Is your automaton good for playing games ?

Marc Bagnol, **Denis Kuperberg**

CNRS, LIP, ENS Lyon

Highlights 2018, Berlin

Solving an ω-regular game

Input: G game with ω-regular winning condition $W \subseteq V^\omega$.
Question: Who wins G ? How ?

Solving an ω-regular game

Input: G game with ω-regular winning condition $W \subseteq V^\omega$.
Question: Who wins G ? How ?

Solution: Turn G into a game with more states but simpler acceptance condition.

1. Build Det Parity automaton $\mathcal{A}_{\mathrm{Det}}$ for W,
2. Solve the parity game $G' = G \circ \mathcal{A}_{\mathrm{Det}}$.

Theorem

$G \circ \mathcal{A}_{\mathrm{Det}}$ has same winner as G.

Solving an ω-regular game

Input: G game with ω-regular winning condition $W \subseteq V^\omega$.
Question: Who wins G ? How ?

Solution: Turn G into a game with more states but simpler acceptance condition.

1. Build Det Parity automaton $\mathcal{A}_{\mathrm{Det}}$ for W,
2. Solve the parity game $G' = G \circ \mathcal{A}_{\mathrm{Det}}$.

Theorem
$G \circ \mathcal{A}_{\mathrm{Det}}$ has same winner as G.

Problem: Determinization is expensive. Maybe too strong ?

Solving an ω-regular game

Input: G game with ω-regular winning condition $W \subseteq V^\omega$.
Question: Who wins G ? How ?

Solution: Turn G into a game with more states but simpler acceptance condition.

1. Build Det Parity automaton $\mathcal{A}_{\mathrm{Det}}$ for W,
2. Solve the parity game $G' = G \circ \mathcal{A}_{\mathrm{Det}}$.

Theorem

$G \circ \mathcal{A}_{\mathrm{Det}}$ has same winner as G.

Problem: Determinization is expensive. Maybe too strong ?

Definition (Henzinger, Piterman 2006)

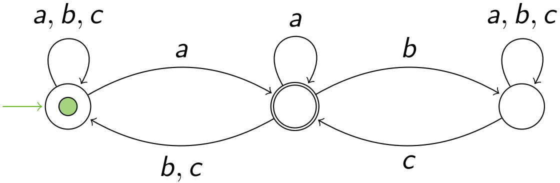
$\mathcal{A}$ is Good-for-Games (GFG) if $G \circ \mathcal{A}$ has same winner as G, for any game G with winning condition $L(\mathcal{A})$.

Definition of GFG via a game

$\mathcal{A}$ Parity automaton, we associate to it a **GFG game**:

Adam plays letters:

Eve: resolves non-deterministic choices for transitions

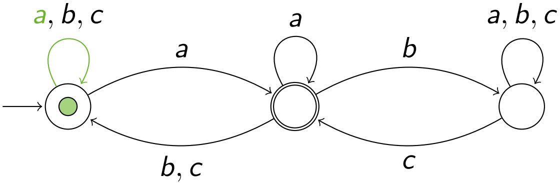


Definition of GFG via a game

$\mathcal{A}$ Parity automaton, we associate to it a **GFG game**:

Adam plays letters: a

Eve: resolves non-deterministic choices for transitions

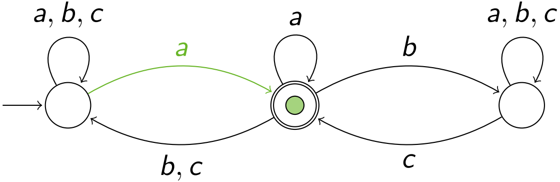


Definition of GFG via a game

$\mathcal{A}$ Parity automaton, we associate to it a **GFG game**:

Adam plays letters: a a

Eve: resolves non-deterministic choices for transitions

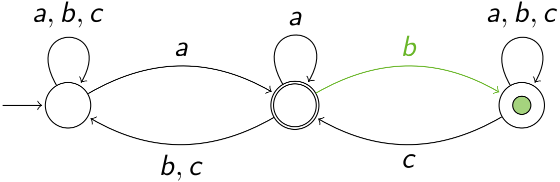


Definition of GFG via a game

$\mathcal{A}$ Parity automaton, we associate to it a **GFG game**:

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

Eve: resolves non-deterministic choices for transitions

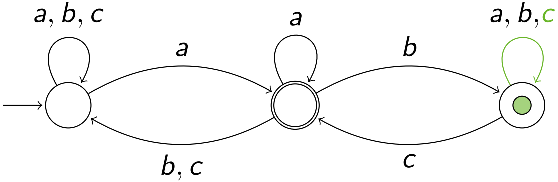


Definition of GFG via a game

$\mathcal{A}$ Parity automaton, we associate to it a **GFG game**:

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

Eve: resolves non-deterministic choices for transitions

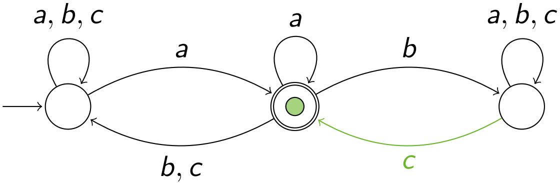


Definition of GFG via a game

$\mathcal{A}$ Parity automaton, we associate to it a **GFG game**:

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

Eve: resolves non-deterministic choices for transitions

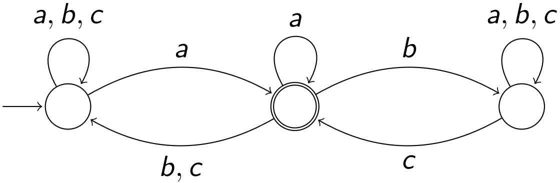


Definition of GFG via a game

$\mathcal{A}$ Parity automaton, we associate to it a **GFG game**:

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

Eve: resolves non-deterministic choices for transitions



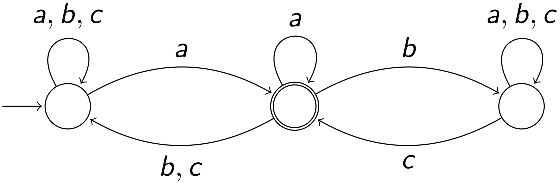
Eve wins if: $w \in L \Rightarrow$ Run accepting.

Definition of GFG via a game

$\mathcal{A}$ Parity automaton, we associate to it a **GFG game**:

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

Eve: resolves non-deterministic choices for transitions



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

$\mathcal{A}$ **GFG** $\Leftrightarrow$ Eve wins the GFG game on $\mathcal{A}$.

Recognizing GFG automata

GFGness problem: input $\mathcal{A}_{\mathrm{ND}}$, is it **GFG**?

Recognizing GFG automata

GFGness problem: input $\mathcal{A}_{\mathrm{ND}}$, is it **GFG**?

Solve the GFG game ?

Recognizing GFG automata

GFGness problem: input $\mathcal{A}_{\mathrm{ND}}$, is it **GFG**?

Solve the GFG game ?
Acceptance condition of the form "$u \in L \Rightarrow$run accepting"

Recognizing GFG automata

GFGness problem: input $\mathcal{A}_{\mathrm{ND}}$, is it **GFG**?

Solve the GFG game ?
Acceptance condition of the form "$u \in L \Rightarrow$run accepting"

Upper bound: EXPTIME

Recognizing GFG automata

GFGness problem: input $\mathcal{A}_{\mathrm{ND}}$, is it **GFG**?

Solve the GFG game ?
Acceptance condition of the form "$u \in L \Rightarrow$run accepting"

Upper bound: EXPTIME

Theorem (Löding)

The GFGness problem is in P for reachability/safety automata.

Theorem (K., Skrzypczak 2015)

The GFGness problem is in P for coBüchi automata.
Parity Games $\equiv$ GFGness for universal Parity automata.

Recognizing GFG automata

GFGness problem: input $\mathcal{A}_{\mathrm{ND}}$, is it **GFG**?

Solve the GFG game ?
Acceptance condition of the form "$u \in L \Rightarrow$run accepting"

Upper bound: EXPTIME

Theorem (Löding)

The GFGness problem is in P for reachability/safety automata.

