2G1505 Automata Theory

EXAMINATION PROBLEMS
WITH SELECTED SOLUTIONS
19 May 2006, 14^{oo} - 19^{oo}

Dilian Gurov
KTH CSC
08-790 81 98

Give solutions in English or Swedish, each problem beginning on a new sheet. Write your name on all sheets. The maximal number of points is given for each problem. The course book and own notes as well as reference material are admissible at the exam.

---

1. Convert the nondeterministic automaton given below to an equivalent deterministic one using the subset construction, textbook Lecture 6. Omit inaccessible states. Draw the graph of the resulting DFA. 2p

	a	b
$\rightarrow$ q_0	$\{q_1\}$	$\{q_2\}$
q_1	$\{q_0, q_1\}$	$\{q_0\}$
q_2 F	$\emptyset$	$\{q_1, q_2\}$

Solution: Easy: has 8 states, and hence no inaccessible ones!

---

2. For the deterministic automaton given below, apply the minimization algorithm of Lecture 14 of the textbook to compute the equvalence classes of the collapsing relation $\approx$ defined in Lecture 13. 4p

	a	b
$\rightarrow$ q_0 F	q_1	q_3
q_1	q_2	q_3
q_2 F	q_5	q_2
q_3	q_4	q_1
q_4 F	q_5	q_4
q_5	q_5	q_5

(a) Show clearly the computation steps (use tables).

(b) List the computed equivalence classes.

(c) Apply the quotient construction of Lecture 13 to derive the minimized automaton. Draw its graph.

Solution: As a table:

	a	b
$\rightarrow$ $\{q_0\}$ F	$\{q_1, q_3\}$	$\{q_1, q_3\}$
$\{q_1, q_3\}$	$\{q_2, q_4\}$	$\{q_1, q_3\}$
$\{q_2, q_4\}$ F	$\{q_5\}$	$\{q_2, q_4\}$
$\{q_5\}$	$\{q_5\}$	$\{q_5\}$

---

3. Recall the Myhill–Nerode Theorem, textbook Lecture 16, with the equivalence relation $\equiv_A$ on strings defined in equation (16.1). The latter gives rise to the following technique for proving that the minimal DFA for a given regular language A over Σ has *at least* k states: $\boxed{4\text{p}}$

   Identify k strings $x_1, \ldots, x_k$ over Σ, for which you can show that:
   $x_i \not\equiv_A x_j$ whenever $i \neq j$.

   Then $\equiv_A$ has at least k equivalence classes, and hence the desired result.

   Now, let m be an arbitrary but fixed even number. Consider the language B over $\{a, b\}$ consisting of all strings of length at least m which have an equal number of a's and b's in the last m positions. Use the technique described above to show that the minimal DFA for language B has at least $2^{\frac{m}{2}}$ states.
   Hint: Consider the strings of length $\frac{m}{2}$.

   Solution: Consider the set X of strings of length $\frac{m}{2}$ over $\{a, b\}$. Let $x, y \in X$ so that $x \neq y$. Let x', y' be the shortest suffices of x and y respectively, such that $|x'| = |y'|$ and $x' \neq y'$. Then obviously $\sharp a(x') \neq \sharp a(y')$. Now let $z = a^r b^s$, where $r = \frac{m}{2} - \sharp a(x')$ and $s = m - (|x'| + r)$. Then $x \cdot z \in B$ while $y \cdot z \notin B$, and hence $x' \not\equiv_B y'$. Since X has $2^{\frac{m}{2}}$ elements, $\equiv_B$ has at least $2^{\frac{m}{2}}$ equivalence classes, and therefore the minimal DFA for language B has at least $2^{\frac{m}{2}}$ states.

---

4. A context–free grammar $G = (N, \Sigma, P, S)$ is called *strongly right–linear* (or SRLG for short) if all its productions are of shape $A \to aB$ or $A \to \epsilon$. Prove that SRLGs generate precisely the regular languages. $\boxed{6\text{p}}$
   Hint: Define appropriate transformations between SRLGs and Finite Automata – one for each direction! – and prove that these transformations are language preserving. State and prove appropriate lemmas where needed to structure the proofs.

   Solution: In two parts: the first part shows that the languages generated by SRLGs are regular, while the second shows that every regular language is the language of some SRLG.

   a) Let $G = (N, \Sigma, P, S)$ be a SRLG. Define the NFA $N_G \stackrel{\text{def}}{=} (N, \Sigma, \Delta, \{S\}, F)$ where we define $\Delta(X, a) \stackrel{\text{def}}{=} \{Y \in N \mid (X \to aY) \in P\}$ and $F \stackrel{\text{def}}{=} \{X \in N \mid (X \to \epsilon) \in P\}$. Then:

$$
\begin{array}{rcll}
x \in L(N_G) & \Leftrightarrow & \hat{\Delta}(\{S\}, x) \cap F \neq \emptyset & \{\text{Def. } L(N)\} \\
& \Leftrightarrow & \{X \in N \mid S \to_G^* xX\} \cap F \neq \emptyset & \{\text{Lemma A}\} \\
& \Leftrightarrow & \exists X \in N. \, (S \to_G^* xX \wedge X \to_G \epsilon) & \{\text{Def. } F\} \\
& \Leftrightarrow & S \to_G^+ x & \{\text{Def. } \to_G^*\} \\
& \Leftrightarrow & x \in L(G) & \{\text{Def. } L(G)\}
\end{array}
$$

