Lecture 7: Büchi Automata

In this lecture we shall study finite automata as acceptors of infinite words. The study of such automata goes back to Büchi ([1]) and continues to be a topic of research ([5]).

Let $\Sigma$ be a finite alphabet. An infinite word (or $\omega$-word) over $\Sigma$ is simply an infinite sequence $a_1a_2\ldots$ where each $a_i \in \Sigma$. We shall use $\Sigma^\omega$ to denote the set of all infinite words over the alphabet $\Sigma$.

Let $A = (Q, \Sigma, \delta, s, F)$ be a finite automaton. There is a natural generalization of the notion of a run from finite to infinite words. A run over an infinite word $\sigma = a_1a_2\ldots$ is a sequence $\rho = sq_1q_2\ldots$ with $s \xrightarrow{a_1} q_1 \xrightarrow{a_2} q_2\ldots$. But, when is such a run accepting? One obvious choice is to define an accepting run as one that visits some state in $F$. But with such a definition one can’t even describe the set of words which have infinitely many $a$s (Why?).

In any run $\rho$, some states of $Q$ are visited only finite number of times and some others are visited infinitely often. Let us call these sets $\text{fin}(\rho)$ and $\text{inf}(\rho)$. There is an (infinite) suffix of the run where none of the states from $\text{fin}(\rho)$ appear and the states from $\text{inf}(\rho)$ appear infinitely often. Thus, it is reasonable to assume that the classification of a run as accepting or rejecting must rely on its behaviour in the limit and hence must depend only on $\text{inf}(\rho)$. Büchi’s suggestion was to classify a run as accepting if it visits the set $F$ infinitely often. Since there are only finitely many states in $Q$ and $F$, this is equivalent to demanding that the run visit some fixed state in $F$ infinitely often.

Formally, a Büchi Automaton is a finite automaton $A = (Q, \Sigma, \delta, s, F)$, and the language accepted by such an automaton is $L(A) = \{\sigma \mid \text{there is a run } \rho \text{ over } \sigma \text{ such that } \text{inf}(\rho) \cap F \neq \emptyset\}$. A language $L \subseteq \Sigma^\omega$ is said to be $\omega$-regular if it is accepted by some Büchi automaton.

The automaton

\[
\begin{array}{c}
\text{b} \\
\downarrow \\
\text{a} \\
\text{q}_0 \\
\text{b} \\
\end{array}
\hspace{1cm}
\begin{array}{c}
\text{a} \\
\downarrow \\
\text{q}_1 \\
\text{a} \\
\end{array}
\hspace{1cm}
\begin{array}{c}
\text{b} \\
\downarrow \\
\text{q}_2 \\
\text{b} \\
\end{array}
\]

accepts all infinite words over $\{a, b\}$ in which, every $a$ has a $b$ occurring some where to its right. In this lecture we shall omit the $\omega$ and call a $\omega$-word as simply a word.

The following automaton

\[
\begin{array}{c}
\text{a} \\
\downarrow \\
\text{a} \\
\text{q}_0 \\
\text{a,b} \\
\end{array}
\hspace{1cm}
\begin{array}{c}
\text{b} \\
\downarrow \\
\text{b} \\
\text{q}_1 \\
\text{b} \\
\end{array}
\hspace{1cm}
\begin{array}{c}
\text{a} \\
\downarrow \\
\text{a} \\
\text{q}_2 \\
\text{a} \\
\end{array}
\]

accepts all words that have infinitely many $a$s.

The automaton
accepts all words that have finitely many as. Call this language \( \text{Finite}(a) \).

1 Deterministic Büchi Automata

Of the three automata described above the first two are deterministic whilst the last one is not. Can one design a deterministic Büchi automaton that accepts the set of words with finite number of as? The answer is negative and this can be seen as follows: Suppose there is a deterministic Büchi automaton \( A \) accepting this language. This automaton must have an accepting run on the word \( ab^n \) (where \( b^n = bbb \ldots \)). Suppose this (unique) run enters a state in \( F \) after \( ab^{n_1} \) for some \( n_1 \geq 1 \). Now, \( ab^{n_1}ab^n \) is also in the language and there is a unique run on this word, which extends the aforementioned run on \( ab^{n_1} \), that is accepting and such a run must visit a state in \( F \) after reading \( ab^{n_1}ab^{n_2} \) for some \( n_2 \geq 1 \). Repeating this argument we can construct a sequence \( ab^{n_1}ab^{n_2} \ldots ab^n \ldots \) on which the unique run visits a state in \( F \) after reading \( ab^{n_1}, ab^{n_1}ab^{n_2}, \ldots ab^{n_1}ab^{n_2} \ldots ab^n, \ldots \). Thus, this run visits the set \( F \) infinitely often and hence this string with infinitely many as is accepted by \( A \). This contradicts our assumption that \( A \) accepted the language of words with finite number of as. Thus, nondeterministic Büchi automata are more powerful than deterministic Büchi automata.

