CNF-Satisfiability: SAT

- **Input:** Expression over Boolean variables in conjunctive normal form (CNF).
  - “and” of clauses;
  - each clause “or” of variables or negations ($x_i$ or not($x_j$))

- **Question:** Is the expression satisfiable?
  I.e., can we give each variable a value (true or false) such that the expression becomes true?
Cook-Levin theorem

**Theorem:** CNF-Satisfiability is NP-complete.

- Most well known is Cook’s proof, using Turing machine characterization of NP.
- It design a Turing machine that verifies yes-Instances of SAT
Lemma (key in the proof)

1. Let $P' \leq_p P$ and let $P'$ be NP-complete. Then $P$ is NP-hard.

2. Let $P' \leq_p P$ and let $P'$ be NP-complete, and $P \in \text{NP}$. Then $P$ is NP-complete.
Proving problems NP-complete: General recipe for a reduction

Suppose that you want to show NP-completeness of problem $B$;
1. Show that $B$ belongs to the class NP.
2. Assume that you know that problem $A$ is NP-complete. Show that $A$ is reducible to $B$: $A \leq_p B$.
   - Take an arbitrary instance $I$ of problem $A$.
   - Indicate how you can construct a special instance $f(I)$ of problem $B$ on basis of the instance of $A$ that you selected: the answers to the instance of $A$ and the special instance of $B$ must be equal.
   - This transformation (or construction) must be possible in polynomial time.
Reduction: first example

- the **Subset sum** problem
  - Given $n$ non-negative integral values $a_1, a_2, \ldots a_n$ and a nonnegative integer $B$ does there exist a subset $S$ of the index-set $\{1, 2, \ldots, n\}$, such that $\sum_{j \in S} a_j = B$

- To prove: **Subset sum is NP-complete**

- the **Partition** problem
  - Given $n$ non-negative integral values $a_1, a_2, \ldots a_n$ does there exist a subset $S$ of the index-set $\{1, 2, \ldots, n\}$, such that $\sum_{j \in S} a_j = \frac{1}{2} \sum_{j=1}^{n} a_j$

- Suppose that we know that the Partition problem is NP-complete;
Proof Subset sum is NP-complete

**Subset sum** is in \textbf{NP}:
- Input size $O(n \log B)$.
- A solution leading to yes is a subset of \{1,2,\ldots,n\}. Can be encoded in polynomial time.
- Checking if a solution leads to yes is adding the included numbers $a_i$ and comparing to $B$: polynomial.

**Reduction from \textit{Partition}**.
- Let $a_1, a_2, \ldots a_n$ an instance from Partition. Construct an instance from Subset sum with the same values $a'_i = a_i \forall i$ and $B = \frac{1}{2} \sum_{j=1}^{n} a_j$.
- We have yes in Partition if and only if we have yes in Subset sum.
- Transformation can be performed in polynomial time.
3-SAT

3-Sat is the special case of CNF-Satisfiability where each clause has exactly three literals.

Lemma: CNF-Satisfiability \(\leq_p\) 3-Sat

- Clauses with one or two literals:
  - Use two extra variables \(p\) and \(q\).
  - Replace \((x \lor y)\) by \((x \lor y \lor p)\) and \((x \lor y \lor \neg p)\).
  - Similarly, replace a 1-literal clause by 4 clauses.

- Clauses with more than three literals:
  - For \((l_1 \lor l_2 \lor \ldots \lor l_r)\) add new variable \(t\) and take as replacement clauses \((l_1 \lor l_2 \lor t)\) and \((\neg t \lor l_3 \lor \ldots \lor l_r)\).
  - Repeat this until the last clause has size three.
3-SAT is NP-complete

Proof:
1. Membership in NP (easy to check).
2. Reduction (previous slide).

3-Sat is important starting problem for many NP-completeness proofs.
Clique

- Clique: set of vertices $W \subseteq V$, such that for all $v, w \in W$: 
  $\{v, w\} \in E$.

Clique

- Given: graph $G = (V, E)$, integer $k$.
- Question: does $G$ have a clique with at least $k$ vertices?
Clique is NP-complete

- Clique in **NP**.
  - Graph \((V,E)\) with \(n\) nodes can be encoded in \(O(n^2)\) bits; \(k\) can be encoded in \(\log(n)\) bits
  - A solution is a set of nodes \(S\), can be encoded in \(O(n)\) bits
  - Checking if \(S\) is a clique of size at least \(k\), takes \(O(n^2)\)

- Reduction from 3-SAT
  - Let \(x\) be an instance from 3-SAT
  - Define instance \(f(x)\) from Clique
  - Make clear that \(f\) can be performed in polynomial time
  - Show \(x\) is yes-instance iff \(f(x)\) is yes-instance
  - *All detailed in next slides*
Let \( x \) be an instance of 3-SAT

Define the following instance of clique:
- One vertex per literal per clause.
- Edges between vertices in different clauses, except edges between \( x_i \) and \( \neg x_i \).
- If \( x \) has \( m \) clauses, look for clique of size \( m \).

