A Tight Lower Bound for Determinization of Transition Labeled Büchi Automata

Thomas Colcombet
(joint work with Konrad Zdanowski)

LIAFA, CNRS
University Paris 7 – Denis Diderot

2 avril 2010
Séminaire automate du Liafa

Outline

Theorem

A non-deterministic automaton over finite words of size n can be transformed into (an equivalent) deterministic automaton of size $2^n - 1$.

Furthermore, this construction is optimal: there exists a language accepted by a deterministic automaton of size n which is not accepted by any deterministic automaton of size $2^n - 2$.

Question

What about infinite words (of length ω*)?*

Theorem (McNaughton'1966)

For every Büchi automaton $\mathcal{A}$ on infinite words there exists a deterministic (Müller) automaton $\mathcal{B}$ such that $L(\mathcal{A}) = L(\mathcal{B})$.

Let n be the number of states of a Büchi automaton.

- Safra (1988) described a deterministic *Rabin automaton* with at most $(3.5n)^{2n}$ states (with n Rabin pairs).
- Piterman (2007) described a deterministic *parity automaton* of at most $2n(0.6n)^{2n}$.
- Schewe (2009) described a Rabin automaton with $o((2.66n)^n)$ states (with 2^{n-1} Rabin pairs).
- Schewe (2009) described a Rabin automaton with the condition labeling transitions with $o((1.65n)^n)$ states (with 2^{n-1} Rabin pairs).

Goal

Prove that the latest construction of Schewe is optimal.

Automata over infinite words

Safra's constructions (Schewe's variant)

Games and memory

Our lower bound proof

Conclusion

Automata

A finite automaton is a tuple $\mathcal{A} = (Q, \Sigma, I, \Delta, C)$, where:

- Q is the set of *states*,
- Σ is the *input alphabet*,
- I is the set of *initial states*,
- $\Delta \subseteq Q \times \Sigma \times Q$ is the *transition relation*,
- $C \subseteq \Delta^\omega$ is the *accepting condition*.

A Büchi accepting condition is a language of the form $\{u : (\exists^\infty u_i \in B)\}$ for some $B \subseteq \Delta$.

A Rabin accepting condition is a language of the form $\{u : \exists (I, F) \in R, (\exists^\infty u_i \in I) \wedge \neg(\exists^\infty u_i \in F)\}$ for some $R \subseteq \mathcal{P}(\Delta) \times \mathcal{P}(\Delta)$. Each $(I, F) \in R$ is a Rabin pair.

The automaton is deterministic if for all state p and all letter a, there is at most one transition of the form (p, a, q).

History trees

Fix a set of states Q (of a non-deterministic Büchi automaton) of size n.

An history tree (over state Q) is an unranked ordered tree, each node being labelled by subsets of Q such that:

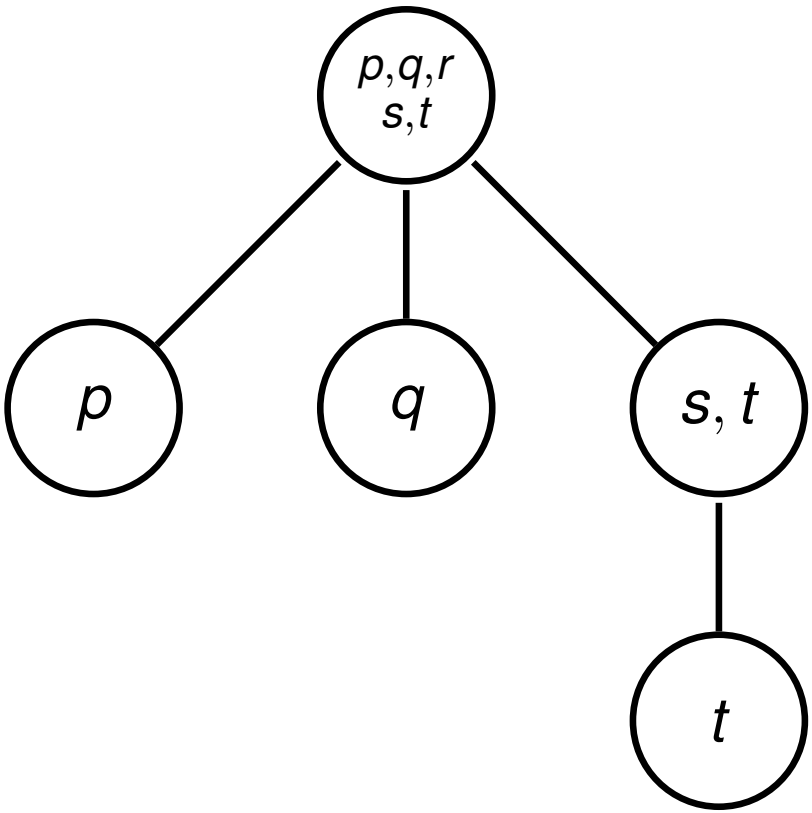
- Labels of children are included in the label of their parent,
- Labels of siblings are disjoint,
- The union of labels of children is strictly included in the label of the parent.

The function $hist(n)$ is the number of such trees.

Proposition (Schewe, Bouvel&Rossin)

$hist(n) \in \Omega(1.64n)^n$ *and* $hist(n) \in o(1.65n)^n$.

Example of an history tree



Safra's construction (Schewe's variant)

Each state of the deterministic Rabin automaton $\mathcal{R}$ is an history tree.
What about the transitions while reading letter $a \in \Sigma$?

1. update the labels of states nodes using the transition function of $\mathcal{B}$
2. create new youngest (i.e. rightmost) children for states obtained from Büchi transitions (in B)
3. eliminate every state which appears also in a label of a left sibling
4. eliminate empty nodes, and mark the position Red
5. collapse every node such that the union of labels of its children equals its label, and mark the position Green.

Rabin condition: A run is accepting if there is a position x which is infinitely often Green and there are only finitely many occurrences of Red at x, above x and at the left siblings of x.

The construction

Theorem (Safra, Schewe)

The automaton $\mathcal{R}$ accepts the same language as $\mathcal{B}$.
The automaton $\mathcal{R}$ has $hist(n)$ states.

The construction

Theorem (Safra, Schewe)

The automaton $\mathcal{R}$ accepts the same language as $\mathcal{B}$.
The automaton $\mathcal{R}$ has $hist(n)$ states.

Theorem (our result)

For all n, there exists an n states Büchi automaton such that no $hist(n) - 1$ state deterministic Rabin automaton accepts the same language.

The construction

Theorem (Safra, Schewe)

The automaton $\mathcal{R}$ accepts the same language as $\mathcal{B}$.
The automaton $\mathcal{R}$ has $hist(n)$ states.

Theorem (our result)

For all n, there exists an n states Büchi automaton such that no $hist(n) - 1$ state deterministic Rabin automaton accepts the same language.

Remark (A difficulty)

The construction is not unique: there is a choice possible for transitions over the same set of states!

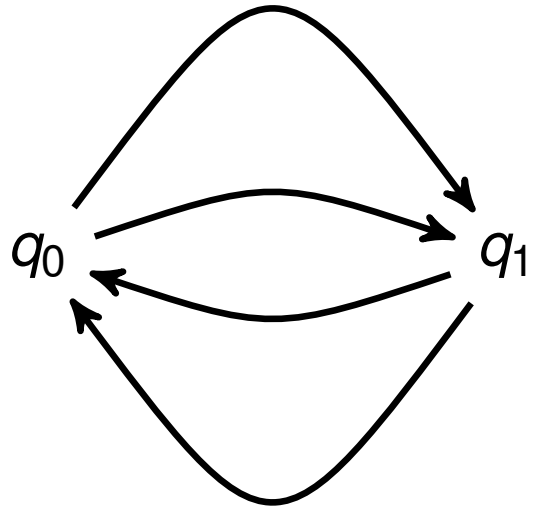
What about other constructions?

Constructions with conditions on state need to further encode the transition informations in the states.

