Computational Complexity

Lecture 7: the Polynomial Hierarchy

Ronald de Haan
me@ronalddehaan.eu

University of Amsterdam

February 22, 2021

Recap

What we saw last time..

- Space-bounded computation
- Limits on memory space
- L, NL, PSPACE
- Logspace reductions
- NL-completeness

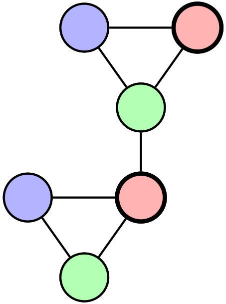
What will we do today?

- The Polynomial Hierarchy
- Bounded quantifier alternation
- Alternating Turing machines

Example problem

- We saw that 3COL is NP-complete, but how about the following problem?

3COL-Extension $= \{ \ (G, V_0) \mid \ G = (V, E)$ *is an undirected graph,* $V_0 \subseteq V$*,*
and each 3-coloring of the vertices in V_0 *can*
be extended to a proper 3-coloring
of the entire graph $G \ \}$



- There seems to be no single (polynomial-size) certificate for yes-inputs
- It is a "∀∃-type" question
- We need a different complexity class to capture the complexity of 3COL-Extension

The complexity class Σ_2^p

Definition (NP)

A language $L \subseteq \{0,1\}^*$ is in the class NP if there is a polynomial $q : \mathbb{N} \to \mathbb{N}$ and a polynomial-time Turing machine $\mathbb{M}$ such that for every $x \in \{0,1\}^*$:

$x \in L$ if and only if **there exists** some $u \in \{0,1\}^{q(|x|)}$ such that $\mathbb{M}(x,u) = 1$.

Definition (coNP)

A language $L \subseteq \{0,1\}^*$ is in the class coNP if there is a polynomial $q : \mathbb{N} \to \mathbb{N}$ and a polynomial-time Turing machine $\mathbb{M}$ such that for every $x \in \{0,1\}^*$:

$x \in L$ if and only if **for all** $u \in \{0,1\}^{q(|x|)}$ it holds that $\mathbb{M}(x,u) = 1$.

The complexity class Σ_2^{p}

Definition (Σ_2^{p})

A language $L \subseteq \{0,1\}^*$ is in the class Σ_2^{p} if there is a polynomial $q : \mathbb{N} \to \mathbb{N}$ and a polynomial-time Turing machine $\mathbb{M}$ such that for every $x \in \{0,1\}^*$:

$$\begin{aligned} x \in L \quad \text{if and only if} \quad & \text{there exists } u_1 \in \{0,1\}^{q(|x|)} \text{ such that} \\ & \text{for all } u_2 \in \{0,1\}^{q(|x|)} \text{ it holds that } \mathbb{M}(x, u_1, u_2) = 1. \end{aligned}$$

The complexity class Π_2^p

Definition (Π_2^p)

A language $L \subseteq \{0,1\}^*$ is in the class Σ_2^p if there is a polynomial $q : \mathbb{N} \to \mathbb{N}$ and a polynomial-time Turing machine $\mathbb{M}$ such that for every $x \in \{0,1\}^*$:

$$\begin{aligned} x \in L \quad \text{if and only if} \quad & \text{for all } u_1 \in \{0,1\}^{q(|x|)} \\ & \text{there exists } u_2 \in \{0,1\}^{q(|x|)} \text{ such that } \mathbb{M}(x, u_1, u_2) = 1. \end{aligned}$$

- It turns out that 3COL-Extension is Π_2^p-complete.

The complexity classes Σ_i^{p}

Definition (Σ_i^{p})

Let $i \geq 1$. A language $L \subseteq \{0,1\}^*$ is in the class Σ_i^{p} if there is a polynomial $q : \mathbb{N} \to \mathbb{N}$ and a polynomial-time Turing machine $\mathbb{M}$ such that for every $x \in \{0,1\}^*$:

$$x \in L \quad \text{if and only if} \quad \begin{array}{ll} \text{there exists } u_1 \in \{0,1\}^{q(|x|)} \text{ such that} & \\ \text{for all } u_2 \in \{0,1\}^{q(|x|)} & \\ \vdots & \\ \text{for all } u_i \in \{0,1\}^{q(|x|)} & \\ \text{it holds that } \mathbb{M}(x, u_1, \ldots, u_i) = 1. & \text{if } i \text{ is even,} \\ \hline \vdots & \\ \text{there exists } u_i \in \{0,1\}^{q(|x|)} & \\ \text{such that } \mathbb{M}(x, u_1, \ldots, u_i) = 1. & \text{if } i \text{ is odd.} \end{array}$$

