Automata Theory - Final (Solutions)

L. Kovalchick
LCSEE,
West Virginia University,
Morgantown, WV
{lynn@csee.wvu.edu}

1 Problems

1. Design a DFA to accept the language L, that consists of all strings having even length *and not* ending in 1. (5 points)

   Solution: □

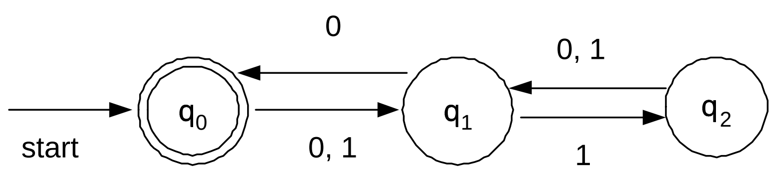

Figure 1: DFA

2. Assume that you are given two DFAs A_1 and A_2; let the languages accepted by these DFAs be $L(A_1)$ and $L(A_2)$ respectively. Design a strategy to check whether $L(A_1) \subseteq L(A_2)$. (5 points)

   Proof: Observe that we want to test whether there exists at least one language (call this language K) such that $K \in L(A_1)$ and $K \notin L(A_2)$. This question can be restated as follows: Is M empty, where $M = L(A_1) \bigcap L(A_2)'$. In order to test M for emptiness, we will first switch the accepting and non-accepting states of A_2 in order to form A_2'. We will then convert A_1 to regular expression R_1 and A_2' to regular expression R_2. Next, we will combine these two regular expressions to form $R = R_1 \cdot R_2$. We can then convert R to an ϵ-NFA, and then to a DFA (call it K). Finally, we will use the graph reachability algorithm to determine if there are any accepting states reachable from the start state; if so, we return "no" (i.e., $L(A_1) \nsubseteq L(A_2)$), otherwise, we return "yes" (i.e., $L(A_1) \subseteq L(A_2)$). □

3. Consider the CFG, G, defined by the following productions:

$$S \rightarrow 0S1S \mid 1S0S \mid \epsilon$$

   Show that $L(G)$ is the set of all strings with an equal number of 0s and 1s. (5 points)
   Hint: Use Induction.

   Proof: Using mathematical induction on the number of steps used to construct the string (say w), we have the following.
   Base Case:
   Consider the string that is derived in one step; the production $S \rightarrow \epsilon$ derives this string, which has 0 0's and 0 1's.
   Let us assume that using a number of steps which is less than or equal to n we produce strings with an

equal number of 0's and 1's (call this string w).
Inductive Step:
Let us assume that the number of steps used to derive the string (call it w') is $n+1$. Clearly, one of the three production rules must have been used for the $n+1$th step. Thus, w' must have been constructed using one of the following:

(a) $S \rightarrow 0w_1 1w_2$. Using the inductive hypothesis, we know that both w_1 and w_2 must have an equal number of 0's and 1's (they were derived in $\leq n$ steps). Using this production rule, we are simply adding an equal number of 0's and 1's to the string; thus the string still contains an equal number of 0's and 1's, even after the $n+1$th production rule.

(b) $S \rightarrow 1w_1 0w_2$. This step is identical to the above step except the position of the 0 and 1 is reversed. Clearly this string contains an equal number of 0's and 1's after the $n+1$th production rule.

(c) $S \rightarrow \epsilon$. Once again, using the inductive hypothesis, we know that w_1 and w_2 have an equal number of 0's and 1's. Thus, simply adding an epsilon does not change the number of 0's and 1's of this string. Therefore, this string contains an equal number of 0's and 1's after the $n+1$th production rule.

Observe that using any one of the three productions for the $n+1$th step produce a string with an equal number of 0's and 1's.
Therefore, $L(G)$ is the set of *all* strings with an equal number of $0's$ and $1's$. □

4. Consider the CFG G_1 defined by:

$$S \rightarrow 0S \mid 0S1S \mid \epsilon$$

Show that

(a) G_1 is ambiguous. (2 points) *Hint: How many leftmost derivations does $w = 001$ have?*
Solution: Observe that $w = 001$ has 2 leftmost derivations.

$$0S1S \rightarrow 00S1S \rightarrow 001S \rightarrow 001$$

and

$$0S \rightarrow 00S1S \rightarrow 001S \rightarrow 001$$

Thus, G_1 is ambiguous. □

(b) Is $L(G_1) = \{0^n 1^n \mid n \geq 0\}$? (3 points)

Solution: No, consider the following leftmost derivation.

$$0S \rightarrow 00S \rightarrow 00$$

This is clearly a valid derivation; thus $L(G_1) \neq \{0^n 1^n \mid n \geq 0\}$. □

5. Design a PDA to accept the language of palindromes L, i.e., the language L consists of all strings w, such that $w = w^R$. (5 points)

Solution: Let $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$, where:
$Q = \{q_0, q_1, q_2\}$
$\Sigma = \{0, 1\}$
$\Gamma = \{0, 1, Z_0\}$
$F = q_2$, and

$\delta =$

$$
\begin{aligned}
\delta(q_0, 1, Z_0) &= \{(q_0, 1Z_0), (q_2, Z_0)\} \\
\delta(q_0, 0, Z_0) &= \{(q_0, 0Z_0), (q_2, Z_0)\} \\
\delta(q_0, \epsilon, Z_0) &= \{(q_2, Z_0)\} \\
\delta(q_0, 1, 0) &= \{(q_0, 10)\} \\
\delta(q_0, 0, 1) &= \{(q_0, 01)\} \\
\delta(q_0, 0, 0) &= \{(q_0, 00)\} \\
\delta(q_0, 1, 1) &= \{(q_0, 11)\} \\
\delta(q_0, \epsilon, 0) &= \{(q_1, 0), (q_1, \epsilon)\}
\end{aligned}
$$

□

6. Design a Turing Machine to accept the regular language described by the expression $1 \cdot 0^* + 0 \cdot 1^*$. (5 points)

Solution: $M =< Q, \Sigma, \Gamma, \delta, q_0, B, q_{accept}, q_{reject} >$, where:
$Q = \{q_0, q_1, q_2, q_{accept}, q_{reject}\}$,
$\Sigma = \{0, 1\}$,
$\Gamma = \{0, 1, B\}$,
$B =$ the blank symbol, and
$\delta =$
□

state	0	1	B
q_0	$< q_1, 0, R >$	$< q_2, 1, R >$	$< q_{reject}, -, - >$
q_1	$< q_{reject}, -, - >$	$< q_1, 1, R >$	$< q_{accept}, -, - >$
q_2	$< q_2, 0, R >$	$< q_{reject}, -, - >$	$< q_{accept}, -, - >$

2 Abbreviations

1. CFG - Context Free Grammar
2. PDA - Pushdown Automaton