The Polynomial Hierarchy

- Optimization problems
- The class **DP**
- The classes $\mathbf{P}^{\mathbf{NP}}$ and $\mathbf{FP}^{\mathbf{NP}}$
- The classes $\mathbf{FP}^{\mathbf{NP}[\log n]}$ and $\mathbf{FP}_{\parallel}^{\mathbf{NP}}$
- The polynomial hierarchy

(C. Papadimitriou: *Computational Complexity*, Chapter 17)

Optimization Problems

- Optimization problems have not been classified satisfactorily within the theory of **P** and **NP**.
- Consider the traveling salesperson problem
  TSP(D) is **NP**-complete but what about the optimization problem TSP of finding the shortest tour and its variants?

  EXACT TSP: Given a distance matrix and an integer B, is the length of the shortest tour equal to B?

  TSP COST: Given a distance matrix, compute the length of the shortest tour.

  TSP: Given a distance matrix, find the shortest tour.

Optimization problems—cont'd

- The four variants can be ordered in "increasing complexity" by reductions: TSP(D) ; EXACT TSP ; TSP COST ; TSP
- All the four variants of TSP are *polynomially equivalent*: there is a polynomial-time algorithm for one iff there is one for all four (because TSP(D) and TSP are).
- Reductions and completeness provide a more refined and interesting characterization of problems.
- For example, the other three versions of TSP are complete for some very natural extensions of **NP**.

The Class DP

- EXACT TSP in **NP**? Most probably not but closely related to **NP** and **coNP**.
- A language L is in the class **DP** iff there are two languages $L_1 \in \mathbf{NP}$ and $L_2 \in \mathbf{coNP}$ such that $L = L_1 \cap L_2$.
- EXACT TSP in **DP** because
  the length of the shortest tour equal to B iff
  - there is a tour of length at most B (a TSP(D) problem) *and*
  - there is a no tour of length at most $B-1$
    (a TSP(D) COMPLEMENT problem).
- Note **DP** *is not* $\mathbf{NP} \cap \mathbf{coNP}$!
  (Most likely **DP** is not contained even in $\mathbf{NP} \cup \mathbf{coNP}$.)

The Class DP–cont'd

- SAT-UNSAT: given two Boolean expressions ϕ, ϕ' both in CNF with three literals per clause. Is it true that ϕ is satisfiable and ϕ' is not?
- SAT-UNSAT is **DP**-complete.
- EXACT TSP is **DP**-complete.
- "Exact cost" versions of **NP**-complete optimization problems (INDEPENDENT SET, KNAPSACK, MAX-CUT, MAX SAT, ...) can be shown **DP**-complete.

The Class DP–cont'd

- Also other types of problems are in **DP**.
- CRITICAL SAT: Given a Boolean expression ϕ, is it true that ϕ is unsatisfiable but deleting any clause makes it satisfiable?
- UNIQUE SAT: Given a Boolean expression ϕ, is it true that ϕ has a unique satisfying truth assignment?
- CRITICAL HAMILTON PATH: Given a graph, is it true that it has no Hamilton path but addition of any edge creates a Hamilton path?
- CRITICAL 3-COLORABILITY: Given a graph, is it true that it has no 3-coloring but deletion of any node makes it 3-colorable?

  ☞ "Critical" versions are known to be **DP**-complete.

The Classes $\mathbf{P}^{\mathbf{NP}}$ and $\mathbf{FP}^{\mathbf{NP}}$

- **DP**: the class of languages decided by two queries to an **NP** (SAT) oracle.
- A generalization of this idea: allow a polynomial number of *adaptive* SAT oracle calls: class $\mathbf{P}^{SAT}$.
- Since SAT is **NP**-complete, $\mathbf{P}^{SAT} = \mathbf{P}^{\mathbf{NP}}$ $(\Delta_2\mathbf{P})$.
- $\mathbf{FP}^{\mathbf{NP}}$: the corresponding functional problem functions computable using a polynomial number of *adaptive* **NP** oracle queries.

