6. The Polynomial Hierarchy

EXACT TSP

Problem EXACT TSP

INSTANCE: \( n \) cities 1, \ldots, \( n \), a nonnegative integer distance \( d_{ij} \) between any two cities \( i \) and \( j \) (such that \( d_{ij} = d_{ji} \)), and an integer \( B \).

QUESTION: Is the length of the shortest tour equal to \( B \)?

Complexity of EXACT TSP

EXACT TSP can be considered as the intersection of two problems – one in NP and one in co-NP:

- in NP: TSP(D) (= asking if the shortest tour has length \( \leq B \)).
- in co-NP: TSP COMPLEMENT (= asking if the shortest tour has length \( \geq B \)).

Definition

A language \( L \) is in the class DP iff there are two languages \( L_1 \in \text{NP} \) and \( L_2 \in \text{co-NP} \) such that \( L = L_1 \cap L_2 \).

Remark. Note that DP is not \( \text{NP} \cap \text{co-NP} \! \). (Most likely DP is not even contained in \( \text{NP} \cup \text{co-NP} \).)

Proposition

- EXACT TSP is DP-complete.
- All exact cost versions of the NP-complete optimization problems studied in the lecture are DP-complete, e.g. INDEPENDENT SET (i.e.: is the size of the biggest independent set equal to some \( K \)?), VERTEX COVER, CLIQUE, etc.
**SAT-UNSAT**

**Problem SAT-UNSAT**

INSTANCE: two Boolean expressions $(\varphi, \varphi')$ (possibly both in 3-CNF).

QUESTION: Is it true that $\varphi$ is satisfiable and $\varphi'$ is unsatisfiable?

**Proposition**

SAT-UNSAT is DP-complete.

**Proof of Membership**

Let $L_1 = \{ (\varphi, \psi) \in \text{PROP} \mid \varphi \text{ satisfiable and } \psi \text{ arbitrary propositional formula} \}$

Let $L_2 = \{ (\varphi, \psi) \in \text{PROP} \mid \varphi \text{ arbitrary propositional formula and } \psi \text{ unsatisfiable} \}$

Clearly $L_1 \in \text{NP}$, $L_2 \in \text{co-NP}$, and SAT-UNSAT = $L_1 \cap L_2$.

**Further DP-complete problems**

“Critical Problems”

**CRITICAL SAT**

INSTANCE: Propositional formula $\varphi$ in CNF

QUESTION: Is it true that $\varphi$ is unsatisfiable but deleting any clause makes $\varphi$ satisfiable?

**CRITICAL HAMILTON PATH**

INSTANCE: (Directed or undirected) graph $G = (V, E)$

QUESTION: Is it true that $G$ has no Hamilton path but addition of any edge creates a Hamilton path?

**CRITICAL 3-COLORABILITY**

INSTANCE: Undirected graph $G = (V, E)$

QUESTION: Is it true that $G$ has no 3-coloring but deletion of any node makes it 3-colorable?

**Proof of Hardness**

Let $L$ be an arbitrary language in DP, i.e., there exists a language $L_1 \in \text{NP}$ and a language $L_2 \in \text{co-NP}$ with $L = L_1 \cap L_2$.

Let $x$ be an arbitrary instance of $L$. We reduce $x$ to the following instance $R(x)$ of SAT-UNSAT:

$L_1 \in \text{NP}$ and $L_2 \in \text{co-NP} \Rightarrow$

there exists a reduction $R_1$ from $L_1$ to 3-SAT and

there exists a reduction $R_2$ from $L_2$ to co-3-SAT.

We define $R(x) := (R_1(x), R_2(x))$.

Clearly, $R(x)$ is a positive instance of SAT-UNSAT $\iff$

$R_1(x)$ is satisfiable and $R_2(x)$ is unsatisfiable $\iff$

(by the correctness of $R_1$ and $R_2$)

$x \in L_1$ and $x \in L_2 \iff$

$x \in L$.

**Remark**

The above problems are called “critical” because the input $x$ is “critical” with respect to some property, i.e., $x$ has some property but the slightest modification of $x$ does not.
Oracle Machines

Motivation

- Intuitively, an oracle is a subroutine with 0 cost (we count the cost of the oracle as 1 for the call – but we neglect the cost of the computation carried out by the oracle). ⇒ We can study complexity in a setting where a part of the computation comes “for free”.
- Oracles allow us to isolate orthogonal (independent) sources of complexity, i.e. we can answer questions like: Suppose that we know the complexity of some sub-task $A$ for solving problem $B$. What is the remaining complexity of problem $B$?

Oracle Machines

Definition

An oracle Turing machine $M'$ has the following additional features:
- an additional tape (= query tape)
- three additional states: query state $q?$, answer states $q_{YES}$, $q_{NO}$

Suppose that $M'$ has an oracle for the problem $A$. Then the call of the oracle works as follows: If $M'$ is in state $q?$, then $M'$ decides if the string $z$ on the query tape is a positive instance of $A$ or not.
⇒ $M'$ either enters state $q_{YES}$ or $q_{NO}$ in one step.

Notation. For any time complexity class $C$ and oracle $A$ (where $A$ is either a problem or a class of problems) we write $C^A$ for the problems which can be decided by a TM within the time bound of $C$, where the TM is allowed to use an oracle for (any problem in the class) $A$.

Examples. $P^{SAT}$, $NP^{SAT}$, $P^{NP}$, $NP^{NP}$, ...

The Polynomial Hierarchy

Definition

The polynomial hierarchy is a sequence of classes:

- $\Delta_0^P = \Sigma_0^P = \Pi_0^P = P$
- $i \geq 0 : \Delta_{i+1}^P = P^{\Sigma_i^P}$
$\Sigma_{i+1}^P = NP^{\Sigma_i^P}$
$\Pi_{i+1}^P = co-NP^{\Sigma_i^P}$
- Cumulative polynomial hierarchy: $PH = \bigcup_{i \geq 0} \Sigma_i^P$

In the literature also the following notation is used: $\Delta_i^P$, $\Sigma_i^P$, $\Pi_i^P$

Properties of the Polynomial Hierarchy

- special case $i = 1$:
  $\Delta_1^P = P^{P_{\Sigma_0^P}} = P^P = P$
  $\Sigma_1^P = NP^{\Sigma_0^P} = NP^P = NP$
  $\Pi_1^P = co-NP^{\Sigma_0^P} = co-NP$

- special case $i = 2$:
  $\Delta_2^P = P^{P_{\Sigma_1^P}} = P^{NP}$
  $\Sigma_2^P = NP^{\Sigma_1^P} = NP^{NP}$
  $\Pi_2^P = co-NP^{\Sigma_1^P} = co-NP^{NP}$

$\Delta_i^P \subseteq \Sigma_i^P \subseteq \Delta_{i+1}^P \subseteq \Sigma_{i+1}^P \subseteq \Delta_{i+2}^P$
Characterization via Certificates

**Theorem**

- Let $L$ be a language and $i \geq 1$. Then $L \in \Sigma_i P$ iff there is a polynomially balanced relation $R$ (i.e., there exists $k$, s.t. $(x, y) \in R$ implies $|y| \leq |x|^k$), such that the language \( \{ x \# y \mid (x, y) \in R \} \) is in $\Pi_{i-1} P$ and
  \[
  L = \{ x \mid \text{there exists a } y \text{ with } |y| \leq |x|^k \text{ s.t. } (x, y) \in R \}
  \]

- Let $L$ be a language and $i \geq 1$. Then $L \in \Pi_i P$ iff there is a polynomially balanced relation $R$ such that the language \( \{ x \# y \mid (x, y) \in R \} \) is in $\Sigma_{i-1} P$ and
  \[
  L = \{ x \mid \text{for all } y \text{ with } |y| \leq |x|^k, (x, y) \in R \}
  \]

**Remark.** Of course, in the definition of $\Sigma_i P$, we could omit the condition $|y| \leq |x|^k$, since we talk about a polynomially balanced relation $R$.