The complexity classes Π_i^{p}

Definition (Π_i^{p})

Let $i \geq 1$. A language $L \subseteq \{0,1\}^*$ is in the class Π_i^{p} if there is a polynomial $q : \mathbb{N} \to \mathbb{N}$ and a polynomial-time Turing machine $\mathbb{M}$ such that for every $x \in \{0,1\}^*$:

$$x \in L \quad \text{if and only if} \quad \begin{array}{ll} \text{for all } u_1 \in \{0,1\}^{q(|x|)} & \\ \text{there exists } u_2 \in \{0,1\}^{q(|x|)} \text{ such that} & \\ \vdots & \\ \text{for all } u_i \in \{0,1\}^{q(|x|)} & \\ \text{it holds that } \mathbb{M}(x, u_1, \ldots, u_i) = 1. & \text{if } i \text{ is odd,} \\ \hdashline \vdots & \\ \text{there exists } u_i \in \{0,1\}^{q(|x|)} & \\ \text{such that } \mathbb{M}(x, u_1, \ldots, u_i) = 1. & \text{if } i \text{ is even.} \end{array}$$

The Polynomial Hierarchy (PH)

Definition (Σ_0^p, Π_0^p, PH)

$$\Sigma_0^p = \Pi_0^p = \mathsf{P} \qquad\qquad \mathsf{PH} = \bigcup_{i \geq 0} \Sigma_i^p .$$

- Some relations:
  - $\Pi_i^p = \{ \ \overline{L} \mid L \in \Sigma_i^p \ \}$
  - $\Sigma_1^p = \mathsf{NP}$, $\Pi_1^p = \mathsf{coNP}$
  - $\Sigma_i^p \subseteq \Pi_{i+1}^p \subseteq \Sigma_{i+2}^p$,
    $\Pi_i^p \subseteq \Sigma_{i+1}^p \subseteq \Pi_{i+2}^p$
  - $\Sigma_i^p \subseteq \Sigma_{i+1}^p$, $\Pi_i^p \subseteq \Pi_{i+1}^p$
  - $\Sigma_i^p \cup \Pi_i^p \subseteq \mathsf{PSPACE}$
  - $\mathsf{PH} \subseteq \mathsf{PSPACE}$

$$\begin{tikzcd}
& \Pi_1^p \arrow[r] \arrow[ddr] & \Pi_2^p \arrow[r] \arrow[ddr] & \Pi_3^p \arrow[r] \arrow[dr] & {} & {} \arrow[dr] & \\
\mathsf{P} \arrow[ur] \arrow[dr] & & & & {} & \cdots & \mathsf{PSPACE} \\
& \Sigma_1^p \arrow[r] \arrow[uur] & \Sigma_2^p \arrow[r] \arrow[uur] & \Sigma_3^p \arrow[r] \arrow[ur] & {} & {} \arrow[ur] &
\end{tikzcd}$$

“Collapse” of the hierarchy

- Statements like “$\mathsf{P} \neq \mathsf{NP}$” and “$\mathsf{NP} \neq \mathsf{coNP}$” are widely believed conjectures
- We can use these as assumptions to show some results
    - E.g., assuming that $\mathsf{P} \neq \mathsf{NP}$, NP-complete problems are not in P.
- For some results, stronger conjectures seem necessary
- Another conjecture: “the PH does not collapse”
    - “the PH collapses to P” $\mathsf{PH} = \mathsf{P}$
    - “the PH collapses to the ith level” $\mathsf{PH} = \Sigma_i^{\mathsf{p}}$

Theorem

Let $i \geq 1$. *If* $\Sigma_i^p = \Pi_i^p$, *then* $\mathsf{PH} = \Sigma_i^p$.
If $\mathsf{P} = \mathsf{NP}$, *then* $\mathsf{PH} = \mathsf{P}$.

