On a Correspondence Between Memory Structures for Muller Games and Rabin Automata

Antonio Casares

Séminaire de l'équipe MOVE

2 December 2021

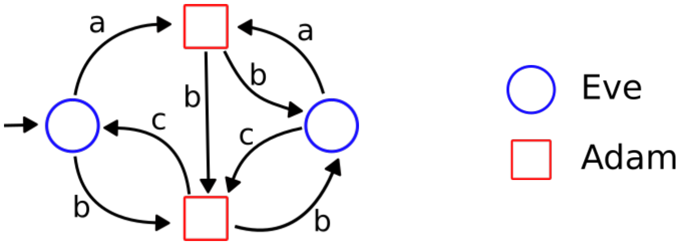
Based in work with Thomas Colcombet and Karoliina Lehtinen.

Index

Index

Infinite Duration Games

$$\mathcal{G} = (V = V_{\mathrm{Eve}} \biguplus V_{\mathrm{Adam}}, v_0, E)$$



Players move a token in turns producing an infinite word $w \in \{a, b, c\}^\omega$.

Muller Conditions

$\mathcal{C}$ → Set of colours (for example, $\mathcal{C} = \{a, b, c\}$).

Muller condition: Family of subsets of colours $\mathcal{F} \subseteq 2^{\mathcal{C}}$.

An infinite word $w \in \mathcal{C}^\omega$ belongs to the Muller condition if

$$Inf(w) \in \mathcal{F}.$$

Example

Produce both "a" and "c" infinitely often:

$$\mathcal{F} = \{\, \{a, c\}, \{a, b, c\} \,\}.$$

Muller Games

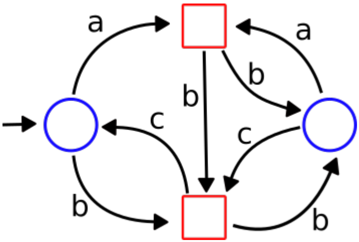


Figure: $\mathcal{F}$-game

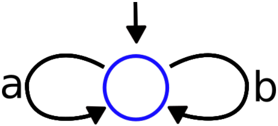
Muller condition:

$$\mathcal{F} = \{\,\{a, c\},\ \{a, b, c\}\,\}.$$

Eve wins if the produced word $w \in \mathcal{C}^\omega$ verifies

$$Inf(w) \in \mathcal{F}.$$

Muller Games Might Require Memory



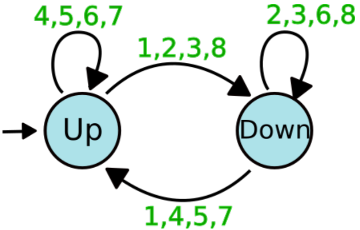
$$\mathcal{F} = \{\{a, b\}\}.$$

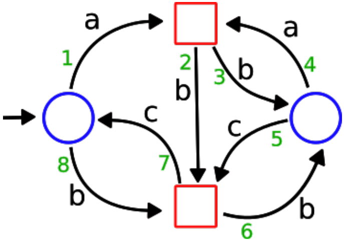
Eve can force a victory, but she needs to remember previous moves.

➔ We use **memory structures**.

General Memory Structures

- Set of states M + initial state.
- $\mu \colon M \times E \to M$, update function.
- $\texttt{next-move} \colon M \times V_{\mathrm{Eve}} \to E$, gives a strategy.

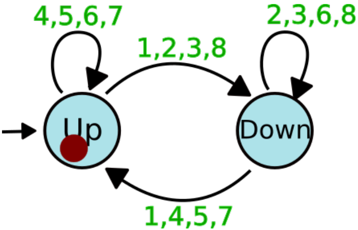


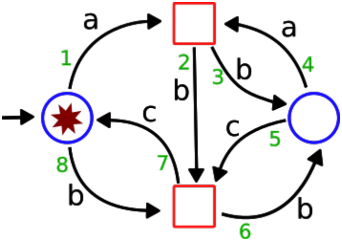


$$\mathcal{F} = \{\, \{a, c\},\ \{a, b, c\}\, \}.$$

General Memory Structures

- Set of states M.
- $\mu\colon M \times E \to M$, update function.
- $\texttt{next-move}\colon M \times V_{\mathrm{Eve}} \to E$, gives a strategy.



