Computer Science & Engineering Department
IIT Kharagpur
Theory of Computation: CS41001
Lecture I

Instructor: Goutam Biswas *Autumn Semester 2014-2015*

1 Büchi Automata

1.1 Finite and Infinite Sequences

An *alphabet* Σ is a finite set of symbols. The collection of finite length sequences of symbols from Σ is called Σ^*.

$$\Sigma^* = \{\sigma_1 \cdots \sigma_k : \sigma_i \in \Sigma,\ k \in \mathbb{N}_0\} = \bigcup_{k \in \mathbb{N}_0} \Sigma^k,$$

where $\mathbb{N}_0 = \{0, 1, 2, \cdots\}$ and $\Sigma^k = \{\sigma_1 \cdots \sigma_k : \sigma_i \in \Sigma\}$ for all $k \in \mathbb{N}_0$.
It is not difficult to prove that Σ^* is a countably infinite set[1].
A Σ-*language* (or a language over Σ) L is a subset of Σ^*. So the set of Σ-*languages* is uncountable[2] and most of these languages cannot have any *effective finite description*.

We may view $x \in \Sigma^k$, as a map from $\{1, 2, \cdots, k\} \to \Sigma$, where $x(i) \in \Sigma$ is the i^{th} symbol of x from the left. We shall write x_i for $x(i)$. The set Σ^k is the collection of all possible maps from $\{1, 2, \cdots, k\} \to \Sigma$. If there are d symbols in Σ, the size of Σ^k is d^k. The *null string* ε is the map from $\emptyset \to \Sigma$.

An infinite string α is a map from $\mathbb{N} \to \Sigma$, where $\mathbb{N} = \{1, 2, 3, \cdots\}$. As usual $\alpha(i)$ is the i^{th} symbol of α, and we write α_i. The collection of all infinite strings over Σ is called Σ^ω. It is also not difficult to show that Σ^ω is uncountable[3] if $|\Sigma| > 1$.
An ω-*language* over Σ is a subset of Σ^ω. We call $\Sigma^\infty = \Sigma^\omega \cup \Sigma^*$.
Example 1. It is clear that $\emptyset$ is a subset of both Σ^* as well as Σ^ω. If $L \subseteq \Sigma^*$ and $\alpha \in \Sigma^\omega$, then $L\alpha = \{x\alpha : x \in L\}$ is an ω-*language* over Σ.

1.2 Finite State Recognisers

We look into languages both over Σ^* and also over Σ^ω, that can be *recognised* by machine models having finite number of states. It has finite memory, so it can remember *finite amount of history* for deciding the future course of action.

1.2.1 Languages over Σ^*

It is known that most of the Σ^*-*languages* cannot have effective descriptions. But some of them (countably many) can be recognised by a mathematical machine model known as *nondeterministic finite automaton (nfa)*.

An *nfa* N, over Σ is specified by four data, $N = (Q, I, F, T)$, where Q is a finite set of *states*, $I \subseteq Q$ is the set of *initial states*, $F \subseteq Q$ is the set of *final states*, and $T \subseteq Q \times \Sigma \times Q$, is the *transition relation*. The transition relation gives a *directed graph* with set of states as vertices. Edges of the graph are labeled with the symbols of Σ.
If $(p, \sigma, q) \in T$, then there is an edge from p to q labeled with σ. The reading is that the machine from the state p will go to state q on input (on event) σ.

A *run* or *computation* of the machine on an input $x = \sigma_1\sigma_2 \cdots \sigma_k \in \Sigma^*$, is a finite sequence of states $s_1, s_2, \cdots, s_{k+1}$, such that

1. $s_1 \in I$, and
2. $(s_i, \sigma_i, s_{i+1}) \in T$, for all i, $1 \le i \le k$.

[1] If $|\Sigma| = d$, then any element of Σ^* may be viewed as a numeral in radix-$(d+1)$ numeral. So there is an injective map from $\Sigma^* \to \mathbb{N}$, the set of natural numbers.
[2] There is no surjective map from $\Sigma^* \to \mathcal{P}\Sigma^*$ (Cantor).
[3] If Σ has two symbols, then a map from $\mathbb{N} \to \Sigma$ may be viewed as the characteristic function of a subset of $\mathbb{N}$. So the collection of all such functions is bijective with the power set of $\mathbb{N}$. And it is known (Cantor) that power set cannot be equinumerous to the original set.

An *nfa* N *accepts* $x = \sigma_1\sigma_2\cdots\sigma_k \in \Sigma^*$, if there is a run of the machine on x such that the last state $s_{k+1} \in F$. The language of the automaton $L(N)$ is the collection of all strings for which there are accepting runs or computations of the automaton. A language $L \subseteq \Sigma^*$ is is called *regular*, if there is an *nfa* N such that $L = L(N)$.

An *nfa* is *deterministic (dfa)* if following two conditions are satisfied.

1. $|I| = 1$,
2. for each $p \in Q$ and $\sigma \in \Sigma$, there is at most one $q \in Q$ so that $(p, \sigma, q) \in T$.

An *nfa* or *dfa* is *complete* if for each $p \in Q$ and $\sigma \in \Sigma$, there is at least one $q \in Q$ so that $(p, \sigma, q) \in T$.
It can be shown that the model of *dfa* and *nfa* are of equivalent in terms of recognising languages of Σ^* i.e. if $L \subseteq \Sigma^*$ is recognised by an *nfa*, then there is a *dfa* for L. But an *nfa* is a more succinct model. In the worst case the equivalent *dfa* model of an n state *nfa* can have $O(2^n)$ states.
Ex 1. Show that the collection of all regular languages over Σ^* is a countable Boolean algebra.

1.2.2 Languages over Σ^ω

In case of languages over Σ^ω we have a model similar to *nfa*, known as *Büchi automaton*[4]. A *Büchi automaton* is also specified by four data, $B = (Q, I, F, T)$, with the symbols having their same meaning.
But a *run* or *computation* of this model on an infinite sequence of elements from Σ is an infinite sequence of states (non-terminating). Let the sequence of events or inputs be $\alpha = \alpha_1\alpha_2\cdots \in \Sigma^\omega$. The corresponding infinite *run* or *computation* is the sequence of states $s_1, s_2, \cdots \in Q^\omega$, such that

1. $s_1 \in I$, and
2. $(s_i, \alpha_i, s_{i+1}) \in T$, for all $i \in \mathbb{N}$.

The obvious question is what can be the an *accepting* or *successful run*?

A *successful* or *accepting run* of a Büchi automaton is the one that starts from an initial state, and at least one of the final states is present infinitely often in it.
Given a run $r : \mathbb{N} \to Q$ we define a subset of states $Inf(r) = \{p \in Q : r^{-1}(p) \text{ is infinite}\}$, where $r^{-1}(p)$ is the set of pre-images of the state $p \in Q$. So an infinite sequence $\alpha = \alpha_1\alpha_2\cdots$ is accepted by a Büchi automaton $B = (Q, I, F, T)$, if $r = s_1, s_2, \cdots$ is a run[5] of the automaton on α and $F \cap Inf(r) \neq \emptyset$. The language of the Büchi automaton B,

$$L(B) = \{\alpha \in \Sigma^\omega : \ B \text{ has a successful run on } \alpha\}.$$

A Büchi automaton is said to be *incomplete* if there is a some $s \in Q$ and $\sigma \in \Sigma$, there is no state $t \in Q$ such that $(s, \sigma, t) \in T$.
An *incomplete* Büchi automaton can be completed by introducing a new state $g \notin Q$ and the following set of transitions.

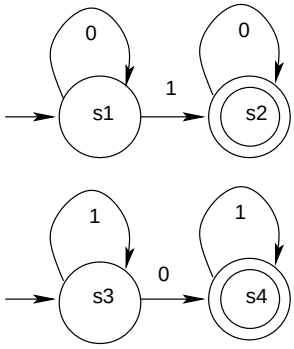
1. $T_1 = \{(s, \sigma, g) : \ (s, \sigma, t) \notin T \text{ for any } t \in Q\}$.
2. $T_2 = \{(g, \sigma, g) : \ \sigma \in \Sigma\}$.

The equivalent *complete* Büchi automaton, $B' = (Q \cup \{g\}, I, F, T \cup T_1 \cup T_2)$.
Example 2. Following are a few examples of Büchi automaton. Let $\Sigma = \{0, 1\}$ in all these cases.

1. Each string has either exactly one 0 or exactly one 1.

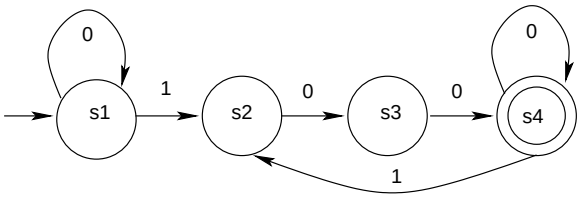
[4] In 1962 it was introduced by J R Büchi in his paper *On a decision method in restricted second order arithmetic*.

[5] There may be more than one run of B on α, as the machine is non-deterministic.

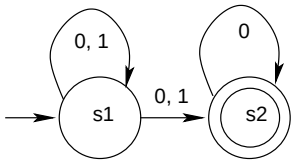


The machine has two start states, $I = \{s_1, s_3\}$ and two final states, $F = \{s_2, s_4\}$. It is incompletely specified.

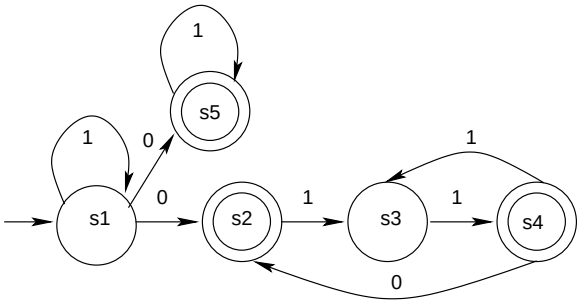
2. Each 1 should be followed by two 0's.



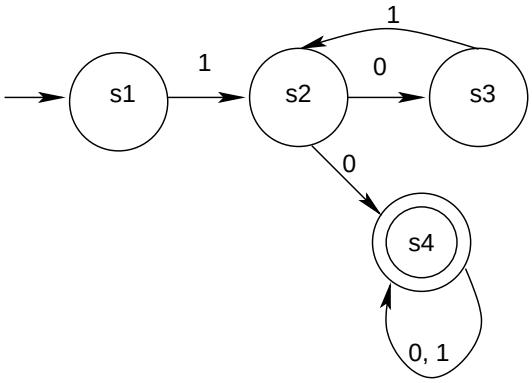
3. There are finitely many 1's in each sequence.



4. After every 0 there must be either even number of consecutive 1's or there must be infinite number of consecutive 1's.



5. Each sequence is of the form $x\alpha$, where $x = (10)^k$ and $\alpha \in \Sigma^\omega$, where $k \geq 1$.



1.2.3 Closure Properties

The class of languages over Σ^ω recognised by Büchi automaton is closed under (i) union, (ii) intersection and (iii) projection. We present this fact in form of following lemmas.

Lemma 1. If L_1 and L_2, languages over Σ^ω, are recognised by Büchi automata $B_1 = (Q_1, I_1, F_1, T_1)$ and $B_2 = (Q_2, I_2, F_2, T_2)$ respectively, then $L_1 \cup L_2$ is also recognised by a Büchi automaton.

Proof: We construct the recogniser for $L = L_1 \cup L_2$. Without any loss of generality we assume that $Q_1 \cap Q_2 = \emptyset$. Our recogniser for L is $B = (Q_1 \cup Q_2, I_1 \cup I_2, F_1 \cup F_2, T_1 \cup T_2)$.

The claim that any $\alpha \in L_1 \cup L_2$ is accepted by B is straight forward. If α has an *successful* run on B_1, then it also has a successful run on B. Similar is the case if α has a successful run on B_2.

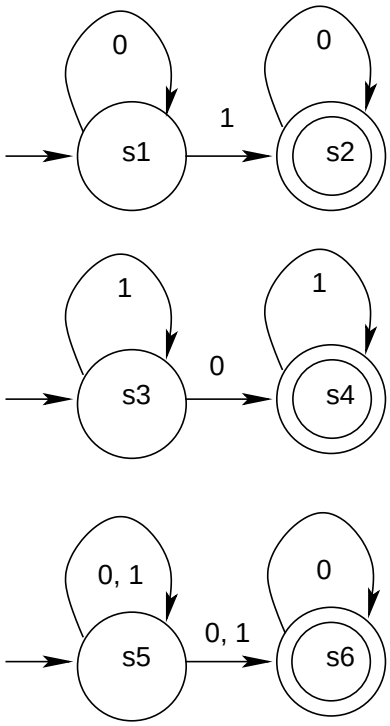
In the other direction, if α has a *successful* run on B, then it must have a *successful* run on B_1 or B_2 component of B. By our construction these two components are disjoint. QED.

Example 3. Consider the following two languages:

L_1: each string has either exactly one 0 or exactly one 1.

L_2: there are finitely many 1's in each sequence.

The automaton for $L = L_1 \cup L_2$ is



Lemma 2. If L_1 and L_2, languages over Σ^ω, are recognised by Büchi automata $B_1 = (Q_1, I_1, F_1, T_1)$ and $B_2 = (Q_2, I_2, F_2, T_2)$ respectively, then $L_1 \cap L_2$ is also recognised by a Büchi automaton.

Proof: The construction of intersection machine requires two things. Given an input $\alpha \in \Sigma^\omega$,

1. we should simulate computations of both B_1 and B_2 simultaneously, and
2. keep track of visiting final states of both the machines.

The first condition calls for the set of states of the intersection machine as the Cartesian product of Q_1 and Q_2. For a successful run on an input α on the intersection machines, the final states of each of the component machines must appear infinitely often in the run. We use a '*switch*' in the states of the intersection machine to remember that the final states of B_1 and B_2 are visited alternately. This will not create any problem as on any successful run on an input $\alpha \in L_1 \cap L_2$, of the intersection machine, as after every occurrence of a final state of either B_1 or B_2, there are infinitely many occurrences of final states of both B_1 and B_2.

We describe the product or intersection machine as follows:

$$B = (Q_1 \times Q_2 \times \{1,2,3\}, I_1 \times I_2 \times \{1\}, Q_1 \times Q_2 \times \{3\}, T).$$

where transition table T is defined as follows:

1. $((s_1, s_2, 1), \sigma, (s_1', s_2', 1))$, where $(s_1, \sigma, s_1') \in T_1$, $(s_2, \sigma, s_2') \in T_2$, and $s_1' \notin F_1$. Expecting a final state of B_1.
2. $((s_1, s_2, 1), \sigma, (s_1', s_2', 2))$, where $(s_1, \sigma, s_1') \in T_1$, $(s_2, \sigma, s_2') \in T_2$, and $s_1' \in F_1$. Expecting a final state of B_2.
3. $((s_1, s_2, 2), \sigma, (s_1', s_2', 2))$, where $(s_1, \sigma, s_1') \in T_1$, $(s_2, \sigma, s_2') \in T_2$, and $s_2' \notin F_2$. Expecting a final state of B_2.
4. $((s_1, s_2, 2), \sigma, (s_1', s_2', 3))$, where $(s_1, \sigma, s_1') \in T_1$, $(s_2, \sigma, s_2') \in T_2$, and $s_2' \in F_2$. Visited both the final states.
5. $((s_1, s_2, 3), \sigma, (s_1', s_2', 1))$, where $(s_1, \sigma, s_1') \in T_1$, $(s_2, \sigma, s_2') \in T_2$, and $s_1' \notin F_1$. Expecting a final state of B_1.
6. $((s_1, s_2, 3), \sigma, (s_1', s_2', 2))$, where $(s_1, \sigma, s_1') \in T_1$, $(s_2, \sigma, s_2') \in T_2$, and $s_1' \in F_1$. Expecting a final state of B_2.

It is possible to work with two-valued switch i.e. $Q = Q_1 \times Q_2 \times \{1,2\}$, where $F = F_1 \times F_2 \times \{2\}$ etc. QED.

Definition 1: Let $\Sigma = \Sigma_1 \times \Sigma_2$, and $\alpha = (\alpha_{11}, \alpha_{12})(\alpha_{21}, \alpha_{22})(\alpha_{31}, \alpha_{32}) \cdots$. The Σ_1-projection of α is $\pi_1(\alpha) = \alpha_1 = \alpha_{11}\alpha_{21}\alpha_{31} \cdots \in \Sigma_1^\omega$. Similarly, Σ_2-projection of α is $\pi_2(\alpha) = \alpha_2 = \alpha_{12}\alpha_{22}\alpha_{32} \cdots \in \Sigma_2^\omega$.

If $L \subseteq \Sigma^\omega = (\Sigma_1 \times \Sigma_2)^\omega$, then the projection languages of L are $\pi_1(L) = \{\pi_1(\alpha) : \alpha \in L\}$ and $\pi_2(L)$, defined similarly.

Lemma 3. If $L \subseteq \Sigma^\omega$, where $\Sigma = \Sigma_1 \times \Sigma_2$, is recognised by a Büchi automaton $B = (Q, I, F, T)$, then $\pi_1(L)$ and $\pi_2(L)$ are also Büchi recognisable.

Proof: We construct the machine $B_1 = (Q, I, F, T_1)$ to recognise $\pi_1(L)$, where for all $\sigma_1 \in \Sigma_1$, $(s, \sigma_1, t) \in T_1$, if there exists some $\sigma_2 \in \Sigma_2$, such that $(s, (\sigma_1, \sigma_2), t) \in T$.

Let $\alpha = (\alpha_{11}, \alpha_{12})(\alpha_{21}, \alpha_{22})(\alpha_{31}, \alpha_{32}) \cdots \in L \subseteq \Sigma^\omega = (\Sigma_1 \times \Sigma_2)^\omega$. So there is a *successful* run $r = s_1 s_2 \cdots$ of B on α, where $s1 \in I$ and $(s_i, (\alpha_{i1}, \alpha_{i2}), s_{i+1}) \in T$, for all $i = 1, 2, \cdots$. But then according to the construction there is a successful run of B_1 on $\pi_1(\alpha) = \alpha_{11}\alpha_{21} \cdots$. The other direction can also be proved similarly.

We can construct recogniser for $\pi_2(L)$ in a similar manner. QED.

1.2.4 Büchi Characterisation Theorem

First we show that for any regular language L over Σ, the language of infinite words L^ω is recognised by some Büchi automaton, where $L^\omega = \{x_1 x_2 \cdots :$ where all $x_i \in L$, $i \geq 1$ and $x_i \neq \varepsilon\}$.

We start with a regular language L over Σ^*. There is an nfa $N = (Q, I, F, T)$ that recognises L. It is well known that the nfa can be converted to an equivalent dfa by subset construction. Let $M = (Q, s_0, F, T)$ be the equivalent dfa. The following lemma claims that there is an equivalent dfa with a start state having no incoming transition.

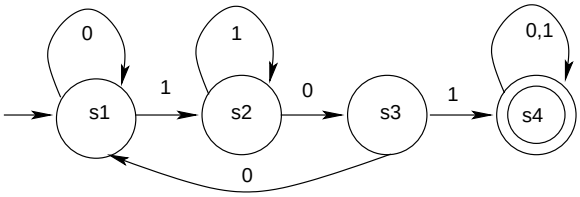
Lemma 4. If a regular language L is recognised by a dfa $M = (Q, s_0, F, T)$, then there is an equivalent dfa $M_1 = (Q_1, s_0, F_1, T_1)$ such that $L = L(M_1)$ and there is no incoming transition to the initial state s_0 of M_1.

Proof: The construction of M_1 is simple. If there is no incoming transition to s_0, then $M_1 = M$. If there are incoming transitions to s_0, we introduce a new state $s_0' \notin Q$. The incoming transitions to s_0 are removed and are directed towards s_0'. The state s_0' has all the outgoing transitions of s_0.

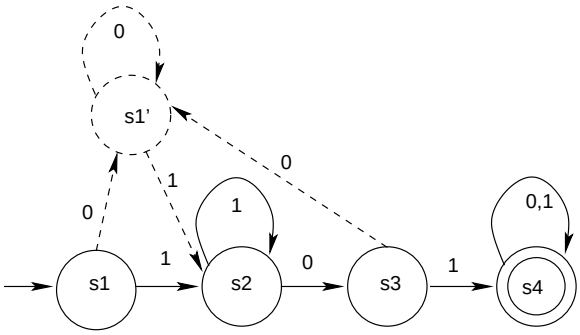
1. $Q_1 = Q \cup \{s_0'\}$.
2. $T_1 = (T \setminus \{(t, \sigma, s_0) : t \in Q, \sigma \in \Sigma\}) \cup \{(t, \sigma, s_0') : \text{ if } (t, \sigma, s_0) \in T\} \cup \{(s_0', \sigma, t) : \text{ if } (s_0, \sigma, t) \in T\}$.

3. $F_1 = F$ if $s_0 \notin F$, else $F_1 = F \cup \{s_0'\}$.

We can prove by induction on the length of $x = \sigma_1 \cdots \sigma_k \in \Sigma^*$, that there is a run $r = s_0 s_1 \cdots s_k$ of M on x if and only if there is a run $r_1 = s_0 s_1' \cdots s_k'$ of M_1 on x, where $s_i' = s_i$ if $s_i \neq s_0$ and $s_i' = s_0'$ otherwise, for all $i = 1, \cdots, k$. QED.
Example 4. Let $L = \{x \in \{0,1\}^* : \ x$ has a substring $101\}$. Following dfa accepts L



We wish to remove incoming transition to the initial state s_1.



Given a dfa $M = (Q, s_0, F, T)$ for the language $L \subseteq \Sigma^*$ where there is no incoming transition to any initial state, the construction of the Büchi automaton for L^ω is simple.
Lemma 5. If $L \subseteq \Sigma^*$ is recognised by a dfa $M = (Q, s_0, F, T)$, there there is a Büchi automaton for L^ω.
Proof: Without any loss of generality we assume that no initial state of M has any incoming transition. We copy all in-coming transitions to the final states of M to the initial state (this makes it non-deterministic).
So the machine after consuming a string of $L(N) \subseteq \Sigma^*$ will start afresh from the initial state and this continues ad infinitum. Formally the Büchi automaton $B = (Q_b, I_b, F_b, T_b)$ is

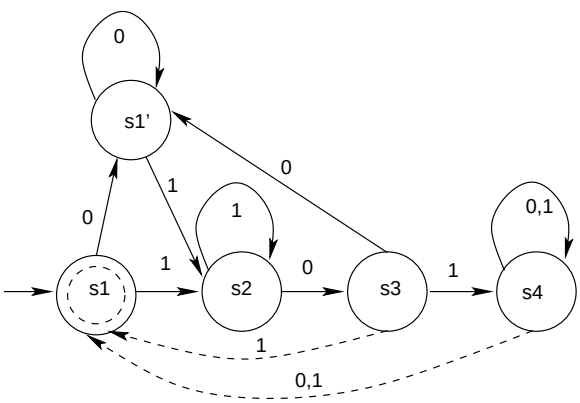
1. $Q_b = Q$, $I_b = \{s_0\}$, $F_b = I_b$.
2. $T_b = T \cup \{(t, \sigma, s_0) :$ if $(t, \sigma, f) \in T$, where $f \in F\}$.

A successful run is as follows:

$$s_0, q_{11}, \cdots, q_{1k_0}, s_0 q_{21}, \cdots, q_{2k_1}, s_0, \cdots$$

The strings consumed in the portion $s_0, q_{j1}, \cdots, q_{jk_j}, s_0$ belongs to L because any $x \in \Sigma^*$ that drives the machine from the initial state s_0 to the state q_{jk_j} and then to s_0 again, also drives M form s_0 to a *final state* of M. QED.

Example 5. The Büchi automaton corresponding to L^ω of Example. 1.2.4 is



Lemma 6. If L_1 and L_2 are regular languages over Σ^*, then $L_1 L_2^\omega$ is Büchi recognisable.
Proof: We have two dfa's $M_1 = (Q_1, s_1, F_1, T_1)$ and $M_2 = (Q_2, s_2, F_2, T_2)$ for L_1 and L_2 respectively. We assume that $Q_1 \cap Q_2 = \emptyset$ and the initial state of Q_2 does not have any in-coming transition.

We know how to construct a Büchi automaton corresponding to L_2^ω. Let it be $B_3 = (Q_2 \cup \{s_2'\}, \{s_2\}, \{s_2\}, T_3)$, where $s_2' \notin Q_1 \cup Q_2$, and T_3 is defined as earlier. Now we can construct a Büchi automaton $B = (Q, I, F, T)$ for $L_1 L_2^\omega$ as follows:

1. $Q = Q_1 \cup Q_2 \cup \{s_2'\}$,
2. $I = \{s_1\}$, $F = \{s_2\}$,
3. $T_1 \cup T_3 \cup \{(t, \sigma, s_2) :$ if $(t, \sigma, f) \in T_1$ and $f \in F_1\}$.

We add transitions (t, σ, s_2) where $(t, \sigma, f) \in T_1$ and $f \in F_1$. An infinite string $\alpha \in \Sigma^\omega$ drives B on a *successful* run if its initial segment is a string of L_1, and then there is an infinite repetition of strings of L_2. QED.

Theorem 7. (Büchi) An ω-language $L \subseteq \Sigma^\omega$ is Büchi recognisable if and only if $L = \bigcup_{i=1}^{k} L_i K_i^\omega$, where $L_i, K_i \subseteq \Sigma^*$ are regular languages.
Proof: We already have done the ground work for the proof of the '*if* ($\Leftarrow$) part. We know that $L_i K_i^\omega$ is Büchi recognisable and Büchi recognisable languages are closed under union.

To prove the '*only if* ($\Rightarrow$) part we proceed as follows. Let $B = (Q, I, F, T)$ be a Büchi automaton such that $L = L(B)$. For every $p, q \in Q$, we define an nfa $M_{p,q} = (S, p, q, T)$ and $L_{p,q} = L(M_{p,q}) \subseteq \Sigma^*$. As Q is finite, there are finite number of such languages. We define the following ω-language using these regular languages.

$$L' = \bigcup_{p \in I, q \in F} L_{p,q} L_{q,q}^\omega.$$

We claim that $L = L'$.
Suppose $\alpha \in L'$, then there is some $p \in I$ and $q \in F$ such that $\alpha \in L_{p,q} L_{q,q}^\omega$. So this α drives B starting from one of its start state p to a final state q and then in the run the state q appears infinitely often. So $\alpha \in L$ i.e. $L' \subseteq L$.
Conversely, suppose $\alpha \in L$. So there is a *successful run* of the machine B on α. The machine starts from a start state p, and in the infinite sequence of states, at least one of the final states q appears infinitely often. So the run may be viewed as $p \cdots q \cdots q \cdots q \cdots$. The initial segment of α that drives B from state p to state q, so the segment belongs $L_{p,q}$ and each portion of α that drives the machine from state q to q belongs to $L_{q,q}$. So $\alpha \in L_{p,q} L_{q,q}^\omega$ i.e. $L \subseteq L'$. QED.

1.2.5 Complement of Büchi Language

We know that $\Sigma^* \setminus L$ is a regular language if and only if $L \subseteq \Sigma^*$ is a regular language. The collection of regular languages over Σ is closed under complementation.
The question is whether the collection of Büchi recognisable languages are closed under complementation. The answer is positive but the proof is more involved than the case of regular languages. The main reason is that as a model *deterministic Büchi* automaton is strictly less powerful than its non-deterministic counterpart.

Let $\alpha \in \Sigma^\omega$. We define $\alpha(m, n)$ or $\alpha_{m,n}$ as the string $\alpha(m)\alpha(m+1)\cdots\alpha(n-1) \in \Sigma^*$, where m and n are two positive integers and $m < n$. Similarly, $\alpha(m, \infty)$ is the ω-string $\alpha(m)\alpha(m+1)\cdots \in \Sigma^\omega$.

Let $L \subseteq \Sigma^*$. We define $\overrightarrow{L} \subseteq \Sigma^\omega$ as follows:

$$\overrightarrow{L} = \{\alpha \in \Sigma^\omega : \alpha(1, n) \in L \text{ for infinitely many } n \in \mathbb{N}\}.$$

So any string $\alpha \in \overrightarrow{L}$ has infinite number of prefixes from L. This may be viewed as some kind of *limit* operator.

1.2.6 Deterministic Büchi Automata

We have the following theorem that characterises the language recognised by a deterministic automaton.

Theorem 8. An ω-language L over Σ is recognisable by a *deterministic* Büchi automaton if and only if there is a regular language L' such that $L = \overrightarrow{L'}$.

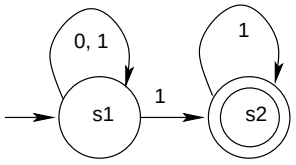
Proof: Let $B = (Q, s_0, F, T)$ be a *deterministic* Büchi automaton and $L_\omega(B)$ be the ω-language accepted by B. We may view B as an dfa accepting $L(B) \subseteq \Sigma^*$. We shall show that $\overrightarrow{L(B)} = L_\omega(B)$.

Let $\alpha = \alpha_1\alpha_2\cdots \in L_\omega(B)$ and the corresponding successful run be $r = s_0s_1\cdots$. In r there are infinitely many occurrences of final states of B. So there are infinite number of prefixes of α belonging to $L(B)$ i.e. $\alpha \in \overrightarrow{L(B)}$. So $L_\omega(B) \subseteq \overrightarrow{L(B)}$.

Let $\alpha = \alpha_1\alpha_1\cdots \in \overrightarrow{L(B)}$. As the machine is deterministic there is only one run and there are infinite number of prefixes of α belonging to $L(B)$. So some final state is visited infinitely often implies that $\alpha \in L_\omega(B)$ i.e. $\overrightarrow{L(B)} \subseteq L_\omega(B)$. QED.

Determinism is important in the second part of the proof. If the machine is non-deterministic, it is possible to have an α with infinite number of prefixes in the regular language, but no run of the corresponding Büchi automaton on α have infinitely many final states. Next example makes it clear.

Example 6. Consider the following non-deterministic automaton N.



An $\alpha = (01)^\omega$ has infinitely many prefixes, $01, 0101, 010101, \cdots$ from the regular language of N. But there is no successful run of N as a Büchi automaton on this α.

Theorem 9. There is a Büchi recognisable language that cannot be accepted by a *deterministic* Büchi automaton.

Proof: We fix the alphabet $\Sigma = \{0, 1\}$. Define the ω-language

$$L = \{\alpha \in \Sigma^\omega : \text{ there are finite number of occurrences of 0 in } \alpha\}.$$

$L = \Sigma^*\{1\}^\omega$ and is recognised by the non-deterministic Büchi automaton of example (1.2.6). We prove by contradiction that L cannot be recognised by a *deterministic* Büchi automaton.

Assume that L is recognised by a *deterministic* Büchi automaton. So there is a regular language $L' \subseteq \Sigma^*$ such that $\overrightarrow{L'} = L$ (deterministic Büchi language).

It is clear that $1^\omega \in L$, so there is some positive integer n_1, such that $1^{n_1} \in L'$ ($\overrightarrow{L'} = L$ and there are infinite number of such prefixes). But the infinite string $1^{n_1}01^\omega \in L$ as it has finite number of 0's. So again there is some positive integer n_2 such that $1^{n_1}01^{n_2} \in L'$. By a similar argument we have $1^{n_1}01^{n_2}01^\omega \in L$ and there is a positive integer n_3 such that $1^{n_1}01^{n_2}01^{n_3} \in L'$.

Continuing this way we construct an ω-string

$$\alpha = 1^{n_1}01^{n_2}01^{n_3}0\cdots01^{n_i}0\cdots, \text{ where each } n_i \geq 1,$$

such that α has infinite number of prefixes belonging to L'. So $\alpha \in \overrightarrow{L'} = L$. But that is contradictory as it has infinite number of 0's. QED.

Corollary 10. There is a deterministic Büchi automaton B such that $\Sigma^\omega \setminus L_\omega(B)$ is not recognised by any deterministic Büchi automaton.

Proof: We define $L' = \{w0 : \ w \in \Sigma^*, 0 \in \Sigma\} \subseteq \{0, 1\}^*$. It is not difficult to construct a dfa that recognises L'. An element $\alpha \in \omega$-language $\overrightarrow{L'}$ has infinitely many prefixes of in L' i.e. α has infinite number of 0's in it. The language $L = \Sigma^\omega \setminus \overrightarrow{L'} = \{\alpha \in \Sigma^\omega :$ only finitely many prefixes of α are in L'. So there are finite number of 0's in α. But we have already proved in theorem (1.2.6) that there is no deterministic Büchi automaton for L. QED.

1.2.7 Equivalences Classes on NFA

Definition 2: Let $B = (Q, I, F, T)$be a Büchi automaton and $p, q \in Q$. We define $L_{p,q} \subseteq \Sigma^*$ as the collection of all strings that drives the machine from state p to state q. This is the language $L(M_{p,q})$, where $M_{p,q} = (Q, \{p\}, \{q\}, T)$. Similarly for $A \subseteq Q$, we define $L^A_{p,q} \subseteq \Sigma^*$ as the collection of all strings that drives the machine (i) from state p to state q, and (ii) one of the states in the run is in A. $L^A_{p,q}$ may be viewed as $\bigcup_{r\in A}(L_{p,r}L_{r,q})$. So both these languages are regular.

We define a binary relation $\sim_B$ on Σ^* in the following way. Let $x, y \in \Sigma^*$, then $x \sim_B y$ if

1. for all pairs of states $p, q \in Q$, the input x drives the machine from p to q ($p \rightarrow_x q$) if and only if y drives the machine from p to q ($p \rightarrow_y q$), and also

2. for all pairs of states $p, q \in Q$, the input x drives the machine from p to q through a final state ($p \rightarrow^F_x q$) if and only if y drives the machine from p to q through a final state ($p \rightarrow^F_y q$).

It is not difficult to prove that $\sim_B$ is an *equivalence* relation. i Afterward we shall characterise the equivalence classes of Σ^* under $\sim_B$.
Definition 3: An equivalence relation on Σ^* is said to be a *congruence relation* if for all $x, y, z \in \Sigma^*$, $x \sim_B y$ implies that $xz \sim_B yz$.
Lemma 11. Given a Büchi automata $B = (Q, I, F, T)$, the equivalence relation $\sim_B$ is a *congruence* relation.
Proof: Suppose $x, y, z \in \Sigma^*$ and $x \sim_B y$ i.e. for all $p, q \in Q$, (i) $p \rightarrow_x q$ if and only if $p \rightarrow_y q$, and (ii) $p \rightarrow^F_x q$ if and only if $p \rightarrow^F_y q$.
$p \rightarrow_{xz} r$ if and only if there is some $q \in Q$, so that $p \rightarrow_x q$ and $q \rightarrow_z r$ if and only if $p \rightarrow_y q$ and $q \rightarrow_z r$ i.e. $p \rightarrow_{yz} r$.

If $p \rightarrow^F_{xz} r$, then there are two possibilities, (i) there is some $q \in Q$, so that $p \rightarrow^F_x q$ and $q \rightarrow_z r$, or (ii) there is some $q \in Q$, so that $p \rightarrow_x q$ and $q \rightarrow^F_z r$. In the first case we have $p \rightarrow^F_y q$ and $q \rightarrow_z r$ i.e. $p \rightarrow^F_{yz} r$. And in the second case we have $p \rightarrow_y q$ and $q \rightarrow^F_z r$ i.e. $p \rightarrow^F_{yz} r$. QED.

Lemma 12. Given a Büchi automata, the congruence relation $\sim_B$ partitions Σ^* in finite number of equivalence classes i.e $\sim_B$ has finite index.
Proof: We define two functions $e_1, e_2 : \Sigma^* \rightarrow 2^{Q\times Q}$ in the following way:

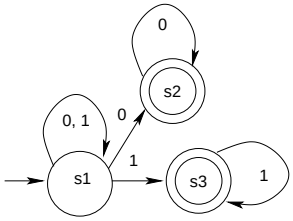
$$e_1(x) = \{(p, q) \in Q \times Q : \ p \rightarrow_x q\},$$

and

$$e_2(x) = \{(p, q) \in Q \times Q : \ p \rightarrow^F_x q\}.$$

Let $A \subseteq Q \times Q$. The preimage of A, $e_1^{-1}(A) = L_A \subseteq \Sigma^*$, if non-empty, is such that for any two $x, y \in L_A$, and for any $p, q \in Q$, $p \rightarrow_x q$ if and only if $p \rightarrow_y q$. So the preimages of two subsets of $Q \times Q$ are disjoint subsets of Σ^*. Number of subsets of $Q \times Q$ are finite, so the partitions are also finite.
Similarly the preimage of $A \subseteq Q \times Q$ under e_2, $e_2^{-1}(A) = L'_A \subseteq \Sigma^*$, if non-empty, is such that for any two $x, y \in L'_A$, and for any $p, q \in Q$, $p \rightarrow^F_x q$ if and only if $p \rightarrow^F_y q$. This also divides Σ^* in finite number of partitions.
Refinement of these two partitions gives the equivalence classes of $\sim_B$. So the number of equivalence classes are finite. QED.

It is well known (Mihill-Nerode theorem) that if there is an equivalence relation of finite index over Σ^*, and the relation is a congruence relation, then the equivalence classes are regular languages.
Example 7. Consider the following Büchi automaton $B = (Q, I, F, T)$ over the alphabet $\Sigma = \{0, 1\}$.



The map $e_1 : \Sigma^* \to 2^{Q\times Q}$ is as follows:

$$\begin{aligned} e_1(\{\varepsilon\}) &= \{(1,1),(2,2),(3,3)\},\\ e_1(\{0^n :\ n \geq 1\}) &= \{(1,1),(1,2),(2,2)\},\\ e_1(\{1^n :\ n \geq 1\}) &= \{(1,1),(1,3),(3,3)\},\\ e_1(\{x0 :\ x \in \Sigma^*1\Sigma^*\}) &= \{(1,1),(1,2)\},\\ e_1(\{x1 :\ x \in \Sigma^*0\Sigma^*\}) &= \{(1,1),(1,3)\}, \end{aligned}$$

So the map e_1 induces an equivalence relation of finite index on Σ^*. The classes are, $E = \{\varepsilon\}$, $A = \{0^n :\ n \geq 1\}$, $B = \{1^n :\ n \geq 1\}$, $C = \{x0 :\ x \in \Sigma^*1\Sigma^*\}$, $D = \{x1 :\ x \in \Sigma^*0\Sigma^*\}$. Similarly, map $e_2 : \Sigma^* \to 2^{Q\times Q}$ is as follows:

$$\begin{aligned} e_2(\{\varepsilon\}) &= \{(2,2),(3,3)\},\\ e_2(\{0^n :\ n \geq 1\}) &= \{(1,2),(2,2)\},\\ e_2(\{1^n :\ n \geq 1\}) &= \{(1,3),(3,3)\},\\ e_2(\{x0 :\ x \in \Sigma^*1\Sigma^*\}) &= \{(1,2)\},\\ e_2(\{x1 :\ x \in \Sigma^*0\Sigma^*\}) &= \{(1,3)\}, \end{aligned}$$

This map also induces the same equivalence relation. So both the map together induces the same equivalence relation of finite index and the equivalence classes are A, B, C, D, E.

Lemma 13. Given a Büchi automaton $B = (Q, I, F, T)$, the equivalence class $[x]_{\sim_B}$ ($[x]$ in short) is the intersection of all languages of the form $L_{p,q}$, $L^F_{p,q}$, $\Sigma^* \setminus L_{p,q}$, $\Sigma^* \setminus L^F_{p,q}$ which contain x.

Proof: Let T_x be the intersection of all the sets of the form $L_{p,q}$, $L^F_{p,q}$, $\Sigma^* \setminus L_{p,q}$ and $\Sigma^* \setminus L^F_{p,q}$, containing x.

If $y \in [x]$, then $y \sim_B x$ and y belongs to all sets of the above form containing x[6]. So y is in the intersection T_x.

Suppose $y \in T_x$, the intersection of all the above mentioned sets. If $p \to_x q$, then $x \in L_{p,q}$. By our definition of $T_x \subseteq L_{p,q}$. So $y \in L_{p,q}$, implies that $p \to_y q$. If $p \not\to_x q$, then $x \in \Sigma^* \setminus L_{p,q}$. Again by definition $y \in T_x \subseteq \Sigma^* \setminus L_{p,q}$ i.e. $p \not\to_y q$. Similarly, if $p \to^F_x q$, then $p \to^F_y q$. And if $p \not\to^F_x q$ then $p \not\to^F_y q$. QED.

It is clear that given a Büchi automaton $B = (Q, I, F, T)$, the corresponding nfa induces an equivalence (congruence) relation $\sim_B$ on Σ^*, where each equivalence class is regular.

1.2.8 Complementation Theorem

The plan for the construction of a Büchi automaton that recognises the complement of a Büchi recognisable language is as follows: Let B be a Büchi automaton. We have already shown how a congruence relation of finite index over Σ^* can be constructed using B as an nfa. We also know that these equivalence classes are regular languages. So for any two equivalence classes L_1 and L_2, the language $L_1L_2^\omega$ is Büchi recognisable. We shall prove that if $L_1L_2^\omega \cap L_\omega(B) \neq \emptyset$, then $L_1L_2^\omega \subseteq L_\omega(B)$. We also prove that each $\alpha \in L_\omega(B)$ belongs to some $L_1L_2^\omega$, where L_1 and L_2 are two equivalence classes.

So the complement language $\Sigma^\omega \setminus L_\omega(B)$ is the union of all $L_1L_2^\omega$ such that $L_1L_2^\omega \cap L_\omega(B) = \emptyset$, where L_1 and L_2 are equivalence classes.

Example 8. In the *example* (**1.2.7**), the equivalence classes are $E = \{\varepsilon\}$, $A = \{0^n :\ n \geq 1\}$, $B = \{1^n :\ n \geq 1\}$, $C = \{x0 :\ x \in \Sigma^*1\Sigma^*\}$, $D = \{x1 :\ x \in \Sigma^*0\Sigma^*\}$. The Büchi recognisable language $L_\omega(B) = A^\omega \cup B^\omega \cup CA^\omega \cup DA^\omega \cup CB^\omega \cup DB^\omega$.

The complement language consists of AC^ω, BC^ω etc.

Definition 4: A congruence relation E on Σ^* *saturates* an ω-language L if for any pair of equivalence classes L_1 and L_2, whenever $L_1L_2^\omega \cap L \neq \emptyset$, then $L_1L_2^\omega \subseteq L$.

Lemma 14. Let $B = (Q, I, F, T)$ be a Büchi automaton and $L = L_\omega(B)$. The congruence relation $\sim_B$ defined on Σ^* *saturates* L. In other words, for any pair

[6] Given a $p, q \in Q$, $x \in L_{p,q}$ if and only if $y \in L_{p,q}$. Similarly $x \in L^F_{p,q}$ if and only if $y \in L^F_{p,q}$.

of equivalence classes L_1 and L_2, if there an $\alpha \in L_1 L_2^\omega$ that has a successful run on B, then all strings of $L_1 L_2^\omega$ have successful runs on B i.e. $L_1 L_2^\omega \subseteq L$.
Proof: Let L_1 and L_2 be equivalence classes and $\alpha = uv_1v_2 \cdots \in L_1 L_2^\omega \cap L_\omega(B)$, where $u \in L_1$, $v_i \in L_2$, $i = 1, 2, \cdots$. So there is a successful run $r = s_0 s_1 s_2 \cdots$ of B on α, where $s_0 \in I$ and $Inf(r) \cap F \neq \emptyset$. Given α and its successful run r, we get the following infinite sequence of states: $s_0 q_1 q_2 \cdots$, such that

1. u drives the machine from the start state s_0 to q_1, $s_0 \rightarrow_u q_1$, and
2. every v_i drives the machine from q_i to q_{i+1}, i.e. $q_i \rightarrow_{v_i} q_{i+1}$, for $i = 1, 2, \cdots$.

As r is a successful run, there are infinitely many final states. So there are infinitely many v_i's such that v_i drives the machine from q_i to q_{i+1} through some final state, $q_i \rightarrow^F_{v_i} q_{i+1}$.

Take any arbitrary word $\beta = u'v_1'v_2' \cdots \in L_1 L_2^\omega$, where $u' \in L_1$ and $v_i' \in L_2$. Moreover, $u \sim_B u'$ and $v_i \sim_B v_i'$, as L_1 and L_2 are equivalence classes. So $s_0 \rightarrow'_u q_1$ and $q_i \rightarrow_{v_i} q_{i+1}$ for $i = 1, 2, \cdots$. Also if $q_i \rightarrow^F_{v_i} q_{i+1}$ then so is $q_i \rightarrow^F_{v_i'} q_{i+1}$. We conclude that β has a *successful* run on B. QED.

Now we go to prove the second part, where we claim that every infinite string α belongs to some $L_1 L_2^\omega$, where L_1 and L_2 are equivalence classes. This will enable us to claim that $L_\omega(B) = \bigcup L_1 L_2^\omega$, where the union is over all pairs of equivalence classes L_1 and L_2 such that $L_\omega(B) \cap L_1 L_2^\omega \neq \emptyset$. So the complement language $\Sigma^\omega \setminus L_\omega(B) = \bigcup L_1 L_2^\omega$, where the union is over all pairs of equivalence classes L_1 and L_2 such that $L_\omega(B) \cap L_1 L_2^\omega = \emptyset$. Hence the complement language is also Büchi recognisable.

Let $B = (Q, I, F, T)$ be a Büchi automaton and $\sim_B$ be the equivalence relation defined earlier.
Let $\alpha = \alpha_1 \alpha_2 \cdots \in \Sigma^\omega$. The symbol at the i^{th}-*position* of α is α_i.
Let k, l and m be positions in α so that $k, l < m$. We say that positions k and l *merge* at m if $\alpha_{k,m} \sim_B \alpha_{l,m}$ i.e. the string $\alpha_k \cdots \alpha_{m-1}$ and $\alpha_l \cdots \alpha_{m-1}$ belong to the same equivalence class of $\sim_B$.
It is also clear that if $\alpha_{k,m} \sim_B \alpha_{l,m}$, then $\alpha_{k,m'} \sim_B \alpha_{l,m'}$, where $m' \geq m$ i.e. if the positions k and l merge at m, then they merge at all positions after this as $\sim_B$ is a congruence relation.
Definition 5: Given a Büchi automata $B = (Q, I, F, T)$ and an ω-string α we define a binary relation on the positions in the string, or on the set of positive integers $\mathbb{N}$ as follows: $k \sim_\alpha l$, if there is a position $m > k, l$, such that $\alpha_{k,m} \sim_B \alpha_{l,m}$. We may also write $k \sim_\alpha l(at\ m)$ or simply $k \sim_\alpha l(m)$.

The binary relation is clearly *reflexive* and *symmetric*. We prove that it is transitive.
Let $j, k, l \in \mathbb{N}$ and $j \sim_\alpha k(m)$, where m is the smallest merge position and $k \sim_\alpha l(n)$, where n is the smallest merge position. But we know that $j \sim_\alpha k$ at any merge position $m' \geq m$ and $k \sim_\alpha l$ at any merge position $n' \geq n$. Let $p = max(m, n)$, then $\alpha_{j,p} \sim_B \alpha_{k,p}$ and $\alpha_{k,p} \sim_B \alpha_{l,p}$. So we have $\alpha_{j,p} \sim_B \alpha_{l,p}$ i.e. $j \sim_\alpha l$ at p. So the binary relation on $\mathbb{N}$ is an equivalence relation.
Lemma 15. Given a Büchi automaton $B = (Q, I, F, T)$ and an ω-string α, the equivalence relation $\sim_\alpha$ has finite index i.e. it partitions $\mathbb{N}$ in finite number of equivalence classes.
Proof: The proof is by contradiction. Assume that $\sim_\alpha$ creates infinitely many equivalence classes. So there are infinitely many positions $k_1 < k_2 < \cdots$ such that for every k_i and k_j $(i \neq j)$, $k_i \not\sim_\alpha k_j$ i.e. there is no position $m > k_i, k_j$, such that $\alpha_{k_i,m} \sim_B \alpha_{k_j,m}$.
But we know that $\sim_B$ is of finite index. Let the number of equivalence classes of Σ^* under $\sim_B$ be n. Consider positions $k_1 < k_2 < \cdots < k_{n+1}$. For all position $m > k_1, \cdots, k_{n+1}$, by our assumption none of $\alpha_{k_1,m}, \cdots, \alpha_{k_{n+1},m}$ are equivalent. But that is impossible as there are only n equivalence classes of Σ^* under $\sim_B$. The is a contradiction and hence the index of $\sim_\alpha$ is finite. QED.

Now we know that given a Büchi automaton B and an ω-word α, the equivalence relation $\sim_\alpha$ partitions $\mathbb{N}$ (positions in α) in finite number of equivalence classes. So there must be an equivalence class with infinite number of positions

(positive integers) in it.
Let points (positions) in such an equivalence class be $k_0 < k_1 < \cdots$ where we may assume that $1 < k_0$. Each k_i, k_j in this sequence of positions are $\sim_\alpha$ equivalent with respect to some position $m > k_i, k_j$. We consider the following sequence of finite words: $\alpha_{k_0,k_1}, \alpha_{k_0,k_2}, \cdots$. Again due to finite number of equivalence classes of Σ^* under $\sim_B$, there will be an infinite subsequence of $\alpha_{k_0,k_1}, \alpha_{k_0,k_2}, \cdots$, whose strings are $\sim_B$ equivalent. We pick-up those k_j's (positions where α_{k_0,k_j}'s are $\sim_B$ equivalent and belong to an equivalence class of infinite size).

So we have an infinite sequence of positions, $k_0 = l_0, l_1, l_2, \cdots$ such that $l_i \sim_\alpha l_j$ for all i, j and $\alpha_{l_0,l_i} \sim_B \alpha_{l_0,l_j}$.

We claim now that there is an infinite sequence of positions $p_0, p_1, p_2, \cdots$ such that for all $i < j$, p_i *merges* with p_j at p_{j+1}. The sequence is constructed as follows:

1. *Basis:* $p_0 = l_0, p_1 = l_1$.
2. *Inductive step:* If we have already identified the positions $p_0, p_1, \cdots, p_n$, where $p_i \sim_\alpha p_j$ at position p_{j+1}, $j = 0, \cdots, n-1$.
   There is a position m such that $p_i \sim_\alpha p_j(m)$, for all i, j, $0 \leq i, j \leq n$. We choose the smallest such m, and take l_k as p_{n+1} such that $m \leq l_k$.

So we have the infinite sequence of positions, $p_0, p_1, p_2, \cdots$, satisfying the following conditions:

1. $1 < p_0$,
2. $\alpha_{p_0,p_i} \sim_B \alpha_{p_0,p_j}$ for all i, j.
3. For all positions p_i and p_j, $i < j$, $p_i \sim_\alpha p_j$, merges at p_{j+1}.

We define two subsets of Σ^* as follows:

$$L_1 = \{x \in \Sigma^* : \ x \sim_B \alpha_{1,p_0}\}, \ L_2 = \{x \in \Sigma^* : \ x \sim_B \alpha_{p_0,p_1}\}.$$

Clearly L_1 and L_2 are two equivalence classes under $\sim_B$.
Lemma 16. The ω-string $\alpha \in L_1 L_2^\omega$.
Proof: We consider the position sequence $p_0 < p_1 < \cdots$ generated earlier. We know that $\alpha_{p_i,p_{i+1}} \sim_B \alpha_{p_0,p_{i+1}}$ (condition (3)). We also have $\alpha_{p_0,p_1} \sim_B \alpha_{p_0,p_i} \sim_b \alpha_{p_0,p_{i+1}}$ (condition (2)). So, $\alpha_{p_0,p_1} \sim_B \alpha_{p_i,p_{i+1}}$ i.e. the string $\alpha_{p_i,p_{i+1}} \in L_2$.
Hence the infinite string α can be written as

$$\alpha = \alpha_{1,p_0} \alpha_{p_0,p_1} \alpha_{p_1,p_2} \cdots,$$

where $\alpha_{1,p_0} \in L_1$ and $\alpha_{p_i,p_{i+1}} \in L_2$ i.e. $\alpha \in L_1 L_2^\omega$. QED.

Theorem 17. Let B be a Büchi automaton over Σ. The language $\Sigma^\omega \setminus L_\omega(B)$ is Büchi recognisable.
Proof: We have already proved that $L_1 L_2^\omega \subseteq L_\omega(B)$, where L_1 and L_2 are equivalence classes generated by $\sim_B$, and $L_1 L_2^\omega \cap L_\omega(B) \neq \emptyset$. Again from the previous lemma we know that if $\alpha \in L_\omega(B)$, then $\alpha \in L_1 L_2^\omega$ for some equivalence classes. So we conclude that $\bigcup L_1 L_2^\omega = L_\omega(B)$, where the union is over all pairs of equivalence classes L_1 and L_2 such that $L_1 L_2^\omega \cap L_\omega(B) \neq \emptyset$.

In the previous theorem we proved that any ω-string is a member of some $L_1 L_2^\omega$. So, the complement language, $\Sigma^\omega \setminus L_\omega(B) = \bigcup L_1 L_2^\omega$, where L_1 and L_2 are equivalence classes generated by $\sim_B$, and the union is over all pairs of such L_1 and L_2 such that $L_1 L_2^\omega \cap L_\omega(B) = \emptyset$. QED.
So the conclusion is that the collection of Büchi recognisable languages are closed under complementation.

References

[BKAN] *Automata Theory and its Applications*, by Bakhadyr Khoussainov and Anil Nerode, Birkhäuser, Springer Int. Ed., 2010, ISBN 978-81-8489-661-9.

[DPJEP] *Infinite Words Automat, Semigroups, Logic and Games*, by Dominique Perrin and Jean-Éric Pin, Elsevier, Academic Press, 2004, ISBN 0-12-532111-2.