Automata and Reactive Systems

Lecture No. 10

Prof. Dr. Wolfgang Thomas

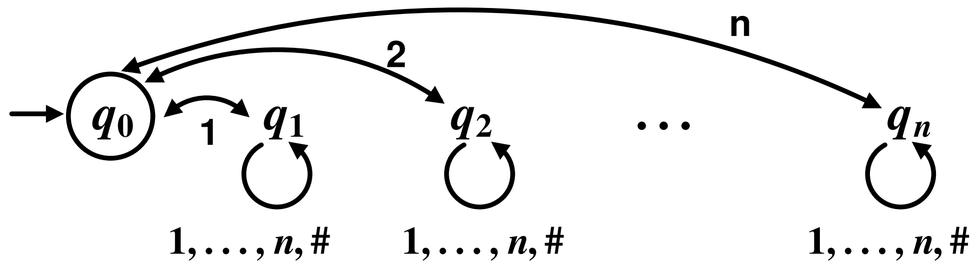
thomas@informatik.rwth-aachen.de

Lehrstuhl für Informatik VII

RWTH Aachen

ω-language L_n

Definition of L_n by a Büchi automaton $\mathcal{B}_n$ over the alphabet $\{1, \ldots, n, \#\}$



4.11 Theorem: **Any deterministic Rabin automaton which recognizes L_n must have $\geq n!$ states.**

Remark:

$\alpha \in L_n \iff$

$(*)$ there are letters $i_1, \ldots, i_k \in \{1, \ldots, n\}$ (pairwise distinct) such that α contains i_1, and the segments of letter pairs $i_1 i_2, i_2 i_3, \ldots, i_{k-1} i_k, i_k i_1$ occur infinitely often in α

Application of $(*)$:

4.12 Lemma (Permutation Lemma):

For each permutation $(i_1 \ldots i_n)$ of $(1, \ldots, n)$, the ω-word $(i_1 \ldots i_n \#)^\omega$ does not belong to L_n.

Union Lemma

Let $\mathcal{R} = (Q, \Sigma, q_0, \delta, \Omega)$ be a Rabin automaton with $\Omega = ((E_1, F_1), \ldots, (E_k, F_k))$. Let $\rho_1, \rho_2, \rho \in Q^\omega$ be runs with $\mathrm{Inf}(\rho_1) \cup \mathrm{Inf}(\rho_2) = \mathrm{Inf}(\rho)$.

If ρ_1 and ρ_2 are not successful, then ρ is not successful.

Proof: Assume ρ_1, ρ_2 are not successful but ρ is.

Then some $j \in \{1, \ldots, k\}$ exists with $\mathrm{Inf}(\rho) \cap E_j = \emptyset$ and $\mathrm{Inf}(\rho) \cap F_j \neq \emptyset$.

By assumption of Lemma: $\mathrm{Inf}(\rho_1) \cup \mathrm{Inf}(\rho_2) = \mathrm{Inf}(\rho)$

Hence $\mathrm{Inf}(\rho_1) \cap E_j = \mathrm{Inf}(\rho_2) \cap E_j = \emptyset$

and ($\mathrm{Inf}(\rho_1) \cap F_j \neq \emptyset$ or $\mathrm{Inf}(\rho_2) \cap F_j \neq \emptyset$).

So ρ_1 or ρ_2 is successful, a contradiction.

Proof of Theorem

Assume the Rabin automaton $\mathcal{R}$ recognizes L_n.

We have to show that $\mathcal{R}$ has $\geq n!$ states.

Consider two distinct permutations $(i_1, \ldots, i_n), (j_1 \ldots, j_n)$ of $1, \ldots, n$.

The ω-words $\underbrace{(i_1 \ldots i_n\#)^\omega}_{\alpha}$, $\underbrace{(j_1 \ldots j_n\#)^\omega}_{\beta}$

are not accepted by $\mathcal{R}$ (Permutation Lemma!)