---

**Proof (continued)**

$\Rightarrow$ Suppose that $L \in \Sigma_i P$, i.e., $L$ is decided by a nondeterministic, polynomial-time TM $M$ with an oracle for some language $K \in \Sigma_{i-1} P$. We must show that an appropriate relation $R$ exists.

By the induction hypothesis, there exists a binary relation $S$, s.t. the language \( \{ u \# v \mid (u, v) \in S \} \) is in $\Pi_{i-2} P$ and $K = \{ u \mid \text{there exists a } v \text{ with } |v| \leq |u|^k \text{ s.t. } (u, v) \in S \}$.

We construct a relation $R$ as follows: We know that $x \in L$ iff there exists an accepting computation of $M$ (with oracle for $K$) on $x$. We define $R$, s.t. $(x, y) \in R$, iff $y$ is a “certificate” of $x$ in the following sense:

1. $y$ encodes the non-deterministic choices of a successful computation of the TM $M$ as in the construction of succinct certificates for NP.
2. In addition, $y$ contains a certificate $v_j$ for every successful call $u_j$ to the oracle $K \in \Sigma_{i-1} P$.

Proof

It suffices to prove the correctness of the characterization of $\Sigma_i P$ for every $i$. The correctness of the characterization of $\Pi_i P$ follows immediately by the equality $\Pi_i P = \text{co-} \Sigma_i P$.

The correctness proof for $\Sigma_i P$ proceeds by induction on $i$.

Recall that $\Sigma_1 P = \text{NP}$. Hence, for $i = 1$, the theorem corresponds to the characterization of NP via succinct certificates. For $i > 1$, we show both directions separately:

$\Leftarrow$ Suppose that such a relation $R$ exists. We must show that $L \in \Sigma_i P$. Indeed, $L$ is decided by the following nondeterministic, polynomial-time Turing machine with $\Sigma_{i-1} P$-oracle:

1. On input $x$, guess an appropriate $y$.
2. Check by means of a $\Pi_{i-1} P$ oracle if $(x, y) \in R$ (or, equivalently, check by a $\Sigma_{i-1} P$ oracle if $(x, y) \not\in R$).

$\Rightarrow$ We must check that the first part of $y$ indeed encodes a successful computation of the TM $M$. This can be done in polynomial time (as in the construction of succinct certificates for NP).

2. We must check for polynomially many pairs $(u_j, v_j)$ that $(u_j, v_j) \in S$ holds. Each such test is in $\Pi_{i-2} P$. Hence, in total, these tests are in $\Delta_{i-1} P \subseteq \Pi_{i-1} P$ (actually, they are even in $\Pi_{i-2} P$).

3. For all “no”-queries $u_j$ to the $K$-oracle, we must check that indeed $u_j \not\in K$. Since $K \in \Sigma_{i-1} P$, each test $u_j \not\in K$ is in $\Pi_{i-1} P$.

All these tests together can be done by a single $\Pi_{i-1} P$ computation since the co-query “there exists a $j$ with $u_j \in K$” is in $\Sigma_{i-1} P$:

Guess $j$ and $v_j$ and check in $\Pi_{i-2} P$ that $(u_j, v_j) \in S$. The correctness proof for $\Pi_i P$ follows immediately by the equality $\Pi_i P = \text{co-} \Sigma_i P$.
Characterization via Certificates and Alternation

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

**Corollary, part 1**
Let $L$ be a language and $i \geq 1$. Then $L \in \Sigma_i P$ iff there is a polynomially balanced, polynomial-time decidable $(i+1)$-ary relation $R$ such that

$$L = \{ x \mid \exists y_1 \forall y_2 \exists y_3 \cdots Qy_i \text{ such that } (x, y_1, \ldots, y_i) \in R \}$$

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

**Proof idea**
Use the above theorem and proceed by induction on $i$. Repeatedly replace languages in $\Pi_j P$ and $\Sigma_j P$ by their certificate forms as in the theorem.

Properties of PH

**Definition**
We say that the polynomial hierarchy collapses to the $i$-th level if $\Delta_i P = \Sigma_i P = \Pi_i P = \Sigma_i P$ holds for every $j > i$.

**Remark**. It is unknown whether PH is indeed an infinite hierarchy, i.e.: $\Sigma_0 P \subset \Sigma_1 P \subset \Sigma_2 P \subset \ldots$ is generally believed but not known.

**Proposition**
- If for some $i \geq 1$, $\Sigma_i P = \Pi_i P$, then the polynomial hierarchy collapses to the $i$-th level. In particular, if $NP = co-NP$, then the polynomial hierarchy collapses to the first level.
- $P = NP$ iff $P = PH$.
- Notice that it can be the case that $P \neq NP$ and $NP \neq co-NP$ but $PH$ collapses to the second level (not expected to happen, though).

QBFs: Quantified Boolean Formulae

**QSAT$_i$**
"quantified satisfiability with $i$ alternating blocks of quantifiers":

**INSTANCE**: Boolean expression $\varphi$ with the Boolean variables partitioned into $i$ sets $X_1, \ldots, X_i$.

**QUESTION**: Is it true that there exists a partial truth assignment for the variables $X_1$ such that for all partial truth assignments for $X_2$ there exists a partial truth assignment for $X_3$ . . . $\varphi$ is satisfied by the overall truth assignment?

**Notation**
A QSAT$_i$-formula is given in the form $\exists X_1 \forall X_2 \exists X_3 \cdots QX_i \varphi$, where $Q$ is $\forall$ if $i$ is even and $\exists$ if $i$ is odd.
### ΣₐP-Completeness

**Theorem**

For all i ≥ 1, QSATᵢ is ΣₐP-complete.

**Proof of ΣₐP-membership**

Recall the characterization of ΣₐP via certificates: A language L is in ΣₐP iff there is a polynomially balanced, polynomial-time decidable (i + 1)-ary relation R, s.t. \( L = \{ x \mid \exists y_1 \forall y_2 \exists y_3 \ldots Q_y \psi \text{ such that } (x, y_1, \ldots, y_i) \in R \} \).

For a QSATᵢ-formula \( \psi = \exists X_1 \forall X_2 \exists X_3 \cdot \cdot \cdot QX_i \varphi \), we take as certificates \( (y_1, \ldots, y_i) \) combinations of variable assignments to the alternating blocks \( X_1, X_2, \ldots, X_i \) of variables (where each \( y_j \) is an assignment on the variables in \( X_j \)) s.t. the formula \( \varphi \) is true in the overall assignment.

**Proof of ΣₐP-hardness (continued)**

Now let \( x \) be an arbitrary instance of the decision problem corresponding to the language \( L \). Moreover, let \( \hat{X} \) denote the values of the variables in \( X \) corresponding to the string \( x \). By \( \varphi(\hat{X}) \) we denote the result of substituting in \( \varphi \) the corresponding Boolean values \( \hat{X} \) for \( X \). We define the desired instance of QSATᵢ as \( \psi = \exists Y_1 \forall Y_2 \ldots \exists Y_i \exists Z \varphi(\hat{X}) \).

For the correctness proof, we observe: Let \( y_1, \ldots, y_i \) be arbitrary strings with Boolean “encoding” \( \hat{Y}_1, \ldots, \hat{Y}_i \) and suppose that we substitute these values \( \hat{Y}_1, \ldots, \hat{Y}_i \) for the variables \( Y_1, \ldots, Y_i \) in \( \varphi \). Then the resulting expression \( \varphi(\hat{X}, \hat{Y}_1, \ldots, \hat{Y}_i) \) is satisfiable (i.e., there exist an appropriate assignment to the variables in \( Z \)) iff \( (x, y_1, \ldots, y_i) \in R \).

It remains to show that \( x \in L \iff \psi \) is a positive instance of QSATᵢ. Indeed, \( x \in L \) iff there is a \( y_1 \) s.t. for all \( y_2, \ldots, y_i \) there is a \( y_i \) s.t. \( (x, y_1, \ldots, y_i) \in R \). In terms of \( \psi \), this means that for these values of \( \hat{X} \) there are values \( \hat{Y}_1 \) for \( Y_1 \) s.t. for all values \( \hat{Y}_2 \) for \( Y_2, \ldots \) there are values \( \hat{Y}_i \) for \( Y_i \) and there are values \( \hat{Z} \) for \( Z \) s.t. the resulting formula \( \varphi(\hat{X}, \hat{Y}_1, \ldots, \hat{Y}_i, \hat{Z}) \) is true, i.e., \( \psi \) is a positive instance of QSATᵢ.

### Further Complete Problems

**Theorem**

We only consider the case that \( i \) is odd. The case that \( i \) is even is treated analogously. Let \( L \) be an arbitrary language in \( \SigmaₐP \). Hence, there exists a polynomially balanced, polynomial-time decidable \( (i + 1) \)-ary relation \( R \), s.t. \( L = \{ x \mid \exists y_1 \forall y_2 \exists y_3 \ldots Q_y \psi \text{ such that } (x, y_1, \ldots, y_i) \in R \} \).

There exists a polynomial-time deterministic TM \( M \) accepting \( x\#y_1\#\ldots\#y_i \iff (x, y_1, \ldots, y_i) \in R \). Following the proof of the Cook-Levin Theorem, there exists a Boolean formula \( \varphi \) that captures the computations of \( M \). We split the variables in \( \varphi \) into \( i + 2 \) classes:

- Variable set \( X \): all propositional variables in \( \varphi \) encoding the first part (before the first \( \# \)) of the input string to \( M \).
- Variable sets \( Y_1 \) to \( Y_i \): encode the remaining input string.
- Variable set \( Z \) captures all other aspects of the computation of \( M \).

**Theorem**

For all \( i \geq 1 \) even, the QSATᵢ problem remains \( \SigmaₐP \)-complete even if the instances \( \exists X_1 \forall X_2 \exists X_3 \ldots \forall X_i \varphi \) are restricted s.t. \( \varphi \) is in 3-DNF.

For all \( i \geq 1 \) odd, the QSATᵢ problem remains \( \SigmaₐP \)-complete even if the instances \( \exists X_1 \forall X_2 \exists X_3 \ldots \exists X_i \varphi \) are restricted s.t. \( \varphi \) is in 3-CNF.

**Theorem**

**MINIMAL MODEL SAT**: Given a propositional formula \( \varphi \) in CNF and an atom \( x \), is \( x \) true in some (subset) minimal model of \( \varphi \)?

**MINIMAL MODEL SAT** is \( \Sigma₂P \)-complete.
MINIMAL MODEL SAT

Proof of the $\Sigma_2^P$-memberhship

We have to show that MINIMAL MODEL SAT can be decided by an NP-algorithm using an NP-oracle (or, equivalently, a co-NP-oracle).

1. Guess a truth assignment $I$, s.t. $x$ is true in $I$. Let $Y \subseteq X$ denote the variables which are true in $I$.
2. Check that $I$ is true in $I$.
3. Check (with an oracle) that there does not exist a “smaller” satisfying truth assignment $J$ of $\psi$, i.e., let $Z$ denote the variables true in $J$, then $Z \not\subseteq Y$ for any satisfying truth assignment $J$ of $\psi$.

The check in step 3 can be done by a co-NP-oracle, i.e.: checking that there does exist a ‘smaller’ satisfying truth assignment $J$ of $\psi$ can be clearly done in NP.

MINIMAL MODEL SAT

Proof of the $\Sigma_2^P$-hardness (continued)

"⇒" Suppose that $\psi = (\exists x_1, \ldots, x_k)(\forall y_1, \ldots, y_l) \varphi$ is true. Then there exists a partial assignment $I$ on $\{x_1, \ldots, x_k\}$, s.t. for any values assigned to $\{y_1, \ldots, y_l\}$, the formula $\varphi$ is true (or, equivalently, $\neg \varphi$ is false).

We define the truth assignment $J$ appropriate to $\chi$ as follows:

$$J(x_i) = I(x_i) \quad \text{and} \quad J(y_j) = I(x_{j'}) \quad \text{for every} \ i, j,$$

$$J(y_j) = \text{true} \quad \text{for every} \ j, \text{and} \ J(z) = \text{true}.$$  

We claim that $J$ is a minimal model of $\chi$ where $z$ is true.

Clearly, $J$ is a model (i.e., satisfiying truth assignment) of $\chi$ since all conjuncts ($\neg x_i \Leftrightarrow x_{j'}$) and the disjunct $(y_1 \land \cdots \land y_l \land z)$ are true in $I$. Moreover, $J(z) = \text{true}$ by definition. It remains to show that there does not exist a strictly “smaller” model of $\chi$.

Suppose to the contrary that there exists a model $J'$ of $\chi$, s.t. $J'$ is strictly smaller than $J$. Then there exists a variable $x_i, x_{j'}, y_j$ or $z,$ s.t. this variable is true in $J$ and false in $J'$. We distinguish 3 cases:
Further Properties of PH

Theorem
If there is a PH-complete problem, then the polynomial hierarchy collapses to some finite level.

Proof
Assume $L$ is PH-complete. Then $L \in \Sigma_i P$ for some $i$. But then any $L' \in \Sigma_{i+1} P$ reduces to $L$. This means that $\Sigma_i P = \Sigma_{i+1} P$ since each level is closed under reductions. Thus PH collapses to the $i$-th level.

By the above theorem, PH probably has no complete problems. But of course each level of PH does (namely QSAT$_i$).

Restrictions on the Oracle Calls

Motivation
We consider two kinds of restrictions:

1. Number of oracle calls.
   - In DP, only 2 calls to an oracle are allowed.
   - Many natural problems require only $O(\log n)$ oracle calls, since they come down to finding the optimal value via binary search, e.g.: max. size of a clique, max./min. cardinality of a model, etc.

2. Adaptive vs. non-adaptive calls:
   - adaptive: The $i$-th question to the oracle may depend on the result of the previous $(i-1)$ calls to the oracle.
   - non-adaptive: otherwise.

Examples in $P^{NP[\log n]}$

CARD-MINIMAL MODEL SAT
INSTANCE: Boolean formula $\varphi$ and an atom $z$.
QUESTION: Is $z$ true in a cardinality-minimal model of $\varphi$?

Proof of $P^{NP[\log n]}$-membership

1. Compute the size $K$ of a cardinality-minimal model of $\varphi$. This can be done by a binary search asking questions like “Does $\varphi$ have a model of size $\leq k$?”. For this task, we need $\log n$ calls to an NP-oracle, where $n$ = number of variables in $\varphi$.

2. Finally, ask an NP-oracle: “Is $z$ true in some model $I$ of $\varphi$, s.t. $I$ sets exactly $K$ variables to true?”

Analogously: CARD-MAXIMAL MODEL SAT
Examples of Optimization Problems in $\text{FP}^{\text{NP}[\log n]}$

Some graph problems

- **MIN-VERTEX COVER, MAX-CLIQUE, MAX-INDEP.-SET**: Given a graph $G = (V, E)$, what is the size of the smallest vertex cover (resp. the biggest clique or the biggest independent set)?
- **CHROMATIC NUMBER**: Given a graph $G = (V, E)$, what is the smallest number $k$, s.t. $G$ has a $k$-coloring?

