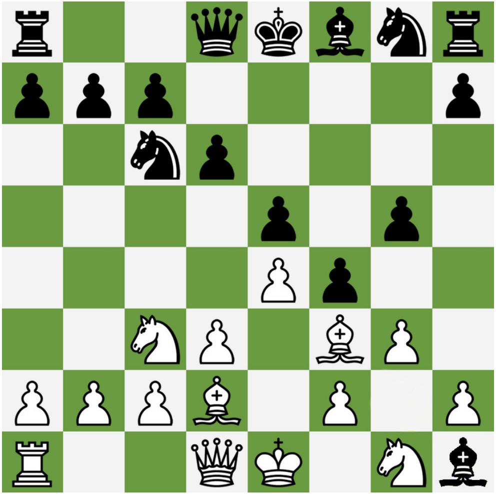
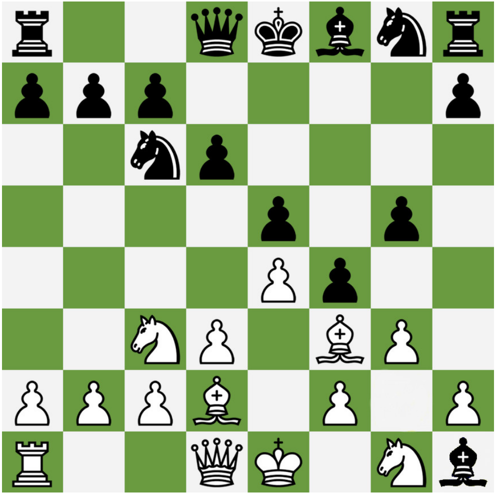
A characterization of half-positional ω-regular languages

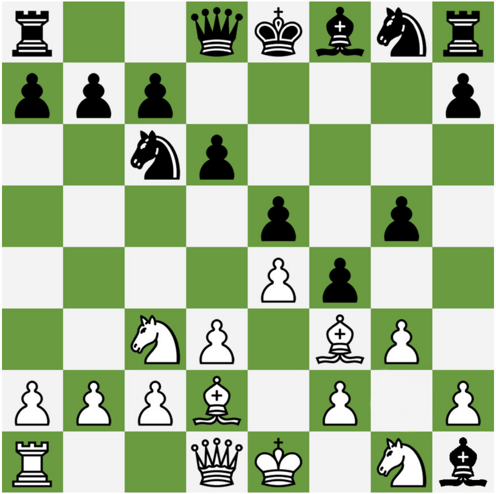
Antonio Casares, Pierre Ohlmann

Automata Seminar IRIF
13 October 2023



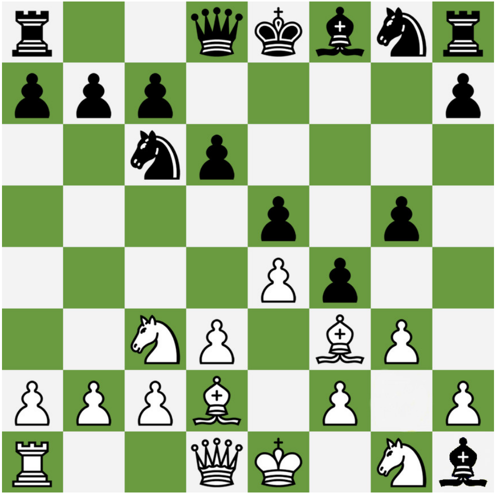


White to move.
What is the optimal move?



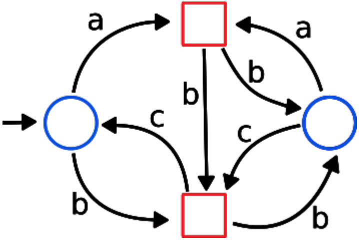
Does not depend on the past of the play**

White to move.
What is the optimal move?



White to move.
What is the optimal move?

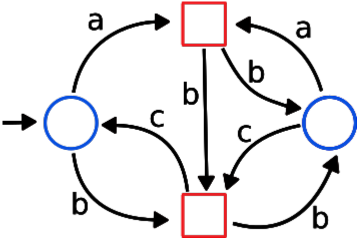
Games on Graphs



$\Sigma = \{a, b, c\}$

- Players move a token in turns producing an infinite word $w \in \Sigma^\omega$.

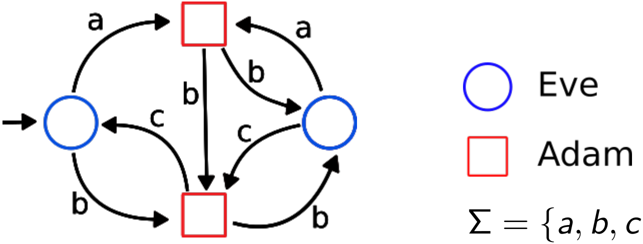
Games on Graphs



$\Sigma = \{a, b, c\}$

- Players move a token in turns producing an infinite word $w \in \Sigma^\omega$.
- The winning condition is given by a language $L \subseteq \Sigma^\omega$.

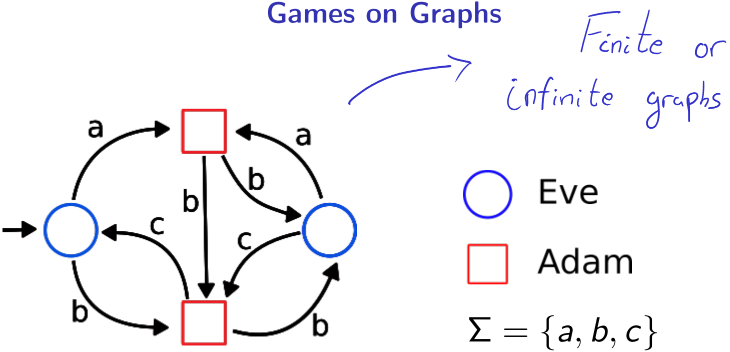
Games on Graphs



$\Sigma = \{a, b, c\}$

- Players move a token in turns producing an infinite word $w \in \Sigma^\omega$.
- The winning condition is given by a language $L \subseteq \Sigma^\omega$.
- Eve wins if the sequence $w \in \Sigma^\omega$ produced belongs to L.

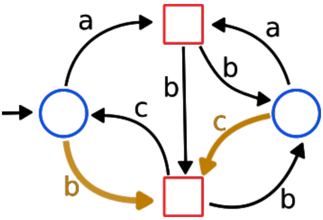
Games on Graphs



$\Sigma = \{a, b, c\}$

- Players move a token in turns producing an infinite word $w \in \Sigma^\omega$.
- The winning condition is given by a language $L \subseteq \Sigma^\omega$.
- Eve wins if the sequence $w \in \Sigma^\omega$ produced belongs to L.

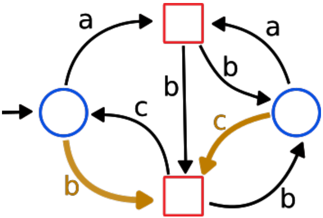
Positional Strategies



A positional strategy (for Eve) is a mapping:

$$\sigma\colon V_{\text{Eve}} \to E.$$

Positional Strategies



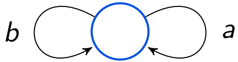
A positional strategy (for Eve) is a mapping:

$$\sigma\colon V_{\text{Eve}} \to E.$$

The positional strategy above ensures the language

$$L = \text{ Words containing the factor } bc \text{ infinitely often} = \mathtt{Inf}(bc).$$

Example: Positional strategies do not always suffice



$$L = \texttt{Inf}(a) \wedge \texttt{Inf}(b)$$

Eve wins, but not positionally.

Positional Languages

Half-positionality

A language L is **half-positional** if, for every game $\mathcal{G}$ using L as winning condition:

Eve can win $\mathcal{G}$ $\implies$ She can win using a positional strategy.

Positional Languages

over finite/arbitrary games

Half-positionality

A language L is **half-positional** if, for every **(finite)** game $\mathcal{G}$ using L as winning condition:

Eve can win $\mathcal{G}$ $\implies$ She can win using a positional strategy.

Positional Languages

Half-positionality

A language L is **half-positional** if, for every **(finite)** game $\mathcal{G}$ using L as winning condition:

Eve can win $\mathcal{G}$ $\implies$ She can win using a positional strategy.

Bipositionality

A language L is **bipositional** if both L and $\Sigma^\omega \setminus L$ are half-positional.

ω-regular languages

The class of ω-**regular languages** can be defined equivalently as those that are:

- Recognizable by a non-deterministic Büchi automaton.
- Recognizable by a deterministic parity automaton.
- Definable by ω-regular expressions.
- Definable in MSO with successor.
- Recognizable by ω-semigroups.

$\vdots$

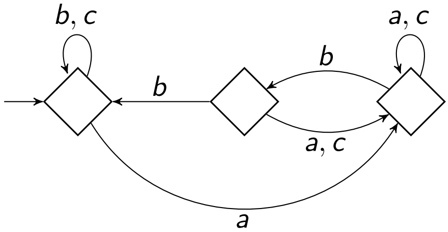
ω-regular languages

The class of ω-**regular languages** can be defined equivalently as those that are:

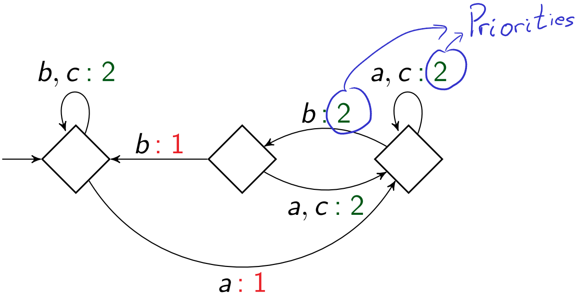
- Recognizable by a non-deterministic Büchi automaton.
- Recognizable by a deterministic parity automaton.
- Definable by ω-regular expressions.
- Definable in MSO with successor.
- Recognizable by ω-semigroups.

$\vdots$

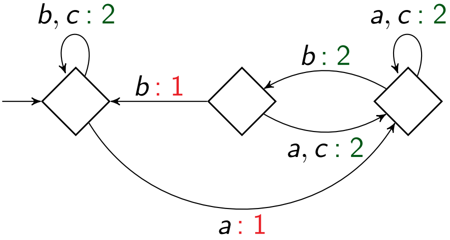
Parity automata (DPA)



Parity automata (DPA)

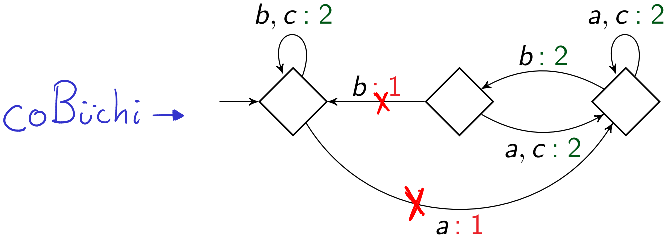


Parity automata (DPA)



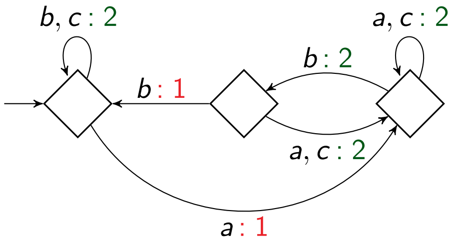
Run accepting if
min{*priorities seen infinitely often*} is even.

Parity automata (DPA)



Run accepting if
$\min\{\textit{priorities seen infinitely often}\}$ is even.

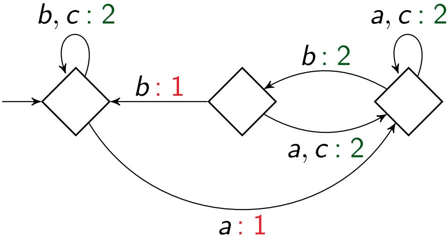
Parity automata (DPA)



Run accepting if
$\min\{\textit{priorities seen infinitely often}\}$ is even.

$\mathcal{L}(\mathcal{A}) = \texttt{Fin}(a) \vee \texttt{Fin}(bb)$

Parity automata (DPA)



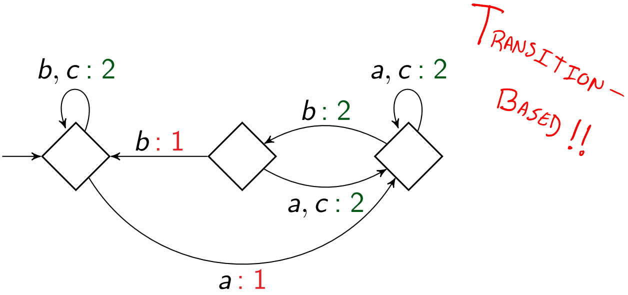
Run accepting if
min{*priorities seen infinitely often*} is even.

$\mathcal{L}(\mathcal{A}) = \texttt{Fin}(a) \vee \texttt{Fin}(bb)$

Prefix-independent:

$w \in L \iff uw \in L$

Parity automata (DPA)



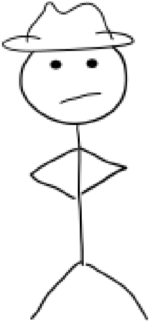
Run accepting if
min{*priorities seen infinitely often*} is even.

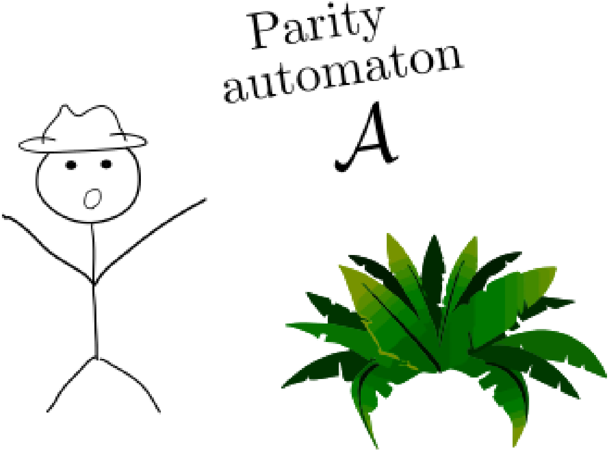
$\mathcal{L}(\mathcal{A}) = \texttt{Fin}(a) \vee \texttt{Fin}(bb)$

Prefix-independent:

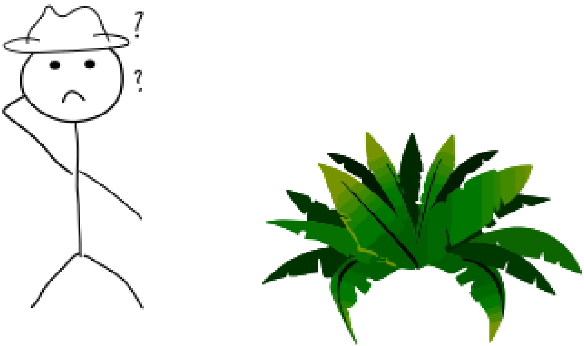
$w \in L \iff uw \in L$

Theoretical Computer Scientist in the Wild




Parity
automaton
$\mathcal{A}$

Is $\mathcal{L}(A)$ half / bi positional?



What do we already know?

Known Results About Bipositionality

Known Results About Bipositionality

Bipositionality over finite graphs

- Characterization of bipositionality over finite graphs (Gimbert, Zielonka 2005).

Known Results About Bipositionality

Bipositionality over finite graphs

- Characterization of bipositionality over finite graphs (Gimbert, Zielonka 2005).
- Corollary: L is bipositional if and only if Eve and Adam can win positionally in 1-player games.

Known Results About Bipositionality

Bipositionality over finite graphs

- Characterization of bipositionality over finite graphs (Gimbert, Zielonka 2005).
- Corollary: L is bipositional if and only if Eve and Adam can win positionally in 1-player games.

→ 1-to-2 player lift

Known Results About Bipositionality

Bipositionality over finite graphs

- Characterization of bipositionality over finite graphs (Gimbert, Zielonka 2005).
- Corollary: L is bipositional if and only if Eve and Adam can win positionally in 1-player games.

Bipositionality over infinite graphs

- Characterization of bipositionality over infinite graphs (Colcombet, Niwiński 2006).

For L prefix-independent:

$$L \text{ is bipositional} \iff L \text{ is the parity objective.}$$

Known Results About Bipositionality

Bipositionality over finite graphs

- Characterization of bipositionality over finite graphs (Gimbert, Zielonka 2005).
- Corollary: L is bipositional if and only if Eve and Adam can win positionally in 1-pl[illegible]s.

Bipositionality over infi[illegible]

- Characterization of bipo[illegible]ty over infinite graphs (Colcombet, Niwiński 2006).

For L prefix-independent:

$$L \text{ is bipositional} \iff L \text{ is the parity objective.}$$

What do we know about half-positionality?

What do we know about half-positionality?

Not much...[*]

[*] This is of course not true. There are many very interesting results about half-positionality.

What do we know about half-positionality?

Decidability for finite graphs (Kopczyński 2007)

The following problem is decidable:

Input:	DPA $\mathcal{A}$ recognising a prefix-independent language.
Question:	$\mathcal{L}(\mathcal{A})$ half-positional over finite graphs?

What do we know about half-positionality?

Decidability for finite graphs (Kopczyński 2007)

The following problem is decidable:

Input: DPA $\mathcal{A}$ recognising a prefix-independent language.

Question: $\mathcal{L}(\mathcal{A})$ half-positional over finite graphs?

What do we know about half-positionality?

Decidability for finite graphs (Kopczyński 2007)

The following problem is decidable in time $\mathcal{O}(n^{n^2})$:

Input: DPA $\mathcal{A}$ recognising a prefix-independent language.

Question: $\mathcal{L}(\mathcal{A})$ half-positional over finite graphs?

What do we know about half-positionality?

Decidability for finite graphs (Kopczyński 2007)

The following problem is decidable in time $\mathcal{O}(n^{n^2})$:

Input: DPA $\mathcal{A}$ recognising a prefix-independent language.

Question: $\mathcal{L}(\mathcal{A})$ half-positional over finite graphs?

Characterization via universal graphs (Ohlmann 2022)

L is half-positional over all graphs

iff

L admits well-ordered monotone universal graphs.

What do we know about half-positionality?

Decidability for finite graphs (Kopczyński 2007)

The following problem is decidable in time $\mathcal{O}(n^{n^2})$:

Input: DPA $\mathcal{A}$ recognising a prefix-independent language.

Question: $\mathcal{L}(\mathcal{A})$ half-positional over finite graphs?

Characterization via universal graphs (Ohlmann 2022)

L is half-positional over all graphs

iff

L admits well-ordered monotone universal graphs.

NOT CONSTRUCTIVE!

What do we know about half-positionality?

Decidability for finite graphs (Kopczyński 2007)

The following problem is decidable in time $\mathcal{O}(n^{n^2})$:

Input: DPA $\mathcal{A}$ recognising a prefix-independent language.

Question: $\mathcal{L}(\mathcal{A})$ half-positional over finite graphs?

Characterization via universal graphs (Ohlmann 2022)

L is half-positional over all graphs

Always $\Uparrow$ iff $\Downarrow$ If L admits a neutral letter

L admits well-ordered monotone universal graphs.

Conjecture: L always admits a neutral letter

Questions about half-positionality (for ω-regular languages)

Decidability for infinite graphs?

Can we decide whether L is half-positional (all graphs)?
Complexity?

Questions about half-positionality (for ω-regular languages)

Decidability for infinite graphs?

Can we decide whether L is half-positional (all graphs)?
Complexity?

Finite vs Infinite

Half-positional for finite graphs $\Longrightarrow$ Half-positional for infinite graphs ?

Questions about half-positionality (for ω-regular languages)

Decidability for infinite graphs?

Can we decide whether L is half-positional (all graphs)?
Complexity?

Finite vs Infinite

Half-positional for finite graphs $\Longrightarrow$ Half-positional for infinite graphs ?

1-to-2-player lift?

Eve can win positionally in 1-player-games $\Longrightarrow$ L is half-positional ?

Questions about half-positionality (for ω-regular languages)

Decidability for infinite graphs?

Can we decide whether L is half-positional (all graphs)?
Complexity?

Finite vs Infinite

Half-positional for finite graphs $\Longrightarrow$ Half-positional for infinite graphs ?

False for non ω-regular languages

1-to-2-player lift?

Eve can win positionally in 1-player-games $\Longrightarrow$ L is half-positional ?

Questions About Half Positionality

Kopczyński's Conjecture (2006)

Let L_1 and L_2 be two prefix-independent half-positional languages. Then:

$$L_1 \cup L_2 \text{ is half-positional.}$$

Questions About Half Positionality

Kopczyński's Conjecture (2006)

Let L_1 and L_2 be two prefix-independent half-positional languages. Then:

$$L_1 \cup L_2 \text{ is half-positional.}$$

- Does not hold for finite graphs (Kozachinskiy, 2022).

  Non ω-regular counterexample!

- Open for infinite graphs.

Contributions

Main Contribution

Characterization of half-positional ω-regular languages

We identify a class $\mathcal{C}$ of deterministic parity automata such that:

- $\mathcal{A} \in \mathcal{C} \implies \mathcal{L}(\mathcal{A})$ is half-positional.
- L half-positional $\implies L$ can be recognized by some $\mathcal{A} \in \mathcal{C}$.

Corollaries of our Characterization

For L an ω-regular language:

Finite vs Infinite

Half-positional for finite graphs $\iff$ Half-positional for infinite graphs

Corollaries of our Characterization

For L an ω-regular language:

Finite vs Infinite

Half-positional for finite graphs $\iff$ Half-positional for infinite graphs

Decidability in polynomial time

Decidability of half-positionality of L in **polynomial time**.

Corollaries of our Characterization

For L an ω-regular language:

Finite vs Infinite

Half-positional for finite graphs $\iff$ Half-positional for infinite graphs

Decidability in polynomial time

Decidability of half-positionality of L in **polynomial time**.

1-to-2-player lift

Eve can win positionally in 1-player-games $\iff$ L is half-positional

Corollaries the Characterization

Kopczyński's Conjecture for ω-regular languages

Let L_1 and L_2 be two prefix-independent, ω-regular, half-positional languages. Then:

$$L_1 \cup L_2 \text{ is half-positional.}$$

Corollaries the Characterization

Kopczyński's Conjecture for ω-regular languages

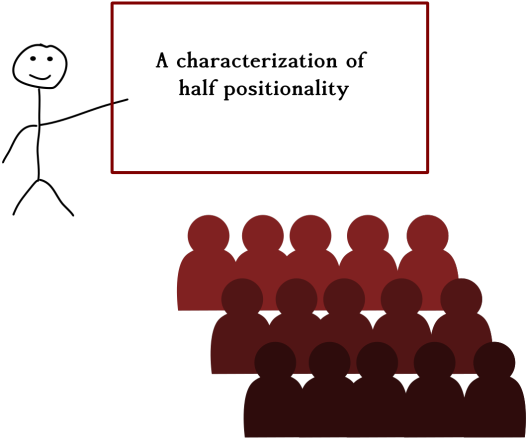
Let L_1 and L_2 be two prefix-independent, ω-regular, half-positional languages. Then:

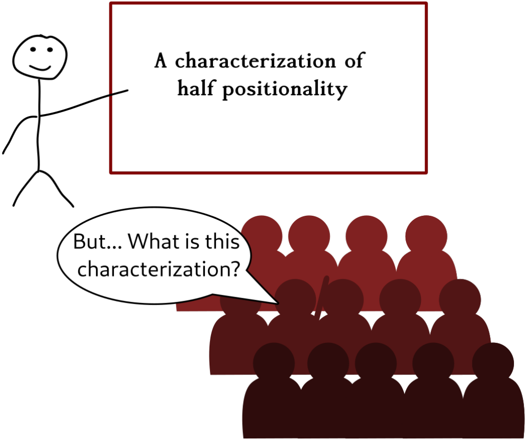
$$L_1 \cup L_2 \text{ is half-positional.}$$

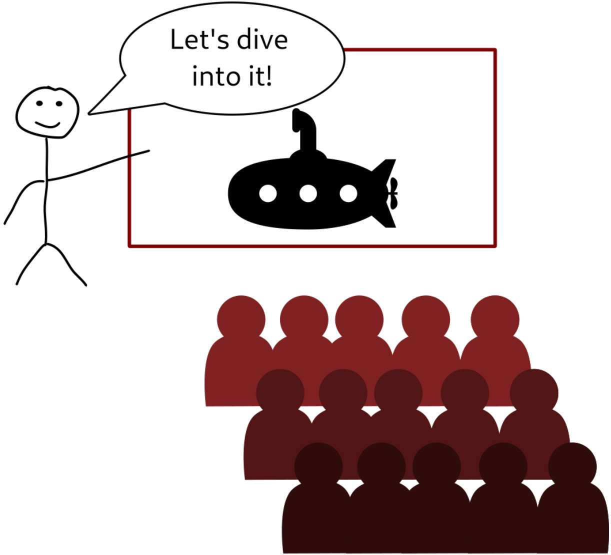
Ohlmann's Neutral Letter Conjecture for ω-regular languages

For L an ω-regular language:

$$L \text{ half-positional} \iff (L + \textit{neutral letter}) \text{ is half-positional.}$$

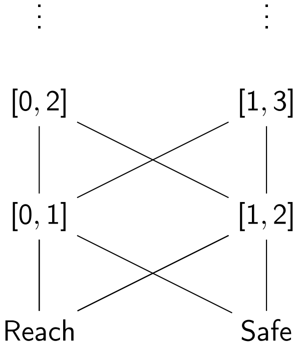

A characterization of half positionality


A characterization of
half positionality
But... What is this
characterization?

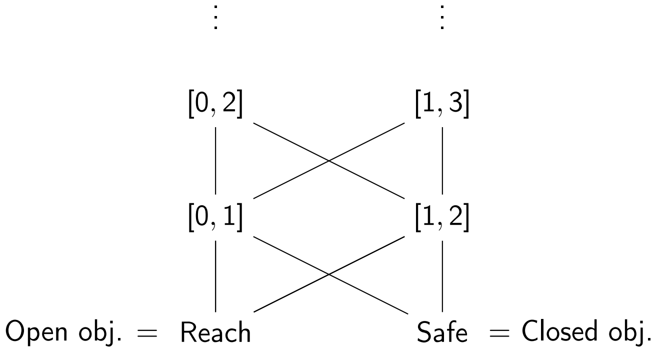

Let's dive into it!

Towards the characterization

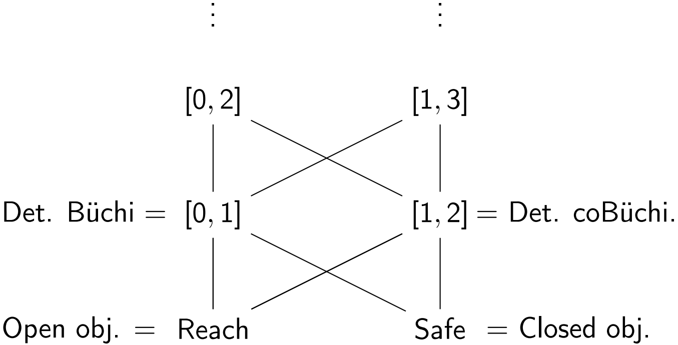
A hierarchy of languages



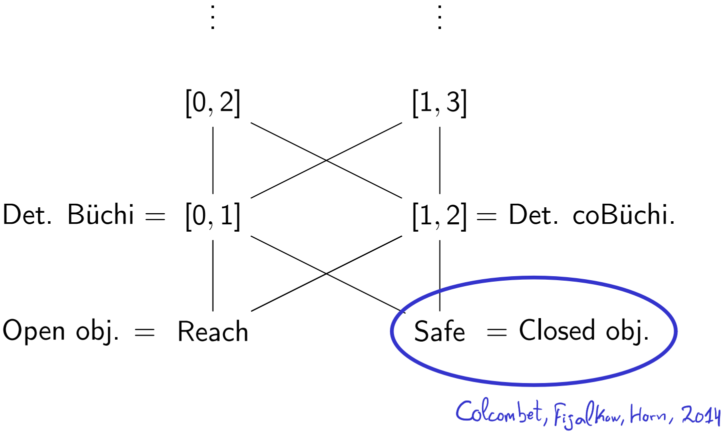
A hierarchy of languages



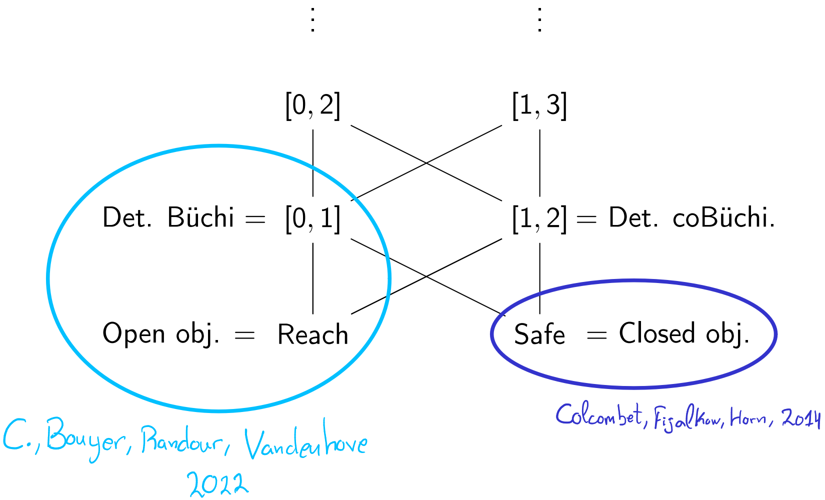
A hierarchy of languages



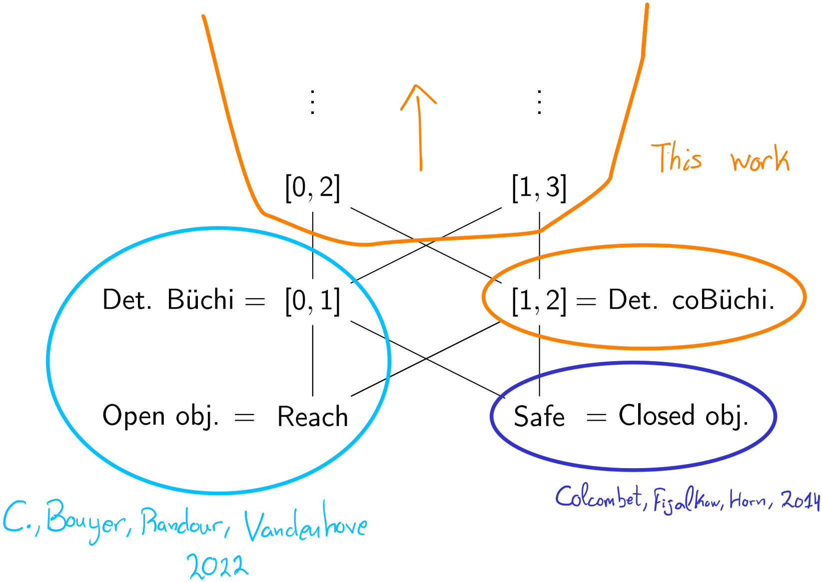
A hierarchy of languages



A hierarchy of languages



A hierarchy of languages



But before climbing the ladder...

But before climbing the ladder...

How to prove a characterization of positionality?

$\mathcal{C}$ a class of languages.

Goal: L half-positional iff $L \in \mathcal{C}$.

$\mathcal{C}$ a class of languages.

Goal: L half-positional iff $L \in \mathcal{C}$.

- L half-positional $\implies L \in \mathcal{C}$

$\mathcal{C}$ a class of languages.

Goal: L half-positional iff $L \in \mathcal{C}$.

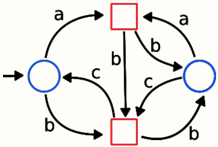
- L half-positional $\implies$ $L \in \mathcal{C}$
- $L \in \mathcal{C}$ $\implies$ L half-positional

$\mathcal{C}$ a class of languages.

Goal: L half-positional iff $L \in \mathcal{C}$.

- L half-positional $\implies$ $L \in \mathcal{C}$
- $L \in \mathcal{C}$ $\implies$ L half-positional

Assume $L \notin \mathcal{C}$



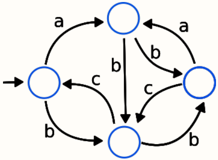
Show game in which Eve can win, but not using a positional strategy.

$\mathcal{C}$ a class of languages.

Goal: L half-positional iff $L \in \mathcal{C}$.

- L half-positional **over finite 1-player graphs** $\implies L \in \mathcal{C}$
- $L \in \mathcal{C} \implies L$ half-positional

Assume $L \notin \mathcal{C}$



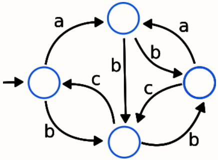
Show **finite 1-player** game in which Eve can win, but not using a positional strategy.

$\mathcal{C}$ a class of languages.

Goal: L half-positional iff $L \in \mathcal{C}$.

- L half-positional **over finite 1-player graphs** $\implies L \in \mathcal{C}$

  Assume $L \notin \mathcal{C}$

  Show **finite 1-player** game in which Eve can win, but not using a positional strategy.

- $L \in \mathcal{C} \implies L$ half-positional

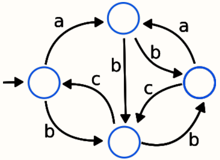
  

$\mathcal{C}$ a class of languages.

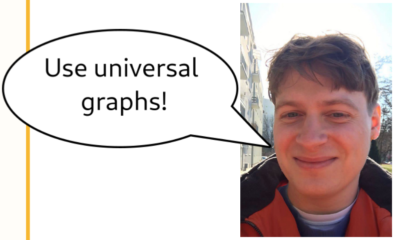
Goal: L half-positional iff $L \in \mathcal{C}$.

- L half-positional **over finite 1-player graphs** $\implies L \in \mathcal{C}$
- $L \in \mathcal{C} \implies L$ half-positional

Assume $L \notin \mathcal{C}$



Show **finite 1-player** game in which Eve can win, but not using a positional strategy.



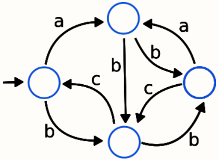
$\mathcal{C}$ a class of languages.

Goal: L half-positional iff $L \in \mathcal{C}$.

- L half-positional **over finite 1-player graphs** $\implies L \in \mathcal{C}$

- $L \in \mathcal{C} \implies L$ half-positional

Assume $L \notin \mathcal{C}$



Show **finite 1-player** game in which Eve can win, but not using a positional strategy.

$\exists$ Well-ordered monotone L-universal graph

$\Downarrow$

L is half-positional

$\mathcal{C}$ a class of languages.

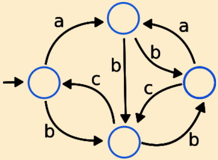
Goal: L half-positional iff $L \in \mathcal{C}$.

- L half-positional **over finite 1-player graphs** $\implies L \in \mathcal{C}$

- $L \in \mathcal{C} \implies L$ half-positional

Assume $L \notin \mathcal{C}$

This talk



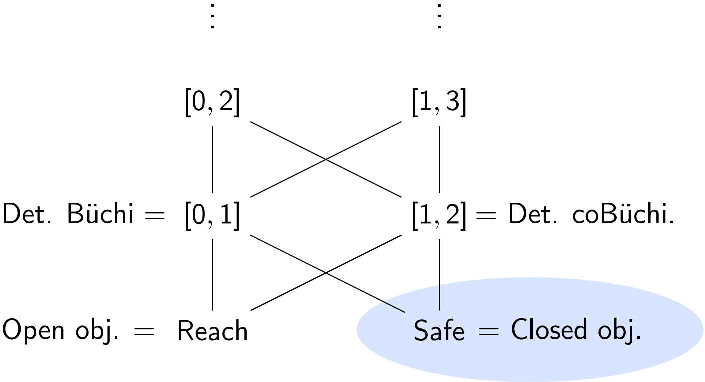
Show **finite 1-player** game in which Eve can win, but not using a positional strategy.

$\exists$ Well-ordered monotone L-universal graph

$\Downarrow$

L is half-positional

Closed objectives



Residuals

We fix an language $L \subseteq \Sigma^\omega$.

For a finite word $u \in \Sigma^*$ we write

$$u^{-1}L = \{w \in \Sigma^\omega \mid uw \in L\}.$$

$$\mathrm{Res}(L) = \text{ Residuals of } L.$$

➔ We consider Res(L) **ordered by inclusion**.

Closed objectives: Total order of residuals

Proposition

If $L \subseteq \Sigma^\omega$ is half-positional, then $\mathrm{Res}(L)$ *is totally ordered by inclusion.*
If L is topologically closed, this condition is also sufficient.

Closed objectives: Total order of residuals

Proposition

If $L \subseteq \Sigma^\omega$ is half-positional, then $\mathrm{Res}(L)$ is totally ordered by inclusion.
If L is topologically closed, this condition is also sufficient.

Proof (necessity):
On the contrary, there are $u_1, u_2 \in \Sigma^*$ and $w_1, w_2 \in \Sigma^\omega$ such that:

$$u_1 w_1 \in L, \quad u_2 w_1 \notin L,$$

$$u_2 w_2 \in L, \quad u_1 w_2 \notin L.$$

Closed objectives: Total order of residuals

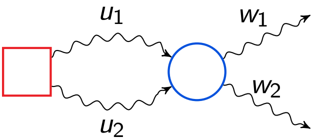
Proposition

If $L \subseteq \Sigma^\omega$ is half-positional, then $\mathrm{Res}(L)$ *is totally ordered by inclusion.*
If L is topologically closed, this condition is also sufficient.

Proof (necessity):
On the contrary, there are $u_1, u_2 \in \Sigma^*$ and $w_1, w_2 \in \Sigma^\omega$ such that:

$$u_1 w_1 \in L, \quad u_2 w_1 \notin L,$$

$$u_2 w_2 \in L, \quad u_1 w_2 \notin L.$$



Closed objectives: Total order of residuals

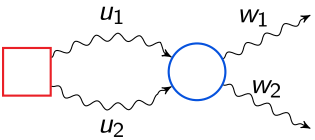
Proposition

If $L \subseteq \Sigma^\omega$ is half-positional, then $\mathrm{Res}(L)$ *is totally ordered by inclusion.*
If L is topologically closed, this condition is also sufficient.

Proof (necessity):
On the contrary, there are $u_1, u_2 \in \Sigma^*$ and $w_1, w_2 \in \Sigma^\omega$ such that:

$$u_1 w_1 \in L, \quad u_2 w_1 \notin L,$$

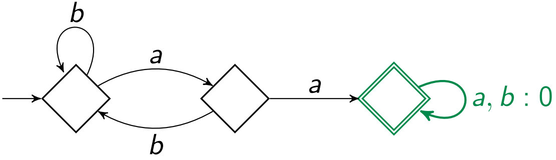
$$u_2 w_2 \in L, \quad u_1 w_2 \notin L.$$



We can make it
- finite
- 1-player

Example: Total order not sufficient in general

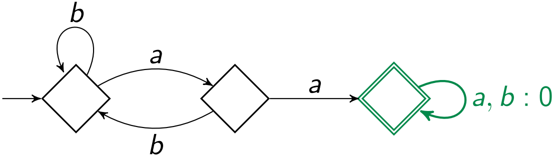
$$L = \{w \in \Sigma^\omega \mid w \text{ contains the factor } aa\}.$$



$$\varepsilon^{-1}L \quad \subseteq \quad a^{-1}L \quad \subseteq \quad aa^{-1}L = \Sigma^\omega$$

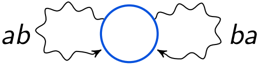
Example: Total order not sufficient in general

$$L = \{w \in \Sigma^\omega \mid w \text{ contains the factor } aa\}.$$

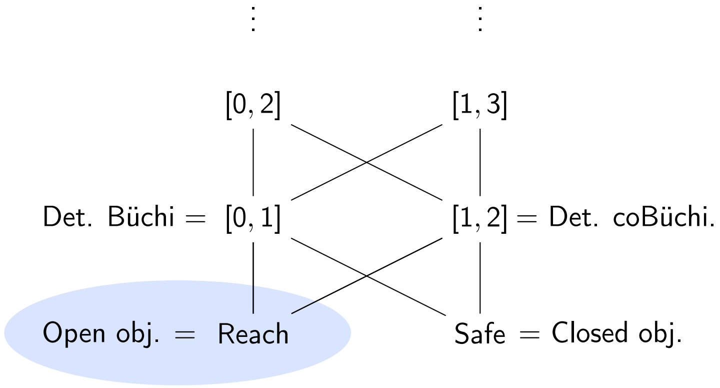


$$\varepsilon^{-1}L \quad \subseteq \quad a^{-1}L \quad \subseteq \quad aa^{-1}L = \Sigma^\omega$$

However:



Open objectives



Progress consistency

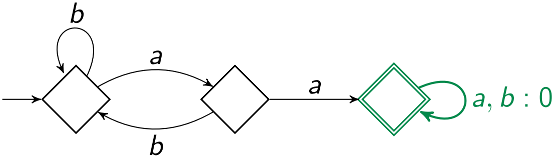
Language L is ***progress consistent*** if for all $u, v \in \Sigma^*$:

$$u^{-1}L \subsetneq uv^{-1}L \implies uv^{\omega} \in L.$$

Progress consistency

Language L is ***progress consistent*** if for all $u, v \in \Sigma^*$:

$$u^{-1}L \subsetneq uv^{-1}L \implies uv^\omega \in L.$$

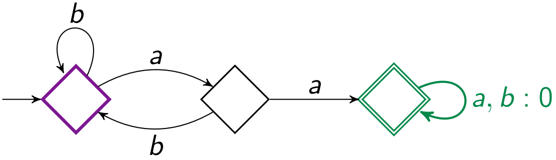


$$\varepsilon^{-1}L \quad \subseteq \quad a^{-1}L \quad \subseteq \quad aa^{-1}L = \Sigma^\omega$$

Progress consistency

Language L is ***progress consistent*** if for all $u, v \in \Sigma^*$:

$$u^{-1}L \subsetneq uv^{-1}L \implies uv^\omega \in L.$$



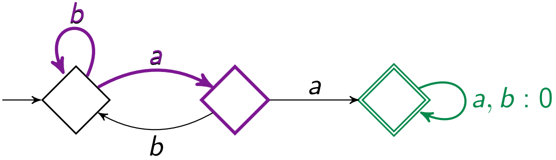
$$\varepsilon^{-1}L \quad \subseteq \quad a^{-1}L \quad \subseteq \quad aa^{-1}L = \Sigma^\omega$$

$u = \varepsilon,$

Progress consistency

Language L is ***progress consistent*** if for all $u, v \in \Sigma^*$:

$$u^{-1}L \subsetneq uv^{-1}L \implies uv^\omega \in L.$$



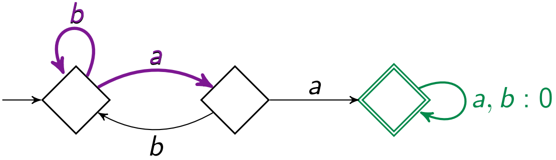
$$\varepsilon^{-1}L \quad \subseteq \quad a^{-1}L \quad \subseteq \quad aa^{-1}L = \Sigma^\omega$$

$u = \varepsilon, \ \ v = ba$

Progress consistency

Language L is ***progress consistent*** if for all $u, v \in \Sigma^*$:

$$u^{-1}L \subsetneq uv^{-1}L \implies uv^\omega \in L.$$



$$\varepsilon^{-1}L \quad \subsetneq \quad a^{-1}L \quad \subsetneq \quad aa^{-1}L = \Sigma^\omega$$

$u = \varepsilon,\ \ v = ba$, but $(ba)^\omega \notin L \implies L$ not progress consistent.

Open objectives: Progress consistency + Total order

Proposition

If L is half-positional, then it is progress consistent.

Open objectives: Progress consistency + Total order

Proposition

If L is half-positional, then it is progress consistent.

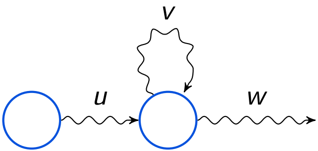
Proof:
Let u, v such that $u^{-1}L \subsetneq uv^{-1}L$ but $uv^\omega \notin L$.
There is $w \in \Sigma^\omega$ s.t. $uw \notin L$ but $uvw \in L$.

Open objectives: Progress consistency + Total order

Proposition

If L is half-positional, then it is progress consistent.

Proof:
Let u, v such that $u^{-1}L \subsetneq uv^{-1}L$ but $uv^\omega \notin L$.
There is $w \in \Sigma^\omega$ s.t. $uw \notin L$ but $uvw \in L$.

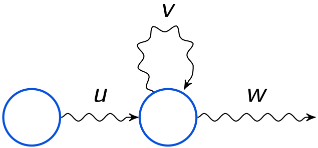


Open objectives: Progress consistency + Total order

Proposition

If L is half-positional, then it is progress consistent.

Proof:
Let u, v such that $u^{-1}L \subsetneq uv^{-1}L$ but $uv^{\omega} \notin L$.
There is $w \in \Sigma^{\omega}$ s.t. $uw \notin L$ but $uvw \in L$.

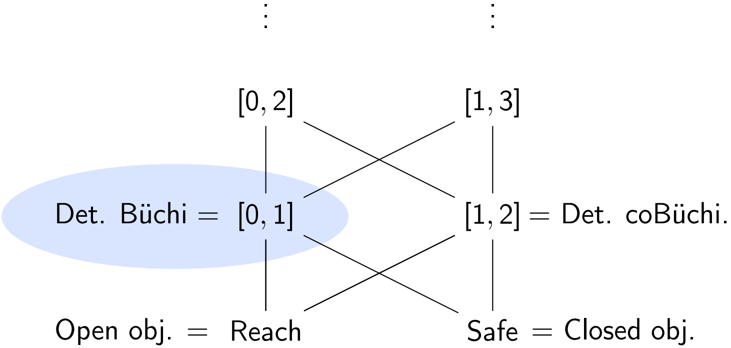


Proposition

If L is topologically open, it is half-positional if and only if:

1. *Total order of residuals, and*
2. *Progress consistent.*

Objectives recognized by deterministic Büchi automata



Deterministic Büchi: Uniformity of 0-transitions

Proposition

Let L be recognizable by a deterministic Büchi automaton. L is half-positional if and only if:

1. *Total order of residuals,*
2. *Progress consistent,*
3. *Can be recognized by the automaton of residuals (one state per residual).*

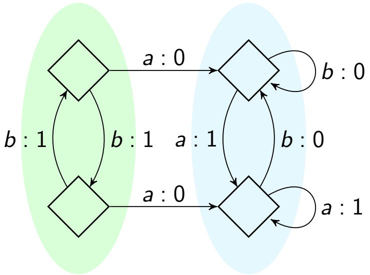
Deterministic Büchi: Uniformity of 0-transitions

Proposition

Let L be recognizable by a deterministic Büchi automaton. L is half-positional if and only if:

1. *Total order of residuals,*
2. *Progress consistent,*
3. *Can be recognized by the automaton of residuals (one state per residual).*

<u>One idea. Uniformity of 0-transitions</u>:



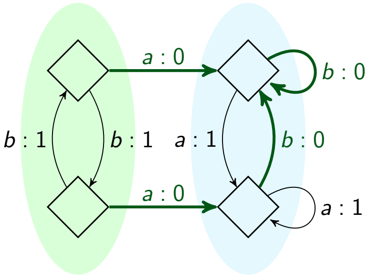
Deterministic Büchi: Uniformity of 0-transitions

Proposition

Let L be recognizable by a deterministic Büchi automaton. L is half-positional if and only if:

1. *Total order of residuals,*
2. *Progress consistent,*
3. *Can be recognized by the automaton of residuals (one state per residual).*

One idea. Uniformity of 0-transitions:



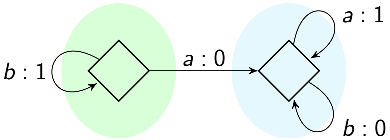
Deterministic Büchi: Uniformity of 0-transitions

Proposition

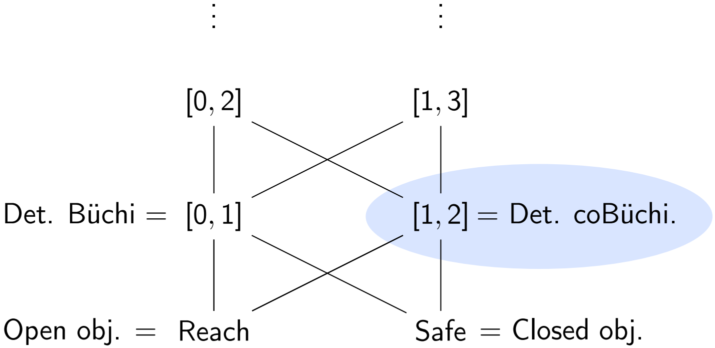
Let L be recognizable by a deterministic Büchi automaton. L is half-positional if and only if:

1. *Total order of residuals,*
2. *Progress consistent,*
3. *Can be recognized by the automaton of residuals (one state per residual).*

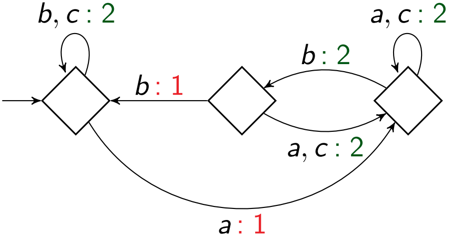
<u>One idea. Uniformity of 0-transitions</u>:



Objectives recognized by deterministic coBüchi automata

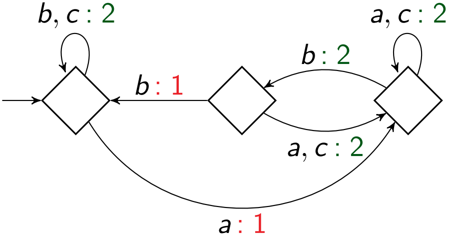


Example: Objective recognized by coBüchi automaton



$$\mathcal{L}(\mathcal{A}) = \texttt{Fin}(a) \vee \texttt{Fin}(bb)$$

Example: Objective recognized by coBüchi automaton



$$\mathcal{L}(\mathcal{A}) = \texttt{Fin}(a) \vee \texttt{Fin}(bb)$$

- Only one residual (prefix-independent)
- Claim: Half-positional

Safe languages and safe components

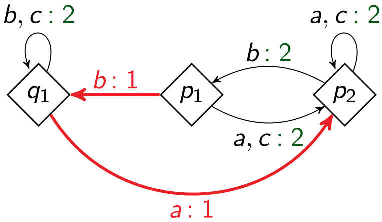
For a state q we write:

$\mathcal{L}_{\mathsf{safe}}(q) = \{w \in \Sigma^* \mid$ the path $q \xrightarrow{w}$ does not produce $1\}$.

Safe languages and safe components

For a state q we write:

$$\mathcal{L}_{\mathsf{safe}}(q) = \{w \in \Sigma^* \mid \text{ the path } q \overset{w}{\rightsquigarrow} \text{ does not produce } 1\}.$$

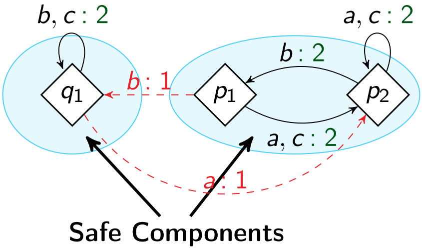


- $\mathcal{L}_{\mathsf{safe}}(q_1) = (b + c)^*$
- $\mathcal{L}_{\mathsf{safe}}(p_1) = (a + c)\overline{(\Sigma^* bb\Sigma^*)}$
- $\mathcal{L}_{\mathsf{safe}}(p_2) = \overline{(\Sigma^* bb\Sigma^*)}$

Safe languages and safe components

For a state q we write:

$$\mathcal{L}_{\mathsf{safe}}(q) = \{w \in \Sigma^* \mid \text{ the path } q \overset{w}{\rightsquigarrow} \text{ does not produce } 1\}.$$



CoBüchi: Total order in safe components

Claim

If $\mathcal{L}(\mathcal{A})$ is half-positional, the states of each safe component are totally ordered by inclusion of safe languages.

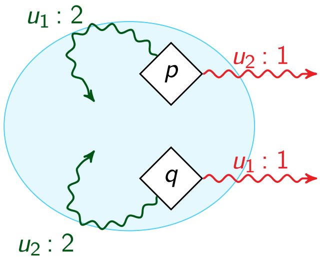
CoBüchi: Total order in safe components

Claim

If $\mathcal{L}(\mathcal{A})$ is half-positional, the states of each safe component are totally ordered by inclusion of safe languages.

<u>(Not a) proof</u>:

On the contrary, there are states p, q and words $u_1, u_2 \in \Sigma^*$ s.t.:



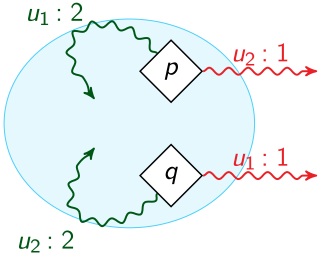
CoBüchi: Total order in safe components

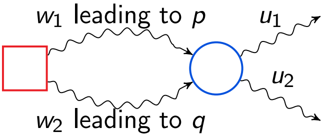
Claim

If $\mathcal{L}(\mathcal{A})$ is half-positional, the states of each safe component are totally ordered by inclusion of safe languages.

(Not a) proof:

On the contrary, there are states p, q and words $u_1, u_2 \in \Sigma^*$ s.t.:





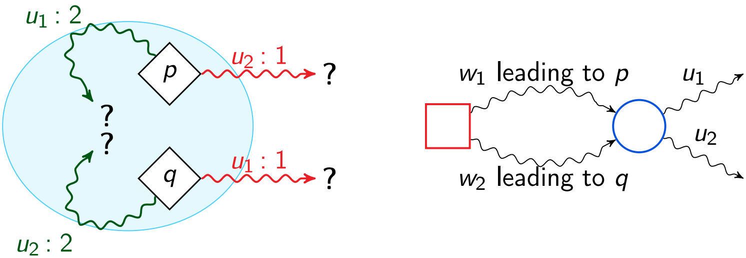
CoBüchi: Total order in safe components

Claim

If $\mathcal{L}(\mathcal{A})$ is half-positional, the states of each safe component are totally ordered by inclusion of safe languages.

(Not a) proof:

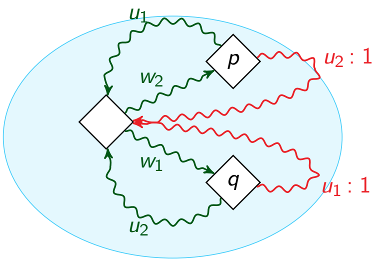
On the contrary, there are states p, q and words $u_1, u_2 \in \Sigma^*$ s.t.:

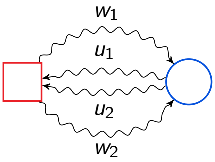


And now what? We need to iterate this process.

Synchronizing words

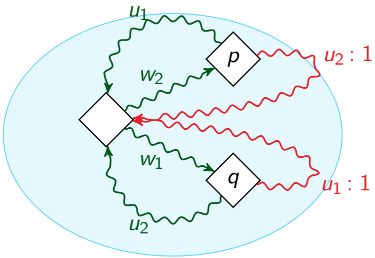
Objective: Obtain words giving the following situation

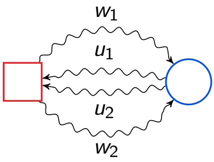




Synchronizing words

Objective: Obtain words giving the following situation

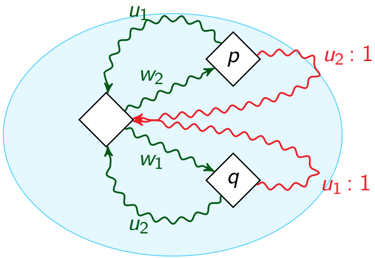


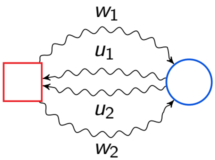


- An arbitrary automaton does not admit synchronizing words.

Synchronizing words

Objective: Obtain words giving the following situation



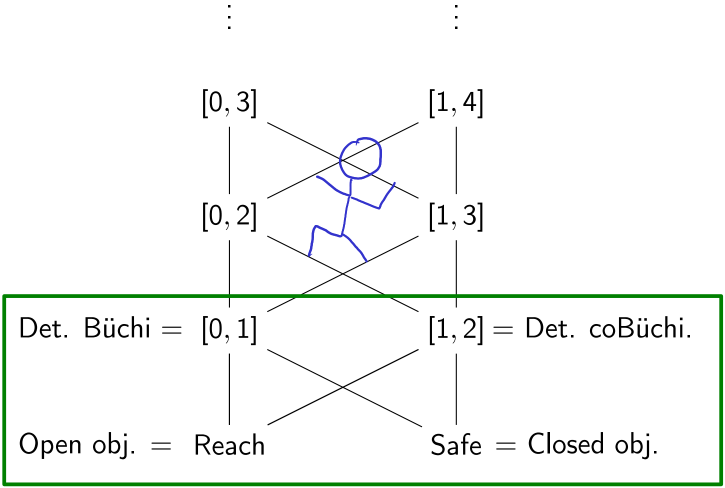


- An arbitrary automaton does not admit synchronizing words.
- But we can obtain a **minimal history-deterministic*** automaton which always admits synchronizing words!†

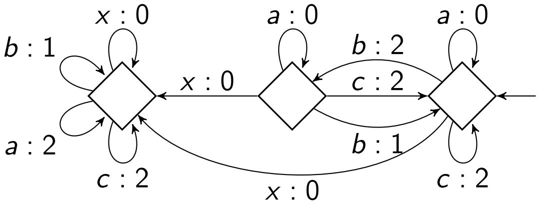
*History-deterministic = Good-for games

†Minimization method from Abu Radi and Kupferman "*Minimizing GFG Transition-Based Automata*" ICALP 2019.

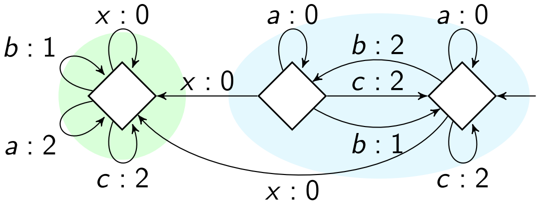
Climbing the ladder



Proof by example



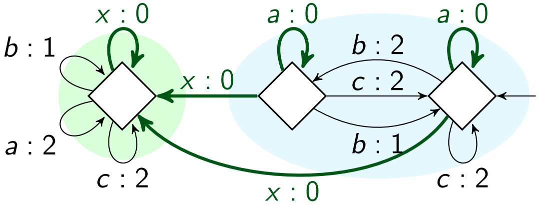
Proof by example



$x^{-1}L \quad \subsetneq \quad \varepsilon^{-1}L$

- Residuals totally ordered ✓
- Progress consistency ✓

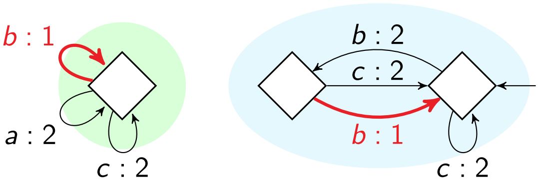
Proof by example



$x^{-1}L \quad \subsetneq \quad \varepsilon^{-1}L$

- Residuals totally ordered ✓
- Progress consistency ✓
- Uniformity of 0-transitions ✓

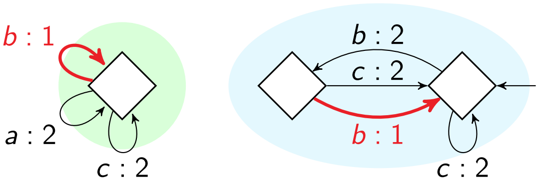
Proof by example



$x^{-1}L \quad \subsetneq \quad \varepsilon^{-1}L$

- Residuals totally ordered ✓
- Progress consistency ✓
- Uniformity of 0-transitions ✓
- Total order of safe languages ✓

Proof by example



$x^{-1}L \quad \subsetneq \quad \varepsilon^{-1}L$

- Residuals totally ordered ✓
- Progress consistency ✓
- Uniformity of 0-transitions ✓
- Total order of safe languages ✓

$\vdots$

Final characterization

Idea:

L half-positional $\iff$ L can be recognized by a parity automaton admitting a decomposition by layers

Final characterization

Idea:

L half-positional $\iff$ L can be recognized by a parity automaton admitting a decomposition by layers

- Formal details quite technical

Deciding half-positionality in polynomial time

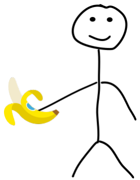
Input: Deterministic parity automaton $\mathcal{A}$.

➔ Apply a sequence of transformations to $\mathcal{A}$

Deciding half-positionality in polynomial time

Input: Deterministic parity automaton $\mathcal{A}$.

➔ Apply a sequence of transformations to $\mathcal{A}$



You can peel it

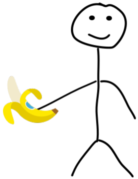
$\Downarrow$

$\mathcal{L}(\mathcal{A})$ half-positional

Deciding half-positionality in polynomial time

Input: Deterministic parity automaton $\mathcal{A}$.

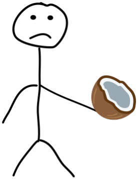
➔ Apply a sequence of transformations to $\mathcal{A}$



You can peel it

$\Downarrow$

$\mathcal{L}(\mathcal{A})$ half-positional



You cannot peel it

$\Downarrow$

It is a coconut

Conclusions

- Advancement in the understanding of the structure of ω-automata

Conclusions

- Advancement in the understanding of the structure of ω-automata
- Critical use of transition-based acceptance

Conclusions

- Advancement in the understanding of the structure of ω-automata
- Critical use of transition-based acceptance
- Unexpected appearance of history-deterministic automata!

Conclusions

- Advancement in the understanding of the structure of ω-automata
- Critical use of transition-based acceptance
- Unexpected appearance of history-deterministic automata!
- Use of universal graphs to show positionality

Conclusions

- Advancement in the understanding of the structure of ω-automata
- Critical use of transition-based acceptance
- Unexpected appearance of history-deterministic automata!
- Use of universal graphs to show positionality

Thanks for your attention!