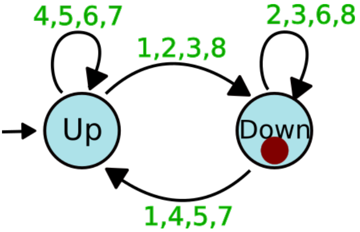


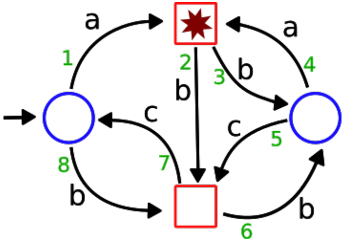
$Output =$

$$\mathcal{F} = \{\, \{a, c\},\ \{a, b, c\}\, \}.$$

General Memory Structures

- Set of states M.
- $\mu: M \times E \to M$, update function.
- $\texttt{next-move}: M \times V_{\mathrm{Eve}} \to E$, gives a strategy.



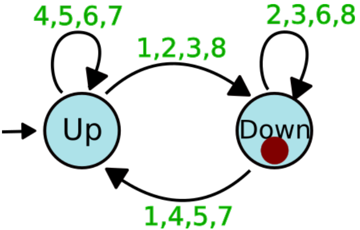


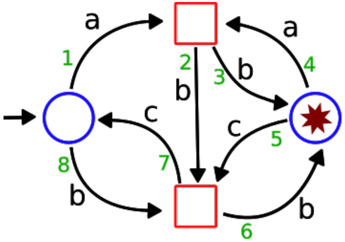
$Output = a$

$$\mathcal{F} = \{\, \{a, c\},\ \{a, b, c\}\,\}.$$

General Memory Structures

- Set of states M.
- $\mu\colon M \times E \to M$, update function.
- $\texttt{next-move}\colon M \times V_{\mathrm{Eve}} \to E$, gives a strategy.



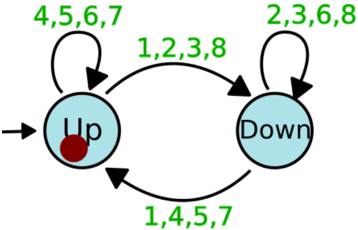


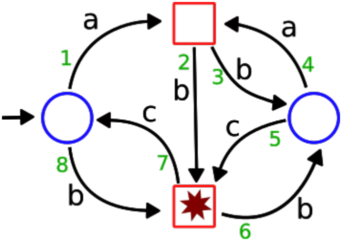
$\mathit{Output} = ab$

$$\mathcal{F} = \{\, \{a, c\},\ \{a, b, c\}\, \}.$$

General Memory Structures

- Set of states M.
- $\mu\colon M \times E \to M$, update function.
- $\texttt{next-move}\colon M \times V_{\mathrm{Eve}} \to E$, gives a strategy.



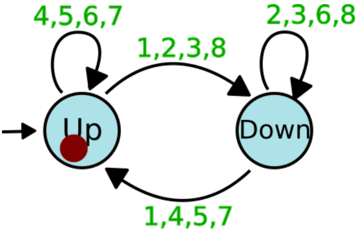


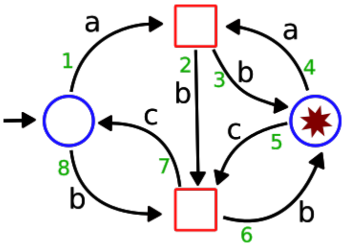
$\mathit{Output} = abc$

$$\mathcal{F} = \{\,\{a, c\},\ \{a, b, c\}\,\}.$$

General Memory Structures

- Set of states M.
- $\mu\colon M \times E \to M$, update function.
- $\texttt{next-move}\colon M \times V_{\mathrm{Eve}} \to E$, gives a strategy.



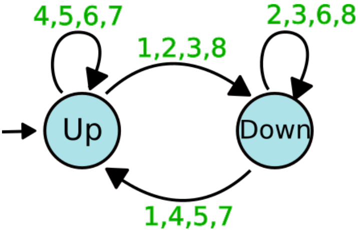


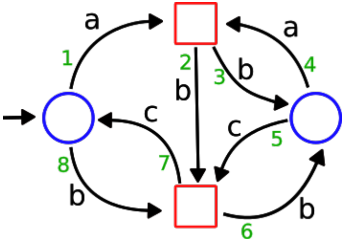
$\mathit{Output} = abcb\ldots$

$$\mathcal{F} = \{\,\{a, c\},\ \{a, b, c\}\,\}.$$

General Memory Structures

- Set of states M.
- $\mu\colon M \times E \to M$, **update function.**
- $\texttt{next-move}\colon M \times V_{\mathrm{Eve}} \to E$, gives a strategy.