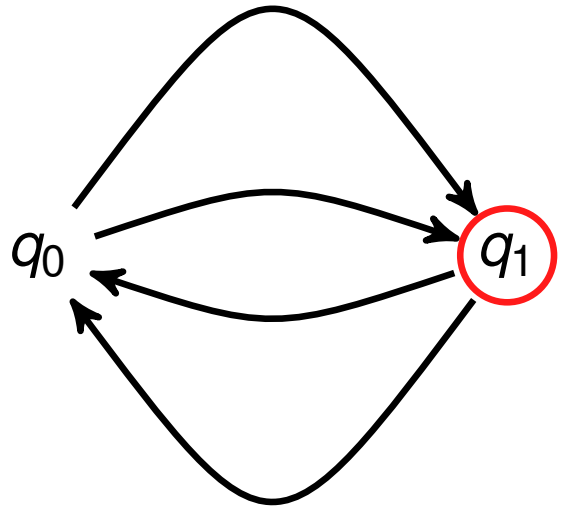
Safra, Piterman and Schewe's construction yielding Parity and n Rabin pairs are obtained by composing with a mechanism of Rabin pair reallocation.

All those modifications increase the number of states of the automaton.

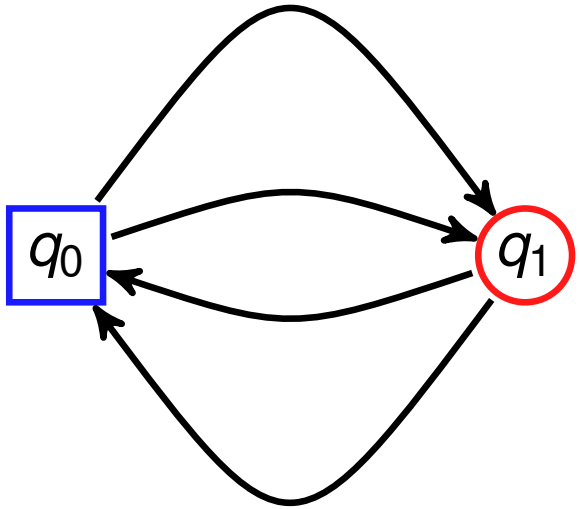
- A game $\mathcal{G} = (V_E, V_A, q_0, \mathit{Move}, \Gamma, L)$ can be seen as a directed graph ($V = V_E \uplus V_A$)



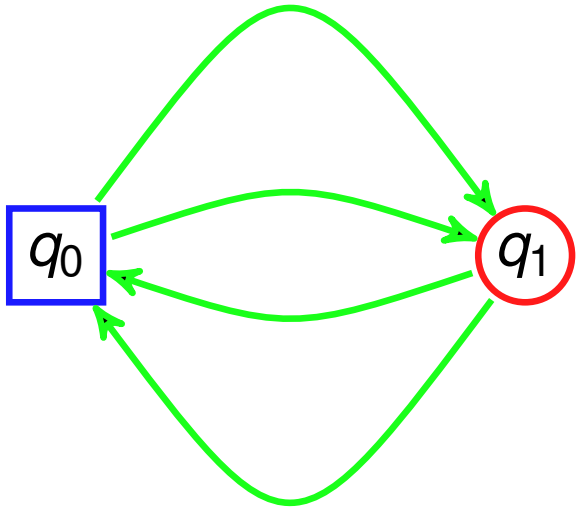
- A game $\mathcal{G} = (V_E, V_A, q_0, \mathit{Move}, \Gamma, L)$ can be seen as a directed graph ($V = V_E \uplus V_A$)
- with nodes V_E belonging to Eve



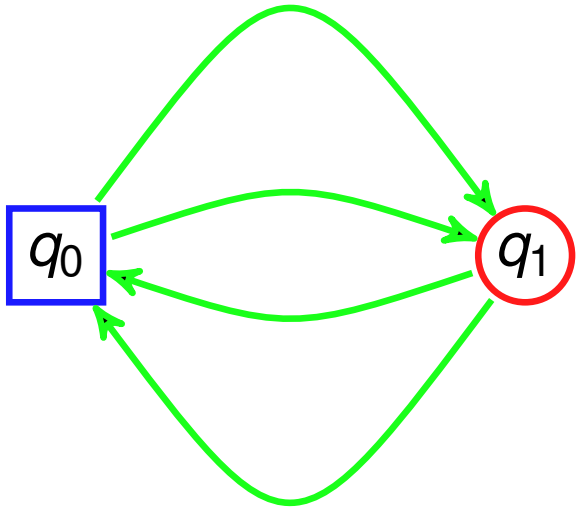
- A game $\mathcal{G} = (V_E, V_A, q_0, \mathit{Move}, \Gamma, L)$ can be seen as a directed graph ($V = V_E \uplus V_A$)
- with nodes V_E belonging to Eve and nodes V_A belonging to Adam.



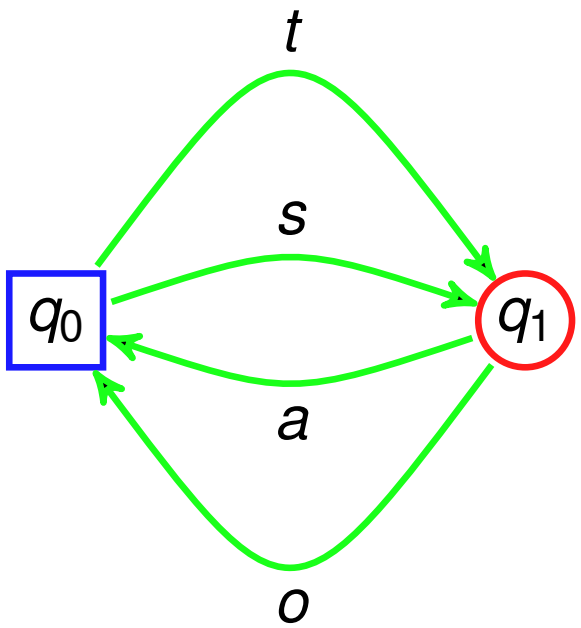
- A game $\mathcal{G} = (V_E, V_A, q_0, Move, \Gamma, L)$ can be seen as a directed graph ($V = V_E \uplus V_A$)
- with nodes V_E belonging to Eve and nodes V_A belonging to Adam.
- Players move sequentially starting from q_0 choosing some edge to go.



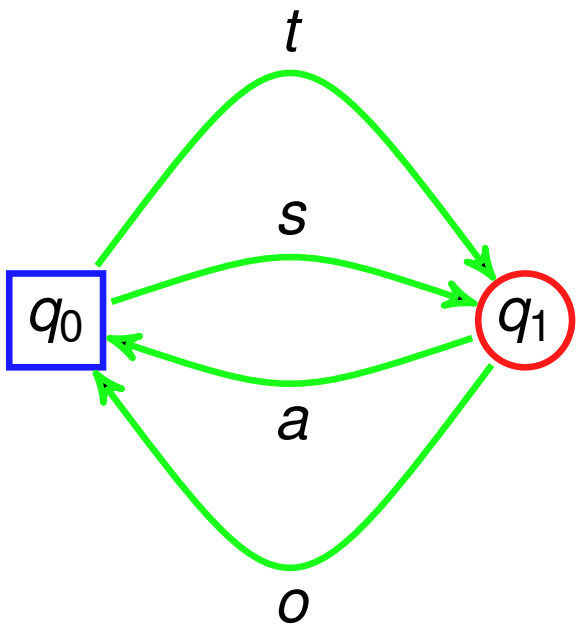
- A game $\mathcal{G} = (V_E, V_A, q_0, \mathit{Move}, \Gamma, L)$ can be seen as a directed graph ($V = V_E \uplus V_A$)
- with nodes V_E belonging to Eve and nodes V_A belonging to Adam.
- Players move sequentially starting from q_0 choosing some edge to go. A player X moves from nodes in V_X.



