Mitschrift

Automata on Infinite Words - Exercises

Dr. S. Wöhrle

Dipl.-Math. Philipp Rohde[*]

Ulrich Loup

Diese Mitschrift ist eine Mitschrift und deswegen nicht unbedingt vollständig oder fehlerfrei! Das Dokument wird bis zum Ende des Semesters sukzessive erweitert. Anmerkungen bitte an die angegebene email-Adresse auf der `s-inf` Seite.

$*$: Ab dem 11.01.2006 wurde die Übung von Philipp Rhode übernommen.

Contents

1 Exercise from 10-19-2005

1.1 Semantics of regular expressions

$[\![\varepsilon]\!] := \{\varepsilon\}$, $[\![\emptyset]\!] := \emptyset$, $[\![a]\!] := \{a\} \quad \forall a \in \Sigma$

If r, s are regular expressions then

- $[\![r + s]\!] := [\![r]\!] \cup [\![s]\!]$
- $[\![r \cdot s]\!] := \{uv \in \Sigma^* \mid u \in [\![r]\!], v \in [\![s]\!]\}$
- $[\![r^*]\!] := \{w \in \Sigma^* \mid \exists n \in \mathbb{N} : w = u_1 \dots u_n \text{ and } u_i \in [\![r]\!] \; \forall i \leq n\}$

Note: We often do distinguish syntax and semantics! E.g. we often write $U \cdot V,\ r \cup s,\ r \cdot U$.

1.2 Definition (ω-regular language)

An ω-regular language over Σ is of the form $r_1 \cdot s_1^\omega + \dots + r_n \cdot s_n^\omega$ for some $n \in \mathbb{N}$ and regular expressions r_i, s_i for all $i \leq n$.

semantics: extend semantics of regular expressions by

- $[\![r^\omega]\!] := \{\alpha \in \Sigma^\omega \mid \alpha = w_1 w_2 \dots, w_i \in [\![r]\!] \; \forall i \in \mathbb{N}\}$
- $[\![r \cdot s^\omega]\!] := \{\alpha \in \Sigma^\omega \mid \alpha = w\beta,\ w \in [\![r]\!], \beta \in [\![s^\omega]\!]\}$
- $r_1 s^\omega + \dots + r_n s^\omega := \bigcup_{i \leq n} [\![r_i s_i^\omega]\!]$

Note:

- $(s^\omega)^\omega$, $s\omega \cdot r$, $s^\omega \cdot r^\omega$, $r \cdot (s_1^\omega + s_2^\omega)$ are not ω-regular expressions!
- $[\![\varepsilon^\omega]\!] = \emptyset$

1.3 Connection to Büchi automata

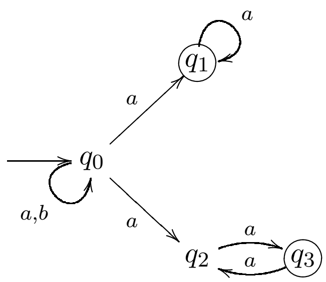
Recall: Given a Büchi automaton $\mathfrak{A} = (Q, \Sigma, q_0, \Delta, F)$. The language $L(\mathfrak{A})$ can be described by the ω-regular expressioin

$$\bigcup_{q \in F} W_{q_0 q} W {q_q}^\omega$$

where W_{qq} is the regular language recognized by the automaton $\mathfrak{A}_{p,q} = (Q, \Sigma, p, \Delta, \{q\})$. From $\mathfrak{A}$ (by Kleene's theorem) we can construct r_{pq} such that $[\![r_{pq}]\!] = L(\mathfrak{A}_{pq})$.

1.4 Example

$\mathfrak{A}$:

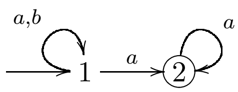


"reaching"	*loop*
$W_{q_0q_1} = (a+b)^*a$	$W_{q_1q_1} = a^*$
$W_{q_0q_3} = (a+b)^*ab(ab)^*$	$W_{q_3q_3} = (ab)^*$

Thus $L(\mathfrak{A}) = (a+b)^*a(a^*)^\omega + (a+b)^*ab(ab)^*((ab)^*)^\omega = (a+b)^*a^\omega + (a+b)^*(ab)^\omega$.

2 Exercise from 10-26-2005

Büchi's complementation procedure

$\mathfrak{A}$: 

recognizes $L(\mathfrak{A}) = (a+b)^*a^\omega$.

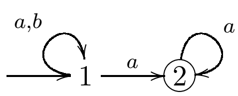
2.1 Equivalence relation $\sim_{\mathfrak{A}}$

$\forall p,q \in Q\ ((p \xrightarrow{u} q \ \Leftrightarrow\ p \xrightarrow{v} q) \wedge (p \overset{u}{\Rightarrow} q \ \Leftrightarrow\ p \overset{v}{\Rightarrow} q)) =: u \sim_{\mathfrak{A}} v$.

Lemma: $\sim_{\mathfrak{A}}$ is even a conguence, i.e. $\forall u,v \in \Sigma^*\ \forall a \in \Sigma\ u \sim_{\mathfrak{A}} v \rightarrow ua \sim_{\mathfrak{A}} va$.

Consequence: $\forall u,v \in \Sigma^*\ \forall a \in \Sigma\ [u] = [v] \Rightarrow [ua] = [va]$.

2.2 Transition profiles

$\mathfrak{A}$: 

$[\varepsilon]$	$[a]$	$[b]$	$[aa]$	$[ab]$	$[ba]$	$[bb]$	$[baa]$	$[bab]$
$1 \rightarrow 1$ $2 \Rightarrow 2$	$1 \rightarrow 1$ $1 \Rightarrow 2$ $2 \Rightarrow 2$	$1 \rightarrow 1$	$1 \rightarrow 1$ $1 \Rightarrow 2$ $2 \Rightarrow 2$	$1 \rightarrow 1$	$1 \rightarrow 1$ $1 \Rightarrow 2$	$1 \rightarrow 1$	$1 \rightarrow 1$ $1 \Rightarrow 2$	$1 \rightarrow 1$
			$= [a]$	$= [b]$		$= [b]$	$= [ba]$	$= [b]$

$[\varepsilon]$, $[a]$, $[b]$, $[ba]$ are *all* equivalence classes. Proof: Use Congruence Lemma.

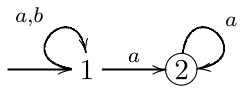
2.3 Computing the equivalence classes as sets

Recall: $W_{p,q} := \{u \mid \mathfrak{A} : p \xrightarrow{u} q\}$ and $W^F_{p,q} := \{u \mid \mathfrak{A} : p \overset{u}{\Rightarrow} q\}$.
Fact: (from the definition of $\sim_{\mathfrak{A}}$)

$[u]$ = the intersection of

- all $W_{p,q}$ with $p \xrightarrow{u} q$, all $W^F_{p,q}$ with $p \overset{u}{\Rightarrow} q$
- all $\Sigma^* \setminus W_{p,q}$ with not $p \xrightarrow{u} q$, all $\Sigma^* \setminus W^F_{p,q}$ with not $p \overset{u}{\Rightarrow} q$

$\mathfrak{A}$:



$$W_{1,1} = (a+b)^* = \Sigma^* \qquad W_{1,1}^F = \emptyset$$
$$W_{1,2} = \emptyset \qquad \mathbf{W_{1,2}^F = (a+b)^*a^+} \qquad \mathbf{\Sigma^* \setminus W_{1,2}^F = (a^*b)^*}$$
$$W_{2,2} = \emptyset \qquad \mathbf{W_{2,2}^F = a^*} \qquad \mathbf{\Sigma^* \setminus W_{2,2}^F = a^*b(a+b)^*}$$

$$\begin{aligned}
[\varepsilon] &= \Sigma^* \setminus W_{1,2}^F \cap W_{2,2}^F = (a^*b)^* \cap a^* = \varepsilon \\
[a] &= W_{1,2}^F \cap W_{2,2}^F = (a+b)^*a^+ \cap a^* = a^+ \\
[b] &= \Sigma^* \setminus W_{1,2}^F \cap \Sigma^* \setminus W_{2,2}^F = (a^*b)^* \cap a^*b(a+b)^* = (a^*b)^+ \\
[ba] &= W_{1,2}^F \cap \Sigma^* \setminus W_{2,2}^F = (a+b)^*a^+ \cap a^*b(a+b)^* = a^*b(a+b)^*a^+
\end{aligned}$$

equivalence classes:

$[\varepsilon] = \varepsilon$

$[a] = a^+$

$[b] = (a^*b)^+$

$[ba] = a^*b(a+b)^*a^+$

$W_{\mathfrak{A}} = \{[\varepsilon], [a], [b], [ba]\}$ and $\Sigma^* = [\varepsilon] \,\dot\cup\, [a] \,\dot\cup\, [b] \,\dot\cup\, [ba]$.

Task: Compute U, V such that $U \cdot V^\omega \nsubseteq L(\mathfrak{A}) = (a+b)^*a^\omega$. Therefore $V = [\varepsilon]$ or $V = [a]$ is not allowed. All other combinations contain $\alpha \notin L(\mathfrak{A})$.

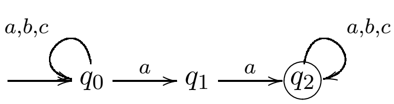
Hence:

$$\begin{aligned}
\Sigma^\omega \setminus L(\mathfrak{A}) &= (a^*b)^\omega + (a^*b(a+b)^*a^+)^\omega \\
&\quad + a^+(a*b)^\omega + a^+(a^*b(a+b)^*a^+)^\omega \\
&\quad + (a^*b)^\omega + (a^*b)^+(a^*b(a+b)^*a^+)^\omega \\
&\quad + (a^*b(a+b)^*a^+)(a^*b)^\omega + (a^*b(a+b)^*a^+)^\omega \\
&= (a^*b)^\omega
\end{aligned}$$

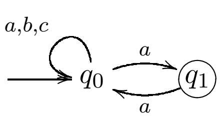
3 Exercise from 11-02-2005

3.1 Exercise 1

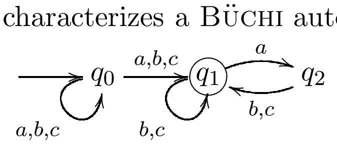
(a)



(b)



(c) Idea: rewrite L_3 into "from some point onwards: after every a either b or c" (such that L_3 characterizes a Büchi automaton).



3.2 Exercise 2

(a) $(a+b+c)^*aa(a+b+c)^\omega$

(b) $((a+b+c)^*aa)^\omega$

(c) $(a+b+c+aa)^*((b+c)+a(b+c))^\omega$

3.3 Exercise 3

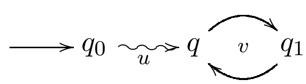
There exists a function $b : \mathbb{N} \to \mathbb{N}$ such that

(i) $\forall$ Büchi automata $\mathcal{A}$ with n states with $L := L(\mathcal{A}) \neq \emptyset$ $\exists w \in \Sigma^*$ with $w = uv$, $uv^\omega \in L$ and $|u| + |v| \leq b(n)$.

(ii) $\exists$ a Büchi automaton $\mathcal{A}$ with n states such that $\nexists w \in L := L(\mathcal{A})$ with $w = uv^\omega$ and $|u| + |v| < b(n)$.

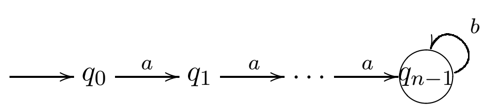
Proof: Set $b(n) := n$.
Since $L(\mathcal{A}) \neq \emptyset$ there exists a loop in the graph of $\mathcal{A}$ such that

- it contains a state $q_1 \in F$
- it is reachable from initial state q_0



Define u, v, we know $uv^\omega \in L(\mathcal{A})$ where we choose u to be as short as possible, i.e. $|u| := m < n$. The path $q_0 \overset{u}{\rightsquigarrow} q$ does not contain a state from loop (except the last one). So only $n-m$ states remain for loop.
The bound $b(n) = n$ cannot be improved:
Consider the family of Büchi automata $(\mathcal{A}_n)_{n\in\mathbb{N}}$.

$\mathcal{A}_n$:



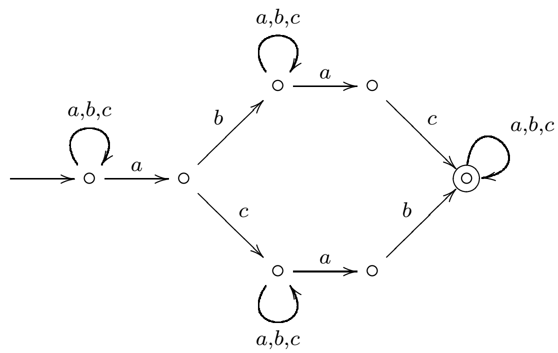
$L(\mathcal{A}_n) = a^{n-1}b^\omega$. Obviously $a^{n-1}b^\omega$ cannot be decomposed in u, v with $|u|+|v|< n$ and $uv^\omega = a^{n-1}b^\omega$.
□

4 Exercise from 11-09-2005

4.1 Exercise 4

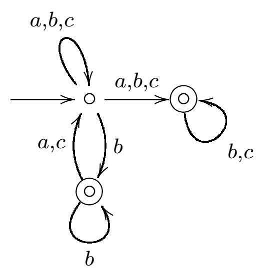
$\Sigma = \{a, b, c\}$.

- $L := \{\alpha \in \Sigma^\omega \mid \alpha$ contains at least one infix ab and one infix $ac\}$.
  Automaton:



  ω-regular expression: $(a+b+c)^* a^* (b(a+b+c)^* ac + c(a+b+c)^* ab)(a+b+c)^\omega$

- $K := \{\alpha \in \Sigma^\omega \mid$ if α contains infinitely many a then α contains infinitely many $b\}$.
  Automaton:



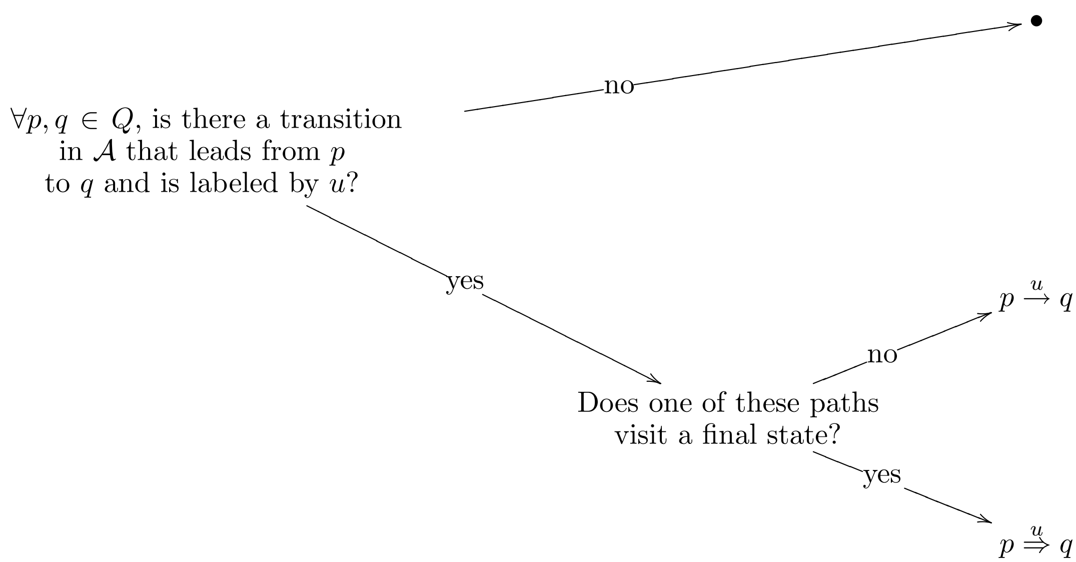
  ω-regular expression: $(a+b+c)^*(a+b+c)(b+c)^\omega + (a+b+c)^* b(b^* + b^*(a+c)(a+b+c)^* b(^\omega$
  direct construction: $\Sigma^*(b+c)^\omega + \Sigma^*(b\Sigma^*)^\omega$ (here Σ^* instead of $(a+b+c)^*$).

4.2 Exercise 5

(a) *ω-regular expression:* $(a+b)^* a(a+ba)^\omega$,
$L(\mathcal{A}) = \{\alpha \in \Sigma^\omega \mid \alpha$ contains finitely often the infix $bb\}$.

(b) $\sim_\mathcal{A}$-class: shortest representatives and transition profiles

Transition profiles of $u \in \Sigma^*$:



$u \sim_\mathcal{A} v \Leftrightarrow (\forall p, q : (p \xrightarrow{u} q \Leftrightarrow p \xrightarrow{v} q) \wedge (p \overset{u}{\Rightarrow} q \Leftrightarrow p \overset{v}{\Rightarrow} q))$

$\sim_\mathcal{A}$ is a congruence (not only right-congruence): If $u \sim_\mathcal{A} v$, then $\forall w_1, w_2 \in \Sigma^*$: $w_1uw_2 \sim_\mathcal{A} w_1vw_2$.

Consequence: $u \sim_\mathcal{A} v$ and $|v| < |u|$, then every word that has u as prefix is equivalent to some word that doesn't have u as prefix and is shorter than the first word. $uw \sim_\mathcal{A} vw = ux \sim_\mathcal{A} vx$.

transition profiles:

$[\varepsilon]$	$[a]$	$[b]$	$[aa]$	$[ab]$	$[ba]$	$[bb]$	$[aba]$	$[abb]$	$[baa]$	$[bab]$
$1 \to 1$	$1 \to 1$	$1 \to 1$	$1 \to 1$	$1 \to 1$	$1 \to 1$	$1 \to 1$	$1 \to 1$	$1 \to 1$	$1 \to 1$	$1 \to 1$
$2 \Rightarrow 2$	$1 \Rightarrow 2$	$2 \Rightarrow 3$	$1 \Rightarrow 2$	$1 \Rightarrow 3$	$1 \Rightarrow 2$		$1 \Rightarrow 2$		$1 \Rightarrow 2$	$1 \Rightarrow 3$
$3 \Rightarrow 3$	$2 \Rightarrow 2$		$2 \Rightarrow 2$	$2 \Rightarrow 3$	$2 \Rightarrow 2$		$2 \Rightarrow 2$		$2 \Rightarrow 2$	$2 \Rightarrow 3$
	$3 \Rightarrow 2$		$3 \Rightarrow 2$				$3 \Rightarrow 2$			
			$= [a]$				$= [a]$	$= [bb]$	$= [ba]$	

$[bba]$	$[bbb]$	$[baba]$	$[babb]$	$[bbaa]$	$[bbaba]$	$[bbabb]$
$1 \to 1$	$1 \to 1$	$1 \to 1$	$1 \to 1$	$1 \to 1$	$1 \to 1$	$1 \to 1$
$1 \Rightarrow 2$		$1 \Rightarrow 2$		$1 \Rightarrow 3$	$1 \Rightarrow 2$	
		$2 \Rightarrow 2$				
	$= [bb]$	$= [ba]$	$= [bb]$		$= [bba]$	$= [bb]$

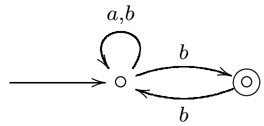
e.g. verify $aa \sim a \Rightarrow baa \sim ba$.

(c) Each $\alpha \in \Sigma^\omega$ can be factorized as $\alpha \in U \cdot V^\omega$ with U and V equivalence classes of $\sim_\mathcal{A}$.

- $\alpha_1 = ababbabbbabbbba\ldots$
  Since $bbb \sim_\mathcal{A} bb$ we get $a\underbrace{b\ldots b}_{n\geq 2} \sim_\mathcal{A} bb$. Thus $\alpha_1 \in [ab] \cdot [bb]^\omega$.

- $\alpha_2 = abaabaaabaaaa\ldots$
  Since $aa \sim_{\mathcal{A}} a$ we get $b\underbrace{a\ldots a}_{n\geq 2} \sim_{\mathcal{A}} ba$. So $\alpha_2 \in [a]\cdot[ba]^\omega$. Alternative: $\alpha_2 \in [\varepsilon]\cdot[a]^\omega$.

(d) direct construction:



$\Sigma^\omega \setminus L(\mathcal{A}) = \{\alpha \in \Sigma^\omega \mid \alpha \text{ contains infinitely many } bb\}$.

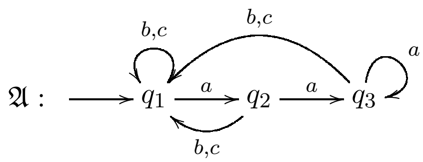
5 Exercise from 11-16-2005

5.1 Exercise 6

UP defines the class of ultimately periodic words.
Claim: UP is not regular.
Assume that UP is regular, then $\Sigma^\omega \setminus$ UP is also regular. But any regular language contains an ultimately periodic word. Contradiction. □

5.2 Exercise 7

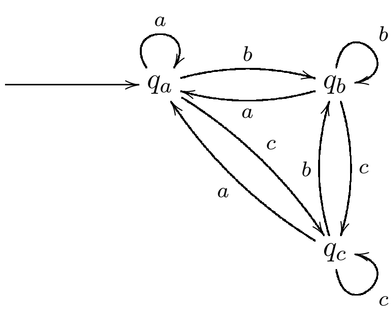
(a)



$\mathcal{F} = \{\{q_1\}, \{q_1, q_2\}\}$
$L(\mathfrak{A}) = L_1$

Remark: Muller automata: $\mathcal{F} \subseteq 1^Q$. Run ρ is accepting $:\Leftrightarrow \text{Inf}(\rho) \in \mathcal{F}$.

(b)

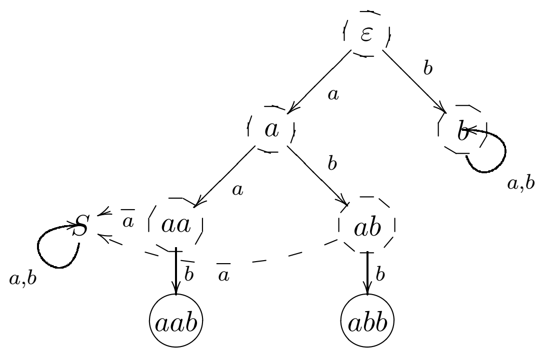


$\mathcal{F} = \{\{q_c\}, \{q_b\}, \{q_b, q_c\}, \{q_a\}, \{q_a, q_c\}\}$.
(Exclude $\{q_a, q_b\}, \{q_a, q_b, q_c\}$, the rest must work.)
$\{q_c\}$: neither a nor b was seen.
$\{q_b\}, \{q_b, q_c\}$: b was ∞-often but a finitely often seen.
$\{q_a\}, \{q_a, q_c\}$: a was ∞-often but b finitely often seen.

