Automata on Infinite Words

Automata: Theory and Practice

Paritosh K. Pandya

(TIFR, Mumbai, India)

Graduate Course Unversity of Trento

10-24 May 2005

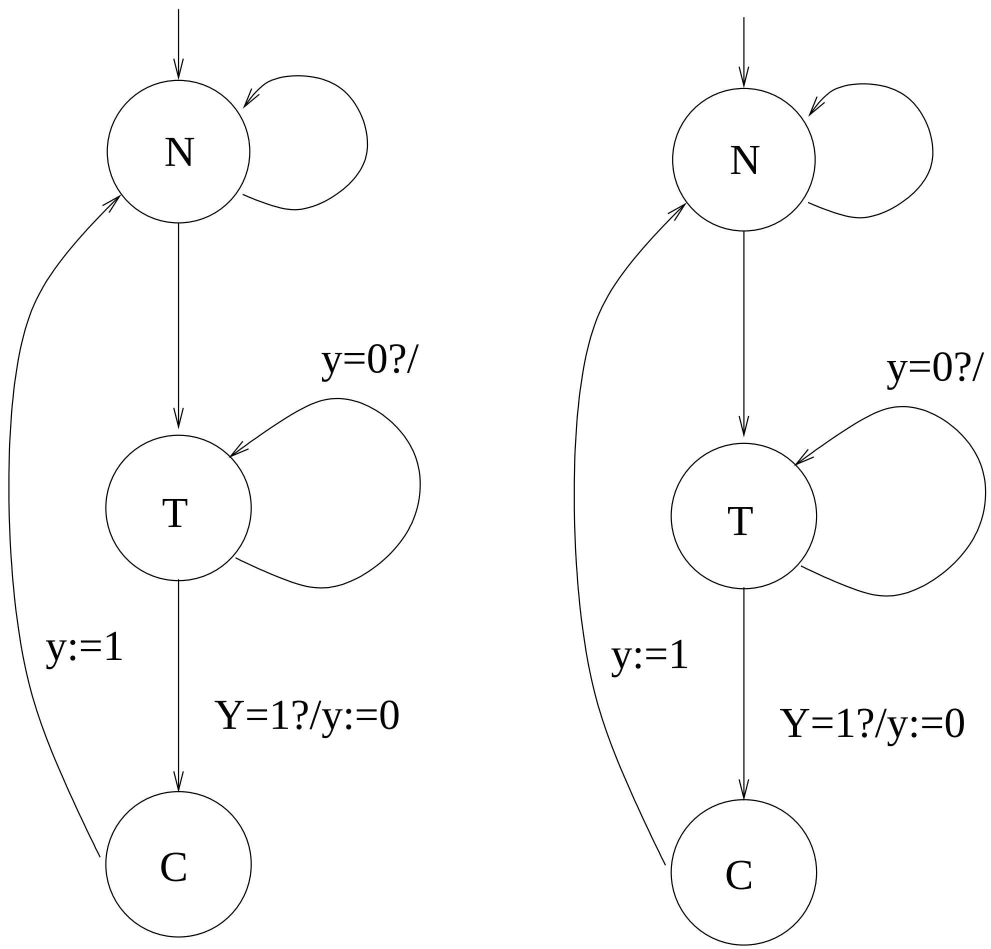
Modelling Infinite Behaviours

Reactive systems

- Control programs, circuits, operating systems, network protocols.
- Infinite computation involving multiple agents
- Nondeterminism and scheduling
- Fairness constraints

Mutual Exclusion Problem

Initialise $y := 1$



- Asynchronous parallelism
- Guarded assignments.

Properties

- Mutual exclusion: In any execution, the system will not reach a state where both processes are in critical region
- In any execution, process 1 will eventually enter critical region.

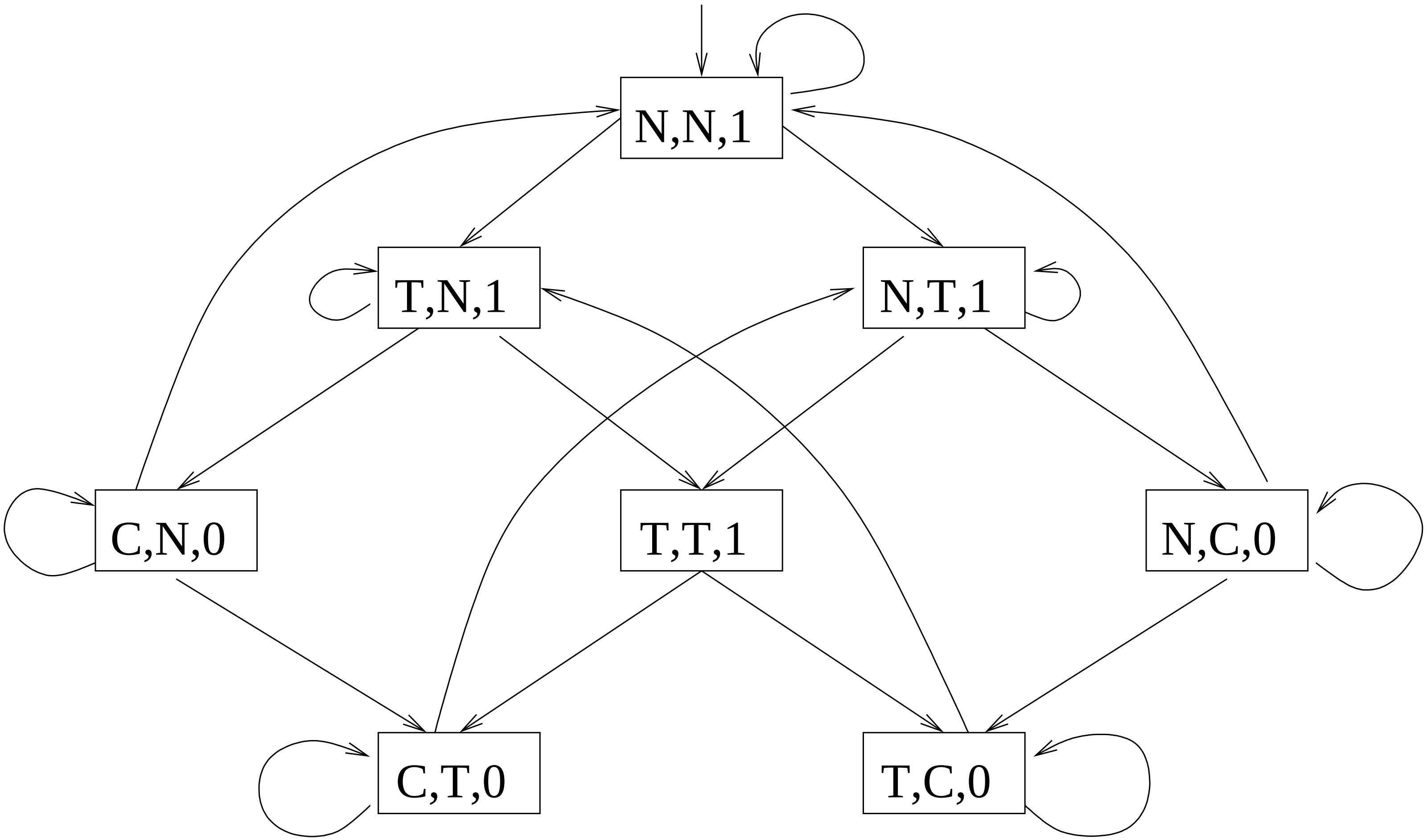
Global State: $\boxed{pc1, pc2, y}$

An Execution:

$\boxed{\text{N,N,1}} \rightarrow \boxed{\text{N,T,1}} \rightarrow \boxed{\text{N,T,1}} \rightarrow \boxed{\text{T,T,1}} \rightarrow \boxed{\text{T,C,0}} \rightarrow \boxed{\text{T,N,1}} \rightarrow \ldots$

Global transition system

Global State: $\boxed{pc1, pc2, y}$



Theory of omega Automata

Topics:

- Buchi Automata: Deterministic and Nondeterministic
- Omega Regular Expressions, Monadic Logic
- Muller Automata
- Rabin and Streett Automata
- Safra's Complementation Theorem (Optional)
- Omega Tree Automata and Rabin's Tree Theorem (Optional)

Infinite Word Languages

Modelling infinite computations of reactive systems.

- An ω-word α over Σ is infinite sequence

  $a_0,\ a_1,\ a_2 \ldots$.

  Formally, $\alpha : \aleph \rightarrow \Sigma$. The set of all infinite words is denoted by Σ^ω.

Infinite Word Languages

Modelling infinite computations of reactive systems.

- An ω-word α over Σ is infinite sequence

  $$a_0,\ a_1,\ a_2 \ldots.$$

  Formally, $\alpha : \aleph \rightarrow \Sigma$. The set of all infinite words is denoted by Σ^ω.

- A ω-language L is collection of ω-words, i.e. $L \subseteq \Sigma^\omega$.

  Example All words over $\{a, b\}$ with infinitely many a.

Infinite Word Languages

Modelling infinite computations of reactive systems.

- An ω-word α over Σ is infinite sequence

  $a_0, \ a_1, \ a_2 \ldots$.

  Formally, $\alpha : \aleph \rightarrow \Sigma$. The set of all infinite words is denoted by Σ^ω.
- A ω-language L is collection of ω-words, i.e. $L \subseteq \Sigma^\omega$.

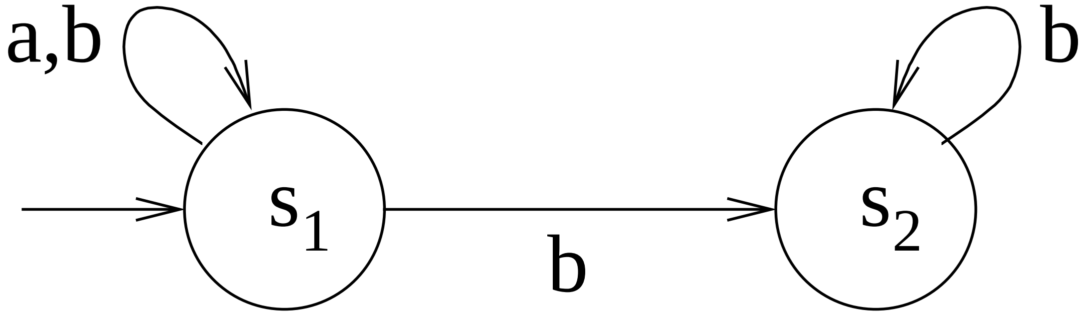
  Example All words over $\{a, b\}$ with infinitely many a.

Notation

omega words $\alpha, \beta, \gamma \in \Sigma^\omega$.
omega-languages $L, L_1 \subseteq \Sigma^\omega$
For $u \in \Sigma^+$, let $u^\omega = u.u.u \ldots$

Omega Automata

We consider automaton runs over infinite words words.



Let $\alpha = aabbbb\ldots$. There are several possible runs.
Run $\rho_1 = s_1, s_1, s_1, s_1, s_2, s_2 \ldots$
Run $\rho_2 = s_1, s_1, s_1, s_1, s_1, s_1 \ldots$

Acceptance Conditions Buchi, Muller, Rabin, Streett.
Acceptance is based on infinitely often occuring states

Notation Let $\rho \in S^\omega$. Then,

$$Inf(\rho) = \{s \in S \mid \exists^\infty i \in \aleph.\ \rho(i) = s\}.$$

Buchi Automata

Nondeterministic Buchi Automaton $A = (Q, \Sigma, \delta, I, F)$ where $F \subseteq Q$ is the set of accepting states.

Buchi Automata

Nondeterministic Buchi Automaton $A = (Q, \Sigma, \delta, I, F)$ where $F \subseteq Q$ is the set of accepting states.

- A run ρ of A on omega word α is infinite sequence $\rho = q_o, q_1, q_2, \ldots$ s.t. $q_0 \in I$ and $q_i \xrightarrow{a_i} q_{i+1}$ for $0 \leq i$.

Buchi Automata

Nondeterministic Buchi Automaton $A = (Q, \Sigma, \delta, I, F)$ where $F \subseteq Q$ is the set of accepting states.

- A run ρ of A on omega word α is infinite sequence $\rho = q_o, q_1, q_2, \ldots$ s.t. $q_0 \in I$ and $q_i \xrightarrow{a_i} q_{i+1}$ for $0 \leq i$.
- The run ρ is accepting if

  $$Inf(\rho) \cap F \neq \emptyset.$$

Buchi Automata

Nondeterministic Buchi Automaton $A = (Q, \Sigma, \delta, I, F)$ where $F \subseteq Q$ is the set of accepting states.

- A run ρ of A on omega word α is infinite sequence $\rho = q_o, q_1, q_2, \ldots$ s.t. $q_0 \in I$ and $q_i \xrightarrow{a_i} q_{i+1}$ for $0 \leq i$.
- The run ρ is accepting if

  $$Inf(\rho) \cap F \neq \emptyset.$$
- Language accepted by A
  $L(A) = \{\alpha \in \Sigma^* \mid A$ has an accepting run on $\alpha\}$
- Langauges accepted by NFA are called ω-regular languages.

