Automata on Infinite Words

gehalten von Univ.-Prof. Dr. rer. nat. Wolfgang Thomas
im Wintersemester 2005/06 an der RWTH Aachen

eine studentische Mitschrift von
Florian Heller
florian@heller-web.net
Diese Mitschrift erhebt keinen Anspruch auf Richtigkeit oder Vollständigkeit.
- Think Different -

February 7, 2006

1 Introduction

18.10.05

1.1 The Theory

The Theory has been formed by Dr. Richard Büchi, Trakhtenbrot, Rabin, McNaughton. The subject is the analysis of finite Automata working on infinite Words.

Motivation

1. Attractive theory with algorithmic content.
2. Framework to non-terminating reactive systems.
3. Connection to logic (temporal logic and others)

A Script of this lecture is available on the website of Informatik 7

1.2 Exercises

Starting tomorrow.
Hand in the Solutions in Groups of three persons.

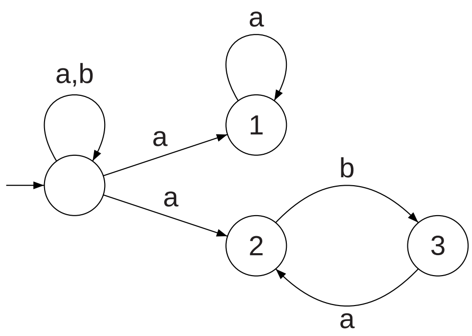
1.3 Büchi automata and regular ω-languages

Σ Alphabet, $\mathbb{B} = \{0, 1\}$, $a, b, c, \ldots$ letters, $u, v, w, \ldots$ finite words, ε empty word, Σ^+, Σ^*
$\alpha = \alpha(0)\alpha(1)\alpha(2)\ldots$ ω-Word [over Σ if $\alpha(i) \in \Sigma$]
Σ^ω set of ω-words over Σ
$\alpha[i \ldots j] = \alpha(i) \ldots \alpha(j)$
$U, V, W, \ldots$ languages of finite words. $K, L, \ldots$ languages of ω-words

Definition (Büchi Automaton)
$\mathcal{A}$ Büchi automaton is of the form $\mathcal{A} = (Q, \Sigma, q_0, \Delta, F)$ with finite state set Q, input alphabet Σ, initial state q_0, transition relation $\Delta \subseteq Q \times \Sigma \times Q$ and a set $F \subseteq Q$ of accepting (or final) states.
accepting ω-wordby the " Büchi condition "
A run of $\mathcal{A}$ on α is a sequence $\rho = \rho(0)\rho(1)\ldots$ s.t. $\rho(0) = q_0, (\rho(i), \alpha(i), \rho(i+1)) \in \Delta$ for $i \geq 0$
ρ satisfies the Büchi condition (ρ accepting) if $\rho(i) \in F$

A accepts α if exists run ρ of A on α wich satisfies the Büchi condition
$L(A) = \{\alpha \in \Sigma^\omega | A accepts \alpha\}$

Example:
$\mathcal{A}_0$:



$\alpha = abbababab\dots$
$\rho_1 = 0000000000\dots$
$\rho_2 = 0000002323\dots$
$\rho_3 = 0000232323\dots$
ρ_2, ρ_3 accepting. $\mathcal{A}_0$ accepts α $L(A_0)$ = the set of all ω-words over $\{a, b\}$ with from some point onwards have $ababab\dots$ or $aaaa\dots$.
Short notation: $(a+b)^*(ab)^\omega + (a+b)^*a^\omega$ "regular expression"

Questions

1. Reduction to deterministic automata?
2. Alternative characterization of accepted (or recognized) ω-languages?
3. Closure under operations like $\cap, \cup$
4. Algorithmic properties

Remark on determinism In a deterministic Büchi automaton replace Δ by a transition function $\delta : Q \times \Sigma \to Q$
Then an ω-wordα induces a unique run ρ of $\mathcal{A}$ on α.
$\rho(0) = q_0$
$\rho(1) = \delta(q_0, \alpha(0))$
$\rho(2) = \delta(\delta(q_0, \alpha(0)), \alpha(1))$
Büchi condition as before.
$L_1 = \{\alpha \in \{a, b\}^\omega |$ from some point onwards, in α only a occurs$\}$
$L_2 = \{\alpha \in \{a, b\}^\omega |$b occurs only finitely often in$\alpha\} = (a+b)^*a^*\omega$
Claim: L_A is Büchi recognizable, but not deterministically Büchi rec.

Proof:
Assume det. Büchi aut. $\mathcal{A}$ recognizes L_1
Consider $\mathcal{A}$ on $aaa\dots$. $\mathcal{A}$ visits final states infinitely often in its unique run, say first time after n_0 letters a.
Consider $\mathcal{A}$: $a^{n_0}baa\dots$ Next visit to final state is guaranteed by assumption, say after prefix

$a^{n_0}ba^{n_1}$
Consider$\mathcal{A}$ on $a^{n_0}ba^{n_1}baa\ldots$
Generate infinite word $a^{n_0}, ba^{n_1}ba^{n_2}b\ldots$ where the $\mathcal{A}$-run visits final states infinitely often.
Contradiction to the assumption on $\mathcal{A}$

1.4 Towards characterization of Büchi recognizable ω-languages

1.4.1 Preparation

- Given $U \subseteq \Sigma^*$, define $U^\omega = \{\alpha \in \Sigma^\omega | \alpha = u_0u_1u_2\ldots, u_i \in U\}$

  Example:
  $U = abba^* + aa$
  U^ω contains $\underbrace{aa}\underbrace{abb}\underbrace{abba}\underbrace{abbaa}abbaaa\ldots$

- Given $U \subseteq \Sigma^*, L \subseteq \Sigma^\omega$, $U \cdot L = \{\alpha \in \Sigma^\omega | \alpha = u\beta, u \in U, \beta \in L\}$

Theorem 1 *$L \subseteq \Sigma^\omega$ is Büchi recognizable $\Leftrightarrow L = \bigcup_{i=1}^{n} U_i \cdot V_i^\omega$ with $U_i, V_i \subseteq \Sigma^*$ regular.*

Proof:
$\Rightarrow$ Given $\mathcal{A} = (Q, \Sigma, q_0, \Delta, F)$ Büchi automaton
Define for $p, q \in Q$ $W_{p,q} = \{w \in \Sigma^* |$ ex. $\mathcal{A}$-run from p to q via $w\}$
$W_{p,q}$ is regular (use $(Q, \Sigma, p, \Delta, \{q\})$)
Remark: $\mathcal{A}$ accepts α if for some $q \in F$ α is in $W_{q_0q} \cdot W_{qq} \cdot W_{qq}\ldots$ $L = \bigcup_{q\in F} W_{q_0q} \cdot W_{qq}^\omega$ $(*)$

Consequence: α is ultimately periodic if $\alpha = uvvv\ldots$ for some fixed words u, v
Proposition: L Büchi recognizable, $L \neq \emptyset \Rightarrow L$ contains an ultimately periodic ω-word

Proof:
Given $\mathcal{A}$, consider the representation $(*)$
For some q $W_{q_0q} \neq \emptyset$ $W_{qq} \neq \emptyset$ Using $u \in W_{q_0q}, v \in W_{qq}$ find $\alpha = uvvv\ldots$

Definition
$L \subseteq \Sigma^\omega$ is Büchi recognizable $\Leftrightarrow$ L is finite union of sets $U \cdot V^\omega$, with $U, V \subseteq \Sigma^*$ regular

Proof:
Proof of $\Leftarrow$) Lemma:

a) $V \subseteq \Sigma^*$ regular $\Rightarrow$ V^ω Büchi recognizable (B-Rec).

b) $U \subseteq \Sigma^* reg, K \subseteq \Sigma^\omega B - Rec \Rightarrow L_1 \cup L_2 B - rec$

Proof a) Given NFA $\mathfrak{A} = (Q, \Sigma, q_0, \Delta, F)$ recognizing V
Preprocessing: Introduce new initial state q_0' wich cannot be reached via nenempty word, obtain equiv. NFA $\mathfrak{A}$
Construct the Büchi-atuomaton $\mathfrak{B}$ for V^ω from a′

- For any transition (p, a, q) with $q \in F$ introduce new transition (p, a, q_0')

- Use $\{q_0'\}$ as set of final states of $\mathfrak{B}$

Proof b) Given NFA $\mathfrak{A}$ for U, Büchi automaton $\mathfrak{B}$ for K Introduce over $Q_{\mathfrak{A}} \cup Q_{\mathfrak{B}}$ for (p, a, q) with $q \in F_{\mathfrak{A}}$ new transistion $(p, a, q_{0\mathfrak{B}}$ if $q_{0\mathfrak{A}} \in F_{\mathfrak{A}}$ for $(q_{0\mathfrak{B}}, b, q)$ new transition $(q_{0\mathfrak{A}}, b, q)$
Proof c) Given $\mathfrak{B}_1, \mathfrak{B}_2$ for L_1, resp L_2
Introduce a new initial state q_0 and new initial transitions

Consequence Büchi-recognizable ω-languages are described by regular ω-expressions (ω-regular expressions) $r_1 \cdot s_1^\omega + \cdots + r_k \cdot s_k^\omega$ where r_i, s_i are standard regular expressions.

1.5 Complementation of Büchi automata.

Theorem 2 *$L \subseteq \Sigma^\omega$ B-rec. $\Rightarrow \Sigma^\omega \setminus L$ B-rec*

Strategy: Given Büchi-automaton $\mathfrak{A} = (Q, \Sigma, q_0, \Delta, F)$ recognizing L
Define finite family $\mathcal{W}_{\mathfrak{A}} = \{W_1, \ldots, w_k\}$ of regular languages $W_i \subseteq \Sigma^*$ such that

- L is finite union of sets $U \cdot V^\omega$ with $U, V \in \sqsupseteq_{\mathfrak{A}}$
- $\Sigma^\omega \setminus L$ ist also finite union of sets with $U, V \in \sqsupseteq_{\mathfrak{A}}$

notation (Given $\mathfrak{A}$) write $p \xrightarrow{w} q[p \overset{w}{\Rightarrow} q] : \Leftrightarrow$ ex run of $\mathfrak{A}$ on w from p to q [Such than a final state is visited in this run]
Definition
Given $\mathfrak{A}$ define for $u, v \in \Sigma^*$
$u \sim_{\mathfrak{A}} v \Leftrightarrow$ for each $p, q \in Q$
$p \xrightarrow{u} q \Leftrightarrow p \xrightarrow{v} q$
$p \xrightarrow{u} q \Leftrightarrow p \xrightarrow{v} q$

Fact 1: $\sim_{\mathfrak{A}}$ is equivalence relation, call the equivalence classes $\sim_{\mathfrak{A}}$-classes $[u]$

Fact 2: Each $\sim_{\mathfrak{A}}$-class is regular
$w \in [u] \Leftrightarrow \forall p, q \in Q$ s.t. $p \xrightarrow{u} q$ [not $p \xrightarrow{u} q$] $w \in \mathcal{W}_{pq}[w \notin \mathcal{W}_{pq}]$ and $\forall p, q \in Q$ s.t. $p \overset{u}{\Rightarrow} q[not p \overset{u}{\Rightarrow} q]$ $w \in \mathcal{W}_{pq}^F$ (allowing $\mathfrak{A}$ to go from p to q visiting F)

$L \subseteq \Sigma^\omega$ Büchi-Rec. $\Rightarrow \Sigma^\omega \setminus$ Büchi-Rec. $\mathcal{A}$ Büchi aut. for L $U \cdot V^\omega$

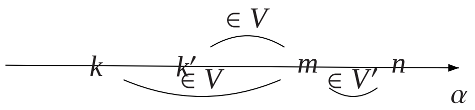
Remark:
From the transition profiles of u, v one can compute the transition profile of uv.
Other Formulation:

- $u \sim_a u'$ $v \sim_a v'$ $uv \sim_a u'v'$

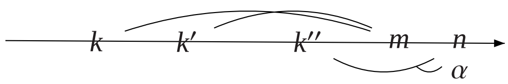
- $\sim_a$ is a congruence
- $U, V \in \mathfrak{W}_a$ ($U, V \sim_a$ classes)
  $U \cdot V \subseteq W$ for some W

Consequence of Lemma 1,2: $\Sigma^\omega \setminus L = \bigcup\{U \cdot V^\omega | U, V \in \mathfrak{W}_a, V \cdot V \subseteq V, U \cdot V^\omega \cap L = \emptyset\}$

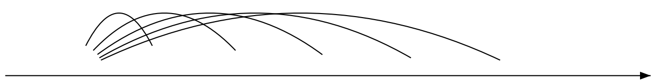
Proof (Proof of Lemma 2):
Given $\alpha \in \Sigma^\omega$ Notation $\alpha[m, n) = \alpha(m) \ldots \alpha(n-1)$
Two positions k, k' merge at $m(> k, k')$



$k \equiv_\alpha k'$: k, k' merge at some m **Remark 1** k, k' merge at $m, m < n \Rightarrow k, k'$ merge at n: clear from Remark on $\sim_a$ being congruence
Remark 2 $\equiv_\alpha$ is an equivalence relation over $\mathbb{N}$ of finite index.



Given α, by Remark 2, choose infinite $\equiv_\alpha$ -class, say with $k_0, k_1, k2_, \ldots$
Consider $\alpha = \alpha[0, k_0)\alpha[k_0, k_1)\alpha[k_1, k_2) \ldots$
Of the segments, $\alpha[k_0, k_i)$ infinitely many must belong to fixed $\sim_a$-class, say V
So choose a subsequence of the k_i, call them k_0, k_1, k_2 again, sucht that $\alpha[0, k_0) \in U$
$\alpha[k_0, k_i) \in V$ for all $i > 0$
By canncelling some k_j we can assume that k_0, k_i merge at $k_i + 1$ Call the subsequence obtained again $k_0, k_1, \ldots$ Show for this sequence $\alpha[k_i, k_{i+1} \in V, i = 0, \alpha[k_0, k_1) \in V$ So $\alpha[k_i, k_i + 1) \sim_\alpha$ Check



wether $U.V \subseteq V$ For construction of complement Büchi automaton we need a test wether $U \cdot V^\omega \cap L = \emptyset$

Lemma (Intersection-Lemma:)
Given Büchi automata a_1, a_2, $L(a_1) \cap L(a_2)$ is Büchi-recognizable.
Emptiness Test: Given Büchi aut. a, one can test wether $L(a) = \emptyset$
$L(a) \neq \emptyset \Leftrightarrow$ ex. final state q, such that

- q is reachable from q_0

- q is reachable from q by nonempty path.

Idea: Given $a_1(Q_1, \Sigma, q_{01}, \Delta_1, F_1)$
$a_2(Q_2, \Sigma, q_{02}, \Delta_2, F_2)$

construct product automaton
Introduce memory component with entries $0, 1, 2$:

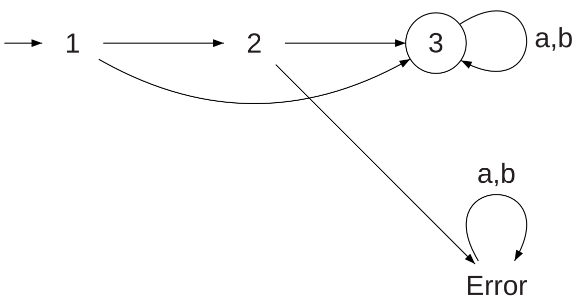
1. wait for state in F_1
2. wait for state in F_2
3. Cycle completed

1.6 Acceptance Conditions

Aim: Obtain expressive of NBA by deterministic automata with other acceptance then Büchi-ac.
Four basic acceptance conditions (given $a = (Q, \Sigma, q_0, \delta, F)$)

The run $\rho \in Q^\omega$ is E-Accepting if $\exists i, \rho(i) \in F$
A-Accepting if $\forall i \rho(i) \in F$
Büchi-Accepting if $\forall_j \exists i > j, \rho(i) \in F$
ω-Büchi-acc. $\exists, \forall i > j \rho(i) \in F$

An E-/A-/Büchi/ω-Büchi-cond is here a det. automaton used over ω-words with E-/A-… acceptance



Example:
$(aa + b) \cdot \Sigma^\omega$ $\Sigma = \{a, b\}$

Example:

For $\rho \in Q^\omega$ $Inf(\rho) = \{q \in Q | \forall j \exists i > j \ \ \rho(I) = q\}$ ρ is visited infinitely often in ρ
Muller aut. has format $a = (Q, \Sigma, q_0, \delta, F)$, where $F = \{F_1, \ldots, F_k\}$, $F_i \subseteq Q$

Run ρ is Muller accepting if $Inf(\rho) \in F$ $\exists i, Inf(\rho) = F_i$
L_1 set of ω-words over $\Sigma = \{a, b\}$ with infinitely many b
15.11.05 L_2set of ω-words over $\Sigma = \{a, b\}$ with only finitely many b. ...

Lemma
a) The class of Muller recognizable Languages is closed under boolean comb.

b) L is Büchi-recognizable $\Rightarrow$ L is Muller recognizable

$\mathcal{A}$

Proof:
We show closure under negation (complementation) an $\wedge$ (intersection)
Complementation proceed from $\mathcal{F} to \mathcal{F}' := q^Q \setminus \mathcal{F}$
Intersection use a product construction, given $\mathcal{A}_i = (Q_i, \Sigma, q_0^i, \delta_i, \mathcal{F}_i), (i = 1, 2)$ construct $\mathcal{A} = (Q_1 \times Q_2, \Sigma, (q_0^1, q_0^2), \delta, \mathcal{F})$ where $\delta((p, q), a) := (\delta_1(p, a), \delta_2(q, a))$ and $\mathcal{F}$ defined $\forall n, \forall p_1 \ldots p_n \in Q_1, \forall q_1 \ldots, q_n \in Q_2 : \{(p_1, q_1), \ldots, (p_n, q_n)\} \in \mathcal{F} \Leftrightarrow \{p_1 \ldots p_n\} \in \mathcal{F}_1$ and $\{q_1 \ldots q_n\} \in \mathcal{F}_2$

Lemma
a) $L \subseteq \Sigma^\omega$ det. Büchi recognizable $\Leftrightarrow$ $\Sigma^\omega \setminus L$ det ca.Büchi recognizable

b) $L \subseteq \Sigma^\omega$ E-recognizable $\Leftrightarrow$ $\Sigma^\omega \setminus L$ is A-recognizable

proof of a) (B is similar
Let $\mathcal{A} = (Q, \Sigma, q_0, \delta, F)$ be a det. Büchi automaton with $L = L(\mathcal{A})$
Define $\mathcal{A}' = (Q, \Sigma, q_0, \delta, Q \setminus F)$
Then $\alpha \in \Sigma^\omega \setminus L \Leftrightarrow Inf(\rho_\alpha \cap F = \emptyset \Leftrightarrow$ From some point onwards only states from $Q \setminus F$ are seen , i.e. $\forall i \geq n \rho_\alpha(i) \in Q \setminus F \Leftrightarrow \mathcal{A}'$ co-Büchi accepts α

Structural Analysis of E- and det. Büchi-recognizable Languages

Lemma
a) $L \subseteq \Sigma^\omega$ is E-recognizable $\Leftrightarrow$ $L = U\Sigma^\omega$ wher $U \subseteq \Sigma^*$ regular

b) $L \subseteq \Sigma^\omega$ is det. Büchi recognizable $\Leftrightarrow$ $L = lim(U), U \subseteq^*$ regular

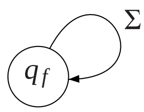
Definition
Let $u \subseteq \Sigma^*$
$Lim(u) := \{\alpha \in \Sigma^\omega | \alpha[0, \ldots, i] \in U$ for inifinitely many$i\}$

Example U = $a^* b a^*$
$lim(u) = \{\alpha \in \Sigma^\omega | \alpha$ contains exactly one b$\}$

Proof:
a) Let $\mathcal{A} = (Q, \sigma, q_0, \delta, F)$ be det. automaton
Let $L \subseteq \Sigma^\omega$ be the ω-language E-recognized by $\mathcal{A}$ and $U \subseteq \Sigma^*$ the language recognized by $\mathcal{A}$ as finite automaton. Then $\alpha \in \Sigma^\omega isE - Acceptedby\mathcal{A} \Leftrightarrow$ in the unique run of $\mathcal{A}$ on α. Finished after finite prefix $u \Leftrightarrow \mathcal{A}$ accepts u as a finite automaton $\Leftrightarrow u \in U$ and for the remainder of $\alpha \in \Sigma^\omega$.

b) Let $\mathcal{A} = (Q, \Sigma, q_0, \delta, F)$ det, $L = L(\mathcal{A}) \subseteq \Sigma^\omega$ Büchi recognized by $\mathcal{A}$, $U \subseteq \Sigma^*$ reg. language accepted by $\mathcal{A}$ as finite DFA.
Then for $\alpha \in \Sigma^\omega$: $\mathcal{A} acceotsα \Leftrightarrow in \rho_\alpha$ a state from F is visited inf. often i.e. $\rho_\alpha(i) = q \in F$ for inf. many $i \Leftrightarrow \alpha[0, \ldots, i]$ acceptesd by DFA $\mathcal{A}$ for inf. many $i \Leftrightarrow \alpha[0, \ldots, i] \in U$ for inf. many $i \Leftrightarrow \alpha \in lim(u)\square$

Comparison of det. recognizable ω-Languages **Hierarchy Theorem** For the classes of det. E,A,Büchi, co-Büchi, Muller-recognizable languages, the following inclusion diagram
All the inclusions are strict!
proof strategy: done. Büchi $\subseteq$ Muller complement Lemma + closure of Muller under boolean comb. $\Rightarrow$ co-Büchi $\subseteq$ Muller , we sho $E \subseteq$ Büchi, $E \subseteq$ co-Büchi.
Then (complement) $\Rightarrow A \subseteq$ Büchi, $A \subseteq$ co-Büchi.
Proof: $W \subseteq$ Büchi, $E \subseteq$ co-Büchi
Let $\mathcal{A}$ be an E-automaton recognizing $L \subseteq \Sigma^\omega$ We construct automaton $\mathcal{A}'$ wich both Bühi and co-Büchi



recognizes L. $\mathcal{A}'$ results from $\mathcal{A}$ by adding a new accepting sink state, , $\{q_f\} = F'$
22.11.05 redirtect every transition to a final state in $\mathcal{A}$ to q_f, i.e. $\delta(q, a) = p \in F \Rightarrow \delta'(q, a) = q_f$