Let \( L \) be a regular language of finite words. We define \( \hat{L} \) to be the \( \omega \)-language consisting of all words that have infinitely many prefixes in \( L \). For example if \( L = \Sigma^*a \) then \( \hat{L} \) is the set of words with infinitely many as. Then, we can characterize the class of languages accepted by deterministic Büchi automata as follows:

**Theorem 1** Let \( A \) be a deterministic Büchi automaton and let \( L_f(A) \) be the language of finite words accepted by \( A \) when treated as a finite automaton and let \( L(A) \) be the language accepted by \( A \) as a Büchi automaton. Then,

\[
L(A) = \hat{L_f(A)}
\]

The proof of this theorem quite easy and we leave it as an exercise. Our proof above showing that \( \text{Finite}(a) \) is not accepted by any deterministic Büchi automaton can be seen as showing that \( \text{Finite}(a) \) is not \( \hat{L} \) for any language \( L \).

We now examine the closure properties of \( \omega \)-regular languages. Given Büchi automata recognizing languages \( L_1 \) and \( L_2 \) it is quite trivial to construct a Büchi automaton accepting the language \( L_1 \cup L_2 \). On the other hand constructing an automaton that accepts \( L_1 \cap L_2 \) requires some ingenuity. We leave that as an interesting exercise.

---

1 If you wonder I write \( \hat{L} \) and not \( \tilde{L} \), the answer is rather embarrassing. I can’t get \( \tilde{L} \) to stretch over longer expressions like \( L_1 \cup \mathcal{L}_2 \).
Exercise: Show how to construct a Büchi automaton accepting $L_1 \cap L_2$ from automata accepting $L_1$ and $L_2$.

Exercise: Show how to construct deterministic Büchi automata accepting $L_1 \cup L_2$ and $L_1 \cap L_2$ from deterministic automata accepting $L_1$ and $L_2$.

2 Complementation of Büchi Automata

We saw three different techniques to establish the closure under complementation of regular languages: via deterministic automata, via Myhill-Nerode congruences and finally via alternating automata. From the previous section it seems that the first route is not available in the case of $\omega$-regular languages. However, using more general acceptance conditions than the Büchi condition, one can obtain deterministic automata accepting all $\omega$-regular languages.

In the next couple of lectures we shall use each of the three routes in demonstrating the closure under complementation of $\omega$-regular languages.

The easiest technique, and the one used by Büchi himself, is via congruences. Our presentation below follows that of Thomas [7]. Recall that we associated a congruence over $\Sigma^*$ with every deterministic finite automaton $A$, given by $x \equiv_A y$ if and only if $\forall q \in Q. \delta(q, x) = \delta(q, y)$. Since acceptance of finite words is decided by where the run starting at $s$ ends up, this is the right notion. This equivalence is of finite index and further it saturates $L(A)$. That is, $[x]_{\equiv_A} \subseteq L(A)$ or $[x]_{\equiv_A} \cap L(A) = \emptyset$ for each $x \in \Sigma^*$. Since each $x \in \Sigma^*$ lies in some class (namely $[x]$) we can write both $L(A)$ as well as $\overline{L(A)}$ as unions of these equivalence classes (each of which is a regular language).

How do we extend these ideas to $\omega$-regular languages? We need to extend the above construction in two ways, firstly we must be able associate a congruence with nondeterministic automata and secondly it must capture the notion of Büchi acceptance. The first step is easy: With any nondeterministic finite automaton $A$ we can associate a congruence $\equiv_A$ defined by $x \equiv_A y$ if and only if $\forall q. \delta(q, x) = \delta(q, y)$ where this equality is an equality of sets. In other words, there is a run from $q$ to $q'$ on the word $x$ if and only if there is a run from $q$ to $q'$ on the the word $y$ (for any $q$ and $q'$). This relation is a congruence, is of finite index and saturates $L(A)$. Extending this relation to capture accepting runs over infinite words requires a little bit of thought. If $x$ and $y$ are equivalent then we would like to be able to replace any number of occurences of $x$ with $y$ in any run without affecting acceptance. This leads us to the following definition: With each Büchi automaton $A$ we associate a relation $\equiv_A$ over $\Sigma^*$ given by

$$x \equiv_A y \overset{\Delta}{=} \forall q, q'. q \xrightarrow{x} q' \iff q \xrightarrow{y} q' \land \forall q, q'. q \xrightarrow{x} q' \iff q \xrightarrow{y} q'$$

where $q \xrightarrow{x} q'$ means that there is a run from $q$ to $q'$ on the word $x$ that passes through some state in $F$ (there may be other runs that do not pass through any state in $F$). We shall often write $[x]$ for $[x]_{\equiv_A}$. 

