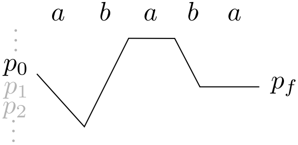
Explorable Automata

Emile Hazard, **Denis Kuperberg**

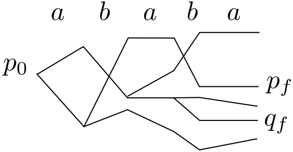
and Marc Bagnol, Udi Boker, Orna Kupferman, Karoliina Lehtinen,
Anirban Majumdar, Michał Skrzypczak, Milla Valnet,. . .

Delta ANR Meeting, Marseille, 30 May 2022

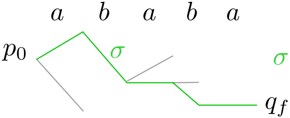
Good-for-Games Automata



Deterministic



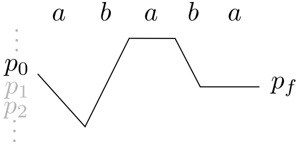
Non-deterministic



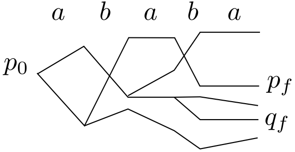
$\sigma : A^* \to Q$

Good-for-Games

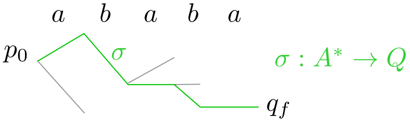
Good-for-Games Automata



Deterministic



Non-deterministic



Good-for-Games

Motivations

- Solve Church Synthesis more efficiently
- Intermediate model between Det. and Nondet.

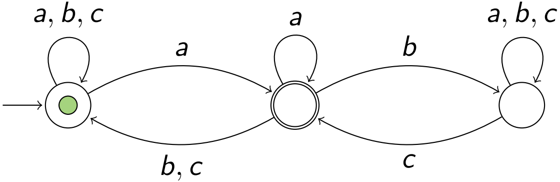
Definition of GFG via a game

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

Letter game of $\mathcal{A}$:

Adam plays letters:

Eve: resolves non-deterministic choices for transitions

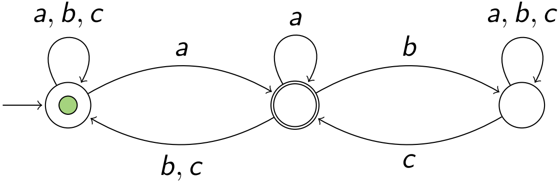


Definition of GFG via a game

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

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

Eve: resolves non-deterministic choices for transitions



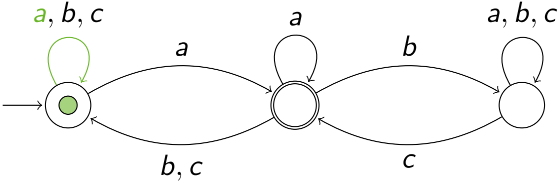
Definition of GFG via a game

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

Letter game of $\mathcal{A}$:

Adam plays letters: a

Eve: resolves non-deterministic choices for transitions

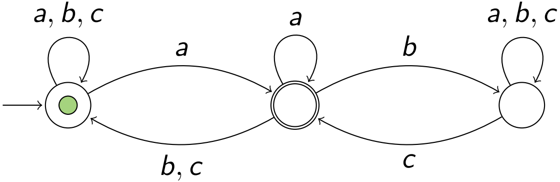


Definition of GFG via a game

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

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

Eve: resolves non-deterministic choices for transitions



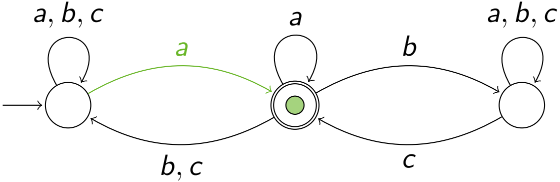
Definition of GFG via a game

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

Letter game of $\mathcal{A}$:

Adam plays letters: a a

Eve: resolves non-deterministic choices for transitions

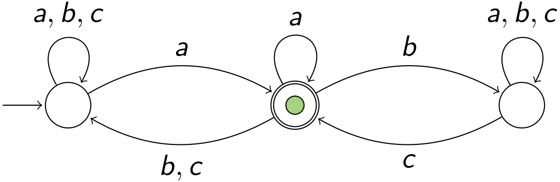


Definition of GFG via a game

$\mathcal{A}$ 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



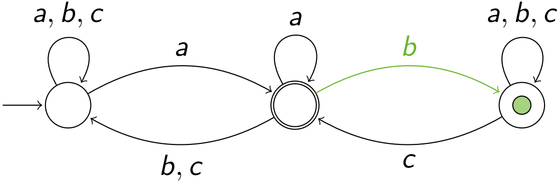
Definition of GFG via a game

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

Letter game of $\mathcal{A}$:

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

Eve: resolves non-deterministic choices for transitions



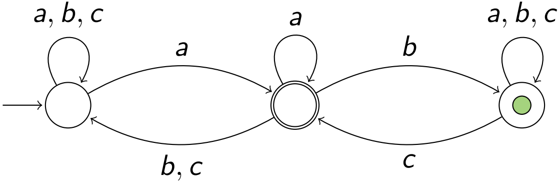
Definition of GFG via a game

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

Letter game of $\mathcal{A}$:

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

Eve: resolves non-deterministic choices for transitions

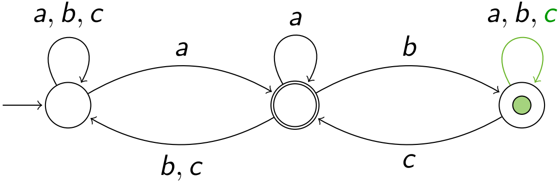


Definition of GFG via a game

$\mathcal{A}$ 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



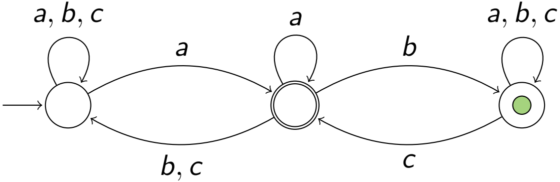
Definition of GFG via a game

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

Letter game of $\mathcal{A}$:

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

Eve: resolves non-deterministic choices for transitions

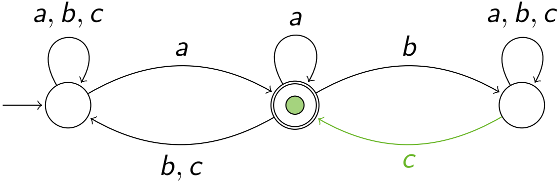


Definition of GFG via a game

$\mathcal{A}$ 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



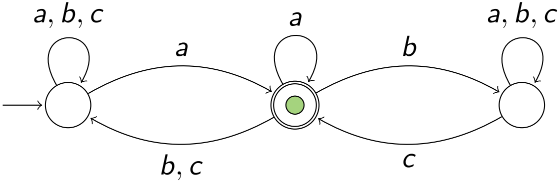
Definition of GFG via a game

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

Letter game of $\mathcal{A}$:

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

Eve: resolves non-deterministic choices for transitions



Eve wins if: $w \in L \Rightarrow$ Run accepting.

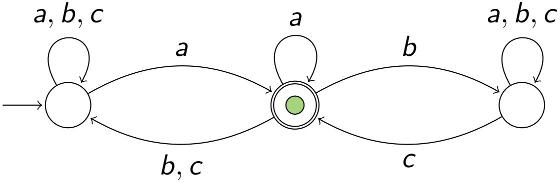
Definition of GFG via a game

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

Letter game of $\mathcal{A}$:

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

Eve: resolves non-deterministic choices for transitions



Eve wins if: $w \in L \Rightarrow$ Run accepting.

