Finite Automata Theory and Formal Languages

TMV027/DIT321 (6 hec) — Responsible: Ana Bove, tel: 1020

Wednesday 1st of June 2016

Total: 60 points	
CTH: $\geqslant$ 27: 3, $\geqslant$ 38: 4, $\geqslant$49: 5	GU: $\geqslant$27: G, $\geqslant$45: 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. (5.5pts) Consider the following context-free grammar with start symbol S:

$$S \to aaS \mid SS \mid ab$$

Prove using induction that any word generated by the grammar has an even length, it starts with an a and it finishes with a b.

Do not forget to clearly state the property you will prove, which kind of induction you will use, the base case(s) and the inductive hypothesis(es)!

2. (4.5pts) Construct a finite automaton without ϵ-transitions that recognises the language over $\{0, 1, 2\}$ where words do not contain 012 as substring OR have an even number of 0's and 1's (together).

3. (5.5pts) Convert the following ϵ-NFA into an equivalent DFA.

	0	1	ϵ
$\to q_0$	$\{q_0, q_1\}$	$\{q_2\}$	$\{q_5\}$
q_1	$\{q_3\}$	$\{q_1\}$	$\emptyset$
q_2	$\{q_2, q_4\}$	$\emptyset$	$\{q_1\}$
q_3	$\{q_5\}$	$\{q_1\}$	$\{q_4\}$
q_4	$\{q_5\}$	$\{q_3, q_4\}$	$\emptyset$
$^*q_5$	$\emptyset$	$\emptyset$	$\{q_0\}$

4. (4.5pts) Compute, using any of the methods given in class (NOT your intuition!), a regular expression generating the language accepted by the DFA below.

	0	1	2
$\to q_0$	q_1	q_0	q_2
q_1	q_1	q_2	q_3
q_2	q_1	q_3	q_2
$^*q_3$	q_3	q_3	q_3

Solve/eliminate first q_1, then q_2 and finally q_3 (if you use equations). Show enough intermediate steps so we can follow what you are doing!

5. (4.5pts) Minimise the following DFA using the method given in class (NOT your intuition!)

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

Show the table that identifies the distinguishable states, give the equivalent classes of states and the new automaton.

6. (a) (1pts) State the Pumping lemma for regular languages.
   (b) (5pts) Which of the following languages over $\{0, 1\}$ are regular? Give a regular expression or use the Pumping lemma for regular languages to justify your answer.
      i. $\mathcal{L}_1 = \{w \mid \#_0(w) = 2 \times \#_1(w) \vee \#_0(w) = 3 \times \#_1(w)\}$, where $\#_0$ and $\#_1$ are the functions that count the numbers of 0's and 1's respectively in a word.
      ii. $\mathcal{L}_2 = \mathcal{L}_1 \cap \{w \mid |w| \leqslant 5\}$.

7. (a) (5.5pts) Give a non-ambiguous context-free grammar generating the language $\{a^n b^m c^k d^l \mid n + m = k - l\}$. Consider that $k - l = 0$ if $l > k$.
   (b) (3.5pts) Explain the grammar, why it produces exactly this language and why it is non-ambiguous.
   (c) (2pts) Give the leftmost derivation of a word with two more c's than d's, and the parse tree of a word with two more d's than c's.

8. Consider the following grammar with start symbol S:

$$S \rightarrow ab \mid ASB \mid CSD \qquad A \rightarrow aA \mid B \qquad B \rightarrow \epsilon \mid Bb \qquad C \rightarrow c \mid cC \mid D$$
$$D \rightarrow dD \mid eE \qquad E \rightarrow EE \qquad F \rightarrow FF \mid f$$

   (a) (2pts) State which are the nullable variables in the grammar and eliminate the ϵ-productions.
   (b) (2pts) State which are the unit productions in the grammar from a) and eliminate them.
   (c) (1.5pt) State which are non-generating symbols in the grammar from b) and eliminate them.
   (d) (1.5pt) State which are non-reachable symbols in the grammar from c) and eliminate them.
   (e) (2pts) Convert the simplified grammar into Chomsky normal form.

9. (4pts) Consider the following grammar with start symbol S:

$$S \rightarrow AB \mid c \qquad A \rightarrow a \mid SA \mid BA \qquad B \rightarrow b \mid SB \mid AB$$

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

10. (a) (4pts) Define a Turing machine that determines if the input tape is of the form $0^{2n}112^{2m}$ for $n, m \geqslant 0$. Give either the transition function of the machine or its transition diagram. You can assume that the initial tape only contains symbols in $\{0, 1, 2\}$ (in addition to the blank symbol).
    (b) (1.5pts) Is your machine a Turing decider? Justify your answer.

Solutions Exam 160601

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

1. Our property is: $P(n)$: if $S \Rightarrow^n w$ then the length of w is even, w starts with an a and it finishes with a b.

   We will use course-of-value induction on the length of the derivation (number of steps) $S \Rightarrow^n w$.

   Base case: $S \Rightarrow w$, hence the rule applied should have been $S \to ab$.

   Here, the length of ab is 2, it starts with a and finishes with b.

