On the Minimisation of Transition-Based Rabin Automata and the Chromatic Memory Requirements of Muller Conditions

Antonio Casares

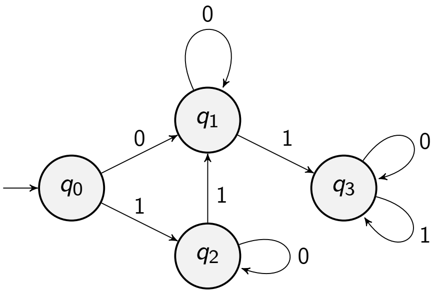
LaBRI

6 July 2021

Index

Automata over infinite words

All automata in this talk will be **deterministic**.

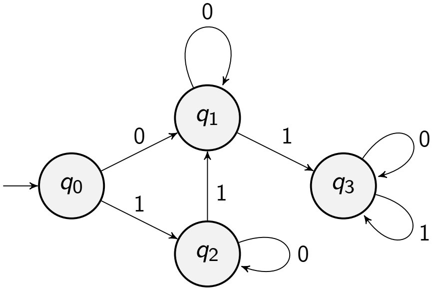


Automaton $\mathcal{A}$, $\Sigma = \{0, 1\}$.

$\textit{Input} = 10101000010 \cdots \in \Sigma^\omega \quad \longrightarrow \quad$ Infinite run over the automaton.

Automata over infinite words

All automata in this talk will be **deterministic**.

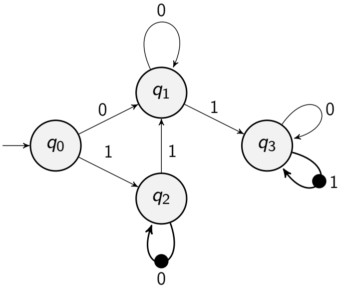


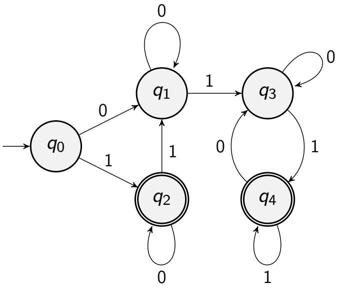
Automaton $\mathcal{A}$, $\Sigma = \{0, 1\}$.

Input $= 10101000010 \cdots \in \Sigma^\omega \quad \longrightarrow$ Infinite run over the automaton.

We have to define which runs will be accepting.

Büchi conditions

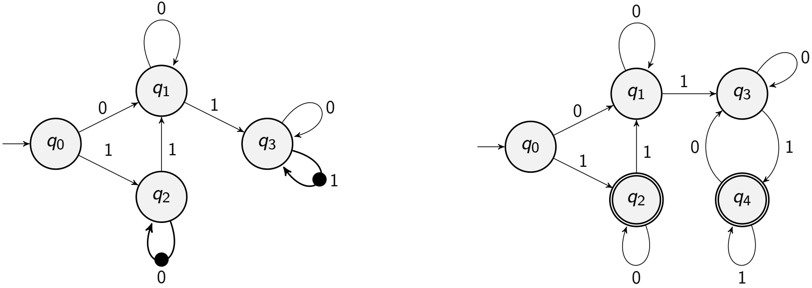




- Condition over transitions.
- Condition over states.

We accept a run if it visits infinitely often a Büchi transition (resp. Büchi state).

Büchi conditions: transition-based vs state-based



- Both models are equivalent (linear blow-up on size in both directions).
- Transition-based automata are always smaller.
- Minimality is not preserved.
- **The minimisation problem is not clear to be equivalent.**

Theorem (Schewe, '10)

Minimisation of state-based Büchi automata is NP*-complete.*

➔ The reduction of the proof (strongly) relies on the state-based assumption!

Minimisation of Büchi automata

Theorem (Schewe, '10)

Minimisation of state-based Büchi automata is NP*-complete.*

➔ The reduction of the proof (strongly) relies on the state-based assumption!

Corollary

Minimisation of state-based co-Büchi, parity and Rabin automata is NP*-complete.*

Minimisation of GFG-automata

Good-For-Games (GFG): class of automata between deterministic and non-deterministic.

Theorem (Abu Radi, Kupferman, '19)

We can minimise transition-based co-Büchi GFG-automata in polynomial time, and there is a canonical minimal automaton.

Minimisation of GFG-automata

Good-For-Games (GFG): class of automata between deterministic and non-deterministic.

Theorem (Abu Radi, Kupferman, '19)

We can minimise transition-based co-Büchi GFG-automata in polynomial time, and there is a canonical minimal automaton.

