Formale Methoden der Informatik
Computability and Complexity: Exercises

→ The name of the student representing the group ←

SS 2013

The deadline for submitting your exercises for review is April 14. If you would like to receive the corrections before the exercise session on April 15, you should submit your solutions to TUWEL no later than April 7.

Exercise 1 *By providing a reduction from the* ***HALTING*** *problem to* ***REACHABLE-CODE****, prove that* ***REACHABLE-CODE*** *is undecidable.*

Solution: your solution

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

Solution: your solution

Exercise 3 *By providing a reduction from* ***CORRECTNESS*** *to* ***HALTING****, prove that* ***CORRECTNESS*** *is semi-decidable.*

Solution: your solution

Exercise 4 *Prove that the following problem is undecidable:*

> ***ALL-FALSE***
>
> *INSTANCE: A program* Π *that takes as input a natural number and returns* true *or* false*. It is guaranteed that* Π *terminates on any input.*
>
> *QUESTION: Is it the case that* $\Pi(k) = false$ *for* all *natural numbers* k*?*

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

(a) Π_{int} *takes as input a program* Π*, a string* I*, and a natural number* n*.*

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

Solution: your solution

Suppose you have n processes, where some processes may need to communicate with each other. Suppose you also have m computers, where some of them are connected by a (fast) direct network connection. Each computer has a limit on the number of processes it can run. Your problem is to assign processes to computers so that the limits are obeyed and all the processes that need to communicate can communicate. This can be formalized as follows:

ASSIGNMENT

INSTANCE: A pair $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ of undirected graphs, and a function *limit* that assigns to each $v \in V_2$ an integer. The graph G_2 is reflexive, i.e. for every $v \in V_2$, $[v, v] \in E_2$.

QUESTION: Does there exist an assignment μ that assigns to each vertex in V_1 a vertex in V_2 such that:

(A) if $[v, v'] \in E_1$, then also $[\mu(v), \mu(v')] \in E_2$, and

(B) for every vertex v in V_2, no more than $limit(v)$ nodes of V_1 are assigned to v.

Exercise 5 *Give a proof that **ASSIGNMENT** 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 6 *Define a polynomial-time reduction from **CLIQUE** to **ASSIGNMENT**.*

Note: the result of Exercise 5 together with the reduction show that **ASSIGNMENT** is NP-complete.

Solution: your solution

We consider a polynomial-time reduction from **INDEPENDENT SET** to **SAT**. Let an arbitrary instance of **INDEPENDENT SET** be given by the undirected graph $G = (V, E)$ and integer k. Let V be of the form $V = \{b_1, \dots, b_m\}$. We construct a propositional formula $\varphi_{G,k}$ (i.e. an instance of **SAT**) as follows. First of all, we use the following propositional variables:

- M_{i,b_j} for each $1 \leq i \leq k$ and $1 \leq j \leq m$ (intended meaning: M_{i,b_j} is set to *true* in a model of $\varphi_{G,k}$ if and only if the number i is assigned to the node b_j of G).

Then the formula $\varphi_{G,k}$ is defined as $\varphi_{G,k} = \alpha_1 \wedge \alpha_2 \wedge \alpha_3$, where

$$\alpha_1 = \bigwedge_{1 \leq i \leq k} \Big(\bigvee_{1 \leq j \leq m} M_{i,b_j} \Big)$$

$$\alpha_2 = \bigwedge_{(1 \leq n \leq m) \wedge (1 \leq i,j \leq k) \wedge (i \neq j)} \neg (M_{i,b_n} \wedge M_{j,b_n})$$

$$\alpha_3 = \bigwedge_{[v_1,v_2] \in E} \ \bigwedge_{1 \leq i,j \leq k} \neg (M_{i,v_1} \wedge M_{j,v_2})$$

Informal explanation of the reduction. Intuitively (!), the formulae $\alpha_1, \alpha_2, \alpha_3$ can be explained as follows.

- The formula α_1 expresses the condition that each number $i \in \{1, \dots k,\}$ must be assigned to at least one node from G.
- The formula α_2 makes sure that a pair of numbers do not share the same vertex in G. Thus α_1 together with α_2 make sure that at least k nodes from G have numbers "assigned".
- The formula α_3 ensures that for every edge $[v_1, v_2]$ of G we have not assigned numbers to both v_1 and v_2.

Remark. All the above comments are *explanations* of the intuition of the problem reduction. They are *not proofs*!! When you are requested to prove the correctness of the problem reduction, you are not allowed to refer to these explanations. *Your proofs have to be self-contained*!

Exercise 7 *Prove formally the "$\Rightarrow$" direction of the correctness of the reduction, i.e. prove the following statement: if G has an independent set I of size $\geq k$, then there exists a truth assignment T that makes $\varphi_{G,k}$ evaluate to true.*

Solution: your solution

Exercise 8 *Prove the "$\Leftarrow$" direction of the correctness of the reduction, i.e. prove the following statement: if $\varphi_{G,k}$ is satisfiable, then there exists some independent set I in G of size $\geq k$.*

Solution: your solution

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

SAME-DIGITS
INSTANCE: A pair L_1, L_2 of lists, where each list contains some digits from $0, \ldots, 9$.
QUESTION: Is the set of digits occurring in L_1 equal to the set of digits occurring in L_2?

Solution: your solution

Exercise 10 *Let $L = \{w \in \{1\}^* \mid w \text{ has length } 3k \text{ for some integer } k \geq 0\}$, i.e. L is the set of all strings w such that (a) w is built using the symbol 1, and (b) the length of w is a multiple of 3. Define a Turing machine M that decides L, i.e. define a tuple $M = (K, \Sigma, \delta, s)$ such that, for all $w \in \{1\}^*$, 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