Buchi Automata

Nondeterministic Buchi Automaton $A = (Q, \Sigma, \delta, I, F)$ where $F \subseteq Q$ is the set of accepting states.

- A run ρ of A on omega word α is infinite sequence $\rho = q_o, q_1, q_2, \ldots$ s.t. $q_0 \in I$ and $q_i \xrightarrow{a_i} q_{i+1}$ for $0 \leq i$.
- The run ρ is accepting if

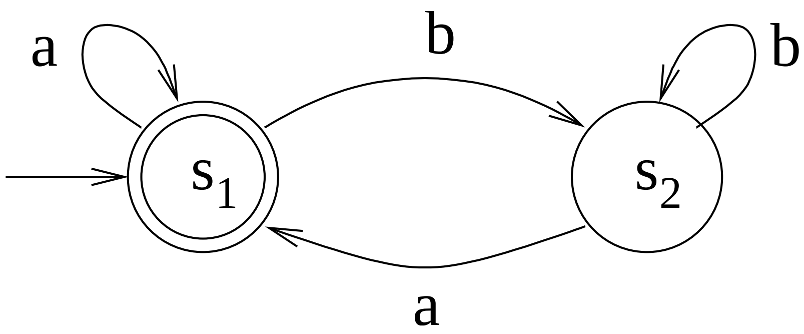
  $$Inf(\rho) \cap F \neq \emptyset.$$

- Language accepted by A
  $L(A) = \{\alpha \in \Sigma^* \mid A$ has an accepting run on $\alpha\}$
- Langauges accepted by NFA are called ω-regular languages.

A Deterministic Buchi Automaton has transition function $\delta : Q \times \Sigma \to Q$ and unique initial state $I = \{q_0\}$.

Buchi Automaton Example

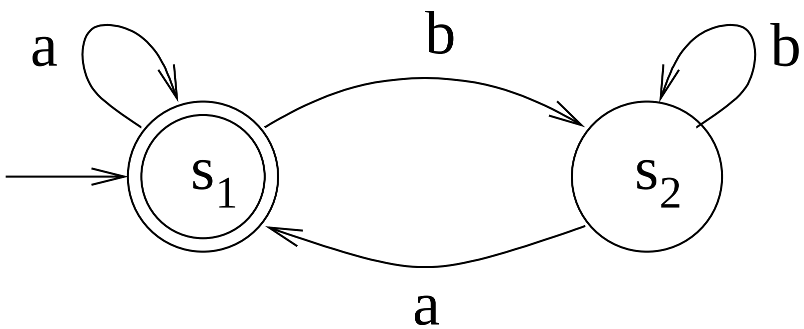
Let $\Sigma = \{a, b\}$.
Let Deterministic Buchi Automaton(DBA) A_1 be



- With $F = \{s_1\}$ the automaton recognises

Buchi Automaton Example

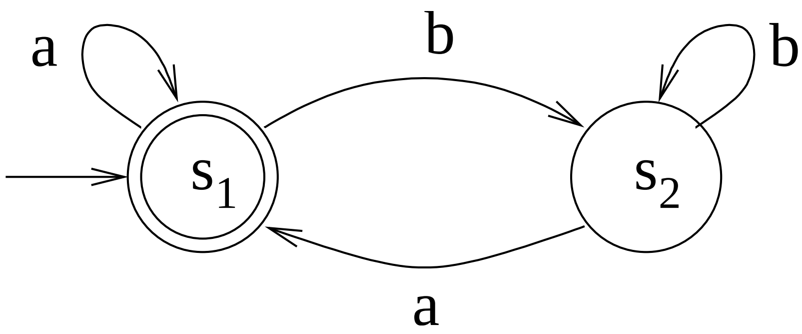
Let $\Sigma = \{a, b\}$.
Let Deterministic Buchi Automaton(DBA) A_1 be



- With $F = \{s_1\}$ the automaton recognises words with infinitely many a.

Buchi Automaton Example

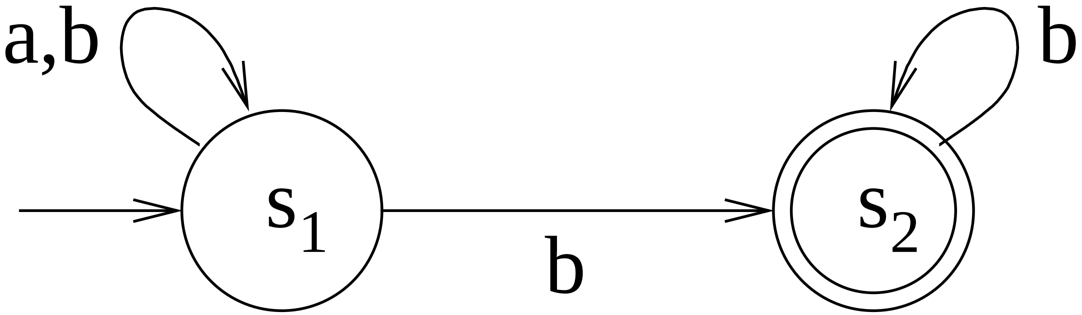
Let $\Sigma = \{a, b\}$.
Let Deterministic Buchi Automaton(DBA) A_1 be



- With $F = \{s_1\}$ the automaton recognises words with infinitely many a.
- With $F = \{s_2\}$ the automaton recognises words with infinitely many b.

Buchi Automaton Example 2

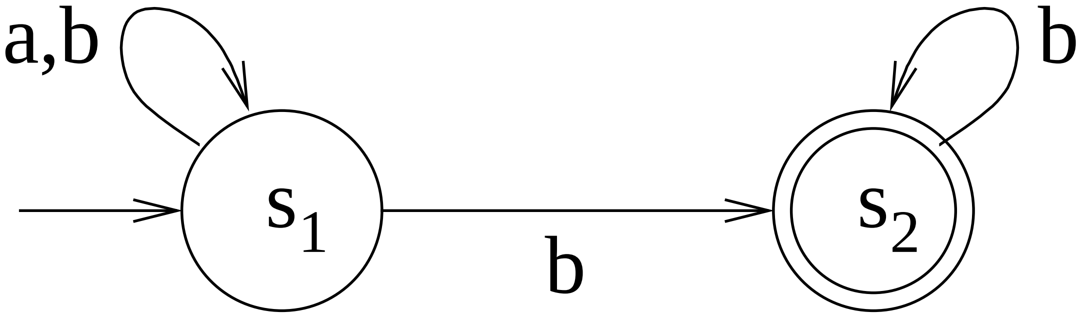
Let Nondeterministic Buchi Automaton(NBA) A_2 be



With $F = \{s_2\}$, automaton A_2 recognises

Buchi Automaton Example 2

Let Nondeterministic Buchi Automaton(NBA) A_2 be



With $F = \{s_2\}$, automaton A_2 recognises words with finitely many a.Thus, $L(A_2) = \overline{L(A_1)}$.