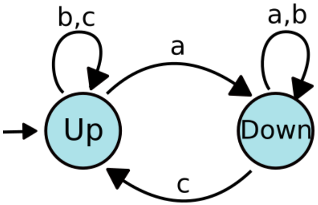


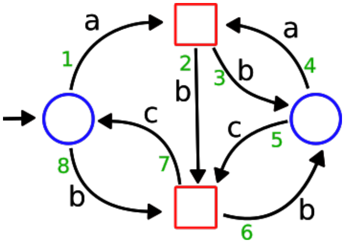


$$\mathcal{F} = \{\, \{a, c\},\ \{a, b, c\}\, \}.$$

Chromatic Memory Structures

- Set of states M.
- $\mu\colon M \times \mathcal{C} \to M$, **update function.**
- `next-move`: $M \times V_{\mathrm{Eve}} \to E$, gives a strategy.



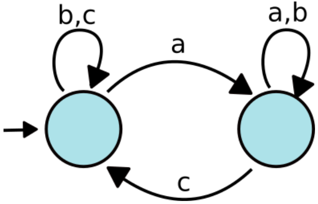


$$\mathcal{F} = \{\,\{a, c\},\ \{a, b, c\}\,\}.$$

Arena-Independent Memory Structures

We fix a Muller condition $\mathcal{F}$.

- Set of states M.
- $\mu\colon M \times \mathcal{C} \to M$, update function.



Memory structure $\mathcal{M} = (M, m_0, \mu)$.

The memory $\mathcal{M}$ is **arena-independent** if for every $\mathcal{F}$-game $\mathcal{G}$ won by Eve, there is a winning strategy given by some function

$$\texttt{next-move}_{\mathcal{G}}\colon M \times V_{\mathrm{Eve}} \to E.$$

Muller games are finite-memory determined

Theorem (Gurevich, Harrington '82)

Every Muller condition $\mathcal{F}$ admits a finite arena-independent memory structure.

➔ A deterministic parity (or Rabin) automaton recognizing the Muller condition gives an arena-independent memory.

We fix a set of colours $\mathcal{C}$ and a Muller condition $\mathcal{F} \subseteq 2^{\mathcal{C}}$.

General Memory Requirements $= \mathfrak{mem}_{gen}(\mathcal{F})$	Minimal n such that for any $\mathcal{F}$-game won by Eve, she can win it using a general memory of size n.
Chromatic Memory Requirements $= \mathfrak{mem}_{chr}(\mathcal{F})$	Minimal n such that for any $\mathcal{F}$-game won by Eve, she can win it using a chromatic memory of size n.
Arena-Independent Memory Requirements $= \mathfrak{mem}_{ind}(\mathcal{F})$	Minimal size of an arena-independent memory for $\mathcal{F}$.

General Memory Requirements $= \mathfrak{mem}_{gen}(\mathcal{F})$	Minimal n such that for any $\mathcal{F}$-game won by Eve, she can win it using a general memory of size n.
Chromatic Memory Requirements $= \mathfrak{mem}_{chr}(\mathcal{F})$	Minimal n such that for any $\mathcal{F}$-game won by Eve, she can win it using a chromatic memory of size n.
Arena-Independent Memory Requirements $= \mathfrak{mem}_{ind}(\mathcal{F})$	Minimal size of an arena-independent memory for $\mathcal{F}$.

$$\mathfrak{mem}_{gen}(\mathcal{F}) \leq \mathfrak{mem}_{chr}(\mathcal{F}) \leq \mathfrak{mem}_{ind}(\mathcal{F})$$

General Memory Requirements $= \mathfrak{mem}_{gen}(\mathcal{F})$	Minimal n such that for any $\mathcal{F}$-game won by Eve, she can win it using a general memory of size n.
Chromatic Memory Requirements $= \mathfrak{mem}_{chr}(\mathcal{F})$	Minimal n such that for any $\mathcal{F}$-game won by Eve, she can win it using a chromatic memory of size n.
Arena-Independent Memory Requirements $= \mathfrak{mem}_{ind}(\mathcal{F})$	Minimal size of an arena-independent memory for $\mathcal{F}$.

