18.404/6.840 Lecture 21

Last time:
- Log-space reducibility
- L = NL? question
- $PATH$ is NL-complete
- $\overline{2SAT}$ is NL-complete
- NL = coNL (unfinished)

Today: (Sipser §9.1)
- Finish NL = coNL
- Time and Space Hierarchy Theorems

NL = coNL (part 1/4)

Theorem (Immerman-Szelepcsényi): NL = coNL

Proof: Show $\overline{PATH} \in$ NL

Defn: NTM M computes function $f\colon \Sigma^* \to \Sigma^*$ if for all w

1) All branches of M on w halt with $f(w)$ on the tape or reject.
2) Some branch of M on w does not reject.

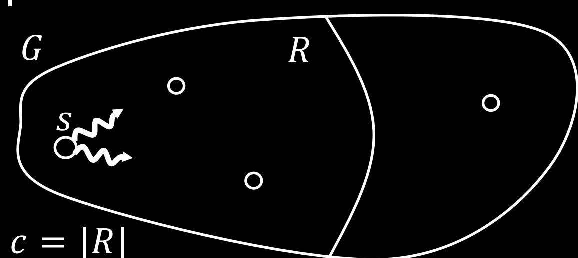
Let $path(G, s, t) = \begin{cases} \text{YES}, & \text{if } G \text{ has a path from } s \text{ to } t \\ \text{NO}, & \text{if not} \end{cases}$

Let $R = R(G, s) = \{u \mid path(G, s, u) = \text{YES}\}$

Let $c = c(G, s) = |R|$

R = Reachable nodes

c = # reachable



Check-in 21.1

Let G be the graph below.

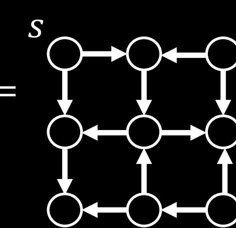
What is the value of $c = c(G, s)$?

(a) 2 (e) 6

(b) 3 (f) 7

(c) 4 (g) 8

(d) 5 (h) 9

$G =$ 

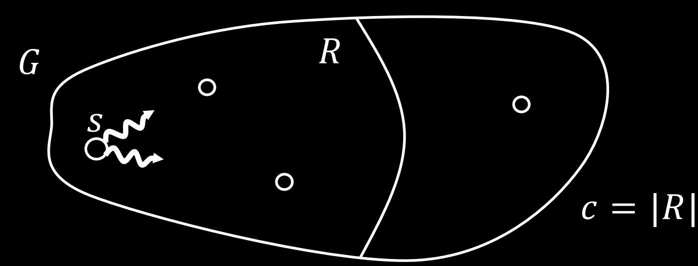
Check-in 21.1

NL = coNL (part 2/4) – key idea

Theorem: If some NL-machine computes c, then some NL-machine computes $path$.

Proof: "On input $\langle G, s, t \rangle$ where G has m nodes

1. Compute c
2. $k \leftarrow 0$
3. For each node u
4. Nondeterministically go to (p) or (n)
   (p) Nondeterministically pick a path from s to u of length $\leq m$.
   If fail, then *reject*.
   If $u = t$, then output YES, else set $k \leftarrow k + 1$.
   (n) Skip u and continue.
5. If $k \neq c$ then *reject*.
6. Output NO." [found all c reachable nodes and none were t}

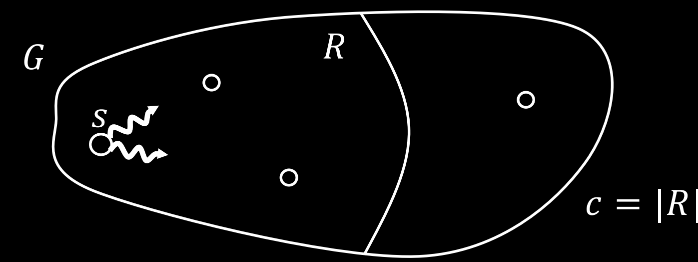


NL = coNL (part 2/4) – key idea SIMPLIFIED!!

Theorem: If some NL-machine computes c, then some NL-machine computes $path$.