1.6.1 Strict inclusion claims

On (4): $O^*1(0+1)^\omega$ is not A-recognizable.
Assume $\mathfrak{A}$ with n states recognizes $0^*1(0+1)^\omega$
$\mathfrak{A}$ on input 0^n10^ω accepts, so run has only final states.
$\mathfrak{A} : q_0 \underbrace{\to p \to p \to}_{0^n} q$ On input o^ω the run repeats $q_0 \to p \to p \to p \ldots$ so only final states. $\mathfrak{A}$ accepts 0^ω contradiction.
On (5): 0^ω, complement of $0^*1(0+1)^\omega$
Assuming $\{=^\omega\}$ is E-recogn., the complement would be A-recognizable. Now use the proof on the language (4).
So $\{0^\omega\}$ is not E-recogn.
On(2): $L_2 = \{\alpha \in \{0,1\}^\omega |$ in $\alpha 101$ occurs, but 11 does not $\}$
Assume L_2 is E-Recognizable, say by $\mathfrak{A}$ with n states. Consider $\mathfrak{A}$ on 1010^ω, reaches a final state, say up to 1010^k. On 1010^k1^ω. $\mathfrak{A}$ reaches final state and accepts. Contradiction.
Assume L_2 is A-recognizable, say by $\mathfrak{A}$ with n states. $\mathfrak{A}$ accepts 0^n1010^ω, have state repetitions on prefix 0^n, so accept 0^ω, so contrad.
On (7): $L_7 = \{\alpha \in \{0,1\}^\omega | 1$ occurs only finitely often$\}$ known: not Büchi-recogn.
On (6): L_6 is complement of L_7, so it is not co-Büchi recognizable.

1.6.2 Deciding the levels

Aim: Given Mulller automaton $\mathfrak{A} = (Qm\Sigma, q_0, \delta, \mathcal{F})$ we want to decide wether $L(\mathfrak{A})$ is in fact E-recognizable or Büchi-recognizable.

A loop of $\mathfrak{A}$ is a subset $S \subseteq Q$ s.t. for all s, s' exists $w \in \Sigma^+$ with $\delta(s, w) = s'$ over $Q \cap S$
Remark A set $Inf(\rho)$ is a loop. We may restrict $\mathcal{F}$ to loops only. Assume $\mathfrak{A}$ has only reacheable states, has acceptance component $\mathcal{F}$ containing only loops.
$\mathcal{F}$ is closed under reacheable then $S' \in \mathcal{F}$
$\mathcal{F}$ is closed under superloops $\Leftrightarrow$ if loop $S \in \mathcal{F}$ and loop $S' \supseteq S$ then $S' \in \mathcal{F}$
Remark Given $\mathfrak{A}$ both properties of $\mathcal{F}$ can be checked effectively.

1.6.3 Landweber's Theorem

Let $\mathfrak{A}$ be a Muller automaton $\mathfrak{A} = (Q, \Sigma, q_0, \delta, \mathcal{F})$

a) $L(\mathfrak{A})$ is E-recognizable if $\mathcal{F}$ is closed under reachable loops.

b) $L(\mathfrak{A})$ is Büchi-recognizable if $\mathcal{F}$ is closed under superloops.

Proof b) $\Leftarrow$ Assume $\mathfrak{A}$ is closed under superloops. Construct Büchi automaton for $L(\mathfrak{A})$ Use set $Q \times 2^Q$: in first component simulate $\mathfrak{A}$ in second component accumulate visited states until superset $S \supseteq S \in \mathcal{F}$ is reached, then go to $\emptyset$ instead (final).
Automaton accepts if given aut. $\mathfrak{A}$ $satisfies Inf(\rho) \supseteq S \in \mathcal{F} \Rightarrow$ Given Muller-automaton $\mathfrak{A} = (Q, \Sigma, q_0, \delta_a, \mathcal{F})$ and a Büchi-atuomaton $\mathfrak{B} = (P, \Sigma, p_0, \delta_b, \mathcal{F}$ with $L(\mathfrak{A}) = L(\mathfrak{B})$
Consider loop $S \in \mathcal{F}$, superloop S'. Show $S \in \mathcal{F}$
Find ω-word $\alpha$with $Inf(\rho) = S'$ for the $\mathfrak{A}$-run ρ, with $\mathfrak{A}$ accepts. Start α with prefix w leading $\mathfrak{A}$ to some $q \in S$. Continue w by γ wich causes $\mathfrak{A}$ to loop through S again and again. $\mathfrak{B}$ on $w\gamma$ visits F-state after w, say adter wu_1. Via word v_1 back to q in $\mathfrak{A}$, via x_1 go once through S in $\mathfrak{A}$ and back to q. On prefix $wu_1v_1x_1$ $\mathfrak{A}$ has looped through S' once, $\mathfrak{B}$ has visited a final state. Repeat the argument with $wu_1v_1x_1\gamma$. Repeateing we obtain $wu_1v_1x_1u_2v_2x_2\ldots$ s.t.: $\begin{cases} \mathfrak{A} loops\ through\ S' again and again \\ \mathfrak{B}\text{ reaches final states inf. often} \end{cases}$ So $\mathfrak{B}$ accepts hence accepts, hence $S' \in \mathcal{F}$

1.7 Weak automata

A Staiger-Wagner automaton (weak Muller automaton) has the same format $(Q, \Sigma, q_0, \delta, \mathcal{F})$ as Muller automaton, but with the following acceptance: $\mathfrak{A}$ accepts α if for unique run ρ of $\mathfrak{A}$ on α: the set of states occuring in ρ is in $\mathcal{F}$. $Occ(\rho) \in \mathcal{F}$ 29.11.200

Staiger-Wagner automaton: $\mathfrak{A} = (Q, \Sigma, q_0, \delta, \mathcal{F})$ $\mathcal{F} \subseteq Pow(Q)$
$\mathfrak{A}$ accepts $\alpha \Leftrightarrow$ for the unique run ρ of $\mathfrak{A}$ in α we have $Occ(\rho) \in \mathcal{F}$ for some $F \in \mathcal{F}$, the states of ρ form F

Theorem 3 *$L \subseteq \sigma^\omega$ is (det.) Büchi- and Co-Büchi recognizable $\Leftrightarrow$ L is Staiger-Wagner recognizable.*

Proof:

$\Leftarrow$ Given $\mathfrak{A} = (Q, \Sigma, q_0, \delta, \mathcal{F})$ $\mathcal{F} = \{F_1, F_2, \ldots, F_k\}$
Construct $\mathfrak{A}'$ over $Q \times \underbrace{2^Q \times \cdots \times 2^Q}_{k\ times}$
State $(q, R_1, \ldots, R_k)$ signals that $\mathfrak{A}$ is in q, and that $R_i (i = 1, \ldots, k)$ is the set of states visited so

far.
Declare $(q, R_1, \ldots, R_k)$ as final if $R_i = F_i$ for some i (For Büchi-automaton $\mathfrak{A}$)

$\mathfrak{A}'$ visits final inifinitely often $\Leftrightarrow$ for some $i, R_i = F_i$ infinitely often
$\Leftrightarrow$ for some $i, R_i = F_i$ from some point onwards.

So $\mathfrak{A}'$ used as büchi or as Co-Büchi automaton, accepts if the visited states for $\mathfrak{A}$ form some F_i.

Preparation for $\Rightarrow$: SCC Decomposition) Given transition graph, a strongly connected component is a maximal strongly connected subset.
Proposition the SCCs and the singletons wich do not belong to an SCC form a partial ordering under the reachability relation.

SCC-Algorithm For directed Graph $G = (V, E)$

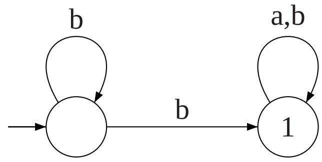
1. Run depth-first search, recording enter/farewell-times for the vertices
2. Reverse edges, get G^T
3. Run depth-first search on G^T, taking as roots of depth-first trees vertices in reversed order of finish times (Starting from vertex with highest farewell

Resulting d-f-trees are the SCCs (the reacheable vertices form a SCC S of G

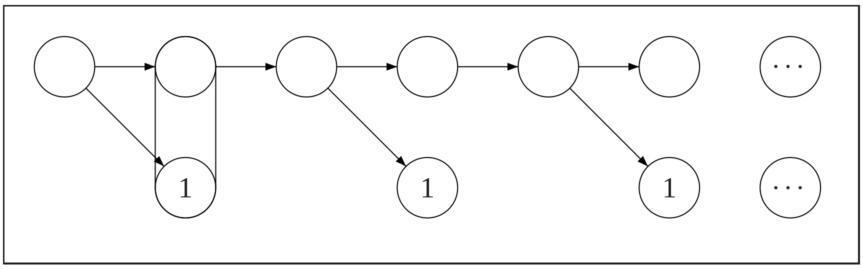
$\Rightarrow$ Given Büchi-automaton wich recognizes L
Take Muller-automaton for L, $\mathfrak{A} = (Q, \Sigma, q_0, \delta, \mathcal{F})$ $\mathcal{F}$ is closed under superloops.
Since $\mathfrak{A}$ recognizes a co-Büchi recogn. set, $\mathcal{F}$ is closed under subloops.
Consequence: All loops of an SCC of $\mathfrak{A}$ are accepting ($\in \mathcal{F}$) or all loops of SCC are rejecting ($\notin \mathcal{F}$).
Call SCC S good, if all its loops are accepting, (otherwise it's bad)
Fiven S, let S_+ be the set of states $q \notin S$ with transition (p, q), $p \in S$
Consequence: Run ρ of $\mathfrak{A}$ is accepting if ρ reaches some good S but does not reach the correpsonding set S_+.
So get a Staiger-Wagner-automaton from $\mathfrak{A}$ with the following acceptance component $\mathcal{F}'$ containing a set $R \subseteq Q$ if for some good S we have $R \cap S \neq \emptyset$ and $R \cap S_+ = \emptyset$.

2 Determinization

Aim: Transformation of undet. Büchi automaton to det. Muller automata. (McNaughton 1966, (Information and Control)).
Safra 1988: Optimal complexity bound for the number of states (Rabin automata)
Muller,Schupp (1992): Optimal complexity bound for the number of states



Problem: Powerset construction is not enough.



infinitely often set visited with final state 1!
Idea of MS-construction: On given input word, build up the "run Tree" of Büchi automaton. Use prefixes of tree up to some level as first approximation of states. Reduction and compression leads to finite number of states.