What makes it possible, GFGness, transition-based, a combination of both...?

Minimisation of GFG-automata

Good-For-Games (GFG): class of automata between deterministic and non-deterministic.

Theorem (Abu Radi, Kupferman, '19)

We can minimise transition-based co-Büchi GFG-automata in polynomial time, and there is a canonical minimal automaton.

What makes it possible, GFGness, transition-based, a combination of both...?

Theorem (Schewe, '20)

Minimisation of Büchi and co-Büchi state-based GFG-automata is NP*-complete.*

Minimisation of co-Büchi automata:

	State-based	**Transition-based**
Deterministic	NP-complete	?
Good-For-Games	NP-complete	Polynomial

Remark: For deterministic automata, minimisation of Büchi and co-Büchi automata is equivalent.

Main result

Theorem

Minimisation of transition-based Rabin automata is NP*-complete.*

Rabin conditions (defined soon) are more general and expressive than Büchi ones.

Interest of Rabin automata

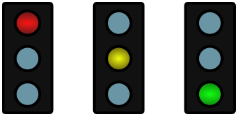
- The results presented before cast doubts on the *NP*-completeness of the minimisation of transition-based Rabin automata. Büchi automata are a special type of Rabin automata.

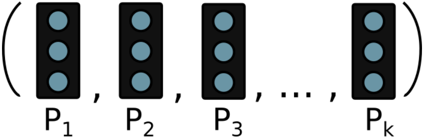
- The determinisation of Büchi automata by Safra's construction provides a Rabin automaton.

- Rabin conditions are exactly the Muller conditions that are half-positional determined (if the existencial player wins a Rabin game, she can always use a positional strategy).

Rabin conditions

Set of Rabin pairs, that can take the values:





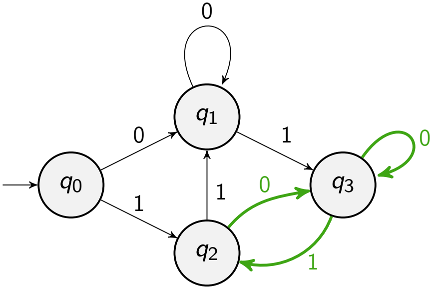
Each transition of the automaton triggers one colour for each Rabin pair P_i (green, orange or red).

We accept a run if some P_i produces infinitely often green and only finitely many times red.

Remark: Büchi = Rabin with just one pair and not using the colour red.

Cycle (or Muller) conditions

A *cycle* of an automaton $\mathcal{A}$ is a set of transitions that forms a closed path (not necessarily simple).



- A run over an automaton eventually gets trapped in one cycle.

A *cycle condition* over $\mathcal{A}$ is a map

$$f : \mathcal{C}ycles(\mathcal{A}) \to \{Accept, Reject\}.$$

Cycle conditions

Remark: each Rabin (or Büchi) condition induces a cycle condition, but the latter are more general.

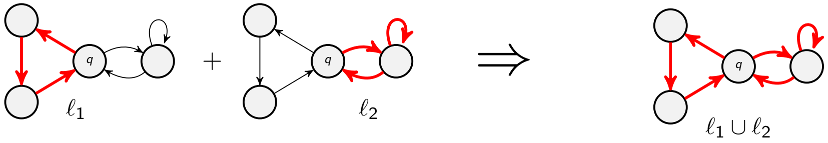
Question

Given an automaton with a cycle condition, when can we replace the condition by a Rabin one?

Proposition (C., Colcombet, Fijalkow '20)

Given a cycle automaton $\mathcal{A}$, the following are equivalent:

- *We can define a Rabin condition on top of $\mathcal{A}$, obtaining an equivalent automaton.*
- *For every pair of cycles ℓ_1, ℓ_2 with some state in common, if both ℓ_1 and ℓ_2 are rejecting, their union is also a rejecting cycle.*



Index

The minimisation problem

For the rest of this talk,

automata = transition-based deterministic automata.

Minimisation of Rabin automata (decision problem):

Input: A Rabin automaton $\mathcal{A}$ and an integer k.

Question: Is there a Rabin automaton with k states recognizing $\mathcal{L}(\mathcal{A})$?

Theorem

This decision problem is NP*-complete.*

Minimisation of Rabin automata is NP-complete

Lemma

Minimisation of Rabin automata is in NP.

Proof.