Consider the two non-accepting runs ρ_α, ρ_β of $\mathcal{R}$ on α, β, and set $R := \mathrm{Inf}(\rho_\alpha)$ and $S := \mathrm{Inf}(\rho_\beta)$.

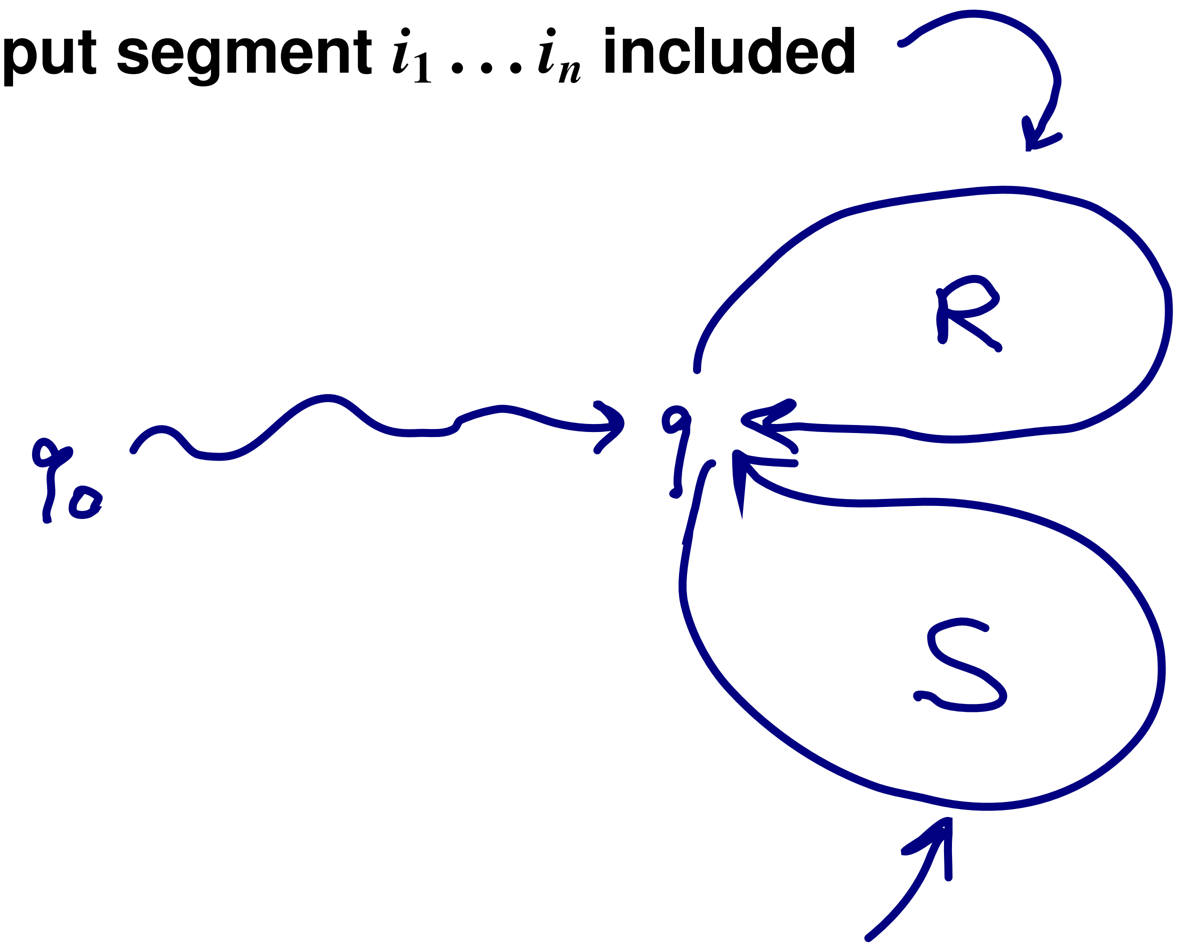
Claim: $R \cap S = \emptyset$.

Then $\mathcal{R}$ must have as many states as permutations, i.e. $\geq n!$ states.