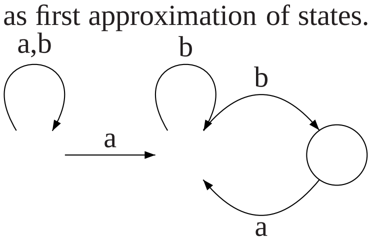


Illustration with

$L = (a + b)^*(b^+a)$
Example input: Run tree of $\mathfrak{A} = (Q, \Sigma, q_0, \Delta, F)$ on input

Remark:
$\mathfrak{A}$ accepts $\alpha \Leftrightarrow$ in run tree of $\mathfrak{A}$ an infinite path exists with infinitely many final states.

Reduct1on 1 Put states together if they are final, respectingly nonfinal (final:"down", vertically display: "left")
Result: Binary branching tree. "Acceptance Tree".

Remark:
$\mathfrak{A}$ accepts α $\Leftrightarrow$ in acceptance of $\mathfrak{A}$ on α exists path pranching down infinitely often.

From a nondet. Büchi automaton, one can construct an equivalent det. Muller automaton.
Given $\mathfrak{A} = (Q, \Sigma, q_0, \Delta, F)$, start on input α from run tree of $\mathfrak{A}$ on α
Convention: Branch left(down) with final states.

1. Reduction Merge states at a branching when they are final (left succ.)
Merge states at a branching when they are non-final (right succ.)
Get "acceptance tree" with a most binary branching.

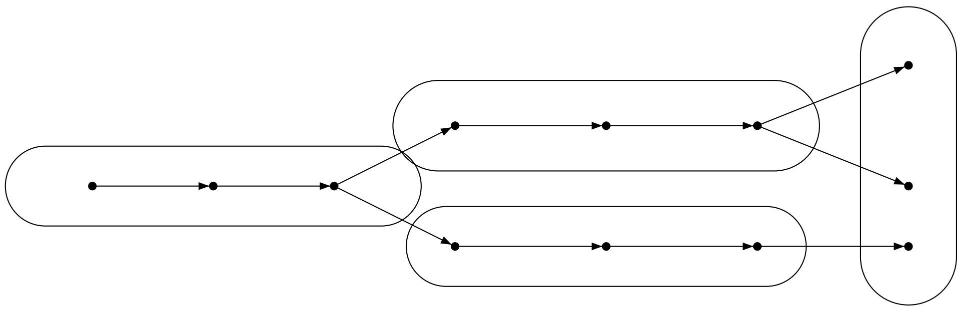
Remark:
$\mathfrak{A}$ accepts $\alpha \Leftrightarrow$ in acceptance tree of $\mathfrak{A}$ on α exists path with infinitley many left turns.

$\Rightarrow$ easy from condition on run tree
$\Leftarrow$ from infinite path in acc. tree obtain a partial run tree wich is infinite and finitely branching.Königs Lemma gives infinite path of run tree of course with infinitley many left turns.

2. Reduction On each Level keep only the leftmost (downmost) occurence of each individual state.

Remark:
$\mathfrak{A}$ accepts $\alpha \Leftrightarrow$ in the resulted left-reduced acc. tree a path exists with infinitely many left turns.

3. Reduction Compress path segments into single nodes:



Merge nodes of a path segment into the topmost one (not a successor of branching node)
Keep states at leaves, color each node of compressed tree by:

- Red: if no final state occurs
- yellow: if final state occurs, no final state added
- green: if final state was added in left update

4. Reduction Delete all nodes wich do not get a new descendent in the last update step.

Result: Muller-Schupp tree (over Q), a finite, strictly binary tree with node names from $\mathbb{N}_+$ where node is colored red,green or yellow, and the leaves are labelled with disjoint state sets (over Q).
Notation: $MS(Q)$ for the finite set of all Muller-Schupp trees over Q.

Remark:
$\mathfrak{A}$ accepts $\alpha \Leftrightarrow$ in the sequence of Muller-Schupp trees of $\mathfrak{A}$ on α, some node stays forever from some point onwards and is colored gren again and again.

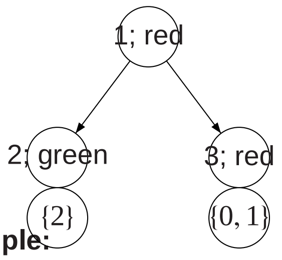
Definition
of Muller automaton $\mathfrak{A}'$ from nondet. Büchi automaton $\mathfrak{A} = (Q, \Sigma, q_0, \Delta, F)$

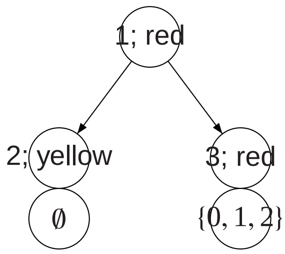
State-set: $MS(Q)$ initial state: $\overset{1;\text{red}}{\{q_0\}}$

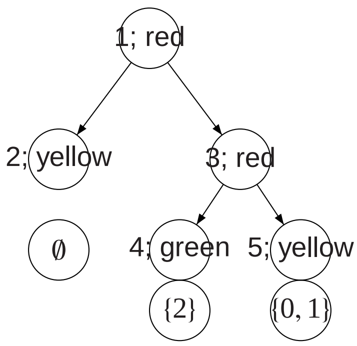
Definition
of $\delta(t,a)$ ($t : MS - tree$) using the following update rule:

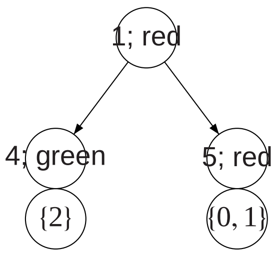
1. Copy tree, replace green by yellow
2. For each leaf, with state set P, introduce son labelled $P' = \{q | \exists p \in P, (p, a, q) \in \Delta\}$
   Delete state p if it occurs more to the left.
   Split any set into left,right son with the final, resp. nonfinal states with colors green,red.
3. Delete all nodes wich did not get a new descnendant with $\neq \emptyset$ set.
4. Compress path semgents into the respective top node giving it colour green if it is merged with nodes either coloured green or yellow.

Convention about use of node names after deletion step a node name can be reused, however not in an immediate successor tree accordings to update.
Observation: Over Q, $3|Q|$ node names suffices.
Acceptance Component: Define $\mathfrak{F}$ as follows:
$R(\subseteq MS(Q)) \in \mathfrak{F} \Leftrightarrow$ some node name k occurs in each tree of R and even colored green in some tree of R.

Example:
t = 

$\delta(t, b)$ to be computed. 





Definition
Define $E_k :=$ set of MS-trees without node k
$F_k :=$ set of MS-trees with node k colored green.
Acceptance Condition: for some k: any tree in E_k occurs only finitely often, some tree in F_k occurs

infinitely often.
Start notation for run ρ (of MS-trees): $\bigwedge_{k=1}^{m} Inf(\rho) \cap E_k = \emptyset \wedge Inf(\rho) \cap F_k \neq \emptyset$

Definition
A (det.) Rabin-automaton has the form $\mathfrak{A} = (Q, \Sigma, q_0, \delta, \Omega)$ where Ω is sequence $(E_1, F_1), \ldots, (E_m, F_m)$ of sets $\subseteq Q$.
$\mathfrak{A}$-run ρ is accepting if for some $k \in \{1, \ldots, m\}$: $Inf(\rho) \cap E_k = \emptyset \wedge Inf(\rho) \cap F_k \neq \emptyset$

13.12.05

Theorem 4 *A nondet. Büchi aut. can be transformed into a deterministic Muller automaton and also into a det. Rabin automaton.*
Rabin aut.: $\mathfrak{A} = (Q, \Sigma, q_0, \delta, \Omega)$ $\Omega = (E_1, F_1), \ldots, (E_m, F_m))$ $E_i, F_i \subseteq Q$
ρ *successfull* $\Leftrightarrow \bigvee_{i=1}^{m}(Inf(\rho) \cap E_i = \emptyset \wedge Inf(\rho) \cap F_i \neq \emptyset)$

Remark on Rabin and (Union Lemma) Given Rabin aut. over Q, whith $\Omega = (E_1, F_1), \ldots, (E_m, F_m))$, ρ_1, ρ_2 non-successful runs
Let ρ be run with $Inf(\rho) = Inf(\rho_1) \cup Inf(\rho_2)$
ρ is not successful

Proof:
$\rho_1, \rho_2 arenotsuccessful$, assume ρ is successful, $Inf(\rho) = Inf(\rho_1) \cup Inf(\rho_2)$
Pick index i: $Inf(\rho) \cap E_i = \emptyset \wedge Inf(\rho) \cap F_i \neq \emptyset$
Then $Inf(\rho) \cap E_i = \emptyset, Inf(\rho_2) \cap E_i = \emptyset$
Also $Inf(\rho_1) \cap F_i \neq \emptyset$ or $Inf(\rho_2) \cap F_i \neq \emptyset$
So ρ_1 or ρ_2 successful

Theorem 5 *MS-construction yelds a Rabin automaton with* $2^{O(nlog\ n)}$ *states from Büchi automaton with* n *states*

Proof:
Estimate number of MS-Trees over Q, $|Q| = n$
MS-trees are built from node names $1, \ldots, 3n$
Fix a MS-Tree by the following functions:

parent $p : N \to N \cup \{0, *\}$

$$p(n) = \begin{cases} parent & \text{if exists} \\ 0 & \text{i k is root} \\ * & otherwise \end{cases}$$

right brother $rb : N \to N \cup \{0, *\}$ anologously
color: $c : N \to$ {green, red,yellow} $\cup \{*\}$
State occurence: $\sigma : Q \to N \cup \{0\}$

$$\begin{cases} \text{node where } q \text{ occurs} & \text{if q occurs} \\ 0 & otherwise \end{cases}$$

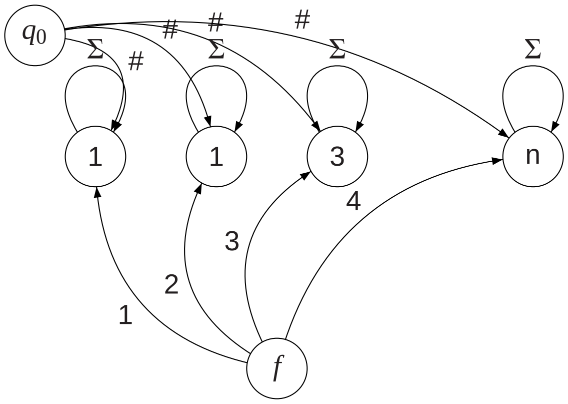
Number od MS-Trees $\leq$ number of quadruples (p, rb, c, σ) of functions.

$$\leq (3n+2)^{3n} \cdot (3n+2)^{3n} \cdot 4^{3n} \cdot (3n+1)^{n} \leq (4n)^{10n} = 2^{O(n\ log\ n)}$$

Optimality of bound:

Theorem 6 *There is $L_n \subseteq \{\#, 1, \ldots, n\}^\omega$ recognized by Büchi aut. with $n+2$ states auch that any det. Rabin automaton recognizing L_n needs $\geq n!$ states. $n! \in 2^{=(n\ log\ n)}$*

Proof:
Büchi automaton for L_n $\Sigma = \{\#, 1, \ldots, n\}$

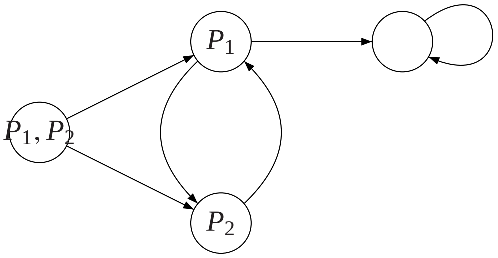


Cycle property: $\alpha \in L_n \Leftrightarrow$ exists letters $i_1, \ldots, i_k \in \Sigma \setminus \{\#\}$ such that the letter paris segments $i_1 i_2, i_2 i_3, i_3 i_4 \ldots i_{k-1} i_k, i_k i_1$ occur infinitely often.

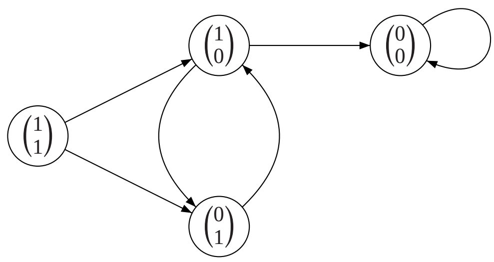
Consider permutation $(i_1, \ldots, i_n)$ of $1, \ldots, n$ $(i_1, i_2, \ldots, i_n\#)^\omega \notin L_n$
Assume $\mathfrak{A}$ does not accept $(i_1, \ldots, i_n\#)^\omega, (j_1, \ldots, j_n\#)^\omega$ with permutations $i_1, \ldots, i_n$ $j_1, \ldots, j_n$
The runs ρ_α, ρ_β of $\mathfrak{A}$ on α, resp β are not successful, $Inf(\rho_\alpha) = R$ Show; $Inf(\rho_\beta) = S$ $R \cap S = \emptyset$
So $\mathfrak{A}$ has $\geq n!$ states.
Assume $q \in R \cap S$. Build ω-word with infinitely many occ. of $i_1 \ldots i_n, j_1 \ldots j_n$
$Inf(\rho) = R \cup S$, ρ not successful.
$i_1 \ldots i_k$
$j_1 \ldots j_k$
get cycle in input word. Contradiction!

3 Monadic Theory of one successor (S1S)

We consider transition systems.



p_i denote properties of the states
arrows = possible behaviour of the system.
Associate boolean vector to properties



p_i is true $\Leftrightarrow$ i-th component is 1
Execution of such a system yields an ω-word over $\mathbb{B}^m$
eg: $\alpha = \binom{1}{1}\binom{1}{0}\binom{0}{1}\binom{1}{0}\binom{0}{0}^{\omega}$
evolvement of single property over time is the projection to the corresponding row
Express specifiication for the behaviour of the system by expressing specification for ω-words over $\mathbb{B}^n$
Use S1S for this: variables $s_it\dots$ for time points, positions
variables $X, Y, Z\dots$ sets of positions, 0 constant, successor $'$ < earlier, = , + boolean connectors + quantification

Example:
Constant: At position 3, p_1 holds. $\phi(X_1) = X_1(\overbrace{\underbrace{0'''}_{=3}})$
Reactivity: Sometimes p_1 holds. $\phi_2(X_1) = \exists t X_1(t)$
Recurrence: again and again p_1 holds: $\forall t \exists s > t : X_1(s)$
Request - Response: Whenever p_1 holds, p_2 holds afterwards.
$\forall s(X_1(s) \rightarrow \exists t(t > s \wedge X_2(t))$

3.1 Formal Syntax

Variables: $s, t, \ldots$
Second-order variables: $X, Y, X_1, X_2 \ldots$
Terms: constant 0, first-order variables, τ term $\rightarrow$ τ' term
Atomic formulas: $X(\tau), \sigma < \tau, \sigma = \tau$ with σ, τ terms.
S1S-formulas are obtained from the atomic formulas by using boolean connectives and quantification.

3.2 Semantics

Use $\mathbb{N}$ as universe for first-order variables
Use $2^{\mathbb{N}}$ as universe for second-order variables
The interpretation of $'$ is $+1$
<= less than on $\mathbb{N}$
Use standard semantics

Write
$(\mathbb{N}, 0, +1, <, P_1, \ldots, P_n) \models \varphi(X_1 \ldots X_n)$ where $X_1 \ldots X_n$ are the free variables od φ
if φ is true in these semantics if the free variable X_i is interpreted as P_i
We need to specify only $P_1 \ldots P_n = \overline{P}$
For $P_1 \ldots P_n \subseteq \mathbb{N}$ we define $\alpha(\overline{P})(\alpha \in (\mathbb{B}^n)^\omega$ by $(\alpha(i)9_j = 1$ iff $i \in P_j$
Then we write $\alpha(\overline{P}) \models \varphi(X_1 \ldots X_n$

Definition
For S1S-formula $\varphi(X_1 \ldots X_n)$ define $L(\varphi) = \{\alpha \in ((B)^{n^\omega} | \alpha \models \varphi(X_1 \ldots X_n)\}$

20.12.05

3.2.1 Connection from S1S to Büchi-automata

S1S: $s, t, \ldots$ positions of ω-words.
$X, Y, \ldots$sets of positions
$0,', <X(s)$ “Monadic second-order logic”
$X(s''')$
$\neg, \wedge, \vee, \rightarrow, \leftrightarrow, \exists, \forall$

Formula $\varphi(X_1, \ldots, X_n)$ satisfied in a model $(\mathbb{N}, 0,', <, P_1, \ldots, P_n) \approx \omega$-word over $\{0, 1\}^\omega$

Example (for correspondance $(p_1, P_2) \sim \alpha\{0, 1\}^2$):

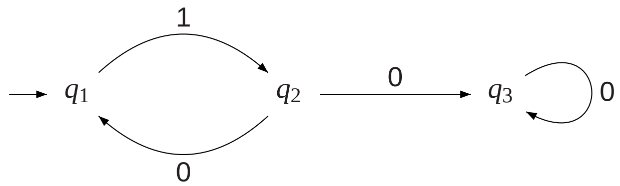
P_1 even numbers:	1	0	1	0	1	0	1	0	1	0 ...
P_2 prime numbers:	0	0	1	0	1	0	1	0 ...		

$\varphi(X_1, X_2$ “There are two successive positions with 1 in second component followed by 1 in first component”

With α from above: $\alpha \vDash \exists s \exists t(s' = t \wedge X_2(s) \wedge X_2(t) \wedge X_1(t'))$
$\exists s(X_2(s) \wedge X_2(s') \wedge X_1(s''))$

$L \subseteq \{0,1\}^n$ S1S-definable $\Leftrightarrow$ exists S1S-formula $\varphi(X_1, \dots, X_n)$ s, t for any $\alpha \in (\{0,1\}^n)^\omega$: $\alpha \in L \Leftrightarrow \alpha \vDash \varphi(X_1, \dots, X_n)$

Theorem 7 *(Büchi 1960) An ω-language $L \subseteq (\{0,1\}^n)^\omega$ is S1S-definable if and only if is Büchi-recognizable.*
Proof:
[$\Leftarrow$]: Given Büchi-automaton

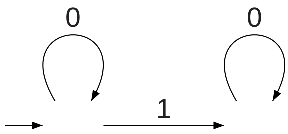


$\Sigma = \{0,1\}$ Find $\varphi(X)$ saying "$\mathfrak{A}$ accepts the ω-word corresponding to X"
$\varphi(X)$ has to say: exists a successful run of $\mathfrak{A}$ over α corresponding to X

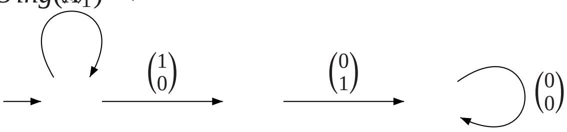
α	=	1	0	1	0	0	0	0
	q_1	*		*				
	q_2		*		*			
	q_3					*	*	0

Idea: Express existence of run by existence of three set Y_1, Y_2, Y_3
Express that Y_1, Y_2, Y_3 represents successful run
$\varphi(X) : \exists Y_1, \exists Y_2 \exists Y_3($ each position belongs to singly $Y_i \wedge Y_1(0) \wedge \forall s(Y_1(s) \wedge X(s) \wedge Y_2(s')) \vee (Y_2(s) \wedge \neg X(s) \wedge Y_1(s')) \vee (Y_2(s) \wedge \neg X(s) \wedge Y_3(s'')) \vee (Y_3(s) \wedge \neg X(s) \wedge Y_3(s'))]$
$\wedge \forall s \exists t(s < t \wedge Y_3(t)))$
General Case $\mathfrak{A} = (Q, \{0,1\}^n, q_1, \Delta, F)$ $Q = \{q_1, \dots, q_m\}$
$\varphi(X_1, \dots, X_n) : \exists Y_1 \dots Y_m(Partition(Y_1, \dots Y_m) \wedge Y_1(0) \wedge \forall s \bigvee_{(q_i, a, q_j \in \Delta}(Y_i(s) \wedge X_a(s) \wedge Y_j(s)) \wedge \forall s \exists t(s < t \wedge \bigvee_{q_i \in F} Y_i(t))$
Partition $(Y_1, \dots, Y_m) : \forall \bigvee_{i=1}^m Y_i(s) \wedge \neg \exists s \bigvee_{i \neq j}(Y_i(s) \wedge Y_j(s))$
For $a = (b_1, \dots, b_n)$ $b_1 \in \{0,1\}$ write $X_a(s)$ for $(b_1)X_1(s) \wedge \dots \wedge (b_n)X_n(s)$ where $b_i = \begin{cases} empty & b_i = 1 \\ \neg & b_i = 0 \end{cases}$
[$\Rightarrow$] From S1S-Formulas to büchi-automata

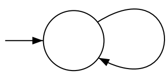
Simplify formalism S1S to S1S$_0$ with second order variables only.
S1S$_0$ has new atomic formulas :
$Sing(X)$ for "X is a singleton"
$Succ(X, Y)$ for $X = \{s\}, Y = \{t\}$ with $s' = t$
$X \subseteq Y$
Lemma S1S formulas can be rewritten as S1S$_0$-formulas
Proof Apply the following steps: Eliminate 0: $X(=) \rightsquigarrow \exists s(X(s) \wedge \exists tt' = s)$
Eliminate iterations of $'$ $X(s'') \rightsquigarrow \exists t(s' = t \wedge X(t'))$
Eliminate $<$: $s < t \rightsquigarrow$ "t is in successor closure of s' " $\forall X(X(s') \wedge \forall z(X(z) \rightarrow X(z')) \rightarrow X(t))$
Get S1S-formulas with atomic formulas $s' = t$ $X(s)$ only
From such formulas obtain an equivalent S1S$_0$ formula.
Example: $\exists X \forall s \exists t(s = t', X(s))$
$\exists X \forall S(Sing(S) \rightarrow \exists T(Sing(T) \wedge Succ(T, S) \wedge S \subseteq X)$ **Lemma** Each S1S$_0$ formula $\varphi(X_1, \dots, X_n) canbe$ transformed into an equivalent Büchi-automaton.
Proof by induction on S!S$_0$-formulas.



