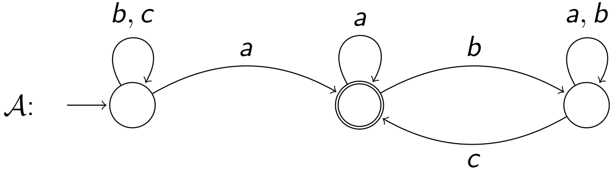
History-deterministic and Explorable Automata

Denis Kuperberg
Marc Bagnol, Udi Boker, Emile Hazard, Olivier Idir, Orna Kupferman, Karoliina Lehtinen, Michał Skrzypczak,. . .

GREYC, Caen, May 14th 2024

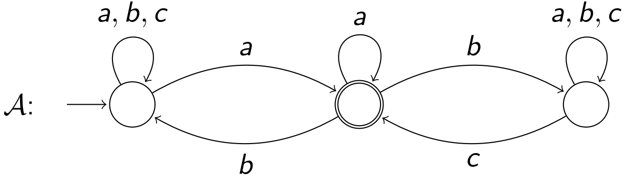
Automata



Deterministic

$$L(\mathcal{A}) = \{\text{accepted words}\}$$

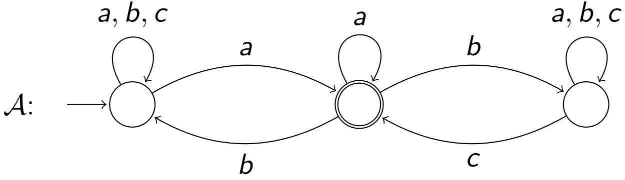
Automata



Non-deterministic (ND)

$$L(\mathcal{A}) = \{\text{accepted words}\}$$

Automata

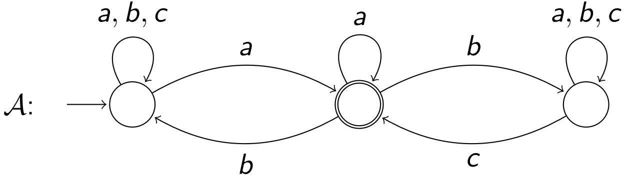


Non-deterministic (ND)

$$L(\mathcal{A}) = \{\text{accepted words}\}$$

Büchi: ∞ accepting states

Automata

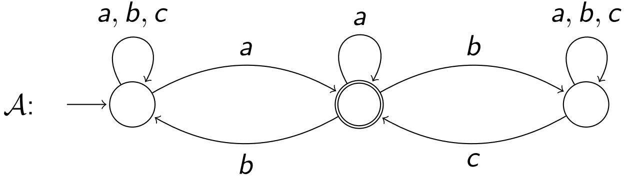


Non-deterministic (ND)

$$L(\mathcal{A}) = \{\text{accepted words}\}$$

Büchi: ∞ accepting states
coBüchi: finitely many non-accepting states

Automata



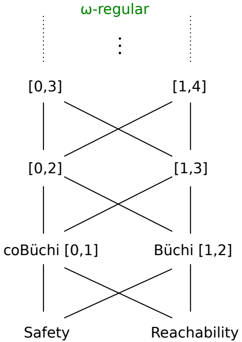
Non-deterministic (ND)

$$L(\mathcal{A}) = \{\text{accepted words}\}$$

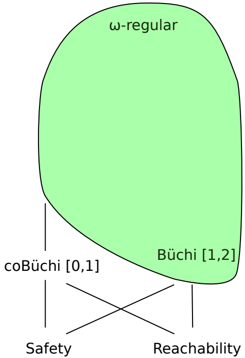
Büchi: ∞ accepting states
coBüchi: finitely many non-accepting states

for coBüchi: $L(\mathcal{A}) \subseteq (a + b + c)^* a^\omega$

The Parity hierarchy

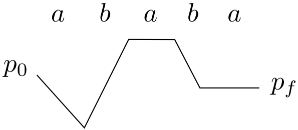


Deterministic

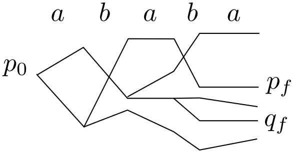


Non-deterministic

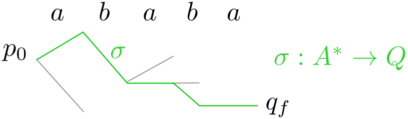
History-Deterministic Automata



Deterministic

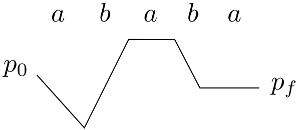


Non-deterministic

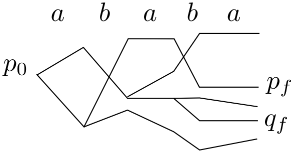


History-Deterministic
(a.k.a. Good-for-Games)

