Finite Automata Theory and Formal Languages

TMV027/DIT321 (6 hec) — Responsible: Ana Bove/Andrea Vezzosi — Tel: 1062

Wednesday 17th of August 2016 — 8:30–12:30

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

$$S \to abS \mid SabS \mid ab$$

Prove using induction that any word w generated from the grammar in n steps is of the form $w = (ab)^i$ for $n \leqslant i \leqslant 2n - 1$.

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. (4pts) Construct a DFA with only <u>one</u> final state and as few states as possible that recognises the language $0(10)^*(0 + 11) + 1(01)^*(1 + 00)$.

To obtain full points, the DFA should contain no more than four states in total.

3. (5pts) Convert the following NFA into an equivalent DFA using the subset construction.

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

4. (4pts) 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_0	q_1	q_3
q_1	q_2	q_0	q_3
q_2	q_1	q_2	q_3
$^*q_3$	$-$	$-$	$-$

First eliminate q_1, then solve/eliminate q_2. Show enough intermediate steps so we can follow what you are doing!

5. (5pts) Minimise the following automaton.

	0	1
$\to q_0$	q_7	q_1
q_1	q_7	q_0
q_2	q_4	q_5
q_3	q_4	q_5
q_4	q_6	q_6
$^*q_5$	q_5	q_5
$^*q_6$	q_5	q_6
q_7	q_2	q_2

Show the table that identifies the distinguishable states and give the resulting minimised automaton.

6. (a) (2pts) Describe as detailed as possible the words contained in the language $\{0,1\}^* - \mathcal{L}((0+01)^*)$.
   (b) (2.5pts) Give a regular expression generating the language in 6a).

7. (a) (3.5pts) Show as formal as possible that the regular expressions $10(10)^*(0+11)+11$ and $1(01)^*(00+1)$ represent the same languages.
   Hint: This can be proved nicely by using the shifting rule ($R(SR)^* = (RS)^*R$).
   (b) (3.5pts) Let $\mathcal{L}_1, \mathcal{L}_2 \subset \{0,1\}^*$. Let $\mathcal{L}_1$ be non empty and regular, $\mathcal{L}_2$ be non regular, and $\mathcal{L}_1 \cap \mathcal{L}_2 = \emptyset$.
      i. Can $\mathcal{L}_1 \cup \mathcal{L}_2$ be non regular? Justify!
      ii. Can $\mathcal{L}_2 - \mathcal{L}_1$ be regular? Justify!

8. (a) (3.5pts) Let w^{rev} represent the reverse of w. Give a non-ambiguous context-free grammar without ϵ-productions generating the language $\{uwvw^{\mathsf{rev}} \mid u, v, w \in \{0,1\}^+ \text{ and } |u| \leqslant 2, |v| \leqslant 2\}$.
   (b) (2pts) Explain the grammar, why it produces exactly this language and why it is not ambiguous.
   (c) (1pt) Show the parse tree for the word of length 9.
   (d) (2.5pts) Convert your grammar into an equivalent grammar in Chomsky normal form.

9. (a) (1.5pts) State the Pumping lemma for context-free languages.
   (b) (4.5pts) Use the Pumping lemma to prove that $\{a^{2i}b^jc^{2i} \mid i \geqslant j > 0\}$ is not a context-free language.

10. (4.5pts) Consider the following grammar with start symbol S:

$$S \to CA \mid BA \mid CB \qquad A \to a \mid SA \mid BC \qquad B \to b \mid SB \mid AC \qquad C \to c \mid SC \mid AB$$

Apply the CYK algorithm to determine if the string *abccba* is generated by this grammar. Show the complete resulting table and justify your YES/NO answer.

11. (5pts) Define a Turing machine that determines if the input tape is of the form 0^i1^j for $j > i$. Give either the transition function of the machine or its transition diagram. You can assume that the initial tape only contains symbols in $\{0, 1, \square\}$. Explain how the machine works.

