Minimisation of (history-)deterministic generalised (co-)Büchi automata

joint work with Antonio Casares, Denis Kuperberg, Olivier Idir and Aditya Prakash

Minimisation of (history-)deterministic generalised (co-)Büchi automata

joint work with Antonio Casares, Denis Kuperberg, Olivier Idir and Aditya Prakash

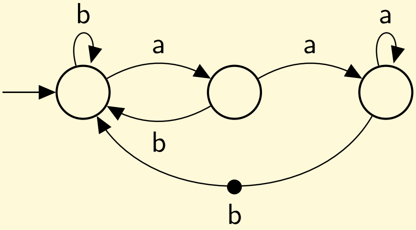



Minimisation of (history-)deterministic generalised (co-)Büchi automata

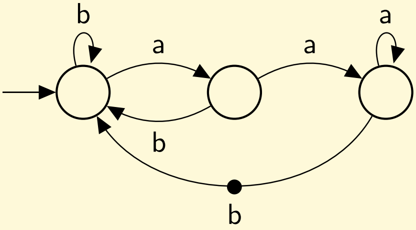
joint work with Antonio Casares, Denis Kuperberg, Olivier Idir and Aditya Prakash




Büchi automaton

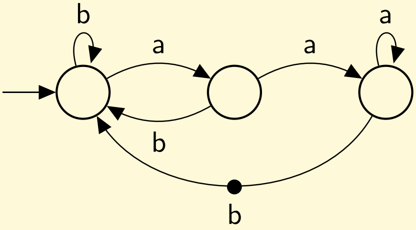


Büchi automaton



$L(\mathcal{A}) = \{\text{words that contain infinitely many } aab\} = (\Sigma^* aab)^\omega$

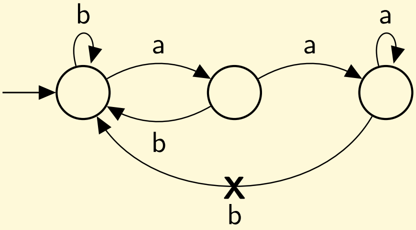
Büchi automaton



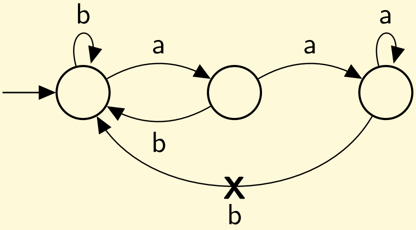
$L(\mathcal{A}) = \{\text{words that contain infinitely many } aab\} = (\Sigma^* aab)^\omega$

Deterministic Büchi automata recognise *Büchi languages*.

Co-Büchi automaton

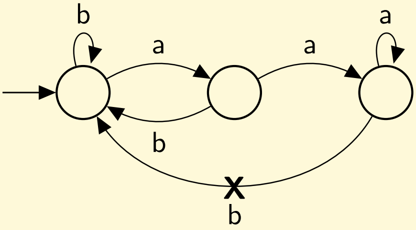


Co-Büchi automaton



$L(\mathcal{A}) = \{\text{words that contain finitely many } aab\} = \Sigma^* a^\omega \cup \Sigma^*(ab^+)^\omega$

Co-Büchi automaton



$L(\mathcal{A}) = \{\text{words that contain finitely many } aab\} = \Sigma^* a^\omega \cup \Sigma^*(ab^+)^\omega$

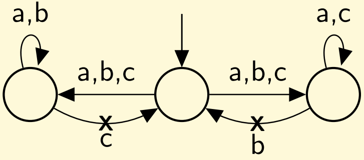
Deterministic co-Büchi automata recognise *co-Büchi languages*.
= complements of Büchi languages.

History-determinism [Henzinger, Piterman 2006]

$\mathcal{A}$ is history-deterministic if there is a *resolver* $\sigma : \Delta^* \times \Sigma \to \Delta$ such that for all $w \in L(\mathcal{A})$, applying σ while reading w yields an accepting run.

History-determinism [Henzinger, Piterman 2006]

$\mathcal{A}$ is history-deterministic if there is a *resolver* $\sigma : \Delta^* \times \Sigma \to \Delta$ such that for all $w \in L(\mathcal{A})$, applying σ while reading w yields an accepting run.

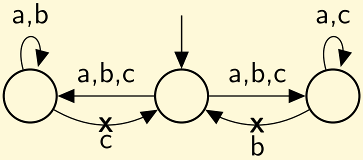


History-deterministic co-Büchi automaton.

Finitely many b or finitely many c.

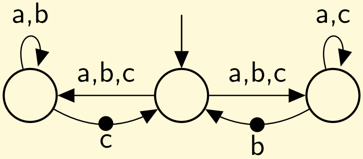
History-determinism [Henzinger, Piterman 2006]

$\mathcal{A}$ is history-deterministic if there is a *resolver* $\sigma : \Delta^* \times \Sigma \to \Delta$ such that for all $w \in L(\mathcal{A})$, applying σ while reading w yields an accepting run.