5.3 Exercise 8

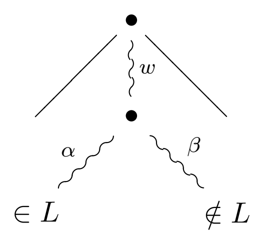
(a) Let $U \subseteq \Sigma^*$, U finite, $L = U \cdot \Sigma^\omega$.
Claim: L is E- and A-recognizable.
Proof: W.l.o.g. (without loss of generality) assume that U contains only max. elements w.r.t. (with respect to) $\sqsubseteq$-relation (prefix-relation).
$w \in U$, $w' \sqsubseteq w$, $w \neq w' \Rightarrow w' \notin U$.

Define $T := \{w' \sqsubseteq w \mid w \in U, w' \in \Sigma^*\}$, $E_a := \{(w, wa) \mid wa \in T\}$.
Example: $U = \{aab, abb, b\}$



E-automaton $\mathfrak{A}$: $L(\mathfrak{A}) = L$
A-automaton:
all final states (dashed)
new sink-state S

(b) Let L be E- and A-recognizable.
complement-lemma: $\Sigma^\omega \setminus L$ is E-recognizable. [swapping final/non-final states]
There is an E-automaton $\mathfrak{A}$ with $L(\mathfrak{A}) = L$.
There is an A-automaton $\mathfrak{A}'$ with $L(\mathfrak{A}') = L$
To contradiction: Assume there is no *finite* $U \subseteq \Sigma^*$ with $L = U \cdot \Sigma^\omega$.
Let $T := \{w \in \Sigma^* \mid \exists \alpha, \beta \in \Sigma^* : w\alpha \in L \wedge w\beta \notin L\}$



$\Rightarrow T$ is prefix-closed and with E_a as before: T is a tree. Assume that T is finite.
For any leaf of T, w and any $a \in \Sigma$

* either $\forall \alpha \in \Sigma^\omega : wa\alpha \in L$
* or $\forall \alpha \in \Sigma^\omega : wa\alpha \notin L$.

Define $U := \{wa \in \Sigma^* \mid w \in T, w \text{ is a leaf}, a \in \Sigma, \forall \alpha \in \Sigma^\omega : wa\alpha \in L\}$.
Assume that T is finite. Σ is finite $\Rightarrow U$ is finite and $L = U \cdot \Sigma^\omega$. Contradiction to T is finite. $\Rightarrow T$ is finite.
Tree T infinite and finitely branching ($|\Sigma| < \infty$).
König‘s lemma $\Rightarrow$ there is an infinite path in T, i.e there is $u\Sigma^\omega$ such that each finite prefix of u belongs to T.
Does $\mathfrak{A}$ accept u? If $\mathfrak{A}$ accepts $u \Rightarrow$ assumes final state after $u_1, \ldots, u_m$ (finitely many) $\beta \notin L$.

$$\left.\begin{array}{l}\text{But } L = L(\mathfrak{A}). \text{ So the answer is NO! } \Rightarrow u \notin L \\ \text{Analog.: } \mathfrak{A}' \text{ cannot accept } u. \Rightarrow u \notin \Sigma^\omega \setminus L \Rightarrow u \in L\end{array}\right\} \text{Contradiction!}$$

□

6 Exercise from 11-23-2005

6.1 Exercise 9

$U, V \subseteq \Sigma^*$, $U^\omega := \{\alpha \in \Sigma^\omega \mid \alpha = u_1u_2u_3\dots, u_i \in U\}$.
$\lim(U) := \{\alpha \in \Sigma^\omega \mid \forall i\ \exists j \text{ with } j \geq i : \alpha[0\dots j] \in U\}$
$U^+ := \{v \in \Sigma^* \mid \exists k \geq 1, v = u_1 \dots u_k, u_j \in U\}$

(a) $U^\omega = \lim(U^+)$? Answer: NO.

- $U^\omega \subseteq \lim(U^+)$: Let $\alpha \in U^\omega$, then $\alpha = u_1u_2u_3\dots$ with $u_i \in U$, thus u_1, u_1u_2, $u_1u_2u_3$,$\dots \in U^+$. Thus $\forall i\ \exists j$ such that $\alpha[0\dots j] \in U^+$ (choose j such that $j = |u_1 \dots u_l| \geq i$). $\Rightarrow$ $\alpha \in \lim(U^+) \Rightarrow U^\omega \subseteq \lim(U^+)$.
- $\lim(U^+) \nsubseteq U^\omega$: Choose $U = ba^*$.
  Choose $\alpha = ba^\omega \in \lim(U) \subseteq \lim(U^+)$.
  By definition of U^ω every word in $(ba^*)^\omega$ will contain infinitely many b. $\Rightarrow \alpha \notin U^\omega$. □

(b) $\lim(U \cup V) = \lim(U) \cup \lim(V)$. Answer: YES.
"$\supseteq$": Let $\alpha \in \lim(U) \cup \lim(V)$, w.l.o.g. $\alpha \in \lim(U)$. $\Rightarrow \alpha \in \lim(U \cup V)$.
"$\subseteq$": Let $\alpha \in \lim(U \cup V)$. Then $\forall i\ \exists j$ such that $\alpha[0\dots j] \in U$ or $\alpha[0\dots j] \in V$.
Let $N_U := \{j \mid \alpha[0\dots j] \in U\}$ and $N_V := \{j \mid \alpha[0\dots j] \in U\}$. At least one of the two sets has to be infinite. $\Rightarrow$ Either $\alpha \in \lim(U)$ or $\alpha \in \lim(V)$. □

6.2 Exercise 10

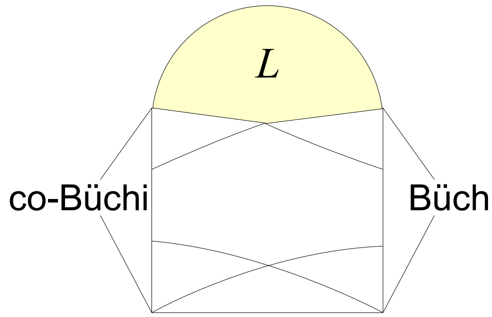
Every deterministically co-Büchi recognizable language is also deterministically Muller recognizable.
Proof: Let $\mathcal{A} = (Q, \Sigma, q_0, \delta, F)$ be a deterministic co-Büchi automaton. $\mathcal{A}$ accepts $\alpha \in \Sigma^\omega$ if $\exists i$ such that $\forall j \geq i$ we have $\rho_\alpha(j) \in F$, i.e $\mathrm{Inf}(\rho_\alpha) \subseteq F$.
Choose $\mathcal{B} = (Q, \Sigma, q_0, \delta, \mathcal{F})$ where $\mathcal{F} = 2^F = \{A \mid A \subseteq F\}$. Then

$$\alpha \in L(\mathcal{B}) \ \Leftrightarrow\ \mathrm{Inf}(\rho_\alpha) \in \mathcal{F} \ \Leftrightarrow\ \mathrm{Inf}(\rho_\alpha) \subseteq F \ \Leftrightarrow\ \alpha \in L(\mathcal{A})$$

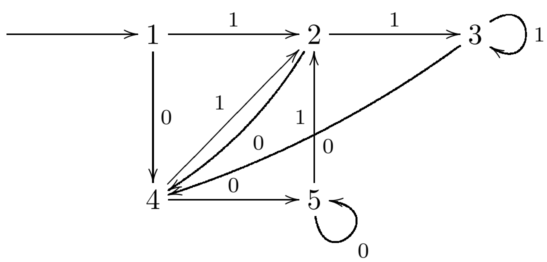
□

6.3 Exercise 11

$L := \{\alpha \in \{0,1\}^\omega \mid \alpha$ contains infix 00 infinitely often, but infix 11 only finitely often$\}$.
Show in 3 steps:



(a) A deterministic MULLER automaton for L (intuitively):



$\mathcal{F} = \{F \subseteq Q \mid 5 \in F, 3 \notin F\}$
equivalent:
$\mathcal{F} = \{\{5\}, \{2,4,5\}\}$

(b) L is not BÜCHI recognizable.

Proof: Assume it is, e.g. by $\mathcal{A}$.

Consider $\alpha_1 = 110^\omega \in L$, on α_1 $\mathcal{A}$ enters a final state, e.g. for the first time after 110^{n_1}.

Consider $\alpha_2 = 110^{n_1}110^\omega \in L$, $\mathcal{A}$ enters a second final state, e.g. after $110^{n_1}110^{n_2}$.

Consider $\alpha_2 = 110^{n_1}110^{n_2}110^\omega \in L, \ldots$.

$\vdots$

We obtain an infinite sequence of ω-words $\alpha_1, \alpha_2, \ldots \in L$, the runs of $\mathcal{A}$ on the initial parts agree. Thus α forming the common extension of $(\alpha_i)_i$ the automaton $\mathcal{A}$ will enter a final state infinitely often, but α contains 11 infinitely often. Contradiction. $\square$

(b) L is not deterministically co-BÜCHI recognizable.

Assume it is, e.g. by $\mathcal{A}$ with n states.

Let $\alpha = (00(01)^{n+1})^\omega \in L$.

From some point onwards $\mathcal{A}$ only enters final states on α, e.g. after reading $(00(01)^{n+1})^m$, so on $(00(01)^{n+1})^m \underbrace{(00(01)^{n+1})}_{\text{only states } \in F}$.

$\Rightarrow \alpha' = (00(01)^{n+1})^m 00(01)^\omega$ is accepted by $\mathcal{A}$, but $\alpha' \notin L$. Contradiction. $\square$

7 Exercise from 11-30-2005

7.1 Exercise 12

(a) Every nondeterministically E-recognizable language is also nondeterministically STAIGNER-WAGNER recognizable.

Proof: Let $\mathcal{A} = (Q, \Sigma, q_0, \Delta, F)$ be E-automaton. Construct $\mathcal{B} = (Q', \Sigma, q'_0, \Delta', \mathcal{F})$ a STAIGNER-WAGNER automaton by choosing $Q' := Q$, $q'_0 := q_0$, $\Delta := \Delta'$, $\mathcal{F} := \{F' \subseteq Q \mid F' \cap F \neq \emptyset\}$.
Then $\forall\ \alpha \in \Sigma^\omega$: $\mathcal{A}$ E-accepts α
$\Leftrightarrow$ there exists an infinite run ρ_α of $\mathcal{A}$ on α such that a state from F is visited
$\Leftrightarrow \text{Occ}(\rho) \cap F \neq \emptyset$
$\Leftrightarrow \text{Occ}(\rho_\alpha) \in \mathcal{F} \Leftrightarrow \mathcal{B}$ accepts α. □

(b) Every nondeterministically STAIGNER-WAGNER recognizable language is also nondeterministically co-BÜCHI recognizable.

Proof: Let $\mathcal{A} = (Q, \Sigma, q_0, \Delta, \mathcal{F})$ be STAIGNER-WAGNER automaton. Construct co-BÜCHI automaton $\mathcal{B} = (Q', \Sigma, q'_0\Delta', F')$ as follows:
$Q' := Q \times 2^Q$, $\quad q'_0 := (q_0, \emptyset)$.
Δ' given $((p, P), a, (q, R)) \in \Delta' \Leftrightarrow (p, a, q) \in \Delta$ and $R = P \cup \{p\}$ $\forall p, q \in Q$ $\forall a \in \Sigma$ $\forall R, P \in 2^Q$.
$F' = \{(p, F) \mid F \in \mathcal{F}\}$.
Then for $\alpha \in \Sigma^\omega$: $\mathcal{A}$ STAIGNER-WAGNER accepts α
$\Leftrightarrow \exists$ infinite run ρ_α of $\mathcal{A}$ on α such that $\text{Occ}(\rho_\alpha) \in \mathcal{F}$.
$\Leftrightarrow \exists$ infinite run ρ'_α of $\mathcal{B}$ on α such that from some point onwards only states $(*, P)$ for some $P \in \mathcal{F}$ are visited.
$\Leftrightarrow \mathcal{B}$ co-BÜCHI accepts α. □

(c) Every nondeterministically co-BÜCHI recognizable language is also nondeterministically E-recognizable.

