Ladner’s Theorem

- Assuming $P \neq NP$, what does the world (inside $NP$) look like?

\[ \text{NP: } \begin{array}{c} \text{P} \\ \text{NPC} \end{array} \]

- Our goal: $L \in NP$ that is neither in $P$ nor $NP$-complete.

\[ \text{Ladner’s Theorem} \hspace{1cm} \text{Theorem (Ladner): If } P \neq NP, \text{ then there exists } L \in NP \text{ that is neither in } P \text{ nor } NP\text{-complete.} \]

- Proof: “lazy diagonalization”
- deal with similar problem as in NTIME
  - Hierarchy proof

- Can enumerate (TMs deciding) all languages in $P$.
  - enumerate TMs so that each machine appears infinitely often
  - add clock to $M$, so that it runs in at most $n^i$ steps

- Can enumerate (TMs deciding) all $NP$-complete languages.
  - enumerate TMs $f$, computing all polynomial-time functions
  - machine $N_i$ decides language SAT reduces to via $f$, if $f$ is reduction, else SAT (details omitted...)

- Our goal: $L \in NP$ that is neither in $P$ nor $NP$-complete

\[ \begin{array}{c} \text{L} \\ \text{inputs} \end{array} \]
Ladner’s Theorem

- Top half, assuming $P \neq NP$:
  - focus on $M_i$
  - for any $x$, can always find some $z \geq x$ on which $M_i$ and SAT differ (why?)
  - input $x$

Ladner’s Theorem

- Bottom half, assuming $P \neq NP$:
  - focus on $N_i$
  - for any $x$, can always find some $z \geq x$ on which $N_i$ and TRIV differ (why?)

Ladner’s Theorem

- Try to “merge”:
  - on input $x$, either
    - answer SAT($x$)
    - answer TRIV($x$)
  - if can decide which one in $P$, $L \in NP$
Ladner’s Theorem

Key: $n$ eventually large enough to notice completed previous stage

- $f(0) = 0$
- $f(n)$: for $n$ steps compute $f(0), f(1), f(2), \ldots$

Inductive definition of $f(n)$

If $k = 2i$:

- for $n$ steps try (lex order) to find $z$ s.t. $\text{SAT}(z) \neq M_i(z)$ and $f(|z|)$ even
- if found, $f(n) = f(n-1)+1$ else $f(n-1)$

If $k = 2i + 1$:

- for $n$ steps try (lex order) to find $z$ s.t. $\text{TRIV}(z) \neq N_i(z)$ and $f(|z|)$ odd
- if found, $f(n) = f(n-1)+1$ else $f(n-1)$

Finishing up:

$L = \{ x | x \in \text{SAT if } f(|x|) \text{ even, } x \in \text{TRIV if } f(|x|) \text{ odd } \}$

$L \in \text{NP}$ since $f(|x|)$ can be computed in $O(n)$ time

Suppose $M$ decides $L$

- $f$ gets stuck at $2i$
- $L$ is SAT for $|z| > n_c$
- implies $\text{SAT} \in \text{P}$. Contradiction.

Suppose $N$ decides $L$

- $f$ gets stuck at $2i+1$
- $L$ is TRIV for $|z| > n_c$
- implies $\text{TRIV} \in \text{P}$. Contradiction.

A puzzle:

Two kinds of trees

- cover up nodes with $c$ colors
- promise: never color “arrow” same as “blank”
- determine which kind of tree in poly(n, c) steps?
A puzzle

Introduction

• Ideas
  – depth-first-search; stop if see arrow color
  – how many times may we see a given “arrow color”? 
  • at most n+1
  – pruning rule?
  • if see a color > n+1 times, it can’t be an arrow node; prune
  – # nodes visited before know answer?
  • at most c(n+2)

Sparse languages and NP

• We often say NP-compete languages are “hard”

• More accurate: NP-complete languages are “expressive”
  – lots of languages reduce to them

Sparse languages and NP

• Sparse language: one that contains at most poly(n) strings of length ≤ n
• not very expressive — can we show this cannot be NP-complete (assuming P ≠ NP)?
  – yes: Mahaney ’82 (homework problem)

• Unary language: subset of 1* (at most n strings of length ≤ n)

Sparse languages and NP

Theorem (Berman ’78): if a unary language is NP-complete then P = NP.

