Automata and Grammars

SS 2018

Assignment 13: Solutions to Selected Problems

Problem 13.1. [Encodings of Turing Machines]

Let $M = (\{q_0, q_1, q_2, q_3\}, \{0, 1\}, \{0, 1, \Box\}, \Box, q_0, q_3)$ be the Turing machine that is given through the following transition function:

δ	0	1	$\Box$	Comments
q_0	$(q_0, 0, R)$	$(q_0, 1, R)$	$(q_2, \Box, L)$	Move right to the last nonblank symbol
q_1	$(q_1, 0, L)$	$(q_1, 1, L)$	$(q_3, \Box, R)$	Move left to the first nonblank symbol
q_2	$(q_1, 1, L)$	$(q_2, 0, L)$	$(q_3, 1, 0)$	Add 1 while moving left
q_3	$-$	$-$	$-$	Final state

that is, M computes the binary +1-function (compare M to the TM in the first example of Section 4.1).

(a) Compute the encoding $c(M)$ of the above TM (see the definition before Lemma 4.9).

(b) Let M' be the TM that is given through the following encoding

$$\begin{aligned} c(M') \quad = \quad & 1110001111101010100100011010010101000110100010010101\,1 \\ & 00101001001011001001001010110010001000100100111. \end{aligned}$$

Reconstruct the TM $M' = (Q, \Sigma, \Gamma, \Box, \delta', q_0, q_n)$ from its encoding $c(M')$.

(c) Construct the encoding $c(\hat{M})$ of the following TM $\hat{M}$ from the encoding $c(M')$ of M' and the input word $x = 10$, where $\hat{M}$ behaves as follows:

(1) erase the given input;
(2) write x;
(3) simulate M' on input x.

Solution. (a) M is already in the form required for the encoding. Hence, we obtain

$$\begin{aligned} c(M) \quad = \quad & 1110^4 11111 \cdot 01010101000 \cdot 11 \cdot 0100101001000 \cdot 11 \cdot 010001000100010 \cdot \\ & 11 \cdot 00101001010 \cdot 11 \cdot 0010010010010 \cdot 11 \cdot 001000100001000 1000 \cdot \\ & 11 \cdot 0001010010010 \cdot 11 \cdot 00010010001010 \cdot 11 \cdot 000100010000100100 \cdot 111. \end{aligned}$$

(b) We obtain the TM $M' = (\{q_0, q_1, q_2\}, \{0, 1\}, \{0, 1, \Box\}, \delta', q_0, q_2)$, where δ' is described by the following table:

δ'	0	1	$\Box$	Comments
q_0	$(q_0, 1, R)$	$(q_0, 0, R)$	$(q_1, 0, L)$	Move right, invert symbols, add suffix 0
q_1	$(q_1, 1, L)$	$(q_1, 0, L)$	$(q_2, 1, 0)$	Move left, invert symbols, add prefix 1
q_2	$-$	$-$	$-$	Final state

(c) We need two new states in order to realize (1) and (2):

	0	1	$\Box$	Comments
q_0	$(q_0, \Box, R)$	$(q_0, \Box, R)$	$(q_1, 0, L)$	Erase input, write 0, move left
q_1	$-$	$-$	$(q_2, 1, 0)$	Write 1, goto M'

Hence, all states q_i of M' have to be replaced by q_{i+2}, which gives the following encoding for the TM $\hat{M}$:

$$\begin{array}{rcl} c(\hat{M}) & = & 1110^5 11111 \cdot 0101010001000 \cdot 11 \cdot 01001010001000 \cdot 11 \cdot 010001001010 \cdot \\ & & 11 \cdot 0010001000100100 \cdot \\ & & 11 \cdot 0001010001001000 \cdot 11 \cdot 0001001000101000 \cdot 11 \cdot 0001000100001010 \cdot 11 \cdot \\ & & 0000101000010010 \cdot 11 \cdot 0000100100001010 \cdot 11 \cdot 00001000100000100100 \cdot 111. \end{array}$$

