This take-home test is worth 10% of your final grade. It is due at 11 a.m. on Friday, 18th October. All work is to be done individually.

You are expected to have read and understood the University’s guidelines on academic conduct. In accordance with this policy, you may discuss with other students the general principles required to understand this test, but the work you submit must be the result of your own effort.

**Submission:**
Completed tests must be submitted before the submission deadline above. Tests can **EITHER** be handed in prior to the start of the lecture **OR** submitted electronically via cssubmit. If submitted electronically, tests should be submitted as a single PDF file with A4 paper size. All answers must be clear and legible.

**Late submission penalties:**
You must submit your completed test before the submission deadline above. The penalties for late submission are described in the University’s guidelines on assessment. For late submissions a penalty of 10% of the total mark allocated to the assessment item must be deducted per day for the first 7 days (including weekends and public holidays). After 7 days the assigned work will not be accepted and will receive a mark of zero (unless an application for mitigation is approved).

<table>
<thead>
<tr>
<th>Name</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Student Number</td>
<td></td>
</tr>
</tbody>
</table>

**SOLUTION:**
Model answers, marking scheme and general feedback.

October 29, 2019
1. (a) **Deterministic FSMs** are comprised of a 5-tuple. For the deterministic FSM depicted by the following diagram, describe each of the items in the relevant tuple.

![ FSM Diagram ]

(b) Give an English-language description of the language recognised by the FSM.

**SOLUTION:**

(a) The given FSM consists of the 5-tuple \((Q, Q_0, \Sigma, F, f)\) with

- (the set of all states) \(Q = \{ q_1, q_2, q_3 \} \)
- (the starting state) \(Q_0 = q_1\)
- (the alphabet) \(\Sigma = \{ a, b \} \)
- (the set of all accepting states) \(F = \{ q_1, q_3 \} \)
- (the transition function) \(f: Q \times \Sigma \rightarrow Q\), which is defined by the following table:

\[
\begin{array}{c|cc}
f & a & b \\
\hline
q_1 & q_2 & q_1 \\
q_2 & q_3 & q_2 \\
q_3 & X & q_3 \\
\end{array}
\]

(b) The FSM accepts all input strings over the alphabet \(\Sigma\) that consist of any number of b’s (including the empty string) and includes either none or exactly two a’s.
2. (a) For each of the following languages over the alphabet $\Sigma = \{a, b, c\}$ specified by the regular expressions (i)–(iii), provide two strings in $\Sigma^*$ that are members and two strings in $\Sigma^*$ that are not members of the language (four strings each).

i. $aa + c$

ii. $((ca)^* + a)b$

iii. $(b + c + abc)^*(a + c)$

Write your answers in the spaces provided in the table below.

<table>
<thead>
<tr>
<th>Regular expression</th>
<th>Two strings in language</th>
<th>Two strings not in language</th>
</tr>
</thead>
<tbody>
<tr>
<td>i.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ii.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>iii.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Question 2 continues on next page
(b) Assume the alphabet $\Sigma = \{0, 1\}$. Give three different regular expressions (besides the one given) that specify the language described by this regular expression:

$$(10 + 0)^* + ((10)^*0^*)^* + (0 + \epsilon)^*$$

In each case, explain why your regular expression specifies the same language.

**Note:** For the purposes of this exercise only, changing the order of a union does not count as a different regular expression. Your examples also should not be more complicated than the original regular expression.

**Solution:**

(a) A few member and non-member examples are given. In most cases there are more members or non-members than the ones given below:

i. members: $aa$, $c$ (These are the only two members in the language.)
   non-members: $b$, $aac$, $aaa$, $bbb$, ...

ii. members: $cacab$, $ab$, $cab$, $cacab$, ...
   non-members: $caca$, $caab$, $aaab$, ...

iii. members: $bcbcbbbabcbbcebbabcb$, $abca$, $cbba$, $abcc$, ...
    non-members: $abc$, $bacb$, $ccabcecbcb$, ...

(b) The regular expressions $(10 + 0)^*$ and $((10)^*0^*)^*$ describe the same language, so it is sufficient to use only one of them to describe their union. The language described by the regular expression $(0 + \epsilon)^*$ is the same as $0^*$ as this already contains the empty string. Furthermore, the set of all strings in $0^*$ is a subset of $(10 + 0)^*$ (or $((10)^*0^*)^*$), so it does not need to be added. Therefore, equivalent regular expressions for the language specified above are:

- $(10 + 0)^*$
- $((10)^*0^*)^*$
- $(10 + 0)^* + 0^*$
- $(10 + 0)^* + ((10)^*0^*)^*$
- $((10)^*0^*)^* + 0^*$
- $(10 + 0)^* + ((10)^*0^*)^* + 0^*$
3. State, for each of the following FSMs, whether it is a deterministic or non-deterministic FSM, explaining your answer. Give a regular expression over the alphabet $\Sigma = \{a, b, c\}$ for each of them.

