A puzzle:
- 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
A puzzle
Introduction

• Ideas
  – depth-first-search; stop if see 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 \textbf{NP}

• We often say \textbf{NP}-complete languages are “hard”

• More accurate: \textbf{NP}-complete languages are “expressive”
  – lots of languages reduce to them
Sparse languages and \textbf{NP}

- **Sparse language**: one that contains at most $\text{poly}(n)$ strings of length $\leq n$
- not very expressive – can we show this cannot be \textbf{NP}-complete (assuming $P \neq \text{NP}$)?
  - yes: Mahaney ’82 (homework problem)

- **Unary language**: subset of $1^*$ (at most $n$ strings of length $\leq n$)
Sparse languages and \textbf{NP}

\textbf{Theorem} (Berman ’78): if a unary language is \textbf{NP}-complete then $P = \textbf{NP}$.

• Proof:
  – let $U \subseteq 1^*$ be a unary language and assume $\text{SAT} \leq U$ via reduction $R$
  – $\varphi(x_1,x_2,\ldots,x_n)$ instance of $\text{SAT}$
Sparse languages and \textbf{NP}

– self-reduction tree for $\varphi$:

$$
\begin{align*}
&\varphi(x_1, x_2, \ldots, x_n) \\
&\varphi(0, x_2, \ldots, x_n) \quad \varphi(1, x_2, \ldots, x_n) \\
&\vdots \\
&\varphi(0, 0, \ldots, 0) \quad \text{gray} \\
&satisfying\ assignment \\
&\varphi(1, 1, \ldots, 1)
\end{align*}
$$
Sparse languages and \textbf{NP}

– applying reduction \( R \):

\[
R(\phi(x_1,x_2,\ldots,x_n)) \\
R(\phi(0,x_2,\ldots,x_n)) \\
R(\phi(0,0,\ldots,0)) \\
R(\phi(1,1,\ldots,1))
\]

satisfying assignment
Sparse languages and \textbf{NP}

- on input of length $m = |\phi(x_1,x_2,\ldots,x_n)|$, $R$ 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 $\Rightarrow$ can solve SAT in $\text{poly}(p(m)+1, n+1) = \text{poly}(m)$ time!
Summary

• nondeterministic time classes: 
  \( NP, \ coNP, \ NEXP \)
• NTIME Hierarchy Theorem: 
  \( NP \neq NEXP \)
• major open questions: 
  \( P \overset{?}{=} NP \quad NP \overset{?}{=} coNP \)
Summary

• **NP-**“intermediate” problems (unless $P = NP$)
  – technique: delayed diagonalization

• unary languages not **NP**-complete (unless $P = NP$)
  – true for sparse languages as well (homework)

• complete problems:
  – circuit SAT is **NP**-complete
  – UNSAT is **coNP**-complete
  – succinct circuit SAT is **NEXP**-complete
Summary

NEXP \rightarrow \text{EXP} \rightarrow \text{PSPACE} \rightarrow \text{NP} \rightarrow \text{P} \rightarrow \text{L}

\text{coNEXP} \rightarrow \text{coNP}
Remainder of lecture

• nondeterminism applied to space
• reachability
• two surprises:
  – Savitch’s Theorem
  – Immerman/Szelepcsényi Theorem
Nondeterministic space

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

April 12, 2017
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 $t$?

**Theorem**: STCONN is $\text{NL}$-complete under logspace reductions.
Reachability

• Proof:
  – in $\textbf{NL}$: guess path from $s$ to $t$ one node at a time
  – given $L \in \textbf{NL}$ 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 $P \neq NP$
- Nondeterminism seems to add enormous power
- For space: Savitch ‘70:
  \[
  \text{NPSPACE} = \text{PSPACE}
  \]
  and
  \[
  \text{NL} \subset \text{SPACE}(\log^2 n)
  \]
Two startling theorems

• Strongly believe $\text{NP} \neq \text{coNP}$
• Seems impossible to convert existential into universal

• For space: Immerman/Szelepscényi ’87/’88:

$$\text{NL} = \text{coNL}$$
Savitch’s Theorem

**Theorem:** \( \text{STCONN} \in \text{SPACE}(\log^2 n) \)

- Corollary: \( \text{NL} \subset \text{SPACE}(\log^2 n) \)
- Corollary: \( \text{NPSPACE} = \text{PSPACE} \)
Proof of Theorem

– input: $G = (V, E)$, two nodes $s$ and $t$
– recursive algorithm:

```c
/* return true iff path from x to y of length at most $2^i$ */
PATH(x, y, i)
if i = 0 return ( x = y or (x, y) ∈ E ) /* base case */
for z in V
    if PATH(x, z, i-1) and PATH(z, y, i-1) return(true);
return(false);
end
```
Proof of Theorem

– answer to STCONN: \( \text{PATH}(s, t, \log n) \)
– space used:
  • \((\text{depth of recursion}) \times (\text{size of “stack record”})\)
– depth = \(\log n\)
– claim stack record: “\((x, y, i)\)” sufficient
  • size \(O(\log n)\)
– when return from \(\text{PATH}(a, b, i)\) can figure out what to do next from record \((a, b, i)\) and previous record

April 12, 2017
Nondeterministic space

- Robust nondeterministic space classes:

\[ \text{NL} = \text{NSPACE}(\log n) \]

\[ \text{NPSPACE} = \bigcup_k \text{NSPACE}(n^k) \]
Second startling theorem

• Strongly believe $\text{NP} \neq \text{coNP}$
• seems impossible to convert existential into universal

• for space: Immerman/Szelepscényi ’87/’88:

$$\text{NL} = \text{coNL}$$
I-S Theorem

**Theorem**: \( \text{ST-NON-CONN} \in \text{NL} \)

- Proof: slightly tricky setup:
  - input: \( G = (V, E) \), two nodes \( s, t \)

```
  s
  ↑
  "yes"

  t
```

```
  s
  ↑
  "no"

  t
```
I-S Theorem

– want nondeterministic procedure using only $O(\log n)$ space with behavior:
I-S Theorem

– observation: given \textbf{count} of # nodes reachable from \(s\), can solve problem
  \begin{itemize}
    \item for each \(v \in V\), \textit{guess} if it is reachable
    \item if yes, \textit{guess} path from \(s\) to \(v\)
      \begin{itemize}
        \item if guess doesn’t lead to \(v\), reject.
        \item if \(v = t\), reject.
        \item else counter++
      \end{itemize}
    \item if counter = \textbf{count} accept
  \end{itemize}
I-S Theorem

– every computation path has sequence of guesses…

– only way computation path can lead to accept:
  • correctly guessed reachable/unreachable for each node v
  • correctly guessed path from s to v for each reachable node v
  • saw all reachable nodes
  • t not among reachable nodes
I-S Theorem

- \( R(i) \) = number of nodes reachable from \( s \) in at most \( i \) steps
- \( R(0) = 1 \): node \( s \)

- we will compute \( R(i+1) \) from \( R(i) \) using \( O(\log n) \) space and nondeterminism

- computation paths with “bad guesses” all lead to reject
I-S Theorem

– Outline: in n phases, compute
  \[ R(1), R(2), R(3), \ldots R(n) \]
– only \( O(\log n) \) bits of storage between phases
– in end, lots of computation paths that lead to reject
– only computation paths that survive have computed correct value of \( R(n) \)
– apply observation.
I-S Theorem

– computing $R(i+1)$ from $R(i)$:

– Initialize $R(i+1) = 0$

– For each $v \in V$, *guess* if $v$ reachable from $s$ in at most $i+1$ steps

$R(i) = R(2) = 6$
I-S Theorem

– if “yes”, guess path from s to v of at most i+1 steps. Increment R(i+1)
– if “no”, visit R(i) nodes reachable in at most i steps, check that none is v or adjacent to v
  • for u ∈ V guess if reachable in ≤ i steps; guess path to u; counter++
  • KEY: if counter ≠ R(i), reject
• at this point: can be sure v not reachable
I-S Theorem

• correctness of procedure:
• two types of errors we can make
• (1) might guess v is reachable in at most \(i + 1\) steps when it is not
  – won’t be able to guess path from s to v of correct length, so we will reject.

“easy” type of error
I-S Theorem

• (2) might guess $v$ is **not** reachable in at most $i+1$ steps when it is
  – then must **not** see $v$ or neighbor of $v$ while visiting nodes reachable in $i$ steps.
  – but forced to visit $R(i)$ distinct nodes
  – therefore must try to visit node $v$ that is **not** reachable in $\leq i$ steps
  – won’t be able to guess path from $s$ to $v$ of correct length, so we will reject.

“easy” type of error
Summary

• nondeterministic space classes \textbf{NL} and \textbf{NPSPACE}

• ST-CONN \textbf{NL}-complete
Summary

• Savitch: \textbf{NPSPACE} = \textbf{PSPACE}
  – Proof: \text{ST-CONN} \in \text{SPACE}(\log^2 n)
  – open question:
    \textbf{NL} = \textbf{L}?

• Immerman/Szelepcsényi: \textbf{NL} = \textbf{coNL}
  – Proof: \text{ST-NON-CONN} \in \textbf{NL}