Proof: Let $\mathcal{A} = (Q, \Sigma, q_0, \Delta, F)$ be a nondeterministic co-BÜCHI automaton. We construct $\mathcal{B} = (Q', \Sigma, q'_0, \Delta', F')$ E-automaton as follows.
$Q' := Q \cup (\{1\} \times F)$, $\quad F' := \{1\} \times F$, $\quad g'_0 := q_0$,
$\Delta' := \Delta \cup \{(p, a, (1, q)) \mid (p, a, q) \in \Delta, q \in F\} \cup \{((1, p), a, (1, q)) \mid (p, a, q) \in \Delta, q, q \in F\}$.
Then $\mathcal{A}$ co-BÜCHI accepts $\alpha \in \Sigma^\omega$
$\Leftrightarrow \exists$ infinite run ρ_α such that only states from F are seen.
$\Leftrightarrow \exists$ run ρ'_α such that finally only states from $\{1\} \times F = F'$ are seen.
$\Leftrightarrow \mathcal{B}$ E-accepts α. □

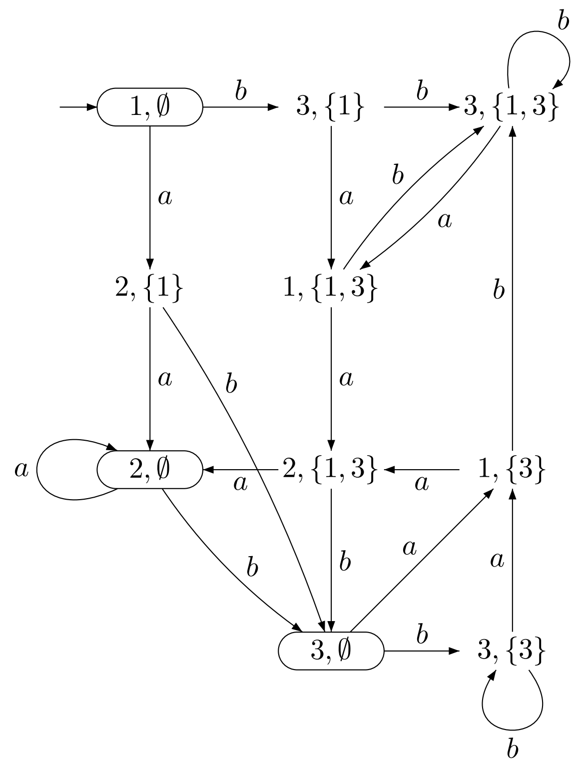
7.2 Exercise 13

Let L be the language recognized by the given automaton. Apply LANDWEBER's theorem:

(a) L is deterministically E-recognizable iff $\mathcal{F}$ is closed under reachable loops. $\Rightarrow$ L is not deterministically E-recognizable.

(b) L is deterministically Büchi recognizable iff $\mathcal{F}$ is closed under superloops. $\Rightarrow$ L is deterministically Büchi recognizable.

algorithmic solution:

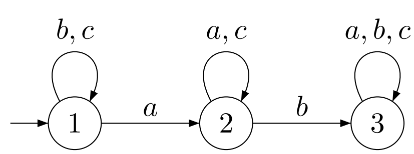


8 Exercise from 12-07-2005

8.1 Exercise 14

$L = \{\alpha \in \Sigma^\omega \mid \text{ if } a \text{ occurs in } \alpha \text{ then } b \text{ occurs later on}\}$.

(a) Staigner-Wagner-automaton for L with $\mathcal{F} = \{\{1\}, \{1, 2, 3\}\}$:

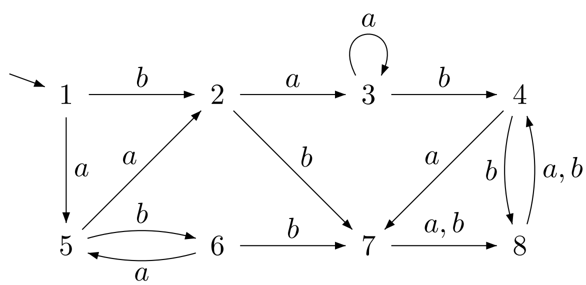


(b) Tarjan's SCC (strongly connected components) algorithm on graph G:

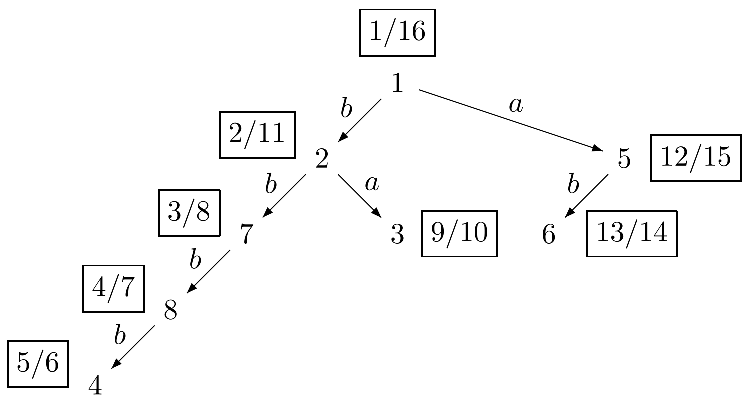
1. Do DFS (depth first search) through G and remember enter/farewell times.
2. Reverse edges of graph G. $\rightarrow \overline{G}$

3. Do DFS on $\overline{G}$ starting from vertex with highest farewell.
   The reachable vertices form a SCC S of G.
4. Repeat step 3 without S.

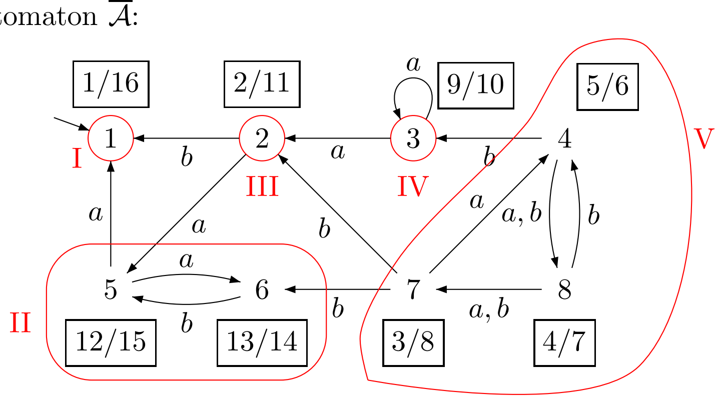
Given Automaton $\mathcal{A}$:



DFS through $\mathcal{A}$:



Notation $\boxed{E/F}$ is used, whereas E indicates the step number at the entering into the current state (”enter”) and F the step number while leaving the current state (”farewell”).
Edge-reversed Automaton $\overline{\mathcal{A}}$:



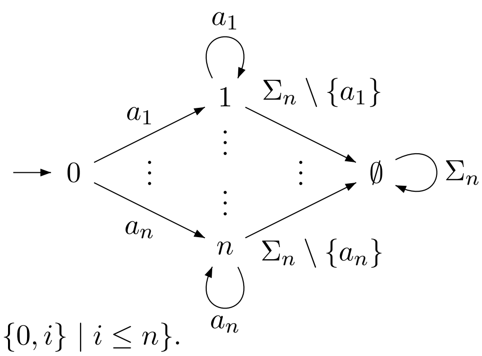
The red marked components are the SCCs.

8.2 Exercise 15

For $n \geq 1$ there exists $L_n := a_1^\omega + \cdots + a_n^\omega$ (with $\Sigma_n = \{a_1, \ldots, a_n\}$, which is accepted by a Staigner-Wagner automaton with n accepting state sets but not by one with $n-1$.

Proof:

1. L is accepted by



with acceptance sets $\mathcal{F} = \{\{0, i\} \mid i \leq n\}$.

2. L_n cannot be accepted by Staigner-Wagner automaton with only $n-1$ acceptance sets. Assume there is such an automaton $\mathcal{A}$. There exist $i \neq j$ and runs ρ of $\mathcal{A}$ on a_i^ω and ρ' of $\mathcal{A}$ on a_j^ω such that $\mathrm{Occ}(\rho) = \mathrm{Occ}(\rho') \in \mathcal{F}$. Let m minimal such that $\mathrm{Occ}(\rho[0 \ldots m]) = \mathrm{Occ}(\rho)$.
Let $\rho(m) = q \in \mathrm{Occ}(\rho')$.
Choose suffix ρ'' of ρ' starting with q, then $\rho(0) \ldots \rho(m-1)\rho''$ is a run on $a_i^{m-1} a_j^\omega$ with same occurrence set. Contradiction. □

8.3 Exercise 16

Surprise: **A nondeterministic Staigner-Wagner automaton cannot be determinized!**

Proof:

1. Every nondeterministic Staigner-Wagner automaton can be completed by creating a new sink state to which all missing edges are connected.

2. Assume Staigner-Wagner automaton can be determinized.
Then the following class inclusions emerge:

$$\text{ndet-co-Büchi} \overset{\text{Exercise 12}}{=} \text{ndet-SW} = \text{det-SW} \subsetneq \text{det-co-Büchi}.$$

But this is a contradiction to the class hierarchy! □

9 Exercise from 12-14-2005

9.1 Example: S1S definable property

- property: P_1 holds from some odd position onwards, e.g.

$$\begin{array}{cccccccccccc} 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & \ldots \\ \uparrow & & & & & & & \uparrow & & & \\ 0 & & & & & & & 7 & & & \end{array}$$

- formula (intuitive): $\phi(X_1) = \exists t(t \text{ odd } \wedge \forall s \geq t : X_1(s))$
  How to formulate "odd"? Answer: *second order logic.*

$$\begin{array}{ccccccccc} 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & \dots \\ 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & \dots \end{array}$$

  Replace "t odd" by "$\exists Y(\neg Y(0) \wedge \forall t(Y(t) \leftrightarrow Y(T')))$". The resulting formula would then be

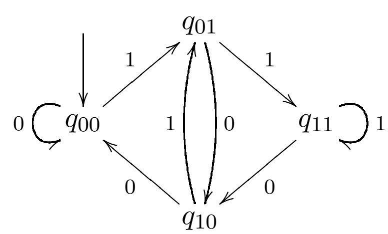
$$\exists Y[(\neg Y(0) \wedge \forall t(Y(t) \leftrightarrow Y(T'))) \wedge \exists t(Y(t) \wedge \forall s \geq t : X_1(s))].$$

- *Notation:* The sequence

$$\left(\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \right) \text{ is written as } \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}.$$

9.2 Exercise 17

(a) $L = \{\alpha \in \{0,1\}^\omega \mid \alpha$ contains 00 infinitely often, but 11 only finitely often$\}$. Deterministic Rabin automaton (thanks to Klaus)



remembers every last two read letters. $\Omega = \{(\{q_{11}\}, \{q_{00}\})\}$.

(b) L recognized by a Rabin automaton $\mathcal{A}$ with $\Omega = \{(Q_1, F_1), \dots, (E_k, F_k)\}$ and $E_i = \emptyset \; \forall 1 \leq i \leq k$.
By definition α is accepted by $\mathcal{A}$ iff $\text{Inf}(\rho_\alpha) \cap F_i \neq \emptyset$ and $E_i \cap \text{Inf}(\rho_\alpha) = \emptyset$ (which is always satisfied in this situation) for some i.
If we choose $F = \bigcup_{i=1}^{k} F_i$ then $\mathcal{A}$ with F as a Büchi automaton accepts L.

9.3 Exercise 18

Tool: `http://www-i7.informatik.rwth-aachen.de/d/research/omegadet.html`
Short description of the Muller-Schupp - algorithm to calculate an update step of the deterministic Muller automaton from an initial tree t on the input symbol $a \in \Sigma$:

1. Copy t, replace green by yellow.
2. (a) Delete state sets P of each leaf.
   (b) Introduce sons labeled with $P' := \{q \mid \exists p \in P : (p, a, q) \in \Delta\}$.

(c) Delete all states which occur also more to the left proceeding from right to left.

(d) Split any set into its final and non-final states producing a left son labeled green and a right son labeled red respectively, each named with a free node name.

3. Delete all nodes which did not get a new non-empty (and hence named) descendant.

4. Compress path segments into their top node, giving it color green if merged with a path segment containing a green or yellow node.

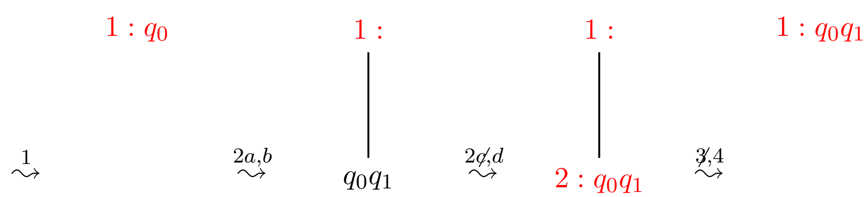
Muller-Schupp - construction (for better readability yellow is set to blue):

Given automaton $\mathcal{A}$ on the input word $\alpha = ababb(ab)^\omega$.

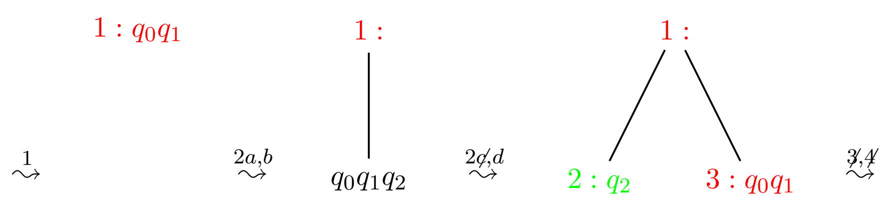
- initial tree:

$$1 : q_0$$

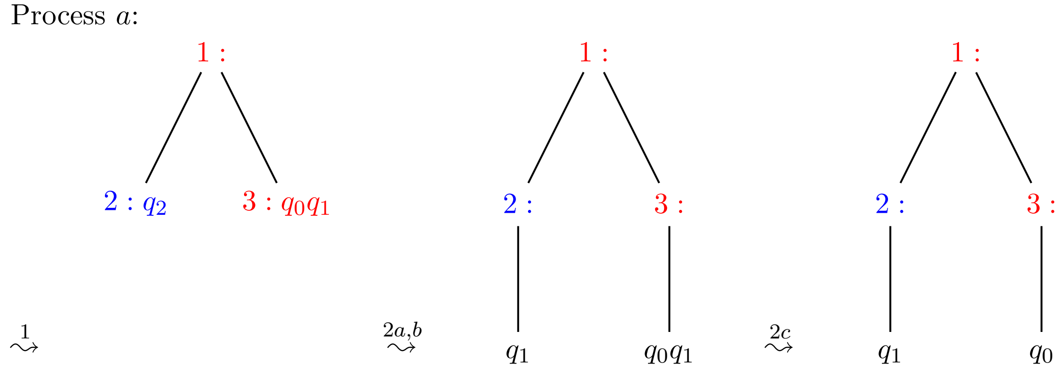
- Process a:

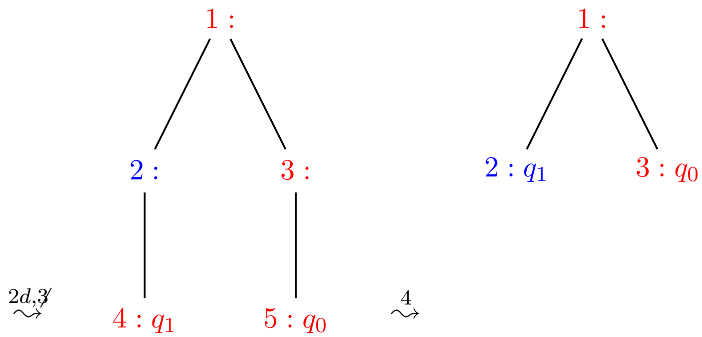


- Process b:

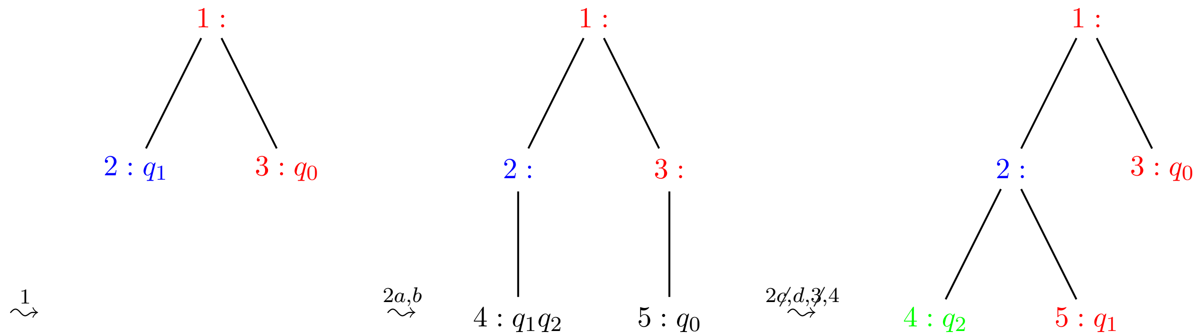


- Process a:

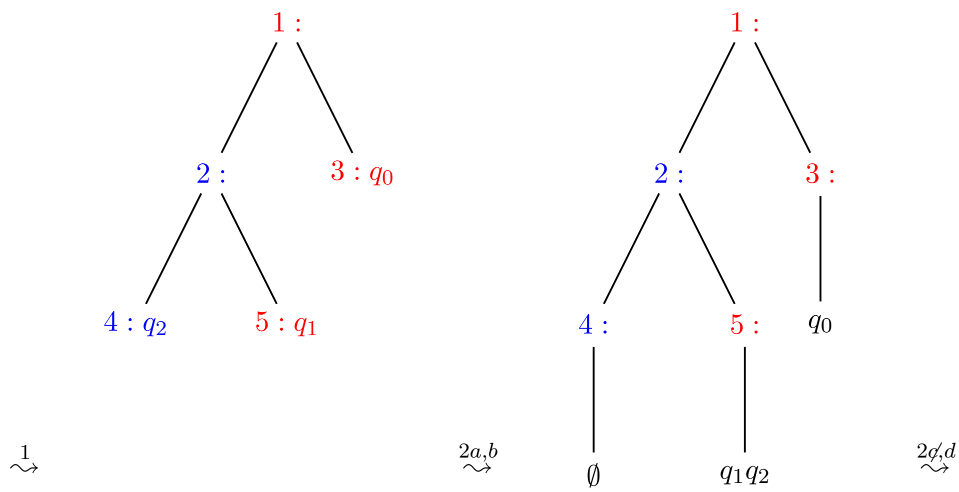


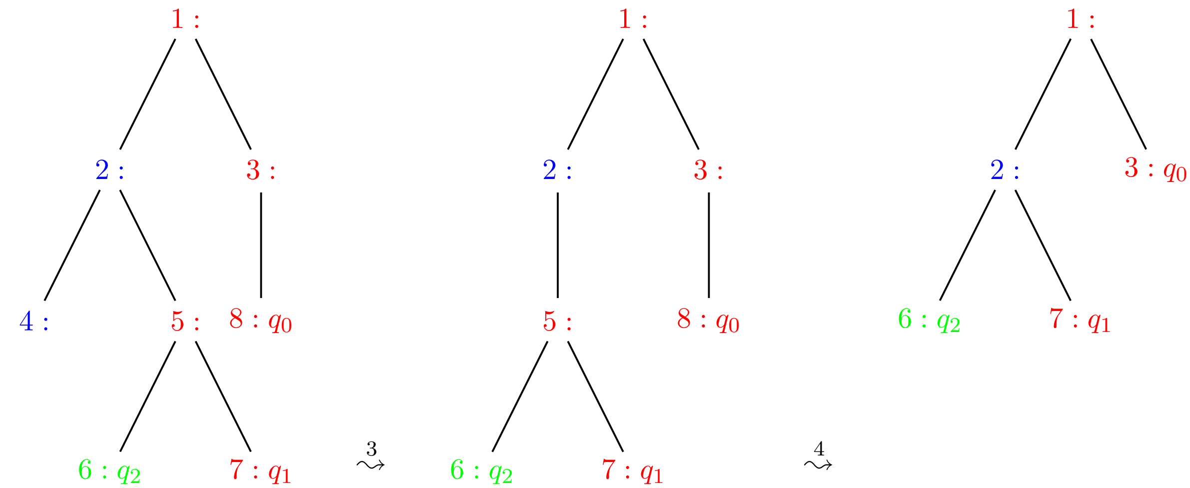


- Process b:

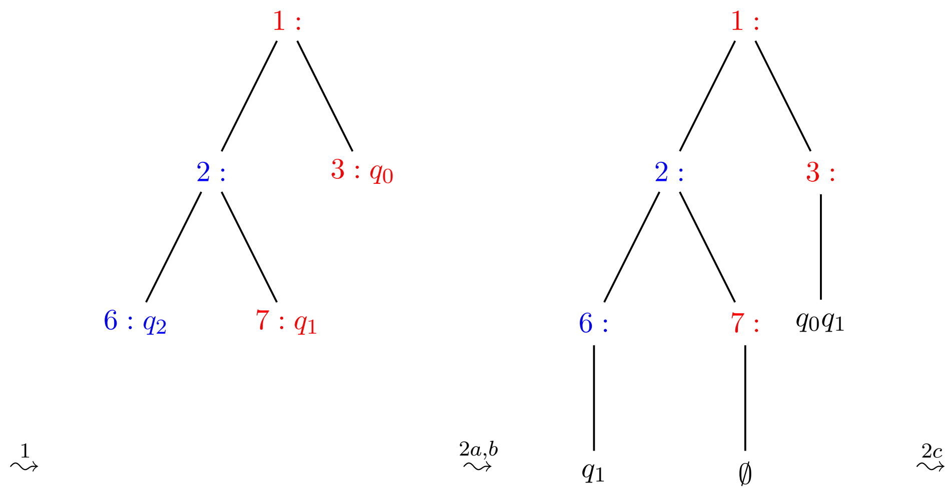


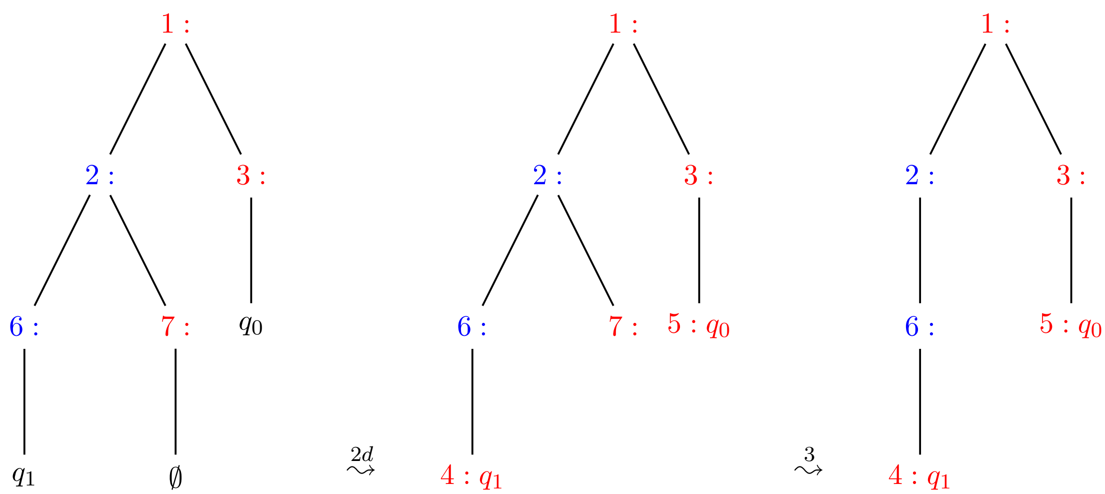
- Process b:

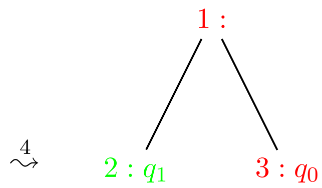



1 :
2 :
3 :
4 :
5 :
8 : q_0
6 : q_2
7 : q_1
$\overset{3}{\rightsquigarrow}$
1 :
2 :
3 :
5 :
8 : q_0
6 : q_2
7 : q_1
$\overset{4}{\rightsquigarrow}$
1 :
2 :
3 : q_0
6 : q_2
7 : q_1

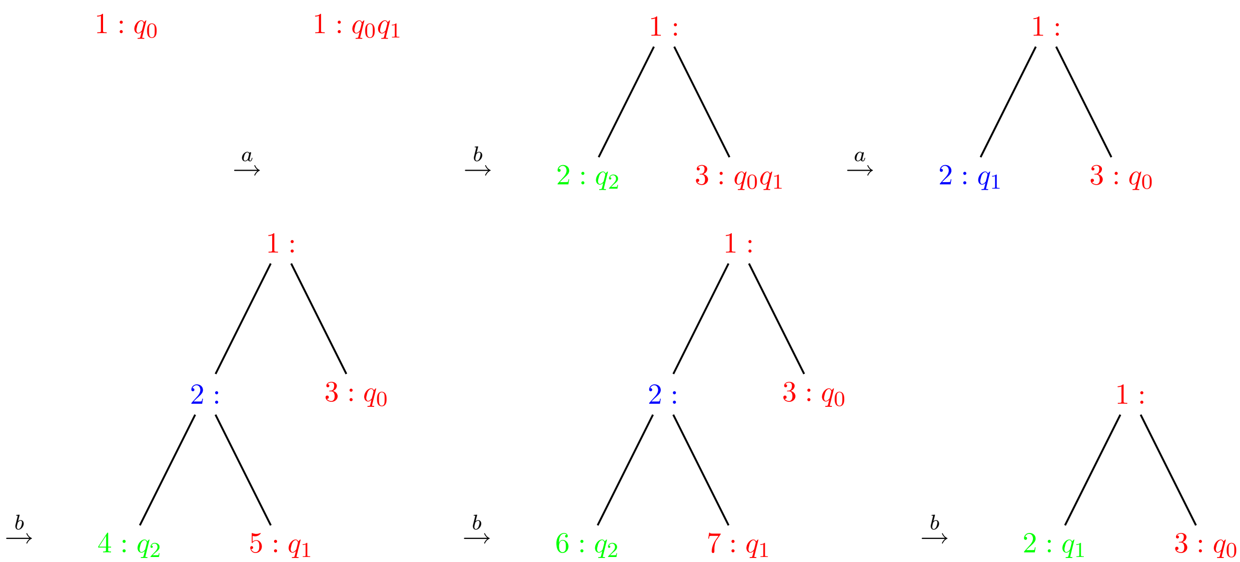
- Process a:


$\overset{1}{\rightsquigarrow}$
1 :
2 :
3 : q_0
6 : q_2
7 : q_1
$\overset{2a,b}{\rightsquigarrow}$
1 :
2 :
3 :
6 :
7 :
q_0q_1
q_1
$\emptyset$
$\overset{2c}{\rightsquigarrow}$


1 :
2 :
3 :
6 :
7 :
q_0
q_1
$\emptyset$
$\overset{2d}{\rightsquigarrow}$
1 :
2 :
3 :
6 :
7 :
5 : q_0
4 : q_1
$\overset{3}{\rightsquigarrow}$
1 :
2 :
3 :
6 :
5 : q_0
4 : q_1



Result: The states of the deterministic RABIN automaton are the last MULLER-SHUPP trees in each processing step. So the run of the resulting deterministic automaton is:



Note: There are other constructions like the SAFRA construction which are more complex but yield better results in the tool.

10 Exercise from 12-21-2005

10.1 Exercise 19

Given $\Sigma_n = \{\#, 1, \dots, n\}$ and nondeterministic BÜCHI automaton $\mathcal{A}_n$ with $L(\mathcal{A}) =: L_n$.

(a) Characterization by sequences.

$$\alpha \in L_n \;\Leftrightarrow\; \exists \text{ a sequence } i_1 i_2 \; i_2 i_3 \; \dots \; i_{k-1} i_k \; i_k i_1, \quad i, j \in \Sigma_n \setminus \{\#\},$$

which is repeated infinitely often and starts with $\#$.
The deterministic MULLER automaton checks wether such a sequence exists.
Choose $Q = \Sigma_n \times \Sigma_n \cup \{q_0, q_s\}$. Δ consists of rules

- (ab, c, bc), $a, b, c \in \Sigma_n$,
- $(q_0, \#, 11)$, (q_0, a, q_s), $a \in \Sigma_n \setminus \{\#\}$,
- (q_s, a, q_s), $a \in \Sigma_n$.

For $\mathcal{F}$ we add $F \subseteq Q$ to $\mathcal{F}$ iff F contains a characteristic sequence.

Note: The relation Δ also defines a transition function here.

(b) Recall LANDWEBER's theorem:
Let L be recognized by a deterministic MULLER automaton with accepting component $\mathcal{F}$, then:
L is recognized by a deterministic BÜCHI automaton iff $\mathcal{F}$ is closed under superloops.
$\Rightarrow L_n$ is deterministically BÜCHI recognizable.

10.2 Exercise 20

(a) $L_1 := \binom{1}{1}\binom{1}{0}^*\left(\begin{smallmatrix}1&0\\1&0\end{smallmatrix}\right)^\omega$.

$$\begin{aligned}\varphi_1(X_1,X_2) \quad = \quad & X_1(0) \wedge X_2(0) && (1)\\ & \wedge \exists t\big(t>0 \wedge X_1(t) \wedge X_2(t) && (2)\\ & \wedge \forall s(0<s<t \to X_1(s) \wedge \neg X_2(s)) && (3)\\ & \wedge \forall s(s \geq t \to \underbrace{((X_1(s) \leftrightarrow \neg X_2(s')) \wedge (X_2(s) \leftrightarrow \neg X_2(s'))))}_{(*)}\big). && (4)\end{aligned}$$

Comment:

(1): Describes: First letter is $\binom{1}{1}$.

(2): $\binom{1}{1}$ appears again at position $t>0$.

(3): In-between positions 0 and t only $\binom{1}{0}$ occurs.

(4): After t $\binom{1}{1}$ and $\binom{0}{0}$ alternate.

Remark: $(*)$ cannot be replaced by

$$(X_1(s) \wedge X_2(s)) \leftrightarrow (\neg X_1(s') \wedge \neg X_2(s'))$$

Counter-example:

$$\left(\begin{smallmatrix}1&0&1&1&\cdots\\1&0&0&0&\cdots\end{smallmatrix}\right)$$

satisfies the formula because the implication at the second position always is true.

(b) $L_2 := \left(\begin{smallmatrix}1&1&1\\1&1&1\end{smallmatrix}\right)^*\binom{0}{1}^\omega$.

$$\begin{aligned}\varphi_2(X_1,X_2) \quad = \quad & \exists Y\big(Y(0) \wedge Y(0') \wedge Y(0'') \wedge \forall t(Y(t) \leftrightarrow \neg Y(t') \wedge \neg Y(t'')) && (5)\\ & \wedge \exists s(Y(s) \wedge \forall t<s(X_1(t) \wedge X_2(t)) \wedge \forall t \geq s(\neg X_1(t) \wedge X_2(t)))\big) && (6)\end{aligned}$$

Comment:

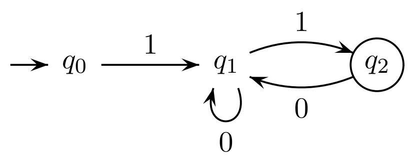
(5): Y is second order variable representing a $\bmod 3$ counter.

(6): $t \bmod 3 = 0$ and: before t only $\binom{1}{1}$ and starting at t only $\binom{0}{1}$ occurs.

11 Exercise from 01-11-2006

11.1 Exercise 21

Given BÜCHI automaton $\mathcal{A}$:



Example situation:

$\mathbb{N}$	0	1	2	3	4	$\cdots$	
X	1	0	0	1	1	$\cdots$	$[X = \{0, 3, 4, \ldots\}]$
Y_0	1	0	0	0	0	$\cdots$	$[Y_0 = \{0, \ldots\}]$
Y_1	0	1	1	1	0	$\cdots$	$[Y_1 = \{1, 2, 3, \ldots\}]$
Y_2	0	0	0	0	1	$\cdots$	$[Y_2 = \{4, \ldots\}]$

S1S-formula for $\mathcal{A}$:

$$\begin{aligned}
\varphi(X) \quad = \quad & \exists Y_0 \exists Y_1 \exists Y_2 [\text{Partition}(Y_0, Y_1, Y_2) \wedge \underbrace{Y_0(0)}_{q_0 \text{ initial state}} \\
& \wedge \forall t \big((Y_0(t) \wedge X(t) \wedge Y_1(t')) \vee \\
& \qquad (Y_1(t) \wedge \neg X(t) \wedge Y_1(t')) \vee \\
& \qquad (Y_1(t) \wedge X(t) \wedge Y_2(t')) \vee \\
& \qquad (Y_2(t) \wedge \neg X(t) \wedge Y_1(t')) \big) \\
& \wedge \forall t \exists s (t < s \wedge Y_s(s))],
\end{aligned}$$

$$\begin{aligned}
\text{Partition}(Y_0, Y_1, Y_2) \quad := \quad & \forall t \big((Y_0(t) \vee Y_1(t) \vee Y_2(t)) \\
& \wedge \neg((Y_0(t) \wedge Y_1(t)) \vee (Y_0(t) \wedge Y_2(t)) \vee (Y_1(t) \wedge Y_2(t))) \big).
\end{aligned}$$

11.2 Exercise 22

(a) 4 states:

$$\begin{array}{cc c ccccc}
\text{unary:} & & & \text{binary:} & & & & \\
Y_0 & 0 & & Z_0 & 1 & 0 & 0 & 1 \\
Y_1 & 1 & \rightsquigarrow & Z_1 & 0 & 0 & 1 & 1 \\
Y_2 & 0 & & & \wr & \wr & \wr & \wr \\
Y_3 & 0 & & & q_1 & q_0 & q_2 & q_3
\end{array}$$

Let $\mathcal{A} = (Q, \{0,1\}, q_0, \Delta, F)$, $Q = \{q_0, \dots, q_7\}$.
For $0 \le i \le 7$: $(i_1, i_2, i_3)_b = i \quad [i = i_2 \cdot 2^2 + i_1 \cdot 2^1 + i_0 \cdot 2^0]$

$$\psi_i(Y_0, Y_1, Y_2, t) \quad := \quad \left\{ \begin{array}{rl} Y_2(t), & i_2 = 1 \\ \neg Y_2(t), & i_2 = 0 \end{array} \right\} \wedge \left\{ \begin{array}{rl} Y_1(t), & i_1 = 1 \\ \neg Y_1(t), & i_1 = 0 \end{array} \right\} \wedge \left\{ \begin{array}{rl} Y_0(t), & i_0 = 1 \\ \neg Y_0(t), & i_0 = 0 \end{array} \right\}.$$

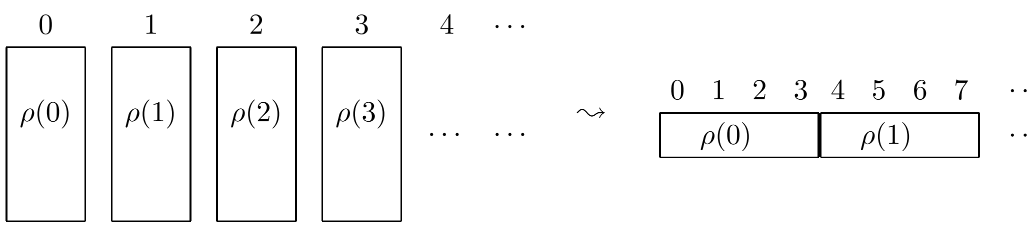
e.g.: $\psi_3(Y_0, Y_1, Y_2, t) = \neg Y_2(t) \wedge Y_2(t) \wedge Y_0(t) \quad [3 = (011)_b]$

$$\delta_{(i,b,j)}(Y_0, Y_1, Y_2, t) \quad := \quad \left\{ \begin{array}{rl} \psi_i(t) \wedge \neg X(t) \wedge \psi_j(t'), & b = 0 \\ \psi_i(t) \wedge X(t) \wedge \psi_j(t'), & b = 1 \end{array} \right\}$$

i.e. $q_i \xrightarrow{b} q_j \quad \sim \quad \delta_{(i,b,j)}$.

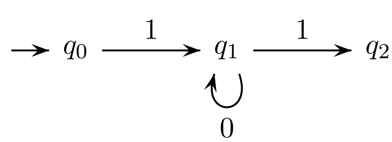
$$\varphi(X) := \exists Y_0 \exists Y_1 \exists Y_2 \left[\psi_0(0) \wedge \forall t \left(\bigvee_{(q_i,b,q_j) \in \Delta} \delta_{(i,b,j)}(Y_0, Y_1, Y_2, t) \right) \wedge \forall t \exists s \left(t < s \wedge \bigvee_{q_i \in F} \psi_i(s) \right) \right].$$

(b*) – First idea: Characterise a position $\rho(t)$ of run ρ not by a vector X with $|Q|$ components but by the transposed one. Then the position of $X(t)$ can be calculated as $t \cdot k$, thus $X(t) \sim Y(t \cdot k)$.

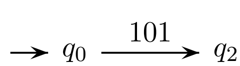


But this of course requires a 2nd set variable!

– Better idea:



corresponds to



1. Successful run: It suffices to consider $n \cdot k$ states for fixed length k.
2. Flag wether we have seen at least one final state in between.
3. Mark beginning of encoding.

12 Exercise from 01-18-2006

12.1 Exercise 23

$\mathrm{S1S}_0$:

- Eliminate 0, $<$.
- Successor $'$ as $\mathrm{Succ}(X,Y)$ what especially means that X and Y are singletons.
- Eliminate FO-variables, use $X \subseteq Y$, $\mathrm{Sing}(X)$.

(a) Given $\varphi(X) = \exists t\big(\neg X(t) \rightarrow X(t')\big)$, the corresponding $\mathrm{S1S}_0$-formula in prenex normal form results in

$$\psi(X) = \exists T \exists S(\underbrace{\mathrm{Succ}(T,S)}_{T,S \text{ singletons and in successor closure}} \wedge \underbrace{(T \subseteq X \vee S \subseteq X)}_{\neg p \rightarrow q \equiv p \vee q}).$$

Note: $T \subseteq X : \{t\} \subseteq X \Rightarrow t \in X$.

(b) Find Büchi automaton $\mathcal{A}$ for $\mathrm{Succ}(T,S) \wedge (T \subseteq X \vee S \subseteq X)$. Notation:

$$\begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} \begin{matrix} \sim X \\ \sim T \\ \sim S \end{matrix} .$$

1. T,S are singletons:

   T :

   $$\ldots \begin{pmatrix} * \\ 0 \\ * \end{pmatrix} \ldots \begin{pmatrix} * \\ 0 \\ * \end{pmatrix} \begin{pmatrix} * \\ 1 \\ * \end{pmatrix} \begin{pmatrix} * \\ 0 \\ * \end{pmatrix} \ldots \qquad \underbrace{\ldots \begin{pmatrix} * \\ 1 \\ * \end{pmatrix}}_{\text{not allowed, because then two positions belong to } T}$$

   S :

   $$\ldots \begin{pmatrix} * \\ * \\ 0 \end{pmatrix} \ldots \begin{pmatrix} * \\ * \\ 0 \end{pmatrix} \begin{pmatrix} * \\ * \\ 1 \end{pmatrix} \begin{pmatrix} * \\ * \\ 0 \end{pmatrix} \ldots$$

2. S successor of T:

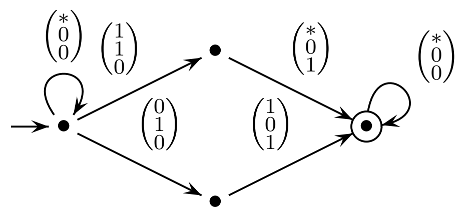
   $$\ldots \begin{pmatrix} * \\ 0 \\ 0 \end{pmatrix} \ldots \begin{pmatrix} * \\ 0 \\ 0 \end{pmatrix} \underbrace{\begin{pmatrix} * \\ 1 \\ 0 \end{pmatrix} \begin{pmatrix} * \\ 0 \\ 1 \end{pmatrix}}_{(*)} \begin{pmatrix} * \\ 0 \\ 0 \end{pmatrix} \ldots$$

   Note: This formula already contains that T,S are singletons.

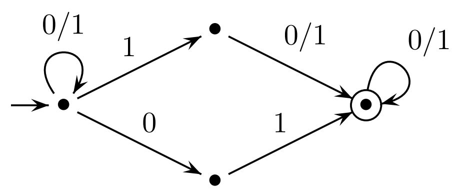
3. $T \subseteq X$: Replace $(*)$ by

$$\begin{pmatrix}1\\1\\0\end{pmatrix}\begin{pmatrix}*\\0\\1\end{pmatrix} \quad \text{or} \quad \begin{pmatrix}0\\1\\0\end{pmatrix}\begin{pmatrix}1\\0\\1\end{pmatrix}$$

4. $\Rightarrow$ automaton for quantifier-free formula:



5. Quantifiers: "$\exists S$" results in a projection of transition labels onto first two components (i.e. delete third components), for "$\exists T$" delete second components.
   $\Rightarrow$ complete automaton:



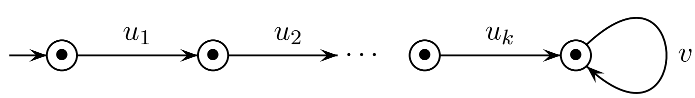
$\Rightarrow L(\mathcal{A}_\varphi) = \{0,1\}^\omega \setminus \{0^\omega\}$. $\square$

12.2 Exercise 24

Ultimately periodic word $\alpha \in \{0,1\}^\omega$:

$$\begin{aligned} \alpha &= u \cdot v^\omega \text{ for some } u, v \in \{0,1\}^* \text{ (finite)} \\ &= u \cdot v \cdot v \cdot v \cdot v \cdots \end{aligned}$$

(a) Define Büchi automaton $\mathcal{A}_\alpha$ for $u = u_1 u_2 \dots u_k$:



1. $L(\mathcal{A}) = \{\alpha\}$.
2. Number of states $|Q_\alpha| = |u| + |v|$.

$\Rightarrow \mathcal{A}$ accepts α iff $L(\mathcal{A}) \cap L(\mathcal{A}_\alpha) = L(\mathcal{A}) \cap \{\alpha\} \neq \emptyset$.
Properties:

(1) The intersection of Büchi automata is effective (there is a Büchi automaton $\mathcal{A}'$ with $L(\mathcal{A}') = L(\mathcal{A}) \cap L(\mathcal{A}_\alpha)$).

(2) The emptiness problem for Büchi automata is decidable.

Reduction to a decidable problem $\Rightarrow$ the original problem is also decidable.

(b) Find a *tight* upper bound.

- Measure for α: $|u| + |v| =: n$.
- Measure for $\mathcal{A} = (Q, \Sigma, q_0, \Delta, F)$: $|Q| + |\Delta| =: m$ (usual graph size).

(1) *intersection:* by construction of the lecture

$$\mathcal{A}' := (\underbrace{Q \times Q_\alpha \times \{1,2,3\}}_{\#=3\cdot n\cdot|Q|}, \ldots, \Delta', F')$$

with $|\Delta'| \leq |\Delta|$ ($Q' = Q \times Q_\alpha \times \{1,2\}$ also possible).

(2) *emptiness:* by Tarjan's algorithm in time $\mathcal{O}(|V| + |E|)$ for graph (V, E). For automaton $\mathcal{A}' = (Q', \ldots, \Delta', F')$ in time $\mathcal{O}(|Q'| + |\Delta'|)$

$\Rightarrow$ original problem can be solved in time

$$\mathcal{O}(3 \cdot n \cdot |Q| + |\Delta|) = \mathcal{O}(n \cdot m).$$

13 Exercise from 01-25-2006

13.1 Exercise 25

$$\begin{array}{llll} (\omega+\omega)\text{-word } \alpha^\frown\beta: & \alpha(0)\alpha(1)\alpha(2)\ldots & \ldots & \beta(0)\beta(1)\beta(2)\ldots \\ & \mathrm{Inf}(\rho_\alpha) & \xrightarrow{\Delta'} & p \in Q \end{array}$$

Emptiness decision procedure: Let $\mathcal{A} = (Q, \Sigma, q_0\Delta, \Delta', F)$.

1. Collect all $q \in Q$ with $(P, q) \in \Delta'$ for some $P \subseteq Q$. Set $Q' := \{q \in Q \mid \exists P \subseteq Q : (P, q) \in \Delta'\}$.
2. Check the Büchi ω-automaton $\mathcal{A}_q = (Q, \Sigma, q, \Delta, F)$ for emptiness $\forall q \in Q$.
3. Collect all sets $P \subseteq Q$ such that there is $q \in Q$ with $(P, q) \in \Delta'$ and $L(\mathcal{A}_q) \neq \emptyset$. We obtain family $\mathcal{F} := \{P \subseteq Q \mid \exists q \in Q : (P, q) \in \Delta' \wedge L(\mathcal{A}_q) \neq \emptyset\}$.
4. Check the nondeterministic Muller automaton $\mathcal{A}' = (Q, \Sigma, q_0, \Delta, \mathcal{F})$ for emptiness.

Claim: $L(\mathcal{A}) \neq \emptyset \Leftrightarrow L(\mathcal{A}') \neq \emptyset$.
Proof: "$\Leftarrow$": Let $\alpha \in L(\mathcal{A}')$ be an accepting run of $\mathcal{A}'$ on α, in particular: $\mathrm{Inf}(\rho_\alpha) \in \mathcal{F}$.
By definition exists $q \in Q$ such that $(\mathrm{Inf}(\rho_\alpha), q) \in \Delta'$ and $L(\mathcal{A}_q) \neq \emptyset$.
Let $\beta \in L(\mathcal{A}_q)$ and ρ_β be an accepting run of $\mathcal{A}_q$ on β.
$\Rightarrow \mathrm{Inf}(\rho_\beta) \cap F \neq \emptyset$, $\rho_\beta(0) = q$.

$\Rightarrow (\rho_\alpha, \rho_\beta)$ is an accepting run of $\mathcal{A}$ on $\alpha^\frown\beta$.
"$\Rightarrow$": Analogously: $(\rho_\alpha, \rho_\beta)$ on $\alpha^\frown\beta \Rightarrow \ldots$ □

Remark: Emptiness-check for nondeterministic MULLER automata:

- *1st way:* nondeterministic MULLER automaton → nondeterministic BÜCHI automaton.

  1. BÜCHI automaton guesses $F \in \mathcal{F}$.
  2. BÜCHI automaton guesses the position from which on only F-states.
  3. Ensures that all F-states are visited and no other state.

- *2nd way:* direct procedure.

  1. Remove non-reachable states and remove $F \in \mathcal{F}$ that contains non-reachable state.
  2. For each $F \in \mathcal{F}$: Restrict $\mathcal{A}$ on F. Check if new $\mathcal{A}_F$ is strongly connected.
  3. Yes, if $\mathcal{A}_f$ is strongly connected for some $F \in \mathcal{F}$.
     No, if $\mathcal{A}_f$ is not strongly connected for all $F \in \mathcal{F}$.

13.2 Exercise 26

To obtain the desired form of $\varphi(X)$ we consider two possible ways.

- *1st way:*

  1. Translate $\varphi(X)$ into equivalent BÜCHI automaton $\mathcal{A}$.
  2. Find complement automaton $\overline{\mathcal{A}}$ with $L(\overline{\mathcal{A}}) = \{0,1\}^\omega \setminus L(\mathcal{A})$.
  3. Find S1S-formula $\overline{\varphi}(X)$ equivalent to $\overline{\mathcal{A}}$ of the form $\exists Y_1 \ldots \exists Y_n \overline{\psi}(Y_1, \ldots, Y_n, X)$ (standard construction).

  Claim: $\neg\overline{\varphi}(X)$ is the desired formula.
  Proof:

  - $\neg\overline{\varphi}(X) = \forall Y_1 \ldots \forall Y_n \psi(Y_1, \ldots, Y_n, X)$ with $\psi(Y_1, \ldots, Y_n, X) \equiv \neg\overline{\psi}(Y_1, \ldots, Y_n, X)$.
  - $\alpha \in \{0,1\}^\omega$, $\alpha \models \neg\overline{\varphi}(X) \Leftrightarrow$ not $\alpha \models \overline{\varphi}(X) \Leftrightarrow$ not $\alpha \in L(\overline{\mathcal{A}}) \overset{2.}{\Leftrightarrow} \alpha \in L(\mathcal{A}) \overset{1.}{\Leftrightarrow} \alpha \models \varphi(X)$.

  So

$$\neg\overline{\varphi}(X) \equiv \varphi(X) \quad \text{over } \omega \text{ words.}$$

- *2nd way:* Use the hint.

  1. As above.
  2. Transform BÜCHI automaton $\mathcal{A}$ into equivalent deterministic MULLER automaton $\mathcal{A}'$. (MCNAUGHTON or MULLER-SHUPP construction)

3. Find S1S-formula $\varphi'(X)$ equivalent to $\mathcal{A}'$ of the denied form.
   Muller automaton $\mathcal{A}' = (Q, \{0,1\}, q_1, \delta, \mathcal{F})$, $Q = \{q_1, \dots, q_n\}$.

$$\begin{aligned}
\varphi'(X) \quad = \quad & \forall Y_1 \dots \forall Y_n \Bigg(\text{Partition}(Y_1, \dots, Y_n) \wedge Y_1(0) \\
& \wedge \forall t \left[\bigwedge_{i=1}^{n} ((Y_i(t) \wedge \neg X(t) \rightarrow Y_{\delta(q_i,0)}(t')) \wedge (Y_i(t) \wedge X(t) \rightarrow Y_{\delta(q_i,1)}(t'))) \right] \\
& \Rightarrow \bigvee_{F \in \mathcal{F}} \left(\bigwedge_{q_i \in F} \exists^{\infty} t \; Y_i(t) \wedge \bigwedge_{q_j \notin F} \neg \exists^{\infty} t \; Y_j(t) \right) \Bigg),
\end{aligned}$$

   whereat $\exists^{\infty} t \; \chi(t) \equiv \forall s \exists t (s < t \wedge \chi(t))$.

14 Exercise from 02-01-2006

14.1 Exercise 27

φ-expansion up to position 10:

p_1	0	0	0	0	1	0	0	1	1	1	1	...
p_2	0	0	1	0	0	1	0	1	1	1	1	...
$\neg p_1$	1	1	1	1	0	1	1	0	0	0	0	...
$p_1 \wedge p_2$	0	0	0	0	0	0	0	1	1	1	1	...
$\neg p_1 U (p_1 \wedge p_2)$	0	0	0	0	0	1	1	1	1	1	1	...
$X p_2$	0	1	0	0	1	0	1	1	1	1	1	...
$X X p_2$	1	0	0	1	0	1	1	1	1	1	1	...
$(p_2 \wedge X X p_2)$	0	0	0	0	0	1	0	1	1	1	1	...
$(\neg p_1 U (p_1 \wedge p_2)) U (p_2 \wedge X X p_2)$	0	0	0	0	0	1	1	1	1	1	1	...

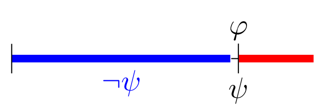
Hints:

- Sort subformulas by increasing complexity.
- If the right site of an until-formula is true at a position the formula itself directly is true. So the first step to evaluate the until-formula would be copying all 1s of the right-site-formula.

14.2 Exercise 28

(a) Two interpretations:

1. "ψ holds sometime"



$$\varphi N \psi \equiv (\neg \psi) U (\psi \wedge \varphi)$$

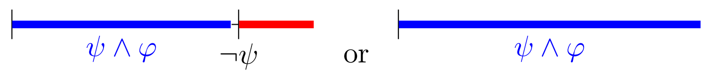
2. "if ψ holds at all then also φ"

$$\varphi N \psi \equiv G\neg\psi \vee (\neg\psi)U(\psi \wedge \varphi)$$

The operator N is not fixed because of the different possible interpretations.

(b) Two interpretations:

1. "φ and ψ from now on until the next $\neg\psi$"

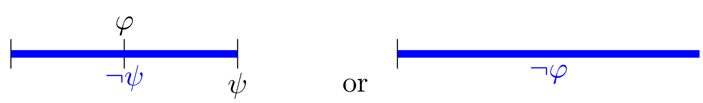


$$\varphi W \psi \equiv (\psi \wedge \varphi)U(\neg\psi) \quad \vee \quad G(\psi \wedge \varphi)$$

2. "always if ψ then φ"

$$\varphi W \psi \equiv G(\psi \rightarrow \varphi)$$

(c) Interpretation:



$$\varphi B \psi \equiv (\neg\psi)U(\varphi \wedge \neg\psi \wedge XF\psi) \quad \vee \quad G\neg\psi$$

14.3 Exercise 29

$\mathcal{A} = (Q, \Sigma, q_0, \Delta, F)$, $\Delta \subseteq Q \times \Sigma \times Q$
Δ: $\# = 2^8 = 512$, $\quad F \subseteq Q$: $\# = 4 \quad \Rightarrow$ about 2000 possible automata to consider!
Problem: All ω-words with suffix $\binom{1}{1}^\omega$ belong to $L(\varphi)$. Solution in three steps:

1. There is a Büchi automaton $\mathcal{A}$ with three states with the interpretations

   q_0: no current request,

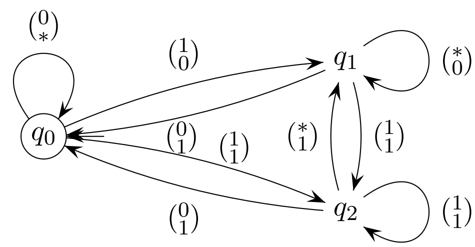
   q_1: wait for response,

   q_2: response and new request

   for the states and

   $$\begin{pmatrix} p_1 \\ p_2 \end{pmatrix} \quad \begin{matrix} \sim \text{request} \\ \sim \text{response} \end{matrix}$$

   for the transition labels. Form of the automaton:



2. *Claim:* There is no Büchi automaton with less than three states.

   (a) Since $L(\varphi) \neq \emptyset \Rightarrow$ each Büchi automaton needs at least one final state q.

   (b) Assume that there is a Büchi automaton $\mathcal{A}$ with one final state such that $L(\varphi) = L(\mathcal{A})$.

   - $\binom{0}{0}^\omega \in L(\mathcal{A})$: Let ρ be an accepting run of $\mathcal{A}$ on $\binom{0}{0}^\omega$.
   - $\binom{1}{1}^\omega \in L(\mathcal{A})$: Let ρ' be an accepting run of $\mathcal{A}$ on $\binom{1}{1}^\omega$.

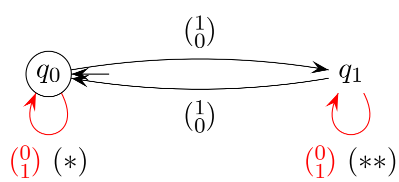
   $\Rightarrow$

   - ρ visits q infinitely often. Let m with $\rho(m) = q$.
   - ρ' visits q infinitely often. Let $m' > 0$ with $\rho'(m') = q$.

   $$\begin{array}{lllllllllll} \rho': & \rho'(0) & \xrightarrow{\binom{1}{1}} & \rho'(1) & \xrightarrow{\binom{1}{1}} & \cdots & \xrightarrow{\binom{1}{1}} & \rho'(m') & \xrightarrow{\binom{1}{1}} & \cdots & \\ \rho: & \rho(0) & \xrightarrow{\binom{0}{0}} & \rho(1) & \xrightarrow{\binom{0}{0}} & \cdots & \xrightarrow{\binom{0}{0}} & \rho(m) & \xrightarrow{\binom{0}{0}} & \rho(m+1) & \xrightarrow{\binom{0}{0}} \cdots \end{array}$$

   Consider new (red colored) run $\hat{\rho}$. $\hat{\rho}$ is accepting run on $\binom{1}{1}^{m'} \binom{0}{0}^\omega \notin L(\varphi)$. Contradiction.

3. Assume Büchi automaton $\mathcal{A} = (\{q_0, q_1\}, \ldots)$, $L(\mathcal{A}) = L(\varphi)$.
   $\Rightarrow q_0, q_1$ must be final, q_0 is the initial state.



   $(*)$: Transition not possible because $\binom{1}{0}\binom{0}{1}^\omega \in L(\varphi)$.
   $(**)$: Transition not possible because $\binom{1}{0}\binom{1}{0}\binom{0}{1}^\omega \in L(\varphi)$.
   But the resulting automaton also doesn't recognize $L(\varphi)$ because $\binom{1}{0}^\omega \notin L(\varphi)$. So the contradiction is complete.