Solutions Exam 160817

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 $w = (ab)^i$ for $n \leqslant i \leqslant 2n-1$.

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

   Base case: $S \Rightarrow^1 w$, hence the rule applied should have been $S \to ab$.

   Here, $w = (ab)^1$ and $1 \leqslant 1 \leqslant 2-1 = 1$.

   Step case: Our IH is: if $S \Rightarrow^m w$ in $0 < m \leqslant n$ steps then $w = (ab)^i$ for $m \leqslant i \leqslant 2m-1$.

   Let $S \Rightarrow^{n+1} w$ with $n > 0$.

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

   In the case the first rule was $S \to abS$ then $w = abw'$ with $S \Rightarrow^n w'$. Then the IH applies for w' so we know that $w' = (ab)^i$ for $n \leqslant i \leqslant 2n-1$. We have then that $w = (ab)^{i+1}$ and also that $n+1 \leqslant i+1 \leqslant 2n-1+1 = 2n \leqslant 2(n+1)-1 = 2n+1$.

   In the case the first rule was $S \to SabS$ then $w = w'abw''$ with $S \Rightarrow^p w'$ and $S \Rightarrow^q w''$ and $0 < p+q = n$, hence $0 < p, q \leqslant n$. Then the IH applies for both w' and w'' so we know that $w' = (ab)^i$ and $w'' = (ab)^j$ for $p \leqslant i \leqslant 2p-1$ and $q \leqslant j \leqslant 2q-1$. We have then that $w = (ab)^i ab (ab)^j = (ab)^{i+j+1}$. Also, we have that $n = p+q \leqslant i+j \leqslant 2p-1+2q-1 = 2(p+q)-2 = 2n-2$, that is, $n \leqslant i+j \leqslant 2n-2$ and hence, $n+1 \leqslant i+j+1 \leqslant 2n-2+1 = 2n-1 \leqslant 2(n+1)-1 = 2n+1$.

2. We define the following DFA:

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

3.

	0	1	2
$\to q_0$	q_0q_1	q_2	q_2q_3
$^*q_0q_1$	$q_0q_1q_3$	q_2	q_2q_3
$^*q_2$	q_1	q_2	q_1
$^*q_2q_3$	q_1	q_2q_3	q_0q_1
$^*q_1$	q_1q_3	q_2	q_2q_3
$^*q_0q_1q_3$	$q_0q_1q_3$	q_2q_3	$q_0q_2q_3$
$^*q_1q_3$	q_1q_3	q_2q_3	$q_0q_2q_3$
$^*q_0q_2q_3$	q_0q_1	q_2q_3	$q_0q_1q_2q_3$
$^*q_0q_1q_2q_3$	$q_0q_1q_3$	q_2q_3	$q_0q_1q_2q_3$

4. We will solve equations:

$$E_0 = 0E_0 + 1E_1 + 2E_3 = 0E_0 + 1E_1 + 2$$
$$E_1 = 0E_2 + 1E_0 + 2E_3 = 0E_2 + 1E_0 + 2$$
$$E_2 = 0E_1 + 1E_2 + 2E_3 = 0E_1 + 1E_2 + 2$$
$$E_3 = \epsilon$$

$$E_0 = 0E_0 + 10E_2 + 11E_0 + 12 + 2 = (0+11)E_0 + 10E_2 + 12 + 2$$

$$E_2 = 00E_2 + 01E_0 + 02 + 1E_2 + 2 = (00+1)E_2 + 01E_0 + 02 + 2$$
$$E_2 = (00+1)^*(01E_0 + 02 + 2) = (00+1)^*01E_0 + (00+1)^*(02+2)$$

$$E_0 = (0+11)E_0 + 10(00+1)^*01E_0 + 10(00+1)^*(02+2) + 12 + 2$$
$$E_0 = (0+11+10(00+1)^*01)E_0 + 10(00+1)^*(02+2) + 12 + 2$$

Hence

$$E_0 = (0+11+10(00+1)^*01)^*(10(00+1)^*(02+2) + 12 + 2)$$

If you eliminated the states in the order you were asked to, you should get the same final expression.