For contradiction assume $q \in R \cap S$.

Construct from ρ_α, ρ_β a new run on a new input word:

input segment $i_1 \ldots i_n$ included



input segment $j_1 \ldots j_n$ included

We repeat the two loops in alternation, get a new input word γ and a new run ρ on γ with $\mathrm{Inf}(\rho) = R \cup S$.

$R \cap S = \emptyset$

$q \in R \cap S$

By the Union Lemma we know that ρ is not successful.

But we now show that γ has the cycle property $(*)$ and hence $\gamma \in L_n$ (contradiction)

In γ both $i_1 \ldots i_n$ and $j_1 \ldots j_n$ occur infinitely often.

Since $i_1 \ldots i_n \neq j_1 \ldots j_n$, we may pick the smallest k with $i_k \neq j_k$. Then:

$$\begin{array}{ccccccccc} i_1 & \ldots & i_{k-1} & i_k & i_{k+1} & \ldots & i_n \\ \parallel & & \parallel & \nparallel & & & \\ j_1 & & j_{k-1} & j_k & j_{k+1} & \ldots & j_n \end{array}$$

$$\begin{array}{cccccccc} i_1 & \ldots & i_{k-1} & i_k & i_{k+1} & \ldots & i_n \\ \shortparallel & & \shortparallel & \nparallel & & & \\ j_1 & & j_{k-1} & j_k & j_{k+1} & \ldots & j_n \end{array}$$

There is i_l with $l > k$, such that $i_l = j_k$.

Similarly there is j_r with $r > k$ such that $j_r = i_k$.

$$\begin{array}{c} i_k i_{k+1}, \ldots, i_{l-1} \underset{\substack{\shortparallel \\ j_k}}{i_l}\ , j_k j_{k+1}, \ldots, j_{r-1} \underset{\substack{\shortparallel \\ i_k}}{j_r}\ , i_k i_{k+1}, \ldots \end{array}$$

We get a cycle of letter pairs as in condition $(*)$.

So $\gamma \in L_n$

in contradiction to the fact that run ρ is not successful.

Open Problem

Question:

Does the same lower bound of $n!$ states also hold for the transformation from Büchi automata to deterministic Muller automata?

This is an open problem.

We would need a new family of example languages.

Exercise: **The language L_n is accepted by a deterministic Muller automaton with n^2 states.**

5 Logical Application: From S1S to Büchi automata

A consequence of McNaughton's Theorem:

The class of Büchi recognizable ω-languages is closed under complement.

Proof:

Given a Büchi automaton $\mathcal{B}$, construct a Büchi automaton for the complement ω-language as follows:

1. **From $\mathcal{B}$ obtain an equivalent deterministic Muller automaton $\mathcal{M}$ by Safra's construction**
2. **In $\mathcal{M}$ declare the non-accepting state sets as accepting and vice versa and thus obtain $\mathcal{M}'$**
3. **From $\mathcal{M}'$ obtain an equivalent Büchi automaton $\mathcal{B}'$**

We have shown:

A Büchi-recognizable ω-language is S1S definable.

Now we prove the converse:

5.1 Theorem:

An S1S-definable ω-language is Büchi recognizable.

There will be two stages:

1. **Reduction of S1S to a simpler formalism $S1S_0$**
2. **Construction of Büchi automata by induction on $S1S_0$-formulas.**

Preparations: From S1S to $S1S_0$

We eliminate some constructs from S1S:

The constant 0 can be eliminated: Instead of $X(0)$ write

$$\exists t(X(t) \wedge \neg\exists s(s < t))$$

The relation symbol $<$ can be eliminated: Instead of $s < t$ write

$$\forall X(X(s') \wedge \forall y(X(y) \rightarrow X(y')) \;\rightarrow\; X(t))$$

("each set which contains s' and is closed under successor must contain t")

The successor function only occurs in formulas $x' = y$: Instead of $X(s'')$ write

$$\exists y \exists z(s' = y \wedge y' = z \wedge X(z))$$

S1S$_0$

Eliminate the use of first-order variables by using different atomic formulas:

$X \subseteq Y$, $\mathrm{Sing}(X)$, $\mathrm{Succ}(X, Y)$, meaning:

"X is subset of Y", "X is a singleton set", and

"$X = \{x\}$, $Y = \{y\}$ are singleton sets with $x + 1 = y$"

Now one can write

$X(y)$ as $\mathrm{Sing}(Y) \wedge Y \subseteq X$

$x' = y$ as $\mathrm{Succ}(X, Y)$

Translation example: $\forall x\, \exists y(x' = y \wedge Z(y))$ is written as

$$\forall X(\mathrm{Sing}(X) \rightarrow \exists Y(\mathrm{Sing}(Y) \wedge \mathrm{Succ}(X, Y) \wedge Y \subseteq Z))$$

5.1 Theorem:

An S1S-definable ω-language is Büchi recognizable.

Proof:

We can assume that S1S-formulas $\varphi(X_1, \ldots, X_n)$ are rewritten as S1S_0-formulas.

Show the claim by induction on S1S_0-formulas:

It suffices to treat

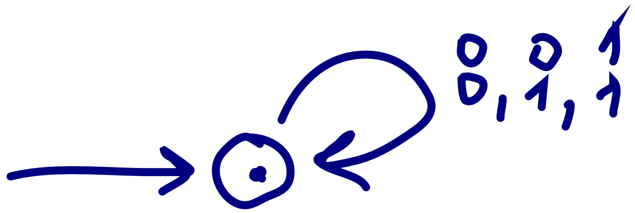
- **the atomic formulas**
  $X_1 \subseteq X_2, \quad \mathrm{Sing}(X_1)\ , \quad \mathrm{Succ}(X_1, X_2)$
- **the connectives $\vee$ and $\neg$ and the existential set quantifier $\exists$.**

Büchi automata for the formulas

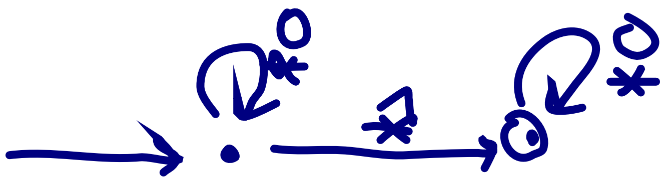
$\overset{1}{X_1 \subseteq X_2}$, $\overset{2}{\mathrm{Sing}(X_1)}$, $\overset{3}{\mathrm{Succ}(X_1, X_2)}$

$X_1 = 0\ 0\ 1\ 1\ 0\ 1 \ldots$
$X_2 = 0\ 1\ 0\ 1\ 0\ 1 \ldots$

1

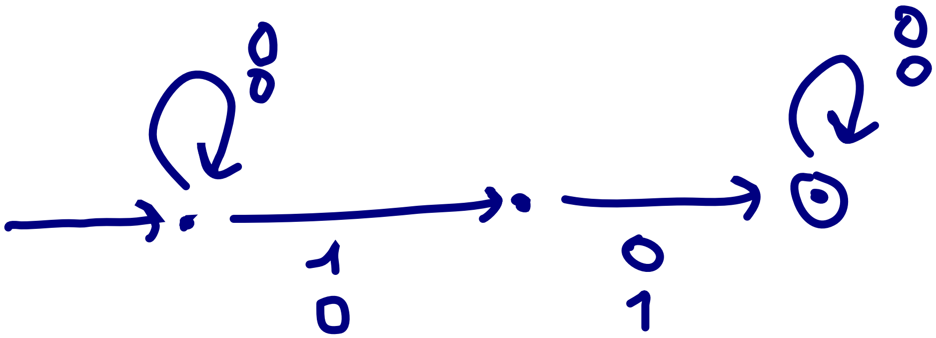


2



$X_1 = 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0 \ldots$

3



$X_1 = 0\ 0\ 0\ 1\ 0\ 0\ 0 \ldots$
$X_2 = 0\ 0\ 0\ 0\ 1\ 0\ 0$

Induction step

1. Consider $\varphi_1(X_1, \ldots, X_n) \vee \varphi_2(X_1, \ldots, X_n)$

   By induction hypothesis assume Büchi automata $\mathcal{A}_1, \mathcal{A}_2$ are equivalent to φ_1, φ_2. Take the Büchi automaton for the union.

