Automata and Reactive Systems

Lecture No. 8

Prof. Dr. Wolfgang Thomas

thomas@informatik.rwth-aachen.de

Lehrstuhl für Informatik VII

RWTH Aachen

4 Deterministic ω-Automata

We have seen:

Deterministic Büchi automata are too weak to recognize the set $L = (a+b)^* a^\omega$

How to define deterministic ω-automata which have the same power as nondeterministic Büchi automata?

Idea: To define successful runs, we fix precisely the states which should be visited infinitely often.

For a sequence $\rho \in Q^\omega$ define

$$\mathrm{Inf}(\rho) = \{q \in Q \mid q \text{ occurs infinitely often in } \rho\}$$

Muller automata

A (deterministic) Muller automaton has the form $\mathcal{A} = (Q, \Sigma, q_0, \delta, \mathcal{F})$ where

- Q, Σ, q_0 are as for Büchi automata
- $\delta : Q \times \Sigma \to Q$ is the transition function
- $\mathcal{F} \subseteq 2^Q$, i.e. $\mathcal{F} = \{F_1, \ldots, F_k\}$ for certain sets $F_1, \ldots, F_k \subseteq Q$

and where a run ρ is called **successful** if $\mathrm{Inf}(\rho) \in \mathcal{F}$ (Muller acceptance)

$\mathcal{A}$ **accepts** α if the unique run of $\mathcal{A}$ on α is successful.

$$L(\mathcal{A}) := \{\alpha \mid \mathcal{A} \text{ accepts } \alpha\}$$

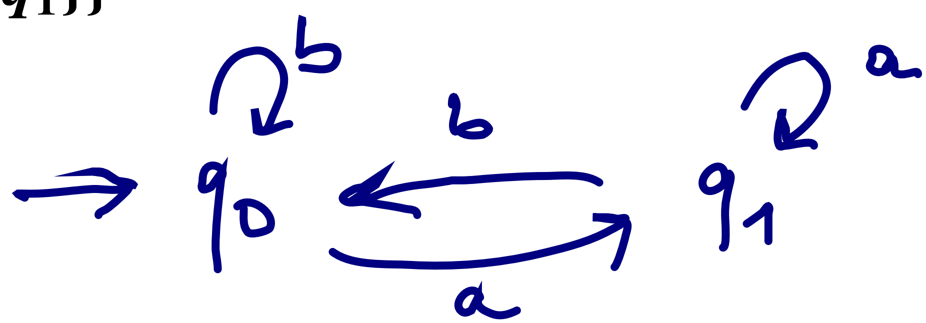
L is **Muller recognizable** if $L = L(\mathcal{A})$ for a Muller automaton $\mathcal{A}$

The Büchi acceptance condition (for the run ρ and the set F of final states) means $\mathrm{Inf}(\rho) \cap F \neq \emptyset$

The Muller condition (for $\mathcal{F} = \{F_1, \ldots, F_k\}$) means $\mathrm{Inf}(\rho) = F_i$ for some $i = 1, \ldots, k$

Example

$L = (a + b)^* a^\omega$ is recognized by the following Muller automaton with $\mathcal{F} = \{\{q_1\}\}$

