Some Variants of Satisfiability

We have already encountered several versions of satisfiability problems:

- intractable: SAT, 3-SAT
- tractable: 2-SAT, HORNSAT

We shall encounter further intractable versions of satisfiability problems:

- restricted (but still intractable) versions of SAT
- CIRCUIT SAT
- Not-all-equal SAT (NAESAT)
- (MONOTONE) 1-IN-3-SAT
- strongly related problem: HITTING SET

Narrowing NP-complete languages

An NP-complete language can sometimes be narrowed down by transformations which eliminate certain features of the language but still preserve NP-completeness.

Restricting SAT to formulae in CNF and a further restriction to 3-SAT are typical examples. Generally, k-SAT (i.e., formulae are restricted to CNF with exactly k literals in each clause) is NP-complete for any $k \geq 3$.

Here is another example of narrowing an NP-complete language:

**Proposition**

3-SAT remains NP-complete even if the Boolean expressions $\varphi$ in 3-CNF are restricted such that

- each variable appears at most three times in $\varphi$
- each literal appears at most twice in $\varphi$. 
**Boolean Circuits**

**Semantics**

Let $C$ be a Boolean circuit and let $X(C)$ denote the set of variables appearing in the circuit $C$. A truth assignment for $C$ is a function $T : X(C) \rightarrow \{\text{true}, \text{false}\}$.

The truth value $T(i)$ for each gate $i$ is defined inductively:

- If $s(i) = \text{true}$, $T(i) = \text{true}$ and if $s(i) = \text{false}$, $T(i) = \text{false}$.
- If $s(i) = x_j \in X(C)$, then $T(i) = T(x_j)$.
- If $s(i) = \neg$, then $T(i) = \text{true}$ if $T(j) = \text{false}$, else $T(i) = \text{false}$ where $(j, i)$ is the unique edge entering $i$.
- If $s(i) = \land$, then $T(i) = \text{true}$ if $T(j) = T(j') = \text{true}$ else $T(i) = \text{false}$ where $(j, i)$ and $(j', i)$ are the two edges entering $i$.
- If $s(i) = \lor$, then $T(i) = \text{true}$ if $T(j) = \text{true}$ or $T(j') = \text{true}$ else $T(i) = \text{false}$ where $(j, i)$ and $(j', i)$ are the two edges entering $i$.
- $T(C) = T(n)$, i.e. the value of the circuit $C$.

**Proof**

The reduction consists in rewriting an arbitrary instance $\varphi$ of 3-SAT in such a way that the forbidden features are eliminated.

Consider a variable $x$ appearing $k > 3$ times in $\varphi$.

1. Replace the first occurrence of $x$ in $\varphi$ by $x_1$, the second by $x_2$, and so on where $x_1, \ldots, x_k$ are new variables.
2. Add clauses $(\neg x_1 \lor x_2), (\neg x_2 \lor x_3), \ldots, (\neg x_k \lor x_1)$ to $\varphi$.

Let $\varphi'$ be the result of systematically modifying $\varphi$ in this way. Clearly, $\varphi'$ has the desired syntactic properties.

Now $\varphi$ is satisfiable iff $\varphi'$ is satisfiable:

For each $x$ appearing $k > 3$ times in $\varphi$, the truth values of $x_1, \ldots, x_k$ are the same in each truth assignment satisfying $\varphi'$.

**CIRCUIT SAT**

**INSTANCE**: Boolean circuit $C$ with variables $X(C)$

**QUESTION**: Does there exist a truth assignment $T : X(C) \rightarrow \{\text{true}, \text{false}\}$ such that $T(C) = \text{true}$?

**Theorem**

CIRCUIT SAT is NP-complete.

**Proof of NP-Membership**

Consider the following NP-algorithm:

1. Guess a truth assignment $T : X(C) \rightarrow \{\text{true}, \text{false}\}$.
2. Check that $T(C) = \text{true}$ holds.
Reduction from SAT to 3-SAT

Motivation

- We have already seen how an arbitrary propositional formula $\varphi$ can be transformed efficiently into a sat-equivalent formula $\psi$ in 3-CNF.
- This transformation (first into CNF and then into 3-CNF) is intuitive and clearly works in polynomial time. However, the log-space complexity of this transformation is not immediate.
- We now give an alternative transformation by reducing CIRCUIT SAT to 3-SAT. In total, we thus have:

$$\text{SAT} \leq L \text{CIRCUIT SAT} \leq L 3\text{-SAT}$$

NAESAT

Not-all-equal SAT (NAESAT)

INSTANCE: Boolean formula $\varphi$ in 3-CNF

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

Remark. Clearly NAESAT $\subset$ 3-SAT.