QBF problems complete for Σ_i^p and Π_i^p

- $\Sigma_i\mathsf{SAT} = \{\, \varphi = \exists \overline{u}_1 \forall \overline{u}_2 \dots Q_i \overline{u}_i \ \psi(\overline{u}_1, \dots, \overline{u}_i) : \varphi$ is a true QBF $\}$,
  where each $\overline{u}_j = (u_{j,1}, \dots, u_{j,\ell})$ is a sequence of propositional variables,
  $\exists \overline{u}_j$ stands for $\exists u_{j,1} \exists u_{j,2} \dots \exists u_{j,\ell}$, and $\forall \overline{u}_j$ for $\forall u_{j,1} \forall u_{j,2} \dots \forall u_{j,\ell}$
- $\Pi_i\mathsf{SAT} = \{\, \varphi = \forall \overline{u}_1 \exists \overline{u}_2 \dots Q_i \overline{u}_i \ \psi(\overline{u}_1, \dots, \overline{u}_i) : \varphi$ is a true QBF $\}$,

Theorem

Let $i \geq 1$. Then $\Sigma_i\mathsf{SAT}$ is Σ_i^p-complete and $\Pi_i\mathsf{SAT}$ is Π_i^p-complete (both under polynomial-time reductions).

Oracle characterizations of Σ_i^{p} and Π_i^{p}

Theorem

Let $i \geq 2$. Then $\Sigma_i^p = \mathsf{NP}^{\Sigma_{i-1}\mathsf{SAT}}$ and $\Pi_i^p = \mathsf{coNP}^{\Sigma_{i-1}\mathsf{SAT}}$.

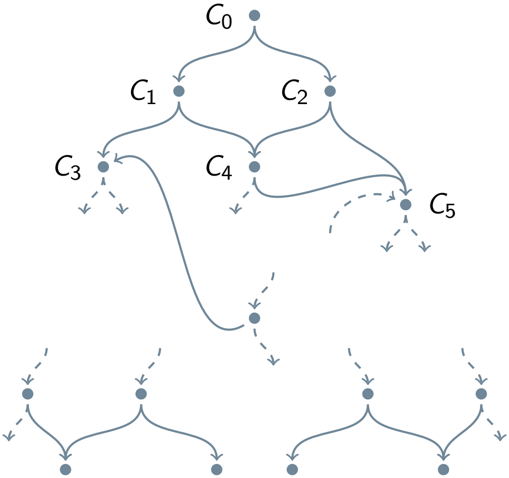
- (Or replace $\Sigma_{i-1}\mathsf{SAT}$ by any $\Sigma_{i-1}^{\mathsf{p}}$-complete or Π_{i-1}^{p}-complete problem.)
- This is often written as: $\Sigma_i^{\mathsf{p}} = \mathsf{NP}^{\Sigma_{i-1}^{\mathsf{p}}}$ and $\Pi_i^{\mathsf{p}} = \mathsf{coNP}^{\Sigma_{i-1}^{\mathsf{p}}}$

Configuration graphs

Configurations C consist of:
(1) tape contents
(2) tape head positions
(3) state $q \in Q$

Configuration graph of a TM $\mathbb{M}$ on some input x:

- Nodes are all the configurations that are reachable from the initial configuration C_0
- Edge from C to C' if applying one of the transition functions in C results in C'



Alternating Turing machines

Definition (Alternating Turing machines; ATMs)

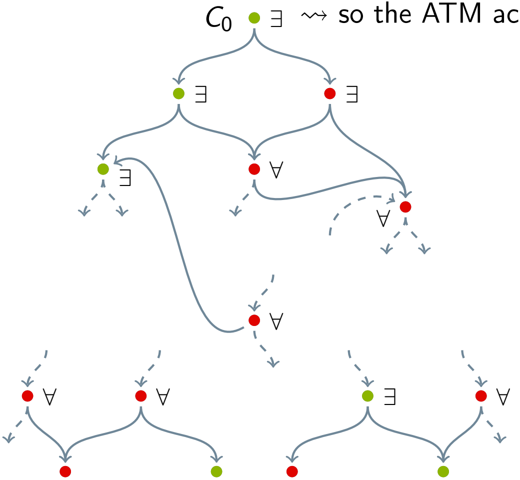
- Instead of a single transition function δ, there are two transition functions δ_1, δ_2.
- The set $Q \setminus \{q_{\mathsf{acc}}, q_{\mathsf{rej}}\}$ is partitioned into $Q_\exists$ and $Q_\forall$.
- Executions of alternating TMs are defined using a labeling procedure on the *configuration graph*. Repeatedly apply, until a fixpoint is reached:
  - Label each configuration with q_{acc} with "accept."
  - If a configuration c with $q \in Q_\exists$ has an edge to a configuration c' that is labeled with "accept," then label c with "accept."
  - If a configuration c has a state $q \in Q_\forall$ and both configurations c', c'' that are reachable from it in the graph are labeled with "accept," then label c with "accept."