The Class $\mathbf{FP}^{\mathbf{NP}}$

- There are several natural $\mathbf{FP}^{\mathbf{NP}}$-complete problems
- MAX-WEIGHT SAT: Given a set of clauses each with an integer weight, find a truth assignment that satisfies a set of clauses with the most total weight.
- MAX OUTPUT: Given a nondeterministic Turing machine N and its input 1^n such that N halts on input 1^n in $O(n)$ steps with a binary string of length n on its output string, determine the largest output (considered as a binary integers) of any computation of N on 1^n.
- MAX OUTPUT is $\mathbf{FP}^{\mathbf{NP}}$-complete
- MAX-WEIGHT SAT is $\mathbf{FP}^{\mathbf{NP}}$-complete

TSP

INSTANCE: n cities $1,\ldots,n$ and a nonnegative integer distance d_{ij} between any two cities i and j (such that $d_{ij} = d_{ji}$).
QUESTION: What is the shortest tour of the cities?

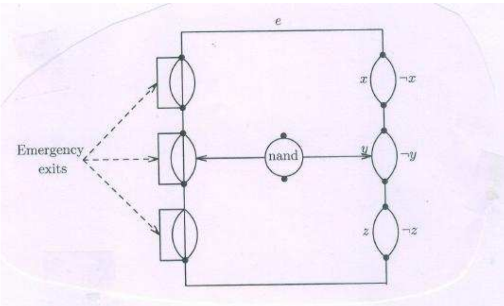
Theorem. TSP is $\mathbf{FP^{NP}}$-complete.

Proof.

- TSP $\in \mathbf{FP^{NP}}$: use binary search and the $\mathbf{NP}$ oracle for TSP(D): is there a tour of length at most B?
- Completeness: reduction (R,S) from MAX-WEIGHT SAT to TSP: Given a set Σ of clauses with weights, $R(\Sigma)$ is a set of cities with distances such that if t is the shortest tour for $R(\Sigma)$, then $S(t)$ is the truth assignment satisfying clauses with the most total weight.

TSP—cont'd

- From clauses with weights build a graph defining the distances: For nodes u,v, (i) if no edge between u and v, distance is W (sum of all weights); (ii) if there is an edge, then distance is 0 except for "emergency edges" which have the weights of the corresponding clauses.



[Papadimitriou, 1994]

TSP—cont'd

- A tour of the graph induces a corresponding truth assignment T.
- A tour uses an "emergency edge" if the corresponding clause is not true in the truth assignment induced by the tour.
- The length of a tour is the sum of weights of clauses not satisfied by T, i.e., $W-$ "tour length" is the total weight of T.
- Hence, minimum length tour corresponds to the maximum weight truth assignment.

TSP—cont'd

Corollary. TSP COST is $\mathbf{FP^{NP}}$-complete.

- Other $\mathbf{FP^{NP}}$-complete problems:
  KNAPSACK, WEIGHTED MAX CUT, WEIGHTED BISECTION WIDTH
- What about CLIQUE SIZE, UNARY TSP, MAX SAT, MAX CUT, BISECTION WIDTH?
- For these only $\log n$ oracle calls are needed: cost polynomially large (logarithmically many bits)
- CLIQUE SIZE:
  Given a graph, determine the size of its largest clique.
  Use binary search with oracle: is the largest clique larger than k?
  Only $\log n$ queries are needed where n is the number of nodes.

The Classes $\mathbf{FP}^{\mathbf{NP}[\log n]}$ and $\mathbf{FP}_{\|}^{\mathbf{NP}}$

- $\mathbf{P}^{\mathbf{NP}[\log n]}$: the class of languages decided by a polynomial time oracle machine which on input x asks a total of $O(\log |x|)$ SAT queries.
- $\mathbf{FP}^{\mathbf{NP}[\log n]}$: the corresponding class of functions
- MAX OUTPUT$[\log n]$ is $\mathbf{FP}^{\mathbf{NP}[\log n]}$-complete.
- MAX SAT SIZE is $\mathbf{FP}^{\mathbf{NP}[\log n]}$-complete.
- CLIQUE SIZE is $\mathbf{FP}^{\mathbf{NP}[\log n]}$-complete.
- UNARY TSP, MAX SAT, MAX CUT, BISECTION WIDTH are $\mathbf{FP}^{\mathbf{NP}[\log n]}$-complete.