Theorem

NAESAT is NP-complete.
NAESAT

Proof of NP-Hardness
Recall the Boolean formula \( \varphi(C) \) resulting from the reduction of CIRCUIT SAT to 3-SAT. For all one- and two-literal clauses in the resulting CNF-formula \( \varphi(C) \), we add the same literal \( z \) (possibly twice) to make them 3-literal clauses.

The resulting formula \( \varphi_z(C) \) fulfills the following equivalence:

\[ \varphi_z(C) \in \text{NAESAT} \iff C \in \text{CIRCUIT SAT}. \]

\( \exists \tau \)

If a truth assignment \( T \) satisfies \( \varphi_z(C) \) in the sense of NAESAT, so does the complementary truth assignment \( \neg T \).

Thus, \( z \) is false in either \( T \) or \( \neg T \) which implies that \( \varphi(C) \) is satisfied by either \( T \) or \( \neg T \). Thus \( C \) is satisfiable.

1-IN-3-SAT

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

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

 MONOTONE 1-IN-3-SAT

 INSTANCE: Boolean formula \( \varphi \) in 3-CNF, s.t. the clauses in \( \varphi \) contain only unnegated atoms.

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

 Remark

 Both 1-IN-3-SAT and MONOTONE 1-IN-3-SAT are NP-complete.
Proof of the NP-hardness of **1-IN-3-SAT**

We prove the NP-hardness by a reduction from **4-SAT:**

Let \( \varphi \) be an arbitrary instance of **4-SAT**, i.e., \( \varphi \) is in 4-CNF.

We construct an instance \( \psi \) of **1-IN-3-SAT** as follows:

For every clause \( l_1 \lor l_2 \lor l_3 \) in \( \varphi \), let \( a_1, a_2, a_4, b_1, b_2, c_1, c_2, d \) be 9 fresh propositional variables. Then \( \psi \) contains the following 7 clauses:

1. \( l_1 \lor a_1 \lor b_1 \)
2. \( l_2 \lor a_2 \lor b_1 \)
3. \( l_3 \lor a_4 \lor c_1 \)
4. \( l_4 \lor a_3 \lor b_2 \)
5. \( l_5 \lor a_3 \lor c_2 \)
6. \( l_6 \lor a_4 \lor c_2 \)
7. \( b_1 \lor b_2 \lor d \)

**Idea.** Suppose that in a truth assignment \( T \) of \( \varphi \) all literals in the clause \( l_1 \lor \cdots \lor l_4 \) are \textbf{false}:

- By (1) – (3): If \( l_1 \) and \( b_2 \) are \textbf{false}, then \( b_1 \) must be \textbf{true}.
- By (4) – (6): If \( l_3 \) and \( b_2 \) are \textbf{false}, then \( b_2 \) must be \textbf{true}.

However, by (7), it is not allowed that both \( b_1 \) and \( b_2 \) are \textbf{true}.

**HITTING SET**

**INSTANCE:** Set \( T = \{ t_1, \ldots, t_p \} \), family \( \{ V_i \}_{1 \leq i \leq n} \) of subsets of \( T \), i.e., for all \( i \in \{ 1, \ldots, n \} \), \( V_i \subseteq T \).

**QUESTION:** Does there exist a set \( W \subseteq T \) s.t. \( |W \cap V_i| = 1 \) for all \( i \in \{ 1, \ldots, n \} \)? (A set \( W \) with this property is called a “hitting set”.

**Corollary**

**HITTING SET** is NP-complete.

**Proof of the NP-hardness**

By reduction from **MONOTONE 1-IN-3-SAT:** Let an instance of **MONOTONE 1-IN-3-SAT** be given by the 3-CNF formula \( \varphi \) over the variables \( X \). We define the following instance of **HITTING SET**:

\( T = X \). Moreover, suppose that \( \varphi \) contains \( n \) clauses. Then there are \( n \) sets \( \{ V_i \}_{1 \leq i \leq n} \). If the \( i \)-th clause in \( \varphi \) is \( l_1 \lor l_2 \lor l_3 \), then \( V_i = \{ l_1, l_2, l_3 \} \).