3
If \( x \) and \( y \) are equivalent then in any infinite run we may replace any number of subruns on \( x \) by an appropriate runs on \( y \) in such a way that if the original run was accepting then the modified run continues to be accepting.

It is easy to check that this relation is a congruence. It is of finite index as the number equivalence classes is bounded by the number of functions from \( Q \) to \( 2^Q \times 2^Q \). Further, we claim that for any \( x \) and \( y \), \([x],[y]^{\omega}\subseteq L(A)\) or \([x],[y]^{\omega}\cap L(A) = \emptyset\). This can be seen as follows: Suppose \( x_1y_1y_2\ldots \in L(A)\) with \( x_1 \in [x] \) and \( y_i \in [y] \) for \( i \). Consider any accepting run \( \rho = s \xrightarrow{x_1} q_1 \xrightarrow{y_1} q_2 \xrightarrow{y_2} q_3 \ldots \). Let \( x'_1 \equiv x_1, y'_1 \equiv y_1 \) and so on. Then, we know that there are runs \( s \xrightarrow{x'_1} q_1, q_1 \xrightarrow{y'_1} q_2 \) and so on, such that whenever the run \( q_i \xrightarrow{y_i} q_{i+1} \) visits a final state so does the run \( q'_i \xrightarrow{y'_i} q'_{i+1} \). Thus, the run \( s \xrightarrow{x'_1} q_1 \xrightarrow{y'_1} q_2 \xrightarrow{y'_2} q_3 \ldots \) visits \( F \) infinitely often and hence \( x'_1y'_1y'_2\ldots \) is also in \( L(A) \).

So what have we got so far? Suppose \( \equiv \) has \( N \) congruence classes. Then, each of the \( N^2 \omega \)-languages obtained as \([x],[y]^{\omega}\) is either completely contained in \( L(A) \) or in \( \overline{L(A)} \). Further, the following exercise guarantees that all these \( N^2 \) languages are \( \omega \)-regular.

**Exercise:** Let \( U \) and \( V \) be regular languages. Show that \( U.V^{\omega} \) is a \( \omega \)-regular language.

So, it seems that we have shown that \( \equiv \) “saturates” \( L(A) \) and should be able to conclude that both \( L(A) \) and \( \overline{L(A)} \) are just finite unions of languages of the form \([x],[y]^{\omega}\). Then, using the above exercise we have a proof that the complement of a \( \omega \)-regular language is also \( \omega \)-regular. However, there is a gap. Unlike the case of finite words where it is a trivial fact that each word in \( \Sigma^* \) lies in some equivalence class of \( \equiv \), it is not clear that every \( \omega \)-word is an element of \([x],[y]^{\omega}\) for some \( x, y \). This needs proof and is in fact the most intricate part of Büchi’s argument.

Following Gastin and Petit [4] (who attribute the original ideas to Perrin and Pin, see for instance, [8]), we shall find it convenient to use the monoid \( \Sigma'/\equiv_A \) and prove following general result about monoids.

**Theorem 2** Let \( M_1 \) be the free monoid over a (possibly infinite) alphabet \( \Sigma \). \( M \) be a finite monoid and let \( h \) be a homomorphism from \( M_1 \) to \( M \). Let \( a_1a_2a_3\ldots \) be any infinite word over \( \Sigma \). Then, there are elements \( s \) and \( e \) in \( M \) such that \( e.e = e \) and \( a_1a_2\ldots \in h^{-1}(e)(h^{-1}(e))^{\omega} \).

