Theory of Computer Science

E5. Some NP-Complete Problems, Part II

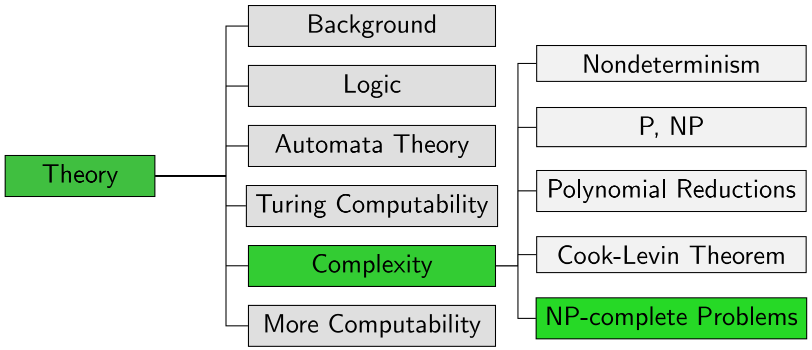
Gabriele Röger

University of Basel

May 13, 2019

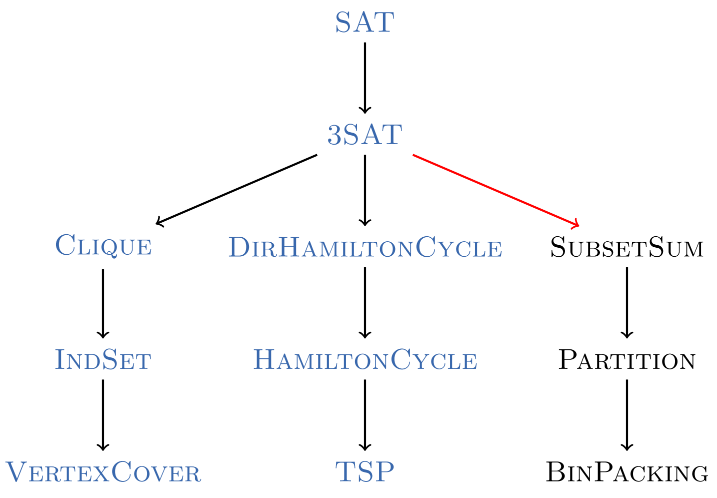
Theory of Computer Science
May 13, 2019 — E5. Some NP-Complete Problems, Part II

Course Overview



E5.1 Packing Problems

3SAT $\leq_p$ SubsetSum



SubsetSum is NP-Complete (1)

Definition (SubsetSum)

The problem SubsetSum is defined as follows:

Given: numbers $a_1, \dots, a_k \in \mathbb{N}_0$ and $b \in \mathbb{N}_0$

Question: Is there a subset $J \subseteq \{1, \dots, k\}$ with $\sum_{i \in J} a_i = b$?

Theorem

SubsetSum *is NP-complete.*

SubsetSum is NP-Complete (2)

Proof.

SubsetSum $\in$ NP: guess and check.

SubsetSum is NP-hard: We show 3SAT $\leq_p$ SubsetSum.

Given a 3-CNF formula φ, we compute a SubsetSum instance that has a solution iff φ is satisfiable.

We can assume that all clauses have exactly three literals and that the literals in each clause are unique.

Let m be the number of clauses in φ,
and let n be the number of variables.

Number the propositional variables in φ in any way,
so that it is possible to refer to "the i-th variable". ...

SubsetSum is NP-Complete (3)

Proof (continued).

The target number of the SubsetSum instance is
$\sum_{i=1}^{n} 10^{i-1} + \sum_{i=1}^{m} 4 \cdot 10^{i+n-1}$
(in decimal digits: m 4s followed by n 1s).

The numbers to select from are:

- one number for each literal (X or $\neg X$):
  if the literal belongs to the j-th variable and occurs (exactly) in the k clauses $i_1, \dots, i_k$, its literal number is
  $10^{j-1} + 10^{i_1+n-1} + \dots + 10^{i_k+n-1}$.
- for each clause, two padding numbers:
  10^{i+n-1} and $2 \cdot 10^{i+n-1}$ for all $i \in \{1, \dots, m\}$.

This SubsetSum instance can be produced in polynomial time.

...

SubsetSum is NP-Complete (4)

Proof (continued).

Observations:

- With these numbers, no carry occurs in any subset sum. Hence, to match the target, all individual digits must match.
- For $i \in \{1, \dots, n\}$, refer to the i-th digit (from the right) as the i-th variable digit.
- For $i \in \{1, \dots, m\}$, refer to the $(n + i)$-th digit (from the right) as the i-th clause digit.
- Consider the i-th variable digit. Its target value is 1, and only the two literal numbers for this variable contribute to it.
- Hence, for each variable X, a solution must contain either the literal number for X or for $\neg X$, but not for both.

...

SubsetSum is NP-Complete (5)

Proof (continued).

- Call a selection of literal numbers that makes the variable digits add up a candidate.
- Associate each candidate with the truth assignment that satisfies exactly the literals in the selected literal numbers.
- This produces a 1:1 correspondence between candidates and truth assignments.
- We now show: a given candidate gives rise to a solution iff it corresponds to a satisfying truth assignment.
- This then shows that the SubsetSum instance is solvable iff φ is satisfiable, completing the proof.

...

SubsetSum is NP-Complete (6)

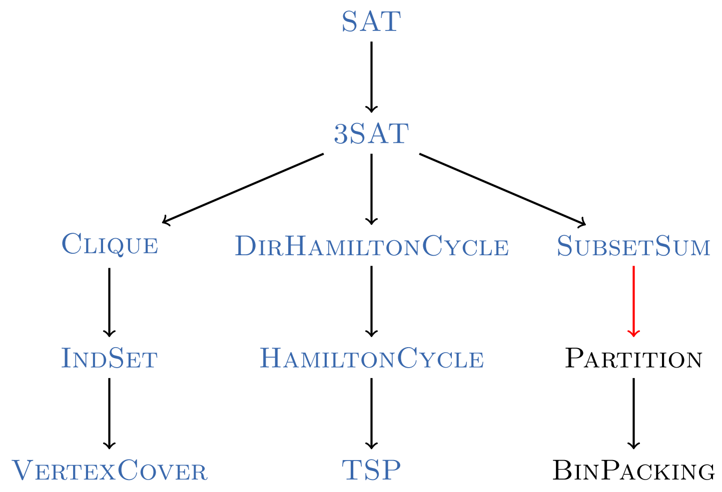
Proof (continued).

Consider a candidate and its corresponding truth assignment.

- Each chosen literal number contributes 1 to the clause digit of each clause satisfied by this literal.
- Satisfying assignments satisfy 1–3 literals in every clause. By using one or both of the padding numbers for each clause digit, all clause digits can be brought to their target value of 4, solving the SubsetSum instance.
- For unsatisfying assignments, there is at least one clause with 0 satisfied literals. It is then not possible to extend the candidate to a SubsetSum solution because the target value of 4 cannot be reached for the corresponding clause digit.

$\square$

SubsetSum $\leq_p$ Partition



Partition is NP-Complete (1)

Definition (Partition)

The problem Partition is defined as follows:

Given: numbers $a_1, \dots, a_k \in \mathbb{N}_0$

Question: Is there a subset $J \subseteq \{1, \dots, k\}$ with $\sum_{i \in J} a_i = \sum_{i \in \{1,\dots,k\} \setminus J} a_i$?

Theorem

Partition is NP-complete.

Partition is NP-Complete (2)

Proof.

Partition $\in$ NP: guess and check.

Partition is NP-hard: We show SubsetSum $\leq_p$ Partition.
We are given a SubsetSum instance with numbers $a_1, \dots, a_k$ and target size b. Let $M := \sum_{i=1}^{k} a_i$.

