Theory of Computer Science

G. Röger
Spring Term 2019

University of Basel
Computer Science

Exercise Sheet 11 — Solutions

Exercise 11.1 (Polynomial Reductions, 2.5 + 0.5 marks)

Consider the decision problem 3Coloring:

- *Given:* undirected graph $G = \langle V, E \rangle$
- *Question:* Is there a total function $f : V \to \{r, g, b\}$ such that $f(v) \neq f(w)$ for all $\{v, w\} \in E$?

and the decision problem 3SAT:

- *Given:* a propositional formula φ in conjunctive normal form with *at most* 3 literals per clause
- *Question:* is φ satisfiable?

(a) Show that 3Coloring $\leq_p$ 3SAT.

Solution:

We require a total and polynomial computable function f, which maps an arbitrary 3Coloring problem to a 3SAT problem. We introduce a variable $v_{i,c}$ for each vertex $v_i \in V$ and color $c \in \{r, g, b\}$.

A valid solution for 3Coloring must have the following properties:

(1) No two neighboring vertices share their color:
$(\neg v_{i,r} \vee \neg v_{j,r}) \wedge (\neg v_{i,g} \vee \neg v_{j,g}) \wedge (\neg v_{i,b} \vee \neg v_{j,b})$ for all $\{v_i, v_j\} \in E$

(2) Each vertex has exactly one color:
(a) $v_{i,r} \vee v_{i,g} \vee v_{i,b}$ for all $v_i \in V$
(b) $(\neg v_{i,r} \vee \neg v_{i,g}) \wedge (\neg v_{i,r} \vee \neg v_{i,b}) \wedge (\neg v_{i,g} \vee \neg v_{i,b})$ for all $v_i \in V$

The function is total (the special case for the empty graph is covered by the convention that an empty set of clauses is equivalent to $\top$) and polynomially computable (there are 4 clauses per vertex and 3 clauses per edge).

To show that $x \in$ 3Coloring iff $f(x) \in$ 3SAT:

($\Rightarrow$): Let $x \in$ 3Coloring. Then, we can model the solution of x as interpretation of φ by setting all variables $v_{i,c}$ to true iff c is the color of v_i in the solution, and to false otherwise. This way all clauses from (2) hold. All clauses from (1) hold, as we model a solution to x.

($\Leftarrow$): Analogously.

(b) What can we say about 3Coloring, knowing that 3SAT is NP-complete?

Solution:

We can only conclude that 3Coloring is in NP (it is *no harder* that 3SAT, and it may be simpler).

Exercise 11.2 (NP-completeness, 2+2 marks)

Consider the decision problem HittingSet:

- *Given:* A finite set T, a set of sets $S = \{S_1, \ldots, S_n\}$ with $S_i \subseteq T$ for all $i \in \{1, \ldots, n\}$, a natural number $K \in \mathbb{N}_0$ with $K \leq |T|$.
- *Question:* Is there a set H with at most K elements that contains at least one element from each set in S?

(a) Prove that HittingSet is in NP by specifying a non-deterministic algorithm for HittingSet whose runtime is limited by a polynomial in $n|T|$.

Solution:

The following algorithm solves HittingSet on the input $\langle T, S \rangle$:

```
H = ∅
FOR x ∈ T DO
   GUESS take ∈ {0, 1}
   IF take = 1 THEN
      H := H ∪ {x}
   END
END
IF |H| > K THEN
   REJECT
END
FOR S_i ∈ S DO
   I := S_i ∩ H
   IF I = ∅ THEN
      REJECT
   END
END
ACCEPT
```

The first part can guess every subset of elements $H \subseteq T$. The second part then verifies that the guessed subset is a hitting set. If there is a hitting set of size K then it can be guessed in the first part. The guessed set then passes all tests and will be accepted. If there is no hitting set of size K, every choice of H leads to a REJECT, either because H has more than K elements or because at least one of the sets S_i is not covered.

Every iteration of the first FOR-loop can be done in constant time, so the first FOR-loop requires time $O(|T|)$ in total.

The test $|H| > K$ is possible in constant time.

The computation of I can naively be done in time $O(|S_i| \cdot |H|) = O(|S_i| \cdot |T|) = O(|T|^2)$ (with suitable data structures it is possible to do it faster, but this is not necessary for this exercise). The loop iterates over all $S_i \in S$, i.e., n times.

In total the algorithm runs in time $O(n|T|^2)$, i.e., in polynomial time with respect to the input size.

