Last time: \(QBF, \overline{QBF} \) are PSPACE-complete. 
\[ \exists x_1 \forall x_2 \exists x_3 \ldots \forall x_n \ \varphi(x_1, \ldots, x_n) \]
\( \rightarrow \text{game on graphs} \)

Today: • IS thus: nondet space closed under complementation

Readings: AB 4.3.2, Papadimitriou 7.3, Sipser 8.6, Cai 3.3

• start randomness unit (#5) prob. basics

Readings: Cai 5.1, Wikipedia

Admin: No OH next week (midterm week)

Midterm released Sun, due end of Fri next week.

Questions?

\[ f(i) = 2f(i-1) \quad f(i) = C + f(i-1) \]

Nondet. space is closed under complement (1)

Recall: • det classes like \(P, L, \text{PSPACE}\) is closed under complement. \(P = \text{co-P} \subseteq \text{co-L}\) etc.

• we \textit{REALLY} think \(NP \neq \text{co-NP}\).

What about space classes? \(NL\) ? First conjecture...
Then: (I T '87) Let \( f(n) \geq \log n \) be a p.c.f. Then \( \text{NSPACE}(f(n)) = \text{coNSPACE}(f(n)) \).

\[ \text{Pf: Fix } L \in \text{NSPACE}(f(n)). \]

Let \( M \) be NTM, \( f(n) \)-space, dec. \( L \).

We'll describe NTM \( N \), \( O(f(n)) \)-space, s.t.

\[ \forall x, N \text{ acc. } x \text{ (on some path)} \iff \text{every comp. path of } M \text{ on } x \text{ rejects}. \]

This means \( N \) decides \( L \).

\[ \text{Fix } \cdot |x| = n \]

\[ \text{Let } \cdot w = c \cdot f(n), c = \text{const s.t. } w > \# \text{config of } M \text{ on } x. \text{ (Note runtime of } M \text{ on } x \text{, on any path, } \leq w. \) \]

\[ \text{Let } \cdot s = \text{init config of } M \text{ on } x \]

\[ \text{Let } \cdot t = 1 \text{ acc config of } M \text{ on } x \text{ (on input } x) \]

\[ \text{Let } \cdot l = \# \text{ configs reachable from } s \text{ along some path.} \]

\[ \text{Let alg: Here's how an NTM that is given } L \text{ as input can correctly determine whether } M \text{ rejects } x \text{ along every path:} \]

\[ \text{Alg 1:} \]

\[ \begin{align*}
\text{\hspace{2cm}} & \cdot \text{let } r = 0 \quad \text{counter of \# reachable configs besides } t \\
\text{\hspace{2cm}} & \cdot \text{For every config } c \text{ among the } m \text{ poss. EXCEPT } t \text{ do:} \\
\text{\hspace{4cm}} & \cdot \text{guess whether } J \text{ can put path (of length } \leq w) \text{ from } s \text{ to } c; \text{ if guess } y, \text{ guess } t \text{ verify path.} \\
\text{\hspace{4cm}} & \text{If successful } (y, \text{ succ. verify path}), \text{ set } r = r + 1. \\
\text{\hspace{4cm}} & \text{If } r = l, \text{ accept. o/w reject.}
\end{align*} \]
Works. \( O(f(n)) \) space.

\( L \) (accepts \( s \)) iff guesses contain \( k \) non-\( t \) nodes are reachable; since \( L = \# \) reachable configs, all other nodes (incl. \( t \)) not reachable from \( s \). So \( s \in L \) iff \( t \) not reachable from \( s \), i.e., \( M \) rejects on every path.

Remaining to show: how to nondet. compute \( L = \# \) configs reachable from \( s \).

We give NTM \( N' \) st some branch outputs right value for \( L \), all other branches reject.

"Inductive counting"

For \( i \in [m] \), let \( A_i = \) set of all configs, dist \( i \) from \( s \).

So \( A_0 = \{s\}; A_0 \subseteq A_1 \subseteq A_2 \subseteq \ldots \) \( A_m \) = all configs reachable from \( s \). Want \( |A_m| \). Have \( |A_0| = 1 \).

\( \Rightarrow \) so \( |A_m| = k \).

