Teoria da Computação

MIEI 2018/2019 - FCT UNL

Aula Prática 6

Deterministic Finite Automata (DFA)

1. Consider the following alphabet

$$\Sigma = \{ \begin{array}{l} \texttt{insertcard}, \\ \texttt{pin}, \\ \texttt{checkbalance}, \\ \texttt{widthdraw}, \\ \texttt{moreops}, \\ \texttt{retrievecard} \end{array} \}$$

   (a) Specify (in)formally a DFA over the alphabet that checks if a word represents a valid interaction of a user with an ATM (considering that the user inserts and retrieves the card only once).

   (b) How could one change the previous DFA to represent the (non-terminating) behaviour of an ATM?

   (c) Check informally if the word `insertcard checkbalance` is accepted by the DFA.

2. Consider the alphabet

$$\mathit{DIGITS} \stackrel{\text{def}}{=} \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

   (a) Specify a DFA over the alphabet *DIGITS* that checks if a word over *DIGITS* is a possible pin for a mobile phone (with 4 digits). Define the transition function in comprehension.

   (b) Check formally if the words 2345, 123, and 12345 are accepted by the DFA.

3. Consider the alphabet $\Sigma = \{\texttt{X}\}$.

   (a) Specify a DFA over the alphabet that checks if a sequence of `X`s is of even length, or in other words, that only accepts the words over Σ of even length.

   (b) Check informally if the following words are accepted by the DFA.

      i. `XXXX`

      ii. `XXX`

      iii. `XY`

   (c) Specify a DFA over the alphabet that checks if a sequence of `X`s is of odd length, or in other words, that only accepts the words over Σ of odd length.

4. Consider the alphabet

$$AB \stackrel{\text{def}}{=} \{\mathtt{a}, \mathtt{b}\}$$

   (a) Specify a DFA over the alphabet AB that only accepts the words over AB that contain an odd number of `b`s and an even number of `a`s.

   (b) Check informally if the following words are accepted by the DFA.

      i. `babab`

      ii. `abab`

      iii. `ababc`

5. Consider the alphabet

$$DNA \stackrel{\text{def}}{=} \{\mathtt{A}, \mathtt{T}, \mathtt{C}, \mathtt{G}\}$$

   (a) Specify a DFA over the alphabet DNA that only accepts the words over DNA that contain at least one occurrence of `ACT` as a substring.

   (b) Check formally if the following words are accepted by the DFA.

      i. `AACT`

      ii. `ACCT`

      iii. `ACTT`