Idea: you can make a clique from the literals that are true.

\[ \text{Clause: } \{x_1, \neg x_2, x_3\} \]
\[ \text{Clause: } \{x_1, x_2, \neg x_3\} \]
Correctness

Let $x$ be a satisfying truth assignment.

- Select from each clause one true literal (there must be at least one since $x$ is true).
  - Since vertices in different clauses, except $x_i$ and $\neg x_i$ are connected, the corresponding vertices form a clique with $m$ vertices.

Suppose $f(x)$ has a clique of size $m$

- Set variable $x_i$ to true, if a vertex representing $x_i$ is in the clique, otherwise set it to false. This is a satisfying truth assignment:
  - It cannot contain a vertex representing $x_i$ and a vertex representing $\neg x_i$, so well-defined
  - The clique must contain one vertex from each 3 vertices representing a clause (vertices within a clause are not connected), so true
Independent set

- **Independent set**: set of vertices $W \subseteq V$, such that for all $v, w \in W$: $\{v, w\} \not\in E$.

- **Independent set problem**:
  - Given: graph $G$, integer $k$.
  - Question: Does $G$ have an independent set of size at least $k$?

- Independent set is NP-complete.
Independent set is NP-complete

- In NP.
- Reduction: transform from Clique.
- $W$ is a clique in $G$, if and only if, $W$ is an independent set in the complement of $G$ (there is an edge in $G^c$ iff. there is no edge in $G$).
Theorem: Independent Set is NP-complete.

Proof:
- The problem belongs to NP:
  - Solutions are subsets of vertices of the input graph; polynomial size
  - We can check in polynomial time for a given subset of vertices that it is an independent set and that its size is at least k.
- We use a reduction from Clique.
  - Let \((G,k)\) be an instance of the clique problem.
  - Transform this to instance \((G^c,k)\) of the independent set problem with \(G^c\) the complement of \(G\).
  - As \(G\) has a clique with \(k\) vertices, if and only if, \(G^c\) has an independent set with \(k\) vertices, this is a correct transformation.
  - The transformation can clearly be carried out in polynomial time.
Writing an NP-Completeness proof

- State the theorem.
- Proof starts with showing that problem belongs to NP.
  - Give/explain the encoding of a solution leading to yes,
  - show/state that it is polynomial in the size of instance and
  - explain how (or state that, if trivial) a yes-solution can verified in
    polynomial time w.r.t. the length of instance and solution.
- State which known NP-complete problem you reduce from.
  - Careful: do not go in the wrong direction.
- Explain the transformation (also called reduction).
- Give the proof: instance to original problem is Yes-instance, if
  and only if, transformed instance is Yes-instance for the known
  NP-complete problem.
  - Remember: you need to prove this in two directions.
- Phrase (or prove if not trivial): transformation can be carried
  out in polynomial time, hence problem is NP-complete.
- QED.
Vertex Cover

- Set of vertices $W \subseteq V$ with for all $\{x, y\} \in E$: $x \in W$ or $y \in W$.

- **Vertex Cover** problem:
  - Given $G$, find vertex cover of minimum size.
  - Vertex cover is NP-complete: exercise
Given: Graph $G=(V,E)$

Question: Can we colour the vertices with 3 colours, such that for all edges $\{v,w\}$ in $E$, the colour of $v$ differs from the colour of $w$.

3-colouring is NP-Complete.
Proof of NP-Completeness of 3-Colouring

- In NP:
  - Encoding a solution: colour for each vertex (O(n))
  - Checking if a solution is a yes instance: O(n^2).
- Reduction from 3-SAT:
  Given an instance from 3-SAT
  We build a graph in 3 steps:
  1. Take a clique with 3 vertices True, False, C.
  2. Take two adjacent vertices for each variable x.

We name the colours: T, F, C
NP-Completeness of 3-Colouring

3. For each clause \{l_1, l_2, l_3\}, take the following gadget:
The transformation takes polynomial time.

Suppose the formula is satisfiable.
- Colour the variables $T$ or $F$ according to their truth value. By making proper case distinction you can show that this is a 3-colouring of $G$.

Suppose there is a 3-colouring of $G$.
- Consider the following solution for the SAT formula. Give the variables with colour $T$ assignment true and the ones with colour $F$ assignment false.
- You can check that $l_1 = l_2 = l_3 = F$ is not a feasible 3-colouring. So SAT formula must be true.

Note: In both cases, the intuition is that a literal vertex is coloured $T$ (true), if and only if, we take it to be true in the formula.