   So $L(G) = L(N_G)$ and hence $L(G)$ is regular. In the proof, Lemma A states that

$$
\hat{\Delta}(\{Y\}, x) = \{X \in N \mid Y \to_G^* xX\}
$$

   which is proved by induction on the structure of x as follows.
   Base case.

$$
\begin{array}{rcll}
\hat{\Delta}(\{Y\}, \epsilon) & = & \{Y\} & \left\{\text{Def. } \hat{\Delta}\right\} \\
& = & \{X \in N \mid Y \to_G^* X\} & \{\text{Def. SRLG}\}
\end{array}
$$

   Induction. Assume the Lemma holds for x (the ind. hyp.); we show that it then also holds for xa.

$$
\begin{array}{rcll}
\hat{\Delta}(\{Y\}, xa) & = & \bigcup_{X \in \hat{\Delta}(\{Y\}, x)} \Delta(X, a) & \left\{\text{Def. } \hat{\Delta}\right\} \\
& = & \bigcup_{X \in \left\{X \in N \mid Y \to_G^* xX\right\}} \Delta(X, a) & \{\text{Ind. hyp.}\} \\
& = & \bigcup_{X \in \left\{X \in N \mid Y \to_G^* xX\right\}} \{Z \in N \mid (X \to aZ) \in P\} & \{\text{Def. } \Delta\} \\
& = & \{X \in N \mid Y \to_G^* xaX\} & \{\text{Def. } \to_G^*\}
\end{array}
$$

b) This part is similar to the previous one and is only sketched here. Let A be a regular language. Then there is a DFA $M_A = (Q, \Sigma, \delta, s, F)$ such that $L(M_A) = A$. We define the SLRG $G_A \stackrel{\text{def}}{=} (Q, \Sigma, P, s)$ where $P \stackrel{\text{def}}{=} \{q_1 \to aq_2 \mid \delta(q_1, a) = q_2\} \cup \{q \to \epsilon \mid q \in F\}$. We then show $x \in L(G_A) \Leftrightarrow x \in L(M_A) = A$, the proof of which is best structured by proving and using Lemma B:

$$q_1 \to^*_{G_A} xq_2 \Leftrightarrow \hat{\delta}(q_1, x) = q_2$$

---

5. Recall the Chomsky–Schützenberger Theorem, textbook Supplementary Lecture G. Show how this Theorem applies to the context–free language PAL of even–length palindromes over $\{a, b\}$, by identifying a suitable number n, a regular language R, and a homomorphism (renaming) h. 3p

Solution: PAL is equal to the language $h(\text{PAREN}_n \cap R)$ for $n = 2$, $R = L(([_1 + [_2)^*(]_1 +]_2)^*)$ and h renaming $[_1$ and $]_1$ to a and $[_2$ and $]_2$ to b.

---

6. Consider the language: 5p

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

(a) Refer to the closure properties of regular languages to argue that A is not regular.
Solution: The regular lanuguages are closed under intersection, but $A \cap L(a^*b^*)$ equals $\{a^n b^n \mid n \geq 0\}$ which is not regular, hence A cannot be regular.

(b) Refer to the closure properties of context–free languages to argue that A is context–free.
Solution: The context–free languages are closed under language concatenation, and since A can be represented as the concatenation of the context–free languages $\left\{a^k b^k \mid k \geq 0\right\}$ and $\{b^m a^m \mid m \geq 0\}$, A must be context–free.

(c) Give a context–free grammar G generating A.
Solution: From the previous observation, and the construction on grammars we used to show this closure property, we directly obtain the grammar:
$S \to AB$
$A \to \epsilon \mid aAb$
$B \to \epsilon \mid bBa$

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

---

7. Give a detailed description (preferably as a graph) of a Turing machine with input alphabet $\{a, b\}$, that on any input x halts with string $y \in L(a^*b^*)$ written on its tape, where $\sharp a(x) = \sharp a(y)$ and $\sharp b(x) = \sharp b(y)$. Explain how your Turing machine achieves its task. 4p

Solution: (Idea) There are many possible solutions, but one simple idea is to use insertion sort: Iteratively "swap" the left–most b with the first following a (using renaming), until there are no more such a's. (6 states suffice!)

---

8. Recall Rice's Theorem, textbook Lecture 34. Explain why the trivial properties of the recursively enumerable sets are decidable, by suggesting suitable total Turing machines for these properties. 2p

Solution: There are exactly 2 trivial properties: the empty set (of r.e. sets) and the set of all r.e. sets. Since we represent every r.e. set by some (encoding of a) Turing machine accepting this set, the two trivial properties can be represented, by using some fixed encoding of Turing machines, as the languages $\left\{\hat{M} \mid \text{false}\right\}$, which is the empty set, and $\left\{\hat{M} \mid \text{true}\right\}$, which is the set of all legal Turing machine encodings.

A total Turing machine M_F accepting the first language is simply one that upon reading $\vdash$ immediately enters its reject state, while a total Turing machine M_T accepting the second language is one which decides whether the input string is a legal encoding of some Turing machine according to the chosen encoding scheme.