How much memory is needed to win regular games?

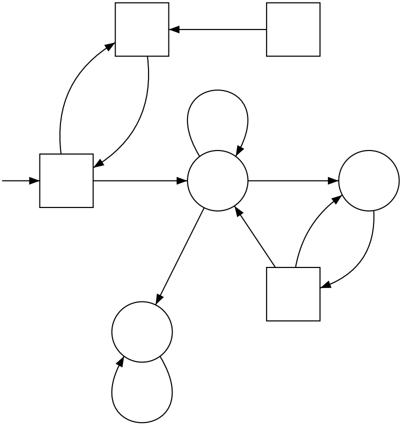
GAMES 2011

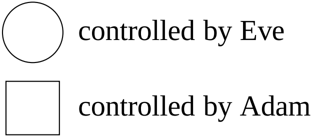
Thomas Colcombet, Nathanaël Fijalkow and Florian Horn

LIAFA, Paris

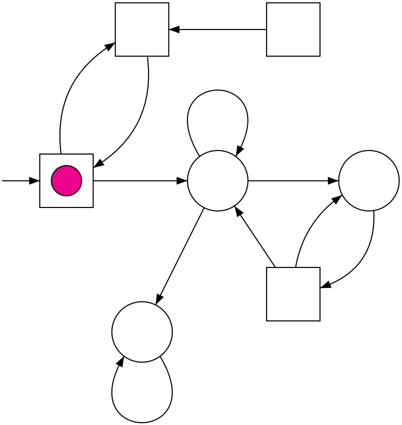
September 1st, 2011

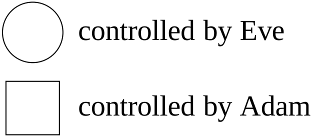
Games



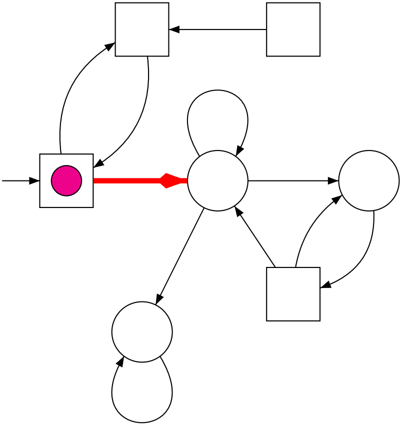


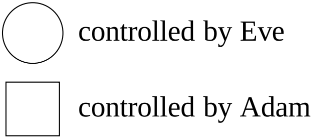
Games



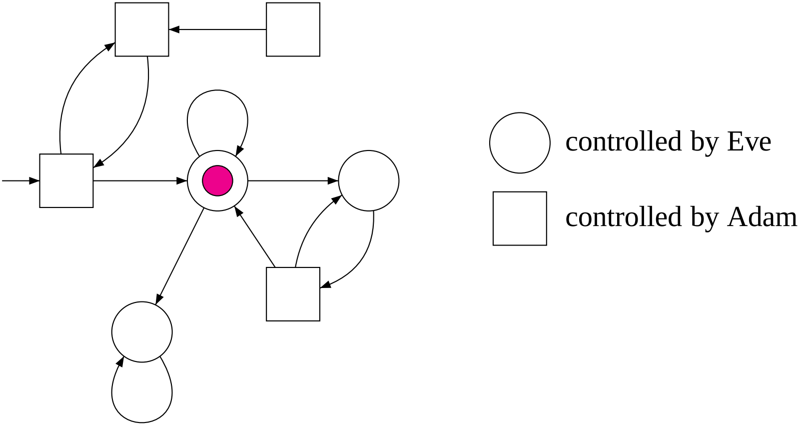


Games

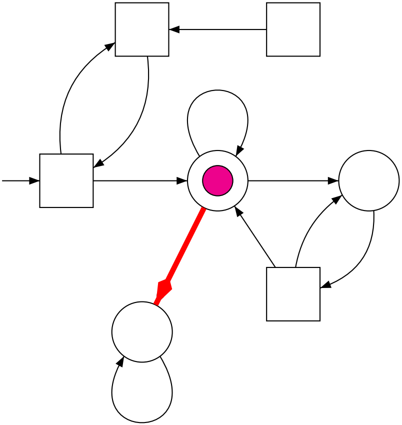


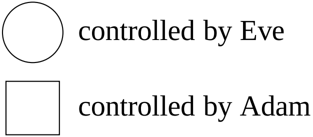


Games

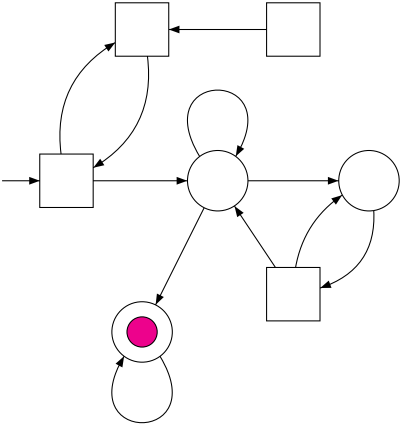


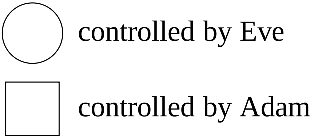
Games



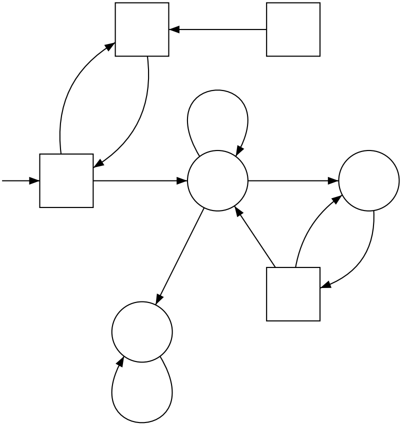


Games



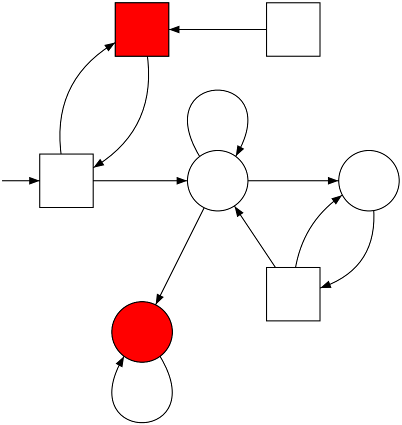


Introduction: reachability games



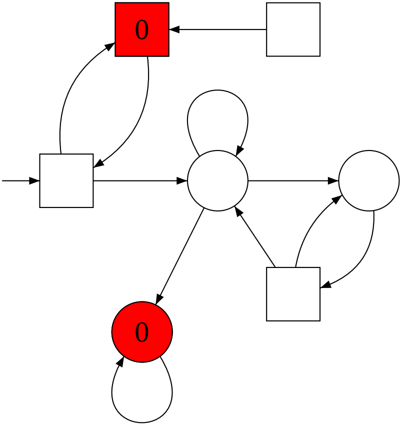
Given $F \subseteq Q$

Introduction: reachability games



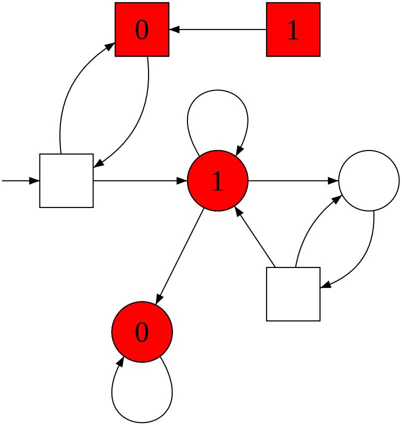
Given $\color{red}{F} \subseteq Q$

Introduction: reachability games



