Denis Trystram
MoSIG1 and M1Info – University Grenoble-Alpes

March, 2021
Agenda

- Horn-SAT
- 2SAT
- analysis of CLIQUE
- Dynamic Programming for SubSetSum
- Bin Packing
Complexity of Horn-SAT

A Horn formula has at most one positive literal per clause.

\[ \text{HORN-SAT} = \{ \langle \mathcal{F} \rangle \mid \mathcal{F} \text{ is a satisfiable Horn formula} \} \]

Recall:

- Positive literal: \( x_i \)
- Negative literal: \( \bar{x}_i \)

Tip:

- What has to happen to clauses that contain only one single literal?
- Consider the case that each clause contains a negative literal.
A Horn formula has at most one positive literal per clause.

\[
\text{Horn-SAT} = \{ \langle \mathcal{F} \rangle \mid \mathcal{F} \text{ is a satisfiable Horn formula} \}
\]

Recall:
- Positive literal: \( x_i \)
- Negative literal: \( \overline{x_i} \)

Prove that Horn-SAT \( \in \mathcal{P} \)

**Tipp:**
- What has to happen to clauses that contain only one single literal?
- Consider the case that each clause contains a negative literal.
Algorithm

1. **While** there are clauses with only one literal
   - pic a clause with only one literal
   - set the corresponding variable to $T$ or $F$ such that the clause is satisfied
   - delete all the other clauses that are satisfied by this assignment and remove the variable from all the other clauses

2. set all non-assigned variables to $F$
Solution Horn-SAT

Algorithm

1. **While** there are clauses with only one literal
   - pic a clause with only one literal
   - set the corresponding variable to $T$ or $F$ such that the clause is satisfied
   - delete all the other clauses that are satisfied by this assignment and remove the variable from all the other clauses

2. set all non-assigned variables to $F$

**Sketch of the analysis:**
After step 1 all the clauses contain at least one negative literal. Therefore, after setting all variables to $F$ in step 2, every clause will contain at least one literal that is $T$. Hence, all the clauses are satisfied.

Complexity is in $O((n \cdot m)^2)$
2SAT

- $X = \{x_1, x_2, \ldots, x_n\}$: set of variables
- $C = \{C_1, C_2, \ldots, C_m\}$: set of clauses for cardinality 2
- $\mathcal{F} = C_1 \land C_2 \land \ldots \land C_m$

SAT = $\{\langle \mathcal{F} \rangle \mid \mathcal{F} \text{ is a satisfiable Boolean formula} \}$

Prove $2\text{SAT} \in \mathcal{P}$

The solution is detailed in the slides of lecture 4: variants of SAT.
CLIQUE = \{ \langle G, k \rangle \mid G = (V, E) \text{ is a graph with a subset of vertices } A \text{ of cardinality } k \text{ and for each pair of vertices in } A, (x, y) \in E \}
CLIQUE ∈ NP-complete

CLIQUE ∈ NP

- given a set of vertices, check if there is an edge between any pair of them
CLIQUE ∈ NP

- given a set of vertices, check if there is an edge between any pair of them

3SAT ≤_P CLIQUE

1. given any formula \( \mathcal{F} \) of SAT, we construct an instance \( I = \langle G, k \rangle \) of CLIQUE
   - add a vertex for each literal
   - add an edge between any two literals except:
     (a) literals in the same clause
     (b) a literal and its negation
   - \( k = m \) (number of clauses)
\[ F = (x_1 \lor x_2 \lor \overline{x}_3) \land (x_1 \lor x_3 \lor x_4) \land (\overline{x}_2 \lor x_3 \lor \overline{x}_4) \]
CLIQUE $\in$ NP-complete

3SAT $\leq_P$ CLIQUE

2. $|V| = 3m$, $|E| = O(m^2)$
CLIQUE ∈ NP-complete

3SAT ≤_P CLIQUE

2. \(|V| = 3m, |E| = O(m^2)\)

3. \(F\) is satisfiable iff there is a clique of size \(k\) in \(G\)
   - assume that \(F\) is satisfiable
   - at least one literal is TRUE in any clause
   - there is an edge between such literals (why?)
   - hence, the corresponding vertices form a \(k\)-clique
CLIQUE ∈ NP-complete

3SAT ≤_P CLIQUE

2. |V| = 3m, |E| = O(m^2)

3. \( \mathcal{F} \) is satisfiable iff there is a clique of size \( k \) in \( G \)
   - assume that \( \mathcal{F} \) is satisfiable
   - at least one literal is TRUE in any clause
   - there is an edge between such literals (why?)
   - hence, the corresponding vertices form a \( k \)-clique

   - assume there is a \( k \)-clique in \( G \)
   - this clique contains at most one vertex from each clause
   - \( k = m \), hence the clique contains exactly one vertex from each clause
   - each pair of these vertices is compatible (no a literal and its negation)
   - set the corresponding literals to TRUE
   - \( \mathcal{F} \) is satisfiable
Solving \textbf{SubsetSum}

\textbf{SubsetSum}

\textbf{Input:} a set of positive integers \( A = \{a_1, a_2, \ldots, a_k\} \)
\( t \in \mathbb{N} \)

\textbf{Question:} is there a set \( B \subseteq A \) such that \( \sum_{a_i \in B} a_i = t \)?

Write a dynamic programming algorithm for solving this problem.
Solving SubsetSum

**SubsetSum**

Input: a set of positive integers \( A = \{a_1, a_2, \ldots, a_k\} \)

\( t \in \mathbb{N} \)

Question: is there a set \( B \subseteq A \) such that \( \sum_{a_i \in B} a_i = t \)?

Write a dynamic programming algorithm for solving this problem.

**Tip:**

- Consider the integers sorted in non-decreasing order:
  \( a_1 \leq a_2 \leq \ldots \leq a_n \)

\[ S[i, q] = \begin{cases} 
  \text{True,} & \text{if there is a SubsetSum among the } i \text{ first} \\
  \text{False,} & \text{otherwise} 
\end{cases} \]

The detailed solution is in the slides of Lecture 4 *pseudo-polynomial algorithms.*
Bin Packing

**Bin-Packing**

**Input:** a set of items $A$, a size $s(a)$ for each $a \in A$, a positive integer capacity $C$, and a positive integer $k$

**Question:** is there a partition of $A$ into disjoint sets $A_1, A_2, \ldots, A_k$ such that the total size of the elements in each set $A_j$ does not exceed the capacity $C$, i.e., $\sum_{a \in A_j} s(a) \leq C$?

Show that this problem is **NP-complete**

Is it strongly or weakly **NP-complete**?

(try to give the strongest result)