History-deterministic co-Büchi automaton.

Finitely many b or finitely many c.

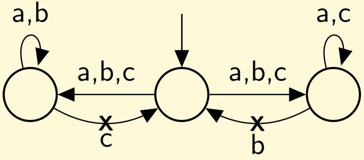


Büchi automaton,
NOT history-deterministic.

Infinitely many b or infinitely many c.

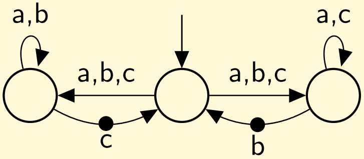
History-determinism [Henzinger, Piterman 2006]

$\mathcal{A}$ is history-deterministic if there is a *resolver* $\sigma : \Delta^* \times \Sigma \to \Delta$ such that for all $w \in L(\mathcal{A})$, applying σ while reading w yields an accepting run.



History-deterministic co-Büchi automaton.

Finitely many b or finitely many c.



Büchi automaton,
NOT history-deterministic.

Infinitely many b or infinitely many c.

[Kuperberg, Skrzypczak 2015]

History-deterministism can be tested in PTIME for Büchi and co-Büchi automata.

Minimisation

Minimise $\mathcal{A}$ = find $\mathcal{B}$ of the same type, the same language and with as few states as possible.

Minimisation

Minimise $\mathcal{A}$ = find $\mathcal{B}$ of the same type, the same language and with as few states as possible.

Remark : Minimising deterministic Büchi and co-Büchi automata is the same problem.

Minimisation

Minimise $\mathcal{A}$ = find $\mathcal{B}$ of the same type, the same language and with as few states as possible.

Remark : Minimising deterministic Büchi and co-Büchi automata is the same problem.

But history-deterministic Büchi and co-Büchi are very different!

[Kuperberg, Skrzypczak 2015]

- History-deterministic co-Büchi automata can be exponentially smaller than deterministic ones,
- Every history-deterministic Büchi automata has an equivalent deterministic one of size $O(n^2)$.

[Schewe 2010]

It is NP-complete to minimise deterministic (co-)Büchi automata.

[Schewe 2010]

It is NP-complete to minimise deterministic (co-)Büchi automata when the acceptance condition is on the states.

[Schewe 2010]

It is NP-complete to minimise deterministic (co-)Büchi automata when the acceptance condition is on the states.

[Abu Radi, Kupferman 2019]

We can minimise history-deterministic co-Büchi automata in polynomial time when the acceptance condition is on the transitions.

[Schewe 2010]

It is NP-complete to minimise deterministic (co-)Büchi automata when the acceptance condition is on the states.

[Schewe 2020]

It is NP-complete to minimise history-deterministic (co-)Büchi automata when the acceptance condition is on the states.

[Abu Radi, Kupferman 2019]

We can minimise history-deterministic co-Büchi automata in polynomial time when the acceptance condition is on the transitions.

State-based

[Schewe 2010]

It is NP-complete to minimise deterministic (co-)Büchi automata when the acceptance condition is on the states.

[Schewe 2020]

It is NP-complete to minimise history-deterministic (co-)Büchi automata when the acceptance condition is on the states.

Transition-based

[Abu Radi, Kupferman 2019]

We can minimise history-deterministic co-Büchi automata in polynomial time when the acceptance condition is on the transitions.

State-based

[Schewe 2010]

It is NP-complete to minimise deterministic (co-)Büchi automata when the acceptance condition is on the states.

[Schewe 2020]

It is NP-complete to minimise history-deterministic (co-)Büchi automata when the acceptance condition is on the states.

Transition-based

[Abu Radi, Kupferman 2019]

We can minimise history-deterministic co-Büchi automata in polynomial time when the acceptance condition is on the transitions.

Can we minimise (history-)deterministic (co-)Büchi automata in polynomial time?

In the transition-based world

	Büchi	Co-Büchi
Deterministic	???	???
History-deterministic	???	PTIME

In the transition-based world

	Büchi	Co-Büchi
Deterministic	???	???
History-deterministic	???	PTIME

When you encounter a difficult problem, switch to a different (related) problem.

-George Pólya (sort of)

In the transition-based world

	Büchi	Co-Büchi
Deterministic	???	???
History-deterministic	???	PTIME

When you encounter a difficult problem, switch to a different (related) problem.

-George Pólya (sort of)

This work: We study generalised (co-)Büchi automata:

In the transition-based world

	Büchi	Co-Büchi
Deterministic	???	???
History-deterministic	???	PTIME

When you encounter a difficult problem, switch to a different (related) problem.

