The McNaughton Theorem

McNaughton Theorem

Theorem 1 *Let Σ be an alphabet. Any ω-recognizable subset of Σ^ω can be recognized by a Rabin automaton.*

Determinisation algorithm by S. Safra (1989) uses a special subset construction to obtain a Rabin automaton equivalent to a given Büchi automaton. The Safra algorithm is optimal $2^{O(n \log n)}$.

This proves that ω-recognizable languages are closed under complement.

Oriented Trees

Let Σ be an alphabet of labels.

An oriented tree is a pair of partial functions $t = \langle l, s \rangle$:

- $l : \mathbb{N} \mapsto \Sigma$ denotes the labels of the nodes
- $s : \mathbb{N} \mapsto \mathbb{N}^*$ gives the ordered list of children of each node

$dom(l) = dom(s) \stackrel{def}{=} dom(t)$

$p \leq q$: q is a successor of p in t

$p \preceq_{left} q$: p is to the left of q in t ($p \preceq q$ and $p \not\leq q$)

Safra Trees

Let $A = \langle S, I, T, F \rangle$ be a Büchi automaton.

A *Safra tree* is a pair $\langle t, m \rangle$, where t is a finite oriented tree labeled with non-empty subsets of S, and $m \subseteq dom(t)$ is the set of *marked positions*, such that:

- each marked position is a leaf
- for each $p \in dom(t)$, the union of labels of its children is a strict subset of $t(p)$
- for each $p, q \in dom(t)$, if $p \not\leq q$ and $q \not\leq p$ then $t(p) \cap t(q) = \emptyset$

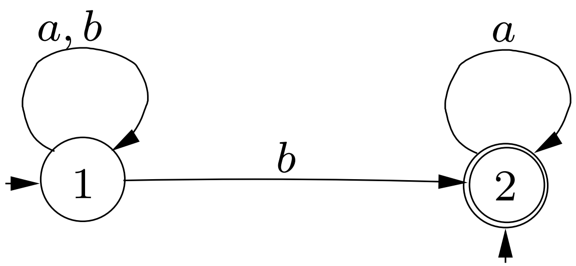
Proposition 1 *A Safra tree has at most $\|S\|$ nodes.*

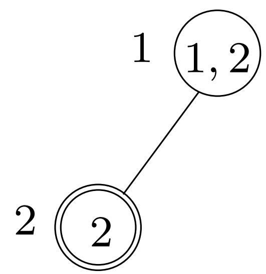
$$
\begin{aligned}
r(p) &= t(p) \setminus \bigcup_{q<p} t(q) \\
\|dom(t)\| &= \sum_{p \in dom(t)} 1 \leq \sum_{p \in dom(t)} \|r(p)\| \leq \|S\|
\end{aligned}
$$

Initial State

We build a Rabin automaton $B = \langle S_B, i_B, T_B, \Omega_B \rangle$, where:

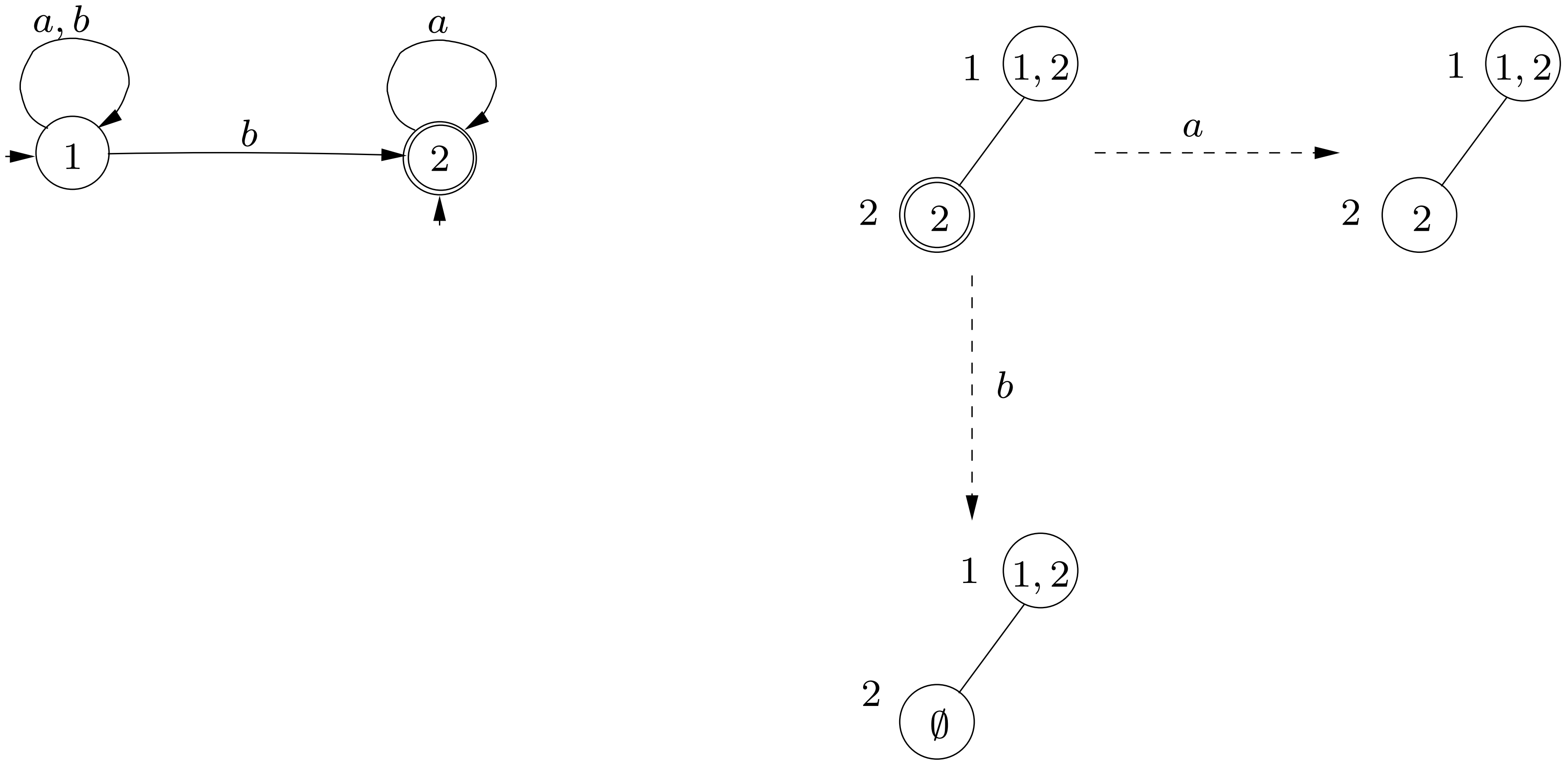
- S_B is the set of all Safra trees $\langle t, m \rangle$ labeled with subsets of S
- $i_B = \langle t, m \rangle$ is the Safra tree defined as either:
  - $dom(t) = \{1\}, t(1) = I$ and $m = \emptyset$ if $I \cap F = \emptyset$
  - $dom(t) = \{1\}, t(1) = I$ and $m = \{\epsilon\}$ if $I \subseteq F$
  - $dom(t) = \{1, 2\}, t(1) = I, t(2) = I \cap F$ and $m = \{2\}$ if $I \cap F \neq \emptyset$