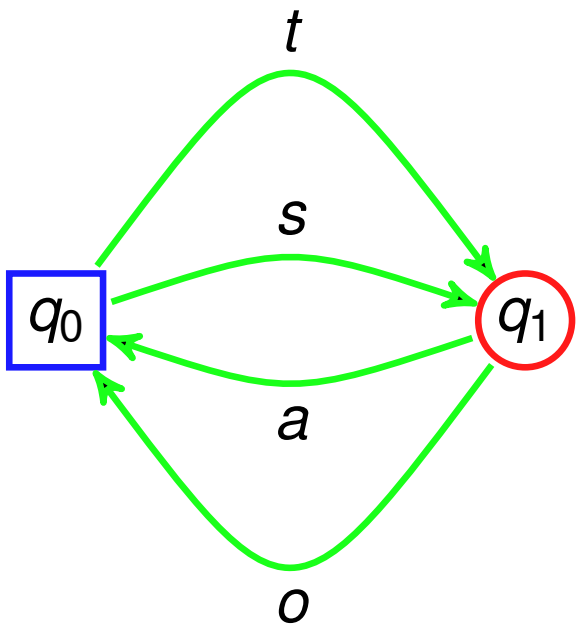
- A game $\mathcal{G} = (V_E, V_A, q_0, Move, \Gamma, L)$ can be seen as a directed graph ($V = V_E \uplus V_A$)
- with nodes V_E belonging to Eve and nodes V_A belonging to Adam.
- Players move sequentially starting from q_0 choosing some edge to go. A player X moves from nodes in V_X.
- During each move they produce some output $a_i \in \Gamma$.



- A game $\mathcal{G} = (V_E, V_A, q_0, Move, \Gamma, L)$ can be seen as a directed graph ($V = V_E \uplus V_A$)
- with nodes V_E belonging to Eve and nodes V_A belonging to Adam.
- Players move sequentially starting from q_0 choosing some edge to go. A player X moves from nodes in V_X.
- During each move they produce some output $a_i \in \Gamma$.
- A play results in an infinite word $a_0 a_1 a_2 \ldots$



- A game $\mathcal{G} = (V_E, V_A, q_0, Move, \Gamma, L)$ can be seen as a directed graph ($V = V_E \uplus V_A$)
- with nodes V_E belonging to Eve and nodes V_A belonging to Adam.
- Players move sequentially starting from q_0 choosing some edge to go. A player X moves from nodes in V_X.
- During each move they produce some output $a_i \in \Gamma$.
- A play results in an infinite word $a_0 a_1 a_2 \ldots$ Eve wins if $a_0 a_1 a_2 \cdots \in L$. $L \subseteq \Gamma$ is the winning condition.



Definition

A strategy for a player X is a function which depends on the finite history of the play which tells player X what move he/she should take (for positions in V_X).
Formally σ_X : Move* $\longrightarrow$ Move.

Definition

A strategy for a player X is a function which depends on the finite history of the play which tells player X what move he/she should take (for positions in V_X).
Formally $\sigma_X : Move^* \rightarrow Move$.

A strategy σ is winning if X wins every play provided that he/she plays according to σ.

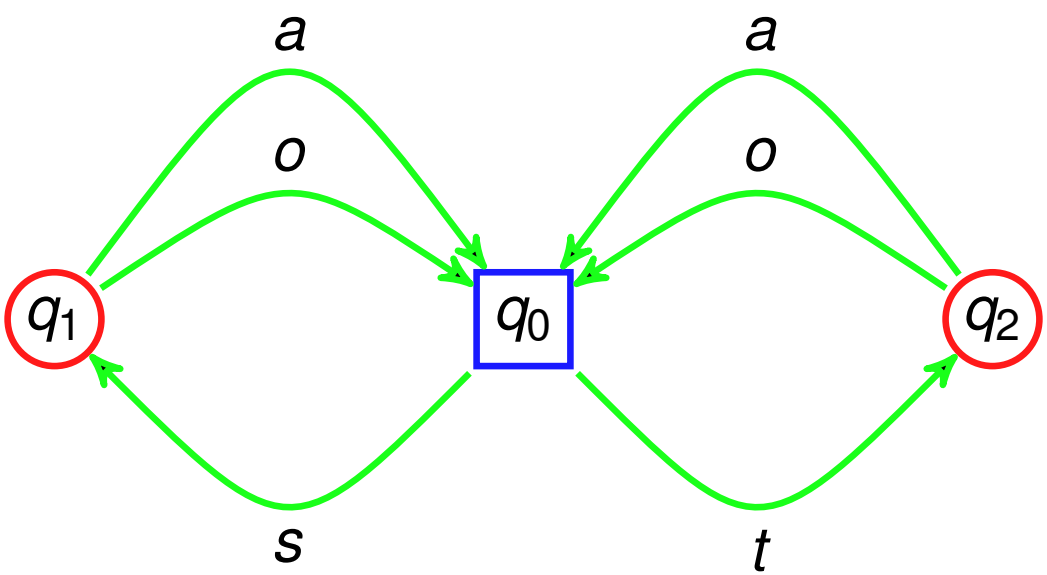
Definition

A strategy for a player X is a function which depends on the finite history of the play which tells player X what move he/she should take (for positions in V_X).
Formally $\sigma_X : Move^* \rightarrow Move$.

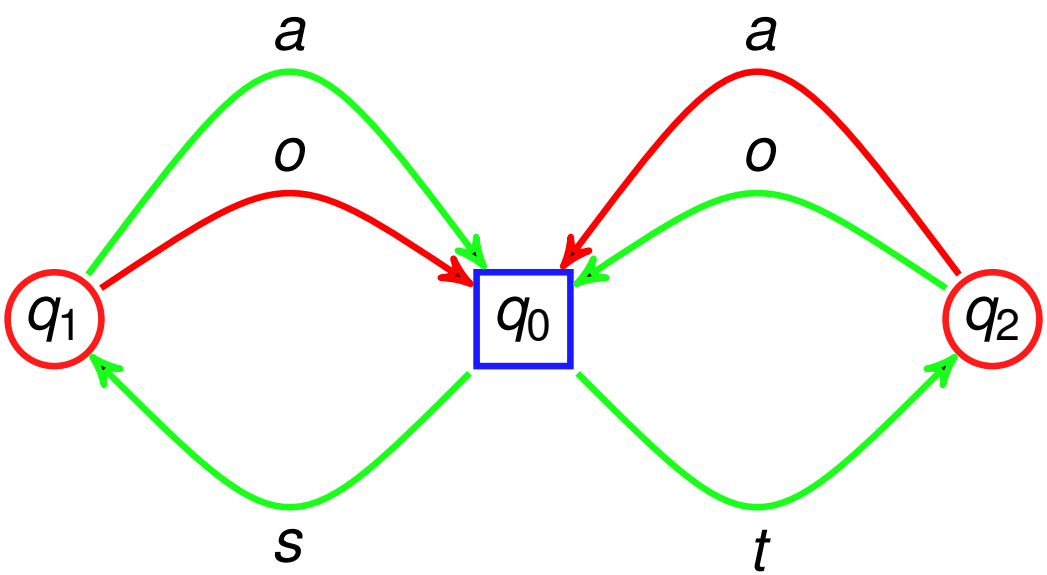
A strategy σ is winning if X wins every play provided that he/she plays according to σ.

A strategy is positional if the choice of σ depends only on the current position.
I.e.: $\sigma_X : V_X \rightarrow Move$.

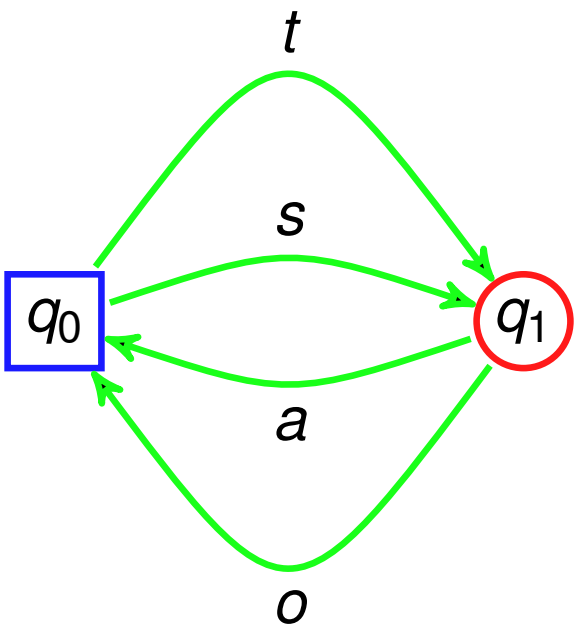
Let the winning condition be $L = (ta + so)^\omega$. Then, Eva wins the game below with a positional strategy.



