Polynomial reductions

• A polynomial reduction of \( L_1 \subseteq \Sigma_1 \) to \( L_2 \subseteq \Sigma_2 \) is a polynomially computable function \( f : \Sigma_1 \rightarrow \Sigma_2 \) with \( w \in L_1 \iff f(w) \in L_2 \).

• Proposition. If \( L_1 \) is polynomially reducible to \( L_2 \), then
  1. \( L_1 \in P \) if \( L_2 \in P \) and \( L_1 \in NP \) if \( L_2 \in NP \)
  2. \( L_2 \not\in P \) if \( L_1 \not\in P \) and \( L_2 \not\in NP \) if \( L_1 \not\in NP \).

• \( L_1 \) and \( L_2 \) are polynomially equivalent if they are polynomially reducible to each other.

NP-complete problems

• A language \( L \subseteq \Sigma^* \) is NP-complete if
  1. \( L \in NP \)
  2. Any \( L' \in NP \) is polynomially reducible to \( L \).

• Proposition. If \( L \) is NP-complete and \( L \in P \), then \( P = NP \).

• Corollary. If \( L \) is NP-complete and \( P \not= NP \), then there exists no polynomial algorithm for \( L \).

Structure of the class NP

\[
\begin{align*}
P & \quad \bullet \quad \bullet \quad \bullet \\
NP & \quad \bullet \quad \bullet \\
NP\text{-complete} & \quad \bullet \quad \bullet
\end{align*}
\]

Fundamental open problem: \( P \not= NP \)?

Proving NP-completeness

• Theorem (Cook 1971). SAT is NP-complete.

• Proposition. \( L \) is NP-complete if
  1. \( L \in NP \)
  2. there exists an NP-complete problem \( L' \) that is polynomially reducible to \( L \).

• Example: INDEPENDENT SET
  Instance: Graph \( G = (V, E) \) and \( k \in \mathbb{N}, k \leq |V| \).
  Question: Is there a subset \( V' \subseteq V \) such that \( |V'| \geq k \) and no two vertices in \( V' \) are joined by an edge in \( E \)?
Reducing 3SAT to INDEPENDENT SET

• Let $F$ be a conjunction of $n$ clauses of length 3, i.e., a disjunction of 3 propositional variables or their negation.

• Construct a graph $G$ with $3n$ vertices that correspond to the variables in $F$.

• For any clause in $F$, connect by three edges the corresponding vertices in $G$.

• Connect all pairs of vertices corresponding to a variable $x$ and its negation $\neg x$.

• $F$ is satisfiable if and only if $G$ contains an independent set of size $n$.

NP-hard problems

• Decision problem: solution is either yes or no

• Example: Traveling salesman decision problem:
  Given a network of cities, distances, and a number $B$, does there exist a tour with length $\leq B$?

• Search problem: find an object with required properties

• Example: Traveling salesman optimization problem:
  Given a network of cities and distances, find a shortest tour.

• Decision problem $NP$-complete $\Rightarrow$ search problem $NP$-hard

• $NP$-hard problems: at least as hard as $NP$-complete problems

NP-hard problems in bioinformatics

Sperschneider 08

• Multiple alignment
• Shortest common superstring
• Protein threading
• Pseudoknot prediction
• Bi-Clustering
  ...

Further complexity classes

cor NP:
Problems whose complement is in $NP$

PSPACE:
Problems solvable in polynomial space

EXPTIME:
Problems solvable in exponential time

...
Literature

- J. E. Hopcroft and J. D. Ullman: Introduction to automata theory, languages and computation. Addison-Wesley, 1979
- C. H. Papadimitriou: Computational complexity. Addison-Wesley, 1994