Proof: "On input $\langle G, s, t\rangle$ where G has m nodes

1. Compute c
2. $k \leftarrow 0$
3. For each node u
4. Nondeterministically pick a path from s of length $\leq m$.
   If it ends at t then output YES and stop.
   If it ends at u, set $k \leftarrow k + 1$.
5. If $k \neq c$ then *reject*.
6. Output NO." [found all c reachable nodes and none were t}



NL = coNL (part 3/4)

Let $path_d(G,s,t) = \begin{cases} \text{YES}, & \text{if } G \text{ has a path } s \text{ to } t \text{ of length} \leq d \\ \text{NO}, & \text{if not} \end{cases}$

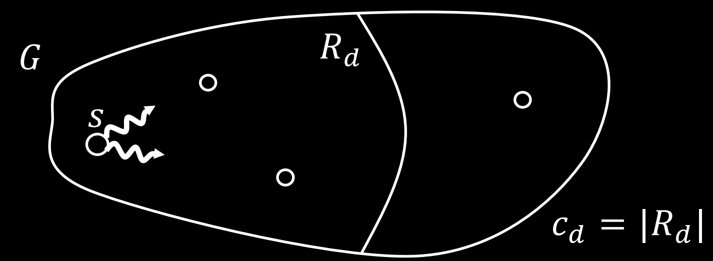
Let $R_d = R_d(G,s) = \{u |\ path_d(G,s,u) = \text{YES}\}$

Let $c_d = c_d(G,s) = |R_d|$

Theorem: If some NL-machine computes c_d, then some NL-machine computes $path_d$.

Proof: "On input $\langle G,s,t \rangle$

1. Compute c_d
2. $k \leftarrow 0$
3. For each node u
4. Nondeterministically go to (p) or (n)
   - (p) Nondeterministically pick a path from s to u of length $\leq d$.
     If fail, then *reject*.
     If $u = t$, then output YES, else set $k \leftarrow k + 1$.
   - (n) Skip u and continue.
5. If $k \neq c_d$ then *reject*.
6. Output NO" [found all c_d reachable nodes and none were t}



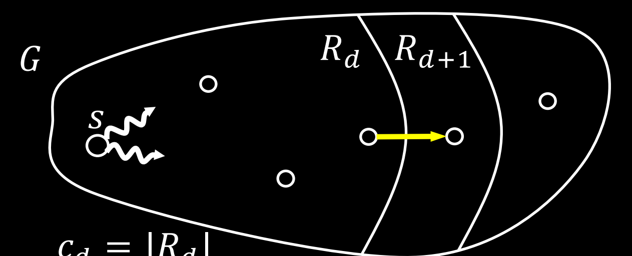
NL = coNL (part 4/4)

Theorem: If some NL-machine computes c_d, then some NL-machine computes $path_{d+1}$.

Proof: "On input $\langle G, s, t\rangle$

1. Compute c
2. $k \leftarrow 0$
3. For each node u
4. Nondeterministically go to (p) or (n)
   - (p) Nondeterministically pick a path from s to u of length $\leq d$.
     If fail, then *reject*.
     If u has an edge to t, then output YES, else set $k \leftarrow k + 1$.
   - (n) Skip u and continue.
5. If $k \neq c_d$ then *reject*.
6. Output NO." [found all c_d reachable nodes and none had an edge to t]

Corollary: Some NL-machine computes c_{d+1} from c_d.



Hence $\overline{PATH} \in$ NL

"On input $\langle G, s, t\rangle$

1. $c_0 = 1$.
2. Compute each c_{d+1} from c_d for $d = 1$ to m.
3. *Accept* if $path_m(G, s, t)$ = NO.
4. *Reject* if $path_m(G, s, t)$ = YES."

Review: Major Complexity Classes

$\mathrm{L} \subseteq \mathrm{NL} \subseteq \mathrm{P} \subseteq \mathrm{NP} \subseteq \mathrm{PSPACE}$

$\neq$

Today

The time and space hierarchy theorems show that
if a TM is given more time (or space) then it can do more.*
* certain restrictions apply.

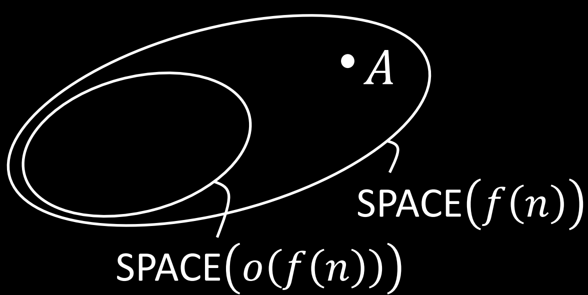
For example:
$\mathrm{TIME}(n^2) \subsetneq \mathrm{TIME}(n^3)$ [$\subsetneq$ means proper subset]
$\mathrm{SPACE}(n^2) \subsetneq \mathrm{SPACE}(n^3)$

Space Hierarchy Theorem (1/2)

Theorem: For any $f: \mathbb{N} \to \mathbb{N}$ (where f satisfies a technical condition)
there is a language A where A requires $O(f(n))$ space, i.e,

1) A is decidable in $O(f(n))$ space, and
2) A is not decidable in $o(f(n))$ space

