Theory of Computation Excercise for TTF (week 7)

Dimitri Tabatadze · Thursday 11-04-2024

Problem 7.1:

Give a precise mathematical description of the Turing machine T with input symbols $\Sigma = \{\texttt{0}, \texttt{1}\}$ informally given by the following.

If T is given a tape with a finite run of consecutive `1`*s and is started on the leftmost cell of the run, it will erase all the* `1`*s and then stop. If T is started on a tape consisting entirely of* `1`*s, it will never stop.*

Solution

Let $Z = \{z_0, z_e\}$, $E = \{z_e\}$ and δ be defined as

$$\delta(z_0, a) = \begin{cases} (z_0, B, R) & \text{if } a = 1 \\ (z_e, a, N) & \text{otherwise.} \end{cases}$$

■

Problem 7.2:

Construct a two-tape Turing machine that simulates a deterministic push down automaton.

Solution

Let $M = (Z, \Sigma, \Gamma, \delta, z_0, Z_A)$ be a DPDA and $N = (Z_T, \Sigma \cup \Gamma, \delta_T, z_0, Z_A)$ where $Z_T = Z \cup Z \times \Gamma$ the corresponding Turing machine. The states can will be the same, the alphabet will be the union of the both of the alphabets of the DPDA, the transition function will be modified, but the starting and the accepting states will be the same.

The first tape will be the whole input and the second tape will be used as the stack of the DPDA. Now for every $z \in Z, a \in \Sigma, s \in \Gamma$ define

$$\delta_T(z, a, s) = \begin{cases} ((z', s'), a, s, R, R) & \text{if } (z', \text{push } s') = \delta(z, a, s) \\ (z', a, B, R, L) & \text{if } (z', \text{pop}) = \delta(z, a, s) \\ (z', a, s, R, N) & \text{if } (z', \text{push } \varepsilon) = \delta(z, a, s) \end{cases}$$

and

$$\delta_T((z, s), a_1, a_2) = (z, a_1, s, N, N).$$

■

Problem 7.3:

What follows are the exercises from the lecture on a Turing machine. See the mentioned lecture for precise definitions.

a) Describe Turing machine "next state" relation $\vdash$ for $k = 2$ tapes. You can aslo try for $k > 2$.

b) Construct a Turing machine which shifts inscription $w \in \{0, 1, \#\}^*$ of tape one cell to the left. Is your machine regular?

Solution

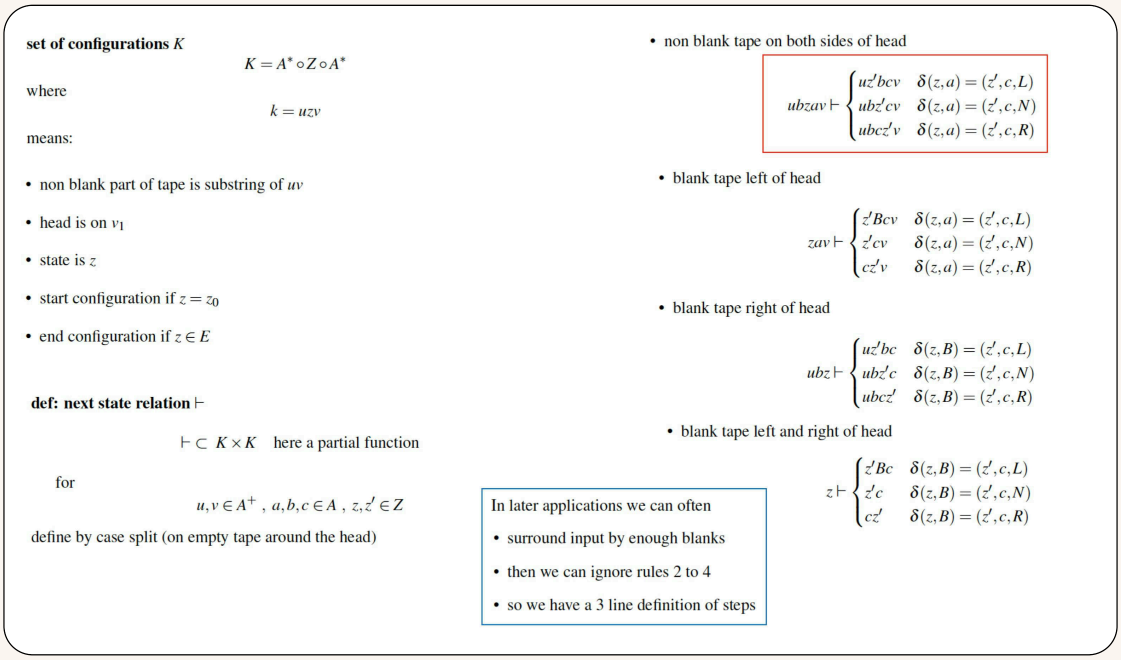
a) 

Figure 1: slide 15 of 08-turing-machines

We can take the set of configurations to be

$$K_2 = (A^* \circ Z \circ A^*)^2,$$

i.e., tuples of configurations where

$$k = \begin{pmatrix} u_1 b_1 z a_1 v_1 \\ u_2 b_2 z a_2 v_2 \end{pmatrix}$$

means

- non-blank part of the tapes 1 and 2 are substrings of $u_1 b_1 a_1 v_1$ and $u_2 b_2 a_2 v_2$ respectively.
- heads of the tapes are on a_1 and a_2, respectively.
- the state for both tapes is z.

Now the next state relation $\vdash$ would be defined as such:

$$\begin{pmatrix} u_1 b_1 z a_1 v_1 \\ u_2 b_2 z a_2 v_2 \end{pmatrix} \vdash \left(\begin{array}{ll} \begin{cases} u_1 z' b_1 c_1 v_1 & \delta(z,a_1,a_2)=(z',c_1,a_2,L,m_2) \\ u_1 b_1 z' c_1 v_1 & \delta(z,a_1,a_2)=(z',c_1,a_2,N,m_2) \\ u_1 b_1 c_1 z' v_1 & \delta(z,a_1,a_2)=(z',c_1,a_2,R,m_2) \end{cases} \\ \begin{cases} u_2 z' b_2 c_2 v_2 & \delta(z,a_1,a_2)=(z',c_1,c_2,m_1,L) \\ u_2 b_2 z' c_2 v_2 & \delta(z,a_1,a_2)=(z',c_1,c_2,m_1,N) \\ u_2 b_2 c_2 z' v_2 & \delta(z,a_1,a_2)=(z',c_1,c_2,m_1,R) \end{cases} \end{array} \right).$$

b) • Let $A = \{0, 1, \#\}$. Now define

$$M = (A \cup \{z_0, z_1, z_e\}, A, \delta, z_0, \{z_e\})$$

and the transition function

$$\begin{aligned}
\delta(z_0, a) &= (z_0, a, R) \\
\delta(z_0, B) &= (z_1, B, L) \\
\delta(z_1, a) &= (a, B, L) \\
\delta(z, a) &= (a, z, L) \qquad \text{where } z \in A \\
\delta(z, B) &= (z_e, z, L) \qquad \text{where } z \in A \cup \{z_1\}.
\end{aligned}$$

- This machine *is* regular.

■

Problem 7.4:

Turing machine M to have accepting states Z_A and rejecting states Z_R. M is acceptor for L if for all w,

$$w \in L \Longleftrightarrow M \text{ started with } w \text{ halts in an accepting state.}$$

So rejection is now possible by halting in a rejecting state or by not halting.

Show: a language has an acceptor in the sense of the new definition if and only if it has an acceptor in the sense of the definition of the lecture on non-computable functions.

Solution

⊠