We show how an arbitrary instance \( \varphi \) of **1-IN-3-SAT** can be transformed into an equivalent instance \( \psi \) of **MONOTONE 1-IN-3-SAT**:

Let \( X = \{ x_1, \ldots, x_n \} \) be the variables in \( \varphi \). Then the variables in \( \psi \) are \( X \cup \{ x_i' \mid 1 \leq i \leq n \} \cup \{ a, b, c \} \). In \( \varphi \), we replace every negative literal of the form \( \neg x_i \) (for some \( i \)) by the unnegated atom \( x_i' \).

Moreover, for every \( i \in \{ 1, \ldots, n \} \), we add the following 3 clauses:

1. \( x_i \lor x_i' \lor a \)
2. \( x_i \lor x_i' \lor b \)
3. \( a \lor b \lor c \)

**Idea.** These three clauses guarantee that in a legal 1-in-3 assignment of \( \psi \), the variables \( x_i \) and \( x_i' \) have complementary truth values. Hence, \( x_i' \) indeed encodes \( \neg x_i \).

Some Graph Problems

We have already proved the NP-completeness of the following graph problems:

- **INDEPENDENT SET**
- **CLIQUE**
- **VERTEX COVER**

We shall now show the following results:

- **3-COLORABILITY** is NP-complete.
- **HAMILTON-PATH \( \leq_L \) HAMILTON-CYCLE \( \leq_L \) TSP(D)**
**INDEPENDENT SET**

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

** QUESTION:** Does there exist an independent set $I$ of size $\geq K$?

i.e., $I \subseteq V$, s.t. for all $i, j \in I$ with $i \neq j$, $[i, j] \notin E$.

---

** 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 $i, j \in C$ 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 $j \in N$.

---

** Complexity**

** Theorem**

The $k$-COLORABILITY-problem is NP-complete for any fixed $k \geq 3$.

The 2-COLORABILITY-problem is in P.

** Proof**

** NP-Membership of k-COLORABILITY:**

1. Guess an assignment $f : V \rightarrow \{1, \ldots, k\}$
2. Check for every edge $[i, j] \in E$ that $f(i) \neq f(j)$.

** P-Membership of 2-COLORABILITY:** (w.l.o.g., $G$ is connected)

1. Start by assigning an arbitrary color to an arbitrary vertex $v \in V$.
2. Suppose that the vertices in $S \subseteq V$ have already been assigned a color. Choose $x \in S$ and assign to all vertices adjacent to $x$ the opposite color.

$G$ is 2-colorable iff step 2 never leads to a contradiction.

---

** Decision 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 $k$)**

** 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?

---

** NP-Hardness Proof of 3-COLORABILITY**

By reduction from NAESAT: Let an arbitrary instance of NAESAT be given by a Boolean formula $\varphi = c_1 \land \ldots \land c_m$ in 3-CNF with variables $x_1, \ldots, x_n$. We construct the following graph $G(\varphi)$:

Let $V = \{a\} \cup \{x_i, \neg x_i | 1 \leq i \leq n\} \cup \{l_1, l_2, l_3 | 1 \leq i \leq m\}$, i.e., $|V| = 1 + 2n + 3m$.

For each variable $x_i$ in $\varphi$, we introduce a triangle $[a, x_i, \neg x_i]$, i.e., all these triangles share the node $a$.

For each clause $c_i$ in $\varphi$, we introduce a triangle $[l_1, l_2, l_3]$. Moreover, each of these vertices $l_j$ is further connected to the node corresponding to this literal, i.e.: if the $j$-th literal in $c_i$ is of the form $x_a$ (resp. $\neg x_a$) then we introduce an edge between $l_j$ and $x_a$ (resp. $\neg x_a$).
Example

The 3-CNF formula \( \varphi = (x_1 \lor \neg x_2 \lor x_3) \land (x_2 \lor x_3 \lor \neg x_4) \) is reduced to the following graph:

```
  a
 / \ /
|  | |
| b1| b2|
 \ | /  \\
  x1 -b1-x2 -b2-x3 -b3-x4
```

Correctness of the Problem Reduction

Proof (continued)

"\( \Rightarrow \)" Suppose that \( G \) has a 3-coloring with colors \( \{0, 1, 2\} \). W.l.o.g., the node \( a \) has the color 2. This induces a truth assignment \( T \) via the colors of the nodes \( x_i \): if the color is 1, then \( T(x_i) = \text{true} \) else \( T(x_i) = \text{false} \). We claim that \( T \) is a legal NAESAT-assignment. Indeed, if in some clause, all literals had the value \( \text{false} \) (resp. \( \text{true} \)), then we could not use the color 0 (resp. 1) for coloring the triangle \( \{l_1, l_2, l_3\} \), a contradiction.