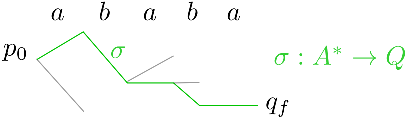
History-Deterministic Automata



Deterministic



Non-deterministic



History-Deterministic
(a.k.a. Good-for-Games)

Context

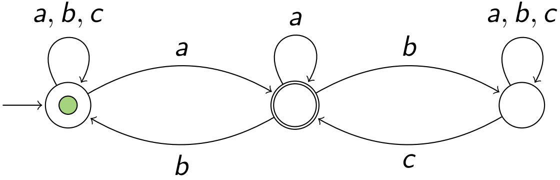
- Introduced in [Henzinger, Piterman 2006] as "Good-for-Games" and in [Colcombet 2009] as "History-determinism".
- Solve Church Synthesis more efficiently
- Intermediate model between Det. and Nondet.

Definition of HD via a game

$\mathcal{A}$ ND automaton on finite or infinite words.

Letter game of $\mathcal{A}$:
Adam plays letters:

Eve: resolves non-deterministic choices for transitions



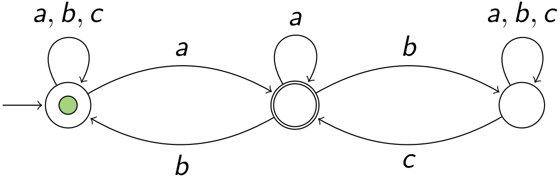
Definition of HD via a game

$\mathcal{A}$ ND automaton on finite or infinite words.

Letter game of $\mathcal{A}$:

Adam plays letters: a

Eve: resolves non-deterministic choices for transitions

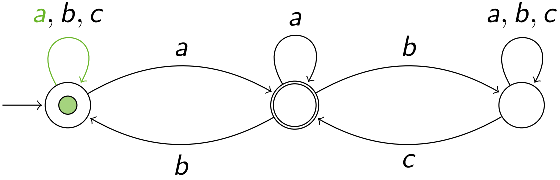


Definition of HD via a game

$\mathcal{A}$ ND automaton on finite or infinite words.

Letter game of $\mathcal{A}$:
Adam plays letters: a

Eve: resolves non-deterministic choices for transitions



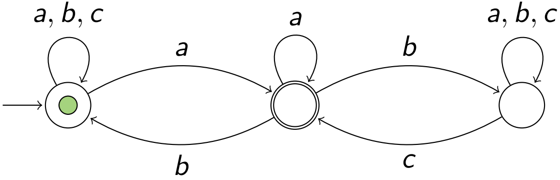
Definition of HD via a game

$\mathcal{A}$ ND automaton on finite or infinite words.

Letter game of $\mathcal{A}$:

Adam plays letters: a a

Eve: resolves non-deterministic choices for transitions

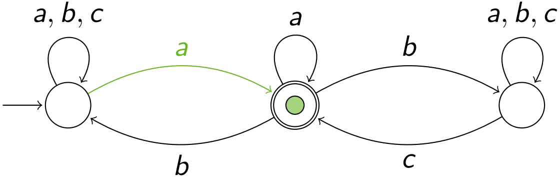


Definition of HD via a game

$\mathcal{A}$ ND automaton on finite or infinite words.

Letter game of $\mathcal{A}$:
Adam plays letters: a a

Eve: resolves non-deterministic choices for transitions



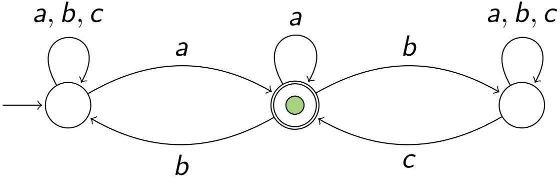
Definition of HD via a game

$\mathcal{A}$ ND automaton on finite or infinite words.

Letter game of $\mathcal{A}$:

Adam plays letters: $a \; a \; b$

Eve: resolves non-deterministic choices for transitions

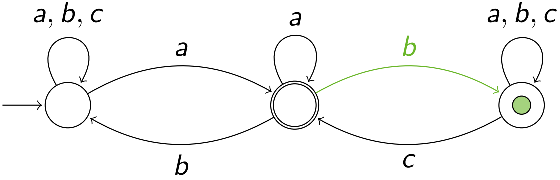


Definition of HD via a game

$\mathcal{A}$ ND automaton on finite or infinite words.

Letter game of $\mathcal{A}$:
Adam plays letters: $a \;\; a \;\; b$

Eve: resolves non-deterministic choices for transitions

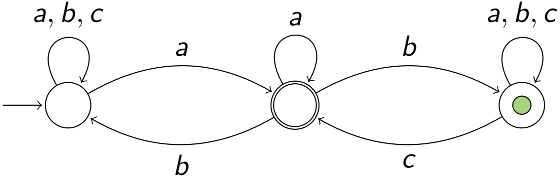


Definition of HD via a game

$\mathcal{A}$ ND automaton on finite or infinite words.

Letter game of $\mathcal{A}$:
Adam plays letters: $a \ a \ b \ c$

Eve: resolves non-deterministic choices for transitions

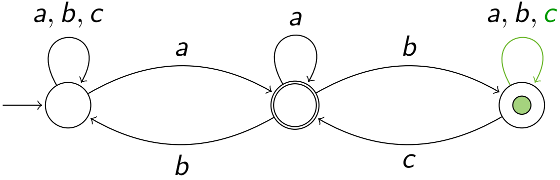


Definition of HD via a game

$\mathcal{A}$ ND automaton on finite or infinite words.

Letter game of $\mathcal{A}$:
Adam plays letters: $a \;\; a \;\; b \;\; c$

Eve: resolves non-deterministic choices for transitions

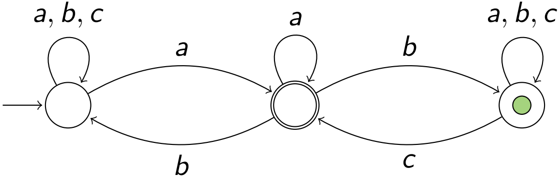


Definition of HD via a game

$\mathcal{A}$ ND automaton on finite or infinite words.

Letter game of $\mathcal{A}$:
Adam plays letters: $a \;\; a \;\; b \;\; c \;\; c$

Eve: resolves non-deterministic choices for transitions



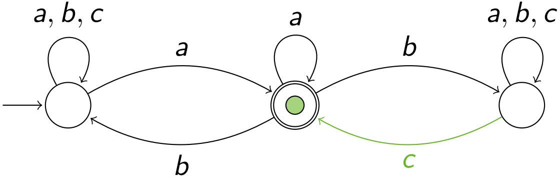
Definition of HD via a game

$\mathcal{A}$ ND automaton on finite or infinite words.

Letter game of $\mathcal{A}$:

Adam plays letters: $a \;\; a \;\; b \;\; c \;\; c$

Eve: resolves non-deterministic choices for transitions



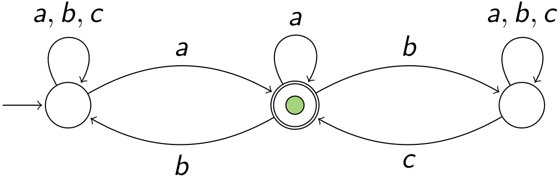
Definition of HD via a game

$\mathcal{A}$ ND automaton on finite or infinite words.

Letter game of $\mathcal{A}$:

Adam plays letters: $a \; a \; b \; c \; c \; \ldots \; = w$

Eve: resolves non-deterministic choices for transitions



Eve wins if: $w \in L(\mathcal{A}) \Rightarrow$ Run accepting.