$\mathcal{A}$ GFG $\Leftrightarrow$ Eve wins the Letter game on $\mathcal{A}$
$\Leftrightarrow$ there is a strategy $\sigma_{\mathrm{GFG}} : A^* \to Q$ accepting all words of $L(\mathcal{A})$.

First facts

Fact

Every deterministic automaton is GFG.

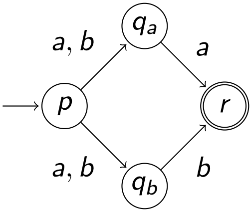
First facts

Fact

Every deterministic automaton is GFG.

Some non-GFG automaton:

$$L = (a + b)(a + b)$$



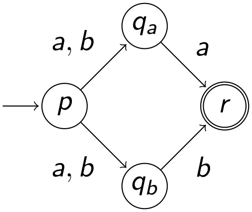
First facts

Fact

Every deterministic automaton is GFG*.*

Some non-GFG automaton:

$$L = (a+b)(a+b)$$



Definition

Nondet automaton $\mathcal{A}$ is **Determinizable by Pruning** (DBP):
Determinizable by removing some transitions.

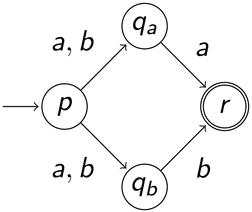
First facts

Fact

Every deterministic automaton is GFG.

Some non-GFG automaton:

$$L = (a+b)(a+b)$$



Definition

Nondet automaton $\mathcal{A}$ is **Determinizable by Pruning** (DBP):
Determinizable by removing some transitions.

Fact

DBP $=$ "GFG *with a positional strategy*".
$\rightarrow$ *Every* DBP *automaton is* GFG.

Some GFG automata

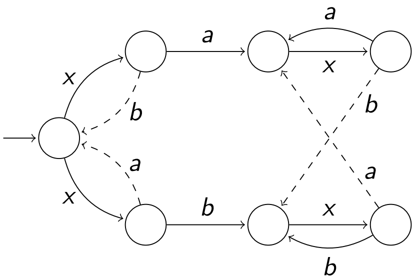
Theorem

On finite words, $\mathrm{DBP} = \mathrm{GFG}$.

Some GFG automata

Theorem
On finite words, DBP $=$ GFG.

Theorem ([Boker, K., Kupferman, Skrzypczak '13])
On infinite words, DBP $\subsetneq$ GFG.

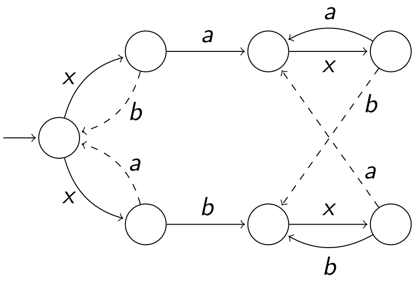


A GFG coBüchi automaton for $(xa + xb)^*[(xa)^\omega + (xb)^\omega]$.

Some GFG automata

Theorem
On finite words, DBP $=$ GFG.

Theorem ([Boker, K., Kupferman, Skrzypczak '13])
On infinite words, DBP $\subsetneq$ GFG.



A GFG coBüchi automaton for $(xa + xb)^*[(xa)^\omega + (xb)^\omega]$.

State-blowup to determinize can be **Exponential** [K., Skrzypczak '15].

Application: Inclusion testing

Simulation game $\mathcal{B} \leq_s \mathcal{A}$: Each round:

- Adam chooses $q \xrightarrow{a} q'$ in $\mathcal{B}$
- Eve has to replicate $p \xrightarrow{a} p'$ in $\mathcal{A}$

Eve wins if $\mathcal{B}$ accepts $\Longrightarrow$ $\mathcal{A}$ accepts

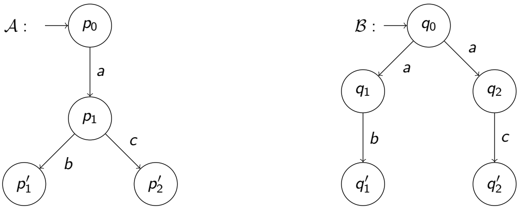
Application: Inclusion testing

Simulation game $\mathcal{B} \leq_s \mathcal{A}$: Each round:

- Adam chooses $q \xrightarrow{a} q'$ in $\mathcal{B}$
- Eve has to replicate $p \xrightarrow{a} p'$ in $\mathcal{A}$

Eve wins if $\mathcal{B}$ accepts $\Longrightarrow$ $\mathcal{A}$ accepts

Example:



$\mathcal{B} \leq_s \mathcal{A}$ but $\mathcal{A} \not\leq_s \mathcal{B}$

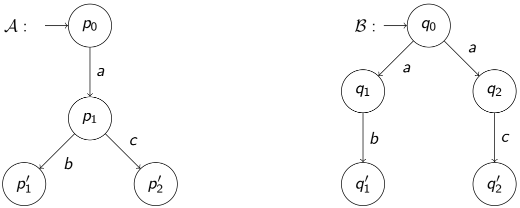
Application: Inclusion testing

Simulation game $\mathcal{B} \leq_s \mathcal{A}$: Each round:

- Adam chooses $q \xrightarrow{a} q'$ in $\mathcal{B}$
- Eve has to replicate $p \xrightarrow{a} p'$ in $\mathcal{A}$

Eve wins if $\mathcal{B}$ accepts $\implies$ $\mathcal{A}$ accepts

Example:



$$\mathcal{B} \leq_s \mathcal{A} \quad \text{but} \quad \mathcal{A} \not\leq_s \mathcal{B}$$

Lemma: If $\mathcal{A}$ GFG, then $\mathcal{B} \leq_s \mathcal{A}$ is equivalent to $L(\mathcal{B}) \subseteq L(\mathcal{A})$

Recognizing GFG automata

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

Recognizing GFG automata

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

- On finite words: $\mathrm{PT{\scriptsize IME}}$ [Löding]

Recognizing GFG automata

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

- On finite words: $\textsc{PTime}$ [Löding]
- On infinite words: Open problem !

Recognizing GFG automata

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

- On finite words: PTIME [Löding]
- On infinite words: Open problem !
  - Upper bound: EXPTIME [Henzinger, Piterman '06]

Recognizing GFG automata

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

- On finite words: PTime [Löding]
- On infinite words: Open problem !
  - Upper bound: ExpTime [Henzinger, Piterman '06]
  - PTime algorithm conjectured to be correct [Bagnol, K. '18]
    Proved correct for Büchi and CoBüchi conditions.

Recognizing GFG automata

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

- On finite words: $\textsc{PTime}$ [Löding]
- On infinite words: Open problem !
  - Upper bound: $\textsc{ExpTime}$ [Henzinger, Piterman '06]
  - $\textsc{PTime}$ algorithm conjectured to be correct [Bagnol, K. '18]
    Proved correct for Büchi and CoBüchi conditions.

What about building GFG automata ?

Recognizing GFG automata

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

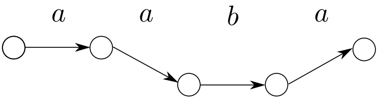
- On finite words: PTIME [Löding]
- On infinite words: Open problem !
  - Upper bound: EXPTIME [Henzinger, Piterman '06]
  - PTIME algorithm conjectured to be correct [Bagnol, K. '18]
    Proved correct for Büchi and CoBüchi conditions.

What about building GFG automata ?

To tackle these questions, we generalize the notion of GFG...