5. First we eliminate q_3 because it is not reachable.

   Then we run the algorithm that identifies equivalent states which give us the following table:

	q_0	q_1	q_2	q_4	q_5	q_6
q_7	X	X	X	X	X	X
q_6	X	X	X	X		
q_5	X	X	X	X		
q_4	X	X	X			
q_2	X	X				
q_1						

   The resulting automaton is:

	0	1
$\rightarrow q_0q_1$	q_7	q_0q_1
q_2	q_4	q_5q_6
q_4	q_5q_6	q_5q_6
$^*q_5q_6$	q_5q_6	q_5q_6
q_7	q_2	q_2

6. (a) The words in $\mathcal{L}((0+01)^*)$ have no 1's (that is, empty word or just 0's) or have always at least one 0 before a 1. Hence a non-empty word starts with 0 and never has two 1's together.
   So words in $\{0,1\}^* - \mathcal{L}((0+01)^*)$ cannot be empty and must contain at least one 1. Any word starting with 1 belongs to this language. If it starts with a 0, it should contain at least two 1's together.

   (b) $(0^*1)^*1(1+0)^*$

7. (a) We know by the shifting rule that $0(10)^* = (01)^*0$ so $10(10)^*(0+11)+11 = 1(01)^*0(0+11)+11 = 1(01)^*00 + 1(01)^*011 + 11$.
   On the other hand $1(01)^*(00+1) = 1(01)^*00 + 1(01)^*1$.
   Observe that, since $R^* = \epsilon + R^*R$, $1(01)^*1 = 1(\epsilon + (01)^*01)1 = 11 + 1(01)^*011 = 1(01)^*011 + 11$.
   Hence both expressions are equal.

   (b) i. (2.25pts) YES. Let $\mathcal{L}_1 = \{\epsilon\}$ which is non empty and regular. Let $\mathcal{L}_2 = \{0^n1^n | n > 0\}$ which is non regular. We have that $\mathcal{L}_1 \cap \mathcal{L}_2 = \emptyset$.
   Now $\mathcal{L}_1 \cup \mathcal{L}_2 = \{0^n1^n | n \geqslant 0\}$ which is non regular.

   ii. (1.25pt) NO. Since $\mathcal{L}_1 \cap \mathcal{L}_2 = \emptyset$ then $\mathcal{L}_2 - \mathcal{L}_1 = \mathcal{L}_2$ and hence non regular.

8. OBS: this solution gives an ambiguous grammar! In the case the word is 00000000 it could be formed by having $u = v = 0$ (via A) and $w = 000$ (via C) or by having $u = v = 00$ (via A) and $w = 00$ (via C).

   (a)

$$\begin{aligned} &S \rightarrow AC \\ &A \rightarrow 0 \mid 1 \mid 00 \mid 11 \mid 01 \mid 10 \\ &C \rightarrow 0C0 \mid 1C1 \mid 0A0 \mid 1A1 \end{aligned}$$

(b) Since $u, v, w \in \{0,1\}^+$ then they cannot be empty.

A generates u and v: any combination of 0's and 1's of length 1 (via the 2 first productions) or of length 2 (via the last 4 productions).

Observe that there is only one way to produce a particular work of length 1 or of length 2.

C will start by generating w and its reverse and when we are done it will insert v (via A). The last 2 productions are used for generating the last symbol of w and the first of w^{rev} (and v), while the first 2 productions are used for generating all other symbols of w and w^{rev}.

Observe that there is a unique sequence of productions to be used from C which is determined by the order in which the 0's and 1's occur in w. After producing w and it reverse we finished by inserting v in the middle (via A).

(c)

(d) There are no ϵ-productions and no unit productions either.

$$\begin{array}{lll}
S \to AC & X \to 0 & Y \to 1 \\
A \to 0 \mid 1 \mid XX \mid YY \mid XY \mid YX & P \to CX & T \to AX \\
C \to XP \mid YQ \mid XT \mid YS & Q \to CY & S \to AY
\end{array}$$

9. (a) See slide 23 lecture 12.

