Complexity Classes

Non-deterministic polynomial time: $\mathbf{NP}$

- $\mathbf{coNP} = \{L \mid \bar{L} \in \mathbf{NP}\}$
- $L \in \mathbf{NP} \cap \mathbf{coNP}$ is equivalent to $L \in \mathbf{NP}$ and $L \in \mathbf{coNP}$

Polynomial time hierarchy: $\mathbf{PH} = \bigcup_{k \geq 0} \Sigma_k^p$

- $\Delta_0^p = \Sigma_0^p = \Pi_0^p = \mathbf{P}$
- $\Delta_{k+1}^p = \mathbf{P}^{\Sigma_k^p}, \quad k \geq 0$
- $\Sigma_{k+1}^p = \mathbf{NP}^{\Sigma_k^p}, \quad k \geq 0$
- $\Pi_{k+1}^p = \mathbf{coNP}^{\Sigma_k^p}, \quad k \geq 0$

Alternating Turing machine: $\mathbf{AL}, \mathbf{AP}$

Parallel computation:

- $\mathbf{NC} = \mathbf{PT/WK}(\log^k n, n^k)$
- $\mathbf{NC}_j = \mathbf{PT/WK}(\log^j n, n^k)$
- Efficient: $\mathbf{NC}_1, \mathbf{NC}_2$

Randomised computation:

- Monte Carlo: $\mathbf{RP}$
- Las Vegas: $\mathbf{ZPP} = \mathbf{RP} \cap \mathbf{coRP}$
- Majority: $\mathbf{PP}$
- Efficient: $\mathbf{BPP}$

Interactive proofs: $\mathbf{IP}$

Counting complexity: $\#\mathbf{P}$

Parameterised complexity:

- Fixed-parameter tractable: $\mathbf{FTP}$
- Problems in $\mathbf{XP}$ have polynomial-time solutions for every constant value of paramter k

Class inclusions:

- $\mathbf{L} \subseteq \mathbf{NL} \subseteq \mathbf{P} \subseteq \mathbf{NP} \subseteq \mathbf{PH} \subseteq \mathbf{PSPACE} \subseteq \mathbf{EPX}$
- $\mathbf{NC}_1 \subseteq \mathbf{L} \subseteq \mathbf{NL} \subseteq \mathbf{NC}_2 \subseteq \mathbf{NC} \subseteq \mathbf{P}$
- $\mathbf{AL} = \mathbf{P}, \mathbf{AP} = \mathbf{PSPACE}, \mathbf{APSPACE} = \mathbf{EXP}$
- $\mathbf{P} \subseteq \mathbf{ZPP} \subseteq \mathbf{RP} \subseteq \mathbf{NP} \subseteq \mathbf{PP}$
- $\mathbf{RP} \subseteq \mathbf{BPP} \subseteq \mathbf{PP}$
- $\mathbf{NP} \subseteq \#\mathbf{P} \subseteq \mathbf{PSPACE}$
- $\mathbf{PH} = \mathbf{P}^{\#\mathbf{P}} = \mathbf{P}^{\mathbf{PP}}$ (Toda's theorem)
- $\mathbf{IP} = \mathbf{PSPACE}$
- $\mathbf{FTP} \subseteq \mathbf{XP}$

Logspace

For input of length n, the total number of possible configurations for **logspace** decider is $c^{O(\log n)} = n^{O(1)}$ where c is constant. Therefore, $\mathbf{L} \subseteq \mathbf{NL} \subseteq \mathbf{P}$.

NL-completeness

Reachability: Given graph $G = (V, E)$ and vertices $s, t \in V$. Is there a path from s to t in G?

Reachability is NL-complete.

1) **Reachability is in NL**: Start from vertex s and nondeterministically walk to every other reachable vertex. Return *yes/no* whether vertex t is reached.
2) **Reachability is NL-hard**: Consider the computation state graph of any other **NL** algorithm. Such algorith will accept only if there is nondeterministic path from start to accepting state.

2SAT is NL-complete

Polynomial Time

Ciruit value and hornsat are **P**-complete. They are least likely to be in **NC**, that is, efficiently solvable by parallel algorithms.

Oracle Turing Machines

An **oracle Turing machine** $M^?$

- Query string y, query state $q^?$, answer states q_{yes}, q_{no}.
- From the query state $q^?$ the machine moves to q_{yes} or to q_{no}, depending on whether $y \in A$ holds or not, where A is the oracle set.
- A query is performed in one step.
- Computation of $M^?$ with oracle A on input x is $M^A(x)$.

Relativised complexity class: C^A where C is complexity class and A is an oracle.

Example: There exists oracle A for which $\mathbf{P}^A = \mathbf{NP}^A$.

Proof: Let A be a **PSPACE**-complete. Then $\mathbf{PSPACE} \subseteq \mathbf{P}^A \subseteq \mathbf{NP}^A \subseteq \mathbf{NPSPACE} \subseteq \mathbf{PSPACE}$.

Padding

Padding $x; 1^d$ of string x.

Padding argument: If $\mathbf{P} = \mathbf{NP}$ the $\mathbf{EXP} = \mathbf{NEXP}$.