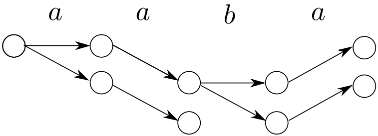
Allowing more runs

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



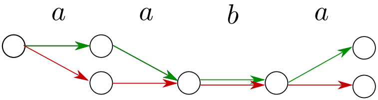
1 state - 1 run

GFG



2 states

Width 2

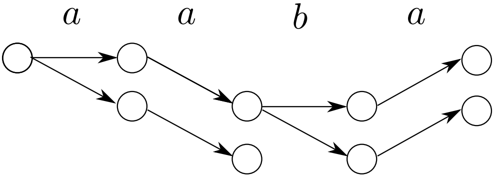


2 runs

2-Explorable

Width of an automaton

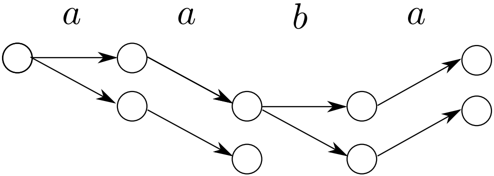
k-width game on $\mathcal{A}$:



Eve wins if $w \in L(\mathcal{A}) \Rightarrow$ her run-DAG contains an accepting run.

Width of an automaton

k-width game on $\mathcal{A}$:

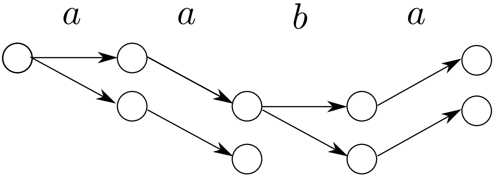


Eve wins if $w \in L(\mathcal{A}) \Rightarrow$ her run-DAG contains an accepting run.

Width of $\mathcal{A}$: Smallest k s.t. Eve wins the k-width game (at most $|Q|$).

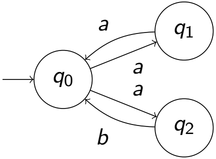
Width of an automaton

k-width game on $\mathcal{A}$:



Eve wins if $w \in L(\mathcal{A}) \Rightarrow$ her run-DAG contains an accepting run.

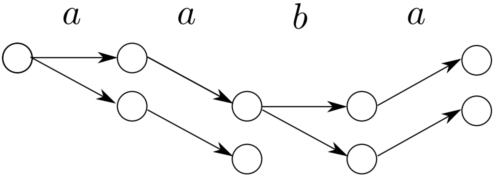
Width of $\mathcal{A}$: Smallest k s.t. Eve wins the k-width game (at most $|Q|$).



A safety NFA of width ?

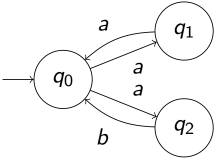
Width of an automaton

k-width game on $\mathcal{A}$:



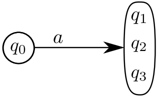
Eve wins if $w \in L(\mathcal{A}) \Rightarrow$ her run-DAG contains an accepting run.

Width of $\mathcal{A}$: Smallest k s.t. Eve wins the k-width game (at most $|Q|$).

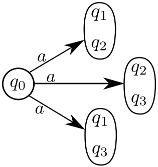


A safety NFA of width 2

k-Determinization

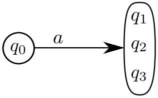


Powerset

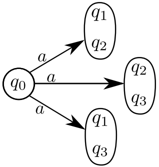


2-Determinization $\mathcal{A}_2$

k-Determinization



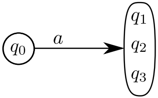
Powerset



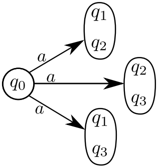
2-Determinization $\mathcal{A}_2$

+ generalization to Breakpoint, Safra: $|\mathcal{A}_k| \approx |\mathcal{A}|^k$

k-Determinization



Powerset



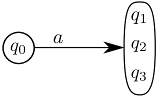
2-Determinization $\mathcal{A}_2$

+ generalization to Breakpoint, Safra: $|\mathcal{A}_k| \approx |\mathcal{A}|^k$

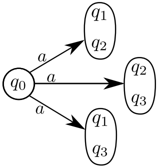
Facts: [K.,Majumdar]

- $\mathrm{width}(\mathcal{A}) \leq k \iff \mathcal{A}_k$ is GFG.
- $\mathcal{B} \subseteq \mathcal{A}$ can be tested in $\approx O(n^{\mathsf{width}(\mathcal{A})})$ via a simulation game.