**Proof:** For \( i < j \), let \( m(i,j) \) denote \( h(a_1a_1\ldots a_{j-1}) \). This mapping is a colouring of all the 2-subsets of \( N \) using the finite set \( M \). Then, the infinite version of Ramsey’s theorem guarantees that there is a subset \( i_1, i_2, \ldots \) such that the colour of any 2-subset in this set is identical. That is \( m(i_j,i_k) = m(i_l,i_m) \) for any \( j < k, l < m \). Let \( s = m(1,i_1) \) and \( e = m(i_1,i_2) \). Then, \( e.e = m(i_1,i_2).m(i_1,i_2) = m(i_1,i_2).m(i_2,i_3) = m(i_1,i_3) = e \). Finally, by the definition of \( m(i,j) \), \( a_1\ldots a_{i_1-1} \in h^{-1}(s) \) and \( a_{i_1}a_{i_1+1}\ldots a_{i_{j+1}} \in h^{-1}(e) \).

If you do not like the infinite version of Ramsey’s theorem, here is a direct argument. Let \( N_0 = \{1,2,\ldots\} \). For \( i = 0,1,2,\ldots \) we define the set \( N_i \), the number \( n_i \) and an element \( s_i \) as follows:

\[
\begin{align*}
n_i &= \text{smallest number in } N_i \\
n_i &= \text{some element of } M \text{ such that for infinitely many } j \in N_i, m(n_i,j) = s_i \\
N_{i+1} &= \{ j \mid m(n_i,j) = s_i \}
\end{align*}
\]
Clearly, \( N_i \) is infinite for each \( i \). Let \( e \) be an element that occurs as \( s_i \) for infinitely many \( i \) and let \( i_1 < i_2 < \ldots \) be such that \( s_{i_j} = e \). Then, \( m(n_{i_j}, n_{i_k}) = e \) for any \( j < k \). Thus, the indices \( i_1, i_2, \ldots \) identify a Ramsey subset and the rest of the proof follows as above. Notice that since \( n_0 = 1 \), we also have that \( s = m(1, n_{i_1}) = m(1, n_{i_2}) = \ldots \) and hence \( s.e = s \). ■

A pair of elements \((s, e)\) in a monoid with \( s.e = s \) and \( e.e = e \) is called a linked pair. Thus we have established that any infinite sequence over \( M_1 \) is in a set of the form \( h^{-1}(s). (h^{-1}(e))^\omega \) for a linked pair \((s, e)\). It turns out that linked pairs play a rather important role in the study of the algebraic theory of \( \omega \)-regular languages, a topic to which we shall return later in the course. Notice that the second element of a linked pair is an idempotent and we shall exploit this fact shortly.

Using \( \Sigma^* \) as \( M_1 \) and \( \Sigma/\equiv_A \) as \( M \) we can conclude that each \( a_1a_2\ldots \) in \( \Sigma^\omega \) is in \( \eta^{-1}([x]). (\eta^{-1}([y]))^\omega \) where \( \eta(x) = [x] \). But \( \eta^{-1}([x]) = [x] \) and thus \( a_1a_2\ldots \in [x.[y]^\omega \) for some \( x, y \). Putting this together with our earlier calculations yields the following result:

**Theorem 3 (Büchi)** Let \( A \) be a Büchi automaton. Then,

\[
\frac{L(A)}{L(A)} = \bigcup_{[x,y] \mid [x][y]^\omega \subseteq L(A)} [x][y]^\omega
\]

We also obtain the following characterization of \( \omega \)-regular languages:

**Theorem 4 ([8])** A language \( L \) over \( \Sigma^\omega \) is a \( \omega \)-regular language if and only if there is a homomorphism \( h \) from \( \Sigma^* \) to a finite monoid \( M \) and a collection \( X \) of linked pairs over \( M \) such that \( L = \bigcup_{(s, e) \in X} h^{-1}(s). (h^{-1}(e))^\omega \).

One direction of this theorem is proved above and the other direction is left as a (rather trivial) exercise.

### 3 Determinizing Büchi Automata

We shall follow the route taken by Eilenberg and Schutzenberger [7], Choueka [2] as well as Rabin [9] as described by Perrin and Pin in [8]. Recall that deterministic Büchi automata recognize limit languages (i.e. languages of the form \( \tilde{L} \) for some regular language \( L \)). We next show that if \( h \) is any homomorphism from \( \Sigma^* \) to a finite monoid \( M \) and \( e \) is an idempotent in \( M \) then \( (h^{-1}(e))^\omega \) is a limit language.

Let \( X_e \) denote \( h^{-1}(e) \) and let \( P_e \) denote the prefix minimal words in \( X_e \) (i.e. \( P_e = \{ x \in X_e \mid y < x \text{ implies } y \notin X_e \} \)). Let \( \sigma \in X_e^\omega \). Then, \( \sigma = x_1x_2\ldots \) with each \( x_i \in X_e \). For each \( i \) there is a \( p_i \in P_e \) with \( p_i \leq x_i \). Thus, there are infinite many prefixes of the word \( \sigma \), namely \( x_1p_2, x_1x_2p_3, \ldots, x_1x_2\ldots x_ip_{i+1}, \ldots \), in \( X_e.P_e \). Thus \( X_e^\omega \subseteq \tilde{X}_e.P_e \). The following lemma establishes the converse.

