as $w_t, s_{t-1}$ and $h_t$. It produces $a_{j,t}, v_{j,t}$, and $s_t$. 
Details of Cell Circuit Design

- $C_1$ changes cell contents if the head is over the cell.
- $C_2$ moves head if head over cell
- $C_3$ outputs cell contents if head over cell else 0.
- O(S) gates used in all cells, $S = mb$. 
Output of Tape Unit

- Output of tape unit is vector OR of cell outputs. O(S) gates used here.
- All but one cell output is vector 0.
Circuit Simulating T-step NTM

- One column/time step
- Inputs to first column are initial values.
- $c_j$ is choice input to control unit on jth step.
Size of Circuit Simulating T-step NTM

- Tape Unit: $O(S)$ gates per time step, $O(ST)$ total
- Control Unit: $O(1)$ gates/time step, $O(T)$ total
Translation to CIRCUIT SAT

• Let $L$ be in $\text{NP}$. There is an NTM $M_L$ that can accept every $w \in L$ in $p(n)$ time steps and accepts no $w \notin L$.

• $M_L$ description is used to construct a circuit with choice inputs that simulates $T = p(n)$ steps of $M_L$ on input $w$, $n = |w|$.

• Add circuit with output $= 1$ if $q_T$ is an accepting halt state and $0$ otherwise. Size is $O(1)$.

• Circuit is satisfied if and only if $w \in L$. 
Final Steps

• Given an L in \textbf{NP} that runs in $p(n)$ steps on accepted inputs $w$, $n = |w|$, we have shown that a circuit exists that is satisfiable iff $w \in L$.

• The circuit has size $O(ST)$. Since $S \leq T$, the size is $O(T^2) = O(p^2(n))$, a polynomial in $n$.

• Is there a program running in polynomial time that prints out the circuit description?
P-Time Program to Print Circuit

- A program that writes circuit simulating T steps of NTM.
- First n inputs specified.
- Set rest to blank.
- Write one copy of CU/time step
  - One copy of m-cell tape/step.
  - One m-input OR/step.
- Running time = $O(p(n))$ on RAM
CIRCUIT SAT is \textbf{NP}-Complete

• We have shown that recognition of an arbitrary language $L$ in \textbf{NP} can be reduced in polynomial time to recognition of CIRCUIT SAT.

• Since CIRCUIT SAT is in \textbf{NP}, as shown earlier, we conclude that CIRCUIT SAT is \textbf{NP}-complete.

• Later we use this result to prove that other languages are \textbf{NP}-complete.
Review

• Definitions of $\textbf{NP}$ and $\textbf{NP}$-complete restated
• CIRCUIT SAT revisited
• Simulating NTM by a circuit
• Simulating the tape and control units
• Reduction of recognition of language in $\textbf{NP}$ to computation by a circuit.
• Proof that this reduction can be done polynomial time.
• CIRCUIT SAT is $\textbf{NP}$-complete.