(b) Let us assume that $\mathcal{L} = \{a^{2i}b^jc^{2i} \mid i \geqslant j > 0\}$ is context-free.

Let n be the constant provided by the Pumping lemma.

Let $w = a^{2n}b^nc^{2n}$. It is clear that $w \in \mathcal{L}$ and that $|w| \geqslant n$.

Since $w = xuyvz$ with $|uyv| \geqslant n$ we have 5 different possibilities:

i. uyv consists only of a's: then by iterating (by letting $k > 1$)/ eliminating ($k = 0$) u and v we will have more/less a's than c's;

ii. uyv consists of a's and b's: if $u \neq \epsilon$ then the relation between the a's and c's is broken for $k \neq 1$ while if $v \neq \epsilon$ then the amount of b's will be more than half the amount of c's for $k > n$ (recall that $uv \neq \epsilon$);

iii. uyv consists only of b's: by taking $k > n$ then the amount of b's will be more than half the amount of a's (and of c's);

iv. uyv consists of b's and c's: if $u \neq \epsilon$ then the amount of b's will be more than half the amount of a's for $k > n$ while if $v \neq \epsilon$ then the relation between the a's and c's is broken for $k \neq 1$;

v. uyv consists only of c's: then by iterating/eliminating u and v we will have more/less c's than a's.

In all cases the resulting word (xu^kyv^kz) will not belong to the language and hence $\mathcal{L}$ cannot be a context-free language.

10.

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

S does NOT belong to the upper-most set, which means that the word is NOT generated by the grammar since S is the starting symbol of the grammar.

11. We should check that there is (at least) one 1 for each 0. We match from left to right. Observe that we need to read at least one extra 1 in the tape, and hence, the tape cannot be empty.

    Let $\Sigma = \{0,1\}$. Let $M = (\{q_0, ..., q_5, q_f\}, \Sigma, \delta, q_0, \Box, \{q_f\})$, with δ is as follows:

$\delta(q_0, 1) = (q_1, 1, R)$	there are no 0's, weto check the rest is only 1's;
$\delta(q_0, 0) = (q_2, X, R)$	if a 0 comes we mark with X and go right searching for the corresponding 1;
$\delta(q_0, Y) = (q_5, Y, R)$	we have matched all the 0's we need to check that the end if the tape is correct;
$\delta(q_1, 1) = (q_1, 1, R)$	there might be several extra 1's;
$\delta(q_1, \Box) = (q_f, \Box, R)$	there is at least one extra one, the tape is correct and we accept;
$\delta(q_2, 0) = (q_2, 0, R)$	we run over remaining 0's;
$\delta(q_2, 1) = (q_4, Y, L)$	we find the first 1 so we mark with Y and move left to find the first unmatched 0;
$\delta(q_2, Y) = (q_3, Y, R)$	we got to the 1's that have already been matched against 0's;
$\delta(q_3, Y) = (q_3, Y, R)$	we run over all matched 1's;
$\delta(q_3, 1) = (q_4, Y, L)$	we found the first unmatched 1, so we mark with Y and move left to find the first unmatched 0;
$\delta(q_4, Y) = (q_4, Y, L)$	we move left until we find the first unmatched 0;
$\delta(q_4, 0) = (q_4, 0, L)$	we move left until we find the first unmatched 0;
$\delta(q_4, X) = (q_0, X, R)$	we found the last matched 0 so we move right to match the next 0 if it exists;
$\delta(q_5, Y) = (q_5, Y, R)$	we run over all matched 1's;
$\delta(q_5, 1) = (q_1, 1, R)$	there is at least an extra 1, we check that the rest of the tape is correct.

    Having q_2 and q_3 guarantee that all 0's come before all 1's. Since one needs to have at least an extra 1, we could actually have had only one state running over 0's and Y's.