6. NP-Completeness

Reinhard Pichler

Institut für Informationssysteme
Arbeitsbereich DBAI
Technische Universität Wien

21 June, 2010

Outline

6. NP-Completeness
6.1 Characterizing NP
6.2 Variants of Satisfiability
6.3 Some NP-complete Graph Problems
6.4 Further NP-complete Problems

Characterizing NP

Definition

1 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.

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

Proposition

Let $L \subseteq \Sigma^*$ be a language. Then $L \in \text{NP}$ if and only if there is a polynomially balanced and polynomially decidable relation $R$ such that $L = \{ x \in \Sigma^* \mid (x, y) \in R \text{ for some } y \in \Sigma^* \}$.

Proof

"$\Rightarrow$" Suppose that there is such a relation $R$. Then $L$ is decided by an NTM which, on input $x$, guesses a $y$ of length at most $|x|^k$ and uses the machine for $R$ to decide in polynomial time whether $(x, y) \in R$.

"$\Leftarrow$" Suppose that $L \in \text{NP}$, i.e. there is an NTM $M$ deciding $L$ in time $|x|^k$ for some $k$. We define the relation $R$ as follows: $(x, y) \in R$ iff $y$ is the encoding of an accepting computation of $M$ on input $x$. Then $R$ has the following properties:

- $R$ is polynomially balanced, since each computation of $M$ is polynomially bounded.
- $R$ is polynomially decidable, since it can be checked in linear time whether $y$ encodes an accepting computation of $M$ on $x$.
- $L = \{ x \mid (x, y) \in R \text{ for some } y \}$, since $M$ decides $L$. 
Succinct Certificates

A problem is in NP if any positive instance $x$ of the problem has at least one succinct certificate (or polynomial witness) $y$. NP contains a huge number of practically important, natural computational problems:

- A typical problem is to construct a mathematical object satisfying certain specifications (path, solution of equations, routing, VLSI layout, ...). This is the certificate.
- The decision version of the problem is to determine whether at least one such object exists for the input.
- The object is usually not very large compared to the input.
- The specifications of the object are usually simple enough to be checkable in polynomial time.

NP-Completeness and Algorithm Design Techniques

Showing that a problem is NP-complete implies that the problem is not in P unless NP = P (which is considered very unlikely).

When a problem is known to be NP-complete, further efforts are usually directed to:

- Attacking special cases
- Approximation algorithms
- Randomized algorithms
- (Exponential) algorithms that are practical for small instances
- etc.

Variants of Satisfiability

SAT
INSTANCE: Boolean formula $\varphi$.
QUESTION: Is $\varphi$ satisfiable?

3-SAT
INSTANCE: Boolean formula $\varphi$ in 3-CNF
QUESTION: Is $\varphi$ satisfiable?

2-SAT
INSTANCE: Boolean formula $\varphi$ in 2-CNF
QUESTION: Is $\varphi$ satisfiable?
Complexity of SAT and 3-SAT

Cook-Levin Theorem

SAT is NP-complete.

Theorem

3-SAT is NP-complete.

Proof of the NP-membership

SAT and also 3-SAT and can be decided by the following NP-algorithm:
1. Guess a truth assignment \( T \) for the variables in \( \varphi \).
2. Check that \( \varphi \) is true in \( T \).

Proof idea of the NP-hardness of SAT

We have to reduce SAT to 3-SAT, i.e.: Let \( \varphi \) be an arbitrary Boolean formula. We have to show that there exists a Boolean formula \( R(\varphi) = \psi \), s.t. \( \psi \) is in 3-CNF and \( \varphi \) is satisfiable \( \Leftrightarrow \psi \) is satisfiable.

Remarks.

- Hence, an arbitrary Boolean formula \( \varphi \) can be transformed in polynomial time into a sat-equivalent formula \( \psi \) in 3-CNF.
- In general, \( \varphi \) and \( \psi \) are not logically equivalent.
- This result is by no means trivial: The “usual” transformation into CNF via de Morgan’s laws and the distributivity of \( \land \) and \( \lor \) usually leads to an exponential blow-up. For instance, consider the CNF, which is logically equivalent to \((x_1 \land y_1) \lor \ldots \lor (x_n \land y_n)\).
Some NP-complete Graph Problems

We have already encountered the **INDEPENDENT SET** problem. The following two problems are closely related:

**CLIQUE**

**INSTANCE:** Undirected graph \( G = (V, E) \) and integer \( K \).

**QUESTION:** Does there exist a clique \( C \) of size \( \geq K \) i.e., \( C \subseteq V \), s.t. for all \( i, j \in I \) with \( i \neq j \), \([i, j] \in E\).

