Lecture 27
Beyond NP

Many complexity classes are worse, e.g. time $2^{2^n}$, $2^{2^{2^n}}$, …

Others seem to be “worse” in a different sense, e.g., not in NP, but still exponential time. E.g., let

$L_p = \text{“assignment } y \text{ satisfies formula } x\text{”}, \in P$

Then:

$SAT = \{ x \mid \exists y \langle x,y \rangle \in L_p \}$

$UNSAT = \{ x \mid \forall y \langle x,y \rangle \notin L_p \}$

$QBF_k = \{ x \mid \exists y_1 \forall y_2 \exists y_3 \ldots \bigcirc_k y_k \langle x,y_1 \ldots y_k \rangle \in L_p \}$

$QBF_{\infty} = \{ x \mid \exists y_1 \forall y_2 \exists y_3 \ldots \langle x,y_1 \ldots \rangle \in L_p \}$
The “Polynomial Hierarchy”

\[ \Sigma^P_1 \text{ (NP): } \{ x | \exists y \langle x, y \rangle \in L_p \} \]
\[ \Pi^P_1 \text{ (co-NP): } \{ x | \forall y \langle x, y \rangle \in L_p \} \]

\[ \Delta^P_0: \text{ P } \]
\[ \Delta^P_1: \text{ P time given SAT } \]
\[ \Sigma^P_2: \{ x | \exists y \forall z \langle x, y, z \rangle \in L_p \} \]
\[ \Pi^P_2: \{ x | \forall y \exists z \langle x, y, z \rangle \in L_p \} \]

Potential Utility: It is often easy to give such a quantifier-based characterization of a language; doing so suggests (but doesn’t prove) whether it is in P, NP, etc. and suggests candidates for reducing to it.
QBF$_k$ in $\Sigma^p_k$

Given graph G, integers j & k, is there a set U of $\leq$ j vertices in G such that every k-clique contains a vertex in U?

Given graph G, integers j & k, is there a set U of $\geq$ j vertices in G such removal of any k edges leaves a Hamilton path in U?
Space Complexity

DTM $M$ has space complexity $S(n)$ if it halts on all inputs, and never visits more than $S(n)$ tape cells on any input of length $n$.

NTM …on any input of length $n$ on any computation path.

$\text{DSPACE}(S(n)) = \{ L \mid L \text{ acc by some DTM in space } O(S(n)) \}$

$\text{NSPACE}(S(n)) = \{ L \mid L \text{ acc by some NTM in space } O(S(n)) \}$
Model-independence

As with Time complexity, model doesn’t matter much. E.g.:

$\text{SPACE}(n)$ on DTM $\approx O(n)$ bytes on your laptop

Why? Simulate each by the other.
Space vs Time

Time $T \subseteq$ Space $T$

Pf: no time to use more space

Space $T \subseteq$ Time $2^{cT}$

Pf: if run longer, looping
Space seems more powerful

Intuitively, space is reusable, time isn’t

Ex.: SAT ∈ DSPACE(n)

Pf: try all possible assignments, one after the other

Even more:

\[ QBF_k = \{ \exists y_1 \forall y_2 \exists y_3 \ldots \bigcirc_k y_k x | \langle x, y_1 \ldots y_k \rangle \in L_p \} \in DSPACE(n) \]

\[ QBF_\infty = \{ \exists y_1 \forall y_2 \exists y_3 \ldots x | \langle x, y_1 \ldots \rangle \in L_p \} \in DSPACE(n) \]
PSPACE = Space(n^{O(1)})

NP ⊆ PSPACE

pf: depth-first search of NTM computation tree
Games

2 player “board” games
E.g., checkers, chess, tic-tac-toe, nim, go, …
A finite, discrete “game board”
Some pieces placed and/or moved on it
“Perfect information”: no hidden data, no randomness
Player I/Player II alternate turns
Defined win/lose configurations (3-in-a-row; checkmate; …)

Winning strategy:
\[
\exists \text{move by player I} \ \forall \text{moves by II} \ \exists \text{a move by I} \ \forall \ldots \ I \text{ wins.}
\]
Game Tree

Config:
Where are pieces
Relevant history
Who goes next

Play:
All moves

Win/lose: 1 1 0 1 1 1 1 1 0 0 0 1 0 1 1 0 1 0 1 0 1 0 1 0
Game Tree

Config:
Where are pieces
Relevant history
Who goes next

Play:
All moves

Win/lose: 1 1 0 1 1 1 1 1 0 1 0 0 1 0 1 1 0 1 0 1 0 1 0
Winning Strategy

Config:
- Where are pieces
- Relevant history
- Who goes next

Play:
- All moves

Win/lose: 1 1 0 1 1 1 1 1 0 1 0 0 1 0 0 1 1 1 1 1 1 0
Complexity of 2 person, perfect information games

From above, \textit{IF}

- config (incl. history, etc.) is poly size
- only poly many successors of one config
- each computable in poly time
- win/lose configs recognizable in poly time, and
- game lasts poly \# moves

\textit{THEN}

- in \textsc{PSPACE}!

\textit{Pf: depth-first search of tree, calc node values as you go.}