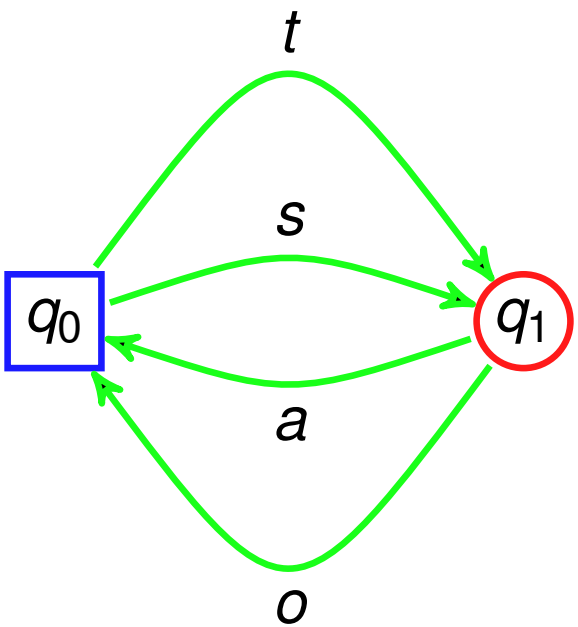
Let the winning condition be $L = (ta + so)^\omega$. Then, Eva wins the game below with a positional strategy.



Let the winning condition be again $L = (ta + so)^\omega$. Eve wins the game below but she has no positional strategy.



Let the winning condition be again $L = (ta + so)^\omega$. Eve wins the game below but she has no positional strategy.



To win Eve needs to remember the last move of Adam.

Definition

A strategy with memory m for Eva is described as $\sigma = (M, update, choice, init)$*, where*

- $|M| = m$,
- $update : M \times Move \to M$,
- $choice : V_E \times M \to Move$ *and* $init \in M$.

Definition

A strategy with memory m for Eva is described as $\sigma = (M, update, choice, init)$*, where*

- $|M| = m$,
- $update : M \times Move \to M$,
- $choice : V_E \times M \to Move$ *and* $init \in M$.

Eva starts the game with memory equal to init.

During a play Eva updates the content of her memory after every played move according to a function update.

Her moves depend only on her actual position and the current content of the memory, following Move.

Definition

A strategy with memory m for Eva is described as $\sigma = (M, update, choice, init)$*, where*

- $|M| = m$,
- $update : M \times Move \to M$,
- $choice : V_E \times M \to Move$ *and* $init \in M$.

Eva starts the game with memory equal to init.

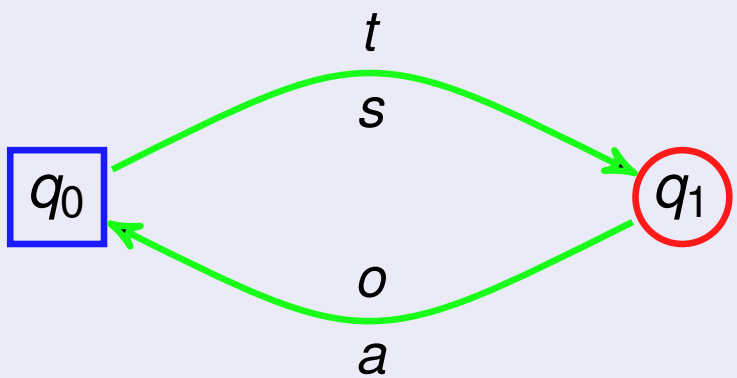
During a play Eva updates the content of her memory after every played move according to a function update.

Her moves depend only on her actual position and the current content of the memory, following Move.

A *positional strategy* corresponds to the case $|M| = 1$.

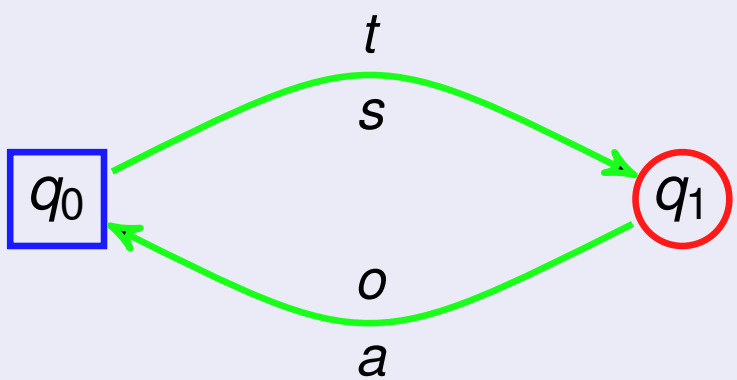
Example

- *Let the winning condition be $L = (ta + so)^\omega$. and a memory for Eva be the set $\{t, s\}$, with arbitrary init.*



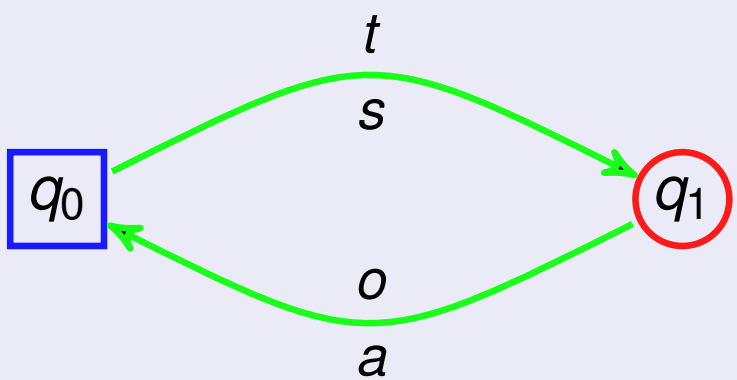
Example

- *Let the winning condition be $L = (ta + so)^\omega$. and a memory for Eva be the set $\{t, s\}$, with arbitrary init.*
- *After each of Adam's move Eva updates the content of its memory by setting the value to the current letter.*



Example

- *Let the winning condition be $L = (ta + so)^\omega$. and a memory for Eva be the set $\{t, s\}$, with arbitrary init.*
- *After each of Adam's move Eva updates the content of its memory by setting the value to the current letter.*
- *At the position q_1, if the content of her memory is t, Eva chooses a move labeled a, otherwise she chooses a move labeled o.*



Definition

A game $\mathcal{G} = (V, V_E, V_A, q_0, Move, \Gamma, L)$ *is called an* L–game.
If L is a Rabin language then we call $\mathcal{G}$ *a* Rabin game.

Definition

A game $\mathcal{G} = (V, V_E, V_A, q_0, Move, \Gamma, L)$ is called an L–game. If L is a Rabin language then we call $\mathcal{G}$ a Rabin game.

Theorem (Klarlund94, Zielonka98)

For every Rabin-game, if Eva wins, she can win using a positional strategy.

Definition

A game $\mathcal{G} = (V, V_E, V_A, q_0, Move, \Gamma, L)$ is called an L–game. If L is a Rabin language then we call $\mathcal{G}$ a Rabin game.

Theorem (Klarlund94, Zielonka98)

For every Rabin-game, if Eva wins, she can win using a positional strategy.

Corollary

Let L be accepted by a deterministic Rabin automaton with n states. If Eva wins an L-game then she wins with memory n.

Definition

A game $\mathcal{G} = (V, V_E, V_A, q_0, Move, \Gamma, L)$ is called an L–game. If L is a Rabin language then we call $\mathcal{G}$ a Rabin game.

Theorem (Klarlund94, Zielonka98)

For every Rabin-game, if Eva wins, she can win using a positional strategy.

Corollary

Let L be accepted by a deterministic Rabin automaton with n states. If Eva wins an L-game then she wins with memory n.

We will use this corollary as follows: if Eva wins an L-game, and requires memory n for that, then every deterministic Rabin automaton for L has size at least n.

The counter-example: the full automaton

Definition (Full Automaton)

Fix n. Let $Q = \{1, \ldots, n\}$.
$\mathcal{F}_n = (Q, \Sigma, Q, \Delta, B)$ *is a full Büchi automaton if*

- $\Sigma = \mathcal{P}(Q \times \{0, 1\} \times Q)$,
- $\Delta = \{(p, A, q) : (p, b, q) \in A\}$,
- $B = \{(p, A, q) \ : \ (p, 1, q) \in A\}$.

The counter-example: the full automaton

Definition (Full Automaton)

Fix n. Let $Q = \{1, \ldots, n\}$.
$\mathcal{F}_n = (Q, \Sigma, Q, \Delta, B)$ is a full Büchi automaton if

- $\Sigma = \mathcal{P}(Q \times \{0, 1\} \times Q)$,
- $\Delta = \{(p, A, q) : (p, b, q) \in A\}$,
- $B = \{(p, A, q) \ : \ (p, 1, q) \in A\}$.

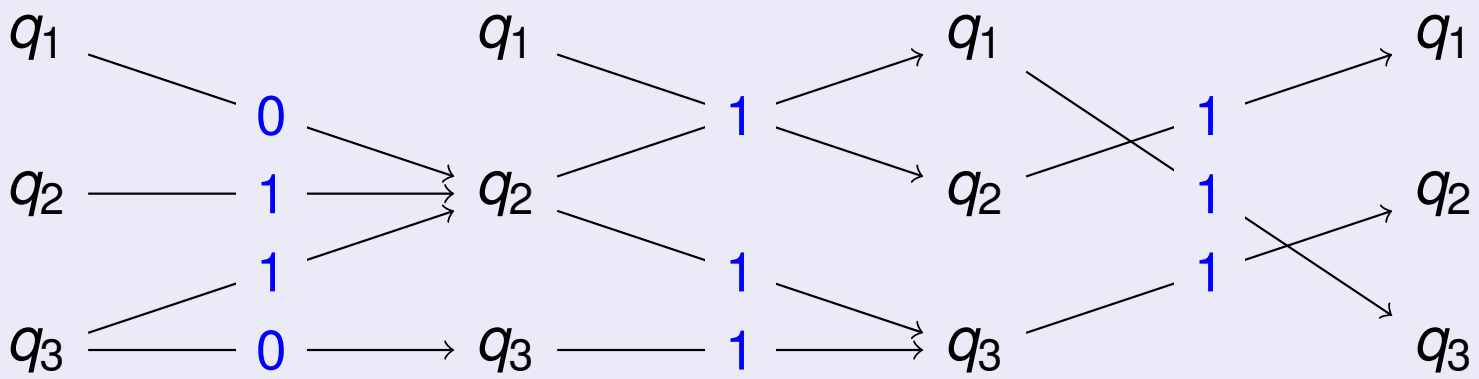
Proposition (Following QiQi Yan)

If $L(\mathcal{F}_n)$ is accepted by a k-state deterministic Rabin automaton, then every Büchi automaton with n states can be accepted by a k-state deterministic Rabin automaton.

Hence $\mathcal{F}_n$ is the best counter-example possible for finding lower bounds! From now, we fix n, and we aim at proving a lower bound for the determinization of $\mathcal{F}_n$.

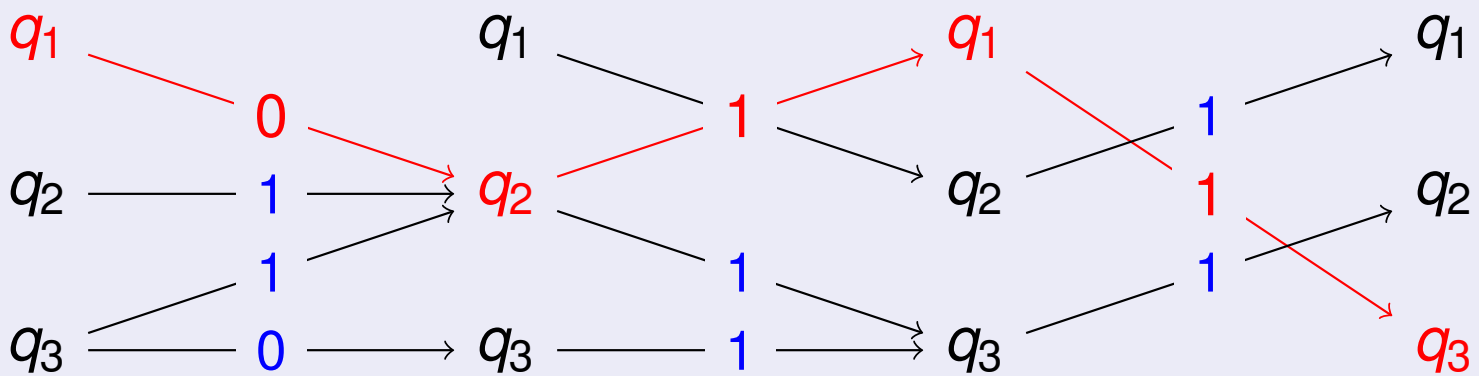
Example

We can write a finite word as:



Example

A possible run:



Good words from an history tree

Let T be a history tree and let u be a word.
Let $\mathcal{R}(T)$ be the state resulting of the reading of u from state T by the automaton $\mathcal{R}$. We write $T \xrightarrow{u} \mathcal{R}(t)$.

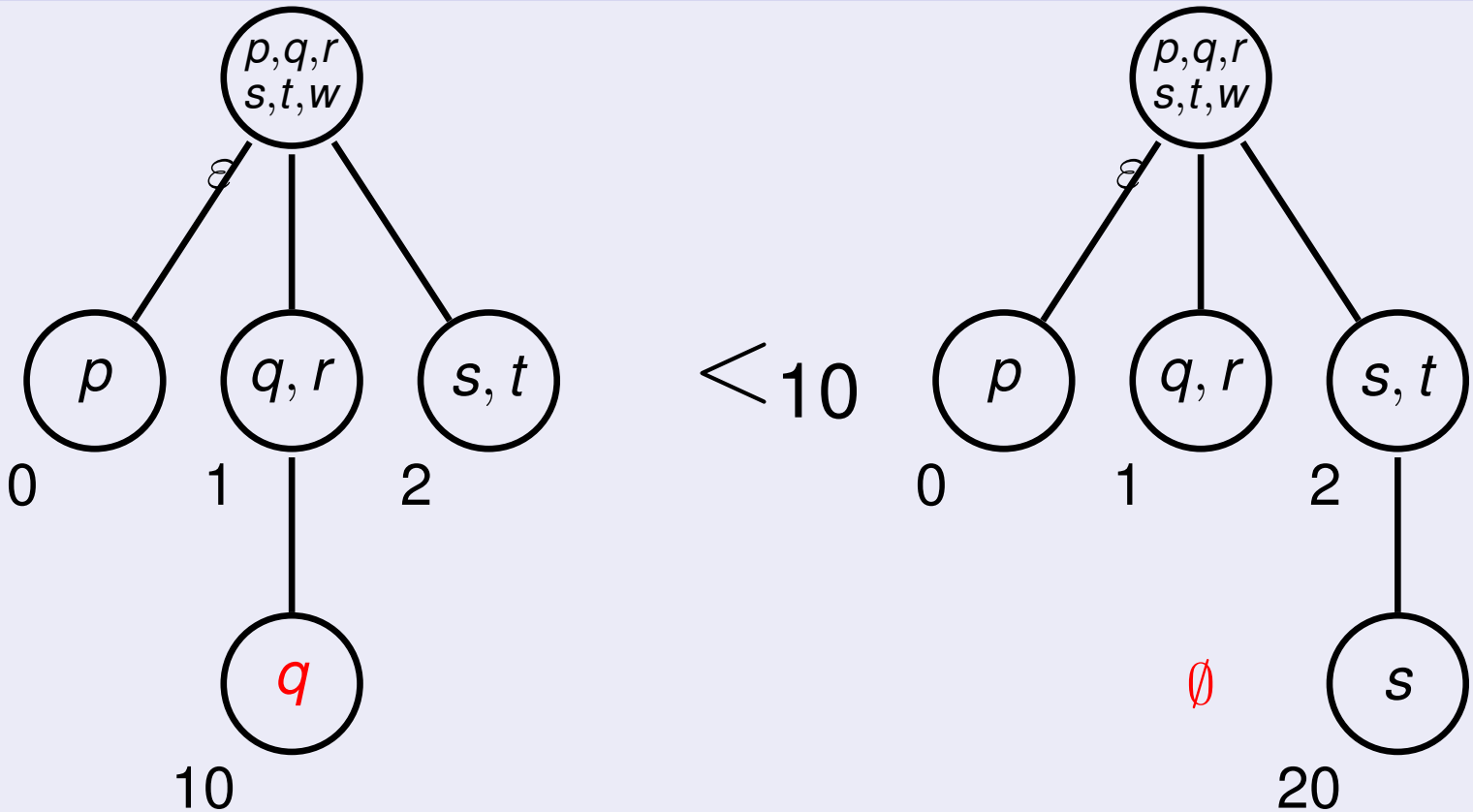
The word u is said good from T if there exist a position x s.t.:

- T and $\mathcal{R}(t)$ coincide on every position lexicograficaly to the left of x
- no position lexicograficaly to the left or equal of x is marked Red when $\mathcal{R}$ reads u from state T
- either $T(x) \subsetneq \mathcal{R}(T)(x)$ or x is marked Green when $\mathcal{R}$ reads u from state T

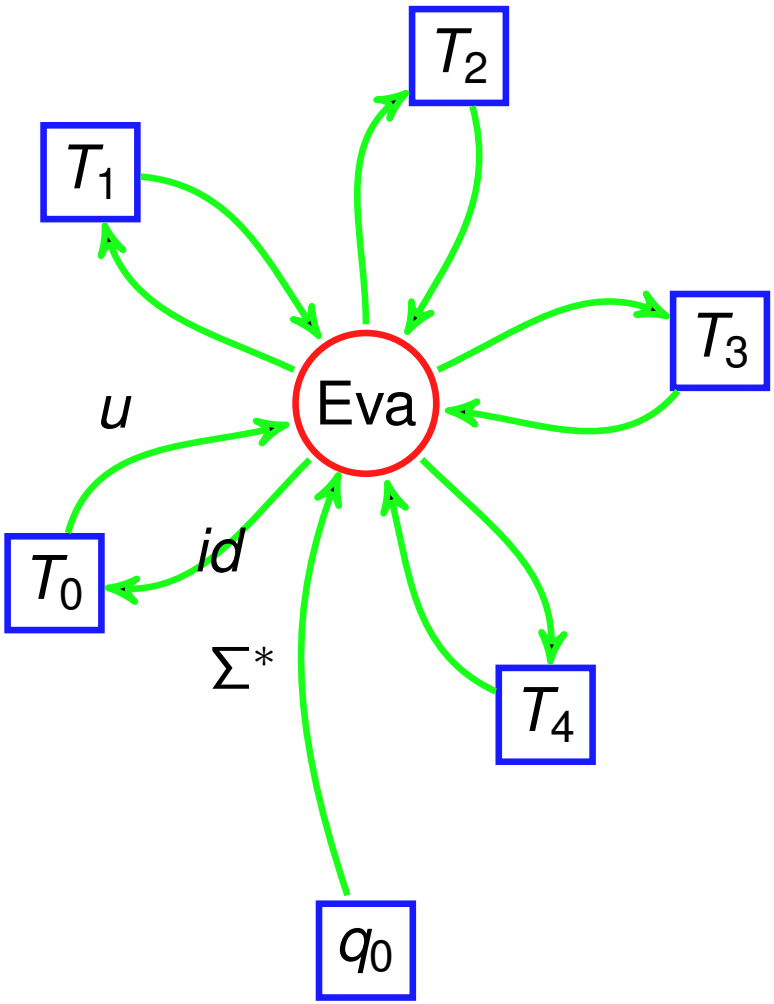
Lemma

Let $T_1 \xrightarrow{u_1} T_2 \xrightarrow{u_2} T_3 \ldots$
If for all i, u_i *is good from* T_i, *then* $\mathcal{R}$ *accepts* $u_1 u_2 \ldots$.

Example

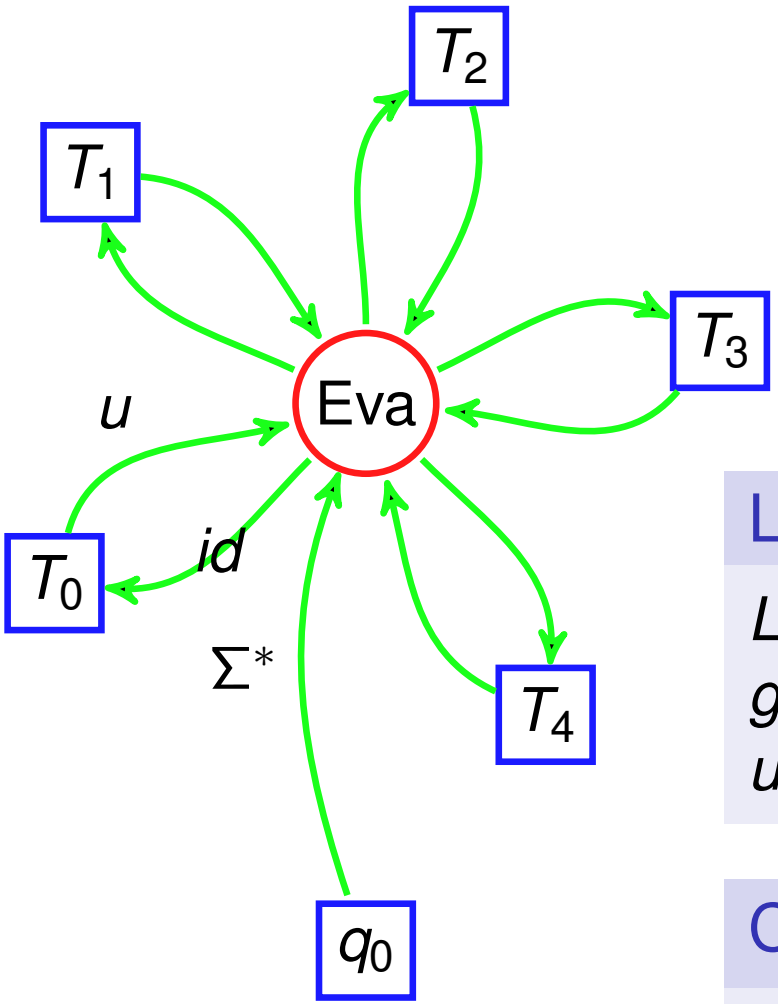


Proving the lower bound: a flower for Eva



- First Adam moves to the center and outputing any word from Σ^*.
- Then, Eva chooses to go to some petal T_i (neutral letter).
- Adam returns to the centre by playing a word u *good* from T_i.

Proving the lower bound: a flower for Eva



- First Adam moves to the center and outputing any word from Σ^*.
- Then, Eva chooses to go to some petal T_i (neutral letter).
- Adam returns to the centre by playing a word u *good* from T_i.

