Contents

Chapter 11

Automata-theoretic Liveness Verification

In this chapter, we extend the automata-theoretic approach studied in Chapter 6 for safety requirements to liveness requirements. In the automata-theoretic liveness verification, a fair module is viewed as a generator of an ω-language, namely, the set of its fair traces, the requirement is specified by an ω-automaton that accepts only the desired ω-traces, and the verification problem corresponds to inclusion between these two ω-languages.

11.1 ω-Automata

A fair structure $\mathcal{K}$ consists of an observation structure K and a fairness assumption F. Each fair structure defines the ω-language $\mathcal{L}_{\mathcal{K}}$ over the set of its observations consisting of the set of ω-traces corresponding to initialized F-fair ω-trajectories. Fair structures can be used to specify requirements also. In their role as a specification language, fairness constraints are usually specified using regions rather than actions. In this role, fairness constraints should be viewed as *accepting conditions* that classify ω-trajectories into accepting and rejecting rather than assumptions about fair resolution of choice. We will concentrate on two types of accepting conditions: Büchi acceptance and Streett acceptance.

Büchi automata

Finite structures with a single weak-fairness constraint specified by a region are called Büchi automata.

> BÜCHI AUTOMATON
>
> A *Büchi automaton* $\mathcal{M}$ consists of (1) a finite observation structure K and (2) [the *repeating region*] a region σ^A of K. An initialized ω-trajectory $\underline{s}$ of K is *accepted* by the Büchi automaton $\mathcal{M}$ if $s_i \in \sigma^A$ for infinitely many positions $i \geq 0$. The *ω-language* $\mathcal{L}_{\mathcal{M}}$ of the Büchi automaton $\mathcal{M}$ is the set of traces corresponding to initialized accepted trajectories of $\mathcal{M}$. The Büchi automaton (K, σ^A) is *deterministic* if the observation structure K is deterministic.

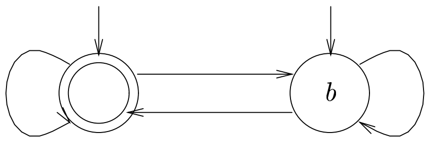
Note that syntactically a Büchi automaton is identical to an ordinary automaton. In an ordinary automaton, a (finite) trajectory is accepted if it terminates in an accepting state; in a Büchi automaton, an ω-trajectory is accepted if its visits a repeating state infinitely often.

Example 11.1 [Büchi languages] Let $A = \{a, b\}$. The Büchi automaton $\mathcal{M}_1$ of Figure 11.1 accepts the response language $(b^*a)^\omega$ consisting of ω-words that contain infinitely many a symbols. The Büchi automaton $\mathcal{M}_2$ of Figure 11.1 accepts the persistence language A^*a^ω consisting of ω-words with a suffix containing only a symbols. Note that the automaton $\mathcal{M}_2$ is nondeterministic (it guesses the beginning of the suffix containing only a symbols).

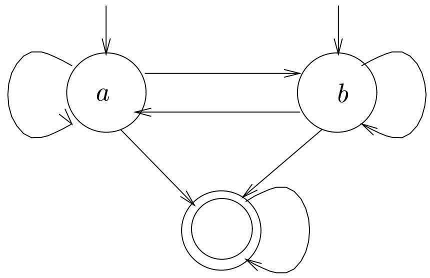
Let $A = \{a, b, c\}$. The nondeterministic Büchi automaton $\mathcal{M}_3$ of Figure 11.1 accepts the reactivity language consisting of ω-words that contain either only finitely many a symbols or infinitely many b symbols. Note that "finitely many a or infinitely many b" is equivalent to "infinitely many b or eventually always c." ■

Remark 11.1 [Multi-Büchi automaton] A *multi-Büchi automaton* $\mathcal{M}$ consists of (1) a finite observation structure K, and (2) a finite set F of *repeating regions* of K. An initialized ω-trajectory $\underline{s}$ of K is accepted by the multi-Büchi automaton $\mathcal{M}$ if for every repeating region $\sigma \in F$, $s_i \in \sigma$ for infinitely many positions $i \geq 0$. Thus, a multi-Büchi automaton is a weak-fair structure all of whose weak-fairness constraints are specified by regions. ■

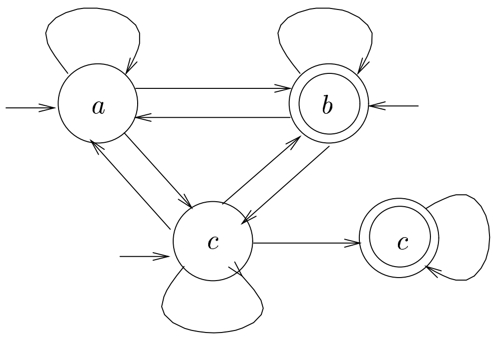
Exercise 11.1 {} [CoBüchi automata] A *CoBüchi automaton* $\mathcal{M}$ consists of (1) a finite observation structure K and (2) [the *stable region*] a region σ^A of K. An initialized ω-trajectory $\underline{s}$ of K is accepted by the CoBüchi automaton $\mathcal{M}$ if it has a suffix all of whose states are in the stable region: there exists $i \geq 0$ such that $s_j \in \sigma^A$ for all $j \geq i$. Note that syntactically a CoBüchi automaton is



Automaton $\mathcal{M}_1$



Automaton $\mathcal{M}_2$



Automaton $\mathcal{M}_3$

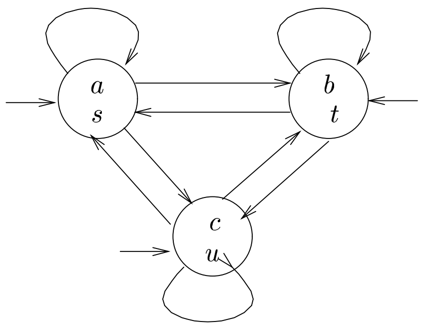
Figure 11.1: Sample Büchi automata

like a Büchi automaton or an ordinary automaton. Semantically, the CoBüchi automaton (K, σ^A) is like the fair structure $(K, \{(\Sigma \backslash \sigma^A, \emptyset)\})$.

Consider the alphabet $A = \{a, b\}$. (1) Find a CoBüchi automaton whose ω-language is the persistence language $A^* a^\omega$ consisting ω-words with a suffix containing only a symbols. (2) Consider the response language $(b^* a)^\omega$ consisting of ω-words with infinitely many a symbols. Can you draw a CoBüchi automaton that accepts this langauge? ■

Streett Automata

Finite structures with a fairness constraints specified by regions are called Streett automata.



Streett Automaton $\mathcal{M}$

Fairness constraint $=(s,t)$

Figure 11.2: Sample Streett automaton

STREETT AUTOMATON

A *Streett automaton* $\mathcal{M}$ consists of (1) a finite observation structure K and (2) [the *Streett constraints*] a finite set F of pairs of regions. An initialized ω-trajectory $\underline{s}$ of K is *accepted* by the Streett automaton $\mathcal{M}$ if for every Streett constraint $(\sigma, \tau) \in F$, if $\underline{s}$ is σ-fair then $\underline{s}$ is τ-fair. The *ω-language* $\mathcal{L}_{\mathcal{M}}$ of the Streett automaton $\mathcal{M}$ is the set of ω-traces corresponding to initialized accepted ω-trajectories of $\mathcal{M}$.

Remark 11.2 [Büchi as a special case of Streett] A Büchi automaton (K, σ^A) can be viewed as the Streett automaton $(K, \{(\Sigma, \sigma^A)\})$ with a single Streett constraint. ■

Example 11.2 [Streett language] Let $A = \{a, b, c\}$. The nondeterministic Büchi automaton $\mathcal{M}_3$ of Figure 11.1 accepts the reactivity language consisting of ω-words that contain either only finitely many a symbols or infinitely many b symbols. The same ω-language is accepted by the deterministic Streett automaton $\mathcal{M}$ of Figure 11.2. There is a single Streett constraint $(\{s\}, \{t\})$. ■

Exercise 11.2 {} [Rabin automata] A *Rabin automaton* $\mathcal{M}$ is syntactically identical to a Streett automaton, and consists of (1) a finite observation structure K and (2) [the *Rabin constraints*] a finite set F of pairs of regions. An initialized ω-trajectory $\underline{s}$ of K is *accepted* by the Rabin automaton $\mathcal{M}$ if there exists a Rabin constraint $(\sigma, \tau) \in F$ such that $\underline{s}$ is σ-fair and but not τ-fair. Thus, semantically a Rabin automaton is the dual of the Streett automaton: Streett acceptance has the form

$$\bigwedge(\sigma,\tau) \in F.\ [\neg(\sigma\text{-fair}) \vee \tau\text{-fair}\ ],$$

while Rabin acceptance requires

$$\bigvee(\sigma,\tau) \in F.\ [\sigma\text{-fair} \wedge \neg(\tau\text{-fair})\ ].$$

Let $A = \{a, b, c\}$. Find a deterministic Rabin automaton $\mathcal{M}$ accepting the reactivity language consisting of ω-words that contain either only finitely many a symbols or infinitely many b symbols. ■

Exercise 11.3 {} [Muller automata] A *Muller automaton* $\mathcal{M}$ is syntactically like a multi-Büchi automaton, and consists of (1) a finite observation structure K and (2) [the *Muller acceptance*] a finite set F of regions of K. An initialized ω-trajectory $\underline{s}$ of K is accepted by the Muller automaton $\mathcal{M}$ if the set $\{s \in \Sigma \mid s_i = s \text{ for infinitely many } i \geq 0\}$ of states repeating infinitely often along $\underline{s}$ is in F. Show that Streett automata as well as Rabin automata are special cases of Muller automata. ■

The ω-language-inclusion problem

The ω-language-inclusion problem asks whether every ω-trace accepted by one ω-automaton is also accepted by another ω-automaton.

> THE ω-LANGUAGE-INCLUSION PROBLEM
> An instance $(\mathcal{M}_1, \mathcal{M}_2)$ of the *ω-language-inclusion problem* consists of two ω-automata $\mathcal{M}_1$ and $\mathcal{M}_2$ over the same observation alphabet A. The answer to the ω-language-inclusion problem $(\mathcal{M}_1, \mathcal{M}_2)$ is YES if $\mathcal{L}_{\mathcal{M}_1} \subseteq \mathcal{L}_{\mathcal{M}_2}$, and otherwise NO.

Note that in an instance $(\mathcal{M}_1, \mathcal{M}_2)$ of the ω-language-inclusion problem, each of the ω-automata $\mathcal{M}_1$ and $\mathcal{M}_2$ may be either a fair structure, or a Büchi automaton, or a Streett automaton.

Automata as specifications

ω-automata can be used for specifying requirements of fair modules. As in case of the logic SAL, the observations of the requirements automaton are boolean expressions over the observable variables of modules. We define the fair state logic LAL whose formulas are Büchi and Streett automata.

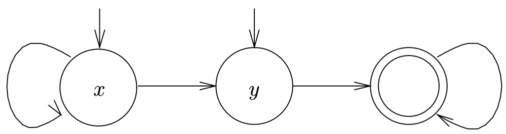


Figure 11.3: The LAL formula $\mathcal{M}_{\forall\mathcal{U}}$

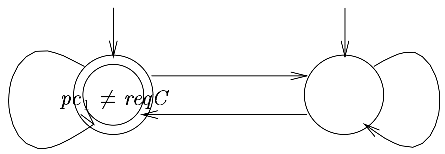


Figure 11.4: Starvation freedom in live automaton logic

LIVE AUTOMATON LOGIC

A formula of the fair state logic *live automaton logic* (LAL) is a Büchi or a Streett automaton $\mathcal{M}$ whose observations are boolean expressions.

Given a formula $\mathcal{M}$ of LAL, a fair structure $\mathcal{K}$ is a $\mathcal{M}$-structure if each observation of $\mathcal{K}$ is a valuation for a superset of the variables appearing in the observations of $\mathcal{M}$.

The satisfaction relation for LAL is defined by:

$s \models_{\mathcal{K}} \mathcal{M}$ iff for every source-s fair ω-trajectory $\underline{s}$ of $\mathcal{K}$ there is an accepting ω-trace $\underline{a} \in \mathcal{L}_{\mathcal{M}}$ such that for all $i \geq 0$, $s_i \models a_i$.

In other words, a state s of $\mathcal{K}$ satisfies the requirement specified by the ω-automaton $\mathcal{M}$ if for every source-s fair ω-trace $\underline{a}$ of $\mathcal{K}$, we can find an initialized accepting ω-trace $\underline{b}$ of M such that every observation in $\underline{a}$ is consistent with the corresponding expression in $\underline{b}$.

Example 11.3 [LAL] The LAL formula $\mathcal{M}_{\forall\mathcal{U}}$ shown in Figure 11.3 asserts that, given a state s, every source-s fair ω-trajectory contains a state satisfying y which is preceded only by states satisfying x. The formula $\mathcal{M}_{\forall\mathcal{U}}$ can be interpreted at states of a fair structure whose observations assign values to x and y. It follows that the LAL formula $\mathcal{M}_{\forall\mathcal{U}}$ is equivalent to the CTL formula $x\forall\mathcal{U}y$. Contrast the specification $\mathcal{M}_{\forall\mathcal{U}}$ with the SAL specification $M_{\mathcal{W}}$ corresponding to the STL formula $x\forall\mathcal{W}y$ (see Example 9.2). ■

Example 11.4 [Starvation freedom in live automaton logic] Recall the starvation freedom requirement for mutual exclusion protocols. The requirement that $pc_1 \neq \mathit{reqC}$ be a recurrent is expressed in LAL by the Büchi automaton of Figure 11.4. ■

LAL model checking

The model-checking problem for LAL can be reduced to the ω-language-inclusion problem. As in case of SAL, we expand each ω-automaton $\mathcal{M}$ of LAL to a larger automaton $E\mathcal{M}$ whose observations are valuations to the variables appearing in the observations of $\mathcal{M}$. Recall the definition of the expansion operator E from Chapter 9. To obtain expansion of an ω-automaton, we apply the expansion operation to the underlying observation structure, and modify the accepting condition appropriately.

EXPANSION OF AN LAL AUTOMATON

For a Büchi automaton $\mathcal{M} = (K, \sigma^A)$ given as a LAL formula, the *expansion* $E\mathcal{M}$ is another Büchi automaton: (1) the observation structure of $E\mathcal{M}$ is EK, and (2) the repeating region of $E\mathcal{M}$ is $\sigma^A \Uparrow$.

For a Streett automaton $\mathcal{M} = (K, F)$ given as a LAL formula, the *expansion* $E\mathcal{M}$ is another Streett automaton: (1) the observation structure of $E\mathcal{M}$ is EK, and (2) for every $(\sigma, \tau) \in F$, the automaton $E\mathcal{M}$ has the Streett constraint $(\sigma \Uparrow, \tau \Uparrow)$.

Exercise 11.4 {} [LAL expansion] Draw the expanded Büchi automaton corresponding to the LAL specification $\mathcal{M}_{\forall\mathcal{U}}$ of Figure 11.3. ■

It follows that checking whether a fair structure satisfies an LAL automaton $\mathcal{M}$ is equivalent to checking whether $\mathcal{K}$ satsifies the expanded ω-automaton $E\mathcal{M}$, which in turn corresponds to checking whether the fair language of $\mathcal{K}$ is contained in the fair language of $E\mathcal{M}$.

Proposition 11.1 [LAL model checking] *The* LAL *model-checking problem* $(\mathcal{K}, \mathcal{M})$ *and the ω-language-inclusion problem* $(\mathcal{K}, E\mathcal{M})$ *have the same answer.*

11.2 Operations on ω-automata

To solve the ω-language inclusion problem $(\mathcal{M}_1, \mathcal{M}_2)$, we first obtain an ω-automaton that accepts the complement of the ω-language accepted by $\mathcal{M}_2$, then construct its product with $\mathcal{M}_1$, and solve the fair emptiness problem on the resulting ω-automaton.

11.2.1 Product

Given two ω-automata $\mathcal{M}_1$ and $\mathcal{M}_2$, we wish to define another ω-automaton that accepts the intersection of the ω-languages of $\mathcal{M}_1$ and $\mathcal{M}_2$. For this purpose, we resort to the product construction described in Section 9.3.2 over observation structures. Consider two observation structures K_1 and K_2, and let $K_1 \times K_2$ be their product. Let $\underline{s}$ be an ω-trajectory of the product. Then, by the definition of the product, there exists an ω-trajectory $\underline{t}$ of K_1 and an ω-trajectory $\underline{u}$ of K_2 such that $\underline{s} = (t_0, u_0)(t_1, u_1)\cdots$. Observe that, for a region σ of K_1, the ω-trajectory $\underline{t}$ of K_1 is σ-fair iff the ω-trajectory $\underline{s}$ of the product is $(\sigma \Uparrow)$-fair. Similarly, for a region σ of K_2, the ω-trajectory $\underline{u}$ of K_2 is σ-fair iff the ω-trajectory $\underline{s}$ of the product is $(\sigma \Uparrow)$-fair. In other words, fairness with respect to a region σ in a component translates to fairness with respect to the lifted region $\sigma \Uparrow$ in the product. Similarly, fairness with respect to an action α in a component translates to fairness with respect to the lifted action $\alpha \Uparrow$ in the product. This leads to a natural definition of product for ω-automata.

PRODUCT OF ω-AUTOMATA

Let $\mathcal{M}_1 = (K_1, F_1)$ and $\mathcal{M}_2 = (K_2, F_2)$ be two Streett automata. The product $\mathcal{M}_1 \times \mathcal{M}_2$ is the Streett automaton $(K_1 \times K_2, \{(\sigma \Uparrow, \tau \Uparrow) \mid (\sigma, \tau) \in F_1 \cup F_2\})$.

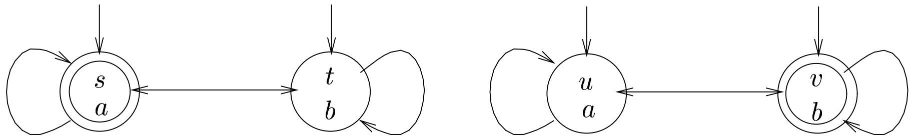
Let $\mathcal{K}_1 = (K_1, F_1)$ and $\mathcal{K}_2 = (K_2, F_2)$ be two fair structures. The product $\mathcal{K}_1 \times \mathcal{K}_2$ is the fair structure $(K_1 \times K_2, \{(\alpha \Uparrow, \beta \Uparrow) \mid (\alpha, \beta) \in F_1 \cup F_2\})$.

Proposition 11.2 [Product of ω-automata] *If $\mathcal{M}_1$ and $\mathcal{M}_2$ are two Streett automata, then $\mathcal{L}_{\mathcal{M}_1 \times \mathcal{M}_2} = \mathcal{L}_{\mathcal{M}_1} \cap \mathcal{L}_{\mathcal{M}_2}$. If $\mathcal{K}_1$ and $\mathcal{K}_2$ are two fair structures, then $\mathcal{L}_{\mathcal{K}_1 \times \mathcal{K}_2} = \mathcal{L}_{\mathcal{K}_1} \cap L_{\mathcal{K}_2}$.*

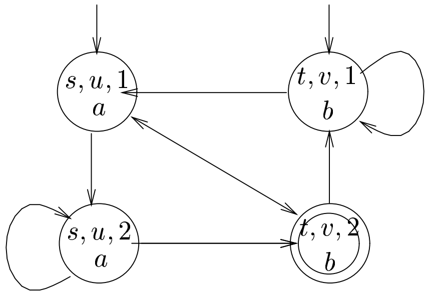
Remark 11.3 [Cost of product] Let $\mathcal{M}_1$ be a Streett automaton with n_1 states, m_1 transitions, and ℓ_1 Streett constraints. Let $\mathcal{M}_2$ be a Streett automaton with n_2 states, m_2 transitions, and ℓ_2 Streett constraints. Then, the product $\mathcal{M}_1 \times \mathcal{M}_2$ has at most $n_1 \cdot n_2$ states, at most $m_1 \cdot m_2$ transitions, and $\ell_1 + \ell_2$ Streett constraints. ■

Product of Büchi automata

The product of two Büchi automata (K_1, σ_1^A) and (K_2, σ_2^A) can be defined to be the multi-Büchi automaton $(K_1 \times K_2, \{\sigma_1^A \Uparrow, \sigma_2^A \Uparrow\})$. However, by introducing a counter, as described in Section 12.3.2, we can define product of Büchi automata to be a Büchi automaton. The states of the product are, then, of the form (s, t, i), where s is a state of K_1, t is a state of K_2, and i is a counter that can be either 1 or 2. The counter is updated from 1 to 2 when an accepting state of K_1 is visited, and from 2 to 1 when an accepting state of K_2 is visited. The



Automaton $\mathcal{M}_2$



Product Automaton $\mathcal{M}_1 \times \mathcal{M}_2$

Figure 11.5: Product construction for Büchi automata

accepting condition of the product requires infinitely many updates from 2 to 1.

> PRODUCT OF BÜCHI AUTOMATA
>
> Let $\mathcal{M}_1 = (\Sigma_1, \sigma_1^I, \rightarrow_1, A, \langle\!\langle\cdot\rangle\!\rangle_1, \sigma_1^A)$ and $\mathcal{M}_2 = (\Sigma_2, \sigma_2^I, \rightarrow_2, A, \langle\!\langle\cdot\rangle\!\rangle_2, \sigma_2^A)$ be two Büchi automata. The *product* $\mathcal{M}_1 \times \mathcal{M}_2$ is the Büchi automaton $(\Sigma, \sigma^I, \rightarrow, A, \langle\!\langle\cdot\rangle\!\rangle, \sigma^A)$:
>
> - $\Sigma = \{(s_1, s_2, i) \mid s_1 \in \Sigma_1,\ s_2 \in \Sigma_2,\ \langle\!\langle s_1\rangle\!\rangle_1 = \langle\!\langle s_2\rangle\!\rangle_2,\ \text{and } i \in \{1, 2\}\}$;
> - $(s_1, s_2, i) \in \sigma^I$ iff $s_1 \in \sigma_1^I$, $s_2 \in \sigma_2^I$, and $i = 1$;
> - $(s_1, s_2, i) \rightarrow (t_1, t_2, j)$ iff $s_1 \rightarrow_1 t_1$, $s_2 \rightarrow_2 t_2$, if $i = 1$ then if $s_1 \in \sigma_1^A$ then $j = 2$ else $j = 1$, and if $i = 2$ then if $s_2 \in \sigma_2^A$ then $j = 1$ else $j = 2$;
> - $\langle\!\langle (s_1, s_2, i)\rangle\!\rangle = \langle\!\langle s_1\rangle\!\rangle_1 = \langle\!\langle s_2\rangle\!\rangle_2$;
> - $(s_1, s_2, i) \in \sigma^A$ if $i = 2$ and $s_2 \in \sigma_2^A$.

Proposition 11.3 [Product of Büchi automata] *If $\mathcal{M}_1$ and $\mathcal{M}_2$ are two Büchi automata, then $\mathcal{L}_{\mathcal{M}_1 \times \mathcal{M}_2} = \mathcal{L}_{\mathcal{M}_1} \cap \mathcal{L}_{\mathcal{M}_2}$.*

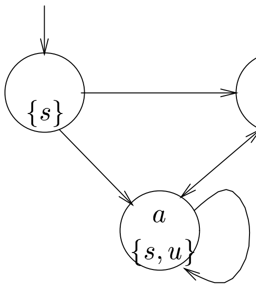


Figure 11.6: Subset construction does not work for Büchi acceptance

Example 11.5 [Product of Büchi automata] Consider the two Büchi automata $\mathcal{M}_1$ and $\mathcal{M}_2$ of Figure 11.5. The automaton $\mathcal{M}_1$ accepts all ω-words that contain infinitely many a symbols, while $\mathcal{M}_2$ accepts all ω-words that contain infinitely many b symbols. The result of applying the product construction contains 4 states, of which the only accepting state is $(t, v, 2)$. Verify that the product accepts precise those ω-words that contain infinitely many a symbols as well as infinitely many b symbols. ■

Remark 11.4 [Product of deterministic Büchi automata] If $\mathcal{M}_1$ and $\mathcal{M}_2$ are deterministic Büchi automata, then so is their product $\mathcal{M}_1 \times \mathcal{M}_2$. Thus, the class of ω-languages definable by deterministic Büchi automata is closed under intersection. ■

Exercise 11.5 $\{\}$ [Product of CoBüchi automata] Given two CoBüchi automata $\mathcal{M}_1$ and $\mathcal{M}_2$, define a CoBüchi automaton $\mathcal{M}_1 \times \mathcal{M}_2$ that accepts the intersection of the ω-languages of $\mathcal{M}_1$ and $\mathcal{M}_2$. ■

11.2.2 Complementation

We turn our attention to the problem of complementing a Büchi automaton. Recall that for an ordinary automaton, its complement is constructed by first determinizing the observation structure using the subset construction, followed by completion by adding dummy states, followed by inversion of the accepting condition. Given a Büchi automaton $\mathcal{M} = (K, \sigma^A)$, can we add accepting conditions to the determinized structure ΔK without changing the ω-language accepted by $\mathcal{M}$? The obstacle in such an approach is illustrated by the following example.

Example 11.6 [Subset construction and Büchi automata] Recall the Büchi automaton $\mathcal{M}_2$ from Figure 11.1 that accepts the persistence language $A^* a^\omega$. The determinized structure obtained by subset construction is shown in Figure 11.6. Declaring the state corresponding to the subset $\{s, u\}$ to be repeating does not preserve the ω-language of $\mathcal{M}_2$. ■

The problem with the subset construction is that states of the determinized structure may contain both repeating and nonrepeating states. Complementing a nondeterministic Büchi automaton turns out to be a nontrivial problem. Consequently, existing model checkers do not support nondeterministic ω-automata as specifications. However, understanding the complementation procedure provides insights into the structure of ω-automata.

We begin by some preliminary definitions. Let A be a finite alphabet. An equivalence relation $\sim \subseteq A^* \times A^*$ over words over A is said to be a *congruence* (with respect to concatenation) if for all words $\overline{a}$, $\overline{b}$, and $\overline{c}$, if $\overline{a} \sim \overline{b}$ then $\overline{a}\cdot\overline{c} \sim \overline{b}\cdot\overline{c}$ and $\overline{c}\cdot\overline{a} \sim \overline{c}\cdot\overline{b}$. By a *finite* equivalence relation, we mean an equivalence relation with finitely many equivalence classes.

Let $\mathcal{M} = (\Sigma, \sigma^I, \rightarrow, A, \langle\!\langle\cdot\rangle\!\rangle, \sigma^A)$ be a Büchi automaton. We are going to establish that both $\mathcal{L}_\mathcal{M}$ and and its complement can be expressed as finite unions of ω-languages of the form $L_1 \cdot L_2^\omega$, where L_1 and L_2 are blocks of a certain finite congruence on A^*.

> For two state s and t of $\mathcal{M}$, and a word $\overline{a}_{0\ldots m}$ over A, define $s\,\alpha^T\overline{a}\,t$ if there is a trajectory $\overline{s}_{0\ldots m}$ of $\mathcal{M}$ such that $s_0 = s$, $s_m = t$, and $\langle\!\langle s_i\rangle\!\rangle = a_i$ for all $0 \leq i \leq m$.

That is, $s\,\alpha^T\overline{a}\,t$ means that the trace $\overline{a}$ can lead the automaton from the initial state s to the final state t.

> For two state s and t of $\mathcal{M}$, and a word $\overline{a}_{0\ldots m}$ over A, define $s\,\alpha^T\overline{a}'\,t$ if there is a trajectory $\overline{s}_{0\ldots m}$ of $\mathcal{M}$ such that $s_0 = s$, $s_m = t$, $\langle\!\langle s_i\rangle\!\rangle = a_i$ for all $0 \leq i \leq m$, and $s_j \in \sigma^A$ for some $0 \leq j \leq m$.

That is, $s\,\alpha^T\overline{a}'\,t$ means that the trace $\overline{a}$ can lead the automaton from the initial state s to the final state t via a trajectory that visits some repeating state. Now we are ready to define the desired equivalence relation on A^* induced by $\mathcal{M}$:

> For two words $\overline{a}$ and $\overline{b}$ over A, $\overline{a} \sim_\mathcal{M} \overline{b}$ iff for all states s and t of $\mathcal{M}$, (1) $s\,\alpha^T\overline{a}\,t$ iff $s\,\alpha^T\overline{b}\,t$, and (2) $s\,\alpha^T\overline{a}'\,t$ iff $s\,\alpha^T\overline{b}'\,t$.

First, we note that the equivalence $\sim_\mathcal{M}$ is a finite congruence:

Lemma 11.1 [Congruence] *The equivalence relation $\sim_\mathcal{M}$ over A^* is a congruence with respect to concatenation.*

Proof. Left as an exercise. ■

Lemma 11.2 [Finiteness] *The equivalence relation $\sim_{\mathcal{M}}$ over A^* is finite, and has at most $\mathbf{2}^{2n^2}$ classes if $\mathcal{M}$ has n states.*

Proof. For every two states s and t of $\mathcal{M}$, let $L_{s,t}$ be the language containing words $\overline{a}$ such that $s\,\alpha^T\overline{a}\,t$, and let $L'_{s,t}$ be the language containing words $\overline{a}$ such that $s\,\alpha^T\overline{a}'\,t$. Let Π be the set of these $2n^2$ languages. Now an equivalence class of $\sim_{\mathcal{M}}$ corresponds to a subset of Π: given a subset $\Pi' \subseteq \Pi$, the intersection

$$[\bigcap L \in \Pi'.\, L] \;\cap\; [\bigcap L \notin \Pi'.\, A^*\backslash L]$$

defines an equivalence class of $\sim_{\mathcal{M}}$. It follows that the number of subsets of Π is an upper bound for the number of equivalence classes of $\sim_{\mathcal{M}}$. ■

The next lemma asserts a saturation property of the $\sim_{\mathcal{M}}$-equivalence classes with respect to the ω-language accepted by $\mathcal{M}$:

Lemma 11.3 [Saturation] *Let L_1 and L_2 be two equivalence classes of the congruence $\sim_{\mathcal{M}}$. Then, if $L_1 \cdot L_2^\omega \cap \mathcal{L}_{\mathcal{M}}$ is nonempty then $L_1 \cdot L_2^\omega \subseteq \mathcal{L}_{\mathcal{M}}$.*

Proof. Let L_1 and L_2 be two equivalence classes of $\sim_{\mathcal{M}}$. Suppose $L_1 \cdot L_2^\omega \cap \mathcal{L}_{\mathcal{M}}$ is nonempty, and contains the ω-word $\underline{a}$. Since $\underline{a} \in L_1 \cdot L_2^\omega$, it is of the form $\overline{b}_0 \cdot \overline{b}_1 \cdot \overline{b}_2 \cdots$, where the word $\overline{b}_0$ is in L_1 and for $i \geq 1$, the word $\overline{b}_i$ is in L_2. Since $\underline{a}$ is accepted by $\mathcal{M}$, there exists an initialized accepting ω-trajectory corresponding to $\underline{a}$. Thus, there exist states $s_0, s_1, \ldots$ such that

$$s_0\ \alpha^T\overline{b}_0\ s_1\ \alpha^T\overline{b}_1\ s_1\ \alpha^T\overline{b}_2\ \cdots$$

Furthermore, for infinitely many indices i, $s_i\ \alpha^T\overline{b}_i{}'\ s_{i+1}$.

Now consider another word $\underline{c} \in L_1 \cdot L_2^\omega$. We need to establish that $\mathcal{M}$ accepts $\underline{c}$ also. The ω-word $\underline{c}$ is of the form $\overline{d}_0 \cdot \overline{d}_1 \cdot \overline{d}_2 \cdots$ such that the word $\overline{d}_0$ is in L_1 and for $i \geq 1$, the word $\overline{d}_i$ is in L_2. Since L_1 and L_2 are equivalence classes of $\sim_{\mathcal{M}}$, $\overline{b}_i \sim_{\mathcal{M}} \overline{d}_i$ for all $i \geq 0$. It follows that

$$s_0\ \alpha^T\overline{d}_0\ s_1\ \alpha^T\overline{d}_1\ s_1\ \alpha^T\overline{d}_2\ \cdots$$

and for infinitely many indices i, $s_i\ \alpha^T\overline{d}_i{}'\ s_{i+1}$. We conclude that there is an initialized accepting trajectory corresponding to the ω-word $\underline{c}$. ■

The next lemma asserts that ω-languages of the form $L_1 \cdot L_2^\omega$ cover the set of all ω-words, provided L_1, L_2 range over equivalence classes of a finite congruence.

Lemma 11.4 [Coverage] *Let $\sim$ be a finite congruence over A^*, and let $\underline{a}$ be an ω-word over A. Then, there exist equivalence classes L_1 and L_2 of $\sim$ such that $\underline{a} \in L_1 \cdot L_2^\omega$.*

Proof. Let $\sim$ be a finite congruence relation over A^*, and let $\underline{a}$ be an infinite word over A. We say that two indices i and j merge at an index $k > i, j$ if $\overline{a}_{i \ldots k} \sim \overline{a}_{j \ldots k}$. For two indices i and j, define $i \cong j$ if they merge at some index. Verify that $\cong$ is an equivalence relation over the set of nonnegative integers. Furthermore, given a finite subset D of nonnegative integers such that D has more elements than the number of equivalence classes of $\sim$, if we choose k such that $k > i$ for all $i \in D$, then the set $\{\overline{a}_{i \ldots k} \mid i \in D\}$ must contain two $\sim$-equivalent words. It follows that the equivalence relation $\cong$ itself is finite (the number of equivalence classes of $\cong$ is bounded by the number of equivalence classes of $\sim$).

Finiteness of $\cong$ implies that there exists an infinite sequence $i_0 < i_1 < i_2 < \cdots$ of indices that are $\cong$-equivalent to each other. Note that for every $j \geq 1$, all the indices $i_0, i_1, \ldots i_j$ merge at some $k > i_j$. Without loss of generality, we may assume that for every $j \geq 1$, all the indices $i_0, i_1, \ldots i_j$ merge at i_{j+1} (this is because we can delete indices from the original sequence, and if $i_0, i_1, \ldots i_j$ merge at k then they merge at every $k' > k$ as $\sim$ is a congruence). It follows that there is an infinite sequence $i_0, i_1, i_2, \ldots$ of indices such that

1. all the words in $\{\overline{a}_{i_0 \ldots i_j} \mid j \geq 1\}$ belong to the same equivalence class of $\sim$, let this class be L_2,
2. for all $j < j' < k$, the indices i_j and $i_{j'}$ merge at i_k.

From (1), $\overline{a}_{i_0 \ldots i_1}$ is in L_2. For all $j \geq 1$, $\overline{a}_{i_0 \ldots i_{j+1}}$ is in L_2 by (1), and $\overline{a}_{i_0 \ldots i_{j+1}}$ is $\sim$-equivalent to $\overline{a}_{i_j \ldots i_{j+1}}$ by (2). It follows that for all $j \geq 0$, $\overline{a}_{i_j \ldots i_{j+1}}$ is in L_2.

It follows that the suffix $\overline{a}_{i_0 \ldots}$ is in L_2^ω. This completes the proof. ■

Since $\sim_{\mathcal{M}}$ is a finite congruence, it follows that the set

$$\{L_1 \cdot L_2^\omega \mid L_1, L_2 \text{ are equivalence classes of } \sim_{\mathcal{M}}\}$$

covers A^ω, and then, by the saturation property, the ω-language accepted by $\mathcal{M}$ corresponds to a subset of this set, and the complement defines the complementary language.

Corollary 11.1 [Structure of Büchi language] *The ω-language $\mathcal{L}_{\mathcal{M}}$ accepted by the Büchi automaton $\mathcal{M}$ equals*

$$\bigcup \{L_1 \cdot L_2^\omega \mid \ L_1, L_2 \text{ are equivalence classes of } \sim_{\mathcal{M}} \text{ and } L_1 \cdot L_2^\omega \cap \mathcal{L}_{\mathcal{M}} \neq \emptyset\},$$

and the complementary ω-language $A^\omega \backslash \mathcal{L}_{\mathcal{M}}$ equals

$$\bigcup \{L_1 \cdot L_2^\omega \mid \ L_1, L_2 \text{ are equivalence classes of } \sim_{\mathcal{M}} \text{ and } L_1 \cdot L_2^\omega \cap \mathcal{L}_{\mathcal{M}} = \emptyset\}.$$

Proof. Follows from Lemmas 11.1, 11.2, 11.3, and 11.4. ■

The next proposition asserts that if L_1 and L_2 are two regular languages then the ω-language $L_1 \cdot L_2^\omega$ is accepted by a Büchi automaton.

Proposition 11.4 [Regular concatenation] *If L_1 and L_2 are two regular languages over A then the ω-language $L_1 \cdot L_2^\omega$ is accepted by some Büchi automaton.*

Proof. Let L_1 be a regular language accepted by the automaton $M_1 = (\Sigma_1, \sigma_1^I, \rightarrow_1, A, \langle\!\langle\cdot\rangle\!\rangle_1, \sigma_1^A)$, and let L_2 be a regular language accepted by the automaton $M_2 = (\Sigma_2, \sigma_2^I, \rightarrow_2, A, \langle\!\langle\cdot\rangle\!\rangle_2, \sigma_2^A)$. The ω-automaton accepting $L_1 \cdot L_2^\omega$ is obtained by taking disjoint union of the two automata M_1 and M_2, and adding transitions from accepting states of M_1 to the initial states of M_2, and from accepting states of M_2 to the initial states of M_2. Specifically, define the Büchi automaton $\mathcal{M}$ over the alphabet $\mathcal{M}$: (1) the state-space of $\mathcal{M}$ is $\Sigma_1 \cup \Sigma_2$ (assuming Σ_1 and Σ_2 are disjoint sets), (2) the initial region of $\mathcal{M}$ is σ_1^I, (3) the set of transitions of $\mathcal{M}$ equals $\rightarrow_1 \cup \rightarrow_2 \cup (\sigma_1^A \times \sigma_2^I) \cup (\sigma_2^A \times \sigma_2^I)$, (4) the observation of a state s of $\mathcal{M}$ is $\langle\!\langle s\rangle\!\rangle_1$ if $s \in \Sigma_1$ and $\langle\!\langle s\rangle\!\rangle_2$ otherwise, (5) the repeating region of $\mathcal{M}$ is σ_2^A. Verify that an ω-word $\underline{a}$ is accepted by the Büchi automaton $\mathcal{M}$ precisely when it belongs to $L_1 \cdot L_2^\omega$. ■

Remark 11.5 [] If the regular language L_1 is accepted by an automaton with n_1 states, and the regular language L_2 is accepted by an automaton with n_2 states, then $L_1 \cdot L_2^\omega$ is accepted by a Büchi automaton with $n_1 + n_2$ states. ■

Theorem 11.1 [Büchi Theorem on Complementation Closure] *Given a Büchi automaton $\mathcal{M}$, there exists a Büchi automaton $\mathcal{M}'$ such that $\mathcal{L}_{\mathcal{M}'} = A^\omega \backslash \mathcal{L}_{\mathcal{M}}$.*

Proof. Let $\mathcal{M}$ be a Büchi automaton. The languages $L_{s,t}$ and $L'_{s,t}$ defined in the proof of Lemma 11.2 are regular languages. Since regular languages are closed under complement and intersection, from the proof of Lemma 11.2 it follows that every equivalence class of $\sim_{\mathcal{M}}$ is a regular language. By Proposition 11.4, for two equivalence classes L_1 and L_2 of $\sim_{\mathcal{M}}$, the ω-language $L_1 \cdot L_2^\omega$ is accepted by a Büchi automaton. Since Büchi automata are closed under union, from Corollary 11.1, it follows that $A^\omega \backslash \mathcal{L}_{\mathcal{M}}$ is accepted by some Büchi automaton. ■

Complexity of Complementation

Let us now analyze the bound on the size of the Büchi automaton accepting the complement of $\mathcal{L}_{\mathcal{M}}$ obtained by our construction. Suppose $\mathcal{M}$ has n states. Recall that $L_{s,t}$, for two states s and t of $\mathcal{M}$, is the language containing words $\overline{a}$ such that $s\,\alpha^T \overline{a}\, t$. It follows that there an automaton accepting $L_{s,t}$ with n states (use the same states and transitions of $\mathcal{M}$, but declare s to be initial and t to be accepting). The language $L'_{s,t}$ containing words $\overline{a}$ such that $s\,\alpha^T \overline{a}'\, t$ can be accepted by an automaton with $2n$ states (the state-space of $\mathcal{M}$ is doubled to keep track of whether an accepting state has been visited or not). It follows that the language $A^* \backslash L_{s,t}$ is accepted by an automaton with $\mathbf{2}^n$ states, and the language $A^* \backslash L'_{s,t}$ is accepted by an automaton with 4^n states (complementation may require determinization). Recall that an equivalence class of $\sim_{\mathcal{M}}$

corresponds to a subset Π' of the set Π containing $2n^2$ languages $L_{s,t}$, $L'_{s,t}$. Such an equivalence class Π' is defined by the product of the automata accepting L for $L \in \Pi'$ and the automata accepting $A^* \backslash L$ for $L \notin \Pi'$. Consequently, the bound on the size of the automaton accepting an equivalence class of $\sim_{\mathcal{M}}$ is $(\mathbf{2}^n)^{n^2} \cdot (4^n)^{n^2}$, which equals $\mathbf{2}^{3n^3}$.

The number of states of the Büchi automaton accepting $L_1 \cdot L_2^\omega$ equals the sum of the number of states of the automata accepting L_1 and L_2. Thus, for two equivalence classes L_1 and L_2 of $\sim_{\mathcal{M}}$, there is a Büchi automaton with $\mathbf{2}^{3n^3+1}$ states accepting $L_1 \cdot L_2^\omega$.

The number of states of the Büchi automaton accepting the union of ω-languages equals the sum of the number of states of the Büchi automata accepting the disjuncts. Since the number of pairs of equivalence classes of $\sim_{\mathcal{M}}$ is $\mathbf{2}^{4n^2}$, from Corollary 11.1, the next theorem follows.

Theorem 11.2 [Complexity of Büchi complementation] *Given a Büchi automaton $\mathcal{M}$ with n states, there exists a Büchi automaton $\mathcal{M}'$ with $\mathbf{2}^{3n^3+4n^2+1}$ states such that $\mathcal{L}_{\mathcal{M}'} = A^\omega \backslash \mathcal{L}_{\mathcal{M}}$.*

Note that to construct the desired complement automaton, we need to construct, for every pair L_1 and L_2 of equivalence classes of $\sim_{\mathcal{M}}$, the Büchi automaton accepting $L_1 \cdot L_2^\omega$, and check if it has a nonempty intersection with $\mathcal{L}_{\mathcal{M}}$. We have already outlined the product construction to obtain intersection of the languages accepted by two Büchi automata. Algorithms for checking fair cycles from Chapter 12 can be used to check for emptiness.

Remark 11.6 [Safra's Construction] The complementation construction presented here yields an automaton with $\mathbf{2}^{3n^3+4n^2+1}$ states. Better constructions are known. In particular, Safra's complementation algorithm produces an automaton with $\mathbf{2}^{O(n \cdot \log\ n)}$ states. This is essentially optimal: $\mathbf{2}^{n \cdot \log\ n}$ is a lower bound on the number of states necessary to define complements of a family of Büchi automata. ■

Complementing Streett automata

To establish that a Streett automaton can be complemented, we show that every Streett automaton has a language-equivalent Büchi automaton.

Proposition 11.5 [From Streett to Büchi] *Let $\mathcal{M}$ be a Streett automaton over A. There exists a Büchi automaton $\mathcal{M}'$ over A such that $\mathcal{L}_{\mathcal{M}} = \mathcal{L}_{\mathcal{M}'}$.*

Proof. Let $\mathcal{M} = (\Sigma, \sigma^I, \rightarrow, A, \langle\!\langle \cdot \rangle\!\rangle, F)$ be a Streett automaton. Recall that an ω-trajectory $\underline{s}$ is F-fair iff there exists a subset $F' \subseteq F$ of Streett constraints and an index $i \geq 0$ such that (1) for every $(\sigma, \tau) \in F$, $\underline{s}$ is τ-fair, and (2) for every $(\sigma, \tau) \notin F$, $s_j \notin \sigma$ for all $j \geq i$.

Suppose F has ℓ Streett constraints. The Büchi automaton $\mathcal{M}'$ has $\mathbf{2}^\ell + 1$ copies of the observation structure of $\mathcal{M}$, an initial copy together with a copy for every subset $F' \subseteq F$ of Streett constraints of $\mathcal{M}$. The automaton starts in the initial copy, and at some point, guesses the set $F' \subseteq F$ of Streett constraints (σ, τ) such that the region τ is going to repeat infinitely many times, and switches to the copy corresponding to F'. The copy corresponding to the set F' ensures that, for every $(\sigma, \tau) \in F'$, τ is visited infinitely often, and for every $(\sigma, \tau) \notin F'$, σ is not visited. To enforce that, for every $(\sigma, \tau) \in F'$, τ is visited infinitely often, we introduce a counter as in the translation from multi-Büchi constraints to Büchi constraint. To enforce that, for every $(\sigma, \tau) \notin F'$, σ is not visited, we delete the states in σ. The formal definition of $\mathcal{M}'$ is left as an exercise. ■

Remark 11.7 [] If $\mathcal{M}$ is a Streett automaton with n states and ℓ Streett constraints, the corresponding language-equivalent Büchi automaton constructed according to the proof of Proposition 11.5, has $n + n \cdot \ell \cdot \mathbf{2}^\ell$ states. Thus, simulating a set of Streett constraints by a single Büchi constraint causes a blow-up exponential in the number of constraints. Such a blow-up can be shown to be essential. ■

Given a Streett automaton, we can first construct the equivalent Büchi automaton, and then complement it using the complementation construction for Büchi automata.

Theorem 11.3 [Complementation of Streett automata] *Given a Streett automaton $\mathcal{M}$ with n states and ℓ Streett constraints, there exists a Büchi automaton $\mathcal{M}'$ with $\mathbf{2}^{O(n^3 \cdot 2^{3\ell})}$ states such that $\mathcal{L}_{\mathcal{M}'} = A^\omega \backslash \mathcal{L}_{\mathcal{M}}$.*

Exercise 11.6 {} [Complementing deterministic Streett automata] Let $\mathcal{M} = (K, F)$ be a Streett automaton such that K is a deterministic and complete observation structure. Show that if F is interpreted as a Rabin accepting condition, then the resulting language is the complement of $\mathcal{L}_{\mathcal{M}}$. ■

11.3 Expressiveness

11.3.1 ω-regular languages

In Chapter 11, we defined different ways of constructing ω-languages from languages of finite words. In particular, we defined the operators *safe*, *guar*, *recur*, and *persist*. If the languages to which these operators are applied are regular, then the resulting ω-languages are ω-regular.

ω-REGULAR LANGUAGES

The ω-language $\mathcal{L} \subseteq A^\omega$ is a *regular-safety language* if there is a regular language $L \subseteq A^*$ such that $\mathcal{L} = \mathit{safe}(L)$. The ω-language $\mathcal{L} \subseteq A^\omega$ is a *regular-guarantee language* if there is a regular language $L \subseteq A^*$ such that $\mathcal{L} = \mathit{guar}(L)$. The ω-language $\mathcal{L} \subseteq A^\omega$ is a *regular-response language* if there is a regular language $L \subseteq A^*$ such that $\mathcal{L} = \mathit{recur}(L)$. The ω-language $\mathcal{L} \subseteq A^\omega$ is a *regular-persistence language* if there is a regular language $L \subseteq A^*$ such that $\mathcal{L} = \mathit{persist}(L)$.

The ω-language $\mathcal{L} \subseteq A^\omega$ is *ω-regular* if it is a boolean combination of regular-response and regular-persistence languages.

Remark 11.8 [Normal form for ω-regular languages] Every ω-regular language is of the form

$$\bigcap 0 \leq i \leq k.\, \mathit{recur}(L_i) \cup \mathit{persist}(L'_i)$$

for regular languages L_i, L'_i. ■

Thus, the set of regular-safety languages is a subset of the the set of safety languages, etc. The set of ω-regular languages is a subset of the set of reactivity languages.

Example 11.7 [] Let $A = \{a, b, c\}$. The ω-language $(Aa)^\omega$ is regular-safe; the ω-language A^*aA^ω is regular-guarantee; the ω-language $(A^*a)^\omega$ is regular-response; the ω-language A^*a^ω is regular-persistence; and the ω-language consisting of ω-words with infinitely many b symbols or only finitely many a symbols is ω-regular. The ω-language consisting of ω-words $\underline{a}$ such that for all $i \geq 0$, if i is a prime number, then $a_i = a$, is safe but not regular-safe. ■

Closure properties of regular-safety, regular-guarantee, regular-response, regular-persistence, and ω-regular languages coincide with the corresponding closure properties of safety, guarantee, response, persistence, and reactivity languages, respectively. In particular, regular-safety and regular-guarantee classes are duals of each other, and so are regular-response and regular-persistence classes. These properties are summarized in the following proposition, and its proof follows from the constructions of Section 11.1.3.

Proposition 11.6 [Closure properties of ω-regular languages] *Regular-safety languages, regular-guarantee languages, regular-response languages, and regular-persistence languages are closed under union and intersection, but not under complementation. The ω-regular languages are closed under all boolean operations.*

11.3.2 Expressiveness of ω-automata

The ω-language $\mathcal{L} \subseteq A^\omega$ is said to be a *Büchi language* if $\mathcal{L}$ is accepted by some Büchi automaton. First, let us note that Büchi languages are closed under all boolean operations:

Proposition 11.7 [Closure of Büchi languages] *The class of Büchi languages is closed under union, intersection, and complementation.*

Since Büchi acceptance is a special case of Streett acceptance, and by Proposition 11.5 is powerful enough to admit translation from Streett constraints, it follows that

Corollary 11.2 [Streett acceptance vs. Büchi acceptance] *The ω-language $\mathcal{L} \subseteq A^\omega$ is a a Büchi language iff $\mathcal{L}$ is accepted by some Streett automaton.*

Exercise 11.7 {} [Acceptance by fair structures] Show that the ω-language $\mathcal{L} \subseteq A^\omega$ is a Büchi language iff $\mathcal{L}$ is the fair language of some finite fair structure. ■

Recall that every ω-regular language is a boolean combination of regular-response languages. Every regular-response language is accepted by a deterministic Büchi automaton.

Proposition 11.8 [From regular-response to deterministic Büchi] *For every regular language L, there exists a Büchi automaton that accepts the response language recur(L).*

Proof. Let L be a regular language. There exists a deterministic and complete automaton $M = (K, \sigma^A)$ such that $L_M = L$. Consider an ω-word $\underline{a}$. There exists precisely one initialized ω-trajectory $\underline{s}$ of K such that $\langle\!\langle \underline{s} \rangle\!\rangle = \underline{a}$. For every $i \geq 0$, the prefix $\overline{a}_{0 \ldots i}$ belongs to L iff $s_i \in \sigma^A$. Hence, the ω-word $\underline{a}$ belongs to *recur*(L) iff the ω-trajectory $\underline{s}$ is σ^A-fair. It follows that if we interpret M as a Büchi automaton it accepts the ω-language *recur*(L). ■

Corollary 11.3 [Lower bound on Büchi expressiveness] *For every ω-regular language $\mathcal{L}$, there exists a Büchi automaton that accepts $\mathcal{L}$.*

Proof. Follows from the definition of ω-regular languages, Proposition 11.7, and Proposition 11.8. ■

Conversely, the language accepted by every ω-automaton is ω-regular:

Proposition 11.9 [Upper bound on Büchi expressiveness] *Every Büchi language is ω-regular.*

Proof. To be added. ■

Exercise 11.8 {} [Expressiveness of Muller automata] Show that an ω-language ω-regular iff it is accepted by a Muller automaton. ■

Exercise 11.9 {} [ω-regular expressions] We have been using ω-regular expressions (e.g. $(A^*a)^\omega$) to specify ω-languages. Let us now formally define the syntax of ω-regular expressions. Given a finite alphabet A, the set of ω-regular expressions is defined by the grammar

$$\varphi := a \mid \varphi \cdot \varphi \mid \varphi + \varphi \mid \varphi^* \mid \varphi^\omega$$

where $a \in A$. Show that an ω-language $\mathcal{L}$ is defined by an ω-regular expression iff $\mathcal{L}$ is ω-regular. ■

Exercise 11.10 {} [Expressiveness of CoBüchi automata] Does the class of languages accepted by CoBüchi automata coincide with ω-regular languages? ■

11.3.3 Deterministic ω-automata

We have established that the nondeterministic varieties of different types ω-automata accept the same class of languages, namely, ω-regular languages. We proceed to understand the expressive power of different types of deterministic ω-automata.

The next proposition shows that every ω-regular language is accepted by some deterministic Streett automaton, and both nondeterministic and deterministic varieties of Streett automata have the same expressive power, namely, ω-regular languages.

Proposition 11.10 [Expressiveness of deterministic Streett] *For every ω-regular language $\mathcal{L}$, there exists a deterministic Streett automaton that accepts $\mathcal{L}$.*

Proof. Let $\mathcal{L}$ be an ω-regular language. Suppose $\mathcal{L} = \bigcap 0 \leq i \leq k. (\mathit{recur}(L_i) \cup \mathit{persist}(L_i'))$ such that the languages L_i, L_i' are regular. For $0 \leq i \leq k$, let $M_i = (K_i, \sigma_i)$ be a complete deterministic automaton accepting the regular language L_i, and let $M_i' = (K_i', \sigma_i')$ be a complete deterministic automaton accepting the regular language L_i'. Let K be the product of the $2k+2$ observation structures K_i, K_i', for $0 \leq i \leq k$. Since product construction preserves determinism, K is deterministic.

Let $\underline{a}$ be an ω-word. For every $0 \leq i \leq k$, the structure K_i has precisely one ω-trajectory $\underline{s}_i$ with the corresponding trace $\underline{a}$, and the word $\underline{a} \in \mathit{recur}(L_i)$ iff the ω-trajectory $\underline{s}_i$ is σ_i-fair. Similarly, for every $0 \leq i \leq k$, the structure K_i' has precisely one ω-trajectory $\underline{s}_i'$ with the corresponding trace $\underline{a}$, and the word

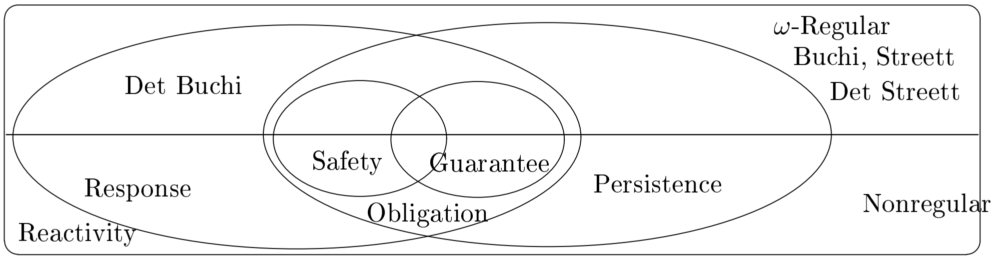


Figure 11.7: Classes of ω-regular languages

$\underline{a} \in \mathit{persist}(L'_i)$ iff the ω-trajectory $\underline{s}_i$ is not σ'_i-fair. The product structure K has precisely one ω-trajectory $\underline{s}$ with the trace $\underline{a}$. For all $0 \leq i \leq k$, the ω-trajectory $\underline{s}_i$ of K_i is σ_i-fair iff the ω-trajectory $\underline{s}$ of K is $(\sigma_i \Uparrow)$-fair (recall the definition of the lifting: $\sigma_i \Uparrow$ contains all product states whose component corresponding to the structure K_i is in σ_i). Similarly, for all $0 \leq i \leq k$, the ω-trajectory $\underline{s}'_i$ of K'_i is σ'_i-fair iff the ω-trajectory $\underline{s}$ of K is $(\sigma'_i \Uparrow)$-fair.

It follows that the word $\underline{a}$ belongs to $\mathcal{L}$ iff the ω-trajectory $\underline{s}$ of the product is, for all $0 \leq i \leq k$, either $(\sigma_i \Uparrow)$-fair or not $(\sigma'_i \Uparrow)$-fair. Hence, the deterministic Streett automaton $(K, \{(\sigma'_i \Uparrow, \sigma_i \Uparrow) \mid 0 \leq i \leq k\})$ accepts the ω-language $\mathcal{L}$. ■

Exercise 11.11 {} [Expressiveness of deterministic Rabin automata] Show that an ω-language is ω-regular iff it is accepted by some deterministic Rabin automaton. ■

It turns out that Büchi accepting condition is not expressive to capture all ω-regular languages, if we restrict to deterministic observation structures. The class of languages accepted by deterministic Büchi automata coincides with the regular-response languages.

Proposition 11.11 [Expressiveness of deterministic Büchi] *The ω-language $\mathcal{L} \subseteq A^\omega$ is accepted by a deterministic Büchi automaton iff it is a regular-response language.*

Proof. By Proposition 11.10, we know that every regular-response language is accepted by a deterministic Büchi automaton. For converse, consider a deterministic Büchi automaton $\mathcal{M} = (K, \sigma^A)$. Let L be the regular language accepted by the automaton with observation strcuture K and accepting region σ^A. Then, the Büchi automaton $\mathcal{M}$ accepts the regular-response language $\mathit{recur}(L)$. ■

It follows that deterministic Büchi automata are not closed under complementation. For instance, the response language "infinitely many a symbols" is accepted by a deterministic Büchi automaton, but its complement "only finitely many a symbols" is a persistence language, and is not accepted by any deterministic Büchi automaton. Intuitively, to define the persistence language consisting of ω-words with a suffix containing only b symbols using Büchi acceptance, the automaton must "guess" when the suffix containing only b symbols has commenced. The relationship between different classes of ω-languages is summarized in Figure 11.7

Exercise 11.12 {} [Union closure of deterministic Büchi automata] Since response languages are closed under union, from Proposition 11.11, it follows that deterministic Büchi automata are closed under union. Closure under union can, alternatively, be established by a direct construction. Give an algorithm to construct, given two deterministic Büchi automata $\mathcal{M}_1$ and $\mathcal{M}_2$, a deterministic Büchi automaton that accepts the union $\mathcal{L}_{\mathcal{M}_1} \cup \mathcal{L}_{\mathcal{M}_2}$. ■

Appendix: Notation

For two sets A and B, if σ is a subset of A then $\sigma \Uparrow$ denotes the subset $\{(a, b) \mid a \in \sigma\}$ of the product-set $A \times B$; if σ is a subset of B then $\sigma \Uparrow$ denotes the subset $\{(a, b) \mid b \in \sigma\}$ of the product-set $A \times B$. The lifting operator $\Uparrow$ can similarly be applied to binary relations over A to obtain binary relations over the product-set. The lifting operation generalizes to products of multiples sets also.