Some SAT-related problems

- **CARD-MINIMAL-MODEL, CARD-MAXIMAL-MODEL**: Given a Boolean formula $\varphi$, what is the size of a minimal (resp. maximal) model of $\varphi$?
- **MAX-SAT**: Given a Boolean formula $\varphi$ in CNF, what is the maximal number of clauses that can be satisfied by a truth assignment?

Examples in $\text{P}^{\text{NP}}$ (Continued)

**LEX-MINIMAL MODEL SAT**

INSTANCE: Boolean formula $\varphi$, order $(x_1, \ldots, x_n)$ of the variables in $\varphi$.
QUESTION: Is $x_n$ true in the lexicographically smallest model of $\varphi$?

Proof of $\text{P}^{\text{NP}}$-membership

**LEX-MINIMAL MODEL SAT** can be decided by the following program with $n$ calls to an NP-oracle.

\[
\text{for } i := 1 \text{ to } n \text{ do } \\
\text{ check if } \varphi \text{ has a model } \mathcal{I}, \text{ s.t. (for all } j < i: \mathcal{I}(x_j) = v_j \text{) and } \mathcal{I}(x_i) = 0; \\
\text{ if yes then set } v_i := 0, \text{ otherwise set } v_i := 1; \\
\text{ if } v_n = 1 \text{ then return true else return false.}
\]

Analogously: **LEX-MAXIMAL MODEL SAT**

Examples of Optimization Problems in $\text{FP}^{\text{NP}}$

Some graph problems

- **MIN-WEIGHT-VERTEX COVER, MAX-WEIGHT-CLIQUE, MAX-WEIGHT-INDEP.-SET**: Given a graph $G = (V, E)$ and weights $w_i$ of the vertices, what is the size of the minimal total weight of a vertex cover, etc.?
- **TSP**: What is the length of the shortest tour through the $n$ cities?

Some SAT-related problems

- **WEIGHT-MINIMAL-MODEL, WEIGHT-MAXIMAL-MODEL**
- **MAX-WEIGHT-SAT**: Given a Boolean formula $\varphi$ in CNF and vector $(w_1, \ldots, w_m)$ of weights of the clauses $(c_1, \ldots, c_m)$ in $\varphi$, what is the maximal total weight of clauses that can be simultaneously satisfied by a truth assignment?
Proof (continued)

"⊆": Suppose that a language $L$ is decided by a TM $M$ with polynomially many non-adaptive SAT queries. Then $L$ can be decided with logarithmically many adaptive NP queries as follows:

- In $O(\log n)$ queries determine the precise number $K$ of “yes” answers to the non-adaptive queries. This can be done by binary search using the oracle: “Given a set of Boolean expressions, does it have satisfying truth assignments for at least $k$ of them?"
- Ask the NP query: “Do there exist $K$ satisfiable Boolean expressions such that if all other expressions were unsatisfiable (at this point, we know that they must be), then $M$ would end up accepting?”

Remark. A succinct certificate for the last query consists of indices $i_1, \ldots, i_K$ of Boolean expressions and models $I_1, \ldots, I_K$ of them.

Learning Objectives

- Oracle machines
- Complexity classes: DP, $\Delta_1^P$, $\Sigma_1^P$, $\Pi_1^P$, PH
- The intuition of these classes and complete problems
- Restrictions on the oracle calls: $\Delta_2^P[\log n] = P^{\text{NP}[\log n]} = P^{\text{NP}}$
- Problem reductions in $\Sigma_2^P$
- Properties of PH (sufficient conditions for PH to collapse)
- Characterization of $\Sigma_1^P$ and $\Pi_1^P$ via certificates
- The power of alternation: limited alternation in QSAT$_i$