Construct the Partition instance $a_1, \dots, a_k, M+1, 2b+1$ (can obviously be computed in polynomial time).

Observation: the sum of these numbers is
$M + (M+1) + (2b+1) = 2M + 2b + 2$

$\rightsquigarrow$ A solution partitions the numbers into two subsets, each with sum $M + b + 1$.

...

Partition is NP-Complete (3)

Proof (continued).

Reduction property:

($\Rightarrow$): construct Partition solution from SubsetSum solution

- Let $J \subseteq \{1, \dots, k\}$ be a SubsetSum solution, i. e. $\sum_{i \in J} a_i = b$.
- Then J together with (the index of) $M+1$ is a Partition solution, since $\sum_{i \in J} a_i + (M+1) = b + M + 1 = M + b + 1$ (and thus the remaining numbers also add up to $M + b + 1$).

...

Partition is NP-Complete (4)

Proof (continued).

($\Leftarrow$): construct SubsetSum solution from Partition solution

- One of the two parts of the partition contains the number $M+1$.
- Then the other numbers in this part sum to $(M+b+1) - (M+1) = b$.

$\rightsquigarrow$ These remaining numbers must have indices from $\{1, \dots, k\}$, since $M+1$ is not one of them and $2b+1$ is too large.

$\rightsquigarrow$ These numbers form a SubsetSum solution.

□

$\textsc{Partition} \leq_p \textsc{BinPacking}$

$$
\begin{tikzcd}
 & \textsc{SAT} \arrow[d] & \\
 & \textsc{3SAT} \arrow[dl] \arrow[d] \arrow[dr] & \\
\textsc{Clique} \arrow[d] & \textsc{DirHamiltonCycle} \arrow[d] & \textsc{SubsetSum} \arrow[d] \\
\textsc{IndSet} \arrow[d] & \textsc{HamiltonCycle} \arrow[d] & \textsc{Partition} \arrow[d, red] \\
\textsc{VertexCover} & \textsc{TSP} & \textsc{BinPacking}
\end{tikzcd}
$$

BinPacking is NP-Complete (1)

Definition (BinPacking)

The problem BinPacking is defined as follows:

Given: bin size $b \in \mathbb{N}_0$, number of bins $k \in \mathbb{N}_0$, objects $a_1, \dots, a_n \in \mathbb{N}_0$

Question: Do the objects fit into the bins?
Formally: is there a mapping $f : \{1, \dots, n\} \to \{1, \dots, k\}$ with $\sum_{i \in \{1,\dots,n\} \text{ with } f(i)=j} a_i \leq b$ for all $1 \leq j \leq k$?

Theorem

BinPacking *is NP-complete.*

BinPacking is NP-Complete (2)

Proof.

BinPacking $\in$ NP: guess and check.

BinPacking is NP-hard: We show Partition $\leq_p$ BinPacking.

Given the Partition input $\langle a_1, \dots, a_k \rangle$, we compute $M := \sum_{i=1}^{k} a_i$ and generate a BinPacking input with objects of sizes $a_1, \dots, a_k$ and 2 bins of size $\lfloor \frac{M}{2} \rfloor$.

This can easily be computed in polynomial time, and clearly $a_1, \dots, a_k$ can be partitioned into two groups of the same size iff this bin packing instance is solvable. □

E5.2 Conclusion

. . . and Many More

Further examples of NP-complete problems:

- 3-Coloring: can the vertices of a graph be colored with three colors in such a way that neighboring vertices always have different colors?
- MinesweeperConsistency: Is a given cell in a given Minesweeper configuration safe?
- GeneralizedFreeCell: Is a given generalized FreeCell tableau (i. e., one with potentially more than 52 cards) solvable?
- . . . and many, many more

https://en.wikipedia.org/wiki/List_of_NP-complete_problems

Summary

- In this chapter we showed NP-completeness of three classical packing problems:
  - SubsetSum,
  - Partition, and
  - BinPacking