1 Regular Languages

The NFA $A$:

(i) The DFA $B$:

(ii) Our initial split differentiates non-final and final states:

$$[[S, S_2], [S_0, S_1, S_{12}]]$$

Testing the non-final group with $b$ produces a split, as for $S$ we stay in the non-final group, while for $S_2$ we go to the final group:

$$[[S], [S_2], [S_0, S_1, S_{12}]]$$
Testing the final group with $a$ produces another split, as for $S_0$ we go to the group $[S]$, while for $S_1$ and $S_{12}$ we go to the group $[S_2]$: 

$[S], [S_2], [S_0], [S_1, S_{12}]$ 

The only non-singleton group left, $[S_1, S_{12}]$, cannot be split by any test: on $a$ we go to $[S_2]$, while on $b$ we stay in $[S_1, S_{12}]$. The algorithm therefore terminates, with $S_1$ and $S_{12}$ the only equivalent states.

(iii) We convert $B$ above into the minimal $C$ by deleting the state $S_{12}$ and re-assigning the arc from $S_1$ to $S_{12}$ to be a loop on $S_1$:

(iv) Note that a string is not in the language $L$ if and only if

- it has $a$ as its first letter, or
- it has $a$ as its last letter, or
- it contains the substring $aa$.

Subgoal one: If a string $w$ is in $L$, then $C$ accepts $w$.

Subgoal two: If $C$ accepts a string $w$, then $w \in L$.

As suggested by the hint, we reformulate both these goals for proof by contrapositive:

Subgoal one (contrapositive): If $C$ rejects a string $w$, then $w \notin L$.

Subgoal two (contrapositive): If $w \notin L$, then $C$ rejects $w$.

Proof of subgoal one (contrapositive): If $C$ rejects $w$ then $N^*(S_0, w) = S_2$ or $N^*(S_0, w) = S$. We consider each case separately.

$N^*(S_0, w) = S_2$: the only arc into $S_2$ is labelled by $a$, so $w$ must have the form $\alpha a$ for some string $\alpha$. But any string ending in $a$ is not in $L$.

$N^*(S_0, w) = S$: because $S$ is not the start state, the DFA must have transitioned into $S$ from other state at some stage. The only arcs into $S$ from other states are labelled by $a$, so $w$ must have the form $\alpha a \beta$ for some strings $\alpha, \beta$. There are two subcases then to consider, depending on which incoming arc was used: $N^*(S_0, \alpha) = S_0$ or $N^*(S_0, \alpha) = S_2$. We consider each subcase separately.
\[ N^*(S_0, \alpha) = S_0 \]: there are no arcs into \( S_0 \) except the ‘start state’ arc, so the only way to validate this equation is if \( \alpha \) is the empty string \( \epsilon \). But then \( w = \alpha a \beta = a \beta \), and any string starting with \( a \) is not in \( L \).

\[ N^*(S_0, \alpha) = S_2 \]: as we argued above, this can only hold if \( \alpha = \gamma a \) for some string \( \gamma \). But then \( w = \alpha a \beta = \gamma aa \beta \), and any string \( w \) containing a substring \( aa \) is not in \( L \).

**Proof of subgoal two (contrapositive):**

First, observe the lemma

\[ N^*(S, w) = S \]

(Lemma)

for all strings \( w \). This holds because all arcs out of \( S \) go back to \( S \), so there is ‘no escape’.

Suppose \( w \notin L \). There are, as noted above, three cases by which this could be, and we consider each separately.

\( w \) starts with \( a \):

\[
N^*(S_0, w) = N^*(S_0, a\alpha) \\
= N^*(N(S_0, a), \alpha) \\
= N^*(S, \alpha) \\
= S
\]

(Lemma)

and \( S \) is a non-final state, so \( w \) is rejected.

