Last time: relations between different resources

**Theorem:** For $f$ a p.c.f, have:

1. $\text{TIME}(f(n)) \leq \text{SPACE}(f(n))$.
2. $\text{TIME}(f(n)) \leq \text{NTIME}(f(n))$
3. $\text{SPACE}(f(n)) \leq \text{NSPACE}(f(n)) \leq \text{SPACE}(f(n)^2)$

**Theorem:** For $f(n)$ a p.c.f, have

$$\text{NTIME}(f(n)) \leq \bigvee_{k \geq 1} \text{TIME}(K^{f(n)k})$$

**Theorem:** For $f(n)$ a p.c.f, have

$$\text{NSPACE}(f(n)) \leq \bigvee_{k \geq 1} \text{TIME}(K^{f(n) + \log n})$$

$config. graph G_{m \times n}, \text{REACHABILITY}$

Today: space complexity

(AB Chap. 4, Sipser Chap. 8, Cai: Chap. 3)

Nondet. space: Switch's Theorem

$\text{NL, } \text{NL-completeness}$

Questions?

Recall space basics:

(iso space #4)
- $L \in \text{SPACE}(f(n))$ if there's a multi-tape TM that decides $L$ and has space exactly $f(n)$.

- 1 tape
  - happy birthday
  - $n = 1 \times 1$

- C-tape:
  - happy birthday

- 2 work tapes
  - $f(n)$

- Linear compressor for space: $L \in \text{SPACE}(f(n)) \Rightarrow$
  - for any $\varepsilon > 0$, $L$ also $\in \text{SPACE}(\varepsilon \cdot f(n))$.
  - really $\text{SPACE}(\max\{1, \varepsilon \cdot f(n)\})$

- For TMs with output, the output tape (output length)
  - doesn't "count" as space usage.

- Ex: TM for addition of 2 $n$-bit #s:
  - output length $= \Theta(n)$, space used $= O(\log n)$

Interesting space classes: $\log n$ up.

- $O(\log n)$ space: can't even store 2 pters into input, or count to $n$, write name of a single
O(\log n) space: can keep O(1) ptrs into input/vertices of G "in mind".

Key space classes:

\[ L = \text{SPACE}(\log n) \]
\[ NL = \text{NSPACE}(\log n) \]
\[ \text{PSPACE} = \bigcup_{k \geq 1} \text{SPACE}(n^k) \]
\[ \text{NPSPACE} = \bigcup_{k \geq 1} \text{NSPACE}(n^k) \]

SPACE CAN BE REUSED

Nondet. space

Switch Thm: For any p.c.f. \( f(n) \geq \log n \), have \( \text{NSPACE}(f(n)) \subseteq \text{SPACE}(f(n)^2) \).

\[ \text{PSPACE} = \text{NPSPACE}; \]
\[ NL \subseteq \text{SPACE}(\log(n)^3) \subseteq \text{PSPACE}. \]

Study \text{REACH}. Recall \text{REACH} \in \text{P}, (BFS, DFS, "mark reachable nodes" algs.) these are linear-space algs. Stronger result:
Claim: \( \text{REACH \in NL} \).

**Pf:** Guess path from \( s \) to \( t \) of length \( \leq n \), check that it's okay.

More detail: NTH with 3 tapes.

- \( T_1 \): Holds current node \( i \) in \( O(\log n) \) space suff.
- \( T_2 \): Machine nondet. writes node \( j \).
- Scans input to verify that \( i \rightarrow j \) edge is in \( G \).
- Rej: if not, if is, replace \( i \) on \( T_1 \) with \( j \).
- \( T_3 \): Keep counter of # times curr. node updated, rej: if reaches \( n+1 \).

Next step:

Claim: \( \text{REACH \in SPACE}(\log(n)^2) \).

**Pf:** Input is \( (G,s,t) \) \( G = (V,E) \) n-node digraph.

Define

\[
\text{PATH}(x,y,i) = \begin{cases} 
\text{TRUE} & \text{if } G \text{ has } x \rightarrow y \text{ path of length } \leq i \\
\text{FALSE} & \text{otherwise.}
\end{cases}
\]

\( G \in \text{REACH} \iff \text{PATH}(s,t,\log n) = \text{TRUE}. \)

How to decide \( \text{PATH}(x,y,i) = \text{TRUE} \) in small space?
Recursion + midpoints.