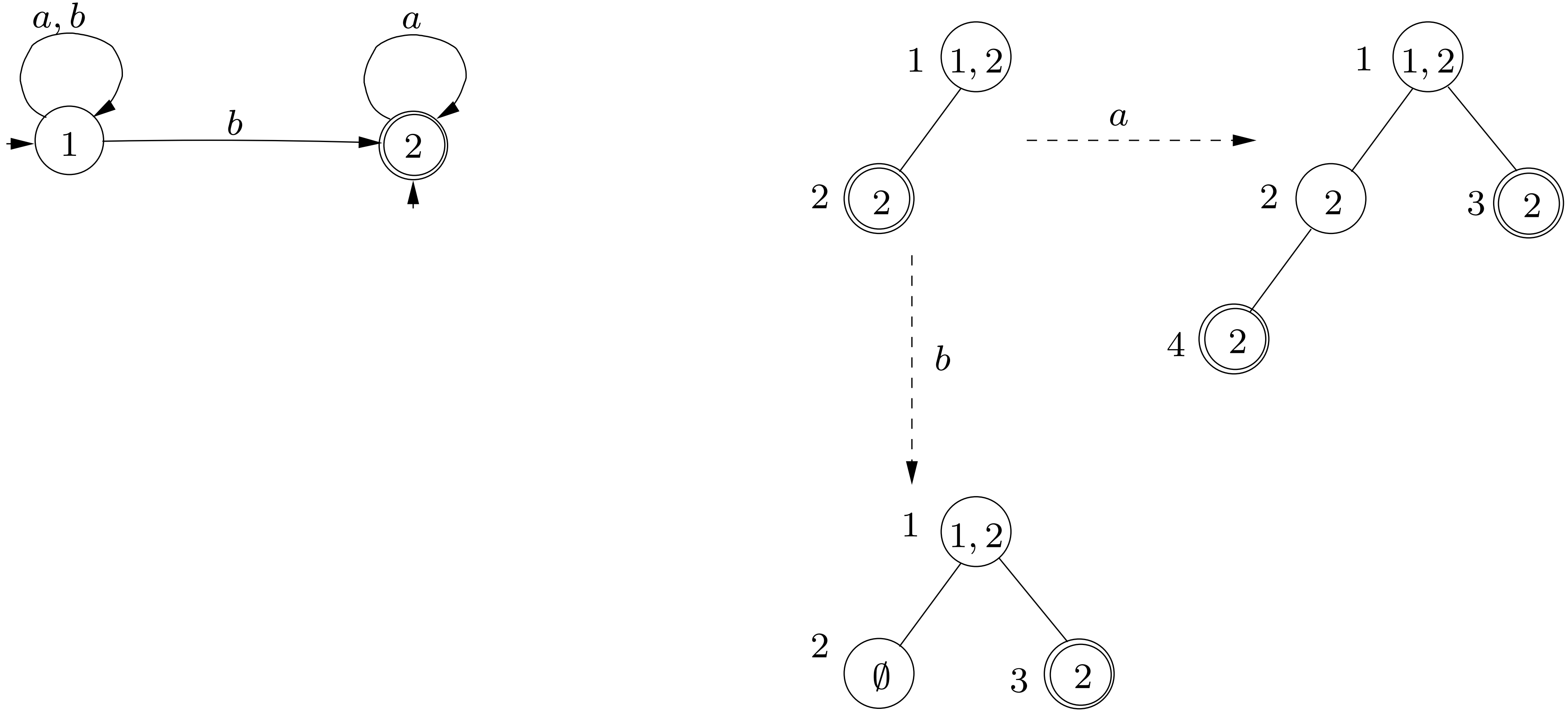
Classical Subset Move

[Step 1] $\langle t_1, m_1 \rangle$ is the tree with $dom(t_1) = dom(t), m_1 = \emptyset$, and $t_1(p) = \{s' \mid s \xrightarrow{\alpha} s',\ s \in t(p)\}$, for all $p \in dom(t)$



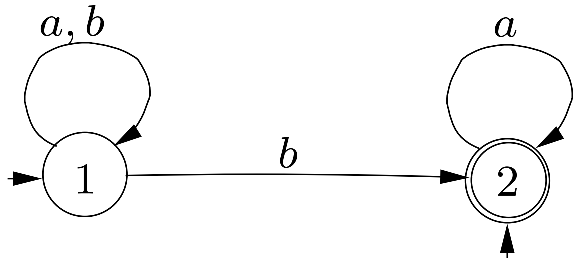
Spawn New Children

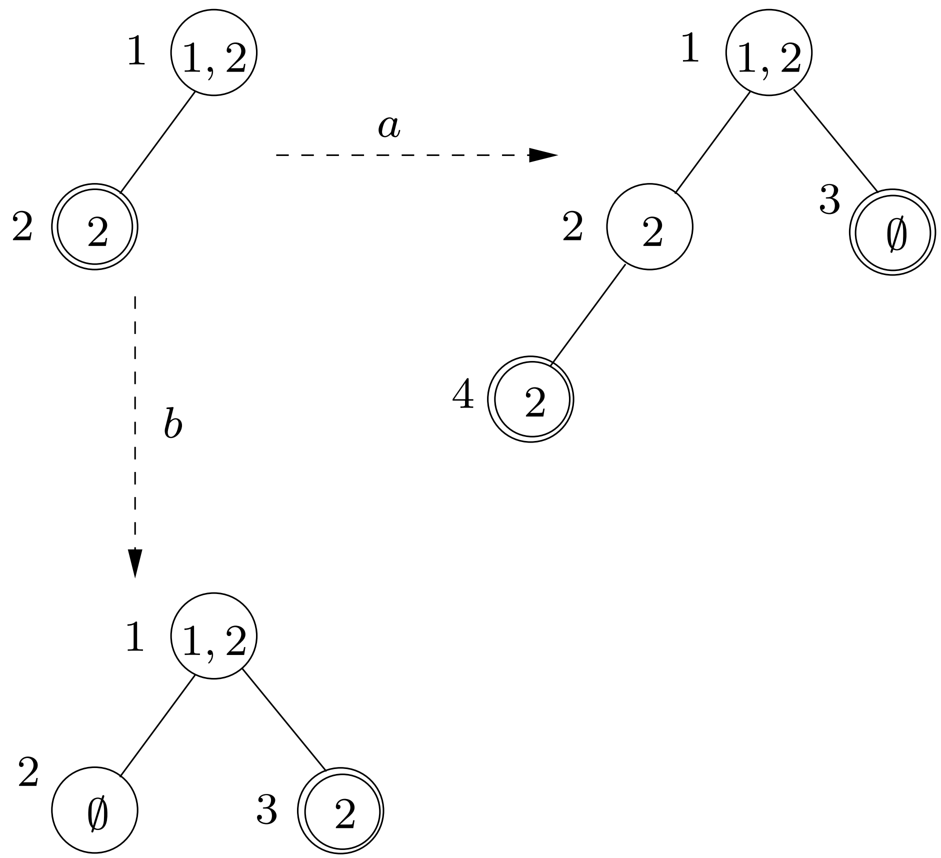
[Step 2] $\langle t_2, m_2 \rangle$ is the tree such that, for each $p \in dom(t_1)$, if $t_1(p) \cap F \neq \emptyset$ we add a new child to the right, identified by the first available id, and labeled $t_1(p) \cap F$, and m_2 is the set of all such children



Horizontal Merge

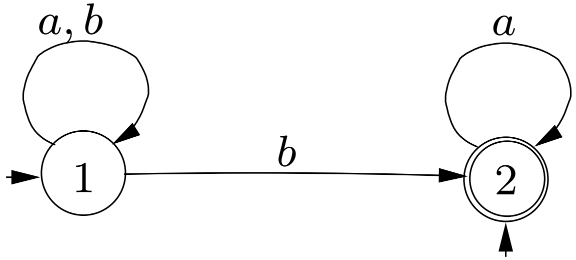
[Step 3] $\langle t_3, m_3 \rangle$ is the tree with $dom(t_3) = dom(t_2)$, $m_3 = m_2$, such that, for all $p \in dom(t_3)$, $t_3(p) = t_2(p) \setminus \bigcup_{q \prec_{left} p} t_2(q)$

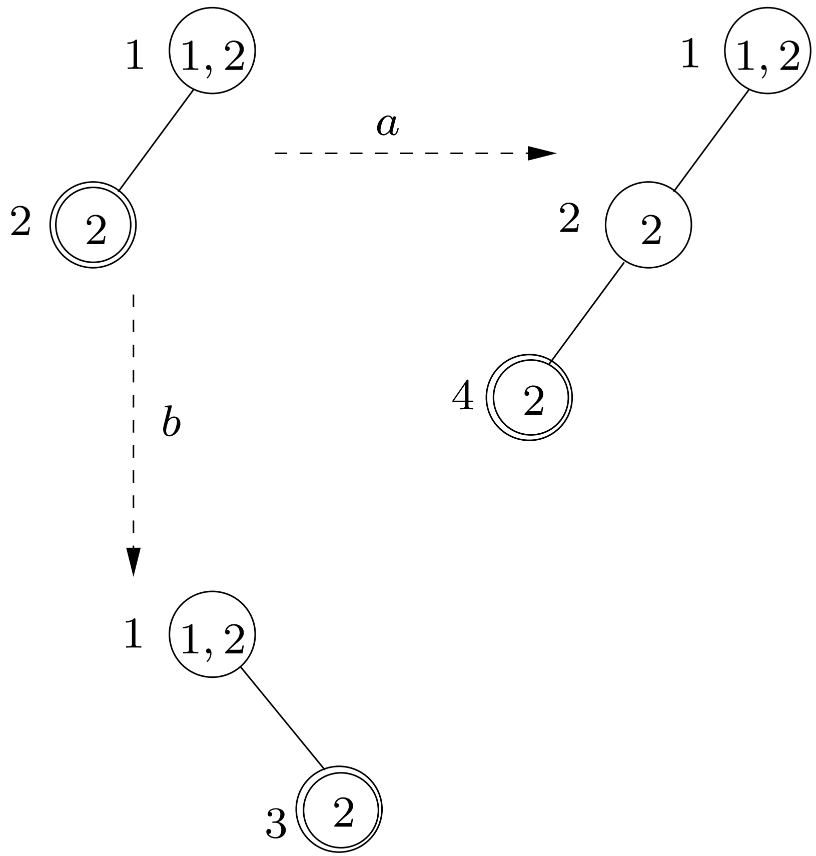




Delete Empty Nodes

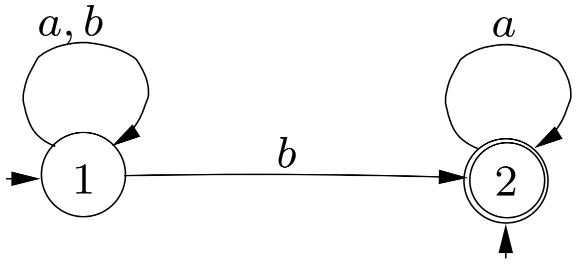
[Step 4] $\langle t_4, m_4 \rangle$ is the tree such that $dom(t_4) = dom(t_3) \setminus \{p \mid t_3(p) = \emptyset\}$ and $m_4 = m_3 \setminus \{p \mid t_3(p) = \emptyset\}$

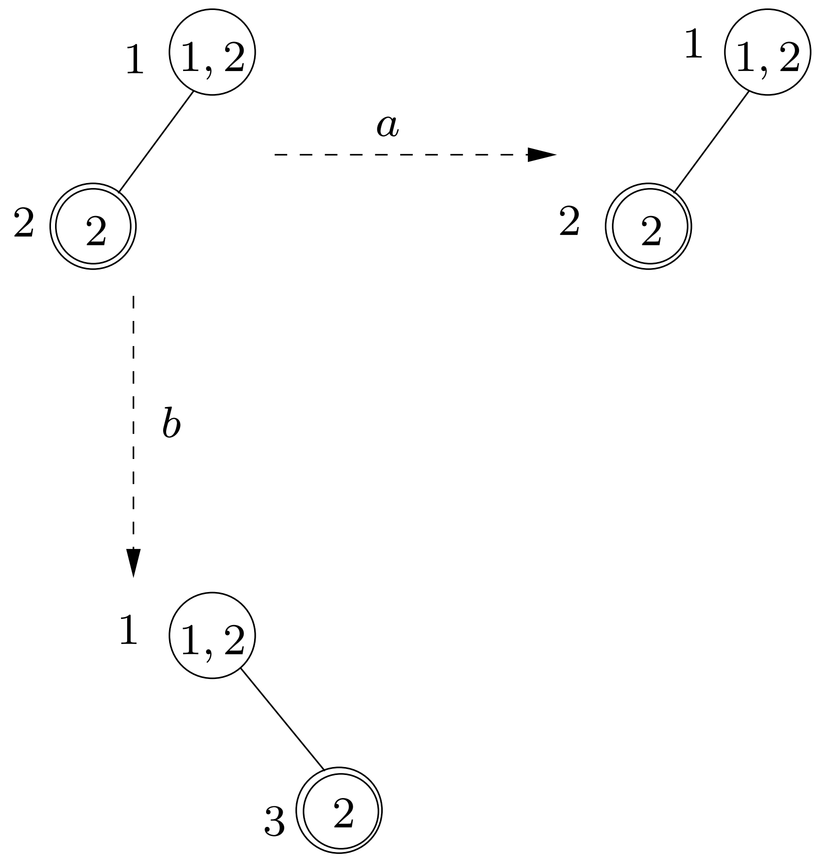




Vertical Merge

[Step 5] $\langle t_5, m_5 \rangle$ is $m_5 = m_4 \cup V$, $dom(t_5) = dom(t_4) \setminus \{q \mid p \in V,\ p < q\}$,
$V = \{p \in dom(t_4) \mid t_4(p) = \bigcup_{p<q} t_4(q)\}$

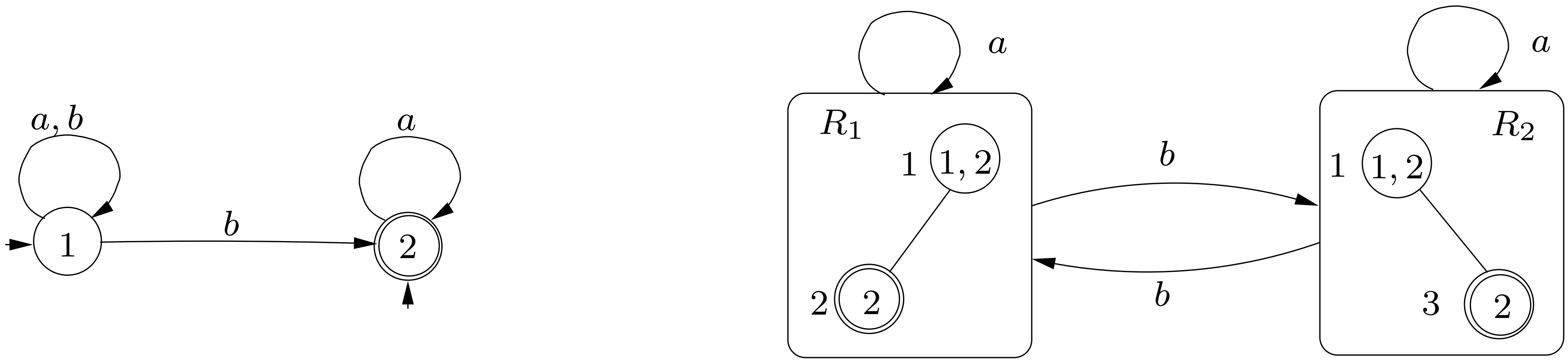




Accepting Condition

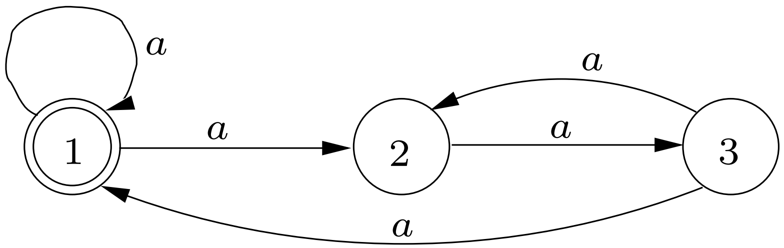
The Rabin accepting condition is defined as $\Omega_B = \{(N_q, P_q) \mid q \in \bigcup_{\langle t,m\rangle \in S_B} dom(t)\}$, where:

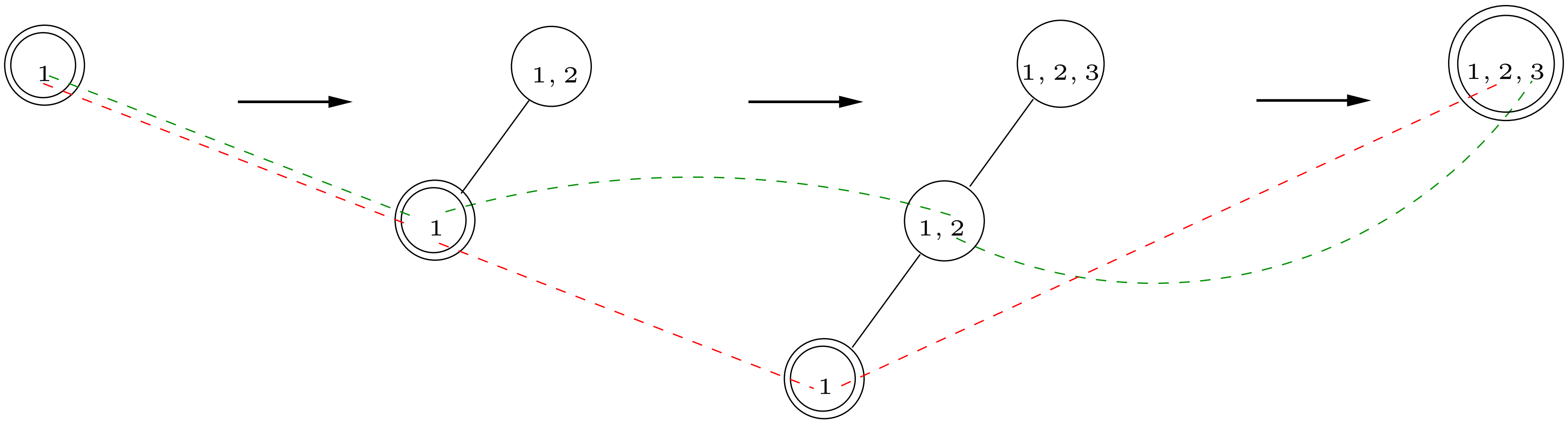
- $N_q = \{\langle t, m\rangle \in S_B \mid q \notin dom(t)\}$
- $P_q = \{\langle t, m\rangle \in S_B \mid q \in m\}$



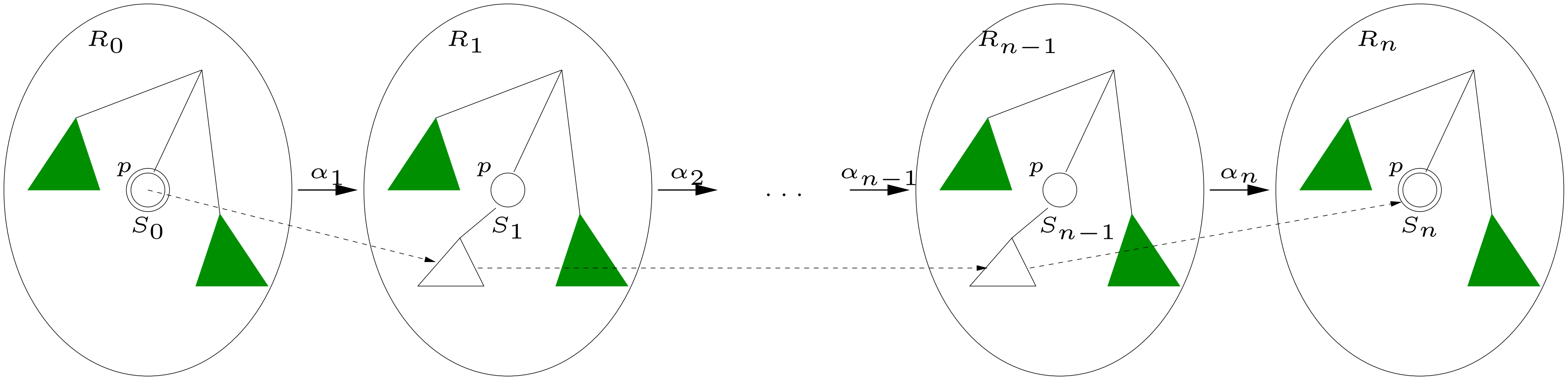
$$\Omega_B = \{(\{R_1\}, \{R_2\}), (\{R_2\}, \{R_1\})\}$$

Example





Correctness of Safra Construction



Lemma 1 *For* $0 \leq i \leq n-1$, $S_{i+1} \subseteq T(S_i, \alpha_{i+1})$. *Moreover, for every* $q \in S_n$, *there is a path in* A *starting in some* $q_0 \in S_0$, *ending in* q *and visiting at least one final state after its origin.*

An infinite accepting path in B corresponds to an infinite accepting path in A (König's Lemma)

Correctness of Safra Construction

Conversely, an infinite accepting path of A over $u = \alpha_0\alpha_1\alpha_2\ldots$

$$\pi \ : \ q_0 \xrightarrow{\alpha_0} q_1 \xrightarrow{\alpha_1} q_2 \ldots$$

corresponds to a unique infinite path of B:

$$i_B = R_0 \xrightarrow{\alpha_0} R_1 \xrightarrow{\alpha_1} R_2 \ldots$$

where each q_i belongs to the root of R_i

If the root is marked infinitely often, then u is accepted. Otherwise, let n_0 be the largest number such that the root is marked in R_{n_0}. Let $m > n_0$ be the smallest number such that $q_m \in F$ is repeated infinitely often in π.

Since $q_m \in F$ it appears in a child of the root. If it appears always on the same position p_m, then the path is accepting. Otherwise it appears to the left of p_m from some n_1 on (step 3). This left switch can only occur a finite number of times.

Complexity of the Safra Construction

Given a Büchi automaton with n states, how many states we need for an equivalent Rabin automaton?

- The upper bound is $2^{\mathcal{O}(n \log n)}$ states
- The lower bound is of at least $n!$ states

Maximum Number of Safra Trees

Each Safra tree has at most n nodes.

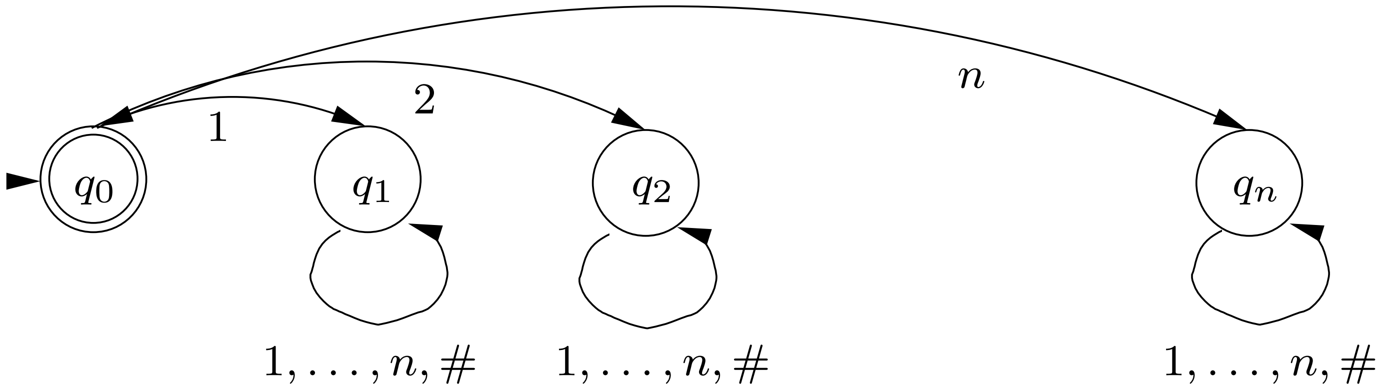
A Safra tree $\langle t, m \rangle$ can be uniquely described by the functions:

- $S \to \{0, \ldots, n\}$ gives for each $s \in S$ the characteristic position $p \in dom(t)$ such that $s \in t(p)$, and s does not appear below p
- $\{1, \ldots, n\} \to \{0, 1\}$ is the marking function
- $\{1, \ldots, n\} \to \{0, \ldots, n\}$ is the parent function
- $\{1, \ldots, n\} \to \{0, \ldots, n\}$ is the older brother function

Altogether we have at most $(n+1)^n \cdot 2^n \cdot (n+1)^n \cdot (n+1)^n \leq (n+1)^{4n}$ Safra trees, hence the upper bound is $2^{\mathcal{O}(n \log n)}$.

The Language L_n

$\Sigma = \{1, \ldots, n, \#\}$



$$(3\#32\#21\#1)^\omega \in L_3$$
$$(312\#)^\omega \notin L_3$$

$\alpha \in L_n$ <u>if</u> there exist $i_1, \ldots, i_n \in \{1, \ldots, n\}$ such that

- $\alpha_k = i_1$ is the first occurrence of i_1 in α and $q_0 \xrightarrow{\alpha_0 \ldots \alpha_k} q_{i_1}$
- the pairs $i_1 i_2, i_2 i_3, \ldots, i_n i_1$ appear infinitely often in α.

The Language L_n

Lemma 2 *(Permutation) For each permutation* $i_1, i_2, \ldots, i_n$ *of* $1, 2, \ldots, n$*, the infinite word* $(i_1 i_2 \ldots i_n \#)^\omega \notin L_n$.

Lemma 3 *(Union) Let* $A = (S, i, T, \Omega)$ *be a Rabin automaton with* $\Omega = \{\langle N_1, P_1 \rangle, \ldots, \langle N_k, P_k \rangle\}$ *and* ρ_1, ρ_2, ρ *be runs of* A *such that*

$$\inf(\rho_1) \cup \inf(\rho_2) = \inf(\rho)$$

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

Proving the $n!$ Lower Bound

Suppose that A recognizes L_n. We need to show that A has $\geq n!$ states.

Let $\alpha = i_1, i_2, \ldots, i_n$ and $\beta = j_1, j_2, \ldots, j_n$ be two permutations of $1, 2, \ldots, n$. Then the words $(i_1 i_2 \ldots i_n \#)^\omega$ and $(j_1 j_2 \ldots j_n \#)^\omega$ are not accepted.

Let ρ_α, ρ_β be the non-accepting runs of A over α and β, respectively.

Claim 1 $\mathrm{inf}(\rho_\alpha) \cap \mathrm{inf}(\rho_\beta) = \emptyset$

Then A must have $\geq n!$ states, since there are $n!$ permutations.

Proving the $n!$ Lower Bound

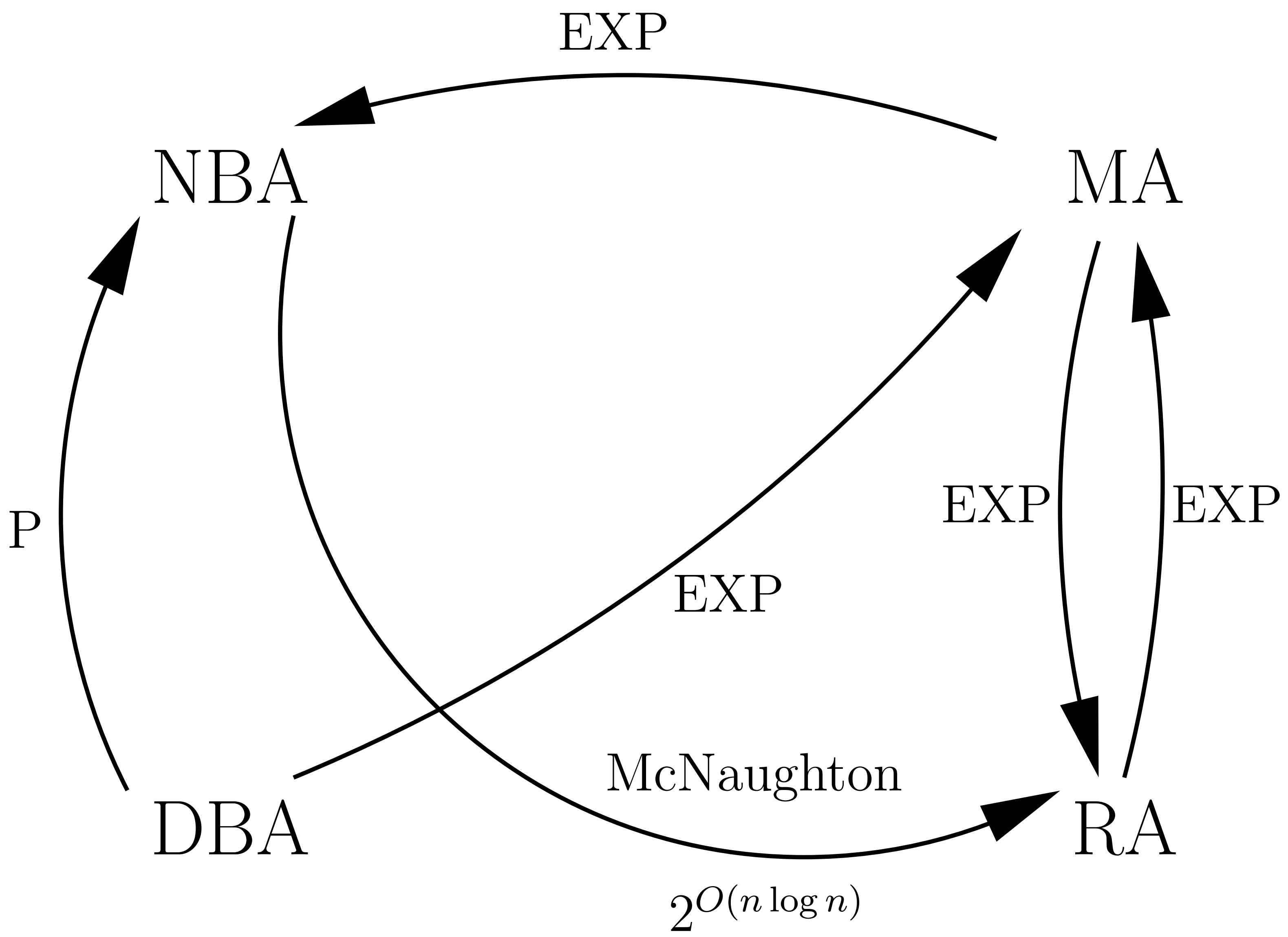
By contradiction, assume $q \in \inf(\rho_\alpha) \cap \inf(\rho_\beta)$. Then we can build a run ρ such that $\inf(\rho) = \inf(\rho_1) \cup \inf(\rho_2)$ and α, β appear infinitely often. By the union lemma, ρ is not accepting.

$$
\begin{array}{cccccccccc}
i_1 & \ldots & i_{k-1} & i_k & i_{k+1} & \ldots & i_{l-1} & i_l & \ldots & i_n \\
= & & = & \neq & & & & & & \\
j_1 & \ldots & j_{k-1} & j_k & j_{k+1} & \ldots & & j_{r-1} & j_r \quad \ldots & j_n
\end{array}
$$

$$
i_k \quad i_{k+1}, \quad \ldots \quad i_l = j_k \quad j_{k+1}, \quad \ldots \quad j_{r-1}, \quad j_r = i_k
$$

The new word is accepted since the pairs $i_k i_{k+1}, \ldots, j_k j_{k+1}, \ldots, j_{r-1} i_k$ occur infinitely often. Contradiction with the fact that ρ is not accepting.

The Big Picture



Linear Temporal Logic

Safety vs. Liveness

- Safety : *something bad never happens*

  A counterexample is an finite execution leading to something bad happening (e.g. an assertion violation).

- Liveness : *something good eventually happens*

  A counterexample is an infinite execution on which nothing good happens (e.g. the program does not terminate).

Verification of Reactive Systems

- Classical verification à la Floyd-Hoare considered three problems:
  - Partial Correctness :

    $$\{\varphi\}\ \mathbf{P}\ \{\psi\} \text{ iff for any } s \models \varphi, \text{ if } P \text{ terminates on } s, \text{ then } P(s) \models \psi$$

  - Total Correctness :

    $$\{\varphi\}\ \mathbf{P}\ \{\psi\} \text{ iff for any } s \models \varphi,\ P \text{ terminates on } s \text{ and } P(s) \models \psi$$

  - Termination :

    $$P \text{ terminates on } s$$

- Need to reason about infinite computations :
  - systems that are in continuous interaction with their environment
  - servers, control systems, etc.
  - e.g. *"every request is eventually answered"*

Reasoning about infinite sequences of states

- Linear Temporal Logic is interpreted on infinite sequences of states
- Each state in the sequence gives an interpretation to the atomic propositions
- Temporal operators indicate in which states a formula should be interpreted

Example 1 *Consider the sequence of states:*

$$\{p, q\}\ \{\neg p, \neg q\}\ (\{\neg p, q\}\ \{p, q\})^{\omega}$$

Starting from position 2*,* q *holds forever.* □

Kripke Structures

Let $\mathcal{P} = \{p, q, r, \ldots\}$ be a finite alphabet of *atomic propositions*.

A *Kripke structure* is a tuple $K = \langle S, s_0, \rightarrow, L \rangle$ where:

- S is a set of *states*,
- $s_0 \in S$ a designated *initial state*,
- $\rightarrow : \; S \times S$ is a *transition relation*,
- $L : S \rightarrow 2^{\mathcal{P}}$ is a *labeling function*.

Paths in Kripke Structures

A *path* in K is an infinite sequence $\pi : s_0, s_1, s_2 \ldots$ such that, for all $i \geq 0$, we have $s_i \to s_{i+1}$.

By $\pi(i)$ we denote the i-th state on the path.

By π_i we denote the suffix $s_i, s_{i+1}, s_{i+2} \ldots$.

$$\boxed{\inf(\pi) = \{s \in S \mid s \text{ appears infinitely often on } \pi\}}$$

If S is finite and π is infinite, then $\inf(\pi) \neq \emptyset$.

Linear Temporal Logic: Syntax

The alphabet of LTL is composed of:

- atomic proposition symbols $p, q, r, \ldots,$
- boolean connectives $\neg, \vee, \wedge, \rightarrow, \leftrightarrow,$
- temporal connectives $\bigcirc, \Box, \Diamond, \mathcal{U}, \mathcal{R}.$

The set of LTL formulae is defined inductively, as follows:

- any atomic proposition is a formula,
- if φ and ψ are formulae, then $\neg\varphi$ and $\varphi \bullet \psi$, for $\bullet \in \{\vee, \wedge, \rightarrow, \leftrightarrow\}$ are also formulae.
- if φ and ψ are formulae, then $\bigcirc\varphi$, $\Box\varphi$, $\Diamond\varphi$, $\varphi\mathcal{U}\psi$ and $\varphi\mathcal{R}\psi$ are formulae,
- nothing else is a formula.

Temporal Operators

- $\bigcirc$ is read at the next time (in the next state)
- $\Box$ is read always in the future (in all future states)
- $\Diamond$ is read eventually (in some future state)
- $\mathcal{U}$ is read until
- $\mathcal{R}$ is read releases

Linear Temporal Logic: Semantics

$$
\begin{array}{rcl}
K, \pi \models p & \iff & p \in L(\pi(0)) \\
K, \pi \models \neg\varphi & \iff & K, \pi \not\models \varphi \\
K, \pi \models \varphi \wedge \psi & \iff & K, \pi \models \varphi \text{ and } K, \pi \models \psi \\
K, \pi \models \bigcirc\varphi & \iff & K, \pi_1 \models \varphi \\
K, \pi \models \varphi \mathcal{U} \psi & \iff & \text{there exists } k \in \mathbb{N} \text{ such that } K, \pi_k \models \psi \\
 & & \text{and } K, \pi_i \models \varphi \text{ for all } 0 \leq i < k
\end{array}
$$

Derived meanings:

$$
\begin{array}{rcl}
K, \pi \models \Diamond\varphi & \iff & K, \pi \models \top \mathcal{U} \varphi \\
K, \pi \models \Box\varphi & \iff & K, \pi \models \neg\Diamond\neg\varphi \\
K, \pi \models \varphi \mathcal{R} \psi & \iff & K, \pi \models \neg(\neg\varphi \mathcal{U} \neg\psi)
\end{array}
$$

Examples

- p holds throughout the execution of the system (p is invariant) : $\Box p$
- whenever p holds, q is bound to hold in the future : $\Box(p \rightarrow \Diamond q)$
- p holds infinitely often : $\Box\Diamond p$
- p holds forever starting from a certain point in the future : $\Diamond\Box p$
- $\Box(p \rightarrow \bigcirc(\neg q \mathcal{U} r))$ holds in all sequences such that if p is true in a state, then q remains false from the next state and until the first state where r is true, which must occur.
- $p \mathcal{R} q$: q is true unless this obligation is released by p being true in a previous state.

LTL vs. FOL

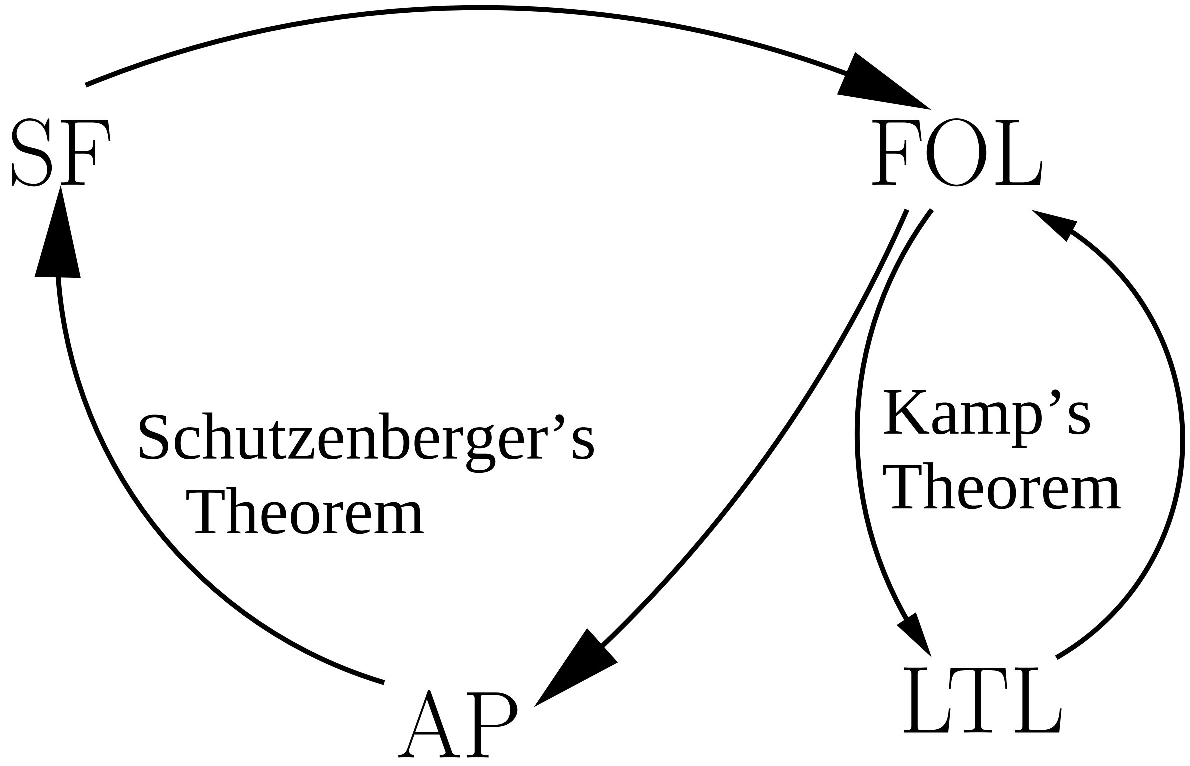
Theorem 2 *LTL and FOL on infinite words have the same expressive power.*

From LTL to FOL:

$$
\begin{array}{rcl}
Tr(q) & = & p_q(t) \\
Tr(\neg\varphi) & = & \neg Tr(\varphi) \\
Tr(\varphi \wedge \psi) & = & Tr(\varphi) \wedge Tr(\psi) \\
Tr(\bigcirc\varphi) & = & Tr(\varphi)[t+1/t] \\
Tr(\varphi \mathcal{U} \psi) & = & \exists x \,.\, Tr(\psi)[x/t] \wedge \forall y \,.\, y < x \rightarrow Tr(\varphi)[y/t]
\end{array}
$$

The direction from FOL to LTL is known as Kamp's Theorem.

The Big Picture



LTL Model Checking

System verification using LTL

- Let K be a model of a reactive system (finite computations can be turned into infinite ones by repeating the last state infinitely often)

- Given an LTL formula φ over a set of atomic propositions $\mathcal{P}$, specifying all bad behaviors, we build a Büchi automaton A_φ that accepts all sequences over $2^{\mathcal{P}}$ satisfying φ.

  Q: Since LTL $\subset$ S1S, this automaton can be built, so why bother?

- Check whether $\mathcal{L}(A_\varphi) \cap \mathcal{L}(K) = \emptyset$. In case it is not, we obtain a counterexample.

Generalized Büchi Automata

Let $\Sigma = \{a, b, \ldots\}$ be a finite alphabet.

A *generalized Büchi automaton* (GBA) over Σ is $A = \langle S, I, T, \mathcal{F} \rangle$, where:

- S is a finite set of *states*,
- $I \subseteq S$ is a set of *initial states*,
- $T \subseteq S \times \Sigma \times S$ is a *transition relation*,
- $\mathcal{F} = \{F_1, \ldots, F_k\} \subseteq 2^S$ is a set of *sets of final states*.

A run π of a GBA is said to be *accepting* iff, for all $1 \leq i \leq k$, we have

$$\inf(\pi) \cap F_i \neq \emptyset$$

GBA and BA are equivalent

Let $A = \langle S, I, T, \mathcal{F} \rangle$, where $\mathcal{F} = \{F_1, \ldots, F_k\}$.

Build $A' = \langle S', I', T', F' \rangle$:

- $S' = S \times \{1, \ldots, k\}$,
- $I' = I \times \{1\}$,
- $(\langle s, i \rangle, a, \langle t, j \rangle) \in T'$ iff $(s, t) \in T$ and:
  - $j = i$ if $s \notin F_i$,
  - $j = (i \mod k) + 1$ if $s \in F_i$.
- $F' = F_1 \times \{1\}$.

The idea of the construction

Let $K = \langle S, s_0, \to, L \rangle$ be a Kripke structure over a set of atomic propositions $\mathcal{P}$, $\pi : \mathbb{N} \to S$ be an infinite path through K, and φ be an LTL formula.

To determine whether $K, \pi \models \varphi$, we label π with sets of subformulae of φ in a way that is compatible with LTL semantics.

Closure

Let φ be an LTL formula written in negation normal form.

The *closure* of φ is the set $Cl(\varphi) \in 2^{\mathcal{L}(LTL)}$:

- $\varphi \in Cl(\varphi)$
- $\bigcirc\psi \in Cl(\varphi) \Rightarrow \psi \in Cl(\varphi)$
- $\psi_1 \bullet \psi_2 \in Cl(\varphi) \Rightarrow \psi_1, \psi_2 \in Cl(\varphi)$, for all $\bullet \in \{\wedge, \vee, \mathcal{U}, \mathcal{R}\}$.

Example 2 $Cl(\Diamond p) = Cl(\top \mathcal{U} p) = \{\Diamond p, p, \top\}\Box$

Q: What is the size of the closure relative to the size of φ ?

Labeling rules

Given $\pi : \mathbb{N} \to 2^{\mathcal{P}}$ and φ, we define $\tau : \mathbb{N} \to 2^{Cl(\varphi)}$ as follows:

- for $p \in \mathcal{P}$, if $p \in \tau(i)$ then $p \in \pi(i)$, and if $\neg p \in \tau(i)$ then $p \notin \pi(i)$
- if $\psi_1 \wedge \psi_2 \in \tau(i)$ then $\psi_1 \in \tau(i)$ and $\psi_2 \in \tau(i)$
- if $\psi_1 \vee \psi_2 \in \tau(i)$ then $\psi_1 \in \tau(i)$ or $\psi_2 \in \tau(i)$

Labeling rules

$$
\begin{aligned}
\varphi \mathcal{U} \psi \quad &\Longleftrightarrow \quad \psi \vee (\varphi \wedge \bigcirc(\varphi \mathcal{U} \psi)) \\
\varphi \mathcal{R} \psi \quad &\Longleftrightarrow \quad \psi \wedge (\varphi \vee \bigcirc(\varphi \mathcal{R} \psi))
\end{aligned}
$$

- if $\bigcirc\psi \in \tau(i)$ then $\psi \in \tau(i+1)$

- if $\psi_1 \mathcal{U} \psi_2 \in \tau(i)$ then either $\psi_2 \in \tau(i)$, or $\psi_1 \in \tau(i)$ and $\psi_1 \mathcal{U} \psi_2 \in \tau(i+1)$

- if $\psi_1 \mathcal{R} \psi_2 \in \tau(i)$ then $\psi_2 \in \tau(i)$ and either $\psi_1 \in \tau(i)$ or $\psi_1 \mathcal{R} \psi_2 \in \tau(i+1)$

Interpreting labelings

A sequence π satisfies a formula φ if one can find a labeling τ satisfying:

- the labeling rules above

- $\varphi \in \tau(0)$, and

- if $\psi_1 \mathcal{U} \psi_2 \in \tau(i)$, then for some $j \geq i$, $\psi_2 \in \tau(j)$ (the eventuality condition)

Building the GBA $A_\varphi = \langle S, I, T, \mathcal{F} \rangle$

The automaton A_φ is the set of labeling rules + the eventuality condition(s) !

- $\Sigma = 2^{\mathcal{P}}$ is the alphabet
- $S \subseteq 2^{Cl(\varphi)}$, such that, for all $s \in S$:
  - $\varphi_1 \wedge \varphi_2 \in s \Rightarrow \varphi_1 \in s$ and $\varphi_2 \in s$
  - $\varphi_1 \vee \varphi_2 \in s \Rightarrow \varphi_1 \in s$ or $\varphi_2 \in s$
- $I = \{s \in S \mid \varphi \in s\}$,
- $(s, \alpha, t) \in T$ iff:
  - for all $p \in \mathcal{P}$, $p \in s \Rightarrow p \in \alpha$, and $\neg p \in s \Rightarrow p \notin \alpha$,
  - $\bigcirc\psi \in s \Rightarrow \psi \in t$,
  - $\psi_1 \mathcal{U} \psi_2 \in s \Rightarrow \psi_2 \in s$ or $[\psi_1 \in s$ and $\psi_1 \mathcal{U} \psi_2 \in t]$
  - $\psi_1 \mathcal{R} \psi_2 \in s \Rightarrow \psi_2 \in s$ and $[\psi_1 \in s$ or $\psi_1 \mathcal{R} \psi_2 \in t]$

Building the GBA $A_\varphi = \langle S, I, T, \mathcal{F} \rangle$

- for each eventuality $\phi \mathcal{U} \psi \in Cl(\varphi)$, the transition relation ensures that this will appear until the first occurrence of ψ

- it is sufficient to ensure that, for each $\phi \mathcal{U} \psi \in Cl(\varphi)$, one goes infinitely often either through a state in which this does not appear, or through a state in which both $\phi \mathcal{U} \psi$ and ψ appear

- let $\phi_1 \mathcal{U} \psi_1, \ldots \phi_n \mathcal{U} \psi_n$ be the "until" subformulae of φ

  $\mathcal{F} = \{F_1, \ldots, F_n\}$, where:

  $$F_i = \{s \in S \mid \phi_i \mathcal{U} \psi_i \in s \text{ and } \psi_i \in s \text{ or } \phi_i \mathcal{U} \psi_i \notin s\}$$

  for all $1 \leq i \leq n$.