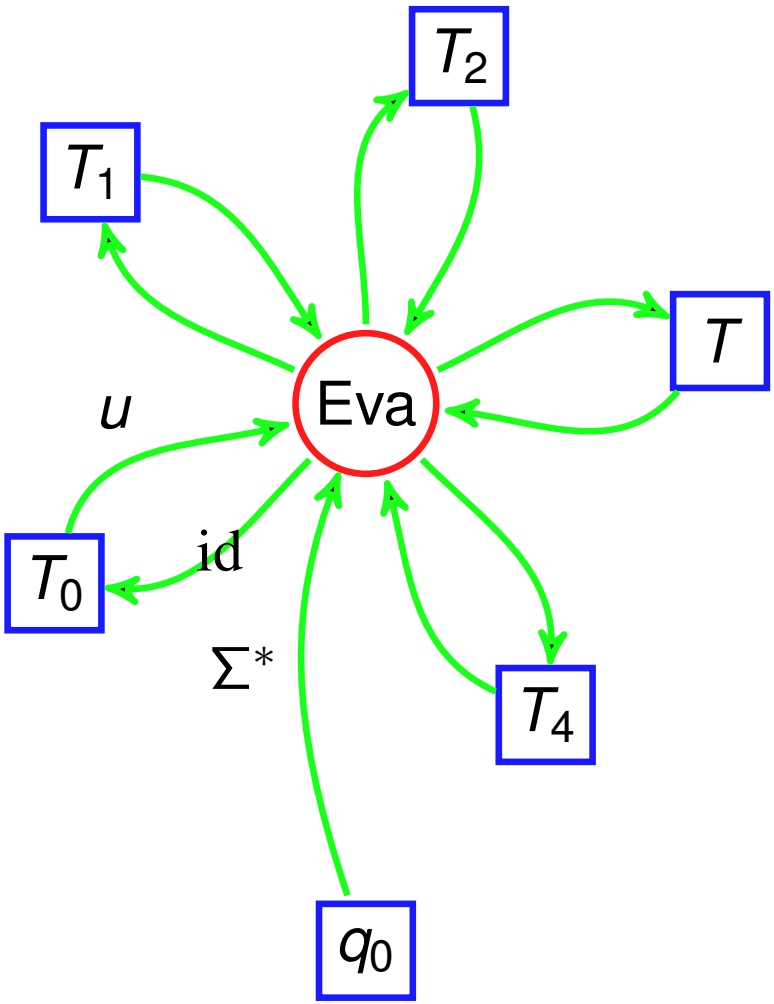
Lemma

Let $T_1 \xrightarrow{u_1} T_2 \xrightarrow{u_2} T_3 \ldots$ *If for all* i, u_i *is good from* T_i, *then* $\mathcal{R}$ *accepts* $u_1 u_2 \ldots$.

Corollary

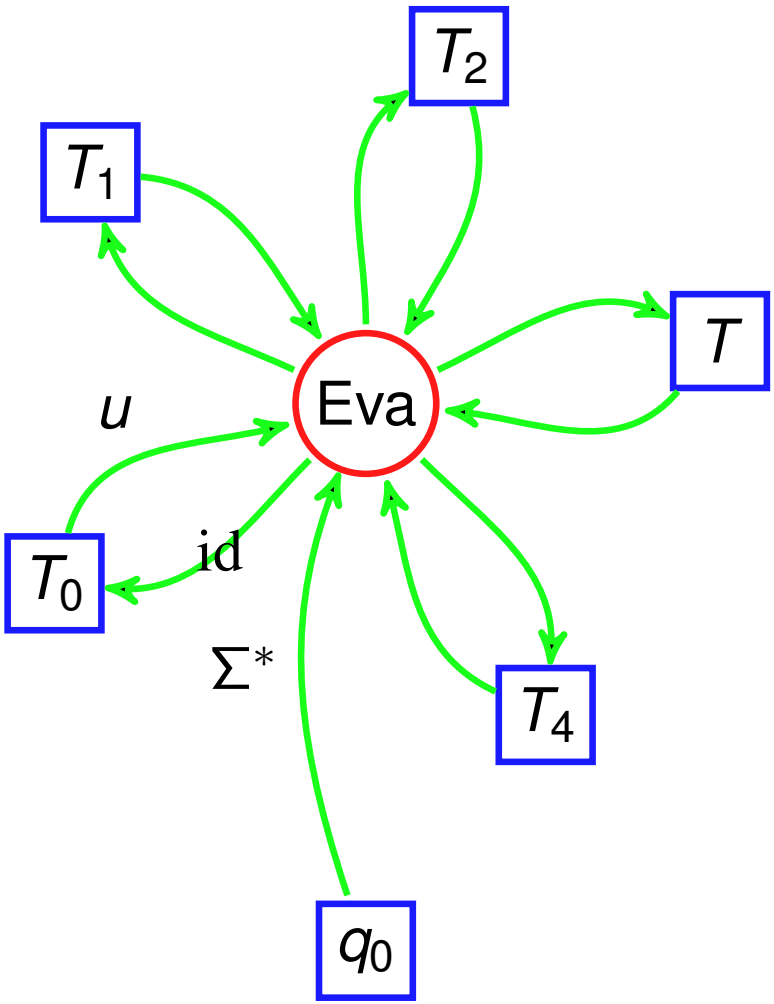
Eva wins the flower game by playing like $\mathcal{R}$.

Assume Eva would win with $hist(n) - 1$ states



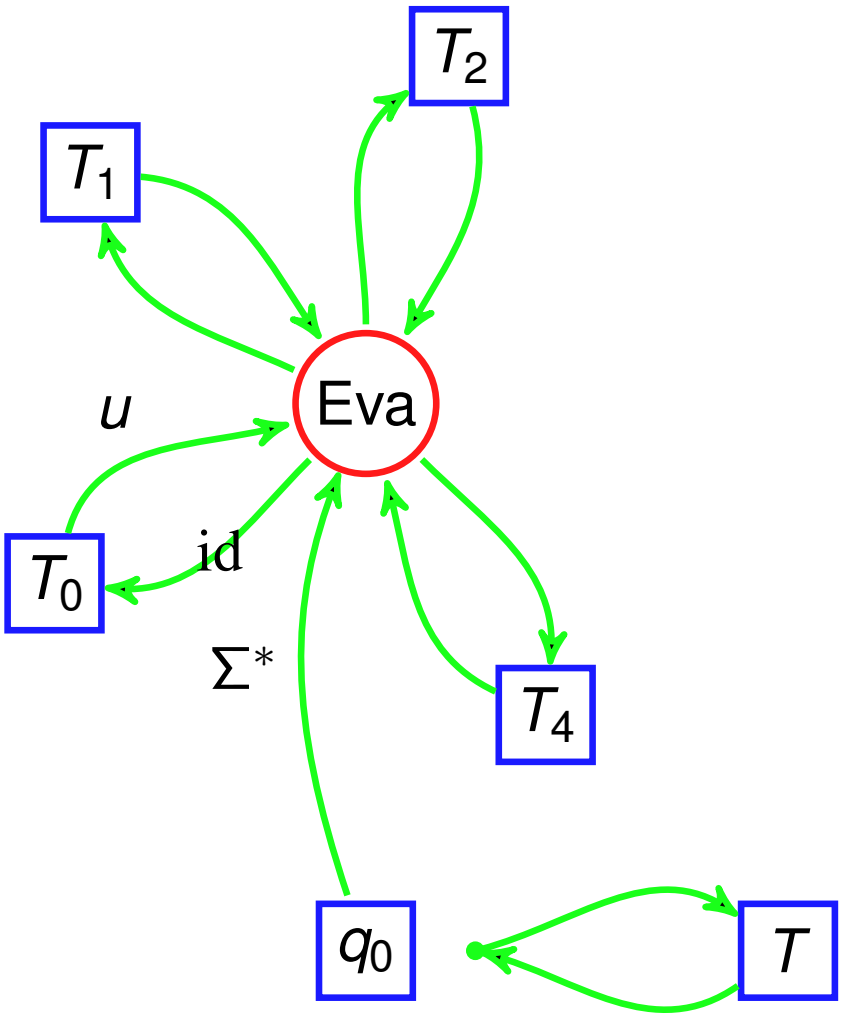
- Then Eva has to never use some petal T in her strategy