   Step case: Our IH is: if $S \Rightarrow^* w$ in at most $n > 0$ steps then the length of w is even, w starts with an a and it finishes with a b.

   Let $S \Rightarrow^* w$ in $n + 1$ steps with $n > 0$.

   Since $n > 0$ then the first rule applied should have been $S \to aaS$ or $S \to SS$.

   In the case the first rule was $S \to aaS$ then $w = aaw'$ with $S \Rightarrow^* w'$ in n steps. Then the IH applies for w'. Since the length of w' is even (by IH) so is the length of aaw'. aaw' clearly starts with a and since w' ends with b (by IH) so does aaw'.

   In the case the first rule was $S \to SS$ then $w = w'w''$ with $S \Rightarrow^* w'$ and $S \Rightarrow^* w''$, each in at most n steps. Then the IH applies for both w' and w''. Since the length of w' and of w'' are even (by IH) so is the lengths of $w'w''$. Since w' starts with an a (by IH) so does $w'w''$. Finally, since w'' ends with a b (by IH) so does $w'w''$.

2. We define a NFA:

	0	1	2
$\to^* q_0$	$\{q_1, q_4\}$	$\{q_2, q_4\}$	$\{q_0\}$
$^*q_1$	$\{q_1\}$	$\{q_3\}$	$\{q_2\}$
$^*q_2$	$\{q_1\}$	$\{q_2\}$	$\{q_2\}$
q_3	$\{q_1\}$	$\{q_2\}$	$\emptyset$
q_4	$\{q_5\}$	$\{q_5\}$	$\{q_4\}$
$^*q_5$	$\{q_4\}$	$\{q_4\}$	$\{q_5\}$

3.

	0	1
$\to^* q_0q_5$	$q_0q_1q_5$	q_1q_2
$^*q_0q_1q_5$	$q_0q_1q_3q_4q_5$	q_1q_2
$^*q_0q_1q_3q_4q_5$	$q_0q_1q_3q_4q_5$	$q_1q_2q_3q_4$
q_1q_2	$q_1q_2q_3q_4$	q_1
q_1	q_3q_4	q_1
$q_1q_2q_3q_4$	$q_0q_1q_2q_3q_4q_5$	$q_1q_3q_4$
$q_1q_3q_4$	$q_0q_3q_4q_5$	$q_1q_3q_4$
q_3q_4	q_0q_5	$q_1q_3q_4$
$^*q_0q_3q_4q_5$	$q_0q_1q_5$	$q_1q_2q_3q_4$
$^*q_0q_1q_2q_3q_4q_5$	$q_0q_1q_2q_3q_4q_5$	$q_1q_2q_3q_4$

4. I will solve equations:

$$\begin{aligned} E_0 &= 0E_1 + 1E_0 + 2E_2 \\ E_1 &= 0E_1 + 1E_2 + 2E_3 \\ E_2 &= 0E_1 + 2E_2 + 1E_3 \\ E_3 &= (0+1+2)E_3 + \epsilon \end{aligned}$$

$$\begin{aligned} E_0 &= 00^*1E_2 + 00^*2E_3 + 1E_0 + 2E_2 = 1E_0 + (00^*1+2)E_2 + 00^*2E_3 \\ E_1 &= 0^*(1E_2 + 2E_3) = 0^*1E_2 + 0^*2E_3 \\ E_2 &= 00^*1E_2 + 00^*2E_3 + 2E_2 + 1E_3 = (00^*1+2)E_2 + (00^*2+1)E_3 \end{aligned}$$

$$\begin{aligned} E_0 &= 1E_0 + (00^*1+2)(00^*1+2)^*(00^*2+1)E_3 + 00^*2E_3 = 1E_0 + ((00^*1+2)^+(00^*2+1) + 00^*2)(0+1+2)^* \\ E_2 &= (00^*1+2)^*(00^*2+1)E_3 \\ E_3 &= (0+1+2)^* \end{aligned}$$

Hence (recall $0^+ = 00^*$)

$$E_0 = 1^*((0^+1+2)^+(0^+2+1) + 0^+2)(0+1+2)^*$$

5.

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

The equivalent classes are $\{q_0\}, \{q_1, q_4\}, \{q_2\}, \{q_3\}, \{q_5\}, \{q_6\}$.

The resulting automaton is:

	a	b
$\to q_0$	q_1q_4	q_0
q_1q_4	q_2	q_3
q_2	q_2	q_2
q_3	q_0	q_6
q_5	q_1q_4	q_6
q_6	q_5	q_6

6. (a) See slide 8 lecture 8.

   (b) i. (3.5pts) Let us assume $\mathcal{L}_1$ is a regular language. Hence the PL should apply.
   Let n be the constant given by the PL.
   Let $w = 0^{3n}1^n$. We have that $w \in \mathcal{L}_1$ and that $|w| \geqslant n$.
   Hence $w = xyz$ with $y \neq \epsilon$ and $|xy| \leqslant n$.
   So y should contain only 0's and at least one 0.
   For any $k > 1$ then xy^kz will contain more than $3n$ 0's (and hence more than $2n$ 0's) while only n 1's. Hence, $xy^kz \notin \mathcal{L}_1$ which contradicts the PL.
   Then, $\mathcal{L}_1$ cannot be regular.

   ii. (1.5pts) $\mathcal{L}_2 = \epsilon + 100 + 010 + 001 + 1000 + 0100 + 0010 + 0001$.

