# Formale Methoden der Informatik
# Computability and Complexity: Exercises

$\rightarrow$ The name of the student representing the group $\leftarrow$

WS 2013

**The deadline for submitting your exercises for review is November 11. If you would like to receive the corrections before the exercise session on November 12, you should submit your solutions to TUWEL no later than November 5.**

**Exercise 1** *Consider the following problem:*

| |
|---|
| ***VARIABLE-VALUE (VV)*** |
| *INSTANCE: A tuple* $(\Pi, I, x)$*, where* |
| *a)* $\Pi$ *is a program that takes as input a string,* |
| *b)* $I$ *is a string,* |
| *c)* $x$ *is a global variable in* $\Pi$ *of type* Boolean. |
| *QUESTION: In the run of* $\Pi$ *on* $I$*, does* $x$ *ever get assigned the value* true*?* |

*By providing a reduction from **HALTING** to **VARIABLE-VALUE**, prove that **VARIABLE-VALUE** is undecidable.*

**Solution:** your solution

**Exercise 2** *By providing a semi-decision procedure, prove that **VARIABLE-VALUE** is semi-decidable.*

**Solution:** your solution

**Exercise 3** *Consider the following problem:*

| |
|---|
| ***EXIST-PAIR*** |
| *INSTANCE: A program* $\Pi$ *that takes as input a pair of natural numbers, and returns* true *or* false*. It is guaranteed that* $\Pi$ *terminates on any input.* |
| *QUESTION: Do there exist* $n_1, n_2$ *such that* $\Pi(n_1, n_2) = true$*?* |

*By providing a semi-decision procedure, prove that **EXIST-PAIR** is semi-decidable.*

**Solution:** your solution

**Exercise 4** *Prove that **EXIST-PAIR** is undecidable.*

*Hint: For your proof you may assume the availability of an interpreter for instances of **HALTING**. In particular, you have available a procedure* $\Pi_{int}$ *that does the following:*

*(a)* $\Pi_{int}$ *takes as input a program* $\Pi$*, a string* $I$*, and a natural number* $n$*.*

*(b)* $\Pi_{int}$ *emulates the first* $n$ *steps of the run of* $\Pi$ *on* $I$*. If* $\Pi$ *terminates on* $I$ *within* $n$ *steps, then* $\Pi_{int}$ *returns* true*. Otherwise,* $\Pi_{int}$ *returns* false*.*

**Solution:** your solution

Recall that the complement of **2-SAT** is as follows:

**2-UNSAT**

INSTANCE: Boolean formula $\varphi$ in 2-CNF.

QUESTION: Is $\varphi$ unsatisfiable?

We consider a polynomial-time reduction from **REACHABILITY** to **2-UNSAT**. Let an arbitrary instance of **REACHABILITY** be given by the undirected graph $G = (V, E)$ and two vertices $a, b \in V$. In the reduction we use a propositional variable $R_v$ for each vertex $v \in V$. The resulting instance $\varphi_{G,a,b}$ of **2-UNSAT** is as follows:

$$\varphi_{G,a,b} = R_a \wedge \neg R_b \bigwedge_{[c,d]\in E} (\neg R_c \vee R_d).$$

**Exercise 5** *Prove formally the "$\Rightarrow$" direction of the correctness of the reduction, i.e. prove the following statement: if* $b$ *is reachable from* $a$ *in* $G$*, then* $\varphi_{G,a,b}$ *is unsatisfiable.*

**Solution:** your solution

**Exercise 6** *Prove the "$\Leftarrow$" direction of the correctness of the reduction, i.e. prove the following statement: if* $\varphi_{G,a,b}$ *is unsatisfiable, then* $b$ *is reachable from* $a$ *in* $G$*.*

Hint: the above is equivalent to proving the following statement: if $b$ is not reachable from $a$ in $G$, then $\varphi_{G,a,b}$ is satisfiable.

**Solution:** your solution

Suppose we are in a country that has cities connected by roads. We know that a robber is in city $a$ and will try to run to city $b$. There are $n$ police cars available, and they have to be used to block the roads so that the robber cannot reach the city $b$. Your problem is to assign the available police cars to roads so that *all* possible paths from $a$ to $b$ are blocked.

The above problem can be formalized as follows:

**EDGE-REMOVAL**

INSTANCE: A tuple $(G, a, b, n)$, where

- $G = (V, E)$ is an undirected graph,

- $a, b \in V$, and

- $n$ is an integer.

QUESTION: Is it possible to remove $n$ edges from $G$ so that the resulting graph does not have a path from $a$ to $b$?

**Exercise 7** *Give a proof that* ***EDGE-REMOVAL*** *is in* NP*, i.e. define a certificate relation and briefly discuss that it is polynomially balanced and polynomial-time decidable.*

**Solution:** your solution

**Exercise 8** *Provide a reduction from 3-**COLORABILITY** to 4-**COLORABILITY**, and prove that your reduction is correct.*

Hint: for the reduction it suffices to introduce one additional node to the input graph.

It is known that 3-**COLORABILITY** is NP-hard. Thus, your reduction shows that 4-**COLORABILITY** is NP-hard as well.

**Solution:** your solution

**Exercise 9** *Argue that the following problem is solvable in logarithmic space:*

***PREFIX***

*INSTANCE: A list* $L = \langle s_1, \ldots, s_n \rangle$*, where each* $s_i$ *is either* $0$ *or* $1$.

*QUESTION: Does there exist* $1 \leq j \leq n$ *such that* $\langle s_1, \ldots, s_j \rangle$ *has the same number of* $0s$ *and* $1s$.

**Solution:** your solution

**Exercise 10** *Let* $L = \{w \in \{0,1\}^* \mid w \text{ has the form } 0^*1^*\}$*, i.e.* $L$ *is the set of all strings* $w$ *that can be split into 2 words* $w = w_1 w_2$ *such that (a)* $w_1$ *is a sequence of* $0s$*, and (b)* $w_2$ *is a sequence of* $1s$*. Define a Turing machine* $M$ *that decides* $L$*, i.e. define a tuple* $M = (K, \Sigma, \delta, s)$ *such that, for all* $w \in \{0,1\}^*$*, we have:*

- *if* $w \in L$*, then* $M(w) =$ "*yes*";
- *if* $w \notin L$*, then* $M(w) =$ "*no*".

*Additionally, provide a high-level description of* $M$.

**Solution:** your solution