2G1505 Automata Theory

SOLUTIONS TO SELECTED PROBLEMS OF EXAM:
10 December 2004

Dilian Gurov
KTH-IMIT

---

1. Convert the nondeterministic automaton given below to an equivalent deterministic one using the subset construction. Omit inaccessible states. Draw the graph of the resulting DFA.

	a	b
$\rightarrow$ q_1	$\{q_2\}$	$\emptyset$
$\rightarrow$ q_2 F	$\emptyset$	$\{q_1, q_3\}$
q_3 F	$\{q_2, q_3\}$	$\{q_1\}$

Solution: (presented as a table)

	a	b
$\rightarrow \{q_1, q_2\}$ F	$\{q_2\}$	$\{q_1, q_3\}$
$\{q_1, q_3\}$ F	$\{q_2, q_3\}$	$\{q_1\}$
$\{q_2, q_3\}$ F	$\{q_2, q_3\}$	$\{q_1, q_3\}$
$\{q_1\}$	$\{q_2\}$	$\emptyset$
$\{q_2\}$ F	$\emptyset$	$\{q_1, q_3\}$
$\emptyset$	$\emptyset$	$\emptyset$

---

2. Consider the following unary operation on languages:

$$min(L) = \{x \in L \mid \text{no proper prefix of } x \text{ is in } L\}$$

Prove that regular languages are closed under this operation; that is, prove that if language A is regular, then so is $min(A)$.

Solution: Assume A is regular. (For simplicity, we shall also assume $\epsilon \notin A$.) Then there is a DFA M_A accepting A. We construct another DFA M'_A from M_A by adding two new states q_F and q_G, making q_F the only accepting state of M'_A, letting $\delta'(q_F, a) = q_G$ and $\delta'(q_G, a) = q_G$ for every $a \in \Sigma$, and by re-directing all edges pointing to a final state in M_A to point to q_F. We can then show $L(M'_A) = min(A)$ as follows:

$$\begin{array}{rcll}
x \in L(M'_A) & \Leftrightarrow & \hat{\delta}'(s, x) = q_F & \{\text{Def. acceptance and } M'_A\} \\
& \Leftrightarrow & \hat{\delta}(s, x) \in F \text{ and for no proper prefix } y \text{ of } x,\ \hat{\delta}(s, y) \in F & \{\text{Def. } M'_A\} \\
& \Leftrightarrow & x \in A \text{ and no proper prefix of } x \text{ is in } A & \{\text{Def. acceptance}\} \\
& \Leftrightarrow & x \in min(A) & \{\text{Def. } min(A)\}
\end{array}$$

thus proving that $min(A)$ is regular.

---

3. Use the constructions we defined on nondeterministic finite automata to inductively build an NFA for the regular expression $(a + ba)(b + ab)^*$. Show all intermediate results.

---

4. For the deterministic automaton given below, apply the minimization algorithm of Lecture 14 to compute the equvalence classes of the collapsing relation $\approx$ defined in Lecture 13. Show clearly the computation steps. List the equivalence classes, and apply the quotient construction to derive a minimized automaton. Draw its graph.

	a	b
$\rightarrow$ q_1	q_3	q_8
q_2 F	q_3	q_1
q_3	q_8	q_2
q_4 F	q_5	q_6
q_5	q_6	q_2
q_6	q_7	q_8
q_7	q_6	q_4
q_8	q_5	q_8

Solution: With the minimization algorithm we establish that $q_1 \approx q_6 \approx q_8$, $q_2 \approx q_4$ and $q_3 \approx q_5 \approx q_7$. The resulting quotient automaton, presented as a table, is:

	a	b
$\rightarrow \{q_1, q_6, q_8\}$	$\{q_3, q_5, q_7\}$	$\{q_1, q_6, q_8\}$
$\{q_3, q_5, q_7\}$	$\{q_1, q_6, q_8\}$	$\{q_2, q_4\}$
$\{q_2, q_4\}$ F	$\{q_3, q_5, q_7\}$	$\{q_1, q_6, q_8\}$

5. Apply the Pumping Lemma – in contra–positive form, as a game with the Demon – to show that the following language:

$$A = \{a^n \mid n \text{ is a power of } 2\}$$

is not regular.

Solution: One possible solution is:

(D) Demon picks k.

(W) We pick $x = \epsilon$, $y = a^{2^k}$ and $z = a^{2^k}$. Then $xyz = a^{2^{k+1}}$ and $|y| > k$.

(D) Demon picks u, v and w so that $uvw = y = a^{2^k}$ and $v \neq \epsilon$.

(W) We pick $i = 2$.

Then $xuv^iwz = a^{2^{k+1}+l}$ for some $0 < l \leq 2^k$. Since $l < 2^{k+1}$ we have $2^{k+1} < 2^{k+1} + l < 2^{k+2}$, and hence $xuv^iwz \notin A$. We have a winning strategy, and A is therefore not regular.

6. Consider the language:

$$A = \left\{ a^k b^l a^m \mid m = k + l \right\}$$

   (a) Use the closure properties of regular languages to show that A is not regular.

