Finite Automata Theory and Formal Languages

TMV026/TMV027/DIT321 – Responsible: Ana Bove

Wednesday 21 of August 2013
Total: 60 points

TMV027/DIT321 registration VT13 Exam valid 6hp	TMV026/DIT321 registration before VT13 Exam valid 7.5 hp
CTH: $\geqslant 27$: 3, $\geqslant 40$: 4, $\geqslant 50$: 5 GU: $\geqslant 27$: G, $\geqslant 45$: VG	CTH: $\geqslant 33$: 3, $\geqslant 43$: 4, $\geqslant 53$: 5 GU: $\geqslant 33$: G, $\geqslant 50$: VG

No help material but dictionaries to/from English or Swedish.

Write in English or Swedish, and as readable as possible (think that what we cannot read we cannot correct).

OBS: 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.

Good luck!

---

1. (5pts) Prove by using induction that $\sum_{i \geqslant 0}^{n} i(i+1) = n(n+1)(n+2)/3$. Do not forget to clearly state which kind of induction you are using, the property you will prove, the base case(s) and the inductive hypothesis(es)!

2. Consider the language $\{a^n b \mid n \neq 4m, m \geqslant 0\}$.
   (a) (3pts) Construct a DFA that accepts this language.
   (b) (3pts) Use your intuition and give a (simple) regular expression that generates this language.

3. Consider the following NFA

	0	1
$\rightarrow q_0$	$\{q_1, q_2\}$	$\{q_4\}$
q_1	$\emptyset$	$\{q_0\}$
q_2	$\emptyset$	$\{q_3\}$
q_3	$\{q_0\}$	$\emptyset$
$^*q_4$	$\emptyset$	$\emptyset$

   (a) (2pts) User your intuition and give a (simple) regular expression that generates the language accepted by this NFA.
   (b) (4pts) Convert this NFA to an equivalent DFA.

4. (4.5pts) Minimise the following automaton. Show the intermediate table and justify the construction of the new automaton.

	0	1
$\rightarrow q_0$	q_1	q_2
q_1	q_2	q_3
q_2	q_2	q_4
$^*q_3$	q_3	q_4
$^*q_4$	q_3	q_4
$^*q_5$	q_5	q_4

5. (4pts) Compute, using any of the methods given in class, a regular expression generating the language accepted by the DFA below. Show enough intermediate steps to follow what you are doing!

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

6. (a) (2pts) Under which operations are regular languages closed? Name all those you can remember.
   (b) (2ts) Show that if $\mathcal{L}_1$ and $\mathcal{L}_2$ are regular, then $\{w \mid w \notin \mathcal{L}_1 \text{ and } w \notin \mathcal{L}_2\}$ is also regular.
   (c) (3pts) Is it true that if $\mathcal{L}_1$ is regular, $\mathcal{L}_2$ is not regular and $\mathcal{L}_1 \cap \mathcal{L}_2$ is regular, then $\mathcal{L}_1 \cup \mathcal{L}_2$ is not regular? Justify your answer as formal as you can.

7. (a) (2pts) What is the definition of a context-free grammar?
   (b) (2pts) What are the steps one needs to perform, and in which order, if one wants to simplify a grammar?

8. (a) (5pts) Give a context free grammar that generates the language $\{a^n b^m c^k \mid k = |n-m|\}$. Explain your grammar!
   (b) (1.5pts) Is your grammar ambiguous? Justify.
   (c) (2.5pts) Give the leftmost derivation and the parse tree of the word *aabbbc*.

9. (a) (2pts) Formulate the Pumping lemma for context-free languages.
   (b) (4.5pts) Use this lemma to show that $\{a^i b^j a^i b^j \mid i, j \geqslant 0\}$ is not context-free.

10. (4pts) Consider the following grammar with start symbol S:

$$S \to AB \qquad A \to BB \mid a \qquad B \to AB \mid b$$

Apply the CYK algorithm to determine if the string *aabab* is generated by this grammar. Show the resulting table and justify your answer.

11. (4pts) Assume your input tape has at most one word written in it. Define a Turing machine (or give its high-level description) over the alphabet $\{a, b\}$ which does nothing if the input tape has less than three non-empty symbols, and otherwise writes the 3rd symbol from the left at the end of the input. For example, if the input is *aabaaba* the output becomes *aabaabab*. Explain your machine.

Solutions Exam 130821

Here we only give a brief explanation of the solution. Your solutions should in general be more elaborated than these ones.

1. Let $P(n)$ be $\sum_{i\geqslant 0}^{n} i(i+1) = n(n+1)(n+2)/3$.
   We shall use mathematical induction to prove that $\forall n.P(n)$.

   Base case is n = 0: Then $P(0)$ amounts to proving that $0(0+1) = 0(0+1)(0+2)/3$, that is, $0 = 0$ which is indeed the case.

   Our IH is that $P(n)$ holds.
   We need now to prove $P(n+1)$, that is, $\sum_{i\geqslant 0}^{n+1} i(i+1) = (n+1)(n+2)(n+3)/3$.

   $\sum_{i\geqslant 0}^{n+1} i(i+1) = \sum_{i\geqslant 0}^{n} i(i+1) + (n+1)(n+2) =$ by IH $= n(n+1)(n+2)/3 + (n+1)(n+2) = n(n+1)(n+2)/3 + 3(n+1)(n+2)/3 = (n+1)(n+2)(n+3)/3$.