• Proof:
  – let U ⊆ 1* be a unary language and assume SAT ≤ U via reduction R
  – φ(x₁, x₂, ..., xₙ) instance of SAT
 Sparse languages and NP

- self-reduction tree for $\varphi$:

```
\varphi(x_1, x_2, \ldots, x_n)
\varphi(0, x_2, \ldots, x_n)
\varphi(1, x_2, \ldots, x_n)
\varphi(0, 0, \ldots, 0)
\varphi(1, 1, \ldots, 1)
```

satisfying assignment

 Sparse languages and NP

- applying reduction $R$:

```
R(\varphi(x_1, x_2, \ldots, x_n))
R(\varphi(1, x_2, \ldots, x_n))
R(\varphi(0, x_2, \ldots, x_n))
R(\varphi(0, 0, \ldots, 0))
R(\varphi(1, 1, \ldots, 1))
```

satisfying assignment

 Sparse languages and NP

- on input of length $m = |\varphi(x_1, x_2, \ldots, x_n)|$,$R$ produces string of length $\leq p(m)$
- $R$’s different outputs are “colors”
  - 1 color for strings not in 1
  - at most $p(m)$ other colors
- puzzle solution ⇒ can solve SAT in $\text{poly}(p(m)+1, n+1) = \text{poly}(m)$ time!

Summary

- nondeterministic time classes:
  - $\text{NP}$, $\text{coNP}$, $\text{NEXP}$
- NTIME Hierarchy Theorem:
  - $\text{NP} \neq \text{NEXP}$
- major open questions:
  - $P \overset{?}{=} \text{NP}$
  - $\text{NP} \overset{?}{=} \text{coNP}$

Summary

- $\text{NP}$-“intermediate” problems (unless $P = \text{NP}$)
  - technique: delayed diagonalization
- unary languages not $\text{NP}$-complete (unless $P = \text{NP}$)
  - true for sparse languages as well (homework)
- complete problems:
  - circuit SAT is $\text{NP}$-complete
  - UNSAT is $\text{coNP}$-complete
  - succinct circuit SAT is $\text{NEXP}$-complete

Summary
Remainder of lecture

• Nondeterminism applied to space
• Reachability
• Two surprises:
  – Savitch’s Theorem
  – Immerman/Szelepcsényi Theorem

Nondeterministic space

• \( \text{NSPACE}(f(n)) \) = languages decidable by a multi-tape NTM that touches at most \( f(n) \) squares of its work tapes along any computation path, where \( n \) is the input length, and \( f: \mathbb{N} \rightarrow \mathbb{N} \)

Nondeterministic space

• Robust nondeterministic space classes:
  \( \text{NL} = \text{NSPACE}(\log n) \)
  \( \text{NPSPACE} = \bigcup_k \text{NSPACE}(n^k) \)

Reachability

• Recall: at most \( n^k \) configurations of given NTM \( M \) running in \( \text{NSPACE}(\log n) \).
  • Easy to determine if \( C \) yields \( C' \) in one step
  • Configuration graph for \( M \) on input \( x \):

Reachability

• Conclude: \( \text{NL} \subseteq \text{P} \)
  – and \( \text{NPSPACE} \subseteq \text{EXP} \)

• S-T-Connectivity (STCONN): given directed graph \( G = (V, E) \) and nodes \( s, t \), is there a path from \( s \) to \( t \) ?
  \[ \text{Theorem: STCONN is NL-complete under logspace reductions.} \]

Reachability

• Proof:
  – In \( \text{NL} \): guess path from \( s \) to \( t \) one node at a time
  – Given \( L \in \text{NL} \) decided by NTM \( M \) construct configuration graph for \( M \) on input \( x \) (can be done in logspace)
  – \( s \) = starting configuration; \( t = q_{\text{accept}} \)
Two startling theorems

• Strongly believe $P \neq \text{NP}$
• Nondeterminism seems to add enormous power
• for space: Savitch '70:
  \[ \text{NPSPACE} = \text{PSPACE} \]
  \[ \text{NPSPACE} = \text{PSPACE} \]  
  \[ \text{NL} \subseteq \text{SPACE}(\log^2 n) \]

Two startling theorems

• Strongly believe $\text{NP} \neq \text{coNP}$
• Seems impossible to convert existential into universal
• for space: Immerman/Szelepscényi '87/88:
  \[ \text{NL} = \text{coNL} \]