□

Problem 13.2. [Recursively Enumerable Languages]

Prove that the following languages are recursively enumerable:

$$\begin{array}{lrcl} \text{(a)} & L_1 & = & \{\, w \in \{a,b,c\}^* \mid |w|_a = |w|_b + |w|_c \,\}, \\ \text{(b)} & L_2 & = & \{\, w \in \{a\}^* \mid \exists n \geq 0 : w = a^{2^n} \,\}. \end{array}$$

Solution. We must present Turing machines that halt exactly for the words from the language L_i ($1 \leq i \leq 2$).

(a) The TM M_1 proceeds as follows:

- While scanning its input w from left to right on tape 1, M_1 realizes two unary counters on tapes 2 and 3 for counting the number $|w|_a$ (on tape 2) and the number $|w|_b + |w|_c$ (on tape 3).
- The counters on tape 2 and tape 3 are compared by moving the corresponding heads simultaneously and synchronously across them.
- If $|w|_a = |w|_b + |w|_c$, then M_1 halts and accepts; otherwise, it enters an infinite loop.

Thus, $L(M_1) = L_1$, which proves that L_1 is recursively enumerable.

(b) The TM M_2 proceeds as follows:

- While moving across its tape, M_2 checks that the input is from $\{a\}^+$, and it replaces every second letter a by b.
- Now M_2 moves repeatedly across its tape, in each round replacing every second letter a encountered by b. This continues until after round n all but one symbols a have been replaced, which means that the input was the word a^{2^n}, or until the process gets stuck, as the number of a-symbols is uneven, but larger than one. In the former case M_2 halts and accepts, while in the latter case it enters an infinite loop.

Thus, $L(M_2) = L_2$, which proves that L_2 is recursively enumerable. □

Problem 13.3. [Undecidable Languages]

Prove that the following languages are undecidable:

(a)	H_0	$=$	$\{\, w \in \{0,1\}^* \mid$ The TM M_w halts on empty input $\}$,
(b)	$H_\forall$	$=$	$\{\, w \in \{0,1\}^* \mid$ The TM M_w halts for every input $\}$,
(c)	T_2	$=$	$\{\, u\#v \mid u, v \in \{0,1\}^*$ and the TM M_u halts for all inputs for which the TM M_v halts $\}$.

Solution. (a) Assume that H_0 is decidable, that is, there exists a TM M_0 such that M_0 halts for each input $w \in \{0,1\}^*$, and it yields the output 1 if M_w halts on empty input and the output 0 if M_w does not halt on empty input. For each $w \in \{0,1\}^*$, we now construct a TM M'_w that proceeds as follows:

(1) check that the tape is empty and go into an infinite loop, if not;
(2) write the word w onto the tape;
(3) simulate the TM M_w on input w.

Then M'_w halts on empty input iff M_w halts on input w. Thus, given the encoding $c(M'_w)$ of M'_w as input, M_0 will halt and return 1 iff M_w halts on input w, and it will halt and return 0 iff M_w does not halt on input w. Hence, by applying M_0 to the encoding $c(M'_w)$, we decide whether or not $w \in K$. As K is undecidable by Theorem 4.10, this is a contradiction. Thus, H_0 is undecidable, too.

(b) Assume that $H_\forall$ is decidable, that is, there exists a TM $M_\forall$ such that $M_\forall$ halts for each input $w \in \{0,1\}^*$, and it yields the output 1 if M_w halts for all inputs and the output 0 if M_w does not halt for all inputs. For each $w \in \{0,1\}^*$, we now construct a TM M'_w as follows:

(1) erase the tape;
(2) simulate the TM M_w on empty input.

Then M'_w halts an any input iff M_w halts on empty input. Thus, given the encoding $c(M'_w)$ of M'_w as input, $M_\forall$ will halt and return 1 iff M_w halts on empty input, and it will halt and return 0 iff M_w does not halt on empty input. Hence, by applying $M_\forall$ to the encoding $c(M'_w)$, we decide whether or not $w \in H_0$. As H_0 is undecidable by (a), this is a contradiction. Thus, $H_\forall$ is undecidable, too.

