1 Reminder/Review material

Definition 1. A non-deterministic finite automaton (NFA) is a tuple \( A = (S, \Sigma, \rightarrow, \text{Init, Final}) \), where \( S \) is a finite set of states, \( \Sigma \) is the alphabet, \( \rightarrow \) is a transition function, \( \text{Init} \) is the set of initial states (subset of \( S \)) and \( \text{Final} \) is the set of final states (subset of \( S \)). Moreover the language accepted/recognized by a NFA \( A \) is denoted \( \mathcal{L}(A) \).

Example 1. The following NFA, \( A_1 \), recognizes the regular set \((a + b)^*b(a + b)\).

\[
\begin{array}{c}
\text{start} \\
\node(s0)[state, initial, accepting] {s_0} \\
\node(s1) [state, below of=s0] {s_1} \\
\node(s2) [state, right of=s1] {s_2} \\
\end{array}
\]

\[
\begin{array}{c}
a \\
\node(s0)[state, initial, accepting] {s_0} \\
\node(s1) [state, below of=s0] {s_1} \\
\node(s2) [state, right of=s1] {s_2} \\
\end{array}
\]

Fact 1. Given a NFA \( A \) over the alphabet \( \Sigma \), the set of finite words accepted by \( A \) is a regular set over \( \Sigma \).

Fact 2. Given a regular set \( X \) over \( \Sigma \), we can construct a NFA \( A \) which recognizes/accepts the set \( X \).

Fact 3. Given a non-deterministic finite automaton \( A \), we can construct a deterministic automaton \( B \) equivalent to \( A \).

Example 2. The following NFA, \( A_2 \), recognizes the regular set \((a + b)b(a + b)^*\).

\[
\begin{array}{c}
\text{start} \\
\node(t0)[state, initial, accepting] {t_0} \\
\node(t1) [state, below of=t0] {t_1} \\
\node(t2) [state, right of=t1] {t_2} \\
\end{array}
\]

Example 3. The product \( A_1 \otimes A_2 \) of \( A_1 \) and \( A_2 \) is
\[ \mathcal{A}_1 \otimes \mathcal{A}_2 \text{ recognizes the regular set } (a + b)b(a + b) + bb \]

**Fact 4.** Given two NFA \( \mathcal{A}_1 \) and \( \mathcal{A}_2 \), we have \( \mathcal{L}(\mathcal{A}_1) \cap \mathcal{L}(\mathcal{A}_2) = \emptyset \) if and only if \( \mathcal{L}(\mathcal{A}_1 \otimes \mathcal{A}_2) = \emptyset \).

**Example 4.** Automaton \( \mathcal{A}_3 \), given below, is deterministic and equivalent to \( \mathcal{A}_1 \)

The language recognized/accepted by \( \mathcal{L}_3 \) is \( \mathcal{L}(\mathcal{A}_3) = (a + b)^*b(a + b) \).

## 2 Büchi Automata

**Definition 2.** An \( \omega \)-regular expression \( G \) over the alphabet \( \Sigma \) has the form

\[ \ldots \]
\[
G = E_1 F_1^\omega + \cdots + E_n F_n^\omega
\]
where \(E_1, \ldots, E_n, F_1, \ldots F_n\) are regular expressions over \(\Sigma\), with \(n \geq 1\), and the empty string \(\epsilon\) is not in \(L(F_i)\) for every \(1 \leq i \leq n\).

**Definition 3.** The language defined by the \(\omega\)-regular expression \(G\) is denoted \(L_\omega(G)\):

\[
L_\omega(G) = L(E_1) \cdot L(F_1)^\omega \cup \cdots \cup L(E_n) \cdot L(F_n)^\omega
\]
where \(L(E) \subseteq \Sigma^*\) denotes the language of finite strings/words defined by the regular expression \(E\).

**Definition 4.** Two \(\omega\)-regular expressions \(G_1\) and \(G_2\) are equivalent, denoted \(G_1 \equiv G_2\), if and only if \(L_\omega(G_1) = L_\omega(G_2)\).

**Example 5.** \((a + b)^* a(ab + c)\omega\) and \((ab + b(a + c))\omega\) are examples of \(\omega\)-regular expressions over the alphabet \(\Sigma = \{a, b, c\}\).

**Fact 5.** If \(\epsilon \notin L(E)\) where \(E\) is a regular expression, then we can view \(E^\omega\) as an \(\omega\)-regular expression, since \(E^\omega\) is the same as \(EE^\omega\) or also \(\epsilon \cdot E^\omega\).

**Definition 5.** A language \(L \subseteq \Sigma^\omega\) is \(\omega\)-regular if and only if \(L = L_\omega(G)\) for some \(\omega\)-regular expression \(G\) over \(\Sigma\).

**Example 6.**
1. All infinite words over \(\{a, b\}\) that contain infinitely \(a\)'s is \(\omega\)-regular. This \(\omega\)-regular language is induced by the \(\omega\)-regular expression \((b^* a)^\omega\).
2. All infinite words over \(\{a, b\}\) that contain finitely many \(a\)'s is \(\omega\)-regular. This \(\omega\)-regular language is induced by \((a + b)^* b^\omega\).
3. The empty set is an \(\omega\)-regular language induced by the \(\omega\)-regular expression \(\emptyset^\omega\).

**Fact 6.** \(\omega\)-regular languages, just like regular languages, are closed under: (i) union, (ii) intersection and (iii) complementation. The proof for (i) is easy but for (ii) and (iii) are complicated.

**Definition 6.** (Non-deterministic Büchi Automata - NBA) NBA’s are the automata that accept/recognize \(\omega\)-regular languages. A NBA \(A\) is defined just like a NFA, say \(A = (S, \Sigma, \rightarrow, \text{Init}, \text{Final})\), except that acceptance/recognition of words is defined differently.

A run \(\sigma\) for \(A\) is an \(\omega\)-sequence over \(\Sigma\), say,

\[
\sigma = a_0 a_1 a_2 \ldots
\]

which induces an \(\omega\)-sequence of states, say,

\[
s_0, s_1, s_2, \ldots
\]

such that \(s_i \xrightarrow{a_i} s_{i+1}\) for every \(i \geq 0\).

The run \(\sigma\) is accepting run if \(s_i \in \text{Final}\) for infinitely many \(i\)'s.

The language accepted/recognized by \(A\) is \(L_\omega(A) \triangleq \{\sigma \in \Sigma^\omega | \text{there is an accepting run for } \sigma \text{ in } A\}\).
Example 7. The following is a NBA $A$ over $\Sigma = \{a, b, c\}$:

![Diagram of NBA A]

We can view $A$ as a NFA or as a NBA. As a NBA, $A$ recognizes the $\omega$-regular language corresponding to the $\omega$-regular expression $c^*ab(b^+ + bc^*ab)^\omega$.

Fact 7. The languages accepted by NBA’s are exactly the $\omega$-regular languages.

Example 8. We should be careful when we compare the behaviours of NFA’s and NBA’s.

1. The following automata $A_1$ and $A_2$ accept the same finite words:

![Diagram of automata $A_1$ and $A_2$]

Namely, $L(A_1) = L(A_2) = \{a^n | n \geq 1\}$. However, $L_\omega(A_1) = \{a^\omega\}$ and $L_\omega(A_2) = \emptyset$.

2. The following automata $A_1$ and $A_2$ accept the same infinite words:

![Diagram of automata $A_1$ and $A_2$]

Namely, $L_\omega(A_1) = L_\omega(A_2) = \{a^\omega\}$. However $L(A_1) = \{a^{2n} | n \geq 0\}$ and $L(A_2) = \{a^{2n+1} | n \geq 0\}$.

Fact 8. If $A_1$ and $A_2$ are deterministic, then $L(A_1) = L(A_2)$ implies $L_\omega(A_1) = L_\omega(A_2)$.

Fact 9. NBA’s are more powerful than DBA’s (Deterministic Büchi Automata). Specifically, there does not exist a DBA $A$ such that $L_\omega(A) = L_\omega((a + b)^*b^\omega)$ (the proof of this fact is not trivial).