$$\mathfrak{mem}_{gen}(\mathcal{F}) \overset{C.\text{'}21}{<} \mathfrak{mem}_{chr}(\mathcal{F}) \overset{Kopczynski\text{'}08}{=} \mathfrak{mem}_{ind}(\mathcal{F})$$

Contributions

$\mathcal{C}$ set of colours, $\mathcal{F} \subseteq 2^{\mathcal{C}}$.

$\mathfrak{mem}_{chr}(\mathcal{F}) = \mathfrak{mem}_{ind}(\mathcal{F})$ = Size of a minimal **deterministic Rabin automaton** recognizing $\mathcal{F}$.

$\mathfrak{mem}_{gen}(\mathcal{F})$ = Size of a minimal **Good-For-Games Rabin automaton** recognizing $\mathcal{F}$.

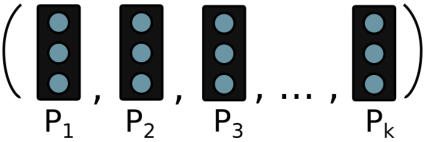
➔ The gap between $\mathfrak{mem}_{gen}(\mathcal{F})$ and $\mathfrak{mem}_{chr}(\mathcal{F})$ can be exponential in $|\mathcal{C}|$.

Index

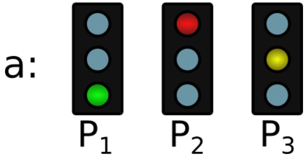
Rabin Conditions

Class of Muller conditions $\mathcal{F} \subseteq 2^{\mathcal{C}}$ defined by:

A set of Rabin pairs:



Each colour in $\mathcal{C}$ triggers one light for each Rabin pair P_i:



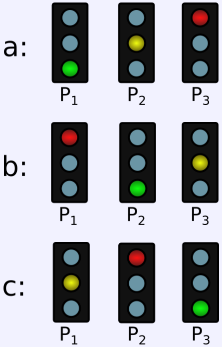
We accept a word $w \in \mathcal{C}^\omega$ if some P_i produces infinitely often green and only finitely many times red.

Rabin Conditions

Example

“See at most two colours infinitely often” is a Rabin condition:

$$\mathcal{F} = \{\, \{a\}, \{b\}, \{c\}\, \{a, b\}, \{a, c\}, \{b, c\}\, \}.$$



Rabin Conditions

Theorem (Klarlund'94, Zielonka'98)

Rabin conditions are exactly the family of Muller conditions that are ***positionally determined***[1] *(that is, if Eve wins a Rabin game, she can win it using a memoryless strategy).*

[1] "Positional" will mean "positional from the point of view of Eve" in this talk. This is sometimes called "half-positional".

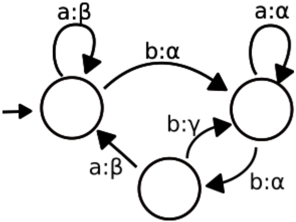
Rabin Automata

Input alphabet
$\mathcal{C} = \{a, b\}$.

Output alphabet
$\mathcal{C}' = \{\alpha, \beta, \gamma\}$.

Acceptance condition ➔ Rabin condition over $\mathcal{C}'$.

$\mathcal{A} =$



RABIN CONDITION OVER TRANSITIONS!

The automaton $\mathcal{A}$ *recognizes a Muller condition* $\mathcal{F}$ over $\mathcal{C}$ if

$$\mathcal{L}(\mathcal{A}) = \{w \in \mathcal{C}^\omega \ : \ Inf(w) \in \mathcal{F}\}.$$

Rabin Automata are Arena-Independent Memories

Proposition (Folklore)

If $\mathcal{A}$ is a deterministic Rabin automata recognizing a Muller condition $\mathcal{F}$, then $\mathcal{A}$ ***is an arena-independent memory*** *for $\mathcal{F}$.*

➔ We just have to show how to define a `next-move` function.

Rabin Automata are Arena-Independent Memories

$\mathcal{G}$ → $\mathcal{F}$-game

$\mathcal{A}$ → Automaton recognizing $\mathcal{F}$, Acceptance condition $\mathcal{F}'$.