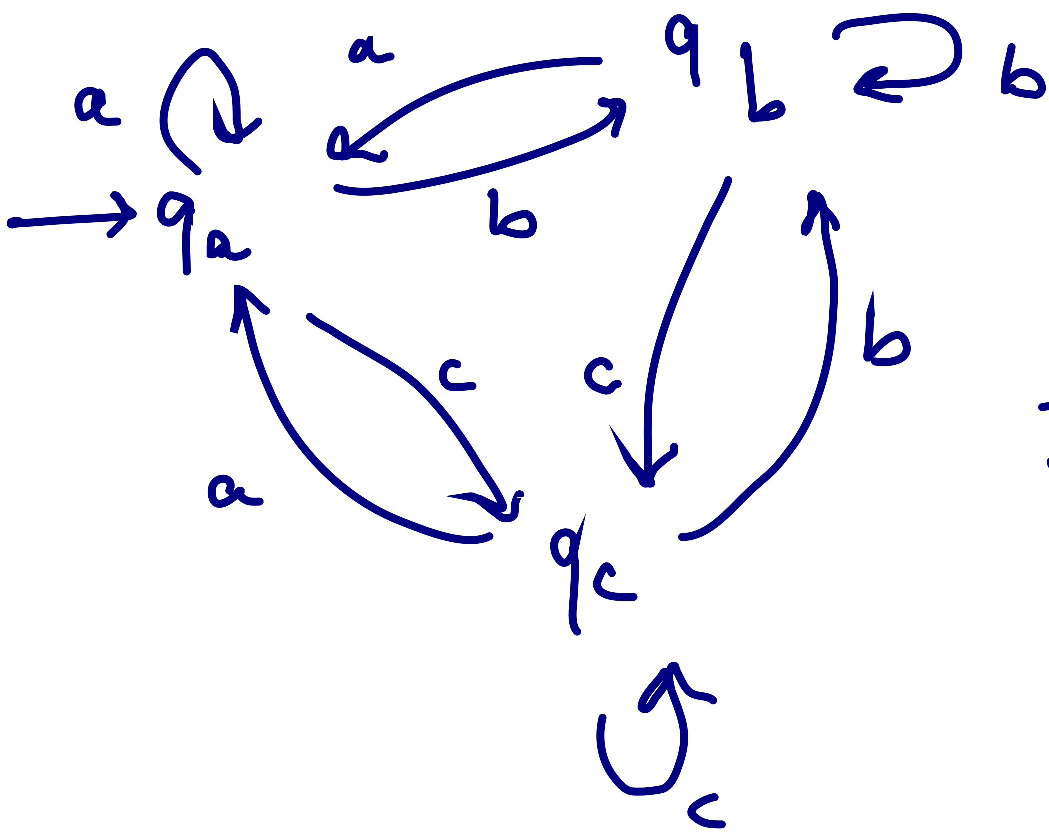


Example: Fairness condition

Let L be the set of ω-words over $\{a, b, c\}$ with the property

"if a occurs infinitely often, then b occurs infinitely often"

Muller automaton:



$P \subseteq Q$

$P \in \mathcal{F} :\Leftrightarrow$

$q_a \in P \to q_b \in P$

$\mathcal{F} = \{\{q_b\}, \{q_c\}, \{q_b, q_c\},$
$\{q_a, q_b\}, \{q_a, q_b, q_c\}\}$

Muller versus deterministic Büchi

4.1 Theorem: The Muller recognizable ω-languages are the boolean combinations of deterministic Büchi recognizable ω-languages.

Proof: First direction from left to right.

Assume $\mathcal{A} = (Q, \Sigma, q_0, \delta, \mathcal{F})$ recognizes L.

$\mathcal{A}$ accepts α

iff for some $F \in \mathcal{F}$:
$\mathcal{A}$ on α visits precisely the F-states infinitely often

iff $$\bigvee_{F \in \mathcal{F}} \Big(\bigwedge_{q \in F} \exists^\omega i : \delta(q_0, \alpha(0) \dots \alpha(i)) = q$$
$$\wedge \bigwedge_{q \in Q \setminus F} \neg \exists^\omega i : \delta(q_0, \alpha(0) \dots \alpha(i)) = q \Big)$$

Büchi automata as Muller automata

For the other direction show:

Boolean combinations of deterministic Büchi recognizable ω-languages are Muller recognizable.

Remark: Each deterministic Büchi automaton $\mathcal{A} = (Q, \Sigma, q_0, \delta, F)$ can be represented as a Muller automaton $\mathcal{A}' = (Q, \Sigma, q_0, \delta, \mathcal{F})$

with $P \in \mathcal{F} \; :\Leftrightarrow \; P \cap F \neq \emptyset$

4.2 Lemma: The class of Muller recognizable ω-languages is closed under boolean operations.

Boolean operations on Muller automata

The class of Muller recognizable ω-languages is closed under boolean operations.

Proof:

Complementation: From $\mathcal{A} = (Q, \Sigma, q_0, \delta, \mathcal{F})$

proceed to $\mathcal{A} = (Q, \Sigma, q_0, \delta, 2^Q \setminus \mathcal{F})$

Intersection: Given $\mathcal{A}_1$, $\mathcal{A}_2$ over Q_1, Q_2 and with acceptance components $\mathcal{F}_1, \mathcal{F}_2$.

Construct the product automaton over $Q_1 \times Q_2$ and define the acceptance component $\mathcal{F}$ as follows:

$\{(p_1, q_1), \ldots, (p_n, q_n)\} \in \mathcal{F}$

iff $\{p_1, \ldots, p_n\} \in \mathcal{F}_1$ and $\{q_1, \ldots, q_n\} \in \mathcal{F}_2$

Overview

We show that the following are equivalent for ω-languages

- being a Boolean combination of deterministic Büchi recognizable ω-languages
- deterministic Muller recognizable
- nondeterministic Büchi recognizable

4.3 Theorem: If L is Muller-recognizable then L is (nondeterministic) Büchi recognizable.

Proof: Assume $\mathcal{M} = (Q, \Sigma, q_0, \delta, \mathcal{F})$ recognizes L, with $\mathcal{F} = \{F_1, \ldots, F_k\}$.

