1 Regular Languages

Consider the non-deterministic finite automaton \( A \):

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

\( A \) is intended to recognise the language \( L \) of strings over \( \{a, b\}^* \) where every occurrence of \( a \) is preceded by and followed by an occurrence of \( b \).

(i) Use the ‘subset construction’ algorithm given in lectures to produce an deterministic finite automaton \( B \) which recognises the same language as \( A \).

Be clear on which states of \( B \) represent which subsets of states of \( A \). \([2 \text{ marks}]\)

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

Be careful to explain how you know that the algorithm has terminated, so that you have discovered all possible groups of equivalent states. \([2 \text{ marks}]\)

(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. \([1 \text{ mark}]\)
(iv) Prove that the DFA \( C \) recognises the intended language \( L \). Ensure that you clearly state your two main proof obligations and prove them separately, and give your proofs in full rigorous detail. \[5 \text{ marks}\]

**Hint 1:** both proofs will be easier if you use the proof by contrapositive technique, where instead of proving the subgoal \( P \Rightarrow Q \), you prove the equivalent \( \neg Q \Rightarrow \neg P \).

**Hint 2:** Think about what all the different cases are in which a string would not be in \( L \).

(v) Using the procedure given in lectures, convert the original NFA \( A \) into a right-linear grammar. \[1 \text{ mark}\]

(vi) Let \( K \) be the language of strings over \( \{a, b\}^* \) where every occurrence of \( a \) is preceded by or followed by (or both) an occurrence of \( b \). Design a NFA that accepts \( K \). You do not need to prove that it is correct. \[2 \text{ marks}\]

2 Context-Free Languages

Consider the language

\[ M = \{ a^m b^n \mid 2m > n > m \} \]

(i) Prove that \( M \) is not regular, i.e. that it is not recognised by any deterministic finite automaton. Make sure that you give your proof in full rigorous detail. \[4 \text{ marks}\]

(ii) Give a context-free grammar \( H \) that generates \( M \). \[2.5 \text{ marks}\]

(iii) Using the grammar \( H \) you defined above, draw a parse tree for the string \( a^4 b^6 \), i.e. \( aaaaabbbbb \). \[0.5 \text{ marks}\]