Albert-Ludwigs-Universität, Inst. für Informatik
Prof. Dr. Fabian Kuhn
M. Ahmadi, P. Bamberger, H. Ghodselahi, Y. Maus, P. Schneider

Theoretical Computer Science - Bridging Course
Summer Term 2017
Exercise Sheet 6 – Sample Solution

Exericse 1: Decidability? (3 Points)

Let n be a fixed positive integer, $\Sigma = \{0, 1\}$ a fixed alphabet, M a fixed TM and $w \in \Sigma^*$ a fixed word.

$$L_{\Sigma,M,n,w} := \begin{cases} \{1^n\}, & M \text{ stops on } w \text{ in at most } n \text{ steps} \\ \{0^n\}, & M \text{ stops on } w \text{ after} > n \text{ steps} \\ \emptyset, & M \text{ does not stop on } w. \end{cases}$$

Is $L_{\Sigma,M,n,w}$ decidable?
Remark: For some $a \in \Sigma$, a^n denotes the word which repeats a n times.

Sample Solution

The problem is decidable: The language equals either $\{1^n\}$, $\{0^n\}$ or $\emptyset$. n, M, Σ and w are fixed and all languages only consist of a single word or are empty. We do not know which one equals L but in either case there is a TM which decides it.

Exersive 2: Semi-Decidable vs. Recursively Enumerable (4 Points)

Very often people in computer science use both terms equivalently. The following exercise shows in which way they actually are equivalent. We first recal the definition of both terms.

A language L is *semi-decidable* if there is a Turing machine which accepts every $w \in L$ and does not accept any $w \notin L$ (this means the TM can either reject $w \notin L$ or simply not stop for $w \notin L$.

A language is *recursively enumerable* if there is a Turing machine which eventually outputs every word $w \in L$ and never outputs a word $w \notin L$.

(a) Show that any recursively enumerable language is semi-decidable.

(b) Show that any semi-decidable language is recursively enumberable.

Sample Solution

1. Let M_L be the TM which enumerates L. Construct a TM which, on input w, simulates M_L. If M_L outputs w the TM accepts w, otherwise it might run forever.

2. Let M_L be a TM which semi-decides L. We use a tricky simulation of M_L to construct a TM which recursively enumerates L. We order all words lexicographically $w_1, w_2, w_3, \ldots$ and then we simulate M_L as follows

   (a) Simulate one step of M_L on w_1

(b) Simulate one (further) step of M_L on w_1 and w_2

(c) Simulate one (further) step of M_L on w_1, w_2 and w_3

(d) Simulate one (further) step of M_L on w_1, w_2, w_3 and w_4

(e) etc.

Exercise 3: Halting Problem (3+2+2+1 points)

The *special halting problem* is defined as

$$H = \{\langle M \rangle \mid \langle M \rangle \text{ encodes a TM and } M \text{ halts on } \langle M \rangle\}.$$

(a) Show that H is undecidable.

Hint: Assume that M is a TM which decides H and then construct a TM which halts iff M does not halt. Use this construction to find a contradiction.

(b) Show that the special halting problem is recursively enumerable.

(c) Show that the complement of the special halting problem is not recursively enumerable.

Hint: What can you say about a language L if L and its complement are recursively enumerable? (if you make some observation for this, also prove it)

(d) Let L_1 and L_2 be recursively enumerable languages. Is $L_1 \setminus L_2$ recursively enumerable as well?

Sample Solution

1. Assume that H is decidable. Then there is a TM M which decides it. Now define a TM $\tilde{M}$ which terminates on the inputs on which M does not terminate: The TM $\tilde{M}$ on input w uses M to test whether $w \in H$. If $w \in H$ it enters a non terminating loop, otherwise it terminates. We now apply $\tilde{M}$ on input $\langle \tilde{M} \rangle$ and construct a contradiction.

   $\langle \tilde{M} \rangle \notin H$: Then M rejects $\langle \tilde{M} \rangle$. Thus $\tilde{M}$ terminates on $\langle \tilde{M} \rangle$ by the definition of $\tilde{M}$. Thus $\langle \tilde{M} \rangle \in H$, a contradiction.

   $\langle \tilde{M} \rangle \in H$: Then M accepts $\langle \tilde{M} \rangle$, i.e., $\tilde{M}$ enters a non terminating loop on $\langle \tilde{M} \rangle$ and does not halt on $\langle \tilde{M} \rangle$ which means that $\langle \tilde{M} \rangle \notin H$, a contradiction.

   (actually both cases are similar as in both cases $\tilde{M}$ enters a non terminating loop and we do have the statement

$$\langle \tilde{M} \rangle \in H \Leftrightarrow \langle \tilde{M} \rangle \notin H.$$

2. The special halting problem is semi-decidable because we can construct a TM which semi-decides it as follows: If the input is not a valid coding of a TM the TM rejects it. If the input is the coding of a TM M it simulates M on $\langle M \rangle$ and accepts if this simulation stops.

   With the previous exercise it follows that the halting problem is recursively enumerable.

3. First note that if a language L and its complement are recursively enumerable the language L is a recursive language: Assume that L is recursively enumerable by TM M_1 and its complement by TM M_2. Then we construct a TM which, on input w interchangebly simulates one step of M_1 and one step of M_2. Eventually one of the two TMs will output w. If M_1 outputs w we accept w and if M_2 outputs w we reject w.

   If the complement of the special halting problem was recursively enumerable, then H and its complement would be recursively enumerable. But then H would be a recursive language which is a contradiction.

4. This does not hold in general. Let $L_1 = \{0,1\}^*$ be the language of all words over $\Sigma = \{0,1\}$ and let L_2 be the special halting problem. Then L_1 and L_2 are recursively enumerable (L_1 is even a recursive language) but $L_1 \setminus L_2$ equals the complement of the special halting problem and is not recursively enumerable.

Exercise 4 (2+3 points)

(a) Show that every finite language is a decidable.

(b) Assume that π is a fixed order of the words in Σ^* such that a Turing machine can decide in finite time whether $\pi(w) \leq \pi(w')$ for all $w, w' \in \Sigma^*$. Furthermore assume that for a given language L there is a Turing machine that enumerates the words of L in order $w_1, w_2, w_3, \ldots$ such that $\pi(w_i) \leq \pi(w_j)$ holds for all $i \leq j$.

Show that L is decidable.

Sample Solution

1. If a language L is finite it can be recognized by a DFA. As every DFA can be simulated by a TM there is also a TM which recognizes L.

2. We construct a TM which decides L as follows: If L is a finite language then we can construct a TM as in the first part of the question.

   Thus assume that L is infinite and we want to decide whether $w \in L$. Now, to decide whether $w \in L$ we simulate M_L. Let w' be the current word that M_L has output. Our TM moves to an accepting state if $w' = w$. If $\pi(w') > \pi(w)$ (the TM can test this in finite time) we move to the rejecting state. Otherwise we continue the simulation and wait for the next word.

   The TM does always halt as a w' with $\pi(w') > \pi(w)$ does always exist if the language is infinite.

   Correctness: If we accept w we also have $w \in L$ because M_L had w as output. If we reject w then M_L output a w' with $\pi(w') > \pi(w)$. Because M_L outputs the words in order π this means every word w'' that M_L would output in a further simulation has the property $\pi(w'') > \pi(w') > \pi(w)$, i.e., M_L will never output w, i.e., $w \notin L$.