Assume Eva would win with $hist(n) - 1$ states



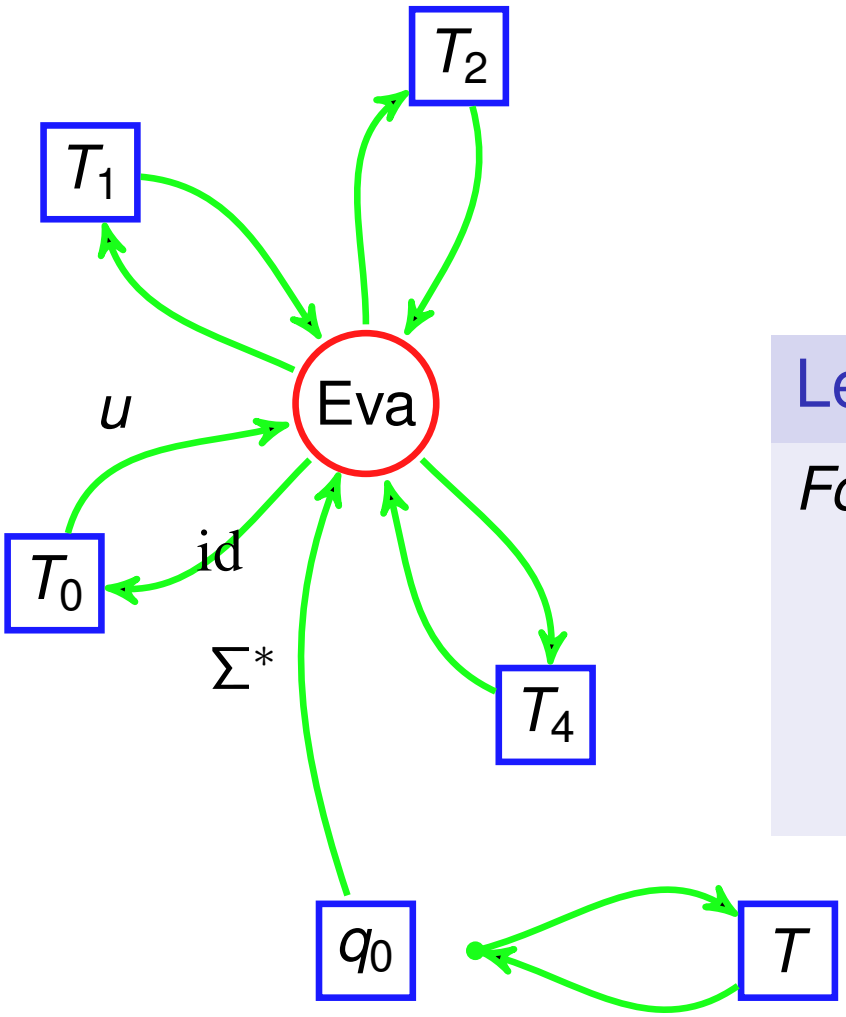
- Then Eva has to never use some petal T in her strategy
- Hence Eva wins the new game $\mathcal{G}_T$ with petal T removed, but:

Assume Eva would win with $hist(n) - 1$ states



- Then Eva has to never use some petal T in her strategy
- Hence Eva wins the new game $\mathcal{G}_T$ with petal T removed, but:

Assume Eva would win with $hist(n) - 1$ states



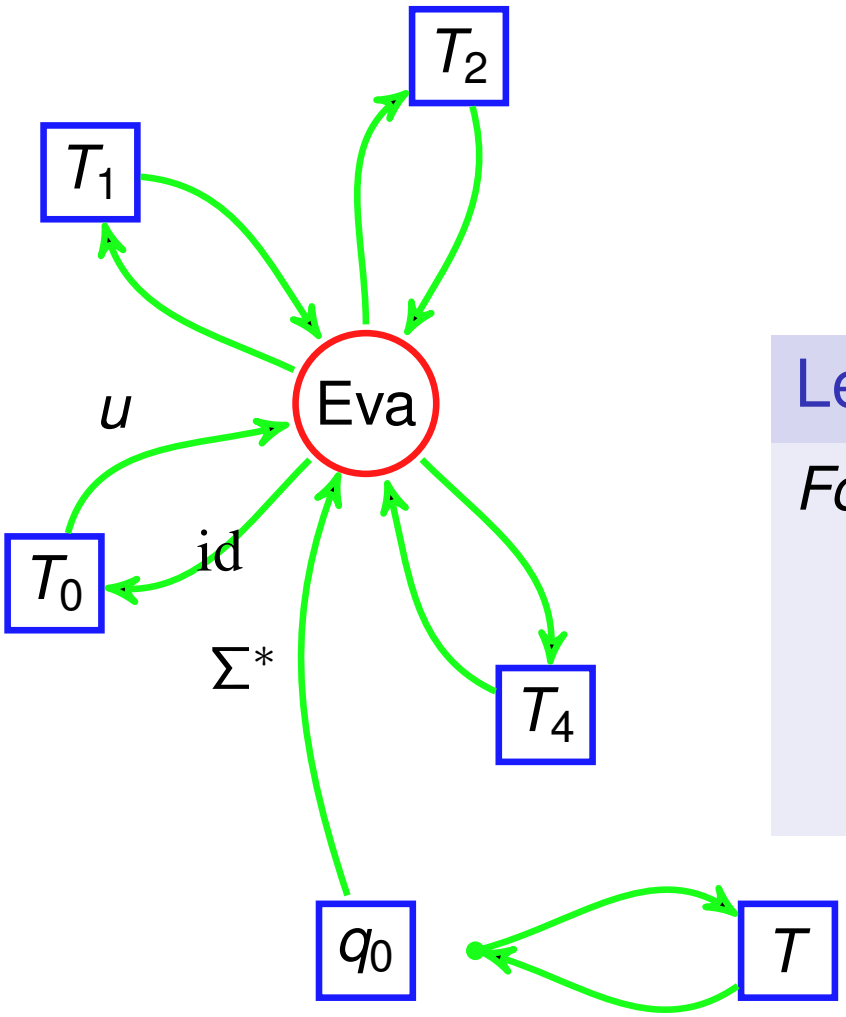
- Then Eva has to never use some petal T in her strategy
- Hence Eva wins the new game $\mathcal{G}_T$ with petal T removed, but:

Lemma (combinatorial core)

For all $T' \neq T$, there is $u \in \Sigma^*$ st:

- *The word u is good from T',*
- *$T \xrightarrow{u} T$,*
- *no position is marked Green in $T \xrightarrow{u} T$.*

Assume Eva would win with $hist(n) - 1$ states



- Then Eva has to never use some petal T in her strategy
- Hence Eva wins the new game $\mathcal{G}_T$ with petal T removed, but:

Lemma (combinatorial core)

For all $T' \neq T$, there is $u \in \Sigma^*$ st:

- *The word u is good from T',*
- *$T \xrightarrow{u} T$,*
- *no position is marked Green in $T \xrightarrow{u} T$.*

- Adam wins the game $\mathcal{G}_T$ by playing the above u when in T', hence Eva can't win $\mathcal{G}_T$ with memory $hist(n) - 1$.

Hence Eva wins the flower game.

But Eva can't win the flower game with memory $hist(n) - 1$.

But recall:

Corollary

Let L be accepted by a deterministic Rabin automaton with k states. If Eva wins an L-game then she wins with memory k.

Hence, all deterministic Rabin automaton for the language $L(\mathcal{F}_n)$ has at least $hist(n)$ states. □

We proved a lower bound on determinization using games. Our lower bound is optimal.

The proof boils down to comparing history-trees, i.e., states of the optimal construction.

Remark: as opposed to the finite case, there can be more than one minimal deterministic Rabin automaton for the same language!

We proved a lower bound on determinization using games. Our lower bound is optimal.

The proof boils down to comparing history-trees, i.e., states of the optimal construction.

Remark: as opposed to the finite case, there can be more than one minimal deterministic Rabin automaton for the same language!

Why transition labeled?

Why counting the number of states?

Why Rabin condition for the output automaton? What about the number of Rabin pairs? What about Müller conditions?

Why only a Büchi condition as input?

Thanks for your attention!