Atomic formulas $Sing(X_1) \rightsquigarrow$



$Succ(X_1, X_2) \rightsquigarrow$



$X_1 \subseteq X_2 \rightsquigarrow$

For induction step assume that only connectives $\neg \vee \exists$ remain ($\wedge, \rightarrow, \leftrightarrow$ have been eliminated)

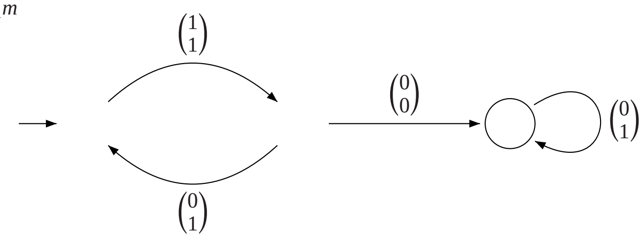
$\neg$ Consider $\neg\psi(X_1, \dots X_m)$, assume by ind Büchi automaton $\mathfrak{A}$ for ψ

Use Büchi-aut. complementation to find automaton for $\neg\psi$

$\vee$: $\psi_1(X_1 \dots) \vee \psi_2(X_1 \dots)$, assuming Büchi aut. $\mathfrak{A}_{\psi_1}, \mathfrak{A}_{\psi_2}$. Use union automation of $\mathfrak{A}_{\psi_1}, \mathfrak{A}_{\psi_2}$

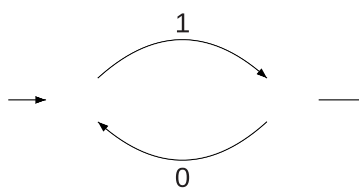
$\exists$ Consider $\exists X\psi(X_1, \dots, X_m, X)$ assuming Büchi-aut. $\mathfrak{A}_\psi$ over $\{0,1\}^{m+1}$

Find autom. over $\{0,1\}^m$



Example: $\psi(X_1, X)$

New automaton reads only first component ans messes second comp. with this simulating given automaton.



Implementation: Delete second components in the given automaton:

10.1.06

From S1S-formulas to Büchi-automaton $\varphi(X_1, \dots, X_n) \mapsto \mathfrak{A}_\varphi$ over $\Sigma = \{0,1\}^n$ such that for each $\alpha \in (\{0,1\}^n)^\omega$

$\alpha \models \varphi(X_1, \dots, X_n) \Leftrightarrow \mathfrak{A}_\varphi$ accepts α

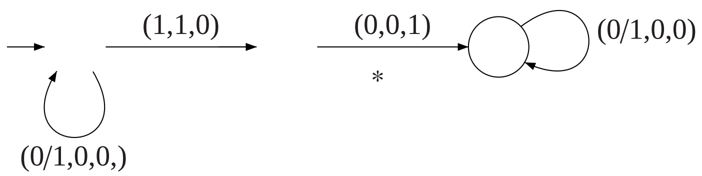
Illustration $\varphi(X_1) : \exists s(X_1(s) \wedge \neg X_1(s'))$

First Step: Rewriting as $S1S_0$-Formula:

$\exists s \exists t(X_1(s) \wedge s' = t \wedge \neg X_1(t))$

$\exists X_2 \exists X_3 \underbrace{(X_2 \subseteq X_1 \wedge Succ(X_2, X_3) \wedge \neg X_3 \subseteq X_1)}_{\psi(X_1, X_2, X_3)}$

$X_2 = \{s\}$, $X_3 = \{t\}$
Automaton for $X_2 \subseteq X_1 \land Succ(X_2, X_3)$



For intersection with $\neg X_3 \subseteq X_1$ take only (0,0,1) at $(*)$
Automaton for full formula: forget 2^{nd} / 3^{rd} component.

1st Step **Theorem 8** *For each S1S-formula $\varphi(X_1, \ldots X_n)$ one can construct an equivalent Büchi-automaton over $\Sigma = \{0,1\}^n$, and conversely*

Recall: Given Büchi automaton, an equivalent formula can be written as $\exists Y_1 \ldots \exists Y_m \underbrace{\varphi(X_1, \ldots, X_m, Y_1, \ldots Y_m)}_{first\ Order}$

$(*)$
Consequence 1: Call S1S-formula existential if it has form $(*)$
Each S1S-formula $\varphi(X_1 \ldots X_n)$ is equivalent to an existential one.
Proof by translation: $\varphi \mapsto \mathfrak{A}_\varphi \mapsto$ formula equivalent to $\mathfrak{A}_\varphi$ (of the form $(*)$) existential.
Consequence 2: Decidability of aritmetical theorics
Use Theorem for $n = 0$, i.e. for sentences φ (without free variables).
$\mathfrak{A}_\varphi$ has unlabelled transitions.
$(\mathbb{N},', <, 0) \models \varphi \Leftrightarrow \mathfrak{A}_\varphi$ has a successful run (state sequence with infenitely many visits of final state)
Hence: For each S1S-sentence one can decide whether it is true in $(\mathbb{N},' m, <, 0)$

Example:
$\forall s \exists t t < s$ false (take s=0)
$\forall X\,(X(0) \land \forall s(X(s) \to X(s')) \to \forall t X(t))$ (induction princ.) True
"The monadic second-order theory of $(\mathbb{N},', <, 0)$ is decidable"
Background: Gödel's result on "undecidability" of first-order arithmetic (for the structure $(\mathbb{N}, +, \cdot, 0, 1, <)$

Example:
$\forall x \exists (x < y \land \underbrace{\forall z_1 \forall z (z_1 \cdot z_2 = y \to z_1 = 1 \lor z_2 = 1)}_{y\ is\ prime}$
There are infinitely many primes.
$\forall x \exists y (x < y \land y\ is\ prime \land y + 1 + 1\ is\ prime$ "There are inifinitely many twin primes."

Remark:
Remak (Gödel): The full second-order theory (with quantification over relations) of $(\mathbb{N},', <, 0)$ is undecidable.

Proof:
By second-order definitions of + ans $'$
$x + y = z \Leftrightarrow$ each relation wich contains $(0, x)$ and is closed under successor in both components mus contain (y, z)
$\Leftrightarrow \forall R((0, x) \in R \land \forall (s, t)((s, t) \in R \to (s', t') \in R) \to (y, z) \in R)$
$x \cdot y = z$ analogously, using +

Decidability? (Tarstei) What about "monadic quantification"? (Quantifiers over sets only) Solution by Büchi.
$Th(\mathbb{N}, +, <, 0, 1)$ $Th(\mathbb{N}, \cdot, <, 0, 1)$ decidable (Presburger, Skolem)
Consequence 3 Monadic theory of a structure $(\mathbb{N}, ', <, 0, P)$ with some fixed $P \subseteq \mathbb{N}$

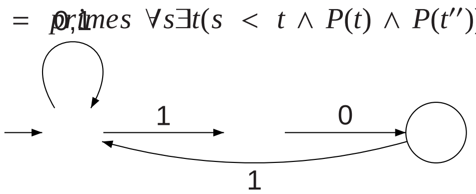
Example:
P= set of primes
$\forall s \exists t (s < t \wedge P(t) \wedge P(t''))$ twin prime statement.

Question: For wich P is the monadic theory decidable. Approach: Use Büchi's theorem about $\varphi(X) \mapsto \mathfrak{A}_\varphi$ for a fixed set/sequence P.

Given φ, find Büchi $\mathfrak{A}_\varphi$ such that $(\mathbb{N}, ', <, 0, P) \models \varphi \Leftrightarrow \mathfrak{A}_\varphi$ accepts α_p $\alpha_p(i) = \begin{cases} 1 & i \in P \\ 0 & i \notin P \end{cases}$

P primes: 001101010001
dots

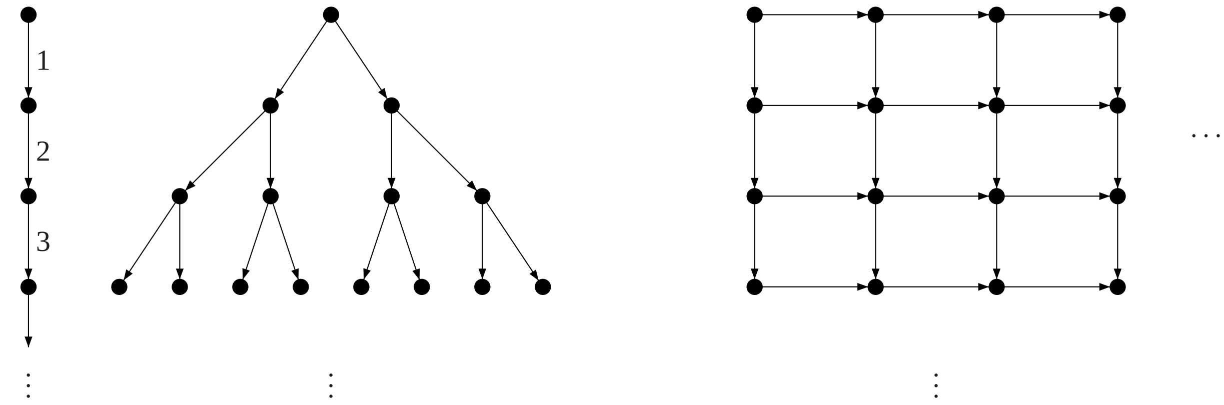
Example:
For $P = primes$ $\forall s \exists t (s < t \wedge P(t) \wedge P(t''))$ is true $\Leftrightarrow$ the following Büchi automaton accepts α_P



$MTh(\mathbb{N}, ', <, 0, P)$ is decidable if the following decision problem is decidable: Given Büchi-automaton over $\Sigma = \{0, 1\}$ Does $\mathfrak{A}$ accept α_P

Consequence 4: Method for model-checking.
Basic situation: P :Program (System) given as transition graph.
Here:represented as a (Büchi) automaton (with all states final)
Specification: S :Formula about the desired system runs.
Here: S1S formula about the transition labels.
P is corrext with respect to S: all runs wich are possible in P satisfy S
$L(\mathfrak{A}_p) \subseteq L(\mathfrak{A}_S)$, equivalently $L(\mathfrak{A}_p \cap L(\mathfrak{A})_S = \emptyset$ 17.01.06

3.3 The binary Tree and the two-dimensional Grid



Format of binary tree: $(\{0,1\}^*, succ_0, succ_1, \varepsilon)$ $succ_0(w) = w0$ and $succ_1(w) = w1$
Introduce monadic second order language as before with $succ_0, succ_1$ instead of $'$.
$S2S$ is the corresponding logical system.

Theorem 9 *(Rabin, 1969)* $MTh(\{0,1\}^*, succ_0, succ_1, \varepsilon)$ *is decidable.*

Format of the grid: $G_2 = (\mathbb{N} \times \mathbb{N}, succ_1, succ_2, (0,0))$
$succ_1(x,y) = (x+1, y)$ $succ_2(x,y) = (x, y+1)$

Theorem 10 *(Seese, ~1975)* $Mth(G_2)$ *is undecidable*

Proof:
Use reduction of the halting problem for Turing machines.
Task: Given TM M, construct sentence φ_M s.t. M halts started on empty tape $\Leftrightarrow$ $G_2 \models \varphi_M$
Use TM on left-bounded tape
TM-computation is sequence of configurations $C_0, C_1, C_2, \ldots$
Convention: Repeat halting configuration
Halting signalled by "stop state" q_s
Idea Express existence of halting computing computation of M by requiring a corresponding labelling

	C_0	q_0	$\sqcup$	$\sqcup$	$\sqcup$	$\sqcup$	$\ldots$	$q_0 \sqcup 1Rq_1$
od G_2	C_1	1	q_1	$\sqcup$	$\sqcup$	$\sqcup$	$\ldots$	$q_1 \sqcup 1Nq_0$
	C_2	1	q_0	1	$\sqcup$	$\sqcup$	$\ldots$	

For construction of φ_M use work-alphabet $\{a_0, \ldots, a_n\}$ and M-states $q_0, \ldots, q_k$
Introduce $X_0, \ldots, X_n, Y_1, \ldots, Y_k$
X_i = set of positions where a_i occurs
Y_j = set of positions where q_j occurs
$\varphi_M : \exists X_0, \ldots, X_n Y_1, \ldots, Y_k$ (Partition $(X_0, \ldots, Y_k)\wedge$
"first row corresponding to inital conf. (empty tape)" $[Y_0(0,0) \wedge \forall y(S_2^+((0,0), y) \rightarrow X_0(y))]$
$*\wedge$ "each successor row corresponds to successor configuration of preceeding row"
$\wedge \exists x Y_k(x)$

For $(*)$ with down condition on 2×4 boxes of grid. $\begin{array}{cccc} \sqcup & q_0 & \sqcup & \sqcup \\ \sqcup & 1 & q_1 & \sqcup \end{array}$ Because the Turing Maching is deterministic: For each labelling wich starts in the first row and wich continues in admissible windows, a stop state will be reached.

4 Model-Checking and Temporal Logics

Model-Checking-Problem: Given Structure/System SYS, specification $SPEC$.
Does SYS satisfy $SPEC$?
Plan: Formalisation/ automata-theoric approach.

1. Kripke structures as system models

2. Basic specifications

3. Formal specification languages

   $[S1S]$: model-checking is very hard($O(2)^{2^{2^{\iddots^{2}}}}$ (exp. k times), non elementary
   Introduce temporal logic LTL, show that M.C. is PSPACE-complete

4. Use automata to solve the m.c. problem

1. Kripke Structures:
   Let $p_1 \dots p_n$ atomic propositions (base "state properties")
   A Kripke structure over $p_1 \dots p_n$ is a tuple $M = (S, R, \lambda)$ where

   - S is a finite set of "states"
   - R is a transition relation, $R \subseteq S \times S$ (($s_1, s_2 \in R$: the system can go from s_1 to state s_2)
   - λ is a "labelling function", $\delta : S \to S^{p_1 \dots p_n}$
   - $p_i \in \lambda(S)$: the base property p_i is true at state s.

   Example: traffic light, three atomic propositions:
   p_1 ~ red light is on
   p_2 ~ yellow light is on
   p_3 ~ green light is on

$$
\begin{tikzcd}
p_1 & p_1, p_2 \\
S_1 \arrow[r] & S_2 \arrow[d, "p_3"] \\
S_4 \arrow[u] & S_3 \arrow[l]
\end{tikzcd}
$$

   Notations:

   a) A pointed K.S. is a K.S. $\mathcal{M} = (S, R, \lambda)$ with an initial state $s \in S$

b) Usually, we write $\lambda(s)$ as a bit vector $\in (B)^n : \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix} : b_i = 1$ iff $p_i \in \lambda(S)$

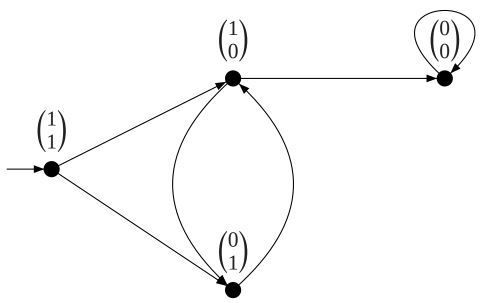
c) Convention we don't allow...

Definition
1) A Path through a K.S. $\mathcal{M} = (S, Q, \lambda)(\mathcal{M}, S)$ is an inifinite sequence of states $s_0, s_1, s_2 \ldots$ with:
$s_0 = s$
$(s_i, s_{i+1}) \in R$ for all $i \in \mathbb{N}$
2) Label sequences for a path $s_0 s_1 s_2 \ldots$ is the ω-word $\lambda(s_0)\lambda(s_1)\ldots$
3) The language of $(\mathcal{M}, S)$ is the set of label sequences of paths through $(\mathcal{M}, S)$, we write $L(\mathcal{M}, s) \in (B)^\omega$

Model-checking Problem revisited:
Given a Kripke structure $(\mathcal{M}, S)$ over $p_1 \ldots p_n$, and a specification φ on ω-words over $\ltimes$, does every path through $(\mathcal{M}, S)$ satisfy φ?
$L(\mathcal{M}, S) \subseteq L(\varphi)$?

24.01.06

Review Model-Checking-Problem:
Given Kripke-structure specification:Logical formula
$(\mathcal{M}, s) \models \varphi$?
Does $(\mathcal{M}, s)$ satisfy φ?



$L(\mathcal{M}, s)$
Label sequence: $\binom{1}{1}\binom{1}{0}\binom{0}{1}\binom{1}{0}\binom{0}{0}\binom{0}{0}\ldots$

Approach for solution: Construct Büchi automata $\mathfrak{A}_{\mathcal{M},s}$ for $L(\mathcal{M}, s)$ and $\mathfrak{A}_\varphi$ for $L(\varphi)$ and check wether $L(\mathfrak{A}_{\mathcal{M},s}) \subseteq L(\mathfrak{A}_\varphi)$
Formulation of φ given often in "linear time temporal logic" LTL (in fact, subsystem of S1S).

Plan: Introduce LTL
Sketch translation from LTL $\rightarrow$ Büchi aut.
Solve MC Problem

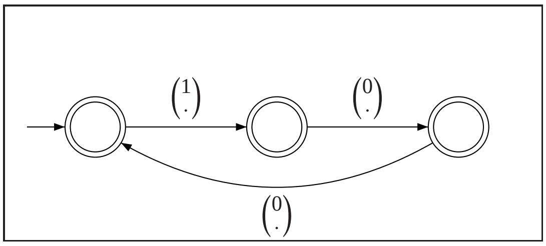
4.0.1 LTL

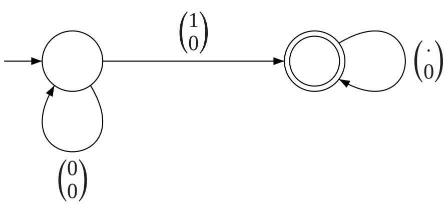
Basic sequence properties (over two state properties p_1, p_2)
Guaranteed property: "sometime p_1 becomes true" (E-aut.) [Fp_1]
Safety property: "alwys p_1 is true" (A-aut.) [Gp_1]
Periodicity property: "Initially p_1 is trie , and p_1 is true precisely every third moment." (A-aut.) [$p_1 \wedge X\neg p_1 \wedge XX\neg p_1 \wedge G(p_1 \leftrightarrow XXXp_1)$]
Obligation property: "Sometimes p_1 is true, and p_2 is never true" (SW-aut.) [$Fp_1 \wedge \neg Fp_2 \equiv Fp_1 \wedge G\neg p_2$]
Recurrence property: "Again and again, p_1 is true" (Büchi-condition) [GFp_1]
Request-response property: "Always when p_1 holds, then sometime later p_2 holds" [$G(p_1 \rightarrow XFp_2$]
Until property: "Always when p_1 holds, sometime later p_1 holds and in the meantime p_2 holds". [$G(p_1 \rightarrow X(p_2Up_1))$]
Fariness property: "If p_1 is true again and again, so is p_2" [$GFp_1 \rightarrow GFp_2$]

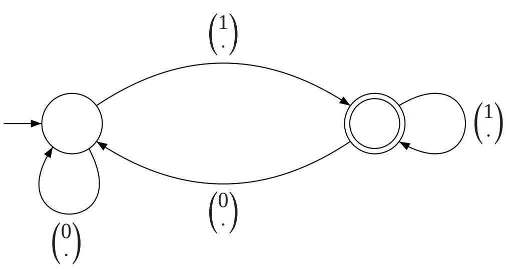
Fomalisations with temporal operators:

X "next"	F "sometimes"
G "always"	U "until"

Remark:
All formulas can be expressed by Büchi automata over $\{0,1\}^2$

Periodicity:


obligtion:


recurrence:


LTL-Syntax

LTL-formulas over $p_1 \dots p_n$ are defined inductively as follows:

- p_i ist LTL formula ($i = 1, \dots, n$)
- If φ, ψ are LTL-formulas, then also $\neg\varphi, \varphi \vee \psi, \varphi \wedge \psi, \varphi \to \psi, \varphi \leftrightarrow \psi$ [$\neg, \vee$] suffices
- If φ, ψ are LTL-formulas, then also $X\varphi$ $F\varphi$ $G\varphi$ $\varphi U \psi$

In ω-sequences over $\Sigma = \{0, 1\}^n$
Notation:
For $\alpha \in (\{0, 1\}^n)^\omega$, $\alpha = \alpha(0)\alpha(1)\cdots$
$\alpha^i = \alpha(i)\alpha(i + 1)\alpha(i + 2)\dots$
$(\alpha(i))_j$ = j-th component of $\alpha(i)$
Satisfaction relation "$\alpha^i \vDash \varphi$" is defined inductively:
$\alpha^i \vDash p_j \Leftrightarrow (\alpha(i))_j = 1$
$\alpha^i \vDash \neg\varphi \Leftrightarrow$ *not* $\alpha^i \vDash \varphi$ similarly for $\vee, \wedge, \to\leftrightarrow$
$\alpha^i \vDash X\varphi \Leftrightarrow \alpha^{i+1} \vDash \varphi$
$\alpha^i \vDash F\varphi \Leftrightarrow \exists j \geq i\ \alpha^j \vDash \varphi$
$\alpha^i \vDash G\varphi \Leftrightarrow$
$forall j \geq i\ \alpha^j \vDash \varphi$
$\alpha^i \vDash \varphi U \psi \Leftrightarrow \exists j \geq i(\alpha^j \vDash \psi \wedge \forall k(i \leq k < j \Rightarrow \alpha^k \vDash \varphi))$

Example:
GFp_1

$$\begin{array}{rcl} \alpha(0\alpha^0 & \vdash & GFp_1 \\ & \Leftrightarrow & \forall j \geq 0\alpha^j \vDash Fp_1 \\ & \Leftrightarrow & \forall j \exists k \geq j\, \underbrace{\alpha^k \vDash p_1}_{(\alpha(k))_1=1} \\ & \Leftrightarrow & \text{infinitely often } p_1 \text{ is true} \end{array}$$

Evaluation of LTL-formulas

$\varphi : F(\neg p_1 \wedge X(\neg p_2 U p_1))$

$$\varphi\text{–Expension of } \alpha \quad \begin{array}{rccccccccc} \alpha = & \binom{1}{0} & \binom{0}{1} & \binom{1}{1} & \binom{0}{0} & \binom{1}{0} & \binom{0}{1} & \binom{0}{1} & \binom{0}{1} & \dots \\ \neg p1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & \dots \\ \neg p_2 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & \dots \\ \neg p_2 U p_1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & \dots \\ X(\neg p_2 U p_1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & \dots \\ \neg p_1 \wedge X(\neg p_2 U p_1) & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & \dots \\ F(\neg p_1 \wedge X(\neg p_2 U p_1)) & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & \dots \end{array}$$

Theorem 11 (Main Theorem) *An LTL formula φ over $p_1 \dots p_n$ can be transformed into a Büchi automaton $\mathfrak{A}_\varphi$ over $\Sigma = \{0, 1\}^n$ such that for all $\alpha \in \Sigma^\omega$ $\alpha \vDash \varphi \Leftrightarrow \mathfrak{A}_\varphi$ can be constructed with state set $\{q_0\} \cup \{0, 1\}^m$*

31.01.06

Idea: Declare the bit vectors for truth of subformulas as states of Büchi automaton
Technical Preparation:

a) $tt : p_1 \vee \neg p_1$
$ff : p_1 \wedge \neg p_1$
$F : ttU\varphi$
$G : \neg F \neg \varphi$ Temp op. X, U suffices.

b) $\varphi U \psi \Leftrightarrow \psi \vee (\varphi \wedge X(\varphi U \psi))$

c) Generalized Büchi automaton: $\mathfrak{A} = (Q, \Sigma, q_0, \Delta, F_1, \ldots, F_k)$
$\mathfrak{A}$ accepts $\alpha \;\Leftrightarrow$ exists run of $\mathfrak{A}$ ond α such that each $F_i (i = 1, \ldots, k)$ is visited infinitely often.

Lemma (Translation Lemma)
Any LTL formula φ over $p_1 \ldots p_n$ wih temporal operators X, U only and with m subformulas $\psi_1 \ldots \psi_m (\neq p_i)$ can be transformed into a generalized Büchi-automaton with state set $\{q_0\} \cup \{0,1\}^m$
Aim: Automaton has a unique successful run, namely the sequence of truth-value vectors for $\psi_1 \ldots \psi_n$

Given α, subformulas $\psi_1, \ldots, \psi_m$ of φ, the φ-expansion of α satisfies the compatibility conditions:
$\psi_j = \neg \psi_{j_1} : (\beta(i))_j = 1 \Leftrightarrow (\beta(i))_{j_1} = 0$
$\psi_j = \psi_{j_1} \vee \psi_{j_2} : (\beta(i))_j = 1 \Leftrightarrow (\beta(i))_{j_1}$ or $\beta(i))_{j_2} = 1$
$\psi_j = X\psi_{j_1} : (\beta(i))_j = 1 \Leftrightarrow (\beta(i+1))_{j_1} = 1$ $\psi_j = \psi_{j_1} U \psi_{j_2} : (\beta(i))_j = 1 \Leftrightarrow (\beta(i))_{j_2} = 1 \vee ((\beta(i))_{j_1} =$
7.2.06 $1 \wedge ((\beta(i))_{j_2} = 0$ $*$in the last case (U-Formula): there is no k s.t. for $i > k$ $(\beta(i))_j = 1$ but $(\beta(i))_{j_2} = 0$.

4.0.2 LTL $\rightarrow$ Büchi automata

Comaprison LTL - FO (first order logic over ω-words)

Example:
$G(p_1 \rightarrow X(p_2 U p_1))$ $\qquad$ p_1 at time $x \rightarrow X_1(x)$
$\forall s\, (X_1(s) \rightarrow \exists t\, (s < t \wedge X_1(t) \wedge \forall r (s < r < t \rightarrow X_2(r)))$

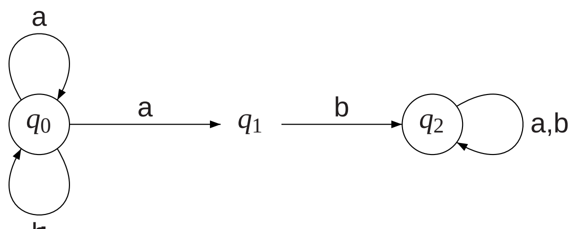
Theorem 12 *LTL and FO are of same expressive Power.*

Proof:
LTL $\rightarrow$ FO: easy by induction
FO $\rightarrow$ LTL: Difficult. (superexponential blowup in formula length)
Intuition: FO-Quantification can be restricted to intervals $[s, t]$ $(\forall r (s < r < t \rightarrow \ldots))$
Illustration for LTL: $p_1 \wedge X(p_2 \wedge Fp_3) U p_1)$

Theorem 13 *a) An LTL-formula with m distinct subformulas can be translated into a Büchi automaton with $O(2^m)$ states.*

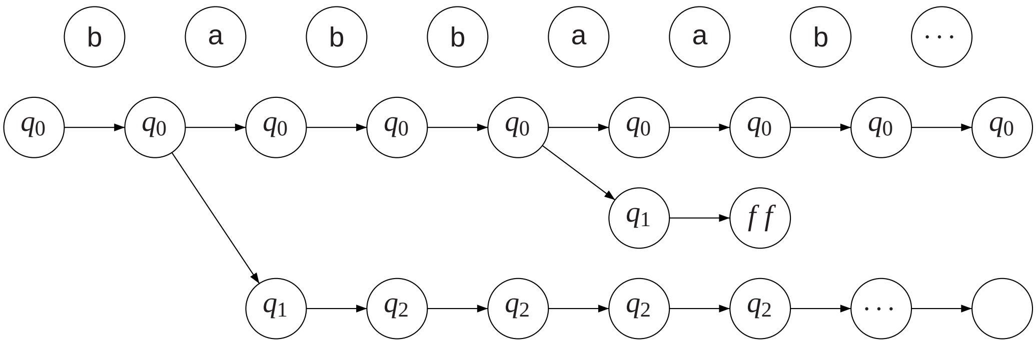
b) An FO-formula with m connectives is translatable to Büchi aut with $m \left\{ 2^{\cdot^{\cdot^{\cdot^{2^m}}}} \right.$ *states*

Translation LTL → Büchi automata via alternating Büchi automata (ABA)
Idea of alternating automaton: Allow existential (or-) branching as in nondet. aut. and universal (and-) branching.



Example:

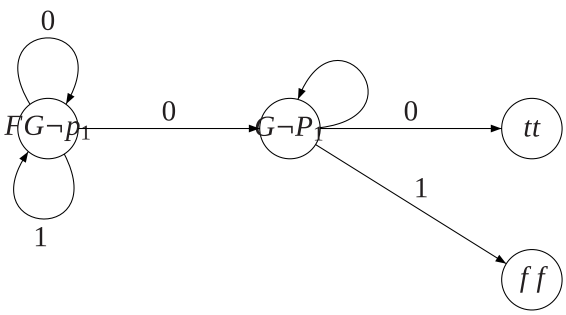
Run tree on input

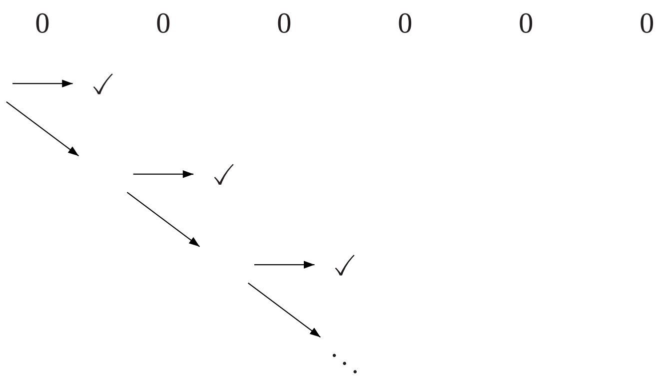


Nondetdermin-ism generates different run trees (for each nondet. choice a new run tree).
Alt. Büchi automaton accepts α iff exists run tree on α such that all branches of it are successful (end in tt or visit final state inifinitely often)

Theorem 14 *An LTL-formula can be translated into an Alt. Büchi automaton where the set of states is the set of subformulas (with ff, tt)*

Illustration$FG\neg p_1$ (input alphabet: $\Sigma = \{0, 1\}$)

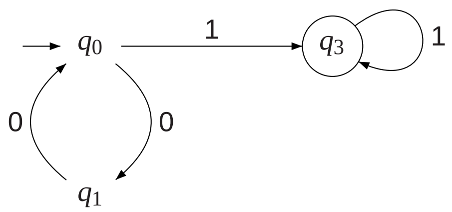




Second Step: Transformation of ABA into standard Büchi automaton
As states use sets of ABA-states, updated according to the growth of ABA run tree(s)
Comparison of LTL (or FO) with Büchi automata (or S1S)

Theorem 15 *Büchi automata are strictly more expressive then LTL*

Example:
$L_0 = (00)^* 1^\omega$ is not LTL-definable.



$L = (10)^* 1^\omega$ is LTL-definable.

Proof:
Proof strategy

- Introduce property “non-counting” for ω-languages L
- Show that each LTL-def ω-language has this property
- $L_0 = (00)^* 1^\omega$ violoates this property

$L \subseteq \Sigma^\omega$:⇔ for sufficiently large n: $\forall xy\beta \quad xy^n\beta \in L \Leftrightarrow xy^{n+1}\beta \in L$
Negation: L “counting”: there are inifinitely many n and $xy\beta$ sucht that $xy^n\beta \in L$, $x, y^{n+1}\beta \notin L$ or conversely.
L_0 is counting: take any even n, $x = \varepsilon$, $y = 0$, $\beta = 1^\omega$
$(oo)^* 1^\omega$: $\quad xy^n\beta \in L_0, xy^{n+1}\beta \notin L_0$

4.1 Beyond reular ω-languages

Scale of complexity for ω-languages:

Level 1: ω-languages of form $L = W \cdot \Sigma^\omega$ $\alpha \in L \Leftrightarrow$ ex. α prefix in W, $W \subseteq \Sigma^*$

Level 2: ω-languages of form $L = limW$ $(W \subseteq \Sigma^*)$

General construction: Borel hierarchy

Level 1 $\underline{\Sigma_1}$ class of $L = W \cdot \Sigma^\omega$ with $W \subseteq \Sigma^*$
$\overline{\Pi_1}$: class of complements of Σ_1-languages

Level (n+1) $\Sigma_{(n+1)}$: class of countable unions $\bigcup_i L_i$ with $L_i \in \Pi_n$
Π_{n+1} class of countable intersections $\bigcap_i L_i$ with $L_i \in \Sigma_n$
Remark: Π_2 = class of languages *limW*

Contents