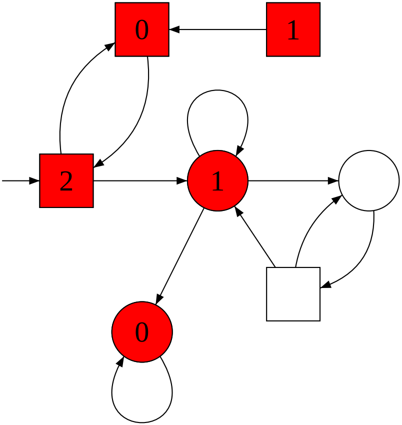
Given $F \subseteq Q$

Introduction: reachability games



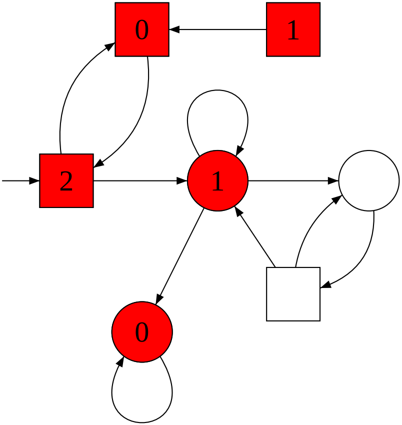
Given $F \subseteq Q$

Introduction: reachability games



Given $\color{red}{F} \subseteq Q$

Introduction: reachability games

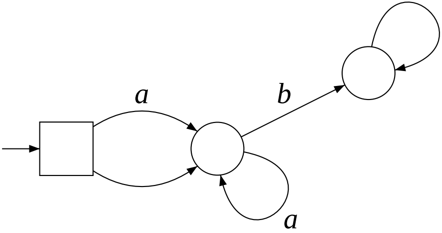


Given $F \subseteq Q$

Memoryless winning strategies
$\sigma : V \to E$ and $\tau : V \to E$

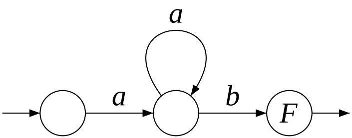
Introduction: regular games

arena



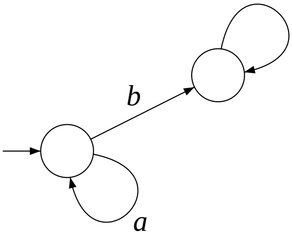
winning condition

$$L = a^+ \cdot b$$



Regular games need memory

arena

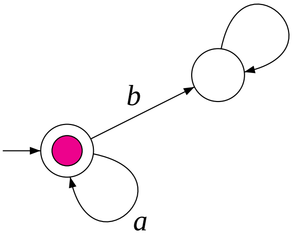


winning condition

$$L = a^+ \cdot b$$

Regular games need memory

arena

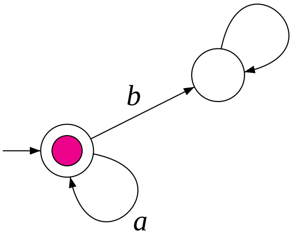


winning condition

$$L = a^{+} \cdot b$$

Regular games need memory

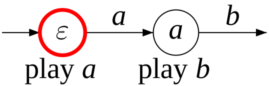
arena



winning condition

$$L = a^{+} \cdot b$$

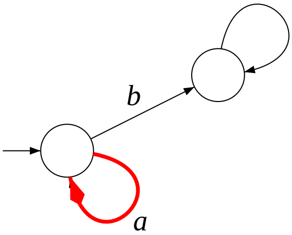
A winning strategy for Eve uses two memory states.



a memory structure

Regular games need memory

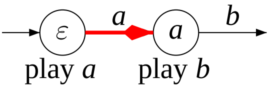
arena



winning condition

$$L = a^+ \cdot b$$

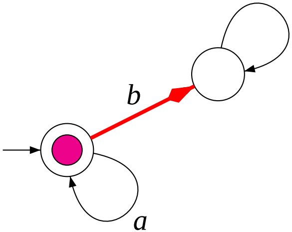
A winning strategy for Eve uses two memory states.



a memory structure

Regular games need memory

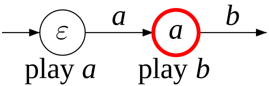
arena



winning condition

$$L = a^{+} \cdot b$$

A winning strategy for Eve uses two memory states.



a memory structure

Introduction: how much memory is needed to win?

Question: given a regular language L, what is the memory required by winning strategies?

Introduction: how much memory is needed to win?

Question: given a regular language L, what is the memory required by winning strategies?
In other words, compute $m_L \in \mathbb{N}^*$ such that:

- in any arena, if Eve wins the regular game for L, then she has a winning strategy with m_L memory states,

Introduction: how much memory is needed to win?