Idea for an equivalent Büchi automaton $\mathcal{B}$:

$\mathcal{B}$ simulates $\mathcal{M}$ and guesses which set F_i is the correct infinity set and from which point onwards precisely the F_i-states are visited again and again.

Introduce memory for accumulating F_i-states. When F_i is full then reset to $\emptyset$ (“final state”).

Implementation

Define the Büchi automaton $\mathcal{B} = (Q', \Sigma, q_0{}', \Delta, F)$:

- $Q' = Q \cup (Q \times 2^Q \times \{1, \dots, k\})$
- $q_0{}' = q_0$
- $F = \{(p, \emptyset, j) \mid p \in Q, j \in \{1, \dots, k\}\}$
- Δ contains the following transitions ($j \in \{1, \dots, k\}$):

$$
\begin{array}{ll}
(p, a, q) \text{ and } (p, a, (q, \emptyset, j)) & \text{if } \delta(p, a) = q \\
((p, P, j), a, (q, P \cup \{q\}, j)) & \text{if } \delta(p, a) = q \text{ and } P \cup \{q\} \subsetneqq F_j \\
((p, P, j), a, (q, \emptyset, j)) & \text{if } \delta(p, a) = q \text{ and } P \cup \{q\} = F_j
\end{array}
$$

McNaughton's Theorem (1966)

4.4 Theorem: If L is Büchi recognizable then L is recognizable by a deterministic Muller automaton.

This is the main theorem of the theory of ω-automata.

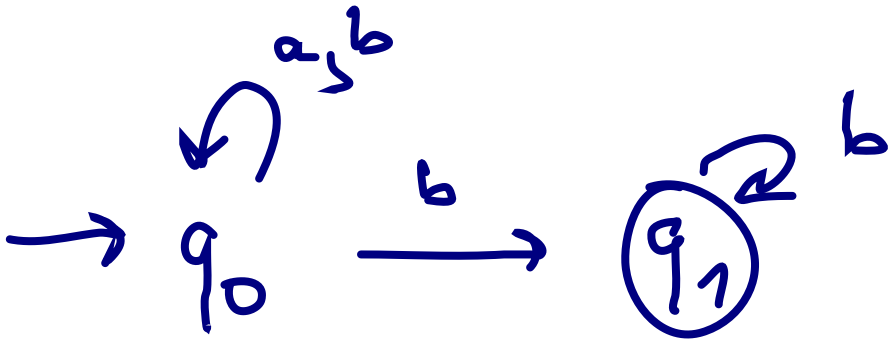
Given Büchi automaton $\mathcal{B} = (Q, \Sigma, q_0, \Delta, F)$.

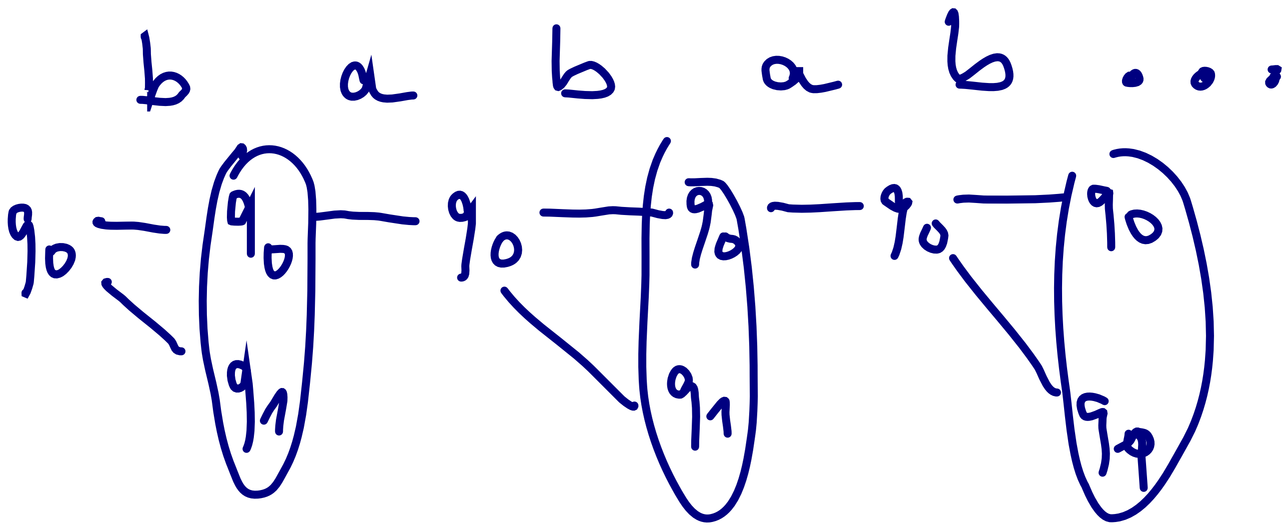
First try: Powerset construction

Determine after each input-prefix w the set of states reachable via w,

and declare as final states the sets which contain an F-state.