Key idea:
There exists a path of length $\leq 2^i$ in $G$ iff for some $z$, there exist paths each of length $\leq 2^{i-1}$.

Rec alg for $PATH(x,y,i)$:

PATH $(x,y,i)$
\begin{enumerate}
  \item if $i = 0$ then if $G$ has $x \to y$ edge return TRUE
  \item else return FALSE
  \item if $i > 0$:
  \begin{enumerate}
    \item for each vertex $z \in V$
      \begin{enumerate}
        \item if $PATH(x,z,i-1) \land PATH(z,y,i-1)$ return TRUE
        \item at end of loop (no $z$ worked), return FALSE
      \end{enumerate}
  \end{enumerate}
\end{enumerate}

This can be run in $log(n)^2$ space ($T_M$ with 2 worktapes):

tape 1: stack stores rec. call history.
Contains some triples $(x,z,i)$ each of length $\leq 3 \log n$.

<table>
<thead>
<tr>
<th>$(x,y,i)$</th>
<th>$(x,z,i-1)$</th>
</tr>
</thead>
</table>

Reuse space when you're checking $z$s.
One $M$ has determined truth value of $PATH(x,z,i-1)$:
- if $T$, erase $(x,z,i-1)$ and write $(z,y,i-1)$ and check that.
  If $PATH(z,y,i-1)$ is T, erase $(z,y,i-1)$.
M knows \( \text{PATH}(x, y, i) \) (so that gets erased...)

- If \( \text{PATH}(x, z, i) \) is \( F \), erase \( + \) try \( (x, z', i-1) \) (next \( z' \))

Depth of stack (# levels of recursion) \( \leq i \).

So space used \( \leq i \cdot (3 \log n) \) on this tape.

\( G \leq O(\log^2 n) \) for \( i = \log n \).

2nd tape: scratch space for counters through nodes of \( G \); input tape position, etc. \( O(1) \) counters, \( O(\log n) \) space.

---

Runtime of \( \alpha \)?

\[ T(i) = \text{time for } \text{PATH}(x, y, i) \]

\[ T(i) \leq 2n \cdot T(i-1) \quad T(n) \leq (2n)^i. \]

\( i = \log n \)  

\[ T(\log n) = (2n)^{\log n}. \]

---

Is there a poly \( (\cdot) \) time, \( \log(n)^2 \) space \( \alpha \)

for \( \text{REACH} \)?

---

Switch Thm: For any p.c.f. \( f(n) > \log n \), have \( \text{NSPACE}(f(n)) \leq \text{SPACE}(f(n)^2) \).

---

Pf: Let \( M \) be \( f(n) \)-space NTM for \( \text{LENSPACE}(f(n)) \)

\[ G_{M, x} \xrightarrow{1 \leq \gamma} \text{config graph: } \leq N = k^{f(n)} \] nodes.

Run our \( \text{REACH} \) alg on \( G_{M, x} \); it needs space

\[ \log(N)^2 = \log(k^{f(n)})^2 = (f(n) \log k)^2 = O(f(n)^2). \]
PATH (C_{init}, C_{core}, \log (k^f(n)))

Note: input string is x, not G_{x, x}. No problem: our \log(n)^2 space algo only consults G to check if x \rightarrow y (two nodes conn.) in i = 0 case.

Given 2 configs C_1, C_2 (each of length O(f(n))) can decide if M yields C_2 from C_1 in one step by "looking at" C_1, C_2, transit. rule of M.

So can run this algo. in O(f(n)^2) space.

Does this use that f(n) is a p.a.c.f.? How did we "know" \log (k^f(n)) (param. for PATH)? Use P.C.F. to first compute f(kx) = f(x), use that value for...

(Can also do the kind of thing from last time: try f(n) = 1, 2, 3, ..., works at right value.)

\underline{NL Completeness}

Readings: AB Ch. 4.3
Papad. Ch. 16.1
Sipser 8.4, 8.5

Q: is L = NL?

probably... not?
Believed REACH \notin L. Can’t answer this, but can simplify landscape: analog. to NPC theory.
there are problems complete for NL: if anyone of them is in \( L \), then \( \text{NL} \subseteq \text{L} \).

\( \text{REACH} \) is such a problem.

Next time: log space reductions.

\( \text{NL completeness} \)

\( \text{REACH is NL-complete}. \)

\( \text{PSPACE completeness}. \)