   Solution: The language $L(b^*a^*)$ is regular, but $A \cap L(b^*a^*) = \{b^n a^n \mid n \geq 0\}$, as we already know, is not regular. Since regular languages are closed under intersection, A is not regular.

   (b) Give a context–free grammar G generating A.

   Solution: One possibility is:

$$\begin{array}{rcl} S & \to & aSa \mid B \\ B & \to & \epsilon \mid bBa \end{array}$$

   (c) Prove your grammar correct; that is, prove $S \xrightarrow{+}_G x \Leftrightarrow x \in A$.

   Solution: The proof is standard, and is made easy by the fact that we already know that $B \xrightarrow{+}_G x \Leftrightarrow x \in \{b^n a^n \mid n \geq 0\}$.

   (d) Construct an NPDA accepting $A - \{\epsilon\}$ on empty stack. Explain your choice of productions.

   Solution: One possibility is to build an NPDA with three states, having the following productions:

$$\begin{array}{lll} \langle q_0, \perp \rangle \overset{a}{\hookrightarrow} \langle q_0, C \rangle & \langle q_1, C \rangle \overset{a}{\hookrightarrow} \langle q_2, \varepsilon \rangle & \langle q_2, C \rangle \overset{a}{\hookrightarrow} \langle q_2, \varepsilon \rangle \\ \langle q_0, \perp \rangle \overset{a}{\hookrightarrow} \langle q_2, C \rangle & \langle q_1, C \rangle \overset{b}{\hookrightarrow} \langle q_1, CC \rangle & \\ \langle q_0, \perp \rangle \overset{b}{\hookrightarrow} \langle q_1, C \rangle & & \\ \langle q_0, C \rangle \overset{a}{\hookrightarrow} \langle q_0, CC \rangle & & \\ \langle q_0, C \rangle \overset{a}{\hookrightarrow} \langle q_2, CC \rangle & & \\ \langle q_0, C \rangle \overset{b}{\hookrightarrow} \langle q_1, CC \rangle & & \end{array}$$

   The first state counts the initial a's, the second state counts the b's which follow, and the third state checks for the sum. In addition, the first state can nondeterministically decide that no b's are going to come and that exactly half of the a's have been read.

---

7. Give a detailed description of a Turing machine with input alphabet $\{a, \sharp\}$ that on input $a^m \sharp a^n$ halts with $a^{(m \bmod n)}$ written on its tape. Explain the underlying algorithm.

Solution: Here is one possible algorithm, consisting of three phases. It implements modulo division by repeated subtraction.

Preparatory phase: scan right to first blank symbol and replace it with $\dashv$.

Main phase: repeat in rounds, in each round performing:
repeatedly scan from right to left, matching the rightmost a on the right of $\sharp$ with the rightmost a on the left of $\sharp$. The matching is done by replacing the corresponding a's with $\dot{a}$'s.
A round terminates in one of two possible ways:
(a) if there are no more a's on the right of $\sharp$, then delete (that is, replace with the blank symbol) all $\dot{a}$'s on the left of $\sharp$, and restore all $\dot{a}$'s on the right of $\sharp$ to a's. Start new round.
(b) if there is no matching a on the left of $\sharp$, then go to the next phase.

Finalizing phase:
- replace all $\dot{a}$'s by a's on the left of $\sharp$, and
- delete all other symbols on the tape.

---

8. Show that the problem of whether a Turing machine, when started on a blank tape, ever writes a given symbol (say a) of its input alphabet on its tape is not decidable.

   Hint: You could reduce the undecidable problem of acceptance of the null string (problem (f), page 235) to the problem above.

   Solution: Assume the problem was decidable. Then there must be a total Turing machine M_a deciding it. We shall use M_a to build a new total Turing machine M_ϵ deciding the problem of acceptance of ϵ. (Since the latter is known to be undecidable, we shall conclude that the present problem is also undecidable.)

   We construct M_ϵ which, on input $\hat{M}$, converts $\hat{M}$ to $\hat{M}'$ such that M' is like M but:

   - a is renamed to a new letter which is added to the alphabet of $\hat{M}'$,
   - a new state q is added, which becomes the accepting state of $\hat{M}'$, and
   - transitions $\delta(t, b) = (q, a, R)$ are added for every $b \in \Gamma$.

   Then rewind and run as M_a on $\hat{M}'$.

   We can now deduce:

$$
\begin{array}{rcll}
M_\epsilon \text{ accepts } \hat{M} & \Leftrightarrow & M_a \text{ accepts } \hat{M}' & \left\{\text{Def. } M_\epsilon \text{ and } \hat{M}\right\} \\
 & \Leftrightarrow & M' \text{ reaches } t \text{ starting from } \epsilon & \left\{\text{Def. } M_a \text{ and } \hat{M}'\right\} \\
 & \Leftrightarrow & M \text{ accepts } \epsilon & \left\{\text{Def. } \hat{M}'\right\}
\end{array}
$$

   So, M_ϵ decides the problem of acceptance of ϵ. Since the latter is known to be undecidable, we conclude that the present problem is also undecidable.