1 Finite State Automata and Regular Language

Consider the non-deterministic finite automaton $A$:

\[
\begin{align*}
S_0 & \xrightarrow{a} S_1 & & b & \xrightarrow{} S_2 \\
\{a, b, c\} & & \{a, b, c\}
\end{align*}
\]

$A$ is intended to recognise the language

\[L = \{\gamma ab \delta | \gamma, \delta \in \{a, b, c\}^*\}\]

of strings over the alphabet $\{a, b, c\}$ that contain the substring $ab$.

(i) Use the 'subset construction' algorithm given in lectures to produce a deterministic finite automaton $B$ which recognises the same language as $A$.  

[2 marks]

Be clear on which states of $B$ represent which subsets of states of $A$.

(ii) Are any of the states of the automaton $B$ equivalent to each other? Answer this question using the algorithm given in lectures.  

[2 marks]

Be careful to explain how you know that the algorithm has terminated, so that you have discovered all possible groups of equivalent states.
(iii) If you discovered any groups of states that are equivalent, use this and the procedure outlined in lectures to give a DFA $C$ that recognises the same language as $B$ but is **minimal**.  

[2 marks]

(iv) Prove that the DFA $C$ recognises the language $L$.  

[5 marks]

Ensure that you clearly state your two main proof obligations. Make sure that you give your proof in full rigorous detail. For example, be explicit about any use of the append theorem.

(v) Using the NFA $A$, derive a **right-linear grammar** using the algorithm given in the lectures.  

[2 marks]

(vi) Consider the language 

$$M = \{ a^m b^n \mid 2m > n > m, \text{ where } m, n \in \mathbb{N} \}$$

Prove that $M$ is **not** regular, i.e., it cannot be recognised by any DFA.  

[5 marks]

Make sure that you give your proof in full rigorous detail. Do not use pumping lemma.

### 2 Pushdown Automata and Context Free Language

(i) Give a **context free** grammar $H$ that generates the language $M$ in the previous question.  

[4 marks]

**Hint**: $m$ can’t be 0, because in that case $2m = m$. $m$ can’t be 1, because in that case $2 > n > 1$, such a natural number $n$ doesn’t exist. So the shortest string in the language $M$ is $aabb$. For longer strings, you need to make sure that the number $n$ of $b$s and the number $m$ of $a$s satisfy $2m > n > m$.

(ii) Using the grammar $H$ you defined above, draw a parse tree for the string $a_4 b_6$, i.e., $aaaabbbb$.  

[2 marks]

(iii) Consider the following context free grammar in the lectures:

$$S \rightarrow T \mid W$$

$$T \rightarrow UV$$

$$U \rightarrow aUb \mid \epsilon$$

$$V \rightarrow cV \mid \epsilon$$

$$W \rightarrow XY$$

$$X \rightarrow aX \mid \epsilon$$

$$Y \rightarrow bYc \mid \epsilon$$

Convert this grammar to a (non-deterministic) PDA.  

[4 marks]

(iv) Give a PDA trace for processing the string $abbcc$ using the PDA you constructed in the previous question. State whether the string is accepted or rejected.  

[3 marks]
3 Turing Machine and Computability

(i) Design a Turing machine that accepts strings over \( \{0, 1\} \) containing even number of 1s.

[5 marks]

Hint: use a state to represent “you have read odd number of 1s”, and use another state to represent “you have read even number of 1s”.

(ii) Recall that every Turing machine can be encoded as a binary string; and each binary string can be seen as a Turing machine, if it is not a valid encoding, then we think of that string as a Turing machine that accepts nothing. Also recall that the diagonal language \( L_{di} \) is not recursively enumerable. Use a similar argument to show that the following language is not recursively enumerable either:

\[
\{w_i \mid w_i \notin L(M_{2i})\}
\]

where \( L(M_{2i}) \) is the language accepted by the Turing machine \( M_{2i} \).

[4 marks]

Note that \( w_i \) is a string of binary number \( i \) (so essentially \( w_i \) and \( i \) are the same thing), and \( M_i \) is a Turing machine with an encoding \( w_i \).

Hint: by the encoding method given in lectures, each valid Turing machine ends with at least one 0. So a Turing machine \( M_j \)’s encoding \( j \) must be an even binary number, which means there must be some number \( i \) such that \( j = 2i \). Now consider whether \( w_i \) is accepted by \( M_{2i} \) or not. You’d need to prove by contradiction.