Testing equivalence of Rabin automata can be done in polynomial time. Therefore, we can guess an equivalent Rabin automaton of size k, and check its equivalence to the given one. □

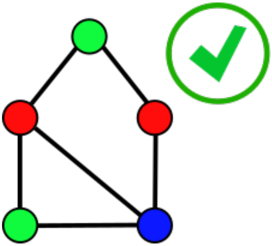
The chromatic number of a graph

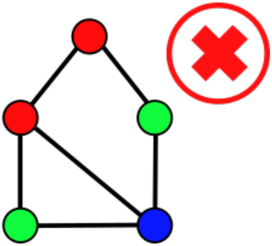
Let $G = (V, E)$ be an undirected graph.

A *colouring* of size k of G is a function

$$c : V \to C, \quad |C| = k,$$

verifying that two vertices connected by an edge have different colours.





The chromatic number of a graph

The *chromatic number* of G, $\chi(G)$, is the minimal number of colours needed to colour G.

The chromatic number problem (decision problem):

Input: A simple undirected graph G and an integer k.

Question: Is there a colouring of G using k colours?

Lemma (Karp, '72)

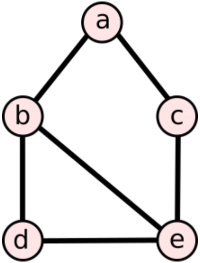
The chromatic number problem is NP*-complete.*

NP-hardness

Let $G = (V, E)$ be a graph. We define the language over the alphabet V:

$$L_G = \bigcup_{(v,u)\in E} V^*(v^+u^+)^\omega$$

A word $w \in V^\omega$ is in L_G iff the set of vertices appearing infinitely often contains exactly 2 vertices connected by an edge.



- $aebcabab(ab)^\omega$ ✓
- $adbaeae(ae)^\omega$ ✗
- $adbaaaa(a)^\omega$ ✗
- $abd(abd)^\omega$ ✗

NP-hardness

Let $G = (V, E)$ be a graph. We define the language over the alphabet V:

$$L_G = \bigcup_{(v,u) \in E} V^*(v^+u^+)^\omega$$

A word $w \in V^\omega$ is in L_G iff the set of vertices appearing infinitely often contains exactly 2 vertices connected by an edge.

Remark

The language L_G verifies that the acceptance of a word $w \in V^\omega$ ***only depends on the set of letters appearing infinitely often*** *on it. (Is what we call a Muller condition).*

NP-hardness

$$L_G = \bigcup_{(v,u)\in E} V^*(v^+u^+)^\omega$$

Proposition

The graph G provides a natural Rabin automaton for L_G (polynomial-time reduction).

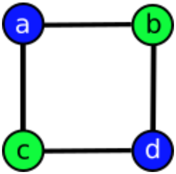
Proposition

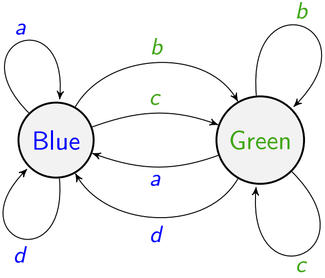
The size of a minimal Rabin automaton recognising L_G coincides with the chromatic number of G.

Colouring of size k ⇢ Rabin automaton with k states:

Let $c : V \to [1, k]$ be a colouring of G. We define an automaton $\mathcal{A}$ as:

- Set of states $= \{1, \dots, k\}$.
- If we read a letter v, we go to $c(v)$.



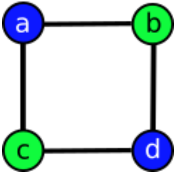


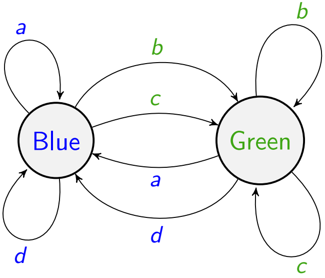
Colouring of size k ⇢ Rabin automaton with k states:

We have to be able to put a Rabin condition accepting

$$L_G = \bigcup_{(v,u)\in E} V^*(v^+u^+)^\omega.$$

We first define a cycle condition:



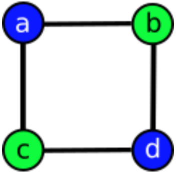


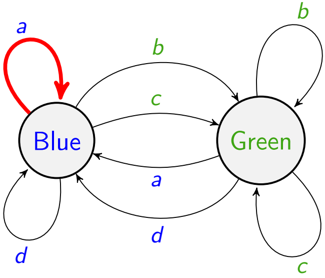
Colouring of size k --→ Rabin automaton with k states:

We have to be able to put a Rabin condition accepting

$$L_G = \bigcup_{(v,u)\in E} V^*(v^+u^+)^\omega.$$

We first define a cycle condition:



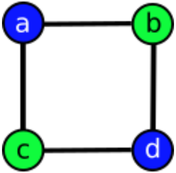


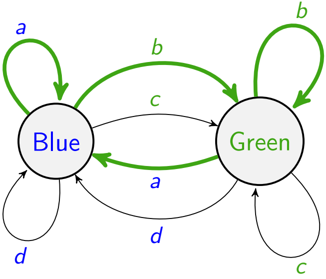
Colouring of size k $\dashrightarrow$ Rabin automaton with k states:

We have to be able to put a Rabin condition accepting

$$L_G = \bigcup_{(v,u)\in E} V^*(v^+u^+)^\omega.$$

We first define a cycle condition:



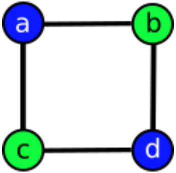


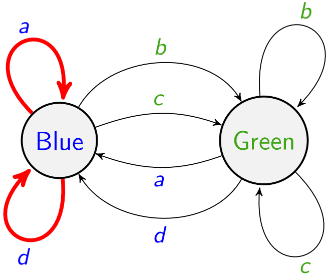
Colouring of size k ⇢ Rabin automaton with k states:

We have to be able to put a Rabin condition accepting

$$L_G = \bigcup_{(v,u)\in E} V^*(v^+u^+)^\omega.$$

We first define a cycle condition:



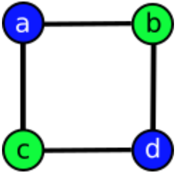


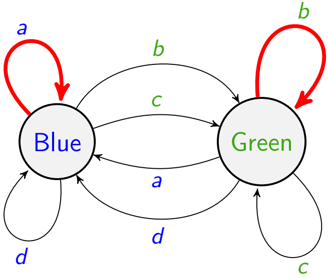
Colouring of size k ⇢ Rabin automaton with k states:

We have to be able to put a Rabin condition accepting

$$L_G = \bigcup_{(v,u)\in E} V^*(v^+u^+)^\omega .$$

We check that the union of two rejecting cycles is rejecting: we cannot form an accepting cycle from two rejecting ones because cycles corresponding to vertices connected by some edge are in different states.





Colouring of size k ⇢ Rabin automaton with k states:

Conclusion: We can put a Rabin condition on top of that automaton and therefore:

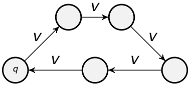
$$\text{Size of a minimal Rabin automaton} \leq \chi(G).$$

Rabin automaton $\mathcal{A}$ ⇢ Colouring of size $|\mathcal{A}|$:

Let $\mathcal{A}$ be a Rabin automaton for L_G with set of states Q.

For each $v \in V$ consider:

$$Q_v = \{q \in Q \ : \ \text{a cycle labelled only with } v \text{ passes throught } q\}.$$



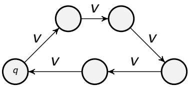
(Q_v non-empty).

Rabin automaton $\mathcal{A} \dashrightarrow$ Colouring of size $|\mathcal{A}|$:

Let $\mathcal{A}$ be a Rabin automaton for L_G with set of states Q.

For each $v \in V$ consider:

$$Q_v = \{q \in Q \ : \ \text{a cycle labelled only with } v \text{ passes throught } q\}.$$



(Q_v non-empty).

For each $v \in V$ we pick $q_v \in Q_v$, and define the colouring

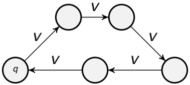
$$\begin{aligned} c \colon V &\to Q \\ v &\mapsto q_v. \end{aligned}$$

Rabin automaton $\mathcal{A} \dashrightarrow$ Colouring of size $|\mathcal{A}|$:

Let $\mathcal{A}$ be a Rabin automaton for L_G with set of states Q.

For each $v \in V$ consider:

$$Q_v = \{q \in Q \ : \ \text{a cycle labelled only with } v \text{ passes throught } q\}.$$



(Q_v non-empty).

For each $v \in V$ we pick $q_v \in Q_v$, and define the colouring

$$\begin{aligned} c \colon V &\to Q \\ v &\mapsto q_v. \end{aligned}$$

Correct colouring $\Leftrightarrow$ We don't associate the same state to any pair of vertices connected by an edge $(v, u) \in E$.

Rabin automaton $\mathcal{A}$ ⇢ Colouring of size $|\mathcal{A}|$:

Proposition

Let $v, u \in V$ be two vertices connected by an edge, $(v, u) \in E$. Then

$$Q_v \cap Q_u = \emptyset.$$

Rabin automaton $\mathcal{A} \dashrightarrow$ Colouring of size $|\mathcal{A}|$:

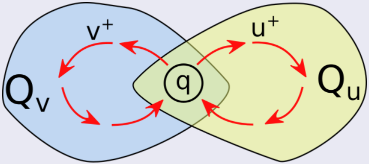
Proposition

Let $v, u \in V$ be two vertices connected by an edge, $(v, u) \in E$. Then

$$Q_v \cap Q_u = \emptyset.$$

Proof.