k-Determinization



Powerset



2-Determinization $\mathcal{A}_2$

+ generalization to Breakpoint, Safra: $|\mathcal{A}_k| \approx |\mathcal{A}|^k$

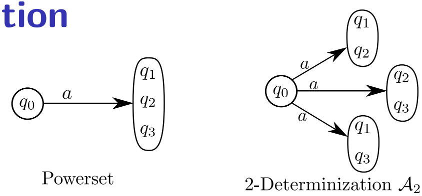
Facts: [K.,Majumdar]

- $\mathrm{width}(\mathcal{A}) \leq k \iff \mathcal{A}_k$ is GFG.
- $\mathcal{B} \subseteq \mathcal{A}$ can be tested in $\approx O(n^{\mathsf{width}(\mathcal{A})})$ via a simulation game.

Application: Building a GFG aut. from $\mathcal{A}$

Start from $\mathcal{B} = \mathcal{A}$ and $k = 1$;
while $\mathcal{B}$ *is not GFG* **do**
    $k := k + 1$;
    $B := \mathcal{A}_k$;
end

k-Determinization



+ generalization to Breakpoint, Safra: $|\mathcal{A}_k| \approx |\mathcal{A}|^k$

Facts: [K.,Majumdar]

- $\mathrm{width}(\mathcal{A}) \leq k \iff \mathcal{A}_k$ is GFG.
- $\mathcal{B} \subseteq \mathcal{A}$ can be tested in $\approx O(n^{\mathsf{width}(\mathcal{A})})$ via a simulation game.

Application: Building a GFG aut. from $\mathcal{A}$

Start from $\mathcal{B} = \mathcal{A}$ and $k = 1$;
while $\mathcal{B}$ *is not GFG* **do**
  $k := k + 1$;
  $B := \mathcal{A}_k$;
end

⚠ We need to test GFGness !

Computing the width

Can we compute $k = \mathrm{width}(\mathcal{A})$ to build $\mathcal{A}_k$ directly ?

Theorem [K, Majumdar]

Computing the width of an NFA is $\textsc{ExpTime}$-complete.
Even deciding whether it is $\leq |Q|/2$.

Computing the width

Can we compute $k = \text{width}(\mathcal{A})$ to build $\mathcal{A}_k$ directly ?

Theorem [K, Majumdar]

Computing the width of an NFA is $\textsc{ExpTime}$-complete.
Even deciding whether it is $\leq |Q|/2$.

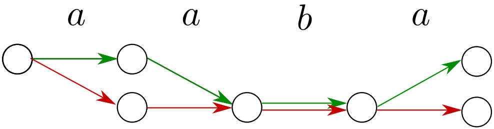
Reduction via a SAT game, introduced by [Robson]
to show $\textsc{ExpTime}$-completeness of popular games:

(Japanese rules)

Explorable Automata

We now bound the number of runs.

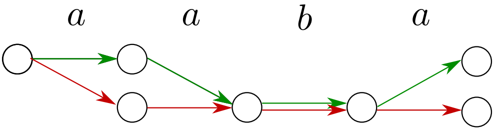
***k*-explorability game**:



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

Explorable Automata

We now bound the number of runs.
k-explorability game:



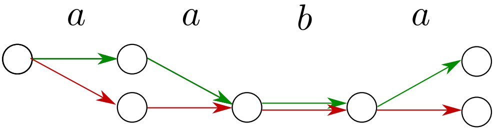
Eve wins if $w \in L(\mathcal{A}) \Rightarrow$ at least one token follows an accepting run.

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

Explorable Automata

We now bound the number of runs.

***k*-explorability game**:



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

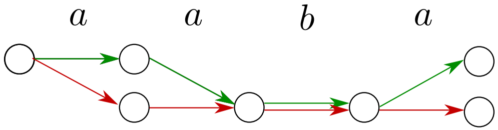
$\mathcal{A}$ is k-explorable if Eve wins the k-explorability game.

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

Explorable Automata

We now bound the number of runs.

