

**Solutions to 8th Coursework**

26/04/2004

**Question 1: (Compulsory)** The following questions are multiple choice. There is at least one correct choice but there may be several. To get all the marks you have to list all the correct answers and none of the wrong ones.

Each part is marked as follows:

**No error** 5

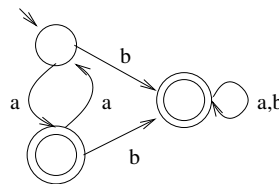
**1 error** 3

**2 errors** 1

**> 2 errors** 0

where an error is either a correct answer left out or an incorrect answer which is included.

- a. Given the following deterministic finite automaton (DFA)  $A_1$  over  $\Sigma_1 = \{a, b\}$ .



Which of the following statements about  $A_1$  are correct?

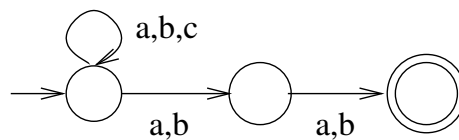
i)  $aabb \in L(A_1)$

ii)  $bbaa \in L(A_1)$

iv)  $A_1$  accepts all words with an odd number of  $a$ s or at least one  $b$ .

(5)

- b. Given the following nondeterministic finite automaton (NFA)  $A_2$  over  $\Sigma_2 = \{a, b, c\}$ .



Which of the following statements about  $A_2$  are correct?

ii)  $cba \in L(A_2)$

iv)  $A_2$  accepts all words which do not have a  $c$  in their last two symbols.

(5)

c. Which of the following is equivalent to “ $L$  is regular”, where  $L$  is a language?

**iii)**  $L$  is recognized by a computer program using only fixed finite memory.

**v)**  $L$  is recognized by a nondeterministic finite automaton (NFA).

(5)

d. Given the following context free grammar

$$G = (\{S, T\}, \{(\, , \, \), \mathbf{a}, \mathbf{b}\}, S, P)$$

with productions  $P$ :

$$S \rightarrow (T) \mid \mathbf{a}$$

$$T \rightarrow (S) \mid \mathbf{b}$$

Which of the following are in the language of  $G$  (i.e. are elements of  $L(G)$ )?

**i)**  $((\mathbf{a}))$

e. Which of the following propositions are correct?

**iv)** A Turing machine may fail to terminate.

**v)** A language is a set of words.

(5)

**Question 2**

a. Given the alphabet  $\Sigma = \{a, b, c\}$  we define the language  $L \subseteq \Sigma^*$  as:

$$L = \{a^i b^j c^k \mid i + j \equiv k \pmod{2}\}$$

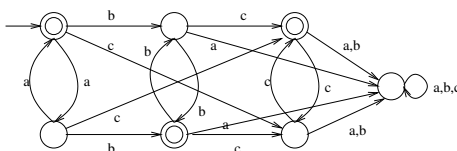
that is a word in the language is a sequence of as followed by a sequence of bs followed by a sequence of cs such that the sum of the number of as and bs is even if and only if the number of cs is even. E.g. in the language are

$$\epsilon, aa, ab, abcc, aabccc$$

and not in the language are

$$abc, aabcc, c, cb, cba$$

Devise a deterministic finite automaton (DFA) which accepts  $L$ . The automaton should be presented using a transition diagram where the initial and final states are marked as usual.



*Perfect answer is 12 points, points are subtracted for errors in the automaton such as: no error state, not deterministic, order not checked etc. Completely incomprehensible answers earn no points.*

(12)

b. Let  $\Sigma = \{a, b\}$  and let  $\#_a, \#_b \in \Sigma^* \rightarrow \mathbb{N}$  be functions which count the numbers of as and bs respectively. E.g.  $\#_a(aababa) = 4$  and  $\#_b(aababa) = 2$ . We use those to define the following languages:

$$\begin{aligned} L_1 &= \{w \mid \#_a(w) \equiv \#_b(w) \pmod{2}\} \\ L_2 &= \{w \mid \#_a(w) = \#_b(w)\} \\ L_3 &= \{w \mid \#_a(w) > 0 \iff \#_b(w) > 0\} \end{aligned}$$

(i) Which of the languages  $L_1, L_2, L_3$  are regular?