"\( \Leftarrow \)" Suppose that there exists an NAESAT-assignment \( T \) of \( \varphi \). Then we can extract a 3-coloring for \( G \) from \( T \) as follows:

(i) If \( T(x_i) = \text{true} \), then color \( x_i \) with 1 and \( \neg x_i \) with 0 else vice versa.
(ii) From each \( \{l_1, l_2, l_3\} \), color two literals having opposite truth values with 0 (\text{true}) and 1 (\text{false}). Color the third with 2.

\[ a \]

\[ x_1 \]
\[ \neg x_1 \]
\[ x_2 \]
\[ \neg x_2 \]
\[ x_3 \]
\[ \neg x_3 \]
\[ x_4 \]
\[ \neg x_4 \]

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.

TSP(D)

INSTANCE: \( n \) cities \( 1, \ldots, n \) and a nonnegative integer distance \( d_{ij} \) between any two cities \( i \) and \( j \) (such that \( d_{ij} = d_{ji} \)), and an integer \( B \).
QUESTION: Is there a tour through all cities of length at most \( B \)?
i.e., a permutation \( \pi \) s.t. \( \sum_{i=1}^{n} d_{\pi(i)\pi(i+1)} \leq B \) with \( \pi(n+1) = \pi(1) \).
Complexity Theory 5. NP-Completeness 5.7. HAMILTON-PATH, etc.

Complexity

Theorem

HAMILTON-PATH, HAMILTON-CYCLE, and TSP(D) are NP-complete.

Proof

We shall show the following chain of reductions:

HAMILTON-PATH ≤L HAMILTON-CYCLE ≤L TSP(D)

It suffices to show NP-membership for the hardest problem:
1. Guess a tour π through the n cities.
2. Check that \( \sum_{i=1}^{n} d_{π(i)π(i+1)} \leq B \) with \( π(n+1) = π(1) \).

Likewise, it suffices to prove the NP-hardness of the easiest problem.

The NP-hardness of HAMILTON-PATH (by a reduction from 3-SAT) is quite involved and is therefore omitted here (see Papadimitriou's book).

HAMILTON-CYCLE vs. TSP(D)

HAMILTON-CYCLE ≤L TSP(D)

Let an arbitrary instance of HAMILTON-CYCLE be given by the graph \( G = (V, E) \). We construct an equivalent instance of TSP(D) as follows:

Let \( V' := V \cup \{ z \} \) for some new vertex \( z \) and \( E' := E \cup \{ [v, z] \mid v \in V \} \).

G has a Hamilton path ⇔ \( G' \) has a Hamilton cycle

“⇒” Suppose that \( G \) has a Hamilton path \( π \) starting at vertex \( a \) and ending at \( b \). Then \( π \cup \{ z \} \) is clearly a Hamilton cycle in \( G' \).

“⇐” Let \( C \) be a Hamilton cycle in \( G' \). In particular, \( C \) goes through \( z \). Let \( a \) and \( b \) be the two neighboring nodes of \( z \) in this cycle. Then \( C \setminus \{ z \} \) is a Hamilton path (starting at vertex \( a \) and ending at \( b \)) in \( G \).

Summary of Reductions

SAT
4-SAT
3-SAT
CIRCUIT-SAT

1-in-3-SAT

MON 1-in-3-SAT

HITTING SET

3-COL

INDEPENDENT SET

VERTEX COVER

CLIQUE

HAM.-PATH

HAM.-CYCLE

TSP(D)
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)
- Basic techniques to prove problems NP-complete
- A basic repertoire of NP-complete problems (in particular, versions of SAT and some graph problems) to be used in further NP-completeness proofs.
- Reductions, reductions, reductions, . . .