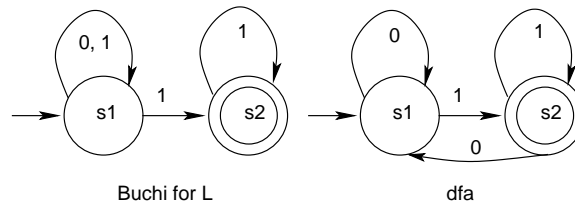


## 1 Müller and Rabin Automata

Müller defined the first deterministic finite automaton on  $\omega$ -strings with a different acceptance condition. As the machine is deterministic, the language complementation problem is easier.

### 1.1 Müller Automaton

We already know that the  $\omega$ -language  $L = \{\alpha \in \{0, 1\}^\omega : \text{where } \alpha \text{ has finite number of 0's}\}$  is not recognised by any deterministic Büchi automaton. In the following diagram, we consider the Büchi automaton for  $L$  and a dfa.



The deterministic machine has two possible types *successful runs*. In one of those the state  $s_2$  only will occur infinitely often, and in the other both  $\{s_1, s_2\}$  will occur infinitely often. If we can exclude those runs  $r$ , where  $Inf(r) = \{s_1, s_2\}$ , the deterministic machine can accept  $L$ . This possibly was the motivation of the following definition of Müller automaton.

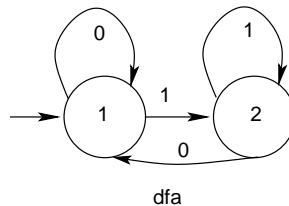
**Definition 1:** A Müller automaton  $M$  over the alphabet  $\Sigma$  is a 4-tuple,  $M(Q, s_0, F, T)$ , where

- $Q$  is a finite set of states,
- $s_0 \in Q$  is the initial or start state,
- $F \subseteq 2^Q$  is the collection of final state sets,
- $T : Q \times \Sigma \rightarrow Q$ , is the transition table.

Given an  $\alpha \in \Sigma^\omega$ , the *run* of  $M$  is defined as usual. As the machine is deterministic (we assume it to be complete), there is a unique run on each  $\alpha$ . The *run*  $r$  on an  $\omega$ -word  $\alpha$  is said to be *successful* if  $Inf(r) \in F$  i.e. the collection of all states appearing infinitely often is one of the sets of final states.

It is important to appreciate that in the successful run  $r$  of a Müller automaton on the input  $\alpha$ , all states except a finite number of initial states are all elements of  $Inf(r) \in F$ .

**Example 1.** Consider the following Müller automaton,  $M = (\{1, 2\}, 1, \{\{2\}\}, T)$ , where the transition table is shown in the figure.



If  $\alpha \in \{0, 1\}^*1^\omega$ , then  $Inf(r_\alpha) = \{2\}$ . So there is a successful run on  $M$  on  $\alpha$ . On the other hand, if  $\alpha \in \{00^*11^*\}^\omega$ , then  $Inf(r_\alpha) = \{1, 2\}$ , and there is no successful run on  $\alpha$ .

The language of  $M$  is the collection of the  $\omega$ -strings for which there is the successful run. If we change the set of final state sets, the accepted language also changes.

Example 2. If we modify the Müller automaton  $M$  of (1.1) only by changing the final state set to  $\{\{1\}\}$ , then the language automaton  $M'$ ,  $L_\omega(M') = \{\alpha \in \{0,1\}^\omega : \alpha \text{ has finite number of 1's}\}$ .

Example 3. If we modify the same machine (1.1) by making the set of final state sets to  $\{\{1,2\}, \{2\}\}$ . Then any  $\omega$ -string of its language have either finite number of 0's and infinitely many 1's, or both 0's and 1's are occur infinitely many times. Strings that does not have successful run has finite number of 1's and infinitely many 0's.

## 1.2 Closure properties

The collection of  $\omega$ -languages recognisable by Müller automaton is closed under union, intersection and complementation.

Theorem 1. Let  $M_1 = (Q_1, s_{01}, F_1, T_1)$  and  $M_2 = (Q_2, s_{02}, F_2, T_2)$  be two Müller automaton on the alphabet  $\Sigma$ .

1.  $L_\omega(M_1) \cup L_\omega(M_2)$  is recognisable by some Müller automaton.
2.  $\Sigma^\omega \setminus L_\omega(M_1)$  is recognisable by some Müller automaton.
3.  $L_\omega(M_1) \cap L_\omega(M_2)$  is recognisable by some Müller automaton.

**Proof:**

1. We define  $M_\cup = (Q_\cup, s_{0\cup}, F_\cup, T_\cup)$  as follows:

- $Q_\cup = Q_1 \times Q_2$ ,
- $s_{0\cup} = (s_{01}, s_{02})$ ,
- $F_\cup = \{\{(f_1, q_1), \dots, (f_k, q_k)\} : \{f_1, \dots, f_k\} \in F_1 \wedge q_1, \dots, q_k \in Q_2\} \cup \{\{(p_1, r_1), \dots, (p_l, r_l)\} : \{r_1, \dots, r_l\} \in F_2 \wedge p_1, \dots, p_l \in Q_1\}$ ,
- $T_\cup : ((p_1, p_2), \sigma) \mapsto (q_1, q_2) : \text{where } T_1(p_1, \sigma) \mapsto q_1 \text{ and } T_2(p_2, \sigma) \mapsto q_2$ .

The transition table is such that both the machines will run in parallel on the input.

Let  $\alpha = \alpha_1\alpha_2 \dots \in \Sigma^\omega$  has the successful run  $r_1 = s_{01}s_{11}s_{21} \dots$  on  $M_1$ . So  $Inf(r_1) \in F_1$ . The run of  $M_\cup$  on  $\alpha$  is  $r = (s_{01}, s_{02})(s_{11}, s_{12})(s_{21}, s_{22}) \dots$ . The set  $Inf(r) = \{(s_{i1}, s_{i2}) : s_{i1} \in F_1 \wedge s_{i2} \in Q_2\} \in F_\cup$ . So  $\alpha \in L_\omega(M_\cup)$  i.e.  $L_\omega(M_1) \subset L_\omega(M_\cup)$ . Similarly we can show that  $L_\omega(M_2) \subset L_\omega(M_\cup)$  i.e.  $L_\omega(M_1) \cup L_\omega(M_2) \subset L_\omega(M_\cup)$ .

In the other direction, if  $\alpha$  has a successful run  $r$  on  $M_\cup$ , then  $Inf(r) \in F_\cup$ , and either the collection of the first components of the states in  $Inf(r)$  is in  $F_1$ , or the collection of the second components is in  $F_2$ . So  $\alpha$  belongs to either  $L_\omega(M_1)$  or  $L_\omega(M_2)$ .

2. We define  $M_c = (Q_1, s_{01}, F_c, T_1)$  where  $F_c = \{A \subseteq Q_1 : \text{where } A \notin F_1\}$ . If  $\alpha \in \Sigma^\omega$  does not belong to  $L_\omega(M_1)$ , and  $r$  is the run of  $M_1$  on  $\alpha$ , then  $Inf(r) \notin F_1$ . So it must belong to  $F_c$ . So  $L_\omega(M_c) = \Sigma^\omega \setminus L_\omega(M_1)$ .
3. The proof of closure of intersection follows from the De Morgan's law and previous two proofs.  $L_\omega(M_1) \cap L_\omega(M_2) = \Sigma^\omega \setminus ((\Sigma^\omega \setminus L_\omega(M_1)) \cup (\Sigma^\omega \setminus L_\omega(M_2)))$ .

QED.

## 1.3 Characterisation of Müller Automata

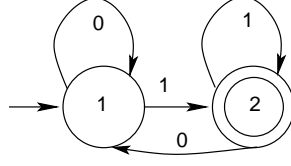
Our next question is about the relative power of Büchi and Müller automata. It is not difficult to guess that every language accepted by a deterministic Büchi automaton will have an equivalent Müller automaton. This we present in the form of a lemma.

Lemma 2. Let  $B = (Q, s_0, F, T)$  be a deterministic Büchi automaton. There is an equivalent Müller automaton  $M$  so that  $L_\omega(B) = L_\omega(M)$ .

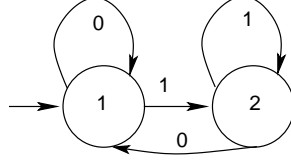
**Proof:** We construct the Müller automaton  $M = (Q, s_0, F', T)$ , where  $F' = \{A \subseteq Q : A \cap F \neq \emptyset\}$ .

An  $\alpha \in \Sigma^\omega$  has the successful run  $r = s_0 s_1 \dots$  on  $B$  if and only if  $\text{Inf}(r) \cap F \neq \emptyset$  if and only if  $\text{Inf}(r) \in F'$  of  $M$  (by definition of  $F'$ ) if and only if  $r$  is the successful run of  $M$  on  $\alpha$ . QED.

Example 4. The deterministic Büchi automaton  $B = (\{1, 2\}, 1, \{2\}, T)$ , where the transition function is given by the following diagram,



Has the equivalent Müller automaton  $M = (Q, 1, \{\{2\}, \{1, 2\}\}, T)$ .



Note that it is not enough to have  $\{\{2\}\}$  as the collection of final state sets.

Theorem 3. An  $\omega$ -language  $L$  over  $\Sigma$  is Müller recognisable if and only if  $L$  is a Boolean combination of  $\vec{L}_i$ 's, where each  $L_i \subseteq \Sigma^*$  and is regular.

**Proof:** Let  $L_i \subseteq \Sigma^*$  be regular. We know that the limit of  $L_i, \vec{L}_i$  is recognised by a deterministic Büchi automaton. We also know that any language accepted by a deterministic Büchi automaton is also accepted by a Müller automaton. So  $\vec{L}_i$  is recognised by a Müller automaton. Finally languages recognised by Müller automata are closed under Boolean operations.

In the other direction of the proof we start with a language  $L \subseteq \Sigma^\omega$  recognised by the Müller automaton  $M = (Q, s_0, F, T)$ . For every  $f \in F$ , we define the  $\omega$ -language  $L_f$  as follows:  $L_f = \{\alpha \in \Sigma^\omega : r \text{ is the run of } \alpha \text{ on } M \text{ and } \text{Inf}(r) = f\}$ . So  $L = L_\omega(M) = \bigcup_{f \in F} L_f$ .

For each  $q \in Q$ , we have the dfa  $M_q = (Q, s_0, \{q\}, T)$  over the alphabet  $\Sigma$ . Let  $L(M_q) = L_q \subseteq \Sigma^*$ . We know that  $\vec{L}_q = \{\alpha \in \Sigma^\omega : \text{there are infinitely many } n \in \mathbb{N} \text{ such that } \alpha_{1,n} \in L_q\}$ .

An  $\omega$ -string  $\alpha \in L_f$  if and only if  $\alpha \in \bigcap_{q \in f} \vec{L}_q$  and  $\alpha \notin \bigcup_{p \in Q \setminus f} \vec{L}_p$ . The second condition is equivalent to  $\bigcap_{p \in Q \setminus f} (\Sigma^\omega \setminus \vec{L}_p)$ . So we can write

$$L_f = \left( \bigcap_{q \in f} \vec{L}_q \right) \cap \left( \bigcap_{p \in Q \setminus f} (\Sigma^\omega \setminus \vec{L}_p) \right).$$

Finally  $L_\omega(M) = \bigcup_{f \in F} L_f$ . QED.

We have the following theorem shows all Müller recognisable  $\omega$ -languages are Büchi recognisable. The other direction of the proof is a harder work.

Theorem 4. Any  $\omega$ -language recognised by a Müller automaton is also recognised by a Büchi automaton ( $\omega$ -regular).

**Proof:** Let  $M = (Q, i, F, T)$  be a Müller automaton. It is clear that  $L_\omega(M) = \bigcup_{f \in F} L_\omega(M_f)$ , where  $M_f = (Q, i, \{f\}, T)$  i.e. the final state set of  $M_f$  has a single subset  $f$  of  $Q$ . If we can prove that  $L_\omega(M_f)$  is Büchi recognisable, we are done as Büchi recognisable languages are closed under union.

We start with the Müller automaton  $M_f = (Q, i, \{f\}, T)$  with the single set of final states  $f = \{q_0, \dots, q_k\} \subseteq Q$ . Restriction of the transition function  $T$  to  $f$  is  $T_f : f \times \Sigma \rightarrow f$ , where  $T_f(q_i, \sigma) = T(q_i, \sigma)$ . We define the following dfa's:  $N = (Q, i, \{q_0\}, T)$ ,  $N_j = (f, q_j, \{q_{j+1}\}, T_f)$  for all  $j$ ,  $0 \leq j < k$ , and  $N_k = (f, q_k, \{q_0\}, T_f)$ .

Let the regular languages of these dfa's be  $L = L(N)$  and  $L_j = L(N_j)$ ,  $j = 0, \dots, k$ . We claim that  $L_\omega(M_f) = L(L_0 \dots L_k)^\omega$ .

Let  $\alpha \in L(L_0 \dots L_k)^\omega$ . So  $\alpha = uv_1v_2\dots$ , where  $u \in L$  and  $v_i \in L_0 \dots L_k$ . The run of  $M_f$  on  $\alpha$  will be identical to the run of  $N$  followed by repeated run of  $N_0 \dots N_k$ . By design all and only states of  $f$  will appear infinitely many times in the run. So,  $\alpha \in L_\omega(M_f)$ .

If  $\alpha \in L_\omega(M_f)$ , then  $\text{Inf}(r) = f$ . If  $r$  is a run of  $M_f$  on  $\alpha$ , then after the initial segment (finite number) of states, there will be infinite repetition of all states of  $f$ . So there will be an infinite repetition of  $q_0, q_1, \dots, q_k, q_0, q_1, \dots, q_k, q_0, \dots$ . The initial segment  $u$  of  $\alpha$  is in  $L$  and  $v_i$  is the label of  $i^{\text{th}}$  segment,  $q_0 \xrightarrow{v_i} q_0$ . So  $\alpha \in L(L_0 \cdots L_k)^\omega$ .

QED.

## 1.4 Rabin Automaton

A sequential Rabin automaton<sup>1</sup> is a deterministic finite state machine. It distinguishes “good( $P$ )” and “bad( $N$ )” states in its successful runs. It is defined as follows.

**Definition 2:** A sequential Rabin automaton over  $\Sigma$  is a 4-tuple  $R = (Q, s_0, F, T)$ , where  $Q, s_0$  and  $T$  have their usual meanings in a dfa, but

$$F = \{(N_1, P_1), \dots, (N_k, P_k)\},$$

where  $N_i, P_i \subseteq Q$ ,  $i = 1, \dots, k$ . We may view  $N_i$ 's as “bad” and  $P_i$ 's as “good” states. Given an  $\omega$ -string  $\alpha$ , there is a unique run  $r$  of the machine  $R$  on  $\alpha$ . The run is successful if there is some  $i$ ,  $1 \leq i \leq k$  such that  $\text{Inf}(r) \cap N_i = \emptyset$  and  $\text{Inf}(r) \cap P_i \neq \emptyset$ .

Following two theorems show the equality of the class of  $\omega$ -languages recognisable by Müller automata and the class of  $\omega$ -languages recognisable by sequential Rabin automata.

**Theorem 5.** Let  $R = (Q, s_0, F, T)$  be a sequential Rabin automaton. There is a Müller automaton  $M$  such that  $L(R) = L(M)$ .

**Proof:** Let  $F = \{(N_1, P_1), \dots, (N_k, P_k)\}$ , where  $N_i, P_i \subseteq Q$ , for  $1 \leq i \leq k$ .

For every pair  $N_i$  and  $P_i$  we define two dfa's as follows:

$M_i = (Q, s_0, P_i, T)$  and  $\overline{M}_i = (Q, s_0, N_i, T)$ . Let  $L_i = L(M_i)$  and  $\overline{L}_i = L(\overline{M}_i)$ .

We claim that  $L_\omega(R) = \bigcup_{i=1}^k \left( \overrightarrow{L}_i \cap (\Sigma^\omega \setminus \overrightarrow{\overline{L}_i}) \right)$ .

Clearly the right-hand side language is recognised by a Müller automaton (characterisation theorem).

Let an  $\omega$ -word  $\alpha \in L_\omega(R)$ . So there is the successful run  $r$  of  $R$  on  $\alpha$ . By the Rabin acceptance condition there is some pair  $(N_i, P_i) \in F$  so that  $\text{Inf}(r) \cap P_i \neq \emptyset$  and  $\text{Inf}(r) \cap N_i = \emptyset$ . As the machine is deterministic the run  $r$  is unique, and there are infinitely many occurrences of at least one state of  $P_i$  in the run. As  $P_i$  is the set of final state of  $M_i$ , there are infinitely many prefixes of  $\alpha$  in  $L(M_i) = L_i$ , so  $\alpha \in \overrightarrow{L}_i$ .

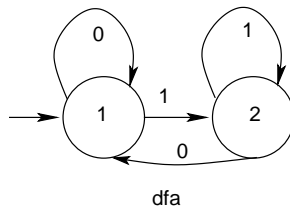
But  $\text{Inf}(r) \cap N_i = \emptyset$ , so  $\text{Inf}(r) \cap (Q \setminus N_i) \neq \emptyset$ . All states in the run that occur infinitely often are from non-final states of  $\overline{M}_i$ . So  $\alpha$  has infinitely many prefixes from the complement language  $\Sigma^* \setminus L(\overline{M}_i)$  and has only finite number of prefixes from  $L(\overline{M}_i)$ . It implies that  $\alpha$  is an element of  $\Sigma^\omega \setminus \overrightarrow{L(\overline{M}_i)}$ . So  $\alpha \in \overrightarrow{L}_i \cap (\Sigma^\omega \setminus \overrightarrow{\overline{L}_i})$ .

A similar proof can be given in the other direction.

QED.

Our next job is to prove that, given a Müller automaton  $M = (Q, s_0, F, T)$  there is a sequential Rabin automaton  $R$  so that  $L_\omega(M) = L_\omega(R)$ . Let  $F = \{f_1, \dots, f_k\}$ , where  $f_i \subseteq Q$ ,  $1 \leq i \leq k$ . The first temptation is to take  $R = (Q, s_0, F', T)$ , where  $F' = \{(Q \setminus f_1, f_1), \dots, (Q \setminus f_k, f_k)\}$  i.e.  $f_i$  are ‘good’ and  $Q \setminus f_i$  are ‘bad’ states. But the following example shows that it does not work, where  $L_\omega(M)$  is a proper subset of  $L_\omega(R)$  constructed in the above method.

**Example 5.** Let  $M = (\{1, 2\}, s_1, \{\{2\}, \{1, 2\}\}, T)$ , where  $T$  is shown in the diagram.



dfa

<sup>1</sup>There is a Rabin tree automaton which we shall not cover in this course.

Following our scheme the sequential Rabin automaton is

$$R = (\{1, 2\}, s_1, \{(\{1\}, \{2\}), (\emptyset, \{1, 2\})\}, T).$$

Due to the presence of the pair  $(\emptyset, \{1, 2\})$ ,  $\{0, 1\}^\omega = L_\omega(R)$ . Whereas  $L_\omega(M) = \{\alpha \in \{0, 1\}^\omega : \text{where } \alpha \text{ has either finite number of 0's and infinitely many 1's, or has infinitely many both 0's and 1's}\}$ .  $\omega$ -strings with finite number of 1's and infinite number of 0's do not have successful run on  $M$ .

The actual construction is slightly more involved. The equivalent sequential Rabin automaton  $R = (Q', s'_0, F', T')$  is defined as follows:

- $Q' = 2^{f_1} \times \dots \times 2^{f_k} \times Q$ ,
- $s'_0 = (\emptyset, \dots, \emptyset, s_0)$ ,
- If  $q = (A_1, \dots, A_k, p) \in Q'$  and  $\sigma \in \Sigma$ , then  $T'(q, \sigma) = q' = (A'_1, \dots, A'_k, p')$ , where
  - $T(p, \sigma) = p'$ ,
  - If  $A_i = f_i$ , then  $A'_i = \emptyset$ , else  $A'_i = A_i \cup (\{p'\} \cap f_i)$ , for all  $i = 1, \dots, k$ ,
- For all  $i = 1, \dots, k$ ,
  - $P_i = \{(A_1, \dots, A_k, p) : A_i = f_i\}$ , and
  - $N_i = \{(A_1, \dots, A_k, p) : p \notin f_i\}$ . If  $(A_1, \dots, A_k, p) \in N_i$ , then  $A_i \neq f_i$  as  $p \notin f_i$ .

The computation of  $M$  is simulated by the last component (state of  $Q$ ) of  $Q'$ . First  $k$  components of  $Q'$  maintains subsets of  $f_1, \dots, f_k$  for each visited state of the original machine. Once the subset is full (equal to  $f_i$ ) it is reset to an empty set.

Let an  $\omega$ -string  $\alpha \in L_\omega(M)$  (accepted by the Müller automaton), and  $r$  be the run of  $M$  on  $\alpha$ . So  $\text{Inf}(r) = f_i$  for some  $i = 1, \dots, k$ .

If  $r'$  is a run of  $R$  on  $\alpha$ , then all elements of  $f_i$  will occur infinitely often in the last component (state of  $M$ ) of  $q = (A_1, \dots, A_k, p) \in Q'$ . This implies that the condition  $A_i = f_i$  will be satisfied infinitely often (it will be emptied after it is full), i.e. elements of  $P_i$  will occur infinitely often.

On the other hand  $\text{Inf}(r') \cap N_i = \emptyset$ . Otherwise, if some state  $(A_1, \dots, A_k, p) \in N_i$  appears infinitely often in  $r'$ , then  $p \notin f_i$  will appear infinitely often in  $r$  of  $M$  - a contradiction.

In the other direction of the proof we start with an unsuccessful run  $r$  of  $M$  on some  $\omega$ -word  $\beta$ . So  $\text{Inf}(r) \neq f_i$  for any  $i = 1, \dots, k$ . Let the corresponding run of  $R$  on  $\beta$  be  $r'$ . No state in  $P_i$  for any  $i = 1, \dots, k$  can occur infinitely often as none of  $A_i$ 's will be full infinitely often.

## 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 Automata, Semigroups, Logic and Games*, by Dominique Perrin and Jean-Éric Pin, Elsevier, Academic Press, 2004, ISBN 0-12-532111-2.