**Lemma 5** Let \( h \) be a homomorphism from \( \Sigma^* \) to a finite monoid \( M \) and let \( e \) be an idempotent and let \( X_e \) and \( P_e \) be as defined above. Then, \( \tilde{X}_e.P_e = X_e^\omega \).
Proof: The discussion above established that \( X_e^\omega \subseteq \hat{X}_e P_e \). Let \( \sigma \in \hat{X}_e P_e \). Then, there are infinitely many \( x_i \)'s in \( X_e \) and \( p_i \)'s in \( P_e \) such that \( x_1 p_1 < x_2 p_2 < \ldots x_ip_i \ldots < \sigma \). Suppose the lengths of these \( x_i \)'s were bounded. Then for some \( i < j \), \( x_i = x_j \) and this in turn means that \( x_i p_i < x_i p_j \) contradicting the requirement that \( p_j \) has no prefixes in \( X_e \). Thus, without loss of generality we may assume that \( x_i p_i < x_{i+1} \). Let us define \( v_1, v_2, \ldots \) as words such that \( x_i v_i = x_{i+1} \). Note that \( p_i < v_i \).

\[
\begin{align*}
\begin{array}{c}
\hline
x_1 & p_1 \\
\hline
x_2 & v_1 & p_2 \\
\hline
x_3 & v_2 & p_3 \\
\hline
\end{array}
\end{align*}
\]

Thus, \( \sigma = x_1 v_1 v_2 v_3 \ldots \). Now, we use the fact that \( x_1, v_1, v_2, \ldots \) are elements of the monoid \( \Sigma^* \) and Theorem 2 to conclude the existence of \( s \) and \( f \) in \( M \) such that \( x_1 v_1 v_2 \ldots \) can be factored as \( h^{-1}(s).h^{-1}(f)^\omega \) with \( f.f = f \) and \( s.f = s \). Since we treat \( x_1, v_1, v_2, \ldots \) as elements of the monoid \( \Sigma^* \) the factoring respects these word boundries. That is, there is a factorization where \( h^{-1}(s) \) looks like \( x_1 v_1 \ldots v_i \) and \( v_{i+1} \ldots v_j \) is in \( h^{-1}(f) \) for some \( j > i \) and so on. Then, \( h(x_1 v_1 \ldots v_i) = h(x_{i+1}) = e \). Thus \( s = e \).

From the factoring, we have that \( v_{i+1} \ldots v_j \in h^{-1}(f) \) and since \( p_{i+1} < v_{i+1} \) there must be a \( g = h(v_{i+1} \ldots v_j/p_{i+1}) \) (where \( x/y \) is \( z \) if \( x = yz \)) with \( eg = f \).

Here is a cute fact about idempotents. If \( e \) and \( f \) are idempotents and \( eg = f \) then \( ef = f \). In proof note that \( ef = eeg = eg = f \). Thus, \( ef = f \). But recall that \( s.f = s \) and \( s = e \) and thus \( e.f = e \). Thus \( e = f \) we have actually factored \( \sigma = x_1 v_1 v_2 \ldots \) as \( (h^{-1}(e))^\omega \) and thus \( \sigma \in X_e^\omega \).

Now, putting together Theorem 4 and Lemma 5 we have the following characterization of \( \omega \)-regular languages.

**Theorem 6** A language \( L \) is \( \omega \)-regular if and only if there is a finite set \( I \) and regular languages \( U_i, V_i \), for each \( i \in I \), such that

\[
L = \bigcup_{i \in I} U_i \hat{V}_i
\]

Actually, there is a monoid \( M \) and a homomorphism \( h \) to \( M \) and a a linked pair \( (s_i, e_i) \) for each \( i \) such that \( U_i \) is recognized via \( s_i \) and \( V_i \) via \( e_i \). Note, that there is no hope of replacing \( U_i \hat{V}_i \) by some \( \hat{W}_i \) (Why? Use the fact that limit languages are closed under union).
Notes: The most widely used introductory article on $\omega$-automata is [7]. Another good introduction to this topic is found in [6]. Our presentation has drawn from [7] and [8].

References


[3] S. Eilenberg and M.P. Schützenberger:


[9] Michael Rabin: