# EXAMPLE EXAMINATION PAPER and SOLUTIONS

## QUESTION 1: Languages [10]

(a) Let S = {*a*, *bb*, *bab*, *abaab*}. For each of the following strings, state whether or not it is a word in S*:

(i) *abbabaabab*
(ii) *abaabbabbbaabb* (2)

(i)

(ii)

(b) Give an example of a set S such that S* only contains <u>all possible</u> strings of combinations of *a*'s and *b*'s that have length divisible by three. (4)

(c) Give an example of two sets S and T of strings such that
S* = T* but S $\not\subset$ T and T $\not\subset$ S. (4)

## QUESTION 2: Regular expressions [10]

(a) Does the regular expression

[**b***+ (**abb***)* + (**aabb***)*]* **bbb** [**b*** + (**abb***)* + (**aabb***)*]*

define the language of *all* words in which the substring *bbb* appears at least once, but the substring *aaa* does not appear at all? If not, give a counter-example. (5)

(b) Give a regular expression generating the language consisting of all words containing exactly one occurrence of the substring *aa* and no occurrence of the substring *bb*. (5)

**QUESTION 3: Recursive definitions** **[10]**

A recursive definition for the language ODDAB should be compiled. Consider the alphabet ∑ = {*a*, *b*} and the language ODDAB, where ODDAB consists of all words of **odd** length that **contain the substring** ***ab***. Provide

(i) an appropriate universal set, (1)
(ii) the generator(s) of ODDAB, (2)
(iii) an appropriate function on the universal set, and then (1)
(iv) use these concepts to write down a recursive definition for the language ODDAB. (6)

(i)

(ii)

(iii)

(iv)

**QUESTION 4: Mathematical induction** **[10]**

(i) Give a recursive definition of the set P of all positive integers greater than or equal to 5, (1)
(ii) formulate an appropriate induction principle, and (2)
(iii) then apply the induction principle to prove that
$2n - 3 \leq 2^{n-2}$ for all integers $n \geq 5$. (7)

| |
|---|
| (i) P is the<br><br>(ii) If a subset A of P is such that<br><br>(iii) Define $A \subseteq P$ as follows:<br><br>Show for n =<br><br>Assume<br><br>Required to prove<br><br>LHS<br>=<br><br>Thus |

**QUESTION 5: Finite automata** **[10]**

Build an FA (finite automaton) that accepts the language of all words that satisfies both of the following conditions:

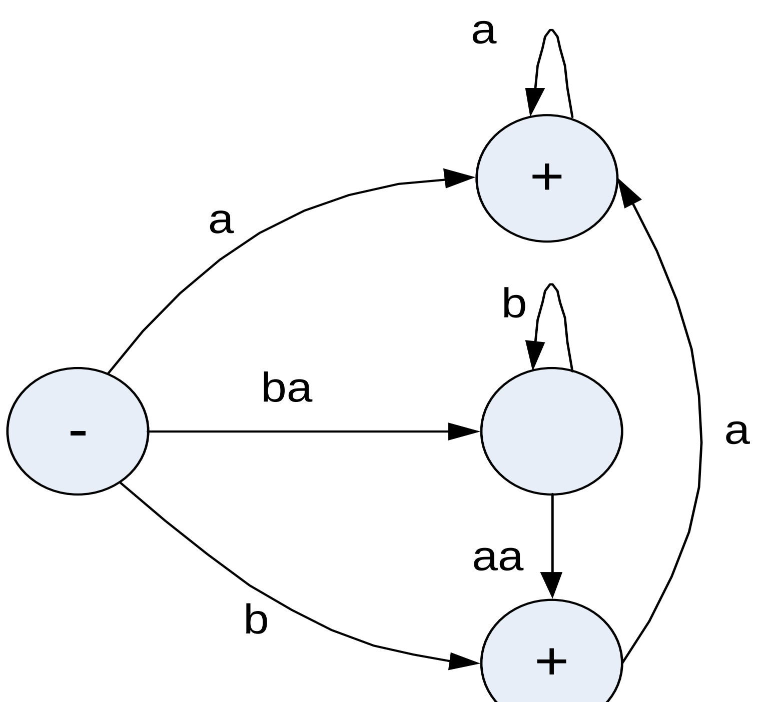
- NO word contains the substring *bba*, and
- ALL words end with a double letter, thus all words end with either *aa* or *bb*.

Note: Only one FA must be build.

NOTE: If you provide a nondeterministic finite automata (NFA) or transition graph (TG) for (c), the maximum mark you may be awarded is 2. Ensure that you build a deterministic FA.

**QUESTION 6: Kleene's theorem (TG to RE)** **[10]**

By using Kleene's theorem, find a regular expression that generates the language accepted by the following TG (transition graph):



Step 1 - Create a unique start state and a unique final state:

Step 2 - Eliminate state 3:

Step 3 - Eliminate state 4:

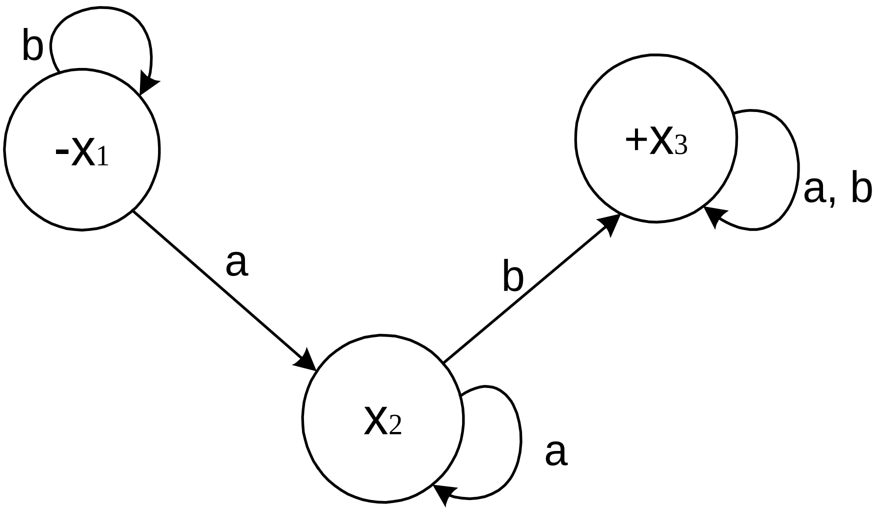
Step 4 - Eliminate state 2:

Step 5 – Eliminate state 1:

A possible regular expression is:

**QUESTION 7: Kleene's theorem (RE to FA)** **[10]**

Consider the following FA with the corresponding regular expression $r_1$:

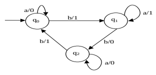



Build an FA for the regular expression $r_1^*$ by applying Kleene's theorem. (Do not formulate any regular expression.)

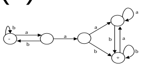
| Use the table below to find your solution. Some z states have been provided for you, and they are not an indication of how many you will need. Remember to indicate start and final state(s). | | | |
|---|---|---|---|
| New state | *a* | *b* | |
| $z_1$ = | | | |
| $z_2$ = | | | |
| $z_3$ = | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| Now draw the new FA for $\mathbf{r_1}^*$ | | | |

## QUESTION 8: Regular language acceptors [10]

**(a)** Convert the following Mealy machine to a Moore machine: (5)




(b) Consider the following FA:




Find an NFA (non-deterministic FA) with four states that accepts the same language. (5)

## QUESTION 9: Pumping lemma with length [10]

Use the Pumping Lemma **with** length to prove that the following language is nonregular:

L = {$a^nb^na^m$, where $n \in \{0, 1, 2, 3, ...\}$ and $m \in \{0, 1, 2, 3, ...\}$}.

Use the prompts below to complete the proof.

Assume

Then there exists

We choose any word *w* =

Thus *w* may be written as

Then according to the pumping lemma with length,

There is/are __ possible choice(s) for *y*:

If *y* is pumped in each of the above case(s)

We conclude that

And,

We conclude that L is not regular.

## QUESTION 10: Decidability [10]

Assume we have an alphabet ∑ = {*a*, *b*}. Complete the following table:

| | FA | TG | NFA |
|---|---|---|---|
| Number of start states | | | |
| Number of final states | | | |
| Permissible edge labels | | | |
| Number of lines leaving each state | | | |
| Deterministic or not | | | |

**ADDITIONAL QUESTION 10: DECIDABILITY** **[10]**

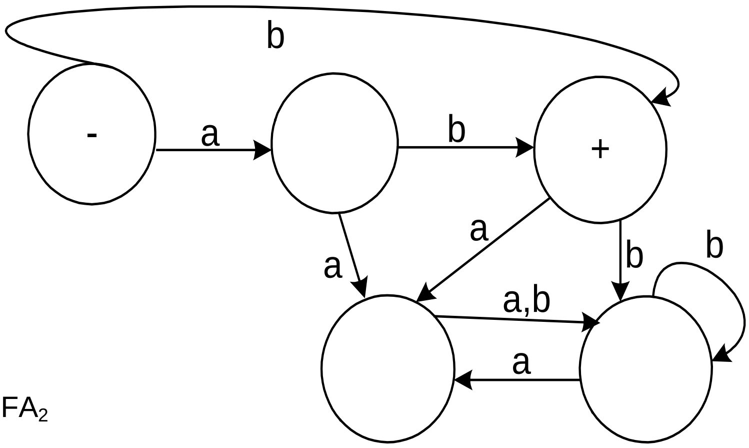
(a) Define a decision procedure.

(b) Use the definition in (a) to define decidability.

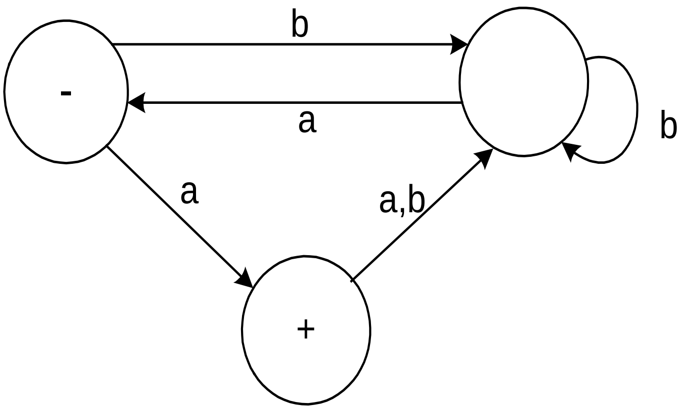
(c) Describe an effective procedure to decide whether a given FA accepts a finite or infinite language.

(d) Using the decision procedure described in (c) above, determine for each $FA_1$ and $FA_2$ below whether it accepts a finite or an infinite language.

$FA_1$




$FA_2$



In the case of $FA_1$:

In the case of $FA_2$: