Computational Complexity

Lecture 7: Non-Uniform Complexity

Ronald de Haan
me@ronalddehaan.eu

University of Amsterdam

April 23, 2024

Recap

What we saw last time..

- 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

What will we do today?

- Non-uniform complexity
- Circuit complexity
- TMs that take advice
- The Karp-Lipton Theorem

Non-uniformity

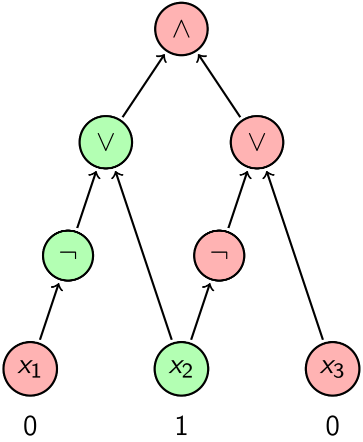
- *"Uniform":* the algorithm is the same, regardless of the input size

  vs.

- *"Non-uniform":* there can be different algorithms for different input sizes

Boolean circuits

- Boolean circuits are very similar to propositional formulas
- Directed acyclic graphs (instead of trees)
- We view binary strings as truth assignments
- Example: $(\neg x_1 \vee x_2) \wedge (\neg x_2 \vee x_3)$, $x = 010$, and $\alpha_x = \{x_1 \mapsto 0, x_2 \mapsto 1, x_3 \mapsto 0\}$



Definition (Circuits)

An *n-input single-output Boolean circuit* C is a directed acyclic graph with:

- n sources (nodes with no incoming edges), labelled 1 to n, and
- one sink (a node with no outgoing edges).

All non-source vertices are called *gates*, and are labelled with $\wedge$, $\vee$, or $\neg$:

- $\wedge$-gates and $\vee$-gates have in-degree 2 (exactly two incoming edges),
- $\neg$-gates have in-degree 1 (exactly one incoming edge).

If C is an n-input single-output Boolean circuit and $x \in \{0,1\}^n$ is a string, then the output $C(x)$ of C on x is defined by plugging in x in the source nodes and applying the operators of the gates, and taking for $C(x)$ the resulting value in $\{0,1\}$ of the sink gate.

SIZE

Definition (Circuit families)

Let $t : \mathbb{N} \to \mathbb{N}$ be a function. A $t(n)$*-size circuit family* is a sequence $\{C_n\}_{n\in\mathbb{N}}$ of Boolean circuits, where each C_n has n inputs and a single output, and $|C_n| \leq t(n)$ for each $n \in \mathbb{N}$.

Definition (SIZE($t(n)$))

Let $t : \mathbb{N} \to \mathbb{N}$ be a function. A language $L \subseteq \{0,1\}^*$ is in SIZE($t(n)$) if there exists a constant $c \in \mathbb{N}$ and a $(c \cdot t(n))$-size circuit family $\{C_n\}_{n\in\mathbb{N}}$ such that for each $x \in \{0,1\}^*$:

$$x \in L \quad \text{if and only if} \quad C_n(x) = 1, \text{ where } n = |x|.$$

The complexity class P/poly

Definition (P/poly)

$$\mathsf{P/poly} = \bigcup_{c \geq 1} \mathsf{SIZE}(n^c).$$

- In other words, P/poly is the class of all decision problems that can be decided by a polynomial-size circuit family.

P $\subseteq$ P/poly

- *(We consider only decision problems $L \subseteq \{0,1\}^*$—i.e., binary alphabets.)*

Theorem

P $\subseteq$ P/poly.

- Main idea:
  - Like in the proof of the Cook-Levin Theorem, we encode polynomial-time computation in logic
  - Instead of using new, fresh variables we use nodes in the Boolean circuit (to encode tape contents, tape head positions, etc)
- In fact, P $\subsetneq$ P/poly

Turing machines that take advice

- We can characterize P/poly (or more generally, non-uniform complexity classes) also using TMs
- The algorithm might differ per input size n, so we will have to give the TM something that depends only on the input size
- This is called advice

Advice characterization of P/poly

Definition (TIME($t(n)$)/$a(n)$)

Let $t, a : \mathbb{N} \to \mathbb{N}$ be functions. The class DTIME$(t(n))/a(n)$ of languages decidable by *$O(t(n))$-time Turing machines with $a(n)$ bits of advice* contains every decision problem $L \subseteq \{0,1\}^*$ such that:

- there exists a sequence $\{\alpha_n\}_{n\in\mathbb{N}}$ with $\alpha_n \in \{0,1\}^{a(n)}$ for each $n \in \mathbb{N}$ and an $O(t(n))$-time deterministic Turing machine $\mathbb{M}$ such that for each $x \in \{0,1\}^*$:

$$x \in L \quad \text{if and only if} \quad \mathbb{M}(x, \alpha_n) = 1, \text{ where } n = |x|.$$

Advice characterization of P/poly (ct'd)

Theorem

$$\mathsf{P/poly} = \bigcup_{c,d \geq 1} \mathsf{DTIME}(n^c)/n^d.$$

- Main idea (for "$\subseteq$"):
  - Use a description of C_n as α_n, and then compute $C_n(x)$ in polynomial time
- Main idea (for "$\supseteq$"):
  - The computation of $\mathbb{M}(x, \alpha_n)$ on inputs $x \in \{0,1\}^n$ can be encoded as a polynomial-size circuit $D_n(\cdot, \alpha_n)$, using ideas from the proof of the Cook-Levin Thm
  - The circuit C_n is D_n with α_n "hardwired in"

P-uniform circuit families

Definition

A circuit family $\{C_n\}_{n\in\mathbb{N}}$ is P-*uniform* if there exists a polynomial-time deterministic TM that on input 1^n outputs a description of C_n, for each $n \in \mathbb{N}$.

Theorem

A decision problem $L \subseteq \{0,1\}^*$ is in P if and only if decidable by a P-uniform circuit family $\{C_n\}_{n\in\mathbb{N}}$.

The Karp-Lipton Theorem

- Question: is SAT decidable by polynomial-size circuits (is it in P/poly)?
  - Perhaps by allowing the algorithm to change per input size, this might work
- The answer: No (assuming that the PH does not collapse)

Theorem (Karp, Lipton 1980)

If NP $\subseteq$ P/poly, *then* $\Sigma_2^p = \Pi_2^p$.

Proof of the Karp-Lipton Thm

The general argument

- Suppose that $\mathsf{NP} \subseteq \mathsf{P}/\mathsf{poly}$.
- We show that then $\Pi_2^p \subseteq \Sigma_2^p$, by showing $\Pi_2\mathsf{SAT} \in \Sigma_2^p$.
- We use the following lemma to swap the order of the quantifiers:

Lemma

If $\mathsf{NP} \subseteq \mathsf{P}/\mathsf{poly}$, *then there exists a polynomial-time algorithm that:*

- *takes polynomial-length advice, and*
- *given a propositional formula* φ*:*
  - *if* φ *is unsatisfiable, it outputs* 0*;*
  - *if* φ *is satisfiable, it outputs a satisfying truth assignment* α *for* φ*.*

- Idea behind the proof of the lemma: use self-reducibility of SAT.

Proof of the Karp-Lipton Thm

Completing the proof

- Take an arbitrary instance of Π_2SAT: $\varphi = \forall \overline{u}.\exists \overline{v}.\psi(\overline{u}, \overline{v})$.
- Let q be the polynomial bounding the size of the advice $\{\alpha_n\}_{n\in\mathbb{N}}$ that can be used to compute satisfying assignments for SAT, in polynomial time with TM $\mathbb{M}$.
- $\varphi = \forall \overline{u}.\exists \overline{v}.\psi(\overline{u}, \overline{v}) \in \Pi_2\mathrm{SAT}$ if and only if for all $\overline{z} \in \{0,1\}^m$, $\psi[\overline{u} \mapsto \overline{z}] \in \mathrm{SAT}$.
- This is the case if and only if:

$\exists$ there exists some $\overline{w} \in \{0,1\}^{q(n)}$ such that

$\forall$ for all $\overline{z} \in \{0,1\}^m$

poly $\mathbb{M}$ uses $\overline{w}$ as advice to output the assignment γ on input $\psi[\overline{u} \mapsto \overline{z}]$ and γ satisfies $\psi[\overline{u} \mapsto \overline{z}]$

Key: we check that γ is correct; because we don't know whether $\overline{w}$ is the right advice

- Thus, $\Pi_2\mathrm{SAT} \in \Sigma_2^{\mathrm{p}}$, and therefore $\Pi_2^{\mathrm{p}} = \Sigma_2^{\mathrm{p}}$.

Recap

- Non-uniform complexity
- Circuit complexity
- TMs that take advice
- The Karp-Lipton Theorem: if $\mathsf{NP} \subseteq \mathsf{P}/\mathsf{poly}$, then $\Sigma_2^p = \Pi_2^p$

Next time

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