Here's nondet proc. to compute \( |A_{i+1}| \) given \( |A_i| \)

(some path gets it right, all others reject) (Apply this repeatedly, starting w/ \( |A_0| = 1 \) & reusing space, to get \( |A_m| = k \).)

Alg 2:

Outer loop: Sim. to earlier: alg goes through all \( m \) poss configs \( c \), \( t \) for each one decides whether \( c \in A_{i+1} \),

\( t \) maintains counter of \# of these \( c \)'s that are in \( A_{i+1} \).
To decide whether \( c \in A_{i+1} \):

- **Inner loop:** Loop over all possible configurations \( c' \). For each \( c' \):
  - guess whether \( c' \in A_i \); if yes, guess and check \( c' \)-to-
  \( c' \)-path of length \( \leq i \); if guessed and checked \( c' \in A_i \), check if
  \( M \) transitions from \( c' \) to \( c \) in one step; if yes,
  decide \( c \in A_{i+1} \).

  As loop thru \( c' \)'s, count # of them that
  were verified as being in \( A_i \).

- **At end of inner loop, after all \( c' \)'s were processed:**
  - If the count (\# configs that were
    found to be in \( A_i \)) is \( \neq |A_i| \), it must be \( < |A_i| \) that
    means the guesses missed some \( c' \) in \( A_i \); \( \textbf{Reject} \)
  - If the count (\# configs that were found to be in \( A_i \))
    \( IS = |A_i| \) (remember \( IS \) is given \( |A_i| \)), \( \text{and } c \) was never
    found to be 1-step-reachable from any of the \( c' \)'s in
    \( A_i \), means \( c \) is not in \( A_{i+1} \). So \( a_{i+1} \) decides \( c \notin A_{i+1} \).

So... \( coNL = NL \).

Surprises happen...

**Unit #5 Randomness (in computation)**

Randomized comput: \( a_{i+1} \) can "toss coins" - make
random choices during its execution

- allow \( R.A. \) to err w/ some...
probability.

**Motiv:** - still realistic model
- helpful!

Much like nondet. comput. (multiple choices / execution paths), but more realistic success crit.: "most" paths / "a random path" should succeed.

Randomness key to crypto: unpredictability; adversary kept off balance.
Alg. design is an adv. scenario: given a fixed det alg, worst-case analysis is like adv. choosing input.
If alg is randomized, no fixed alg, so can help thwart adv. inputs.

**Probability basics**

We will always consider discrete/finite sample spaces (no worries about "measurability")

**Basic setup:** Finite sample space $S$; set of all outcomes for "probabilistic experiment" (P.E.)

- P.E. could be "pick a random person in world w. prob. a their height"; $S = \text{all people}$
- P.E. could be "pick a uniform string from $\{0,1\}^n$"; $S = \{0,1\}^n$.

$\mathbb{P}(S) = \frac{1}{57}$
Prob. dist. \( Y \) over \( S \): defined by weight
\[ Y(s) \text{ for each } s \in S, \]
- \( Y(s) \geq 0 \text{ for all } s \in S \)
- \( \sum_{s \in S} Y(s) = 1 \).

If \( Y \) is clear, we may write

"\( Pr\{s\} \)" to mean \( Y(s) \).

\[ \sum_{s \in S} Pr\{s\} = 1. \]

An event: a subset \( A \subseteq S \). "something that does or doesn't happen."

Have \( Pr\{A\} = \sum_{s \in A} Pr\{s\}. \)

For any two events \( A, B \), have

\[ Pr\{A \cap B\} = \sum_{s \in A \cap B} Pr\{s\} = Pr\{A \mid B\} \cdot Pr\{B\}, \]

where \( Pr\{A \mid B\} = \frac{\sum_{s \in A \cap B} Pr\{s\}}{Pr\{B\}} = \frac{Pr\{A \cap B\}}{Pr\{B\}}. \)

"Conditional prob. of \( A \text{ given } B \)"

Ex: P.E. = roll red die \& blue die (both fair)

\( A = \text{total roll } \leq 5 \)

\( B = \text{one die even, other odd.} \)

\( A = \{ (1, 1); (1, 2); \ldots; (4, 1) \} \)
Next time: independence, random variables, tail bounds, rand. alg. for identity testing

Readings for next time:

Papad. 11.1, AB 7.2, 3 (you may like Sipser 10.2)