Equivalence of alternating and non-deterministic Büchi automata

Master Project

Rafael Penaloza

s9717668@inf.tu-dresden.de

supervised by Carsten Lutz

TU Dresden

Trees

Labeled k-ary Tree (T, V):

- $T \subseteq \{1, \ldots, k\}^*$ such that if $x \cdot n \in T$:
  - $x \in T$
  - $x \cdot m \in T$ for $0 < m < n$
- $V : T \to \Sigma$

Full if $T = \{1, \ldots, k\}^*$

Path $P \subseteq T$:

- $\epsilon \in P$
- if $x \in P, \exists! i . x \cdot i \in P$.

Non-deterministic Automata

$M = (Q, \Sigma, \Delta, I, F)$:

- Q, Σ finite sets
- $I, F \subseteq Q$
- $\Delta \subseteq Q \times \Sigma \times Q^k$

Run (T, r) on (T, V):

- $r(\epsilon) \in I$
- $(r(x), V(x), r(x \cdot 1), \ldots, r(x \cdot k)) \in \Delta$

Language accepted

Run (T, r) accepting if every path of T satisfies accepting condition:

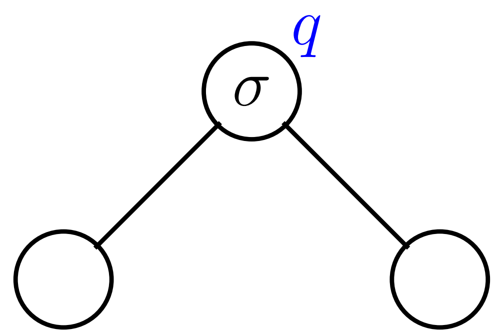
- Büchi
- looping ($F = Q$)

Language accepted by M:

$$\{(T, V) \mid \text{there is an accepting run on } (T, V)\}$$

Intuition

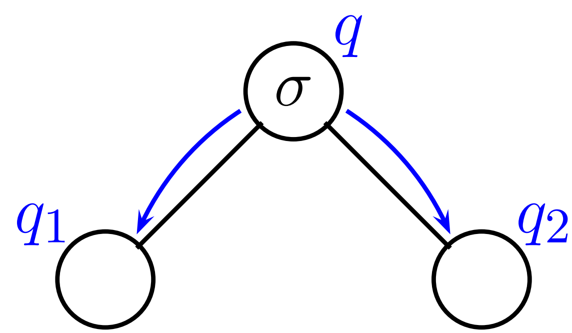
$\Delta \subseteq Q \times \Sigma \times Q^k$
$\delta : Q \times \Sigma \to \mathcal{P}(Q^k)$



$\delta(q, \sigma) = \{(q_1, q_2), (q'_1, q'_2)\}$

Intuition

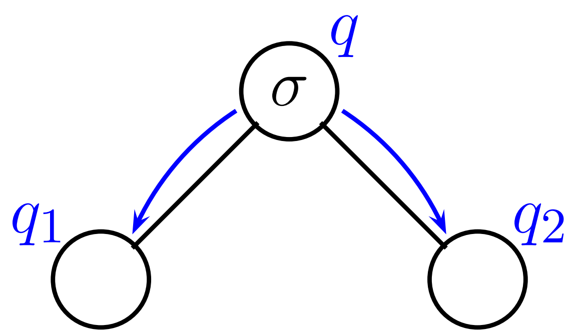
$\Delta \subseteq Q \times \Sigma \times Q^k$
$\delta : Q \times \Sigma \rightarrow \mathcal{P}(Q^k)$



$$\delta(q, \sigma) = \{(q_1, q_2), (q_1', q_2')\}$$

Intuition

$\Delta \subseteq Q \times \Sigma \times Q^k$
$\delta : Q \times \Sigma \to \mathcal{P}(Q^k)$