Question: given a regular language L, what is the memory required by winning strategies?
In other words, compute $m_L \in \mathbb{N}^*$ such that:

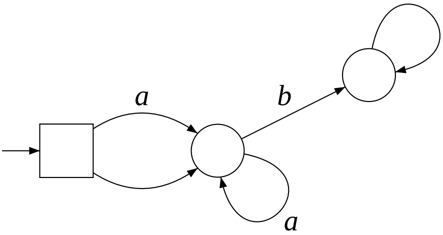
- in any arena, if Eve wins the regular game for L, then she has a winning strategy with m_L memory states,
- there is an arena where Eve wins but there are no winning strategies with less than m_L memory states.

Outline

1. Examples
2. Playing safety
3. Playing reachability
4. Playing optimally in the stochastic case

A first remark

arena

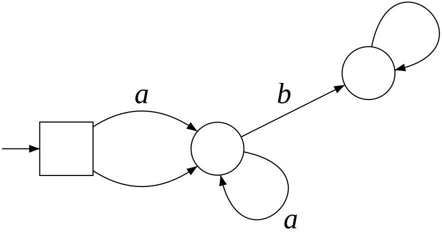


winning condition

$$L = a^+ \cdot b$$

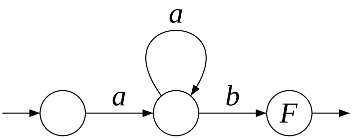
A first remark

arena



winning condition

$$L = a^+ \cdot b$$



Any deterministic automaton that recognizes L is a good memory structure.
Proof: the synchronized product is a reachability game.

An upper bound for both players

We describe a good memory structure for L using left quotients: for $u \in \Sigma^*$,

$$u^{-1}L = \{v \mid u \cdot v \in L\}.$$

- the initial memory state is $\varepsilon^{-1}L = L$,
- each time a letter a is read from $u^{-1}L$, the memory is updated to $(u \cdot a)^{-1}L$.

An upper bound for both players

We describe a good memory structure for L using left quotients: for $u \in \Sigma^*$,

$$u^{-1}L = \{v \mid u \cdot v \in L\}.$$

- the initial memory state is $\varepsilon^{-1}L = L$,
- each time a letter a is read from $u^{-1}L$, the memory is updated to $(u \cdot a)^{-1}L$.

Lemma (An upper bound for both players)

For all regular games $\mathcal{G} = (\mathcal{A}, \mathrm{Reach}(L))$, both players have winning strategies using this memory structure (denoted $\mathcal{M}_L$).

Another example

“read at most ten consecutive b’s, and then an a”.

$$L = a + b \cdot a + bb \cdot a + \ldots + b^{10} \cdot a.$$

Another example

“read at most ten consecutive b’s, and then an a”.

$$L = a + b \cdot a + bb \cdot a + \ldots + b^{10} \cdot a.$$

In every regular game for L, Eve wins without memory.

Another example

"read at most ten consecutive b's, and then an a".

$$L = a + b \cdot a + bb \cdot a + \ldots + b^{10} \cdot a.$$

In every regular game for L, Eve wins without memory.
This shows that the memory structure $\mathcal{M}_L$ is not optimal.

Outline

Outline

Order left quotients inclusion-wise

If Adam wins in $\mathcal{G} \times \mathcal{M}_L$ from $(q, u^{-1}L)$ and $v^{-1}L \subseteq u^{-1}L$, then he wins from $(q, v^{-1}L)$

Order left quotients inclusion-wise

If Adam wins in $\mathcal{G} \times \mathcal{M}_L$ from $(q, u^{-1}L)$ and $v^{-1}L \subseteq u^{-1}L$, then he wins from $(q, v^{-1}L)$ *using the same strategy*.

Order left quotients inclusion-wise

If Adam wins in $\mathcal{G} \times \mathcal{M}_L$ from $(q, u^{-1}L)$ and $v^{-1}L \subseteq u^{-1}L$, then he wins from $(q, v^{-1}L)$ *using the same strategy*.
Let k the maximal number of incomparable (with respect to inclusion) left quotients of L.

Order left quotients inclusion-wise

If Adam wins in $\mathcal{G} \times \mathcal{M}_L$ from $(q, u^{-1}L)$ and $v^{-1}L \subseteq u^{-1}L$, then he wins from $(q, v^{-1}L)$ *using the same strategy*.
Let k the maximal number of incomparable (with respect to inclusion) left quotients of L.

Lemma (A tighter upper bound for Adam)

For all regular games $\mathcal{G} = (\mathcal{A}, \mathrm{Reach}(L))$, Adam has a winning strategy from his winning set that uses k memory states.

Order left quotients inclusion-wise

If Adam wins in $\mathcal{G} \times \mathcal{M}_L$ from $(q, u^{-1}L)$ and $v^{-1}L \subseteq u^{-1}L$, then he wins from $(q, v^{-1}L)$ *using the same strategy*.
Let k the maximal number of incomparable (with respect to inclusion) left quotients of L.

Lemma (A tighter upper bound for Adam)

For all regular games $\mathcal{G} = (\mathcal{A}, \mathrm{Reach}(L))$, Adam has a winning strategy from his winning set that uses k memory states.

Idea: whenever in $(q, v^{-1}L)$, play as from $(q, u^{-1}L)$, where $u^{-1}L$ is maximal winning from q.

Optimality

Lemma (Matching lower bound for Adam)

For all regular languages L, there exists an arena $\mathcal{A}$ such that Adam needs k memory states to win in $\mathcal{G} = (\mathcal{A}, \mathrm{Reach}(L))$.

Optimality

Lemma (Matching lower bound for Adam)

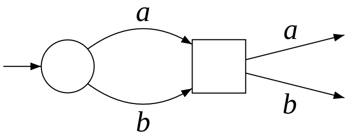
For all regular languages L, there exists an arena $\mathcal{A}$ such that Adam needs k memory states to win in $\mathcal{G} = (\mathcal{A}, \mathrm{Reach}(L))$.

Exemplified: $L = (a+b)^* \cdot (aa + bb)$.

inclusion

$$\begin{tikzcd}
& {\color{red} a^{-1}L} \arrow[dr] & \\
\varepsilon^{-1}L \arrow[ur] \arrow[dr] & & (aa)^{-1}L \\
& {\color{red} b^{-1}L} \arrow[ur] &
\end{tikzcd}$$

arena



Outline

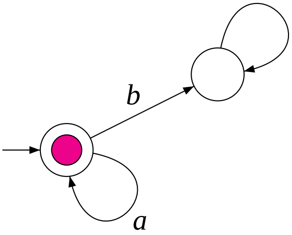
A wrong intuition

If Eve wins in $\mathcal{G} \times \mathcal{M}_L$ from $(q, u^{-1}L)$ and $u^{-1}L \subseteq v^{-1}L$, then she wins from $(q, v^{-1}L)$

A wrong intuition

If Eve wins in $\mathcal{G} \times \mathcal{M}_L$ from $(q, u^{-1}L)$ and $u^{-1}L \subseteq v^{-1}L$, then she wins from $(q, v^{-1}L)$ ... however the *same strategy* might fail!

arena



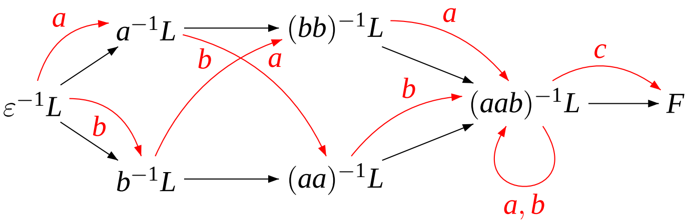
winning condition

$$L = a^{+} \cdot b$$

$$\varepsilon^{-1}L \subset a^{-1}L$$

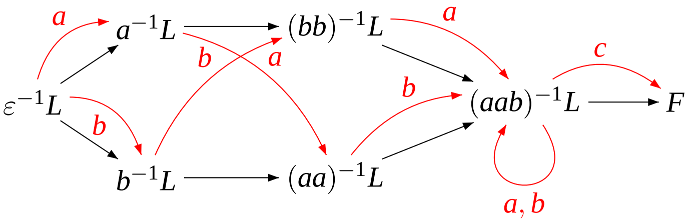
A global merging policy

$$L = (aab + baa) \cdot (a + b)^* \cdot c$$



A global merging policy

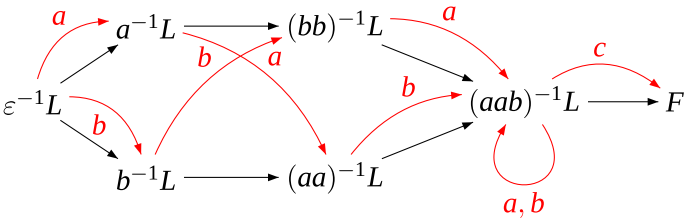
$$L = (aab + baa) \cdot (a + b)^* \cdot c$$



Eve can play from $(bb)^{-1}L$ as from $a^{-1}L$, hence merge the two memory states.

A global merging policy

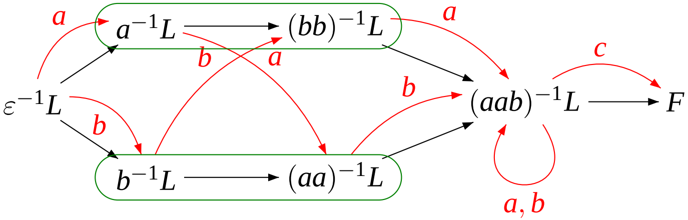
$$L = (aab + baa) \cdot (a + b)^* \cdot c$$



Eve can play from $(bb)^{-1}L$ as from $a^{-1}L$, hence merge the two memory states. Similarly, she merges $(aa)^{-1}L$ and $b^{-1}L \ldots$

A global merging policy

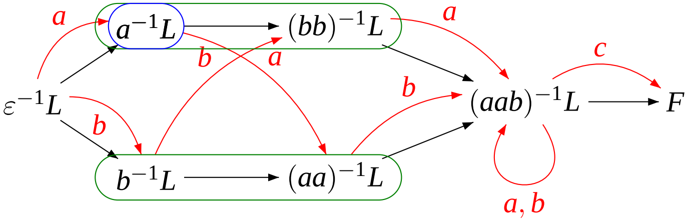
$$L = (aab + baa) \cdot (a + b)^* \cdot c$$



Eve can play from $(bb)^{-1}L$ as from $a^{-1}L$, hence merge the two memory states. Similarly, she merges $(aa)^{-1}L$ and $b^{-1}L \ldots$

A global merging policy

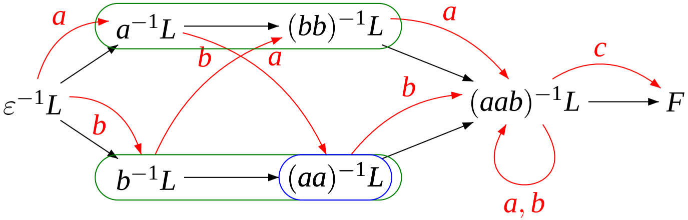
$$L = (aab + baa) \cdot (a + b)^* \cdot c$$



Eve can play from $(bb)^{-1}L$ as from $a^{-1}L$, hence merge the two memory states. Similarly, she merges $(aa)^{-1}L$ and $b^{-1}L \ldots$

A global merging policy

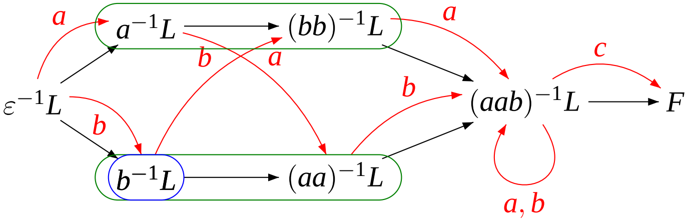
$$L = (aab + baa) \cdot (a + b)^* \cdot c$$



Eve can play from $(bb)^{-1}L$ as from $a^{-1}L$, hence merge the two memory states. Similarly, she merges $(aa)^{-1}L$ and $b^{-1}L$ ...

A global merging policy

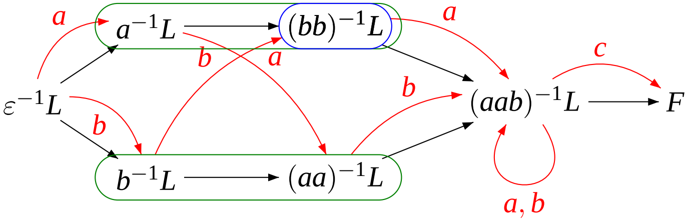
$$L = (aab + baa) \cdot (a + b)^* \cdot c$$



Eve can play from $(bb)^{-1}L$ as from $a^{-1}L$, hence merge the two memory states. Similarly, she merges $(aa)^{-1}L$ and $b^{-1}L$ ...

A global merging policy

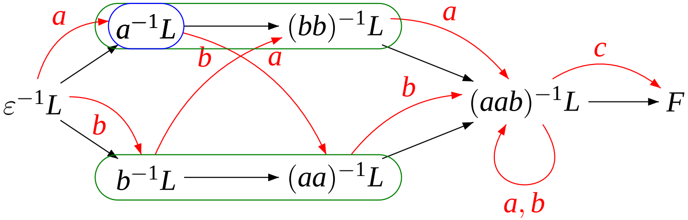
$$L = (aab + baa) \cdot (a + b)^* \cdot c$$



Eve can play from $(bb)^{-1}L$ as from $a^{-1}L$, hence merge the two memory states. Similarly, she merges $(aa)^{-1}L$ and $b^{-1}L \ldots$

A global merging policy

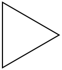
$$L = (aab + baa) \cdot (a + b)^* \cdot c$$



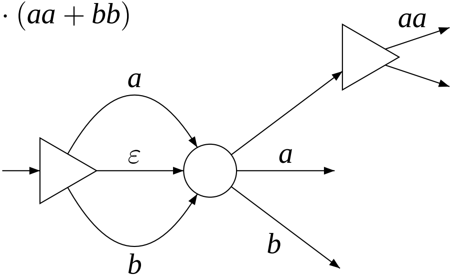
Eve can play from $(bb)^{-1}L$ as from $a^{-1}L$, hence merge the two memory states. Similarly, she merges $(aa)^{-1}L$ and $b^{-1}L$ ...

Outline

Playing optimally in stochastic arenas



$L = (a + b)^* \cdot (aa + bb)$



Upper bound for both players

Since stochastic reachability games enjoy memoryless determinacy:

Lemma (Upper bound for both players)

For all stochastic regular games $\mathcal{G} = (\mathcal{A}, \mathrm{Reach}(L))$, both players have winning strategies using $\mathcal{M}_L$ as memory structure.

Lower bound for Eve

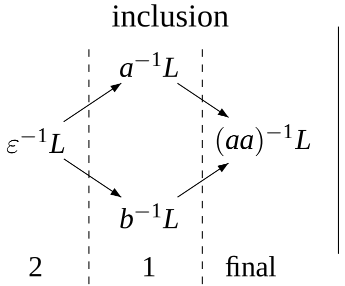
Lemma (Memory lower bound in stochastic games for Eve)

For all regular languages L, let n be the number of non-final left quotients of L, there exists an arena $\mathcal{A}$ such that Eve needs n memory states to play optimally in $\mathcal{G} = (\mathcal{A}, \mathrm{Reach}(L))$.

Again, we order left quotients inclusion-wise.

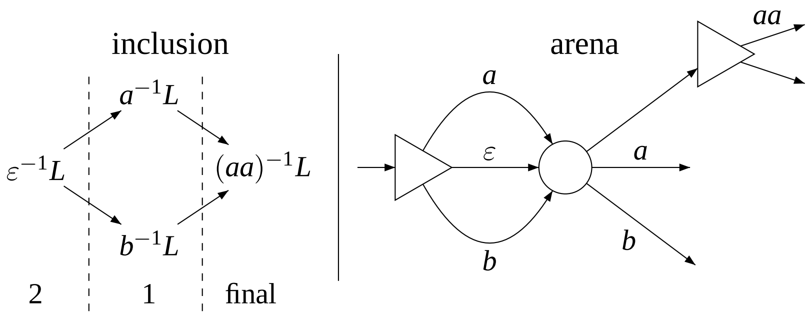
The construction exemplified

We construct an arena for the condition $L = (a+b)^* \cdot (aa+bb)$.



The construction exemplified

We construct an arena for the condition $L = (a+b)^* \cdot (aa+bb)$.



Adam case

The same applies to Adam, using a very similar construction.

Adam case

The same applies to Adam, using a very similar construction.
As opposed to the deterministic case, memory requirements are **symmetric** in stochastic regular games!

The end

Thank you for your attention!