\( w \) ends with \( a \): here \( w = \alpha a \) for some \( \alpha \). Now \( N^*(S_0, \alpha a) = N(N^*(S_0, \alpha), a) \) by the corollary to the Append Theorem. We have no idea what \( N^*(S_0, \alpha) \) is, so we consider all four possible cases:

\[
N^*(S_0, a) = S \\
N^*(S_1, a) = S_2 \\
N^*(S_2, a) = S \\
N^*(S, a) = S
\]

\( S \) and \( S_2 \) are both non-final, so \( w \) is rejected.

\( w \) contains substring \( aa \): here \( w = \alpha aa \beta \) for some \( \alpha, \beta \). Now \( N^*(S_0, \alpha aa \beta) = N^*(N^*(S_0, \alpha), aa \beta) = N^*(N^*(S_0, \alpha), aa), \beta \) by the Append Theorem. We do not know what \( N^*(S_0, \alpha) \) is, so we investigate what \( N^*(N^*(S_0, \alpha), aa) \) might be by all four cases again:

\[
N^*(S_0, aa) = S \\
N^*(S_1, aa) = S \\
N^*(S_2, aa) = S \\
N^*(S, aa) = S
\]

So \( S \) is the answer no matter what. Now

\[
N^*(S_0, w) = N^*(N^*(S_0, \alpha), \beta) \\
= N^*(S, \beta) \\
= S
\]

(Lemma)

\( S \) is non-final, so \( w \) is rejected.

\( v \)

\[
S_0 \to bS_1 | \epsilon \\
S_1 \to bS_1 | aS_2 | bS_2 | \epsilon \\
S_2 \to bS_1
\]

where \( S_0 \) is the start symbol.
The automaton below will be at state $S_2$ if it has just read $b$; it will be at state $S_1$ if it has just read $a$ without $b$ to its left (in this case we will reject unless the next token is $b$); it will be at $S_0$ if the string has just started, or we have just read $ba$ (in this case reading an $a$ next is legal, but will put us ‘in danger’ of rejection by transitioning to $S_1$).

2 Context-Free Languages

(i) Suppose for contradiction that there exists a DFA $D$ that recognises $M$.

There are infinitely many strings of the form

$$a^0, a^1, a^2, \ldots$$

but only finitely many states, so by the pigeonhole principle there must be some state $S$ such that $S = N^*(S_0, a^i)$ for infinitely many $i$ (where $S_0$ is the start state). In particular, pick some $j \geq 2$ and $k \geq j + 1$ with $S = N^*(S_0, a^j) = N^*(S_0, a^k)$.

Now $2j > j + 1 > j$ (because $j \geq 2$), so $a^j b^{j+1} \in M$ and will be accepted by $D$, i.e. $N^*(S_0, a^j b^{j+1}) \in F$, where $F$ is $D$’s set of final states.

By the Append Theorem $N^*(S_0, a^j b^{j+1}) = N^*(N^*(S_0, a^j), b^{j+1})$, so because $N^*(S_0, a^j) = N^*(S_0, a^k)$ we have $N^*(N^*(S_0, a^k), b^{j+1}) \in F$. By the Append Theorem again we have $N^*(S_0, a^k b^{j+1}) \in F$. But $k \geq j + 1$, so it is not the case that $2k > j + 1 > k$ (as $k$ is too big!), so $a^k b^{j+1} \notin M$ and should not have been accepted, contradicting the existence of the DFA $D$ accepting $M$.

(ii) It is impossible for the number of $a$s to be less than two, so the grammar below starts off by adding two $a$s and three $b$s, then proceeds by adding either one or two $b$s for each $a$ added.

$$S \rightarrow aaTbbb$$

$$T \rightarrow \epsilon \mid aTb \mid aTbb$$

where $S$ is the start symbol.

This is not the only correct answer – most context-free languages are definable via many different context-free grammars, and the language $M$ is no exception to this. Of note, the grammar above is ambiguous (see below); designing an unambiguous grammar for $M$ is an worthwhile exercise.

(iii) For the grammar above either of these trees are correct: