Finite Automata and Formal Languages

TMV026/DIT321

Thursday 27th of May 2010

CTH: Total 60 points: $\geq$ 26: 3, $\geq$ 38: 4, $\geq$50: 5
GU: Total 60 points $\geq$26: G, $\geq$42: VG

No help material.

Answers can be written in English or Swedish. Write as clear as possible.

All answers should be well motivated. Points will be deduced when you give an unnecessarily complicated solution or when you do not properly justify your answer.

1. (6pts) Consider the following context-free grammar with start symbol S:

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

Prove using induction that if $w \in \{0,1\}^*$ and $S \Rightarrow^* w$ then w has the same number of 0's than of 1's.

(Example: the word 01101 has 2 0's and 3 1's.)

2. (3pts) Define a deterministic finite automata accepting the language over $\{0,1\}$ not containing the strings with at least 3 consecutive 0's.

3. (6pts) Minimise the following automaton.

	a	b
$\rightarrow q_0$	q_3	q_5
q_1	q_6	q_3
q_2	q_6	q_4
q_3	q_6	q_6
$^*q_4$	q_0	q_5
$^*q_5$	q_2	q_4
q_6	q_1	q_6

Show the table that identifies the distinguishable states and justify the construction of the automaton.

4. (a) (4pts) Convert the following non-deterministic finite automata to an equivalent deterministic finite automata.

	a	b
$\rightarrow q_0$	$\{q_0, q_1\}$	$\{q_2\}$
q_1	$\{q_3, q_4\}$	$\{q_1, q_2\}$
q_2	$\{q_1, q_2\}$	$\{q_3, q_4\}$
$^*q_3$	$\{q_3\}$	$\{q_4\}$
$^*q_4$	$\{q_4\}$	$\{q_3\}$

   (b) (2.5pts) Describe with words the language accepted by the deterministic finite automata you constructed.

   (c) (2.5pts) Give a regular expression which generates the language you described in **??**).

5. Do these two regular expressions represent the same language? Justify your answer.

   (a) (2.5pts) $(a+b)^*$ and $(a^*b^*)^*$?

   (b) (2.5pts) $a^*(a+b)^*$ and $(a+b)^*$?

6. (5pts) Which of the following languages are regular? Give a regular expression or use the Pumping lemma for regular languages to justify your answer.

   (a) $\mathcal{L} = \{b^n a^{3m} \mid n \geqslant 0, m \geqslant 0\}$

   (b) $\mathcal{L} = \{a^n b^{3n} \mid n \geqslant 0\}$

7. Give examples of languages $\mathcal{L}_1$ and $\mathcal{L}_2$ such that

   (a) (2.5pts) $\mathcal{L}_1$ is regular, $\mathcal{L}_2$ is not regular and $\mathcal{L}_1 - \mathcal{L}_2$ is regular.

   (b) (2.5pts) $\mathcal{L}_1$ and $\mathcal{L}_2$ are not regular but $\mathcal{L}_1 \cup \mathcal{L}_2$ is regular.

8. (a) (5pts) Give a context-free grammar that generates the language $\{a^i b^j c^k \mid i,j,k > 0,\ (i > k \text{ or } i < j)\}$. Explain your grammar!

   (b) (2pts) Is the grammar ambiguous? Justify.

9. Consider the following context-free grammar with starting symbol S:

$$S \rightarrow aSb \mid aSbb \mid ab \mid abb$$

   (a) (2pts) Describe informally the language generated by the grammar.

   (b) (2pts) Is the grammar ambiguous? Justify.

   (c) (2pts) Convert the grammar to an equivalent grammar in Chomsky Normal Form.

   (d) (4pts) Apply the CYK algorithm to see if the word $aabbb$ belongs to the language generated by the grammar. Show the table and justify your answer.

10. (4pts) Define formally what a Turing machine is. Describe informally how it works.

Exam Solutions

In the exam, you need to explain a bit more your solutions.