-George Pólya (sort of)

This work: We study generalised (co-)Büchi automata:

	Generalised Büchi	Generalised co-Büchi
Deterministic	NP-complete	NP-complete
History-deterministic	NP-complete	PTIME

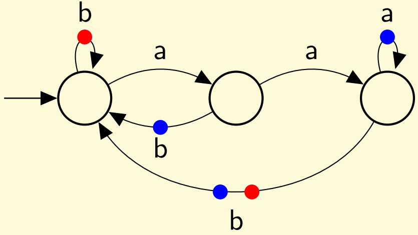
Generalised Büchi automaton

Set of colours $C = \{\bullet, \bullet, \bullet, \ldots\}$, colouring function $col : \Delta \to 2^C$

Generalised Büchi automaton

Set of colours $C = \{\textcolor{blue}{\bullet}, \textcolor{green}{\bullet}, \textcolor{red}{\bullet}, \dots\}$, colouring function $col : \Delta \to 2^C$

Generalised Büchi: See every colour infinitely many times.

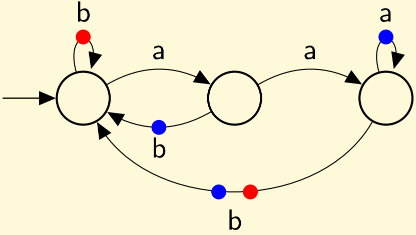


Infinitely many *aab* or infinitely many *ab* and b^2

Generalised Büchi automaton

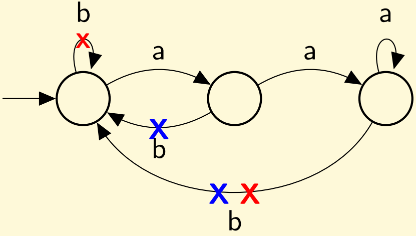
Set of colours $C = \{\textcolor{blue}{\bullet}, \textcolor{green}{\bullet}, \textcolor{red}{\bullet}, \ldots\}$, colouring function $col : \Delta \to 2^C$

Generalised Büchi: See every colour infinitely many times.



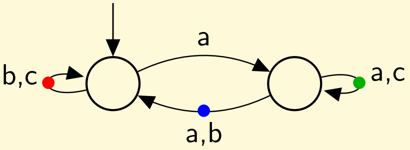
Infinitely many *aab* or infinitely many *ab* and b^2

Generalised co-Büchi: avoid some colour indefinitely after sone point.

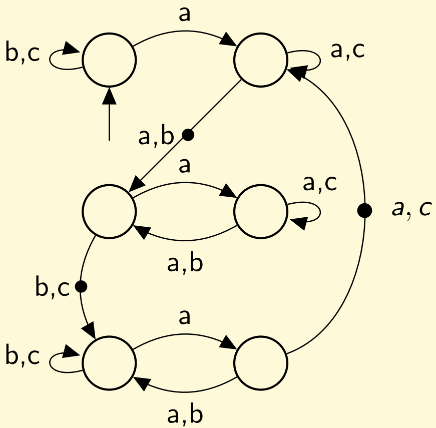


Finitely many *aab* and finitely many *ab* or b^2

From GBA to BA

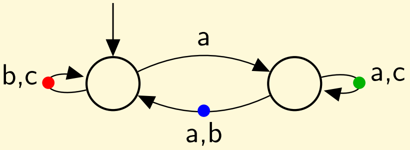


Generalised Büchi with n states and k colours.

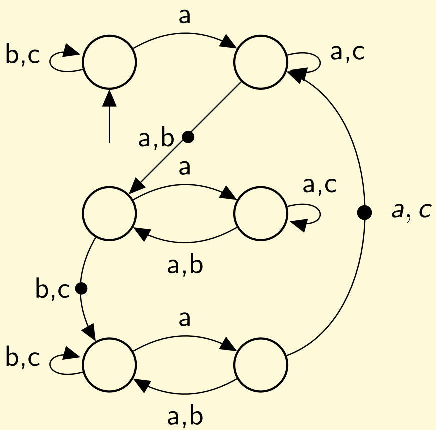


Büchi with nk states.

From GBA to BA



$\rightsquigarrow$



Generalised Büchi with n states and k colours.

Büchi with nk states.

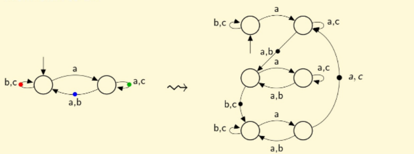
Preserves history-determinism!