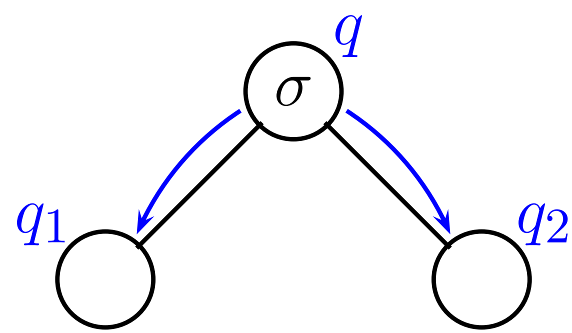


$(1, q_1) \wedge (2, q_2)$

$\delta(q, \sigma) = \{(q_1, q_2), (q'_1, q'_2)\}$

Intuition

$\Delta \subseteq Q \times \Sigma \times Q^k$
$\delta : Q \times \Sigma \to \mathcal{P}(Q^k)$

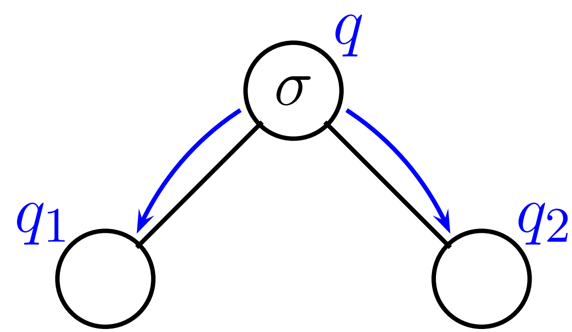


$$(1, q_1) \wedge (2, q_2)$$

$$\delta(q, \sigma) = \{(q_1, q_2), (q_1', q_2')\}$$
$$\sim (1, q_1) \wedge (2, q_2) \vee (1, q_1') \wedge (2, q_2') \in \mathcal{B}^+(\{1, \ldots, k\} \times Q)$$

Intuition

$\Delta \subseteq Q \times \Sigma \times Q^k$
$\delta : Q \times \Sigma \to \mathcal{B}^+(\{0, \ldots, k\} \times Q)$?



$(1, q_1) \wedge (2, q_2)$

$$\delta(q, \sigma) = \{(q_1, q_2), (q_1', q_2')\}$$
$$\sim (1, q_1) \wedge (2, q_2) \vee (1, q_1') \wedge (2, q_2') \in \mathcal{B}^+(\{1, \ldots, k\} \times Q)$$

Alternating Automata

$A = (Q, \Sigma, \delta, q_0, F)$:

- Q, Σ finite sets
- $q_0 \in Q, F \subseteq Q$
- $\delta : Q \times \Sigma \to \mathcal{B}^+(\{0, \ldots, k\} \times Q)$

A has no 0-transitions if $\delta : Q \times \Sigma \to \mathcal{B}^+(\{1, \ldots, k\} \times Q)$.

Run (T_r, r) on (T, V):

- $\epsilon \in T_r, r(\epsilon) = (\epsilon, q_0)$
- if $y \in T_r, r(y) = (x, q)$, then $\exists S$:
  - $S = \{(c_1, q_1), \ldots, (c_n, q_n)\}$ satisfies $\delta(q, V(x))$
  - $y \cdot i \in T_r, r(y \cdot i) = (x \cdot c_i, q_i)$

Language accepted

Run (T_r, r) accepting if every path of T_r satisfies accepting condition:

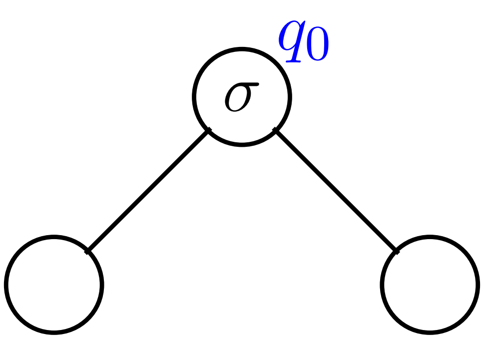
- Büchi
- looping $(F = Q)$

Language accepted by A:

$$\{(T, V) \mid \text{there is an accepting run on } (T, V)\}$$

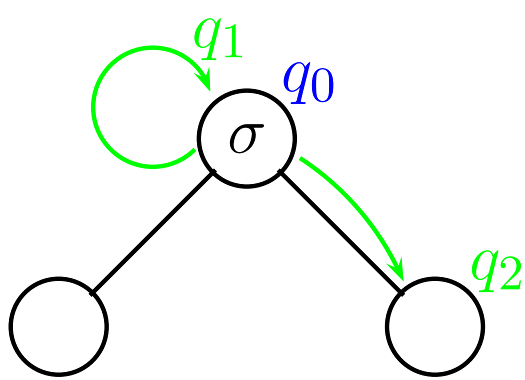
Example (Run)

$$\delta(q_0, \sigma) = (0, q_1) \wedge (2, q_2)$$
$$\delta(q_1, \sigma) = (1, q_3)$$



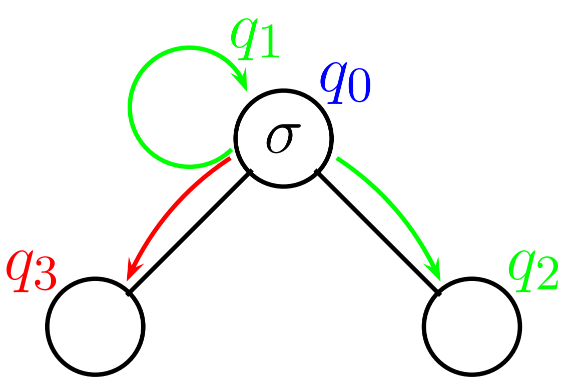
Example (Run)

$$\delta(q_0, \sigma) = (0, q_1) \wedge (2, q_2)$$
$$\delta(q_1, \sigma) = (1, q_3)$$



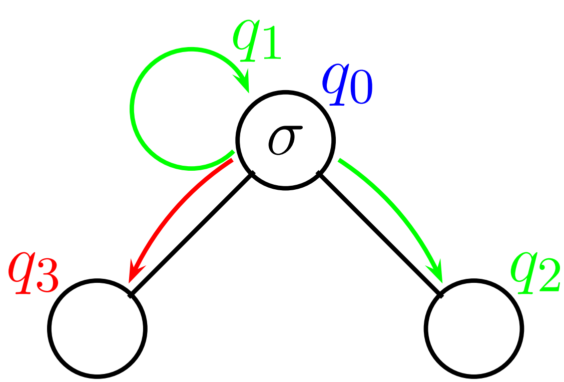
Example (Run)

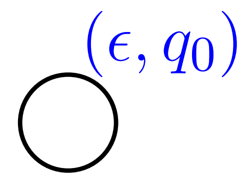
$$\delta(q_0, \sigma) = (0, q_1) \wedge (2, q_2)$$
$$\delta(q_1, \sigma) = (1, q_3)$$



Example (Run)

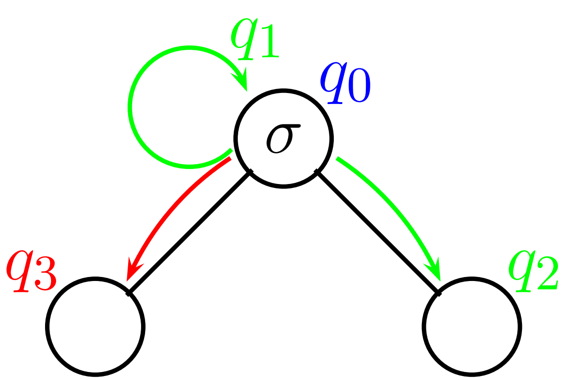
$$\delta(q_0, \sigma) = (0, q_1) \wedge (2, q_2)$$
$$\delta(q_1, \sigma) = (1, q_3)$$

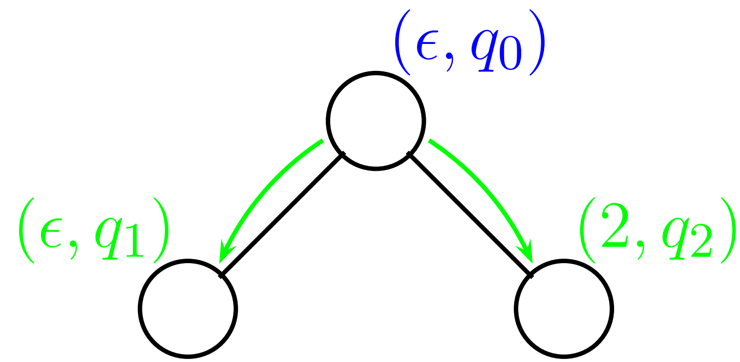




Example (Run)

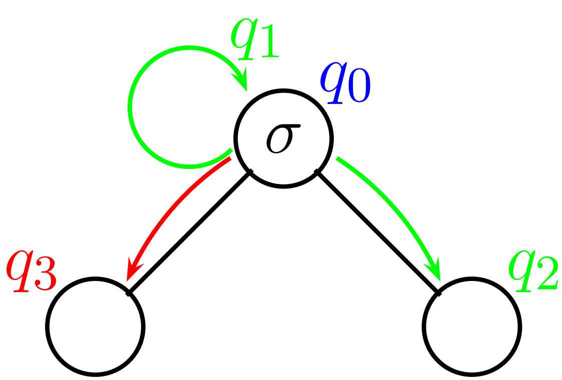
$$\delta(q_0, \sigma) = (0, q_1) \wedge (2, q_2)$$
$$\delta(q_1, \sigma) = (1, q_3)$$

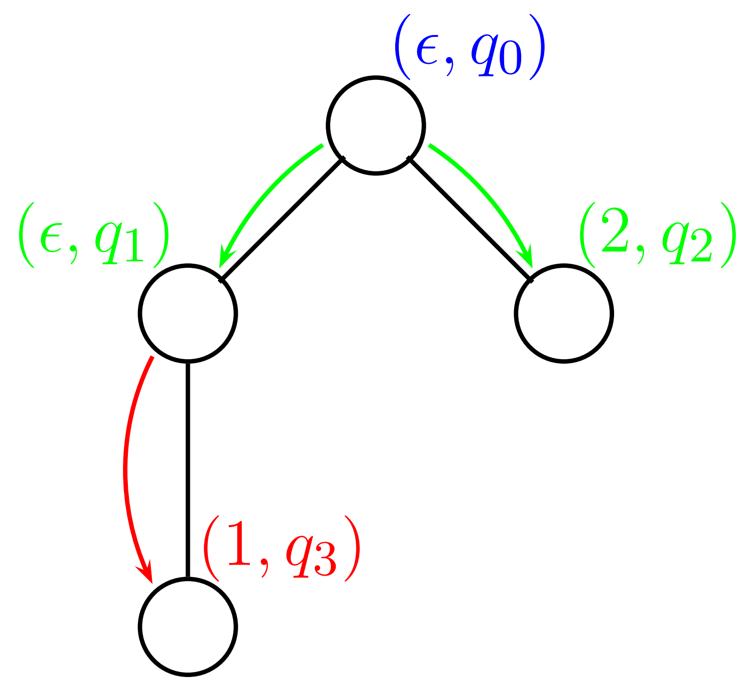




Example (Run)

$$\delta(q_0, \sigma) = (0, q_1) \wedge (2, q_2)$$
$$\delta(q_1, \sigma) = (1, q_3)$$





Strategy trees

Strategy tree τ for A: full k-ary tree with labels from $2^{Q\times\{0,\ldots,k\}\times Q}$.

$state(\zeta) = \{u \mid (u,i,v) \in \zeta\}$.

τ on (T,V) if

- $q_0 \in state(\tau(\epsilon))$
- for $x \in T, q \in state(\tau(x))$:
  - $\{c, q' \mid (q,c,q') \in \tau(x)\}$ satisfies $\delta(q, V(x))$
  - if $(q,c,q') \in \tau(x)$, then $q' \in state(\tau(x \cdot c))$.

Strategy trees (2)

Path β in τ: $(u_1, q_1), (u_2, q_2), \ldots$

- $u_1 = \epsilon$
- $\exists c_i.(q_i, c_i, q_{i+1}) \in \tau(u_i)$ and $u_{i+1} = u_i \cdot c_i$

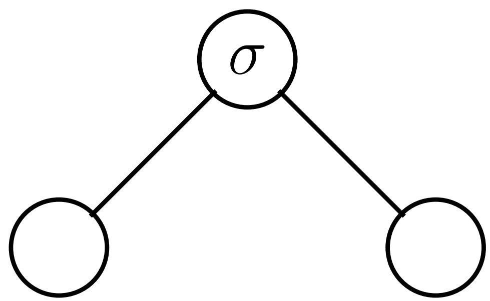
τ accepting if every path is accepting

τ irreducible: if $q \in state(\tau(x \cdot c))$, then $\exists q_1, q_2, \ldots, q_n$ with:

- $(q_1, c, q_2) \in \tau(x)$
- $(q_i, 0, q_{i+1}) \in \tau(x \cdot c)$
- $q_n = q$

Example

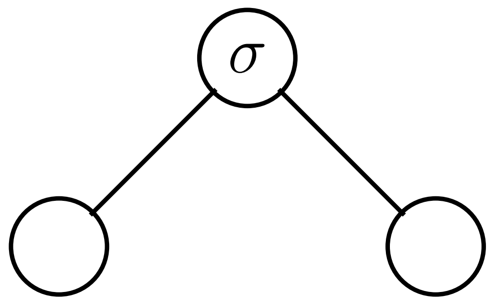
$$\delta(q_0, \sigma) = (0, q_1) \wedge (2, q_2)$$
$$\delta(q_1, \sigma) = (1, q_3)$$



Example

$$\delta(q_0, \sigma) = (0, q_1) \wedge (2, q_2)$$
$$\delta(q_1, \sigma) = (1, q_3)$$

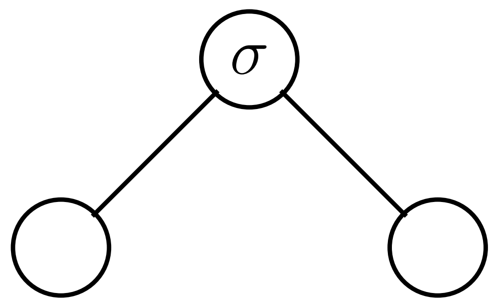
$$\{(q_0, 0, q_1), (q_0, 2, q_2), (q_1, 1, q_3)\}$$



Example

$$\delta(q_0, \sigma) = (0, q_1) \wedge (2, q_2)$$
$$\delta(q_1, \sigma) = (1, q_3)$$

$$\{(q_0, 0, q_1), (q_0, 2, q_2), (q_1, 1, q_3)\}$$



$$\beta = (\epsilon, q_0), (\epsilon, q_1), (1, q_3), \ldots$$
$$\beta = (\epsilon, q_0), (2, q_2), \ldots$$

Auxiliar Result

Theorem [Vardi]: A accepts (T, V) if and only if A has an irreducible accepting strategy tree on (T, V)

Equivalence of looping automata

Given $A = (Q, \Sigma, \delta, q_0)$, construct $M = (S, \Sigma, \Delta, I)$:

- $S = 2^{Q \times \{0,\ldots,k\} \times Q}$
- $I = \{s \mid q_0 \in state(s)\}$
- $\Delta = \{(s, \sigma, s_1, \ldots, s_k \mid$
  - $\{(i, q') \mid (q, i, q') \in s)\}$ satisfies $\delta(q, \sigma)$
  - if $(q, i, q') \in s$ then $q' \in state(s_i)$

  $\}$

A run of M is a strategy tree for A.

Every run (strategy tree) is accepting!

Equivalence of Büchi automata

Given A:

- construct M_1 and M_2
  - M_1 accepts (T, V, τ) if τ is irreducible strategy tree over (T, V)
  - M_2 accepts (T, V, τ) if τ is accepting (assuming is a strategy tree)
- compose M_1 and M_2
- project out τ

The resulting automaton accepts the desired language.

Constructing M_1

Given $A = (Q, \Sigma, \delta, q_0, F)$, construct $M_1 = (2^Q, \Sigma \times S, \Delta, I)$:

- $S = 2^{Q \times \{0,\ldots,k\} \times Q}$
- $I = \{P \mid q_0 \in P\}$
- $\Delta = \{(P, \sigma, s, P_1, \ldots, P_k) \mid$
  - $P = state(s)$
  - $\{(c, q') \mid (q, c, q') \in s\}$ satisfies $\delta(q, \sigma)$
  - $\{q' \mid (q, i, q') \in s\} = P_i$

  $\}$

Same idea as in looping case

Constructing M_2 (idea)

τ is a strategy tree. Is it accepting?

Idea: Two sets of states

- U keeps track of paths not reaching final states
- V keeps track of paths that reached a final state

Constructing M_2 (idea)

τ is a strategy tree. Is it accepting?

Idea: Two sets of states

- U keeps track of paths not reaching final states
- V keeps track of paths that reached a final state

Example:

$\beta = (\epsilon, q_0), (1, q_0), (11, q_1), \ldots$
$\beta = (\epsilon, q_0), (1, q_1), (11, q_0), \ldots$
$\beta = (\epsilon, q_0), (1, q_1), (11, q_1), \ldots$

At node ϵ keep $(\{q_0\}, \emptyset)$;
at $1, (\{q_0\}, \{q_1\})$
at $11, (\emptyset, \{q_0, q_1\})$

$F = \{q_1\}$

Construction of M_2 (restricted case)

No 0-transitions (all paths go downwards)

Given $A = (Q, \Sigma, \delta, q_0, F)$, construct $M = (R, \Sigma \times S, \Delta, I, F)$:

- $R = 2^Q \times 2^Q$
- $S = 2^{Q \times \{1,\ldots,k\} \times Q}$
- $I = 2^Q \times \emptyset$
- $F = \emptyset \times 2^Q$

Construction of M_2 (restricted case)

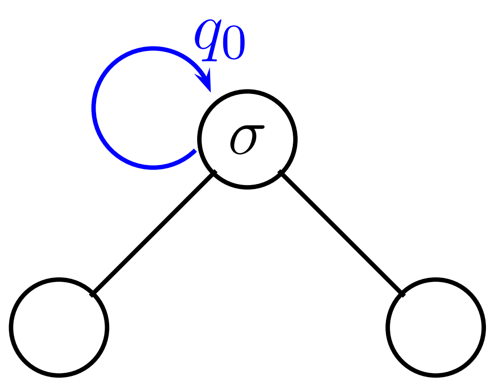
$\Delta = \{(((U,V),\sigma,s,(U_1,V_1),\ldots,(U_k,V_k)) \mid$

- $state(s) \subseteq U \cup V$
- if $U \neq \emptyset$
  - $X_i = \{q' \mid (q,i,q') \in s, q \in U\}$
  - $Y_i = \{q' \mid (q,i,q') \in s, q \in V\}$
  - $U_i = X_i \setminus F$
  - $V_i = Y_i \cup (X_i \cap F)$
- if $U = \emptyset$
  - $Y_i = \{q' \mid (q,i,q') \in s, q \in V\}$
  - $U_i = Y_i \setminus F$
  - $V_i = Y_i \cap F$

$\}$

Towards the General Case

Problem of 0-transitions:
path staying in a node forever (not downwards)



Solution: avoid those paths through transition function

The general case (idea)

$Q_s = \{q \mid \exists q_0, q_1, \ldots$ such that $q_0 = q, (q_i, 0, q_{i+1}) \in s, q_m \notin F\}$

Similar as the restricted case

Take care of paths staying finitely on nodes

No transition if $Q_s \cap (U \cup V) \neq \emptyset$.

Result

For every alternating Büchi automaton exists a non-deterministic Büchi automaton that accepts the same language