1. We will use course of value induction on the length of the derivation (number of steps).

   Base case: Length of the derivation is 1. Then $S \Rightarrow w$. Then $w = \epsilon$ which obviously has the same number of 0's than of 1's.

   Inductive step: Length of the derivation is $n + 1$. Our inductive hypothesis is that any word w' derived from S in at most n steps has the same number of 0's than of 1's.

   We have 2 cases depending on the first step in the derivation:

   - $S \Rightarrow 0S1S \Rightarrow^* w$. Then $w = 0u1v$ such that $S \Rightarrow^* u$ and $S \Rightarrow^* v$. In addition, the derivations for u and v take at most n steps.

     The number of 0's in w is 1 + the number of 0's in u + the number of 0' in v. The number of 1's in w is 1 + the number of 1's in u + the number of 1's in v.

     By inductive hypothesis we know that the number of 0's in u and in v are the same as the number of 1's in u and in v respectively.

     Then we know that the number of 0's in w is the same as the number of 1's in w.
   - $S \Rightarrow 1S0S \Rightarrow^* w$. Then $w = 1u0v$ such that $S \Rightarrow^* u$ and $S \Rightarrow^* v$. In addition, the derivations for u and v take at most n steps.

     The rest is similar to the case above.

2.

	0	1
$\rightarrow^* q_0$	q_1	q_0
$^*q_1$	q_2	q_0
$^*q_2$	q_3	q_0
q_3	q_3	q_3

3. First we need to run the algorithm that identifies equivalent states.

	q_0	q_1	q_2	q_3	q_4	q_5
q_6	X		X		X	X
q_5	X	X	X	X		
q_4	X	X	X	X		
q_3	X		X			
q_2		X				
q_1	X					

   Help distinguishing states:
   $(q_0, q_1) \rightarrow_b (q_5, q_3)$
   $(q_0, q_3) \rightarrow_b (q_5, q_6)$
   $(q_0, q_6) \rightarrow_b (q_5, q_6)$
   $(q_1, q_2) \rightarrow_b (q_3, q_4)$

$(q_2, q_3) \to_b (q_4, q_6)$
$(q_2, q_6) \to_b (q_4, q_6)$

Show the equivalences:
$(q_1, q_6) \to_a (q_6, q_6)$
$(q_1, q_6) \to_b (q_3, q_6)$

$(q_3, q_6) \to_a (q_6, q_1)$
$(q_3, q_6) \to_b (q_6, q_6)$

$(q_4, q_5) \to_a (q_0, q_2)$
$(q_4, q_5) \to_b (q_5, q_4)$

$(q_0, q_2) \to_a (q_3, q_6)$
$(q_0, q_2) \to_b (q_5, q_4)$

$(q_1, q_3) \to_a (q_6, q_6)$
$(q_1, q_3) \to_b (q_3, q_6)$

The resulting automaton is (remember that equivalence of states is transitive):

	a	b
$\to q_0q_2$	$q_1q_3q_6$	q_4q_5
$q_1q_3q_6$	$q_1q_3q_6$	$q_1q_3q_6$
$^*q_4q_5$	q_0q_2	q_4q_5

4. (a)

	a	b
$\to q_0$	q_0q_1	q_2
q_0q_1	$q_0q_1q_3q_4$	q_1q_2
q_2	q_1q_2	q_3q_4
q_1q_2	$q_1q_2q_3q_4$	$q_1q_2q_3q_4$
$^*q_0q_1q_3q_4$	$q_0q_1q_3q_4$	$q_1q_2q_3q_4$
$^*q_1q_2q_3q_4$	$q_1q_2q_3q_4$	$q_1q_2q_3q_4$
$^*q_3q_4$	q_3q_4	q_3q_4

   (b) The language consists of any string containing at least 2 a's or 2 b's.

   (c) $a(a+b)^*a(a+b)^* + b(a+b)^*b(a+b)^*$

5. (a) Yes.
   Words of the form $(a+b)^*$ can be generated by $(a^*b^*)^*$ by taking just one a and the ϵ or ϵ and one b in each iteration.
   Words of the form $(a^*b^*)^*$ can be generated by $(a+b)^*$ by generating as many a's and then as many b's as necessary for each sequence of a's followed by the sequence of b's.

(b) Yes.

Words of the form $a^*(a+b)^*$ can be generated by $(a+b)^*$ by first generating as many a's as needed and the generating the a's and the b's each at a time.

Words of the form $(a+b)^*$ are an especial case of words of the form $a^*(a+b)^*$, where we take ϵ for a^*.

6. (a) The regular expression is $b^*(aaa)^*$.

   (b) The language is not regular. With the n of the Pumping lemma we give the word $a^n b^{3n}$. Then y should contain only a's. Eventually xy^kz will contain more a's than b's (at least for $k > 3n/m$ with $|y| = m$).

7. (a) $\mathcal{L}_1 = a^*$ and $\mathcal{L}_2 = \{b^n c^n \mid n > 0\}$. Here $\mathcal{L}_1 - \mathcal{L}_2 = \mathcal{L}_1$.

   (b) $\mathcal{L}_1 = \{a^i b^j \mid i \leqslant j\}$ and $\mathcal{L}_2 = \{a^i b^j \mid i \geqslant j\}$. Here $\mathcal{L}_1 \cup \mathcal{L}_2 = a^* b^*$.

8. (a)

$$
\begin{aligned}
&S \to P \mid Q \\
&A \to a \mid aA \\
&B \to b \mid bB \\
&C \to c \mid cC \\
&P \to aPc \mid aABc \\
&Q \to RC \\
&R \to aRb \mid aBb
\end{aligned}
$$

   - S is the start symbols of the grammar: P generates the part $a^i b^j c^k$ with $i > k$, and Q generates the part $a^i b^j c^k$ with $i < j$
   - A generates 1 or more a's
   - B generates 1 or more b's
   - C generates 1 or more c's
   - P generates $a^i b^j c^k$ such that $i > k$
     aPc generates equal amount of a's and c's. When we are done adding c's. $aABc$ adds as many a's as we want (at least one) and all the b's as well.
   - R generates $a^i b^j$ such that $i < j$.
     aRb generates as many a's as b's, aBb adds all the extra b's (at least one).
   - Q generates $a^i b^j c^k$ such that $i < j$ by using R and adding all the c's with C.

   (b) Yes, the words $aabbbc$ has the following 2 left-most derivations:

   - $S \Rightarrow P \Rightarrow aABc \Rightarrow aaBC \Rightarrow aabBc \Rightarrow aabbBc \Rightarrow aabbbc$
   - $S \Rightarrow Q \Rightarrow RC \Rightarrow aRbC \Rightarrow aaBbbC \Rightarrow aabbbC \Rightarrow aabbbc$

9. (a) The words are of the form $a^i b^j$ with $0 < i \leqslant j \leqslant 2i$.

   (b) Yes, the word $aabbb$ has the following 2 left-most derivations:

   - $S \Rightarrow aSb \Rightarrow aabbb$
   - $S \Rightarrow aSbb \Rightarrow aabbb$

(c)

$$
\begin{array}{ll}
S \to AP \mid AQ \mid AB \mid AC & \\
P \to SB & Q \to SC \\
A \to a & B \to b \\
C \to BB &
\end{array}
$$

(d)

$$
\begin{array}{|ccccc}
\{S,P\} & & & & \\
\{S\} & \{Q,P\} & & & \\
\emptyset & \{S,P\} & \emptyset & & \\
\emptyset & \{S\} & \{C\} & \{C\} & \\
\{A\} & \{A\} & \{B\} & \{B\} & \{B\} \\
\hline
a & a & b & b & b
\end{array}
$$

S belongs to the upper-most set, which means that the word is generated by the grammar since S is the starting symbol of the grammar.

10. See slides 3 to 7 in lecture 14.