Theory of Computer Science

G. Röger
Spring Term 2019

University of Basel
Computer Science

Exercise meeting 7 — Solutions

Exercise 7.1

Specify the Turing machine M_w encoded by:

$$w = 1111001100110011011110100111100110100110011010011001111001101110111001101$$

Is $w \in K$, i.e. does M_w started on w terminate?

Solution:

We first transform w into a word w' over $\Sigma' = \{0, 1, \#\}$, by scanning the word from the front and replacing each two symbols according to $\{11 \mapsto \#, 00 \mapsto 0, 01 \mapsto 1\}$:

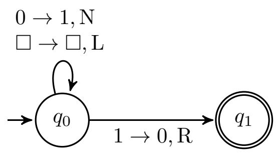
$$w' = \#\#0\#0\#0\#1\#10\#\#0\#10\#0\#10\#0\#\#0\#1\#1\#0\#1$$

The word encodes three transitions:

- ##0#0#0#1#10: $\delta(q_0, 0) = (q_0, 1, N)$
- ##0#10#0#10#0: $\delta(q_0, \Box) = (q_0, \Box, L)$
- ##0#1#1#0#1: $\delta(q_0, 1) = (q_1, 0, R)$

The start state is (per definition) q_0. State q_1 is a terminating state because it does not have any outgoing transitions.

M_w:



On input w TM M_w replaces the first 1 with a 0, moves the head one step to the right and goes into end state q_1. As M_w terminates, we conclude that $w \in K$.

Exercise 7.2

Let A and B be two problems, and $A \leq B$. What can be said about

(a) B, if A is decidable?

(b) B, if A is semi-decidable?

(c) B, if A is undecidable?

(d) B, if A is not semi-decidable?

(e) A, if B is decidable?

(f) A, if B is semi-decidable?

(g) A, if B is undecidable?

(h) A, if B is not semi-decidable?

Solution:

(a) nothing

(b) nothing

(c) B is undecidable

(d) B is not semi-decidable

(e) A is decidable

(f) A is semi-decidable

(g) nothing

(h) nothing

Exercise 7.3

The *equivalence problem* EQUIVALENCE for general (type-0) grammars is defined as:

Given two general grammars G_1 and G_2, is $\mathcal{L}(G_1) = \mathcal{L}(G_2)$?

Show that EQUIVALENCE is undecidable by reducing EMPTINESS to it. The *emptiness problem* EMPTINESS for general (type-0) grammars is defined as:

Given a general grammar G, is $\mathcal{L}(G) = \emptyset$?

(We show that EMPTINESS is undecidable in the next exercise sheet.)

Solution:

Let $G_\emptyset$ be any grammar with $\mathcal{L}(G_\emptyset) = \emptyset$, e.g. a grammar without rules. Let f be the function $f(G) = (G, G_\emptyset)$ for all G.

$$\begin{aligned} G \in \textsc{Emptiness} \text{ iff. } & \mathcal{L}(G) = \emptyset \\ \text{iff. } & \mathcal{L}(G) = \mathcal{L}(G_\emptyset) \\ \text{iff. } & (G, G_\emptyset) \in \textsc{Equivalence} \\ \text{iff. } & f(G) \in \textsc{Equivalence} \end{aligned}$$

The function f is total and computable and reduces EMPTINESS to EQUIVALENCE. Since EMPTINESS is undecidable, EQUIVALENCE must be undecidable as well.