NP-Completeness

**Def** A language $L$ is called \(NP\)-complete if

1. \(L \in NP\)
2. \(\forall\) language \(A \in NP\), \(A \leq_p L\).

**Theorem** The following language is \(NP\)-complete:

\[
L = \{ <M, x, \#^k> | M \text{ is a NTM that has accepting computation on } x \text{ with } \leq k \text{ steps} \}
\]

**Proof**

1. \(L \in NP\) since it is accepted by

\[
M_{sim} := "\text{On input } <M, x, \#^k>, \\
\text{Simulate } M \text{ on } x \text{ for at most } k \text{ steps: } \\
\text{if } M \text{ accepts, accept } \\
\text{else reject}".
\]

Note: Since \(M\) is NTM, so is \(M_{sim}\).
\[ \langle M, x, \#^k \rangle \in L \implies M \text{ has accepting computation on } x \text{ w/ } \leq k \text{ steps} \]

\[ \implies M_{\text{sim}} \text{ has an accepting computation} \]

\[ \langle M, x, \#^k \rangle \notin L \implies M \text{ has no accepting computation on } x \text{ w/ } \leq k \text{ steps} \]

\[ \implies M_{\text{sim}} \text{ has no accepting computation} \]

Further, \( M_{\text{sim}} \) runs in time \( \text{poly}(|\langle M \rangle|, |x|, k) \).

2. Now we show that \( \forall A \in \text{NP}, A \leq_p L \).

Let \( A \) be accepted by polytime NTM \( M_A \).

Here is the polytime \( \sim \) reduction from \( A \) to \( L \):

\[ A \rightarrow L \]

\[ x \sim < M_A, x, \#^{n^c} > \]

where \( |x| = n \) and \( M_A \) has runtime \( n^c \).
$x \in A \iff M_A$ has an accepting computation on $x$ with $\leq n^c$ steps, $|x| = n$

$\iff \langle M_A, x, \#n^c \rangle \in L$.

This shows that NP-complete languages exist. It turns out that many "natural" languages/problems are NP-complete!

**Theorem** [Cook-Levin] 3SAT is NP-complete.

**Theorem** To show that a language $C$ is NP-complete, one shows that

1. $C \in$ NP.

2. For some NP-complete language $B$, $B \leq_p C$. 

\[ B \rightsquigarrow C \quad \text{NP} \]

\[ A \]
Proof

1. Given, \( C \in \text{NP} \).
2. We need to show that \( \forall A \in \text{NP}, \ A \leq_p C \).

Since \( B \) is \( \text{NP}-\text{complete} \), \( A \leq_p B \).

Given, \( B \leq_p C \).

Hence \( A \leq_p C \).

Now we show that the following problems are \( \text{NP}-\text{complete} \):

- \( \text{3SAT} \)
- \( \text{INDEPENDENT SET} \)
- \( \text{CLIQUE} \)
- \( \text{VERTEX COVER} \)
- \( \text{SUBSET SUM} \)
- \( \text{HAMiltonian CYCLE} \)
- \( \text{TRAVELING SALESPerson} \)
**3SAT**

Example of a 3SAT instance:

\[(x \lor y \lor z) \land (x \lor \overline{y} \lor \overline{w}) \land (\overline{y} \lor u \lor v) \land (z \lor \overline{u} \lor \overline{v}) \land (\overline{x} \lor v \lor z \lor w).\]

**Def** A **3CNF** 3SAT formula \(\phi\) is

\[\phi = C_1 \land C_2 \land C_3 \ldots \land C_m\]

where each clause \(C_r\), \(1 \leq r \leq m\), is of form \(C_r = l_i \lor l_j \lor l_k\), \(1 \leq i, j, k \leq n\).

Here \(x_1, x_2, \ldots, x_n\) are Boolean variables and \(l_i = x_i\) or \(l_i = \overline{x_i}\) are literals.

**Def** A 3CNF formula \(\phi\) has a satisfying assignment (\(\phi\) is satisfiable) if there is an assignment \(\sigma: \{x_1, x_2, \ldots, x_n\} \rightarrow \{\text{True}, \text{False}\}\) that makes \(\phi\) evaluate to True (i.e., makes every clause \(C_r\) True).
3SAT = \{ \langle \phi \rangle \mid \phi \text{ has a satisfying assignment} \}.

Note 3SAT ∈ NP. Following polytime NTM accepts it.

M := " on input \langle \phi \rangle,
let \( x_1, \ldots, x_n \) be variables of \( \phi \).
Guess an assign. \( \sigma : \{ x_1, \ldots, x_n \} \rightarrow \{T,F\} \)
Accept if \( \sigma \) satisfies \( \phi \).
Reject otherwise."

Theorem 3SAT is NP-complete.

Note 3SAT can be thought of as the language as above
or equivalently as the problem of
deciding, given instance \( \phi \),
whether \( \phi \) has a satisfying assign.
INDEPENDENT SET

Def In a graph $G(V, E)$, an independent set $I \subseteq V$ is a subset s.t. 
$\forall a, b \in I, (a, b) \notin E$.