Theorem (K., Skrzypczak 2015)

The GFGness problem is in P for coBüchi automata.
Parity Games $\equiv$ GFGness for universal Parity automata.

This talk:

Theorem (Bagnol, K. (unpublished))

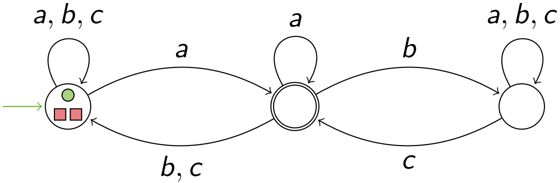
The GFGness problem is in P for Büchi automata.

Abstracting the GFG game

The game G_2:

Adam plays letters:

Eve: moves one token Adam: moves two tokens

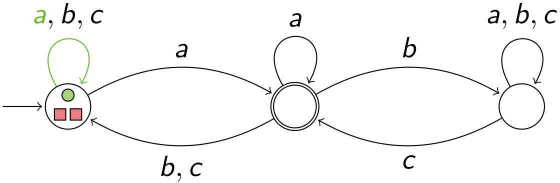


Abstracting the GFG game

The game G_2:

Adam plays letters: a

Eve: moves one token Adam: moves two tokens

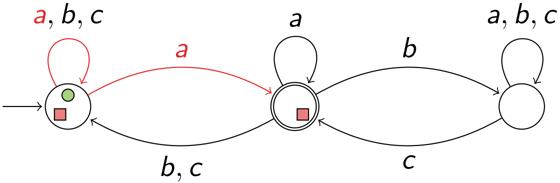


Abstracting the GFG game

The game G_2:

Adam plays letters: a

Eve: moves one token Adam: moves two tokens

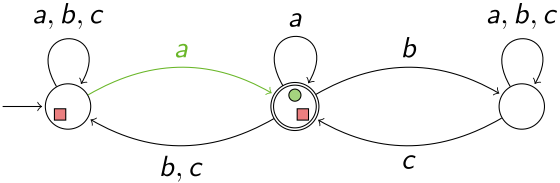


Abstracting the GFG game

The game G_2:

Adam plays letters: a a

Eve: moves one token Adam: moves two tokens

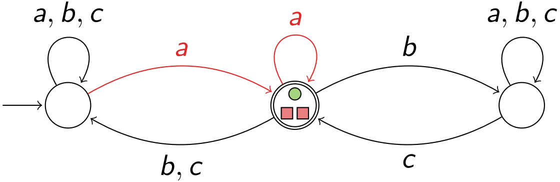


Abstracting the GFG game

The game G_2:

Adam plays letters: a a

Eve: moves one token Adam: moves two tokens

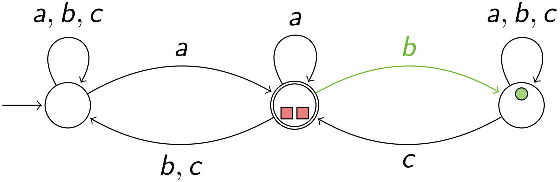


Abstracting the GFG game

The game G_2:

Adam plays letters: a a b

Eve: moves one token Adam: moves two tokens

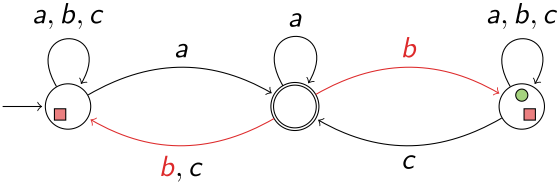


Abstracting the GFG game

The game G_2:

Adam plays letters: a a b

Eve: moves one token Adam: moves two tokens

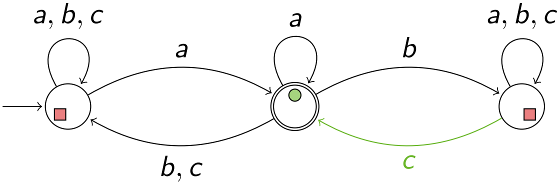


Abstracting the GFG game

The game G_2:

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

Eve: moves one token Adam: moves two tokens

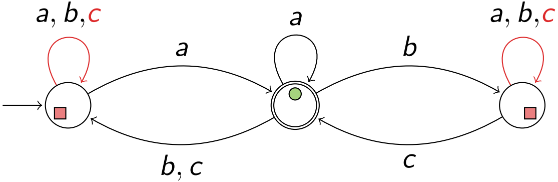


Abstracting the GFG game

The game G_2:

Adam plays letters: a a b c

Eve: moves one token Adam: moves two tokens

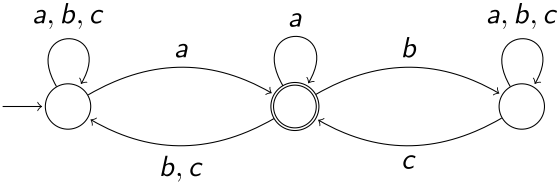


Abstracting the GFG game

The game G_2:

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

Eve: moves one token Adam: moves two tokens



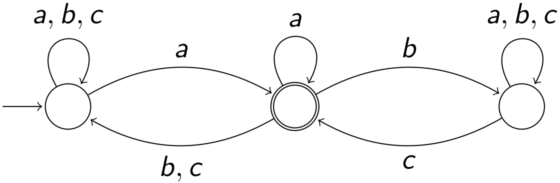
Eve wins if: $\blacksquare_1$ or $\blacksquare_2$ accepts $\Rightarrow$ ● accepts.

Abstracting the GFG game

The game G_2:

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

Eve: moves one token Adam: moves two tokens



Eve wins if: $\blacksquare_1$ or $\blacksquare_2$ accepts ⇒ ● accepts.

Goal: Show that Eve wins G_2 ⇔ Eve wins the GFG game.

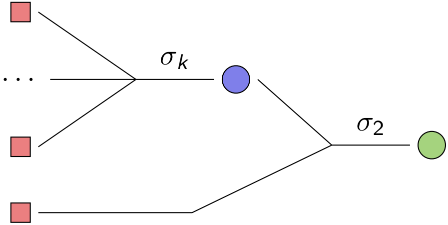
Proof sketch

Lemma

Eve wins $G_2 \Leftrightarrow$ Eve wins G_k for all k.

Proof sketch: how to win against $k+1$ tokens:

- play a *virtual token* ● against the first k tokens
- play the G_2 strategy against the virtual token and the remaining token.



Main proof sketch for $G_2 \Leftrightarrow \mathrm{GFG}$

Assume:

- Adam wins the GFG game with finite-memory strategy τ_{GFG}.
- Eve wins $G_2 \Rightarrow$ wins G_k with strategy σ_k, for a big k.

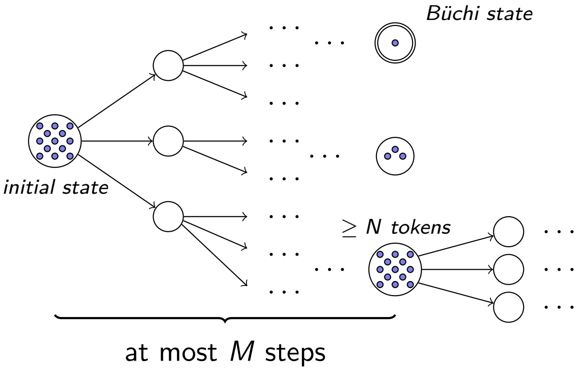
Main proof sketch for $G_2 \Leftrightarrow \mathrm{GFG}$

Assume:

- Adam wins the GFG game with finite-memory strategy τ_{GFG}.
- Eve wins $G_2 \Rightarrow$ wins G_k with strategy σ_k, for a big k.

Build strategy for Eve against τ_{GFG}:

- move k virtual tokens ● against τ_{GFG}
- play σ_k against these k tokens



Conclusion

Results

- Characterisation of Büchi GFG automata via G_2.
- $\rightarrow$ Büchi GFGness $\in$ P, actually in $O(n^4 m(n+m)|\Sigma|^2)$.

Perspectives

- is G_2 equivalent to GFG for coBüchi, Parity ?
- if Yes, Parity GFGness $\in$ P for any fixed parity condition.
- recognizing GFG automata $\rightarrow$ building them.