} Product game $\mathcal{G} \ltimes \mathcal{A}$ Winning condition $\mathcal{F}'$.

Lemma (Folklore)

If $\mathcal{A}$ is deterministic,

$$\text{Eve wins } \mathcal{G} \quad \Leftrightarrow \quad \text{Eve wins } \mathcal{G} \ltimes \mathcal{A}.$$

Rabin Automata are Arena-Independent Memories

$\mathcal{G}$ →	$\mathcal{F}$-game	}	Product game $\mathcal{G} \ltimes \mathcal{A}$
$\mathcal{A}$ →	Automaton recognizing $\mathcal{F}$, Acceptance condition $\mathcal{F}'$.		Winning condition $\mathcal{F}'$.

Lemma (Folklore)

If $\mathcal{A}$ is deterministic,

$$\text{Eve wins } \mathcal{G} \quad \Leftrightarrow \quad \text{Eve wins } \mathcal{G} \ltimes \mathcal{A}.$$

We can transform a positional strategy in $\mathcal{G} \ltimes \mathcal{A}$ into a `next-move` function

$$\texttt{next-move}_{\mathcal{G}} : \mathcal{A} \times V_{\mathrm{Eve}} \to E.$$

Arena-Independent Memories are Deterministic Rabin Automata

Theorem (C. '21)

If $\mathcal{M}$ is an arena-independent memory for $\mathcal{F}$, then we can define a Rabin condition on top of it so that it becomes a deterministic Rabin automaton recognizing $\mathcal{F}$.

Corollary

$\mathfrak{mem}_{chr}(\mathcal{F})$ = *Size of a minimal deterministic Rabin automaton for $\mathcal{F}$.*

Corollary

Determining $\mathfrak{mem}_{chr}(\mathcal{F})$ is NP*-complete, even if the condition $\mathcal{F}$ is represented "quite explicitly".*

Index

Good-For-Games Automata

$\mathcal{G}$ ➔ $\mathcal{F}$-game

$\mathcal{A}$ ➔ **Non-Deteterministic** Automaton for $\mathcal{F}$.

} $\mathcal{G} \ltimes \mathcal{A}$, but **the winner is not necessarily preserved!**

Good-For-Games Automata

$\mathcal{G}$ → $\mathcal{F}$-game

$\mathcal{A}$ → **Non-Deteterministic** Automaton for $\mathcal{F}$.

} $\mathcal{G} \ltimes \mathcal{A}$, but **the winner is not necessarily preserved!**

Definition

$\mathcal{A}$ is *Good-For-Games* (GFG) if for every $\mathcal{F}$-game $\mathcal{G}$,

$$\text{Eve wins } \mathcal{G} \quad \Leftrightarrow \quad \text{Eve wins } \mathcal{G} \ltimes \mathcal{A}.$$

(Remark: in $\mathcal{G} \ltimes \mathcal{A}$ Eve chooses the transitions in $\mathcal{A}$.)

Good-For-Games Automata

$\mathcal{G}$ →	$\mathcal{F}$-game	$\mathcal{G} \ltimes \mathcal{A}$, but **the winner is not necessarily preserved!**
$\mathcal{A}$ →	**Non-Deteterministic** Automaton for $\mathcal{F}$.	

Definition

$\mathcal{A}$ is *Good-For-Games* (GFG) if for every $\mathcal{F}$-game $\mathcal{G}$,

$$\text{Eve wins } \mathcal{G} \quad \Leftrightarrow \quad \text{Eve wins } \mathcal{G} \ltimes \mathcal{A}.$$

(Remark: in $\mathcal{G} \ltimes \mathcal{A}$ Eve chooses the transitions in $\mathcal{A}$.)

Facts

- Deterministic automata are Good-For-Games.
- There are Good-For-Games automata that are not deterministic.

Good-For-Games Automata as Memory Structures

$\mathcal{A}$ a GFG-Rabin automaton for $\mathcal{F}$.

In $\mathcal{G} \ltimes \mathcal{A}$ Eve has a positional strategy. We transform this strategy into a `next-move` function

$$\texttt{next-move}_{\mathcal{G}} \colon \mathcal{A} \times V_{\mathrm{Eve}} \to E.$$

So $\mathcal{A}$ provides a(n) ? memory structure.