**VERTEX COVER**

**INSTANCE:** Undirected graph \( G = (V, E) \) and integer \( K \).

**QUESTION:** Does there exist a vertex cover \( N \) of size \( \leq K \) i.e., \( N \subseteq V \), s.t. for all \([i, j] \in E\), either \( i \in N \) or \( j \in N \).
INDEPENDENT SET vs. CLIQUE

Example

Proposition

Let $G = (V, E)$ be an undirected graph with $I \subseteq V$. Moreover, let $\overline{G} = (V, \overline{E})$ be the complement graph, i.e. $[i, j] \in E \iff [i, j] \not\in \overline{E}$.

$I$ is an independent set in $G$ $\iff$ $I$ is a clique in $\overline{G}$.

INDEPENDENT SET vs. VERTEX COVER

Example
**INDEPENDENT SET vs. VERTEX COVER**

**Example**

![Graph Example](image)

**Proposition**

Let \( G = (V, E) \) be an undirected graph with \( I \subseteq V \).  
\( I \) is an independent set in \( G \) iff \( N = V \setminus I \) is a vertex cover in \( G \).

**Idea.** An independent set never contains both endpoints of an edge. Hence, of every edge in \( E \), at least one endpoint is in \( V \setminus I \).

**Complexity**

**Theorem**

**INDEPENDENT SET, CLIQUE, and VERTEX COVER** are NP-complete.

**Proof**

**Membership.** An NP-algorithm for these problems first guesses a subset \( S \) of the vertices \( V \) and then checks in polynomial time that \( S \) has the desired property (e.g., \( S \) is an independent set of size \( \geq K \)).

**Hardness.** By the above equivalences, it suffices to prove the NP-hardness of one of these 3 problems. In fact, we have already seen a reduction from **3-SAT** to **INDEPENDENT SET**, from which its NP-hardness follows immediately.

**Further Graph Problems**

### 3-COLORABILITY

**INSTANCE:** Undirected graph \( G = (V, E) \)  
**QUESTION:** Does \( G \) have a 3-coloring? i.e., an assignment of one of 3 colors to each of the vertices in \( V \) such that any two vertices \( i, j \) connected by an edge \([i, j] \in E\) do not have the same color?

### \( k \)-COLORABILITY (for fixed value \( k \geq 1 \))

**INSTANCE:** Undirected graph \( G = (V, E) \)  
**QUESTION:** Does \( G \) have a \( k \)-coloring? i.e., an assignment of one of \( k \) colors to each of the vertices in \( V \) such that any two vertices \( i, j \) connected by an edge \([i, j] \in E\) do not have the same color?

**HAMILTON-PATH**

**INSTANCE:** (directed or undirected) graph \( G = (V, E) \)  
**QUESTION:** Does \( G \) have a Hamilton path? i.e., a path visiting all vertices of \( G \) exactly once.

**HAMILTON-CYCLE**

**INSTANCE:** (directed or undirected) graph \( G = (V, E) \)  
**QUESTION:** Does \( G \) have a Hamilton cycle? i.e., a cycle visiting all vertices of \( G \) exactly once.
Further Variants of Satisfiability

**Not-all-equal SAT (NAESAT)**

INSTANCE: Boolean formula \( \varphi \) in 3-CNF

QUESTION: Does there exist a truth assignment \( T \) on \( \varphi \), such that the 3 literals in each clause do not have the same truth value?

**1-IN-3-SAT**

INSTANCE: Boolean formula \( \varphi \) in 3-CNF

QUESTION: Does there exist a truth assignment \( T \) on \( \varphi \), such that in each clause, exactly one literal is true in \( T \)?

---

**Remarks**

- Clearly 1-IN-3-SAT \( \subset \) NAESAT \( \subset \) 3-SAT. The instances of these 3 problems are the same, namely 3-CNF formulae. However, the positive instances of 1-IN-3-SAT are a proper subset of NAESAT, which in turn are a proper subset of the positive instances of 3-SAT.

- Note that the NP-completeness of any of these 3 problems does not immediately imply the NP-completeness of any of the other problems, since it is a priori not clear if further constraining the positive instances makes things easier or harder.
NP-Completeness

Theorem
All of the following problems are NP-complete.
- k-COLORABILITY for any $k \geq 3$ (e.g., 3-COLORABILITY)
- HAMILTON-PATH, HAMILTON-CYCLE, TSP(D)
- k-SAT for any $k \geq 3$, NAESAT, 1-IN-3-SAT

Proof
Membership. see Übungsblatt, Exercise 6.
Hardness. in the Komplexitätstheorie lecture in the summer term

Learning Objectives
- The concept of NP-completeness and its characterizations in terms of succinct certificates.
- You should now be familiar with the intuition of NP-completeness (and recognize NP-complete problems).
- Two fundamental NP-complete problems: SAT and 3-SAT.
- Difference between logical equivalence and sat-equivalence.
- Many more examples of NP-complete problems, e.g.: CLIQUE, INDEPENDENT SET, VERTEX COVER, 3-COLORABILITY, HAMILTON-PATH, HAMILTON-CYCLE, TSP(D), etc.