*$L_1$  and  $L_3$  are regular.*

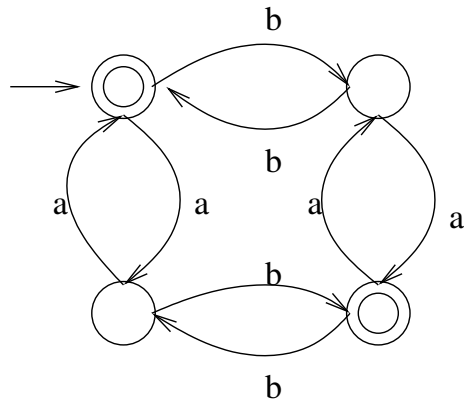
*Marking as for question 1.*

(5)

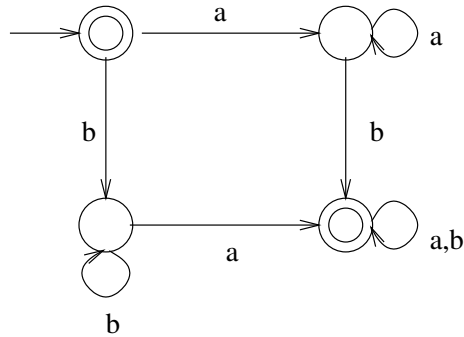
(ii) Provide evidence for your answer to i) by either

- Exhibiting a DFA, NFA or a regular expression defining the language, or

*DFA for  $L_1$ :*



DFA for  $L_3$ :



2 points for each automaton, reduced to 1 if the automaton has minor errors.

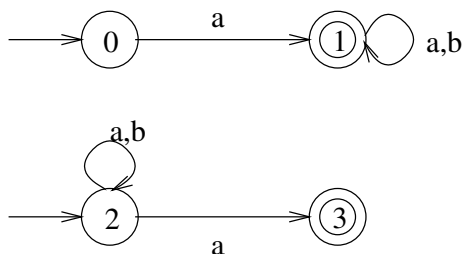
- Use the pumping lemma to show that the language in question is not regular.

Assume that  $L_2$  is regular, then there is a pumping number  $n$ . We know that  $w = a^n b^n \in L_2$  by the pumping lemma there are  $x, y, z \in \Sigma^*$  s.t.  $w = xyz$  and  $|xy| \leq n$ , hence  $y$  contains only  $a$ s and by the 2nd condition at least one. Furthermore  $xz \in L$  by pumping down but this word has fewer  $a$ s but the same number of  $b$ s, hence we have arrived at a contradiction and our assumption that  $L$  is regular must be false.

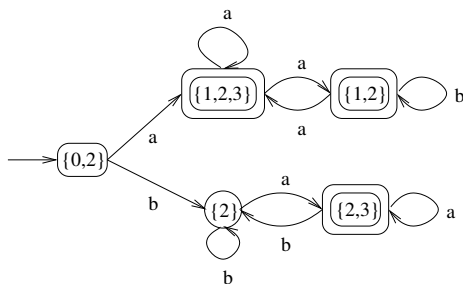
4 points, subtracting for insufficient arguments.

**Question 3**

- a. Given the following nondeterministic finite automaton  $A$  over  $\Sigma = \{a, b\}$ :



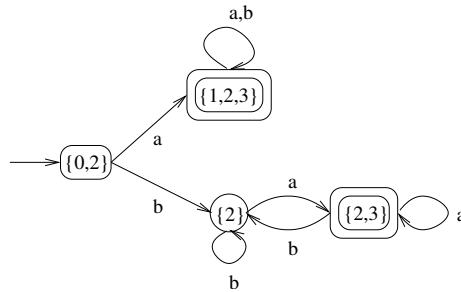
Apply the subset construction to derive a deterministic finite automaton  $D(A)$  which recognizes the same language. You should restrict yourself to the reachable part of  $D(A)$ , i.e. states which are not reachable from the initial state should be left out. (10)



*Correct answer gets 10 points, points are subtracted for minor errors in the automaton. Incomprehensible answers earn no points.*

- b. Is the automaton constructed in a. minimal? I.e. can you find a DFA which accepts the same language but uses fewer states? (5)

*The automaton is not minimal - one state can be saved:*



- c. Give regular expressions for the following languages over  $\Sigma = \{a, b, c\}$ :

- (i) Words where all **as** occur before any **bs**.

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

- (ii) Words that do contain a **c**.

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

- (iii) Words that contain an odd number of symbols.

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

- (iv) Words that end with **aa**.

*Answer:*  $(a + b + c)^*aa$

- (v) Words that do not end with **aa**.

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

(10)

*2 points for each part, minor errors 1 point.*

**Question 4**

The syntax of formulas of propositional logic over the atomic propositions  $p, q, r$  is given by the following rules:

- $p, q, r$  are formulas.
- true, false are formulas.
- If  $A$  is a formula then  $\neg A, (A)$  are formulas.
- If  $A, B$  are formulas then  $A \wedge B, A \vee B$  are formulas.

To avoid ambiguity we also introduce the following conventions:

- $\neg$  binds stronger than  $\wedge$  and  $\vee$
- $\wedge$  binds stronger than  $\vee$

a. Present a Context-free Grammar (CFG) over

$$\Sigma = \{p, q, r, (, ), \wedge, \vee, \neg, \text{true}, \text{false}\}$$

whose languages are the syntactically correct formulas. Design your grammar such that the conventions about binding strength are reflected by the parse trees. (15)

$G = (\{E, F, L\}, \Sigma, E, P)$  with  $P$  given by:

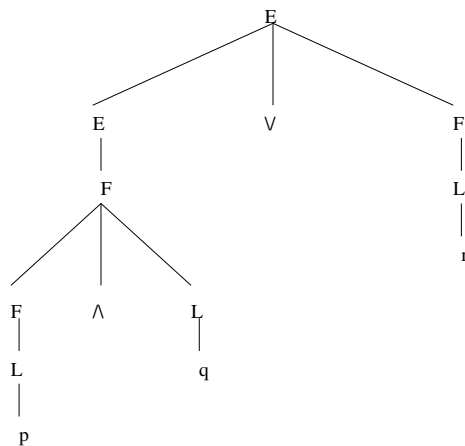
$$E \rightarrow F \mid E \vee F$$

$$F \rightarrow L \mid F \wedge L$$

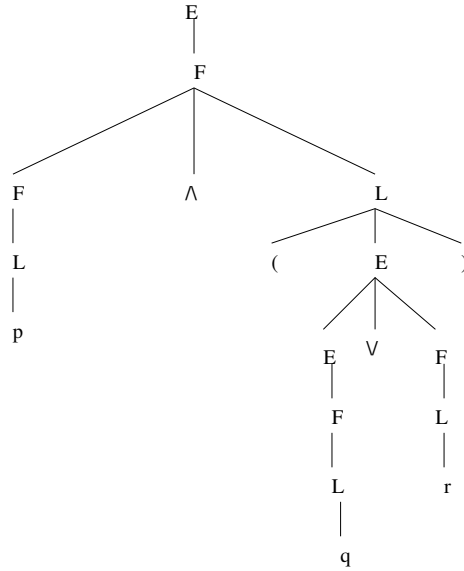
$$L \rightarrow (E) \mid p \mid q \mid r \mid \text{true} \mid \text{false} \mid \neg L$$

b. Draw parse trees for  $p \wedge q \vee r$  and  $p \wedge (q \vee r)$ . (5)

*parse tree for  $p \wedge q \vee r$*



parse tree for  $p \wedge (q \vee r)$



- c. Describe in one or two sentences: *How would you go about implementing a parser for this language?* (5)