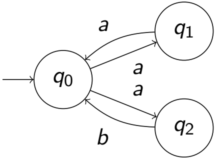
***k*-explorability game**:



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.

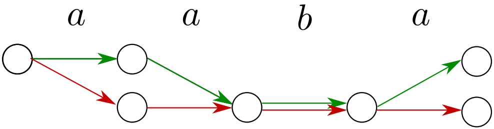


A ?-explorable safety NFA

Explorable Automata

We now bound the number of runs.

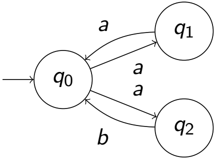
***k*-explorability game**:



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: Deciding $|Q|/2$-explorability is still $\textsc{ExpTime}$-complete.

First results

Theorem: Deciding $|Q|/2$-explorability is still $\textsc{ExpTime}$-complete.

Motivating Theorem [Hazard, K.]

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

First results

Theorem: Deciding $|Q|/2$-explorability is still $\textsc{ExpTime}$-complete.

Motivating Theorem [Hazard, K.]

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

Can we decide explorability ? If yes, how efficiently ?

First results

Theorem: Deciding $|Q|/2$-explorability is still $\textsc{ExpTime}$-complete.

Motivating Theorem [Hazard, K.]

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

Can we decide explorability ? If yes, how efficiently ?

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

First results

Theorem: Deciding $|Q|/2$-explorability is still ExpTime-complete.

> Motivating Theorem [Hazard, K.]
>
> The GFGness problem is in PTime for explorable automata.

Can we decide explorability ? If yes, how efficiently ?

If better than ExpTime: improve on general GFGness !

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.]

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

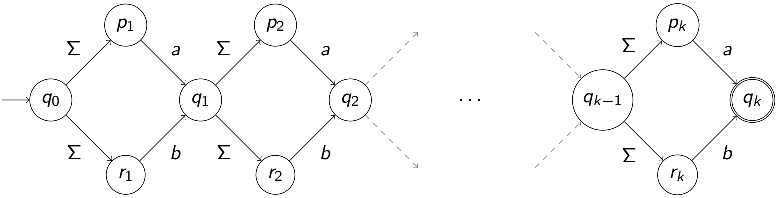
Only up to Büchi condition for now...

Results

Theorems [Hazard, K.]

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

Only up to Büchi condition for now...



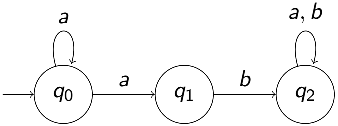
NFA needing exponentially many tokens.

ω-explorability

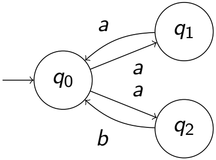
What happens if we allow a countable infinity of tokens ?

ω-explorability

What happens if we allow a countable infinity of tokens ?



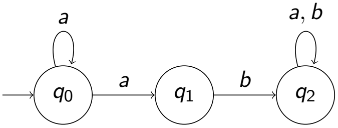
not explorable but
ω-explorable



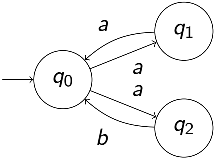
not ω-explorable

ω-explorability

What happens if we allow a countable infinity of tokens ?



not explorable but
ω-explorable



not ω-explorable

Intuition:
Non-explorable: Environment can kill a run **chosen by System**
Non-ω-explorable: Environment can kill a run **of its choice**

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.]

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.]

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

Only up to coBüchi for now... Duality with explorability problem.

Current and future work

- Complexity of (ω)-explorability for parity conditions ?
- Complexity of k-explorability with k in binary?
- Studying GFG and explorable models in other frameworks.
- Practical applications, experimental evaluations.
- $\textsc{PTime}$ GFGness for parity automata.
- ...

Current and future work

- Complexity of (ω)-explorability for parity conditions ?
- Complexity of k-explorability with k in binary?
- Studying GFG and explorable models in other frameworks.
- Practical applications, experimental evaluations.
- PTIME GFGness for parity automata.
- ...

Thanks for your attention!