(b) Prove that HittingSet is NP-complete. You may use without proof that the problem VertexCover (from chapter E5) is NP-complete.

Solution:

To show that HittingSet is NP-complete, we have to show that HittingSet is NP-hard and in NP. For the first part, we reduce VertexCover to HittingSet; we already showed the second part in exercise (a).

Idea: We use the set of nodes V from G as the universe T of the HittingSet instance and the set of edges E as teh set of sets S (each edge is represented as a sets of two nodes in undirected graphs). Every hitting set then uniquely corresponds to a vertex cover of the same size.

Formally:

$$f(\langle\langle V, E\rangle, K\rangle) = \langle V, E, K\rangle$$

The function f is total and computable in polynomial time (other than the restructuring of the data, this is the identity function).

C is a solution for the VertexCover instance if and only if $C \subseteq V$, $|C| \leq K$, and $\{u, v\} \cap C \neq \emptyset$ for all $\{u, v\} \in E$. In exactly these cases, C also is a solution for the HittingSet instance $\langle V, E, K\rangle$.

Exercise 11.3 (NP-hardness, 3 marks)

Consider the following decision problems:

IndSet:

- *Given:* Undirected graph $G = \langle V, E\rangle$, number $k \in \mathbb{N}_0$
- *Question:* Does G contain an independent set of size k or larger, i.e., is there a set $I \subseteq V$ with $|I| \geq k$ and $\{u, v\} \notin E$ for all $u, v \in I$?

SetPacking:

- *Given:* Finite set M, set $\mathcal{S} = \{S_1, \dots, S_n\}$ with $S_i \subseteq M$ for all $i \in \{1, \dots, n\}$, number $k \in \mathbb{N}_0$
- *Question:* Is there a set $\mathcal{S}' \subseteq \mathcal{S}$ with $|\mathcal{S}'| \geq k$, such that all sets in $\mathcal{S}'$ are pairwise disjoint, i.e., for all $S_i, S_j \in \mathcal{S}'$ with $S_i \neq S_j$ it holds that $S_i \cap S_j = \emptyset$?

Prove that SetPacking is NP-hard. You may use that the problem IndSet is NP-complete.

Solution:

Wir müssen zeigen, dass IndSet $\leq_{\mathrm{p}}$ SetPacking.
Hierzu definieren wir $f(\langle V, E\rangle, k) = \langle E \cup V, \mathcal{S}, k\rangle$ mit $\mathcal{S} = \{S_v \mid v \in V\}$, wobei $S_v = \{e \in E \mid v \in e\} \cup \{v\}$. Die Funktion f lässt sich offensichtlich in polynomieller Zeit berechnen.
Wir müssen noch zeigen: $\langle V, E\rangle$ enthält eine unabhängige Menge der Grösse $\geq k$ genau dann, wenn $\mathcal{S}$ mindestens k paarweise disjunkte Mengen enthält:

- Für eine unabhängige Menge $I \subseteq V$ gilt für alle $u, v \in I$, dass $\{u, v\} \notin E$. Betrachte die Menge $\mathcal{S}'_I = \{S_u \mid u \in I\}$. Da jedes $v \in V$ nur genau in der Menge S_v vorkommt, besteht $\mathcal{S}'_I$ aus $|I|$ unterschiedlichen Mengen. Wir zeigen durch Widerspruch, dass diese zudem paarweise verschieden sind:

  Angenommen, es gibt $S_u, S_v \in \mathcal{S}'_I$ mit $S_u \neq S_v$ und es existiert $e \in S_u \cap S_v$. Es gilt $e \in E$ (und damit $|e| = 2$), da jedes $w \in V$ nur in einer Menge vorkommt. Wegen $e \in S_u$ gilt $u \in e$ und wegen $e \in S_v$ gilt $v \in e$. Daraus folgt, dass $\{u, v\} \in E$. $\rightsquigarrow$ Widerspruch zu I unabhängige Menge.

- Sei $\mathcal{S}' \subseteq \mathcal{S}$ eine Menge paarweise disjunkter Mengen. Dann gilt für alle $S_u, S_v \in \mathcal{S}'$ mit $S_u \neq S_v$ (und damit $u \neq v$), dass es kein e gibt mit $u \in e$ und $v \in e$, und somit $\{u, v\} \notin E$. Daher ist $\{v \mid S_v \in \mathcal{S}'\}$ eine unabhängige Menge der Grösse $|\mathcal{S}'|$ in $\langle V, E\rangle$.

Insgesamt hat f also die geforderten Eigenschaften einer polynomiellen Reduktion.