7. (a)

$$\begin{aligned} &S \to A \mid AD \mid BA \\ &A \to \epsilon \mid cAd \qquad B \to ac \mid aBc \mid C \\ &D \to d \mid dD \qquad C \to bc \mid bCc \end{aligned}$$

   (b) We can divide the situation into 3 (mutually exclusive) cases: $l = k$, $l > k$ and $l < k$.
   Observe that if $l \geqslant k$ then $n + m = k - l = 0$ and hence there are neither a's nor b's.
   A will generate the case where $l = k$, even the empty word ($l = k = 0$),

AD generates the case where $l > k$, hence there are more d's than c'. A will generate as many c's as d's (even 0) and then D will generate the extra d's, of which there should be at least one.

BA generates the case where $l < k$, hence there are more c's than d'. Here again A will generate as many c's as d's (even 0).
Now, for every extra c we need to make sure we add either an a or a b.
B will start putting as many a's as needed (adding alway a c for each a). If no a should be present we can directly go to C.
C will then put the necessary b's with their corresponding c's.
Observe that even the case where no b is present is also considered.
Note that any n and m determines a unique derivations.

Given this explanation it is easy to see that there is only one possible derivation for each word and hence the grammar is non-ambiguous.

(c) I present just leftmost derivations here, you will need to give the parse tree of the second one instead.

$$S \Rightarrow BA \Rightarrow aBcA \Rightarrow aCcA \Rightarrow abccA \Rightarrow abcc$$

$$S \Rightarrow AD \Rightarrow cAdD \Rightarrow cdD \Rightarrow cddD \Rightarrow cddd$$

8. (a) B and A are nullable.

$$S \to ab \mid ASB \mid SB \mid AS \mid S \mid CSD \qquad A \to a \mid aA \mid B \qquad B \to b \mid Bb \qquad C \to c \mid cC \mid D$$
$$D \to dD \mid eE \qquad E \to EE \qquad F \to FF \mid f$$

(b) unit productions: $S \to S, A \to B, C \to D$

$$S \to ab \mid ASB \mid SB \mid AS \mid CSD \qquad A \to a \mid aA \mid b \mid Bb \qquad B \to b \mid Bb \qquad C \to c \mid cC \mid dD \mid eE$$
$$D \to dD \mid eE \qquad E \to EE \qquad F \to FF \mid f$$

(c) non-generating symbols: D, E

$$S \to ab \mid ASB \mid SB \mid AS \quad A \to a \mid aA \mid b \mid Bb \quad B \to b \mid Bb \quad C \to c \mid cC \quad F \to FF \mid f$$

(d) non-reachable symbols: C, c, F, f

$$S \to ab \mid ASB \mid SB \mid AS \quad A \to a \mid aA \mid b \mid Bb \quad B \to b \mid Bb$$

(e)

$$S \to PQ \mid AX \mid SB \mid AS \qquad A \to a \mid PA \mid b \mid BQ \qquad B \to b \mid BQ$$
$$P \to a \qquad Q \to b \qquad X \to SB$$

9.

$\{S,B\}$				
$\emptyset$	$\{S,B\}$			
$\emptyset$	$\{A\}$	$\{S,B\}$		
$\emptyset$	$\emptyset$	$\{A\}$	$\{S,B\}$	
$\{A\}$	$\{S\}$	$\{S\}$	$\{A\}$	$\{B\}$
a	a	c	b	a

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. (a) Let $\Sigma = \{0, 1, 2\}$.

Let $M = (\{q_0, ..., q_4, q_f\}, \Sigma, \delta, q_0, \Box, \{q_f\})$, with δ is as follows:

$\delta(q_0, 0) = (q_1, 0, R)$	we have seen an odd nr of 0's so weq_1 will look for another one;
$\delta(q_1, 0) = (q_0, 0, R)$	so far we have seen an even number of 0's;
$\delta(q_0, 1) = (q_2, 1, R)$	we have seen an even nr of 0's and now we see the first one;
$\delta(q_2, 1) = (q_3, 1, R)$	here comes the second 1;
$\delta(q_3, 2) = (q_4, 2, R)$	we have seen an odd nr of 2's so q_4 will look for another one;
$\delta(q_3, \Box) = (q_f, \Box, R)$	we have seen an even nr of 0's, then two 1's and an even nr of 2's; the input has finished so we can accept;
$\delta(q_4, 2) = (q_3, 2, R)$	so far the input is good;

(b) Yes, it is a Turing decider. Whenever the input is not of the correct form the delta function is not defined so the machine will halt.

Observe that no state has a transition to itself (loop). The only "loops" in the running of the machine is when we count the parity of 0's (q_0 and q_1) and of 2's (q_3 and q_4), but the loop will end as soon as another symbol is read.