Good-For-Games Automata as Memory Structures

$\mathcal{A}$ a GFG-Rabin automaton for $\mathcal{F}$.

In $\mathcal{G} \ltimes \mathcal{A}$ Eve has a positional strategy. We transform this strategy into a `next-move` function

$$\texttt{next-move}_{\mathcal{G}} \colon \mathcal{A} \times V_{\mathrm{Eve}} \to E.$$

So $\mathcal{A}$ provides a **general** memory structure.

The automaton is Non-Det, so we can take different transitions in $\mathcal{A}$ when visiting different edges in $\mathcal{G}$ (even if they have the same colour).

Good-For-Games Automata as Memory Structures

$\mathcal{A}$ a GFG-Rabin automaton for $\mathcal{F}$.

In $\mathcal{G} \ltimes \mathcal{A}$ Eve has a positional strategy. We transform this strategy into a `next-move` function

$$\texttt{next-move}_{\mathcal{G}} \colon \mathcal{A} \times V_{\mathrm{Eve}} \to E.$$

Conclusion

$\mathfrak{mem}_{gen}(\mathcal{F}) \ \leq\ $ Size of a minimal GFG Rabin automaton for $\mathcal{F}$.

Good-For-Games Automata Give Optimal Memory Structures

[DJW '97[2]] → Characterization of $\mathfrak{mem}_{gen}(\mathcal{F})$ in terms of the *Zielonka tree* of $\mathcal{F}$.

Using the Zielonka tree we give a construction of a GFG-Rabin automaton of size $\mathfrak{mem}_{gen}(\mathcal{F})$ recognizing $\mathcal{F}$.

[2] S. Dziembowski, M. Jurdziński and I. Walukiewicz, *How much memory is needed to win infinite games?*, LICS 1997.

Good-For-Games Automata Give Optimal Memory Structures

[DJW '97[2]] ➔ Characterization of $\mathfrak{mem}_{gen}(\mathcal{F})$ in terms of the *Zielonka tree* of $\mathcal{F}$.

Using the Zielonka tree we give a construction of a GFG-Rabin automaton of size $\mathfrak{mem}_{gen}(\mathcal{F})$ recognizing $\mathcal{F}$.

Corollary (C., Colcombet, Lehtinen '21)

$$\mathfrak{mem}_{gen}(\mathcal{F}) = \textit{ Size of a minimal GFG Rabin automaton for } \mathcal{F}.$$

And we can construct one such minimal GFG Rabin automaton "efficiently"!

[2] S. Dziembowski, M. Jurdziński and I. Walukiewicz, *How much memory is needed to win infinite games?*, LICS 1997.

General Memory vs Chromatic Memory

Theorem (C., Colcombet, Lehtinen '21[3])

There exists a constant $c > 1$ and a sequence of Muller conditions $\mathcal{F}_1, \dots, \mathcal{F}_n \dots$ over $\mathcal{C}_1, \dots, \mathcal{C}_n \dots$, $|\mathcal{C}_n| = n$ such that:

- $\mathfrak{mem}_{gen}(\mathcal{F}_n) = n/2$.
- $\mathfrak{mem}_{chr}(\mathcal{F}_n) = c^n$.

Corollary (C., Colcombet, Lehtinen '21)

The gap on the size between deterministic and GFG Rabin automata recognizing Muller conditions is exponential.

[3] Thanks to Marthe Bonamy and Pierre Fraigniaud for their help with graph theory!

General Memory vs Chromatic Memory

Theorem (C., Colcombet, Lehtinen '21[3])

There exists a constant $c > 1$ and a sequence of Muller conditions $\mathcal{F}_1, \dots, \mathcal{F}_n \dots$ over $\mathcal{C}_1, \dots, \mathcal{C}_n \dots$, $|\mathcal{C}_n| = n$ such that:

- $\mathfrak{mem}_{gen}(\mathcal{F}_n) = n/2$.
- $\mathfrak{mem}_{chr}(\mathcal{F}_n) = c^n$.

Corollary (C., Colcombet, Lehtinen '21)

The gap on the size between deterministic and GFG Rabin automata recognizing Muller conditions is exponential.

Thank you!

[3] Thanks to Marthe Bonamy and Pierre Fraigniaud for their help with graph theory!