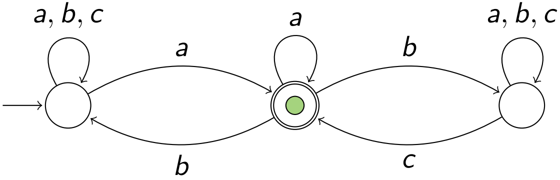
Definition of HD via a game

$\mathcal{A}$ ND automaton on finite or infinite words.

<u>**Letter game of $\mathcal{A}$**</u>:

Adam plays letters: $a \;\; a \;\; b \;\; c \;\; c \;\; \ldots \quad = \; w$

Eve: resolves non-deterministic choices for transitions



Eve wins if: $w \in L(\mathcal{A}) \Rightarrow$ Run accepting.

$\mathcal{A}$ HD $\Leftrightarrow$ Eve wins the Letter game on $\mathcal{A}$

$\Leftrightarrow$ there is a strategy $\sigma_{\mathrm{HD}} : A^* \to Q$ accepting all words of $L(\mathcal{A})$.

First results

Definition (Determinizable By Pruning)

$\mathcal{A}$ DBP if it embeds an equivalent deterministic automaton.

Fact

DBP $\Rightarrow$ HD.

Case where σ_{HD} does not need memory.

First results

Definition (Determinizable By Pruning)

$\mathcal{A}$ DBP if it embeds an equivalent deterministic automaton.

Fact

$\mathrm{DBP} \Rightarrow \mathrm{HD}$.

Case where σ_{HD} does not need memory.

Theorem (PTime inclusion check)

If $\mathcal{A}$ ND and $\mathcal{B}$ HD, checking $L(\mathcal{A}) \subseteq L(\mathcal{B})$ is in PTIME.

(PSPACE-complete for $\mathcal{B}$ ND)

First results

Definition (Determinizable By Pruning)

$\mathcal{A}$ DBP if it embeds an equivalent deterministic automaton.

Fact

$\mathrm{DBP} \Rightarrow \mathrm{HD}$.

Case where σ_{HD} does not need memory.

Theorem (PTime inclusion check)

If $\mathcal{A}$ ND and $\mathcal{B}$ HD, checking $L(\mathcal{A}) \subseteq L(\mathcal{B})$ is in PTIME.

(PSPACE-complete for $\mathcal{B}$ ND)

No need to know σ_{HD} !

First results

Definition (Determinizable By Pruning)

$\mathcal{A}$ DBP if it embeds an equivalent deterministic automaton.

Fact

DBP $\Rightarrow$ HD.

Case where σ_{HD} does not need memory.

Theorem (PTime inclusion check)

If $\mathcal{A}$ ND and $\mathcal{B}$ HD, checking $L(\mathcal{A}) \subseteq L(\mathcal{B})$ is in PTIME.

(PSPACE-complete for $\mathcal{B}$ ND)
No need to know σ_{HD} !

Theorem (Deterministic expressivity)

Any HD automaton can be determinized with exponential blow-up, while preserving its acceptance condition.

Relationship to deterministic automata

Finite words:
$\mathrm{HD} = \mathrm{DBP}$ [Löding]

Relationship to deterministic automata

Finite words:
HD = DBP [Löding]

Büchi (aka Parity [1,2]):

- HD $\neq$ DBP [Boker, K., Kupferman, Skrzypczak 2013]
- Determinization in $O(n^2)$ states [K., Skrzypczak 2015], $O(n)$ conjectured.
- Determinization in PTIME [Acharya, Jurdziński, Prakash 2024]

Relationship to deterministic automata

Finite words:
HD = DBP [Löding]

Büchi (aka Parity [1,2]):

- HD ≠ DBP [Boker, K., Kupferman, Skrzypczak 2013]
- Determinization in $O(n^2)$ states [K., Skrzypczak 2015], $O(n)$ conjectured.
- Determinization in PTime [Acharya, Jurdziński, Prakash 2024]

coBüchi (aka Parity [0,1]):

- Exponential succinctness of HD vs Det. [K., Skrzypczak 2015]
- PTime minimization [Abu Radi, Kupferman 2020]

Recognizing HD automata

Complexity of the HDness problem:
Input: A nondeterministic automaton $\mathcal{A}$
Output: Is $\mathcal{A}$ HD?

Recognizing HD automata

Complexity of the HDness problem:
Input: A nondeterministic automaton $\mathcal{A}$
Output: Is $\mathcal{A}$ HD?

- On finite words: PTIME [Löding]

Recognizing HD automata

Complexity of the HDness problem:
Input: A nondeterministic automaton $\mathcal{A}$
Output: Is $\mathcal{A}$ HD?

- On finite words: PTIME [Löding]
- On infinite words: Open problem !

Recognizing HD automata

Complexity of the HDness problem:
Input: A nondeterministic automaton $\mathcal{A}$
Output: Is $\mathcal{A}$ HD?

- On finite words: PTime [Löding]
- On infinite words: Open problem !
  - Upper bound: ExpTime [Henzinger, Piterman 2006]

Recognizing HD automata

Complexity of the HDness problem:
Input: A nondeterministic automaton $\mathcal{A}$
Output: Is $\mathcal{A}$ HD?

- On finite words: PTime [Löding]
- On infinite words: Open problem !
  - Upper bound: ExpTime [Henzinger, Piterman 2006]
  - PTime algorithm for Büchi [Bagnol, K. 2018]
    works also for coBüchi [Boker, K., Lehtinen, Skrzypczak 2020]
    Conjectured correct for all conditions.

Recognizing HD automata

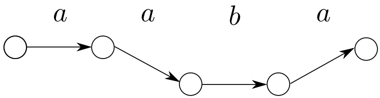
Complexity of the HDness problem:
Input: A nondeterministic automaton $\mathcal{A}$
Output: Is $\mathcal{A}$ HD?

- On finite words: PTIME [Löding]
- On infinite words: Open problem !
  - Upper bound: EXPTIME [Henzinger, Piterman 2006]
  - PTIME algorithm for Büchi [Bagnol, K. 2018]
    works also for coBüchi [Boker, K., Lehtinen, Skrzypczak 2020]
    Conjectured correct for all conditions.

To attack this conjecture and better understand the power of nondeterminism, let us generalize the notion of HD ...

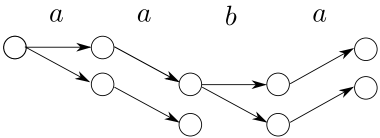
Allowing more runs

Idea: Allow to build several runs, at least one accepting.



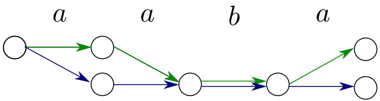
1 state - 1 run

HD



2 states

Width 2
[K., Majumdar 2018]



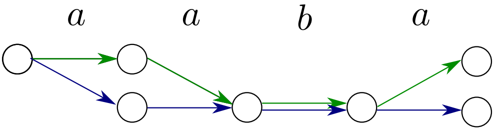
2 runs

2-Explorable
[Hazard, K. 2023]

Explorable Automata

***k*-explorability game**:

Adam plays letters, Eve moves k tokens

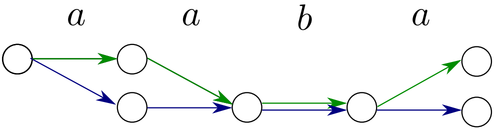


Eve wins if $w \in L(\mathcal{A}) \Rightarrow$ at least one token follows an accepting run.

Explorable Automata

***k*-explorability game**:

Adam plays letters, Eve moves k tokens



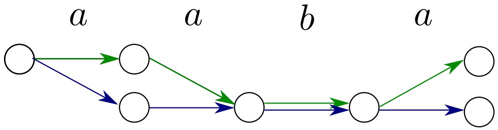
Eve wins if $w \in L(\mathcal{A}) \Rightarrow$ at least one token follows an accepting run.

$\mathcal{A}$ is *k*-explorable if Eve wins the k-explorability game.

Explorable Automata

***k*-explorability game**:

Adam plays letters, Eve moves k tokens



Eve wins if $w \in L(\mathcal{A}) \Rightarrow$ at least one token follows an accepting run.

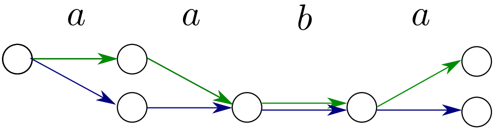
$\mathcal{A}$ is k-explorable if Eve wins the k-explorability game.

$\mathcal{A}$ is explorable if it is k-explorable for some k.

Explorable Automata

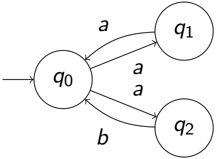
***k*-explorability game**:

Adam plays letters, Eve moves k tokens



Eve wins if $w \in L(\mathcal{A}) \Rightarrow$ at least one token follows an accepting run.

$\mathcal{A}$ is k-explorable if Eve wins the k-explorability game.
$\mathcal{A}$ is explorable if it is k-explorable for some k.

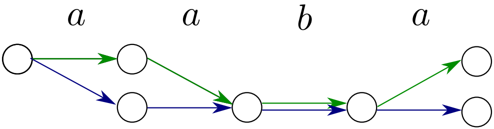


A ?-explorable safety NFA

Explorable Automata

***k*-explorability game**:

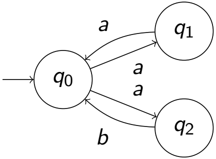
Adam plays letters, Eve moves k tokens



Eve wins if $w \in L(\mathcal{A}) \Rightarrow$ at least one token follows an accepting run.

$\mathcal{A}$ is k-explorable if Eve wins the k-explorability game.

$\mathcal{A}$ is explorable if it is k-explorable for some k.



A non-explorable safety NFA

First results

Theorem [K., Majumdar '18]:
Deciding $|Q|/2$-explorability is EXPTIME-complete.

First results

Theorem [K., Majumdar '18]:
Deciding $|Q|/2$-explorability is ExpTime-complete.

Motivating Theorem [Hazard, K. 2023]

The HDness problem is in PTime for explorable automata.

First results

Theorem [K., Majumdar '18]:
Deciding $|Q|/2$-explorability is $\textsc{ExpTime}$-complete.

Motivating Theorem [Hazard, K. 2023]

The HDness problem is in $\textsc{PTime}$ for explorable automata.

Can we decide explorability ? If yes, how efficiently ?

First results

Theorem [K., Majumdar '18]:
Deciding $|Q|/2$-explorability is $\textsc{ExpTime}$-complete.

Motivating Theorem [Hazard, K. 2023]

The HDness problem is in $\textsc{PTime}$ for explorable automata.

Can we decide explorability ? If yes, how efficiently ?

If better than $\textsc{ExpTime}$: improve on general HDness !

First results

Theorem [K., Majumdar '18]:
Deciding $|Q|/2$-explorability is $\textsc{ExpTime}$-complete.

Motivating Theorem [Hazard, K. 2023]

The HDness problem is in $\textsc{PTime}$ for explorable automata.

Can we decide explorability ? If yes, how efficiently ?

If better than $\textsc{ExpTime}$: improve on general HDness !

How many tokens might be needed in explorable automata ?

A related paper

Similar questions in [Betrand et al 2019: Controlling a population]

k-**population game**: Arena like *k*-explorability game on NFA,
Goal of Adam: bring all tokens to a sink state.

A related paper

Similar questions in [Betrand et al 2019: Controlling a population]

k-**population game**: Arena like *k*-explorability game on NFA,
Goal of Adam: bring all tokens to a sink state.

Population Control Problem (PCP): $\exists k$ s.t. Eve wins ?

A related paper

Similar questions in [Betrand et al 2019: Controlling a population]

k-**population game**: Arena like *k*-explorability game on NFA,
Goal of Adam: bring all tokens to a sink state.

Population Control Problem (PCP): $\exists k$ s.t. Eve wins ?

Results in [Bertrand, Dewaskar, Genest, Gimbert, Godbole]:

- The PCP is ExpTime-complete
- Doubly exponentially many tokens might be needed.

A related paper

Similar questions in [Betrand et al 2019: Controlling a population]

k-**population game**: Arena like *k*-explorability game on NFA,
Goal of Adam: bring all tokens to a sink state.

Population Control Problem (PCP): $\exists k$ s.t. Eve wins ?

Results in [Bertrand, Dewaskar, Genest, Gimbert, Godbole]:

- The PCP is EXPTIME-complete
- Doubly exponentially many tokens might be needed.

Our goal: Generalize to Explorability, but

- Game harder to solve: the input word has to be in $L(\mathcal{A})$
- Must deal with acceptance conditions on infinite words.

Results

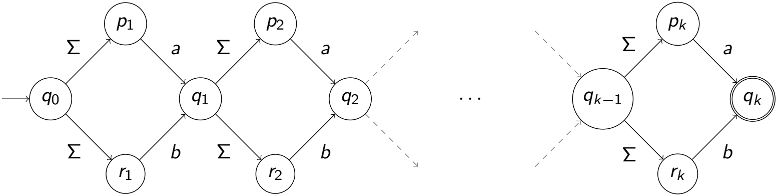
Theorems [Hazard, K. 2023]

Explorability is EXPTIME-complete for NFA, Büchi.
Doubly exponentially many tokens might be needed.

Results

Theorems [Hazard, K. 2023]

Explorability is EXPTIME-complete for NFA, Büchi.
Doubly exponentially many tokens might be needed.

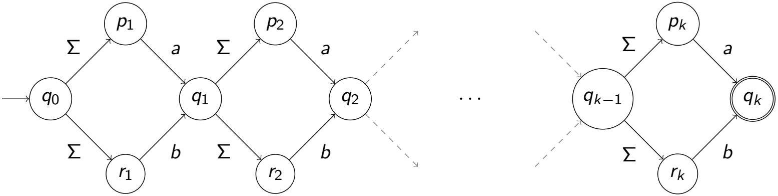


NFA needing exponentially many tokens.

Results

Theorems [Hazard, K. 2023]

Explorability is EXPTIME-complete for NFA, Büchi.
Doubly exponentially many tokens might be needed.



NFA needing exponentially many tokens.

Theorems [Idir, K.]

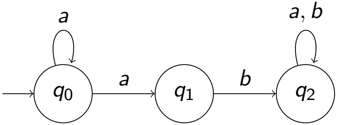
Explorability is EXPTIME for coBüchi, $[0, 2]$-Parity.

ω-explorability

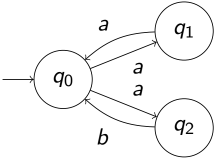
What happens if we allow a countable infinity of tokens ?

ω-explorability

What happens if we allow a countable infinity of tokens ?



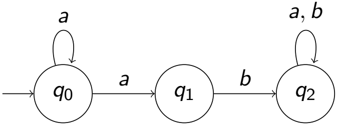
not explorable but
ω-explorable



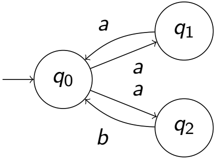
not ω-explorable

ω-explorability

What happens if we allow a countable infinity of tokens ?



not explorable but
ω-explorable



not ω-explorable

Intuition:
Non-ω-explorable: Adam can always kill any run

Results on ω-explorability

Facts:

- any NFA is ω-explorable,
- any automaton $\mathcal{A}$ with $L(\mathcal{A})$ countable is ω-explorable.
- any Reachability automaton is ω-explorable,

Results on ω-explorability

Facts:

- any NFA is ω-explorable,
- any automaton $\mathcal{A}$ with $L(\mathcal{A})$ countable is ω-explorable.
- any Reachability automaton is ω-explorable,

Theorem [Hazard, K. 2023]

ω-explorability is $\textsc{ExpTime}$-complete for safety, coBüchi.

Results on ω-explorability

Facts:

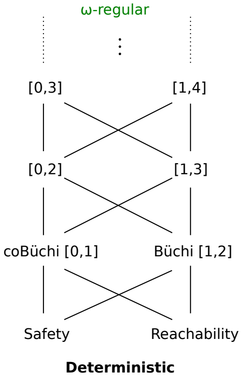
- any NFA is ω-explorable,
- any automaton $\mathcal{A}$ with $L(\mathcal{A})$ countable is ω-explorable.
- any Reachability automaton is ω-explorable,

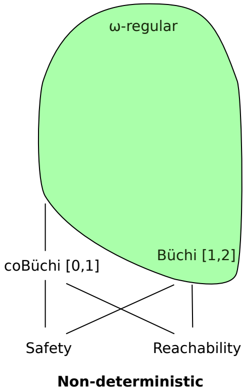
Theorem [Hazard, K. 2023]

ω-explorability is ExpTime-complete for safety, coBüchi.

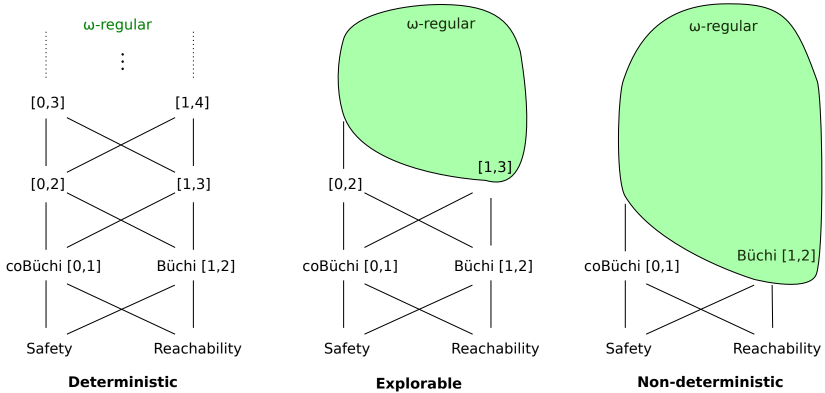
Decidability open for Büchi.

Expressivity of explorable automata

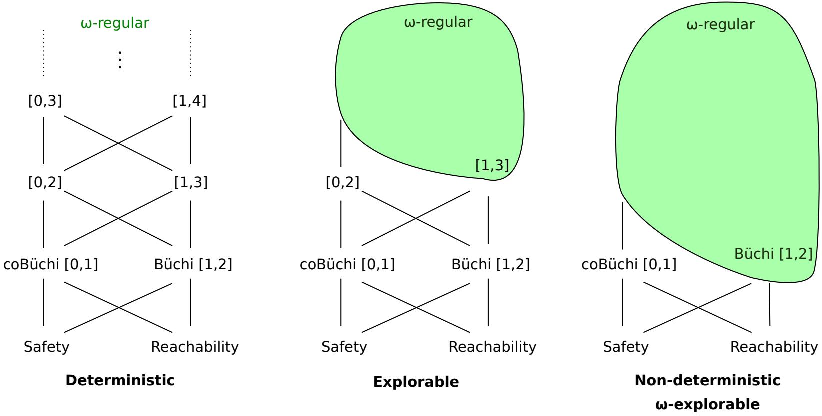




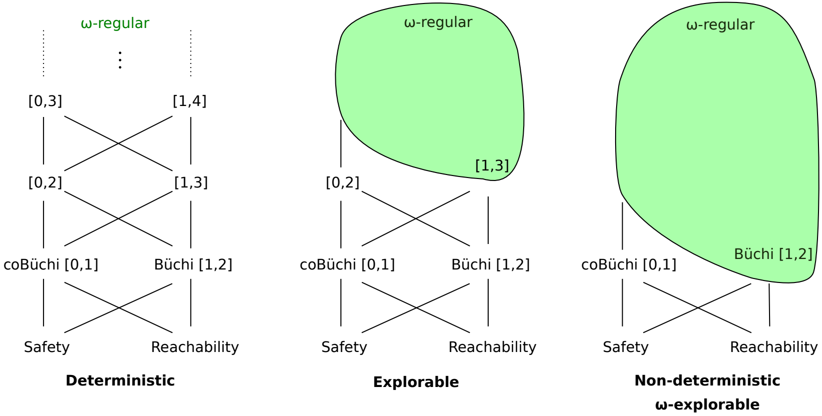
Expressivity of explorable automata



Expressivity of explorable automata



Expressivity of explorable automata



Theorem (Idir, K.)

$[1,3]$-*explorability decidable* $\Leftrightarrow$ *Parity explorability decidable*
Büchi ω-*explorability decidable* $\Leftrightarrow$ *Parity* ω-*explorability decidable*

Future work

- Open decidability: $[1,3]$-expl., Büchi ω-expl.
- Complexity of k-expl. with k in binary?
- Studying HD and expl. models in other frameworks.
- Practical applications, experimental evaluations.
- $\textsc{PTime}$ HDness for parity automata.
- ...

Future work

- Open decidability: $[1,3]$-expl., Büchi ω-expl.
- Complexity of k-expl. with k in binary?
- Studying HD and expl. models in other frameworks.
- Practical applications, experimental evaluations.
- PTIME HDness for parity automata.
- ...

Thanks for your attention!