Deterministic Buchi Automata

Limit Languages Let $U \subseteq \Sigma^*$. Then,

$$lim(U) \stackrel{\text{def}}{=} \{\alpha \in \Sigma^\omega \mid \exists^\infty i \in \aleph.\ \alpha[0:i] \in U\}.$$

Example: $lim((ab)^*) = \{(ab)^\omega\}$.

Deterministic Buchi Automata

Limit Languages Let $U \subseteq \Sigma^*$. Then,

$$lim(U) \stackrel{\text{def}}{=} \{\alpha \in \Sigma^\omega \mid \exists^\infty i \in \aleph.\ \alpha[0:i] \in U\}.$$

Example: $lim((ab)^*) = \{(ab)^\omega\}$.

Theorem $L \subseteq \Sigma^\omega$ is DBA recognisable iff L has the form $Lim(U)$ for some regular langauge $U \subseteq \Sigma^*$.

Proof Method Relate the langauges of DFA for U with DBA for L.

Deterministic Buchi Automata

Limit Languages Let $U \subseteq \Sigma^*$. Then,

$$lim(U) \stackrel{\text{def}}{=} \{\alpha \in \Sigma^\omega \mid \exists^\infty i \in \aleph.\ \alpha[0:i] \in U\}.$$

Example: $lim((ab)^*) = \{(ab)^\omega\}$.

Theorem $L \subseteq \Sigma^\omega$ is DBA recognisable iff L has the form $Lim(U)$ for some regular langauge $U \subseteq \Sigma^*$.

Proof Method Relate the langauges of DFA for U with DBA for L.

Claim Language $L(A_2)$ of words with finitely many a is not of form $Lim(U)$ for any regular U.

Deterministic Buchi Automata

Limit Languages Let $U \subseteq \Sigma^*$. Then,

$$lim(U) \stackrel{\text{def}}{=} \{\alpha \in \Sigma^\omega \mid \exists^\infty i \in \aleph.\ \alpha[0:i] \in U\}.$$

Example: $lim((ab)^*) = \{(ab)^\omega\}$.

Theorem $L \subseteq \Sigma^\omega$ is DBA recognisable iff L has the form $Lim(U)$ for some regular langauge $U \subseteq \Sigma^*$.

Proof Method Relate the langauges of DFA for U with DBA for L.

Claim Language $L(A_2)$ of words with finitely many a is not of form $Lim(U)$ for any regular U.

Corollary DBA are strictly less powerful than NBA.

Closure Properties

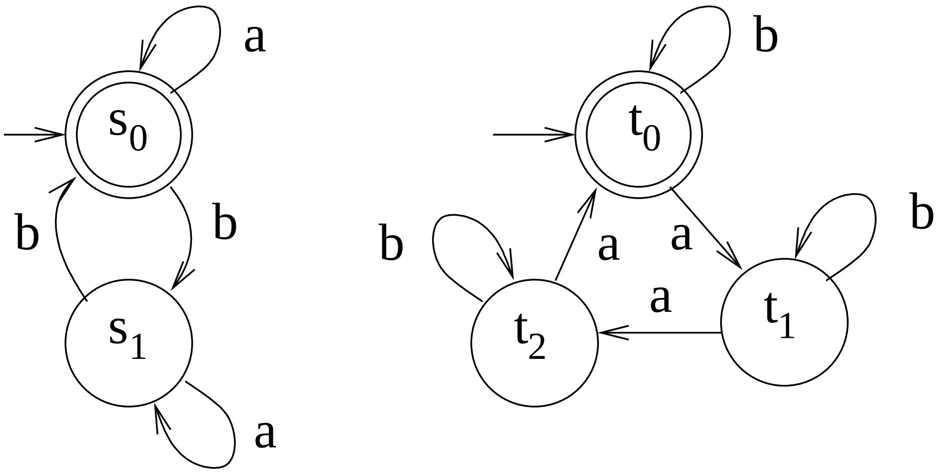
Theorem (Union) For NBA A_1, A_2 we can effectively construct an NBA A s.t. $L(A) = L(A_1) \cup L(A_2)$. The size $|A| = |A_1| + |A_2|$

Construction Take disjoint union of A_1 and A_2.

Theorem (Intersection) For NBA A_1, A_2 we can effectively construct NBA A s.t. $L(A) = L(A_1) \cap L(A_2)$. The size $|A| = |A_1| \times |A_2| \times 2$.

Proof Method Construct product automaton.

Example: Product of NBA



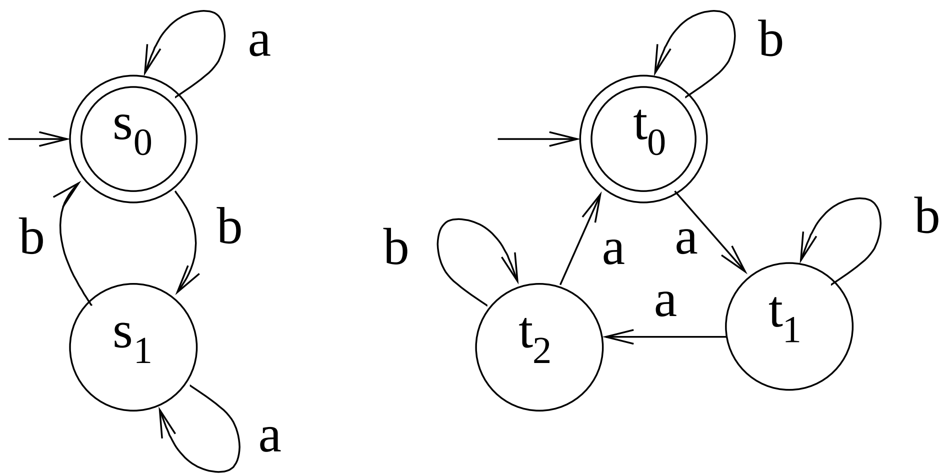
Consider the run on $\alpha = baa\ (bbaaa)\ (bbaaa)\ (bbaaa)\ldots$.

Positions of final states of two automata.

$$\alpha = ||baa\ \ (b|ba|aa)^{\omega}.$$

Does not visit final states simultaneously. But belongs to intersection.

Example: Product of NBA



Consider the run on $\alpha \;=\; baa\ (bbaaa)\ (bbaaa)\ (bbaaa)\ldots$.

Positions of final states of two automata.

$$\alpha \;=\; ||baa\ \ (b|ba|aa)^{\omega}.$$

Does not visit final states simultaneously. But belongs to intersection.

Solution Each component final state must be visited infinitely often, but not necessarily simulataneously.

Synchronous Product of NBA

Let $A_1 = (Q_1, \Sigma, \delta_1, I_1, F_1)$ and $A_2 = (Q_2, \Sigma, \delta_2, I_2, F_2)$.
Then, $A_1 \times A_2 = (Q, \Sigma, \delta, I, F)$ where

$$Q = Q_1 \times Q_2 \times \{1,2\}. \qquad I = I_1 \times I_2 \times \{1\}.$$
$$F = F_1 \times Q_2 \times \{1\}.$$

$< p, q, 1 > \xrightarrow{a} < p', q', 1 >$ iff $p \xrightarrow{a} p'$ and $q \xrightarrow{a} q'$ and $p \notin F_1$.
$< p, q, 1 > \xrightarrow{a} < p', q', 2 >$ iff $p \xrightarrow{a} p'$ and $q \xrightarrow{a} q'$ and $p \in F_1$.
$< p, q, 2 > \xrightarrow{a} < p', q', 2 >$ iff $p \xrightarrow{a} p'$ and $q \xrightarrow{a} q'$ and $q \notin F_2$.
$< p, q, 2 > \xrightarrow{a} < p', q', 1 >$ iff $p \xrightarrow{a} p'$ and $q \xrightarrow{a} q'$ and $q \in F_2$.

Theorem $L(A_1 \times A_2) = L(A_1) \cap L(A_2)$.

Closure Properties (2)

Theorem (projection) For NBA A_1 over Σ_1 and surjection $h : \Sigma_1 \to \Sigma_2$, we can construct A_2 over Σ_2 s.t. $L(A_2) = h(L(A_1))$.

Construction Substitute label a by $h(a)$ in each transition.

This can turn DBA into NBA.

Closure Properties (2)

Theorem (projection) For NBA A_1 over Σ_1 and surjection $h : \Sigma_1 \to \Sigma_2$, we can construct A_2 over Σ_2 s.t. $L(A_2) = h(L(A_1))$.
Construction Substitute label a by $h(a)$ in each transition.
This can turn DBA into NBA.

Theorem (complementation) [Safra, MacNaughten] For NBA A_1 we can construct NBA A_2 such that $L(A_2) = \overline{L(A_1)}$. Size $|A_2| = O(2^{n \log n})$ where $|A_1| = n$.

Decision Problems

Emptiness For NBA A, it is decidable whether $L(A) = \emptyset$.

Method

- Find maximal strongly connected components (SCC) in graph of A disregarding the edge labels.
- A MSC Component C is called non-trivial if $C \cap F \neq \emptyset$ and C has at least one edge.
- Find all nodes from which there is a path to a non-trivial SCC. Call the set of these nodes as N.
- $L(A) = \emptyset$ iff $N \cap I = \emptyset$.

Time Complexity: $O(|Q| + |\delta|)$.

Decision Problems

Emptiness For NBA A, it is decidable whether $L(A) = \emptyset$.

Method

- Find maximal strongly connected components (SCC) in graph of A disregarding the edge labels.
- A MSC Component C is called non-trivial if $C \cap F \neq \emptyset$ and C has at least one edge.
- Find all nodes from which there is a path to a non-trivial SCC. Call the set of these nodes as N.
- $L(A) = \emptyset$ iff $N \cap I = \emptyset$.

Time Complexity: $O(|Q| + |\delta|)$.

Study Topic Courcoubetis *et al*, Memory efficient algorithms of verification of temporal properties, Formal Methods in System Design, 1992.

Omega Regular Expressions

Define $U^\omega \;=\; \{u_0.u_1 \ldots \mid u_i \in U\}$.
Define $U.L \;=\; \{u.\alpha \mid u \in U,\ \alpha \in L\}$.

A language is called ω-regular if it has the form $\bigcup_{i=1}^{n} U_i.(V_i)^\omega$ where U_i, V_i are regular languages.

Theorem A language L is ω-regular iff it is NBA recognisable.
Proof ($\Rightarrow$) Let A be NBA for L. Then,

$$L \quad = \quad \bigcup_{i \in I, f \in F} (\alpha^Q_{i,f}) \cdot (\alpha^Q_{f,f})^\omega.$$

Lemma Let U be regular and L, L_i be NBA recognizable.
Then $U \cdot L$ is NBA recognizable.
U^ω is NBA recognizable.
$\bigcup_{0 \leq i \leq n} L_i$ is NBA recognizable.

Variety of Acceptance Conditions

Consider Automaton Graph $AG = (Q, \Sigma, \delta, I)$. A Buchi automaton is a pair (A, F) where $F \subseteq Q$.
Let $FT = \ < F_1, F_2, \ldots, F_k >$ with $F_i \subseteq Q$.

A Generalised Buchi Automaton is (A, FT) where FT is as above.

Variety of Acceptance Conditions

Consider Automaton Graph $AG = (Q, \Sigma, \delta, I)$. A Buchi automaton is a pair (A, F) where $F \subseteq Q$.
Let $FT = \; < F_1, F_2, \ldots, F_k >$ with $F_i \subseteq Q$.

A Generalised Buchi Automaton is (A, FT) where FT is as above.
A run ρ of A is accepting if $Inf(\rho) \cap F_i \neq \emptyset$ for each $1 \leq i \leq k$.

Variety of Acceptance Conditions

Consider Automaton Graph $AG = (Q, \Sigma, \delta, I)$. A Buchi automaton is a pair (A, F) where $F \subseteq Q$.
Let $FT = < F_1, F_2, \ldots, F_k >$ with $F_i \subseteq Q$.

A Generalised Buchi Automaton is (A, FT) where FT is as above.
A run ρ of A is accepting if $Inf(\rho) \cap F_i \neq \emptyset$ for each $1 \leq i \leq k$.

Theorem For every Generalised Buchi Automaton (A, FT) we can construct a language equivalent Buchi Automaton (A', G').
Construction Let $Q' = Q \times \{1, \ldots, k\}$.
Automaton remains in i phase till it visits a state in F_i. Then, it moves to $i+1$ mode. After phase k it moves to phase 1.

Simulating GBA by BA

Let GBA $A = (Q, \Sigma, \delta, I)$ with $FT = (F_1, \dots, F_k)$. Then we construct the BA $A' = (Q', \Sigma, \delta', I', F')$ where

$$Q' = Q \times \{1, \dots, k\}.$$
$$I' = I \times \{1\}.$$
$$F' = F \times \{1\}.$$

Simulating GBA by BA

Let GBA $A = (Q, \Sigma, \delta, I)$ with $FT = (F_1, \ldots, F_k)$. Then we construct the BA $A' = (Q', \Sigma, \delta', I', F')$ where

$$
\begin{aligned}
Q' &= Q \times \{1, \ldots, k\}. \\
I' &= I \times \{1\}. \\
F' &= F \times \{1\}.
\end{aligned}
$$

The transition relation is:

- $< p, i > \xrightarrow{a} < q, i >$ iff $p \xrightarrow{a} p'$ and $p \notin F_i$.
- $< p, i > \xrightarrow{a} < q, j >$ iff $p \xrightarrow{a} q$ and $p \in F_i$
  where $j = i + 1$ if $i < k$ and $j = 1$ otherwise.

Simulating GBA by BA

Let GBA $A = (Q, \Sigma, \delta, I)$ with $FT = (F_1, \ldots, F_k)$. Then we construct the BA $A' = (Q', \Sigma, \delta', I', F')$ where

$$Q' = Q \times \{1, \ldots, k\}.$$
$$I' = I \times \{1\}.$$
$$F' = F \times \{1\}.$$

The transition relation is:

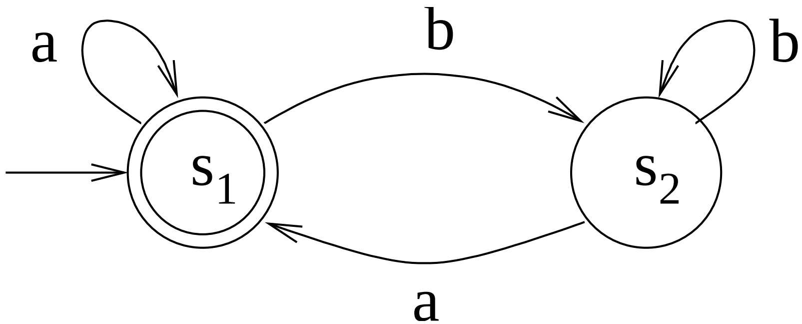
- $< p, i > \xrightarrow{a} < q, i >$ iff $p \xrightarrow{a} p'$ and $p \notin F_i$.
- $< p, i > \xrightarrow{a} < q, j >$ iff $p \xrightarrow{a} q$ and $p \in F_i$
  where $j = i + 1$ if $i < k$ and $j = 1$ otherwise.

Lemma L(A) = L(A'). Size $|A'| = |A| \times k$.

Muller Automata

A Muller automaton is (A, FT). A run ρ of A is Muller-accepting if $Inf(\rho) \in FT$.

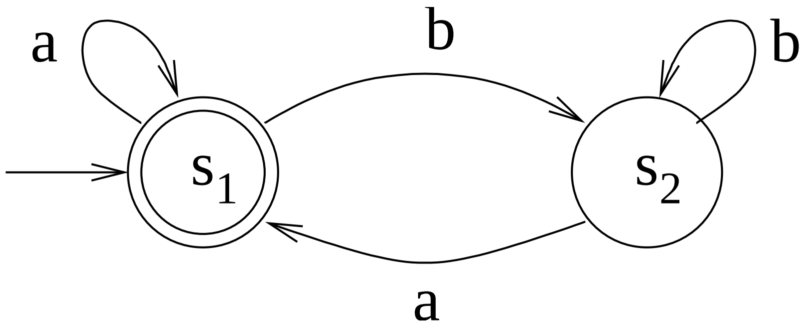
Example Deterministic Muller automaton A_1 recognises:
– for $FT = \langle\{s_1\}, \{s_1, s_2\}\rangle$.
– for $FT = \langle\{s_2\}\rangle$.



Muller Automata

A Muller automaton is (A, FT). A run ρ of A is Muller-accepting if $Inf(\rho) \in FT$.

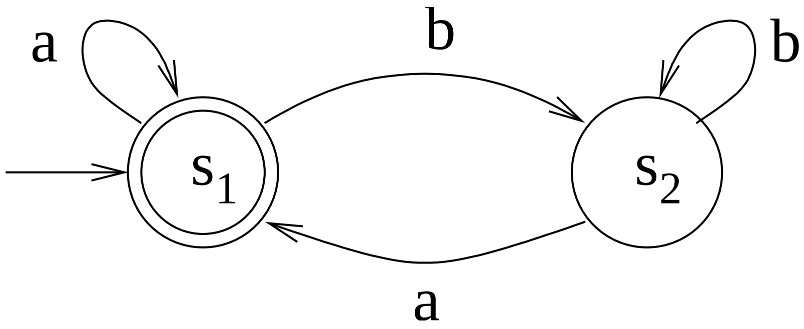
Example Deterministic Muller automaton A_1 recognises:
– words with infinitely many a for $FT = \langle \{s_1\}, \{s_1, s_2\} \rangle$.
– words with finitely many a for $FT = \langle \{s_2\} \rangle$.



Muller Automata

A Muller automaton is (A, FT). A run ρ of A is Muller-accepting if $Inf(\rho) \in FT$.

Example Deterministic Muller automaton A_1 recognises:
– words with infinitely many a for $FT = \langle \{s_1\}, \{s_1, s_2\} \rangle$.
– words with finitely many a for $FT = \langle \{s_2\} \rangle$.



Exercise Describe $L(A_1)$ of the above Muller Aut. when (a) $FT = \langle \{s_1\} \rangle$, and (b) $FT = \langle \{s_1\}, \{s_2\} \rangle$.

Muller Automata (2)

Theorem For every Buchi automaton A_1 there is a language equivalent Muller automaton A_2.

Muller Automata (2)

Theorem For every Buchi automaton A_1 there is a language equivalent Muller automaton A_2.

Construction A_1 and A_2 have same automaton graph. Let F be set of final states of Buchi A_1. Define Muller aut. final table $FT = \{Y \in 2^Q \mid Y \cap F \neq \emptyset\}$.

Muller Automata (2)

Theorem For every Buchi automaton A_1 there is a language equivalent Muller automaton A_2.

Construction A_1 and A_2 have same automaton graph. Let F be set of final states of Buchi A_1. Define Muller aut. final table $FT = \{Y \in 2^Q \mid Y \cap F \neq \emptyset\}$.

Theorem[McNaughten] For every Buchi Automaton A_1 we can construct a language equivalent Deterministic Muller Automaton A_2.

Muller to Buchi

Theorem For every Muller automaton AM we can construct a langauge equivalent Buchi Automaton AB.

Muller to Buchi

Theorem For every Muller automaton AM we can construct a langauge equivalent Buchi Automaton AB.
Construction Let FT be acceptance table of $AM = (A, < F_1, \ldots, F_k >)$.

- We construct a Nondeterministic Buchi Automaton A_i s.t. word α is accepted by A_i iff there is an ρ accepting run of AM on α with $Inf(\rho) \ = \ F_i$.
- Then, $L(AM) \ = \ \cup \ L(A_i)$ which is Buchi recognisable.

Muller to Buchi

Theorem For every Muller automaton AM we can construct a langauge equivalent Buchi Automaton AB.
Construction Let FT be acceptance table of $AM = (A, < F_1, \ldots, F_k >)$.

- We construct a Nondeterministic Buchi Automaton A_i s.t. word α is accepted by A_i iff there is an ρ accepting run of AM on α with $Inf(\rho) = F_i$.
- Then, $L(AM) = \cup L(A_i)$ which is Buchi recognisable.

Construction of A_i Any run of AM has initial finite part followed by infinite part. The finite part follows automaton graph of AM. In infinite part only the F_i states can be visited and each must be visited infinitely often.

Construction of A_i

Let $AM = (Q, \Sigma, \delta, I, FT)$ with $F_i = \{f_1, f_2, \ldots, f_{m-1}\}$. The NBA $A_i = (Q_i, \Sigma, \delta_i, I_i, G_i)$ where

Construction of A_i

Let $AM = (Q, \Sigma, \delta, I, FT)$ with $F_i = \{f_1, f_2, \ldots, f_{m-1}\}$. The NBA $A_i = (Q_i, \Sigma, \delta_i, I_i, G_i)$ where

$$\begin{aligned} Q_i &= \{(q, fin) \mid q \in Q\} \cup \\ &\quad \{(f, inf, j) \mid f \in F_i \land j \in \{1, \ldots, m\}\}. \end{aligned}$$

$I_i = \{(s, fin) \mid s \in I\}$ and $G_i = \{(f_m, inf, m)\}$

Construction of A_i

Let $AM = (Q, \Sigma, \delta, I, FT)$ with $F_i = \{f_1, f_2, \ldots, f_{m-1}\}$. The NBA $A_i = (Q_i, \Sigma, \delta_i, I_i, G_i)$ where

$$Q_i = \{(q, fin) \mid q \in Q\} \cup \{(f, inf, j) \mid f \in F_i \wedge j \in \{1, \ldots, m\}\}.$$

$I_i = \{(s, fin) \mid s \in I\}$ and $G_i = \{(f_m, inf, m)\}$

Transition Relation:

Rabin and Streett Automata

A be automaton graph (Q, Σ, δ, I) as before.
Let $PT = \; < (G_1, R_1), (G_2, R_2), \ldots, (G_k, R_k) >$ with $G_i, R_i \subseteq Q$.

Rabin and Streett Automata

A be automaton graph (Q, Σ, δ, I) as before.
Let $PT = \; < (G_1, R_1), (G_2, R_2), \ldots, (G_k, R_k) >$ with $G_i, R_i \subseteq Q$.

A Rabin automaton is (A, PT). A run ρ of A is Rabin-accepting if for some $i : 0 \leq i \leq k$ we have $Inf(\rho) \cap G_i \neq \emptyset$ and $Inf(\rho) \cap R_i = \emptyset$.

Rabin and Streett Automata

A be automaton graph (Q, Σ, δ, I) as before.
Let $PT = \ < (G_1, R_1), (G_2, R_2), \ldots, (G_k, R_k) >$ with $G_i, R_i \subseteq Q$.

A Rabin automaton is (A, PT). A run ρ of A is Rabin-accepting if for some $i : 0 \leq i \leq k$ we have $Inf(\rho) \cap G_i \neq \emptyset$ and $Inf(\rho) \cap R_i = \emptyset$.

A Streett automaton is (A, PT). A run ρ of A is Streett-accepting if for all $i : 0 \leq i \leq k$ we have $Inf(\rho) \cap G_i \neq \emptyset$ implies $Inf(\rho) \cap R_i \neq \emptyset$.

Rabin and Streett Automata

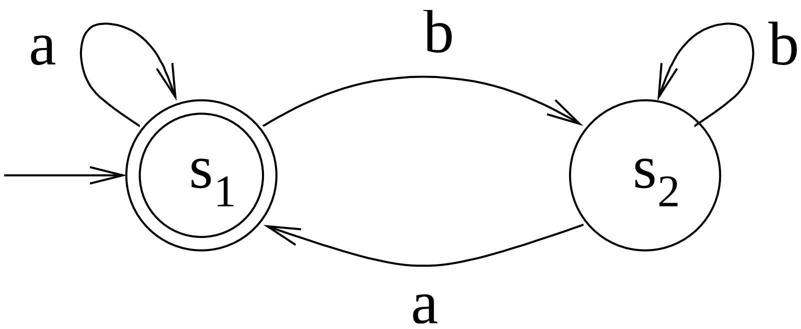
A be automaton graph (Q, Σ, δ, I) as before.
Let $PT = < (G_1, R_1), (G_2, R_2), \ldots, (G_k, R_k) >$ with $G_i, R_i \subseteq Q$.

A Rabin automaton is (A, PT). A run ρ of A is Rabin-accepting if for some $i : 0 \leq i \leq k$ we have $Inf(\rho) \cap G_i \neq \emptyset$ and $Inf(\rho) \cap R_i = \emptyset$.

A Streett automaton is (A, PT). A run ρ of A is Streett-accepting if for all $i : 0 \leq i \leq k$ we have $Inf(\rho) \cap G_i \neq \emptyset$ implies $Inf(\rho) \cap R_i \neq \emptyset$.

Proposition ρ is Rabin accepting iff ρ is not street accepting.

Examples

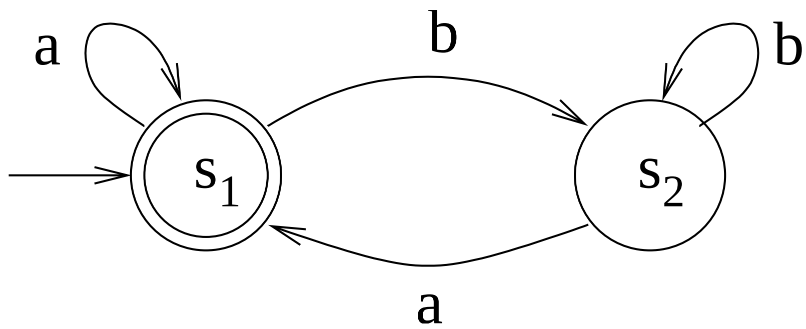


The Rabin Automaton above

- with $PT = < (\{s_1\}, \emptyset) >$

- with $PT = < (\{s_2\}, \{s_1\}) >$

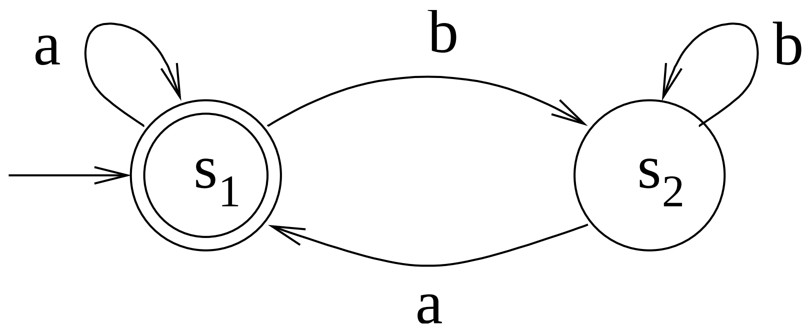
Examples



The Rabin Automaton above

- with $PT =< (\{s_1\}, \emptyset) >$ accepts words with infinitely many a.
- with $PT =< (\{s_2\}, \{s_1\}) >$

Examples



The Rabin Automaton above

- with $PT =< (\{s_1\}, \emptyset) >$ accepts words with infinitely many a.
- with $PT =< (\{s_2\}, \{s_1\}) >$ accepts words with finitely many a.

Simulations

Buchi-to-Rabin Let F final states of Buchi. Let $PT \stackrel{\text{def}}{=} \langle (F, \emptyset) \rangle$.

Simulations

Buchi-to-Rabin Let F final states of Buchi. Let $PT \stackrel{\mathrm{def}}{=} \langle (F, \emptyset) \rangle$.

Buchi-to-Streett Let $PT \stackrel{\mathrm{def}}{=} \langle (Q, F) \rangle$.

Simulations

Buchi-to-Rabin Let F final states of Buchi. Let $PT \stackrel{\text{def}}{=} \langle(F, \emptyset)\rangle$.

Buchi-to-Streett Let $PT \stackrel{\text{def}}{=} \langle(Q, F)\rangle$.

Rabin-to-Buchi Similar to Muller-to-Buchi. Complexity $|Q| \times k$.

Simulations

Buchi-to-Rabin Let F final states of Buchi. Let $PT \stackrel{\mathrm{def}}{=} \langle (F, \emptyset) \rangle$.

Buchi-to-Streett Let $PT \stackrel{\mathrm{def}}{=} \langle (Q, F) \rangle$.

Rabin-to-Buchi Similar to Muller-to-Buchi. Complexity $|Q| \times k$.

Streett-to-Buchi [Vardi] Complexity $|Q| \times 2^k$.

Rabin-to-Buchi

Classroom.

Exercise Give construction for simulating Rabin Automaton using a Muller Automaton.

Streett-to-Buchi

Given Streett Automataon (A, PT) with $A = (Q, \Sigma, \delta, I)$ and $PT = \ < (G_1, R_1), (G_2, R_2), \ldots, (G_k, R_k) >$ we construct NBA (A', G').

Streett-to-Buchi

Given Streett Automataon (A, PT) with $A = (Q, \Sigma, \delta, I)$ and $PT \;=\; < (G_1, R_1), (G_2, R_2), \ldots, (G_k, R_k) >$ we construct NBA (A', G').

Buchi automaton simulates A for initial finite prefix and then nondeterminstically moves to infinite part where it checks that Streett-condition is met.

Streett-to-Buchi

Given Streett Automataon (A, PT) with $A = (Q, \Sigma, \delta, I)$ and $PT = \ < (G_1, R_1), (G_2, R_2), \ldots, (G_k, R_k) >$ we construct NBA (A', G').

Buchi automaton simulates A for initial finite prefix and then nondeterminstically moves to infinite part where it checks that Streett-condition is met.

- For this it keeps two sets $X_1, X_2 \subseteq \{1, \ldots, k\}$.
- If q is occurs, indices i such $q \in G_i$ are added to X_1.
- Similarly if q is occurs, indices i such $q \in R_i$ are added to X_2.
- If $G_i \subseteq R_i$ then all requirements are met. We set $R_i = \emptyset$. This should happen infinitely often.

(Cont)

$$
\begin{aligned}
Q' &= \{(q, fin) \mid q \in Q\} \cup \\
&\qquad \{(q, X_1, X_2) \mid q \in Q \wedge X_1, X_2 \subseteq \{1, \ldots, k\}\}.
\end{aligned}
$$

(Cont)

$$Q' = \{(q, fin) \mid q \in Q\} \cup \{(q, X_1, X_2) \mid q \in Q \wedge X_1, X_2 \subseteq \{1, \ldots, k\}\}.$$

$$G' = \{(q, X, \emptyset) \mid q \in Q \wedge X \subseteq \{1, \ldots, k\}\}.$$

(Cont)

$$Q' = \{(q, fin) \mid q \in Q\} \cup \{(q, X_1, X_2) \mid q \in Q \wedge X_1, X_2 \subseteq \{1, \ldots, k\}\}.$$

$$G' = \{(q, X, \emptyset) \mid q \in Q \wedge X \subseteq \{1, \ldots, k\}\}.$$

- $(p, fin) \xrightarrow{a} (q, fin)$ of $p \xrightarrow{a} q$.
- $(p, fin) \xrightarrow{a} (q, \emptyset, \emptyset)$ if $p \xrightarrow{a} q$.
- $(p, X, Y) \xrightarrow{a} (q, X \cup A, Y \cup B)$ if $p \xrightarrow{a} q$ and $X \cup A \not\subseteq Y \cup B$ and $A = \{i \mid q \in G_i\}$ and $B = \{i \mid q \in R_i\}$.
- $(p, X, Y) \xrightarrow{a} (q, X \cup A, \emptyset)$ if $p \xrightarrow{a} q$ and $X \cup A \subseteq Y \cup B$.

Safra's Determinisation

Theorem For every Nondeterministic Buchi Automaton (A, F) we can construct a language equivalent deterministic Rabin automaton (A_F, PT_F).

Safra's Determinisation

Theorem For every Nondeterministic Buchi Automaton (A, F) we can construct a language equivalent deterministic Rabin automaton (A_F, PT_F).

Compleixty: A_F has $O(2^{(n \ \log \ n)})$ states where A has n states. There is no construction with $O(2^n)$ states.

Safra's Determinisation

Theorem For every Nondeterministic Buchi Automaton (A, F) we can construct a language equivalent deterministic Rabin automaton (A_F, PT_F).

Compleixty: A_F has $O(2^{(n \ \log \ n)})$ states where A has n states. There is no construction with $O(2^n)$ states.

Complementation of Buchi Automata:

(1) Buchi to Deterministic-Rabin.

(2) Deterministic-Rabin to Determinstic Streett (Complement)

(3) Determinstic-Streett to Nodeterministic-Buchi