Let $N = c^{n^{O(1)}}$ and c be a constant. Then

$$\begin{aligned} NEXP(n) &= NTIME(c^{n^{O(1)}}) = NTIME(N) \subseteq NP(N) \\ &\subseteq P(N) = TIME(N^{O(1)}) = TIME(c^{n^{O(1)}}) = EXP(n). \end{aligned}$$

NP

A relation $R \subseteq \Sigma^* \times \Sigma^*$ is **polynomially decidable** if there is a deterministic TM deciding the language $\{x; y \mid (x, y) \in R\}$ in polynomial time.

A relation R is **polynomially balanced** if $(x, y) \in R$ implies $|y| \leq |x|^k$ for some $k \geq 1$.

Succint certificate: Language L is in **NP** iff there is a polynomially balanced and polynomially decidable relation R such that

$$L = \{x \mid (x, y) \in R, \exists y \in \Sigma^*\}.$$

Language L is **NP**-complete if:

1) $L \in \mathbf{NP}$: L is in **NP**.
2) $L' \leq_{\mathrm{L}} L$: Known **NP**-complete language L' has logspace reduction to L.

Polynomial-time Hierarchy

A relation $R \subseteq (\Sigma^*)^{k+1}$ is said to be **polynomially balanced** if whenever $(x, y_1, ..., y_k) \in R$, it holds that $|y_1|, ..., |y_k| \leq |x|^t$ for some t.

Language L is in Σ_k^p for $k \geq 1$ iff there is a polynomially balanced relation R such that the language $\{x; y \mid (x, y) \in R\}$ is in Π_{k-1}^p and

$$L = \{x \mid \exists y, (x, y) \in R\}.$$

Language L is in Π_k^p for $k \geq 1$ iff there is a polynomially balanced relation R such that the language $\{x; y \mid (x, y) \in R\}$ is in Σ_{k-1}^p and

$$L = \{x \mid \forall y, (x, y) \in R\}.$$

Language L is in Σ_k^p for $k \geq 1$ iff there is a **polynomially balanced, polynomial-time decidable** $(k+1)$-ary relation R such that

$$L = \{x \mid \exists y_1 \forall y_2 \exists y_3 \cdots Q y_k, (x, y_1, ..., y_k) \in R\}$$

where Q is $\forall$ if k is even and $\exists$ if k is odd.

Language $L \in \mathbf{PH}$ iff $L \in \Sigma_k^p$ for some k.

Example: $\mathbf{PH} \subseteq \mathbf{PSPACE}$

Proof:

- Base case: $\Sigma_0^p = \mathbf{P} \subseteq \mathbf{PSPACE}$
- Recursive case: $k \geq 0$
  - Assume: $\Sigma_k^p \subseteq \mathbf{PSPACE}$
  - Then: $\Sigma_{k+1}^p = \mathbf{NP}^{\Sigma_k^p} \subseteq \mathbf{NPSPACE} \subseteq \mathbf{PSPACE}$. Last step from Savitch's theorem. Oracle doesn't use additional space, **NP** takes at most polynomial space.

Quantified satisfiability: $QSAT_k$: Given a boolean expression ϕ with its variables partitioned into sets $X_1, ..., X_k$. Is the following true?

$$\exists X_1 \forall X_2 \cdots Q X_k \phi$$

where Q is $\forall$ if k is even and $\exists$ if k is odd.

$QSAT_k$ is Σ_k^p-complete for all $k \geq 1$.

Example: $QSAT_2 \in \Sigma_2^p$ but not in $\Sigma_1^p = \mathbf{NP}$.

In order to verify a certificate for $QSAT_2$, we need to check $2^{|X_2|}$ times boolean formulas of length $|X|$, which means it is not polynomial time verifiable and thus $QSAT_2 \notin \Sigma_1^p = \mathbf{NP}$ assuming $\mathbf{P} \neq \mathbf{NP}$.

- $L \in \Sigma_2^p = \mathbf{NP}^{\mathbf{NP}}$: Covering radius
- $L \in \Pi_2^p = \mathbf{coNP}^{\mathbf{NP}}$: Integer expression equivalence

NP-complete problems

NP-complete problems

Boolean problems: circuit sat, sat, 3SAT, MAX2SAT, NAESAT

Graph cover and packing problems: Independent Set, Clique, Vertex Cover, Max Cut, Max Bisection, Bisection Width

Graph path and coloring problems: Hamilton Path, Travelling Salesperson (TSP), k-coloring

Set problems: Exact Cover by 3-Sets, Set Cover, Set Packing

Number problems: Integer programming, Knapsack

Example reductions:

- circuit sat $\rightarrow$ sat
- sat $\rightarrow$ 3-sat
- circuit sat $\rightarrow$ naesat
- 3-sat $\rightarrow$ independent set
- independent set $\rightarrow$ clique
- independent set $\rightarrow$ vertex cover
- naesat $\rightarrow$ max cut
- max cut $\rightarrow$ max bisection
- naesat $\rightarrow$ 3-coloring
