| | |
|---|---|
| **Intro to Theory of Computing** | **481 Homework 6** |
| **CS481 Fall 2005** | **October 7** |

1. Consider the following automaton. "$\rightarrow$" indicates start state and "$F$" indicates final state.

| | | $a$ | $b$ |
|---|---|---|---|
| $\rightarrow$ | 1 | 8 | 6 |
| | 2 | 7 | 4 |
| | $3F$ | 3 | 8 |
| | $4F$ | 5 | 7 |
| | 5 | 1 | 3 |
| | $6F$ | 6 | 1 |
| | $7F$ | 2 | 4 |
| | 8 | 1 | 3 |

   (a) Say which states are accessible.

   (b) List the equivalence classes of the equivalence relation $\approx$ defined by

$$p \approx q \overset{\text{def}}{\iff} \forall x \in \Sigma^* \; (\widehat{\delta}(p, x) \in F \Leftrightarrow \widehat{\delta}(q, x) \in F).$$

   Include inaccessible states.

   (c) Give the equivalent automaton obtained by collapsing equivalent states and removing inaccessible states.

2. Define a set $U$ to be *ultimately periodic* if there exists integers $m, p$ such that for all $n \geq m$, $n \in U \rightarrow n + p \in U$.

   (a) Prove that every regular language over a unary alphabet is ultimately periodic.

   (b) Prove that if $A$ be a regular language over a finite alphabet then the following set is *ultimately periodic*

$$\mathbf{Length}(A) = \{|x| \mid x \in A\}.$$

3. Let $R \subseteq \Sigma^*$ be a language. Define the following equivalence relation $\equiv_R$ in $\Sigma^*$.

$$x \equiv_R y \overset{def}{\iff} \forall w \in \Sigma^*, \; xw \in R \Leftrightarrow yw \in R.$$

   Show that this is an equivalence relation. Prove that $R$ is a regular language iff the number of equivalence classes of $\equiv_R$ is finite.

4. Till now, the automata that we have considered have been allowed to make only one pass through the input string. What if we allow the automata to read the input string more than once ? Does this increase the power of the automata? In this problem, we are going to prove that it does not. The notation is a bit messy, but is necessary to handle the cases when the machine head either goes into an infinite loop or reaches the end of tape.

   Define a two-way finite automata as an octuple $M = (Q, \Sigma, \vdash, \dashv, \delta, s, t, r)$. Here $Q$ is the finite set of states, $\Sigma$ is the input alphabet. $\{\vdash, \dashv\}$ are letters not in $\Sigma$ and are the left and right endmarkers respectively. $s$ is the start state, $t$ the accept state, and $r$, $r \neq t$ is a reject state. $\delta$, a map from $Q \times (\Sigma \cup \{\vdash, \dashv\})$ to $Q \times \{L, R\}$ is the transition function. $L, R$ stands for left and right. The idea is that the transition function takes a state and a symbol as arguments, and specifies a new state for the finite automata to go into, and a direction for the read head to move. We consider a transition function to be a valid only if it satisfies the following conditions for all states $q$. These are conditions to ensure that the head does not runoff the tape.

$$\exists u \in Q,\ \delta(q, \vdash) = (u, R)$$
$$\exists v \in Q,\ \delta(q, \dashv) = (v, L)$$

   Also, for all symbols $b \in \Sigma \cup \{\vdash\}$,

$$\delta(t, b) = (t, R),\ \ \delta(r, b) = (r, R),$$
$$\delta(t, \dashv) = (t, L),\ \ \delta(r, \dashv) = (r, L).$$

   (a) Construct a finite table (or equivalently, a map) $T_x$ for each string $x$, such that the following is true. Make sure that your state clearly what the intuition behind the definition is.

      - $T_x : Q \cup \{\bullet\} \rightarrow Q \cup \{\perp\}$. Here $\perp$ is a special symbol that we will use to handle the cases when the machine goes into an infinite loop. For instance, if the machine starts at the left end of $x$ and then goes into an infinite loop on string $x$, we will put $T_x(\bullet) = \perp$.
      - For each pair of strings $x$ and $y$, if $T_x = T_y$, then for all strings $w$, $xw$ belongs to the language **iff** $yw$ belongs there too.

   (b) Use the tables $\{T_x \mid x \in \Sigma^*\}$ to show that the language accepted is regular. You can do this either by actually constructing a DFA, or by using the result of problem 3 above.