Def
$INDEP\text{-}SET = \{ \langle G, k \rangle \mid G$ is a $n$-vertex graph that has an independent set of size $\geq k \}$.

Theorem INDEP\text{-}SET is NP-complete.

Proof 1 INDEP\text{-}SET $\in$ NP. Following polytime NTM accepts it.

$M := "$ Given $\langle G, k \rangle$, $G = G(V, E)$,
Non-det. select a subset $I \subseteq V$.
Accept if $|I| \geq k$ and $I$ is an independent set.
Reject otherwise."

2 We show that 3SAT reduces to INDEP\text{-}SET.
3SAT $\rightarrow$ INDEP. SET

$\phi$ $\rightarrow$ $\langle G, k \rangle$

$\text{Sat.}$

$\frac{\phi \in 3\text{SAT}}{\phi \in 3\text{SAT} \iff \langle G, k \rangle \in \text{INDEP. SET}}.$

We need to show that:

(a) $\phi \in 3\text{SAT} \Rightarrow \langle G, k \rangle \in \text{INDEP. SET}$.
(b) $\langle G, k \rangle \in \text{INDEP. SET} \Rightarrow \phi \in 3\text{SAT}$.

**Reduction**

$\phi$

vars: $x_1, x_2, \ldots, x_n$ $\rightarrow$ $\langle G, k \rangle$

clauses: $C_1, C_2, \ldots, C_m$

- For each clause, construct a triangle whose vertices are labeled by the three literals in that clause.
- In addition, every pair of vertices labeled as \( x_i \) and \( \overline{x}_i \) are connected by an edge.
- This yields the graph \( G \). Let \( k = m \).

\[ \phi \text{ has a satisfying assignment} \implies G \text{ has an independent set of size } m. \]

**Proof** Let \( \sigma : \{x_1, \ldots, x_n\} \to \{T, F\} \) be a satisfying assignment. For every \( i, 1 \leq i \leq n \), either \( x_i \) or \( \overline{x}_i \) is set to True (but not both). Since \( \sigma \) satisfies \( \phi \), every clause \( C_r \) contains a True literal.

Let \[ I = \text{Subset of vertices obtained by picking one vertex from each clause/triangle } C_r \text{ that is labeled by a True literal} . \]

Clearly \[ |I| = m. \]
Moreover, I is independent since it consists of only True literals and hence does not contain both $x_i$ and $\overline{x_i}$ for any $1 \leq i \leq n$.

\( G \) has an independent set of size $m$ \implies \( \phi \) has a satisfying assignment.

**Proof** Note first that the said independent set $I$ of size $m$ must contain exactly one vertex from each triangle/clause. Declare all literals (labels of vertices in $I$) in $I$ to be True. This defines an assignment $\sigma$ to the variables $\{x_1, \ldots, x_n\}$ in an unambiguous manner since $I$ does not contain a pair $(x_i, \overline{x_i})$.

Since $I$ contains one vertex from each triangle, each clause contains a True literal. Hence $\sigma$ is a satisfying assignment to $\phi$.  

\[ \square \]
**CLIQUE**

**Def** A clique in a graph $G(V,E)$ is a subset $C \subseteq V$ s.t.: 

$\forall a, b \in C, a \neq b, (a, b) \in E$.

**Def** $\overline{G}$ is the complement graph where:

- $\overline{G}$ has the same vertex set $V$ as $G(V,E)$.
- $\forall a, b \in V, a \neq b,$

$(a, b)$ is an edge $\iff (a, b)$ is not an edge in $G$.

**Theorem** CLIQUE is NP-complete where

$\text{CLIQUE} = \{ <G, k> \mid G \text{ has a clique of size } \geq k \}$.

**Proof**

4) CLIQUE is NP.

Blah Blah Blah ... 

2) INDEPENDENT-SET reduces to CLIQUE.
Theorem \[ \text{INDEP. SET} \leq_p \text{CLIQUE} \]

Proof \[ \text{INDEP. SET} \rightarrow \text{CLIQUE} \]

\[ \langle G', k' \rangle \leadsto \langle G, k \rangle. \]

Let \[ G = \overline{G'}, \quad k = k'. \]

Self-evident that \[ G' \text{ has indep. set } \iff G = \overline{G'} \text{ has a clique of size } k' = k. \]

**VERTEX-COVER**

**Def** A vertex cover in a graph \( G(V, E) \) is a subset \( S \subseteq V \) such that \( \forall (a, b) \in E, \text{ either } a \in S \text{ or } b \in S. \)

**Fact** \( S \) is a vertex cover \[ \iff V \setminus S \text{ is an indep. set.} \]
Theorem \( \text{VERTEX-COVER} \) is \( \text{NP} \)-complete.

\[
\text{VERTEX-COVER} = \left\{ \langle G, k \rangle \mid G \text{ has a vertex cover of size } \leq k \right\}
\]

Proof

1. \( \text{VERTEX-COVER} \in \text{NP} \).

   Blah Blah Blah ...