(a) $M_1$:

(b) $M_2$:

SOLUTION:

(a) $M_1$ is a non-deterministic FSM as for example there are two possible transitions given state $s_1$ and input $c$. One could either move to state $s_2$ or stay in state $s_1$. Similarly, this is true for state $s_2$ and input $a$.

Possible regular expressions for the language recognised by $M_1$ are:

- $(a + b + c)^* c (a^*a + a^*b)$
- $(a + b + c)^* c a^* (b + a)$

(b) $M_2$ is a deterministic FSM as there is only one possible transition from each state for a particular input.

Possible regular expressions for the language recognised by $M_2$ are:

- $(b + c)^* (a (b + c)^* a (b + c)^*)^*$
- $((b + c)^* a (b + c)^* a)^* (b + c)^*$
- $((b + c)^* (a (b + c)^* a)^*)^*$
- $((b + c) + (a (b + c) a))^*$
4. A group of engineering students discovered that they all had something in common: their love of pizza. They were not happy with available options on campus for obtaining pizza, so they decided to build a very simple pizza machine on their own. The machine is able to make simple pizzas (one size only), which all have a standard pizza sauce, one topping and cheese on top included for only $5 each. The students added their favourite three toppings: mushrooms, pepperoni, and ham. The machine has five buttons and one slot to insert money: One button for each of the toppings, one Cancel button to cancel the process and get your money back and one Bake button to start baking the pizza.

Draw a diagram of a deterministic FSM with three states for the pizza machine over the alphabet \( \Sigma = \{ \text{money, cancel, bake, pepperoni, mushrooms, ham} \} \). The only accepted inputs for the machine are described as follows:

- You can leave the machine alone and do not do anything with it. (no input)
- You can insert money and then decide not to continue, pressing “CANCEL” to cancel the process.
- You can insert money, select one of the topping options and then decide not to continue, pressing “CANCEL” to cancel the process.
- You can insert money, select one of the toppings and then press the “BAKE” button. (You cannot cancel the process after pushing the bake button.)

Use three states only, and do not include error states.

\[ \text{SOLUTION: The FSM has the following diagram:} \]

```
\begin{tikzpicture}
  \node[state] (s1) at (0,0) {$s_1$};
  \node[state] (s2) at (1,1) {$s_2$};
  \node[state] (s3) at (0.5,0.5) {$s_3$};

  \draw[->] (s1) -- node[above] {money} (s2);
  \draw[->] (s1) -- node[below] {cancel} (s3);
  \draw[->] (s2) -- node[below] {cancel, bake} (s3);
  \draw[->] (s3) -- node[above] {mushrooms, pepperoni, ham} (s2);
\end{tikzpicture}
```
5. Given the following deterministic FSM $M$ over the alphabet $\Sigma = \{0, 1\}$:

![ FSM Diagram ]

(a) Give an English language description of $L(M)$, the language recognised by $M$.

(b) Add an error state (labelled $X$) to the diagram, and draw all transitions to it.

(c) Describe how to derive an FSM that accepts the complement of $L(M)$ over the set $\Sigma^*$. (That is, an FSM that accepts the language $\Sigma^* - L(M)$.)

(d) Give a regular expression for the complement of $L(M)$.

SOLUTION:

(a) The finite state machine $M$ accepts all binary input strings that do not contain three (or more) consecutive 0s.

(b) There is only one error state to add. The state $s_3$ is the only state that does not have outgoing arrows for both elements of the alphabet $\Sigma = \{0, 1\}$, so we need to add the error state here. The diagram including the error is the following:

![ FSM Diagram with Error State ]
(c) In order to create an FSM diagram that accepts all input strings in the complement of $L(M)$ one considers the diagram that includes all error states. One then turns each accepting state into a non-accepting state and the other way around. Hence, in this example the former error state becomes the only accepting state.

(The description without the diagram is enough.)

(d) One possible solution is: $(0 + 1)^* 000 (0 + 1)^*$ (The complement of the language recognised by $M$ contains all binary strings that contain at least three consecutive 0s. So, the regular expression that specifies the complement can be given as $(0 + 1)^* 000 (0 + 1)^*$.