2. (a)

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

   (b) $(aaaa)^*(aaa + aa + a)b$

3. (a) $(01 + 010)^*1$

   (b)

	0	1
$\rightarrow q_0$	q_1q_2	q_4
q_1q_2	$-$	q_0q_3
q_0q_3	$q_0q_1q_2$	q_4
$q_0q_1q_2$	q_1q_2	$q_0q_3q_4$
$^*q_0q_3q_4$	$q_0q_1q_2$	q_4
$^*q_4$	$-$	$-$

4. First we eliminate q_5 since it is not reachable.

   Then, we run the algorithm that identifies equivalent states (you should explain a bit its construction):

	q_0	q_1	q_2	q_3
q_4	X	X	X	
q_3	X	X	X	
q_2	X			
q_1	X			

   The resulting automaton is (you should explain a bit its construction):

	a	b
$\rightarrow q_0$	q_1q_2	q_1q_2
q_1q_2	q_1q_2	q_3q_4
$^*q_3q_4$	q_3q_4	q_3q_4

5.

$$\begin{array}{lll}
E_0 = 0E_1 + 1E_2 & & E_0 = (00^*11^*0+1)E_2 \\
E_1 = 0E_1 + 1E_3 & E_1 = 0E_1 + 11^*0E_2 & E_1 = 0^*11^*0E_2 \\
E_2 = 0E_1 + 1E_2 + \epsilon & & E_2 = (00^*11^*0+1)E_2 + \epsilon \\
E_3 = 0E_2 + 1E_3 & E_3 = 1^*0E_2 & E_3 = 1^*0E_2
\end{array}$$

Then $E_2 = (00^*11^*0+1)^*$ hence $E_0 = (00^*11^*0+1)(00^*11^*0+1)^* = (0^+1^+0+1)^+$.

6. (a) See lecture 8, slides 12–20.

(b) If $\mathcal{L}_1$ and $\mathcal{L}_2$ are regular then $\overline{\mathcal{L}_1}$ and $\overline{\mathcal{L}_2}$ are also regular and hence $\overline{\mathcal{L}_1} \cap \overline{\mathcal{L}_2}$ (which is the desired set) is also regular.

(c) Yes. Let us assume that $\mathcal{L}_1 \cup \mathcal{L}_2$ is regular. Then $(\mathcal{L}_1 \cup \mathcal{L}_2 - \mathcal{L}_1) \cup (\mathcal{L}_1 \cap \mathcal{L}_2)$ should also be regular. But this is actually $\mathcal{L}_2$ which is not regular.

7. (a) See lecture 10, slide 6.

(b) See lecture 12, slide 17.

8. (a) Let us consider 2 cases: $n \leqslant m$ hence $k = m - n$ (generated by $S \to LM$), and $n \geqslant m$ hence $k = n - m$ (generated by $S \to G$).

L then generates equally many a's and b's (possibly 0 if there are c's in the word).

M generates equally many b's and c's (even 0, allowing the generation of the empty word).

G starts by generating a's and c's and then moves to N to generate a's and b's.

$$\begin{array}{ll}
S \to LM \mid G & \\
L \to ab \mid aLb \mid \epsilon & M \to bc \mid bMc \mid \epsilon \\
G \to ac \mid aGc \mid N & N \to ab \mid aNb
\end{array}$$

(b) Yes, whenever we generate a word $a^n b^n, n \geqslant 0$ this can be done via LM or via G.

(c) Leftmost derivation: $S \Rightarrow LM \Rightarrow aLbM \Rightarrow aabbM \Rightarrow aabbbc$.

Parse tree: root S, subtrees L and M.
Subtrees of L: a, L and b, subtrees of this second L: a and b.
Subtrees of M: b and c.

9. (a) See lecture 12 slide 22.

(b) Let n be the constant of the PL. Let $w = a^n b^n a^n b^n$ which is clearly $|w| \geqslant n$.

We know that $w = xuyvz$ such that $uv \neq \epsilon$ and $|uyv| \leqslant n$.

Hence uyv contains only a's (from the first or the second half of the word), only b's (from the first or the second half of the word), a's and b's (from the first or the second half of the word) or b's and a's (from the middle), but never a's and b's and a's or b's and a's and b'.

If uyv contains only 1 letter, xu^kyv^kz will contain a different amount of that letter in the first than in the second half of the word $\forall k \neq 1$.

If uyv contains a's and b's from the first (second) half of the word, then xu^kyv^kz will contain a different amount of a's and b's in the second (first) half of the word $\forall k \neq 1$.

If uyv contains b's and a's from the middle part of the word, then xu^kyv^kz will contain a different amount of a's and b's in beginning and end of the word $\forall k \neq 1$.

10.

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

Since S does NOT belong to the upper set then the string is NOT generated by the grammar.

11. If in the initial state the TM reads an empty symbols it goes to the final state and does nothing more. Otherwise it moves to the right and goes to the state that reads the second symbol.

    If the second symbols is empty the TM goes to the final state and does nothing more. Otherwise it moves to the right and goes to the state that reads the third symbol.

    If the third symbols is empty the TM goes to the final state and does nothing more. Otherwise it moves to the right and goes to a state that remembers the the symbol that was read (one state for symbol a and one state for symbol b).

    Either of these states will move to the right so long the TM reads a non-empty symbol. When this state reads an empty symbol it will write the symbol that the state has remembered and go to the final state and do nothing more.