**Alternative I** *The given answer is not LL(1) but it can be transformed to an LL(1) grammar which gives rise to a recursive descent parser*

**Alternative II** *Use an LALR parser generator like YACC.*



**Question 5** Given the following PDA  $P$

$$P = (Q = \{q_0, q_1\}, \Sigma = \{a, b\}, \Gamma = \{a, b, \#\}, \delta, q_0, Z_0 = \#, F = \{q_0\})$$

where  $\delta$  is given by the following equations:

$$\begin{aligned} \delta(q_0, a, \#) &= \{(q_1, a\#)\} \\ \delta(q_0, b, \#) &= \{(q_1, b\#)\} \\ \delta(q_1, a, b) &= \{(q_1, \epsilon)\} \\ \delta(q_1, a, a) &= \{(q_1, aa)\} \\ \delta(q_1, b, a) &= \{(q_1, \epsilon)\} \\ \delta(q_1, b, b) &= \{(q_1, bb)\} \\ \delta(q_1, \epsilon, \#) &= \{(q_0, \#)\} \\ \delta(q, w, z) &= \{\} \quad \text{everywhere else} \end{aligned}$$

- a. Construct a sequence of Instantaneous Descriptions (IDs) for the words  $bba, baab, \epsilon$ . Which of those are accepted (using acceptance by final state)?

(12)

**bba**

$$\begin{aligned} (q_0, bba, \#) &\vdash (q_1, ba, b\#) \\ &\vdash (q_1, a, bb\#) \\ &\vdash (q_1, \epsilon, b\#) \end{aligned}$$

The word is not accepted.

**baab**

$$\begin{aligned} (q_0, baab, \#) &\vdash (q_1, aab, b\#) \\ &\vdash (q_1, ab, \#) \\ &\vdash (q_0, ab, \#) \\ &\vdash (q_1, b, a\#) \\ &\vdash (q_1, \epsilon, \#) \\ &\vdash (q_0, \epsilon, \#) \end{aligned}$$

The word is accepted.

**$\epsilon$**

$$(q_0, \epsilon, \#)$$

The word is accepted.

*Each part counts for 4 points.*

- b. Describe the language accepted by  $P$  in one sentence. (5)  
*The set of words with the same number of as and bs*

c. What does it mean for a PDA to be deterministic? Is  $P$  deterministic? (8)

*A PDA is deterministic if there is never a choice between two possible IDs. More formally*

$$|\delta(q, x, z)| + |\delta(q, \epsilon, z)| \leq 1$$

*Indeed,  $P$  is deterministic.*

### Question 6

Write a short essay addressing each of the questions below. Try to be as concise as possible.

- a. What is a Turing machine (TM)? Given an informal description. (5)  
*A TM consists of a tape and a finite state machine. Based on the symbol on the tape and the state the TM writes a new symbol on the current position, changes its states and moves left or right on the tape. The TM may stop and either accept or reject or run forever (which is interpreted as rejection)*
- b. What is the difference between:
- A TM *accepts* a language, and
  - A TM *decides* a language.

(5)

*The language accepted by a TM is the set of words on which the TM stops in an accepting state. A TM decides this language if it always stops, i.e. it rejects word by stopping in a non-accepting state.*

- c. What is the *halting problem*? Can it be considered as a language? (5)  
*The halting problem is the set of pairs of TMs and inputs such that the TM stops on that input. It can be considered as a language once we assume an appropriate encoding of TMs and their inputs.*
- d. How do we show that the *halting problem* is undecidable? (10)  
*Assume  $H(m, x)$  would be a TM which decides the Halting problem. We construct a new TM  $F(m)$*

$$F(m) = \text{if } H(m, m) \text{ then loop else stop}$$

*What happens to  $F(\lceil F \rceil)$  (where  $\lceil F \rceil$  is the code for  $F$ ) ? It stops iff it doesn't stop which is a contradiction, hence our assumption that the halting problem was decidable is wrong.*