2. \( \text{INDEPENDENT-SET} \leq_p \text{VERTEX-COVER} \).

   \( \langle G', k' \rangle \sim \rightarrow \langle G, k \rangle \).

\( G' \) has independent set of size \( k' \) \( \iff \) \( G \) has a vertex cover of size \( k \).

Reduction

Let \( G = G' \).

\[ k = n - k' \quad n = |G'|. \]

Self-evident that \( G' \) has independent set \( I \) of size \( k' \) \( \iff \) \( G = G' \) has vertex cover \( V \setminus I \) of size \( n - k' = k \).
Theorem HAMILTONIAN_CYCLE is NP-complete.

HAM-CYCLE = \{ \langle G \rangle \mid G \text{ has a Hamiltonian cycle} \}.

We skip the proof (= reduction from 3SAT).

Theorem T.S.P. is NP-complete.

T.S.P. = \{ \langle G, wt, l \rangle \mid G \text{ is a complete graph} \\
\text{with wt on the edges} \& \text{has a tour of length } \leq l \}.

Proof

1. T.S.P. \in NP.
   Blah Blah Blah ...

2. HAMILTONIAN-CYCLE \leq_p T.S.P.

   G' \iff \langle G, wt, l \rangle.
   G'(V', E')

Reduction - Let G be complete graph on same vertex set V'.

- wt(e) = \{ 1 \text{ if } e \in E' \\\n 2 \text{ if } e \notin E' \}

- l = n.
Now we show that

\[ G' \text{ has Hamiltonian } \iff <G', \omega_f> \text{ has a tour of length } \leq n. \]

(a) \implies: The Hamiltonian cycle in \( G' \) serves as a tour in \( <G', \omega_f> \); its length is \( n \) since all its edges are edges of \( G' \).

(b) \implies: Consider a tour in \( <G', \omega_f> \) of length \( \leq n \). Since it has \( n \) edges and all edge weights are 1 or 2, the edges of the tour all must have weight 1. Hence the tour must correspond to a Hamiltonian cycle in \( G' \).
SUBSET-SUM

\[ \text{SUBSET-SUM} = \left\{ (a_1, \ldots, a_n; t) \left| a_1, \ldots, a_n, t \text{ are non-negative integers and} \right. \ \exists S \subseteq \{1, \ldots, n\}, \sum_{i \in S} a_i = t \right\} \]

Note: \(a_1, \ldots, a_n, t\) are represented in binary or decimal.

**Theorem**

\[ \text{SUBSET-SUM is NP complete.} \]

**Proof**

1. \[ \text{SUBSET-SUM} \in \text{NP} \]

   Blah Blah Blah ....

2. We show that 3SAT \( \leq_p \) SUBSET-SUM.

   - Let \( \phi \) be a 3SAT formula.

     vars: \( x_1, \ldots, x_n \)

     clauses: \( C_1, C_2, \ldots, C_m \)

   - The SUBSET-SUM instance we construct:
     - has integers in decimal.
     - no carries while adding integers.
     - digits not shown are 0.

   described
- rows represent the (decimal) integers.

- For illustration:  
  \[ C_1 : x_1 \lor \bar{x}_2 \lor \bar{x}_n \]
  \[ C_2 : x_1 \lor x_2 \lor x_n \]
The SUBSET-SUM instance is as follows:

- **Top right block**: An entry is 1 if the literal belongs to the clause and 0 otherwise.

- **Remaining three blocks**: As shown, and the target \( t \).

- **Note**: Only way to pick a subset of rows that sum to \( t \) is to
  - pick exactly one of the two rows labeled \( x_i \) or \( \overline{x_i} \), for every \( 1 \leq i \leq n \).
  - pick none, one, or both of the rows labeled \( g_j \) or \( h_j \), for every \( 1 \leq j \leq m \).

Now we prove that

\( \phi \) has a satisfying assignment \( \iff \) There is a subset of rows that sums to \( t \).
proof of \( \Rightarrow \):

- Let \( \sigma \) be a satisfying assignment.
- For \( 1 \leq i \leq n \), between rows labeled \( x_i \) or \( \bar{x}_i \), pick the one corresponding to the True literal.
- For each clause \( C_j \), at least one of its literals is True and hence one, two, or three rows corresponding to its literals have been picked.
- Thus (in top-right block), column for \( C_j \) has sum 1, 2, or 3.
- Depending on these three cases, we take both, only \( g_j \), or none from the rows \( g_j \), \( h_j \) so that the sum in that column is exactly 3.
Proof of $\subseteq S$

Suppose there is a subset of rows that sums to $t$.

As noted, for every $i$, $1 \leq i \leq n$, exactly one of $x_i$ or $\overline{x}_i$ row is in $S$, and declare that literal to be True. This defines an assignment $\sigma$ to $x_1, x_2, \ldots, x_n$.

To show that $\sigma$ satisfies each clause $C_j$, we note that $C_j$ must contain a True literal. Otherwise (in the top right block), the sum in column $C_j$ is zero and even if one were to take both rows $g_j, h_j$, one wouldn't reach the sum 3 in that column.