Illustration





Safra trees

We present the Safra construction (S. Safra 1988)

Idea: Branch off a separate computation thread starting from final states

To record these different computation branches we use a tree structure.

A Safra tree over Q is an ordered finite tree

- with node names from $\{1, \ldots, 2|Q|\}$,
- where each node is labelled by a nonempty set $R \subseteq Q$, possibly with an extra marker "!"
- where labels of brother nodes are disjoint
- where the union of brother nodes is a proper subset of the parent node

Definition of the Muller automaton

Remark: There are only finitely many possible Safra trees over Q.

For the given Büchi automaton $\mathcal{B} = (Q, \Sigma, q_0, \Delta, F)$

define the Muller automaton $\mathcal{M} = (Q', \Sigma, q_0{}', \delta, \mathcal{F})$:

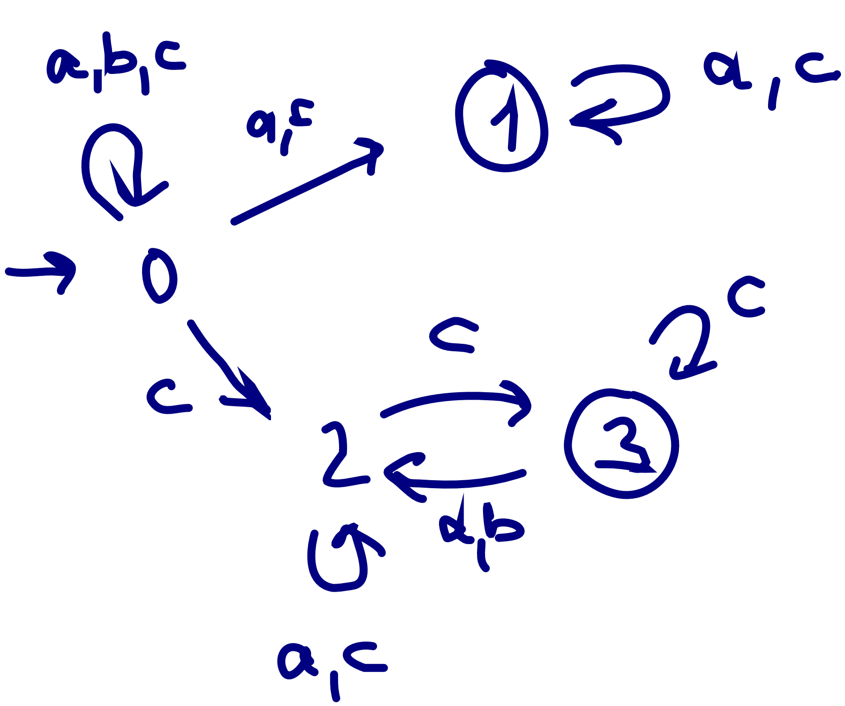
- Q' := set of Safra trees over Q.
- $q_0{}'$:= Safra tree consisting just of root labelled $\{q_0\}$
- Define $\delta(s, a)$ (for Safra tree s, input letter a) in four stages as described below
- Declare a set S of Safra trees to be in $\mathcal{F}$ if some node name appears in each tree $s \in S$, and in some tree $s \in S$ the label of this node name carries the marker “!”

A transition of $\mathcal{M}$

Define $\delta(s, a)$ (for Safra tree s, input letter a) in four stages as follows:

1. For each node whose label contains final states, branch off a new son containing these final states. (Take as node name a free number $\leq 2|Q|$)
2. To each node label apply the powerset construction via input letter a: $R \to \{r' \mid \exists r \in R : (r, a, r') \in \Delta\}$
3. Cancel state q if it occurs also in an older brother node. Cancel a node if it carries label $\emptyset$ (unless it is the root)
4. Cancel all sons (and their descendants) if the union of their labels is the parent label, and in this case mark the parent by “!”

Example



INPUT:
c c b c b