2. Consider $\neg\varphi(X_1, \ldots, X_n)$

   By induction hypothesis there is a Büchi automaton equivalent to φ.

   Apply the closure of Büchi recognizable ω-languages under complement, to obtain a Büchi automaton equivalent to $\neg\varphi$.

3. **Existential quantifier:** Consider $\exists X \varphi(X, X_1, \ldots, X_n)$

Assume $\mathcal{A}$ is a Büchi automaton equivalent to $\varphi(X, X_1, \ldots, X_n)$

In $\mathcal{A}$, change each transition label $(b, b_1, \ldots, b_n)$ into $(b_1, \ldots, b_n)$; thus obtain $\mathcal{A}'$.

Then a transition via $\overline{b}$ in $\mathcal{A}'$ amounts to the existence of a transition via $(0, \overline{b})$ or $(1, \overline{b})$ in $\mathcal{A}$.

$\mathcal{A}'$ accepts $\alpha \in (\mathbb{B}^n)^\omega$

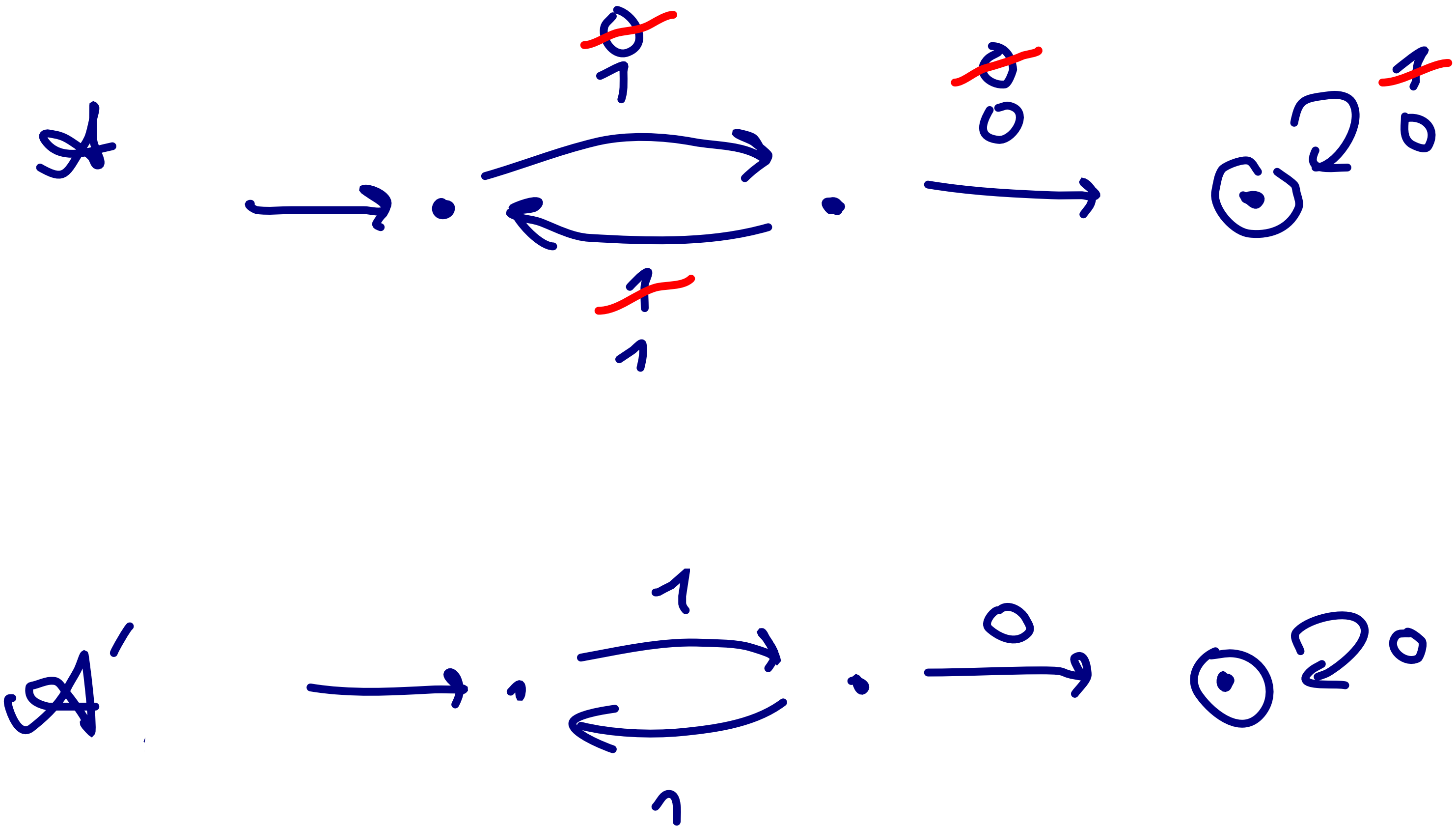
iff there exists a bit sequence $c_0 c_1 \ldots$ such that $(c_0, \alpha(0)), (c_1, \alpha(1)) \ldots$ is accepted by $\mathcal{A}$