If $q \in Q_v \cap Q_u$, then:



But the union of these cycles would be accepting, what is impossible in a Rabin automaton.

□

Rabin automaton $\mathcal{A}$ $\dashrightarrow$ Colouring of size $|\mathcal{A}|$:

Conclusion: The mapping $c\colon V \to Q$ is a correct coloring, and therefore

$$\chi(G) \leq \text{Size of a minimal Rabin automaton.}$$

Minimisation of parity automata

Question

Can we extend this reduction to prove the NP-completeness of the minimisation of transition-based Büchi and parity automata?

Minimisation of parity automata

Question

Can we extend this reduction to prove the NP-completeness of the minimisation of transition-based Büchi and parity automata?

Not easily...

Minimisation of parity automata

➔ The language L_G used in the previous reduction was a Muller condition (acceptance of words only depend on the set of letters appearing infinitely often).

Minimisation of parity automata

➔ The language L_G used in the previous reduction was a Muller condition (acceptance of words only depend on the set of letters appearing infinitely often).

Proposition

We can minimise parity automata recognising Muller conditions in polynomial time.

Minimisation of parity automata

➔ The language L_G used in the previous reduction was a Muller condition (acceptance of words only depend on the set of letters appearing infinitely often).

Proposition

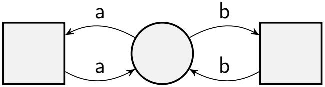
We can minimise parity automata recognising Muller conditions in polynomial time.

Proof idea:

- A minimal parity automaton recognising a Muller condition can be obtained from the Zielonka tree of the condition [C., Colcombet, Fijalkow, '20] and [Meyer, Sickert, '21].
- We can build the Zielonka tree in polynomial time from a given parity automaton.

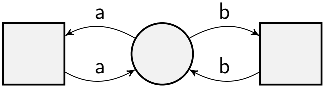
Index

Memories for games

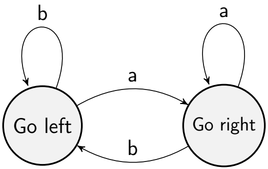


Winning condition = See *a* and *b* infinitely often.

Memories for games



Winning condition = See *a* and *b* infinitely often.



Memory structure for Eve.

Chromatic memories

- In general, transitions between states of the memory structure *depend on the edges of the considered game*.

Chromatic memories

- In general, transitions between states of the memory structure *depend on the edges of the considered game*.

- A memory structure is ***chromatic*** if its transitions only depend on the colours used to define a given condition.
- Given a fixed winning condition, $\mathbb{W}$, an ***arena-independent memory for*** $\mathbb{W}$ is a memory structure that can be used to win any game using this condition.

Chromatic memories

- In general, transitions between states of the memory structure *depend on the edges of the considered game*.

- A memory structure is ***chromatic*** if its transitions only depend on the colours used to define a given condition.
- Given a fixed winning condition, $\mathbb{W}$, an ***arena-independent*** *memory for* $\mathbb{W}$ is a memory structure that can be used to win any game using this condition.

Question (Kopczyński '06)

Given a game $\mathcal{G}$, is there always a minimal *chromatic* memory for it?

Can arena-independent memories be optimal?

Chromatic Memory Requirements for Muller Games

Proposition

- *There are Muller games for which chromatic memories have to be strictly bigger than non-chromatic ones.*
- *There are Muller conditions such that the memory required to win any game using that condition is strictly less than the size of an arena-independent memory.*

Chromatic Memories and Rabin Automata

Theorem

For a given Muller condition, the following quantities coincide:

- *The size of a minimal Rabin automaton recognising the condition.*
- *The size of a minimal arena-independent memory for this Muller condition.*
- *The least number n such that, in any game where Eve wins, she can use a chromatic memory of size n to set up a winning strategy.*

Moreover, determining this quantity is a NP*-complete problem.*

We prove it using the characterisation of Rabin conditions with cycles!

Theorem

For a given Muller condition, the following quantities coincide:

- *The size of a minimal Rabin automaton recognising the condition.*
- *The size of a minimal arena-independent memory for this Muller condition.*
- *The least number n such that, in any game where Eve wins, she can use a chromatic memory of size n to set up a winning strategy.*

Moreover, determining this quantity is a `NP`*-complete problem.*

We prove it using the characterisation of Rabin conditions with cycles!

Thank you!