(c) Assume that T_2 is decidable, that is, there exists a TM M_2 such that M_2 halts for each input $u\#v$ $(u, v \in \{0,1\}^*)$, and it yields the output 1 if M_u halts for all inputs for which M_v halts and the output 0 if M_u does not halt for all inputs for which M_v halts. Let M_c be a fixed TM that halts for all inputs. Then $w\#c \in T_2$ iff $w \in H_\forall$. Thus, by applying the TM M_2 to the input $w\#c$ we can decide whether or not $w \in H_\forall$. As $H_\forall$ is undecidable by (b), this is a contradiction. Thus, T_2 is undecidable, too. $\square$

Problem 13.4 [Non-Recursively Enumerable Languages]

Prove that the following languages are not even recursively enumerable:

$$\begin{array}{lcl} \text{(a)} \ L_1 & = & \{\, w \in \{0,1\}^* \mid \text{The TM } M_w \text{ does not halt on empty input}\,\}, \\ \text{(b)} \ L_2 & = & \{\, w \in \{0,1\}^* \mid \text{The TM } M_w \text{ does not halt on any input}\,\}, \\ \text{(c)} \ L_3 & = & \{\, u\#v \mid u, v \in \{0,1\}^* \text{ and the TMs } M_u \text{ and } M_v \text{ halt on the same inputs}\,\}. \end{array}$$

Solution. (a) By Problem 13.3 (a) the set $H_0 = \{0,1\}^* \smallsetminus L_1 = L_1^c$ is not recursive. Hence, at least one of the languages L_1 and L_1^c is not recursively enumerable by Theorem 4.8. Here we show that $H_0 = L_1^c$ is recursively enumerable. Let M_0' be the TM that proceeds as follows, given a word $w \in \{0,1\}^*$ as input:

(1) simulate the TM M_w on empty input.

Then M_0' halts on input w iff the TM M_w halts on empty input. Thus, $L(M_0') = H_0 = L_1^c$, which shows that L_1^c is recursively enumerable. It follows that L_1 is not recursively enumerable.

(b) Assume that L_2 is recursively enumerable, that is, there exists a TM M_2 such that $L(M_2) = L_2$. For each $w \in \{0,1\}^*$, let M_w' be the following TM:

(1) erase the input;
(2) simulate the TM M_w on empty input.

Then M_w' does not halt on any input iff M_w does not halt on empty input. Hence, $c(M_w') \in L_2$ iff $w \in L_1$. As there is a TM M_c that computes $c(M_w')$ from w, we see that $M_2 \circ M_c$ halts on input w iff $w \in L_1$, that is, $L(M_2 \circ M_c) = L_1$. This contradicts (a), showing that L_2 is not recursively enumerable.

(c) Assume that L_3 is recursively enumerable, that is, there exists a TM M_3 such that $L(M_3) = L_3$. Let M' be a fixed TM such that $L(M') = \emptyset$, which means that M' does not halt for any input. For $w \in \{0,1\}^*$, let M_w' be the following TM:

(1) erase the input;
(2) simulate the TM M_w on empty input.

Then

$$L(M_w') = \begin{cases} \{0,1\}^*, & \text{if } M_w \text{ halts on empty input,} \\ \emptyset, & \text{if } M_w \text{ does not halt on empty input.} \end{cases}$$

Hence, $c(M_w')\#c(M') \in L_3$ iff M_w does not halt on empty input iff $w \in L_1$. As $c(M')$ is a fixed constant, and as $c(M_w')$ can be computed from w by a TM M_c, we see that we get a TM M from M_3, M_c, and $c(M')$ such that $w \in L(M)$ iff $w \in L_1$, that is, $L(M) = L_1$. This, however, contradicts (a), implying that L_3 is not recursively enumerable, either. $\square$