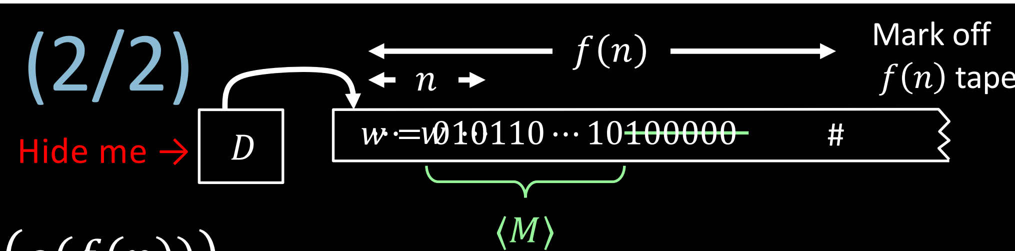
On other words, SPACE$(o(f(n))) \subsetneq$ SPACE$(f(n))$
Notation: SPACE$(o(f(n))) = \{B \mid$ some TM M decides B in space $o(f(n))\}$



Proof outline: (Diagonalization)
Give TM D where
1) D runs in $O(f(n))$ space
2) D ensures that $L(D) \neq L(M)$ for every TM M that runs in $o(f(n))$ space.

Let $A = L(D)$.

Space Hierarchy Theorem (2/2)



Goal: Exhibit $A \in \text{SPACE}(f(n))$ but $A \notin \text{SPACE}(o(f(n)))$

Give D where $A = L(D)$ and

1) D runs in $O(f(n))$ space
2) D ensures that $L(D) \neq L(M)$ for every TM M that runs in $o(f(n))$ space.

D = "On input w

1. Mark off $f(n)$ tape cells where $n = |w|$. If ever try to use more tape, *reject*.
2. If $w \neq \langle M \rangle$ for some TM M, *reject*.
3. Simulate* M on w. *Accept* if M rejects, *Reject* if M accepts

*Note: D can simulate M with a constant factor space overhead.

Issues:

1. What if M runs in $o(f(n))$ space but has a big constant? Then D won't have space to simulate M when w is small. FIX: simulate M on infinitely many w.

Check-in 21.2

What happens when we run D on input $\langle D \rangle 1000000$?

a) It loops
b) It *accepts*
c) It *rejects*
d) We get a contradiction
e) Smoke comes out

Check-in 21.2

Time Hierarchy Theorem (1/2)

Theorem: For any $f: \mathbb{N} \rightarrow \mathbb{N}$ where f is time constructible
there is a language A where A requires $O(f(n))$ time, i.e,

1) A is decidable in $O(f(n))$ time, and
2) A is not decidable in $o(f(n)/\log(f(n)))$ time

On other words, TIME$\left(o\left(\frac{f(n)}{\log(f(n))}\right)\right) \subsetneq$ TIME$(f(n))$

Proof outline: Give TM D where

1) D runs in $O(f(n))$ time
2) D ensures that $L(D) \neq L(M)$ for every TM M that runs in $o(f(n)/\log(f(n)))$ time .

Let $A = L(D)$.

Time Hierarchy Theorem (2/2)

Goal: Exhibit $A \in \text{TIME}\big(f(n)\big)$ but $A \notin \text{TIME}\Big(o\big(f(n)/\log\big(f(n)\big)\big)\Big)$

$A = L(D)$ where

1) D runs in $O\big(f(n)\big)$ time
2) D ensures that $L(D) \neq L(M)$ for every TM M that runs in $o\big(f(n)/\log\big(f(n)\big)\big)$ time.

D = "On input w

1. Compute $f(n)$.
2. If $w \neq \langle M \rangle 10^*$ for some TM M, *reject*.
3. Simulate* M on w for $f(n)/\log\big(f(n)\big)$ steps.

   Accept if M rejects,

   Reject if M accepts or hasn't halted."

*Note: D can simulate M with a log factor time overhead due to the step counter.

Why do we lose a factor of $\log\big(f(n)\big)$?

D must halt within $O\big(f(n)\big)$ time.
To do so, D counts the number of steps it uses and stops if the limit is exceeded. The counter has size $\log\big(f(n)\big)$ and is stored on the tape.
It must be kept near the current head location.
Cost of moving it adds a $O\big(\log\big(f(n)\big)\big)$ overhead factor. So to halt within $O\big(f(n)\big)$ time, D stops when the counter reaches $f(n)/\log\big(f(n)\big)$.

Recap: Separating Complexity Classes

$$L \subseteq NL \subseteq P \subseteq NP \subseteq PSPACE$$

$\neq$

Space Hierarchy Theorem

$$NL \subseteq SPACE(\log^2 n) \subsetneq SPACE(n) \subseteq PSPACE$$

Check-in 21.3

Consider these two famous unsolved questions:

1. Does L = P?
2. Does P = PSPACE?

What do the hierarchy theorems tell us about these questions?

a) Nothing
b) At least one of these has answer “NO”
c) At least one of these has answer “YES”

Check-in 21.3

Quick review of today

1. Finish NL = coNL
2. Space hierarchy theorem
3. Time hierarchy theorem