  The TM accepts the input if the starting configuration is labeled with "accept."
- The TM runs in time $T(n)$ if for every input x and for every possible sequence of transition function choices, the machine halts after at most $T(|x|)$ steps.

Alternating Turing machines (ct'd)

🔴 = reject

🟢 = accept



$\rightsquigarrow$ so the ATM accepts the input

ATIME, Σ_iTIME, and Π_iTIME

Definition (ATIME)

Let $T : \mathbb{N} \to \mathbb{N}$ be a function. A decision problem $L \subseteq \{0,1\}^*$ is in ATIME($T(n)$) if there exists an ATM that decides L and that runs in time $O(T(n))$.

Definition (Σ_iTIME)

Let $T : \mathbb{N} \to \mathbb{N}$ be a function. A decision problem $L \subseteq \{0,1\}^*$ is in Σ_iTIME($T(n)$) if there exists an ATM that decides L, that runs in time $O(T(n))$, whose initial state is in $Q_\exists$, and that on every input and on every path in the configuration graph alternates at most $i - 1$ times between $Q_\exists$ and $Q_\forall$.

- Π_iTIME is defined similarly to Σ_iTIME, with the difference that the initial state of the ATM is in $Q_\forall$

ATM characterizations

Theorem

$$\mathsf{PSPACE} = \bigcup_{c \geq 0} \mathsf{ATIME}(n^c).$$

Theorem

Let $i \geq 1$. Then:

$$\Sigma_i^p = \bigcup_{c \geq 0} \Sigma_i\mathsf{TIME}(n^c) \qquad \Pi_i^p = \bigcup_{c \geq 0} \Pi_i\mathsf{TIME}(n^c).$$

An overview of complexity classes

$$\begin{tikzcd}
 & & & \mathsf{coNP} \arrow[r, "\subseteq"] \arrow[dd, <->, dashed, "\overset{?}{=}"'] \arrow[ddr, "\subseteq"] & \Pi_2^{\mathsf{p}} \arrow[r, "\subseteq"] \arrow[dd, <->, dashed, "\overset{?}{=}"'] \arrow[ddr, "\subseteq"] & \Pi_3^{\mathsf{p}} \arrow[r, dashed] \arrow[dd, <->, dashed, "\overset{?}{=}"'] \arrow[dr, dashed] & & & & \\
\mathsf{L} \arrow[r, "\subseteq"] & \mathsf{NL} \arrow[r, "\subseteq"] \arrow[rrrrrrr, bend left=40, "\subsetneq"] & \mathsf{P} \arrow[ur, "\subseteq"] \arrow[dr, "\subseteq"'] \arrow[rrrrrrr, bend left=50, "\subsetneq"] & & & & {} \arrow[r, dashed, "\subseteq"'] & \mathsf{PH} \arrow[r, "\subseteq"'] & \mathsf{PSPACE} \arrow[r, "\subseteq"'] & \mathsf{EXP} \\
 & & & \mathsf{NP} \arrow[r, "\subseteq"'] \arrow[uur, "\subseteq"'] & \Sigma_2^{\mathsf{p}} \arrow[r, "\subseteq"'] \arrow[uur, "\subseteq"'] & \Sigma_3^{\mathsf{p}} \arrow[r, dashed] \arrow[ur, dashed] & & & &
\end{tikzcd}$$

Recap

- The classes Σ_i^p and Π_i^p
- The Polynomial Hierarchy
- Σ_i^p-complete and Π_i^p-complete QBF problems
- Characterizations using oracles and ATMs

Next time

- A “breather”
- Time to reflect on what we’ve done so far
- Requests for things to recap?