iff $\exists c_0 c_1 \ldots$ such that $\mathcal{A}$ accepts $(c_0, \alpha(0)), (c_1, \alpha(1)) \ldots$

iff $\alpha \models \exists X \varphi(X, X_1, \ldots, X_n)$

So the Büchi automaton $\mathcal{A}'$ is equivalent to $\exists X \varphi(X, X_1, \ldots, X_n)$

Illustration



Complexity of Logic-Automata Translation

We have translated LTL- and S1S-formulas into Büchi automata.

The complexity bounds are very different.

We define the k-fold exponential function g_k by

$$g_0(n) = n, \quad g_{k+1}(n) = 2^{g_k(n)}$$

$$\left.2^{2^{\cdot^{\cdot^{2^{n}}}}}\right\} k$$

5.2 Theorem:

1. **An LTL formula of size n (measured in the number of subformulas) can be translated into a Büchi automaton with 2^n states**
2. **There is no k such that each S1S formula of size n (measured in the number of subformulas) can be translated into a Büchi automaton with $g_k(n)$ states**

The case of sentences

A **sentence** is a formula without free variables.

The translation of a sentence φ into a Büchi automaton $\mathcal{A}_\varphi$ yields an automaton with unlabelled transitions.

The sentence φ is true in the structure $(\mathbb{N}, +1, <, 0)$ iff the automaton $\mathcal{A}_\varphi$ has a successful run.

The latter condition can be checked as in the nonemptiness test.

Consequence: One can decide, for any given S1S-sentence φ, whether φ is true in $(\mathbb{N}, +1, <, 0)$ or not.

Monadic second-order theory of successor

The monadic second-order theory of $(\mathbb{N}, +1, <, 0)$ is the set of S1S-sentences which are true in $(\mathbb{N}, +1, <, 0)$

Notation: $\mathbf{MTh_2}(\mathbb{N}, +1, <, 0)$

Example sentences:

$\forall X \, \exists Y (\forall t (X(t) \rightarrow Y(t)))$ true

$\forall X \, \exists t \, \forall s (X(s) \rightarrow s < t)$ false

$\forall X (X(0) \wedge \forall s (X(s) \rightarrow X(s')) \rightarrow \forall t X(t))$ true

5.3 Theorem (Büchi 1960):

The theory $\mathbf{MTh_2}(\mathbb{N}, +1, <, 0)$ is decidable.