The Classes $\mathbf{FP}^{\mathbf{NP}[\log n]}$ and $\mathbf{FP}_{\|}^{\mathbf{NP}}$ — cont'd

- What if only non-adaptive queries can be asked?
- $\mathbf{P}_{\|}^{\mathbf{NP}}$: the class of languages L decided by an oracle machine M which on input x computes in polynomial time a polynomial number of instances of SAT (or any other problem in $\mathbf{NP}$) and receives correct answers. Based on the answers M decided whether $x \in L$ in polynomial time.

The Classes $\mathbf{FP}^{\mathbf{NP}[\log n]}$ and $\mathbf{FP}_{\|}^{\mathbf{NP}}$ — cont'd

Theorem. $\mathbf{P}_{\|}^{\mathbf{NP}} = \mathbf{P}^{\mathbf{NP}[\log n]}$

Proof. ($\supseteq$) If a machine makes $k \log n$ adaptive queries, there are at most $2^{k \log n} = O(n^k)$ queries in the whole computation.

($\subseteq$) If a language is decidable by polynomially many non-adaptive SAT queries, it can be decided in logarithmically many adaptive $\mathbf{NP}$ queries:

- 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 l of them?
- Ask the $\mathbf{NP}$ query: Do there exist k satisfying truth assignments for k of the expressions such that if all other expressions were unsatisfiable, the oracle machine would end up accepting.

The Polynomial Hierarchy

The polynomial hierarchy is a sequence of classes:

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

In the literature also the following notation is used: Δ_i^{p}, Σ_i^{p}, Π_i^{p}

The polynomial hierarchy—cont'd

Properties:

- $\Delta_1\mathbf{P} = \mathbf{P}^{\Sigma_0\mathbf{P}} = \mathbf{P}^{\mathbf{P}} = \mathbf{P}$
  $\Sigma_1\mathbf{P} = \mathbf{NP}^{\Sigma_0\mathbf{P}} = \mathbf{NP}^{\mathbf{P}} = \mathbf{NP}$
  $\Pi_1\mathbf{P} = \mathbf{coNP}^{\Sigma_0\mathbf{P}} = \mathbf{coNP}$
- $\Delta_2\mathbf{P} = \mathbf{P}^{\Sigma_1\mathbf{P}} = \mathbf{P}^{\mathbf{NP}}$
  $\Sigma_2\mathbf{P} = \mathbf{NP}^{\Sigma_1\mathbf{P}} = \mathbf{NP}^{\mathbf{NP}}$
  $\Pi_2\mathbf{P} = \mathbf{coNP}^{\Sigma_1\mathbf{P}} = \mathbf{coNP}^{\mathbf{NP}}$
- $\Delta_i\mathbf{P} \subseteq \begin{array}{c}\Sigma_i\mathbf{P}\\ \Pi_i\mathbf{P}\end{array} \subseteq \Delta_{i+1}\mathbf{P} \subseteq \begin{array}{c}\Sigma_{i+1}\mathbf{P}\\ \Pi_{i+1}\mathbf{P}\end{array} \subseteq \Delta_{i+2}\mathbf{P}$

Certificates

- Let L be a language and $i \geq 1$. Then $L \in \Sigma_i\mathbf{P}$ iff there is a polynomially balanced relation R such that the language $\{x;y \mid (x,y) \in R\}$ is in $\Pi_{i-1}\mathbf{P}$ and

  $$L = \{x \mid \textit{there is a } y \text{ such that } (x,y) \in R\}$$

- Let L be a language and $i \geq 1$. Then $L \in \Pi_i\mathbf{P}$ iff there is a polynomially balanced relation R such that the language $\{x;y \mid (x,y) \in R\}$ is in $\Sigma_{i-1}\mathbf{P}$ and

  $$L = \{x \mid \textit{for all } y \text{ with } |y| \leq |x|^k, (x,y) \in R\}$$

Certificates—cont'd

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

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

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

PH is fragile

- If for some $i \geq 1$, $\Sigma_i\mathbf{P} = \Pi_i\mathbf{P}$, then for all $j > i$, $\Delta_j\mathbf{P} = \Sigma_j\mathbf{P} = \Pi_j\mathbf{P} = \Sigma_i\mathbf{P}$.
  (The polynomial hierarchy is said to *collapse* to the ith level.)
- If $\mathbf{P} = \mathbf{NP}$, or if $\mathbf{NP} = \mathbf{coNP}$, then the polynomial hierarchy collapses to the first level.
- $\mathbf{P} = \mathbf{NP}$ iff $\mathbf{P} = \mathbf{PH}$.
- Notice that it can be the case that $\mathbf{P} \neq \mathbf{NP}$ and $\mathbf{NP} \neq \mathbf{coNP}$ but the polynomial hierarchy collapses to the second level (not expected to happen, though).

Complete problems

- QSAT$_i$ (quantified satisfiability with i alternations of quantifiers):
  Given a Boolean expression ϕ with the Boolean variables partitioned into i sets $X_1, \ldots, X_i$, is it true that
  there is a partial truth assignment for the variables X_1 such that
  for all partial truth assignments for X_2
  there is a partial truth assignment for X_3
  $\ldots \phi$ is satisfied by the overall truth assignment?
- QSAT$_i$: Is the following *quantified Boolean expression* true

$$\exists X_1 \forall X_2 \exists X_3 \cdots QX_i \ \phi$$

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

Complete problems

Theorem. For all $i \geq 1$, *QSAT$_i$ is $\Sigma_i\mathbf{P}$-complete*.

Example. MINIMUM CIRCUIT: Given a Boolean circuit C, is it true that there is no circuit with fewer gates that computes the same Boolean function?
MINIMUM CIRCUIT $\in \Pi_2\mathbf{P}$.

Example. MINIMAL MODEL SAT: Given a set of clauses S and an atom a, is it true that there is a (subset) minimal model of S with a true?
MINIMAL MODEL SAT is $\Sigma_2\mathbf{P}$-complete.

Complete problems—cont'd

Example. RULE INFERENCE: Given a set of rules of the form
$a_1 \vee \cdots \vee a_n \leftarrow b_1 \wedge \cdots \wedge b_m$
and an atom a, is it true that there is a (subset) minimal set of atoms closed under the rules containing a?
RULE INFERENCE is $\Sigma_2\mathbf{P}$-complete.

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

Proof. Assume L is **PH**-complete. Then $L \in \Sigma_i\mathbf{P}$ for some i. But then any $L' \in \Sigma_{i+1}\mathbf{P}$ reduces to L. This means that $\Sigma_i\mathbf{P} = \Sigma_{i+1}\mathbf{P}$ (all levels are closed under reductions). $\Box$

Complete problems—cont'd

- $\mathbf{PH} \subseteq \mathbf{PSPACE}$
  $L \in \mathbf{PH}$ iff $L \in \Sigma_i\mathbf{P}$ iff $L = \{x \mid \exists y_1 \forall y_2 \cdots Qy_i \text{ s.t. } (x, y_1, \ldots, y_i) \in R\}$
- It is open whether $\mathbf{PH} = \mathbf{PSPACE}$.
  If $\mathbf{PH} = \mathbf{PSPACE}$, then the polynomial hierarchy collapses to some finite level. (There are **PSPACE**-complete problems.)
- If PH does not collapse, *problems are strictly harder in an upper level* when compared to the lower level: if L is a $\Sigma_{i+1}\mathbf{P}$-complete language and $L \in \Sigma_i\mathbf{P}$, then PH collapses to the level i.

Example. Consider a $\Sigma_2\mathbf{P}$-complete problem. It cannot be solved with a polynomial overhead on top of a procedure for a problem in **NP** (unless PH collapses to the level 1).

BPP and polynomial circuits

Theorem. $\mathbf{BPP} \subseteq \Sigma_2\mathbf{P}$

Corollary. $\mathbf{BPP} \subseteq \Sigma_2\mathbf{P} \cap \Pi_2\mathbf{P}$

Proof. $\mathbf{BPP}$ is closed under complement. Hence, if $L \in \mathbf{BPP}$, $\overline{L} \in \mathbf{BPP} \subseteq \Sigma_2\mathbf{P}$ implying $L \in \Pi_2\mathbf{P}$. $\square$

Theorem. If SAT has polynomial circuits, then the polynomial hierarchy collapses to the second level.

Learning Objectives

- The classes $\mathbf{DP}$, $\mathbf{P}^{\mathbf{NP}}$, $\mathbf{FP}^{\mathbf{NP}}$, $\mathbf{FP}^{\mathbf{NP}[\log n]}$ and $\mathbf{FP}_{\|}^{\mathbf{NP}}$
- Classification of optimization problems into these classes
- The concept of the polynomial hierarchy