Minimisation of gen. HD co-Büchi in polynomial time

Step 1: Apply Abu Radi-Kupferman

Given a generalised history-deterministic co-Büchi recognising L,

- Compute an equivalent history-deterministic co-Büchi



Step 1: Apply Abu Radi-Kupferman

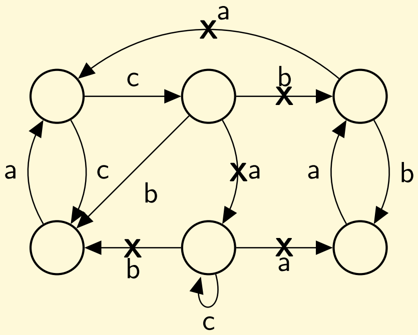
Given a generalised history-deterministic co-Büchi recognising L,

- Compute an equivalent history-deterministic co-Büchi

- We can minimise it $\rightarrow \mathcal{A}^{L}_{min}$ using Abu Radi and Kupferman's algorithm

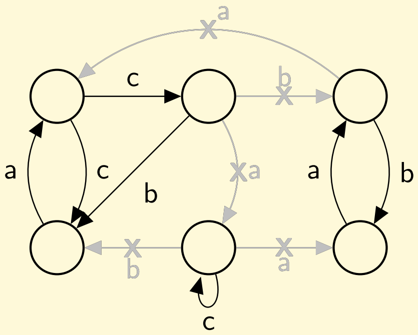
Step 2: Merge safe components

Suppose the language is *prefix-independent*, i.e., all states have the same residual.
(= the language is stable under prefix modification)



Step 2: Merge safe components

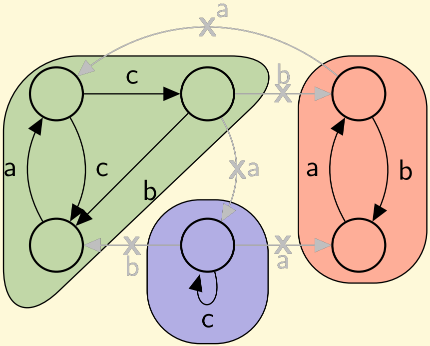
Suppose the language is *prefix-independent*



Accepted = stay in a safe component eventually

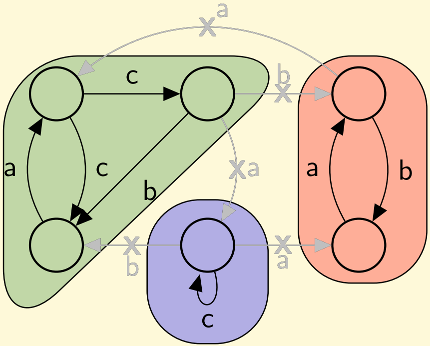
Step 2: Merge safe components

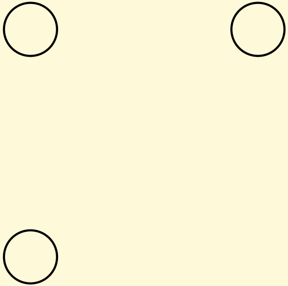
Suppose the language is *prefix-independent*



Step 2: Merge safe components

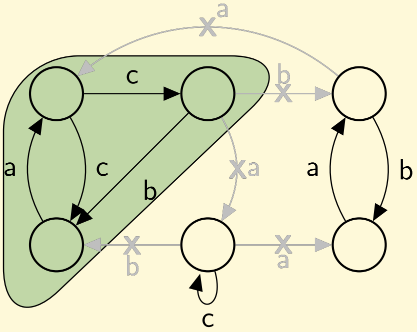
Suppose the language is *prefix-independent*

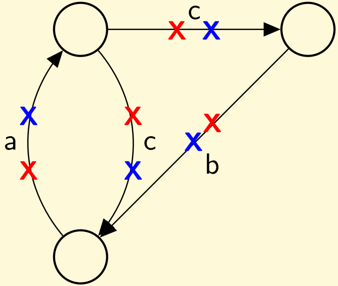




Step 2: Merge safe components

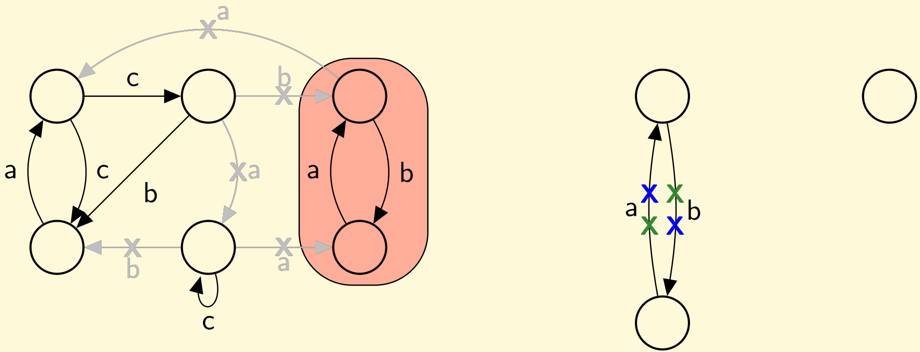
Suppose the language is *prefix-independent*





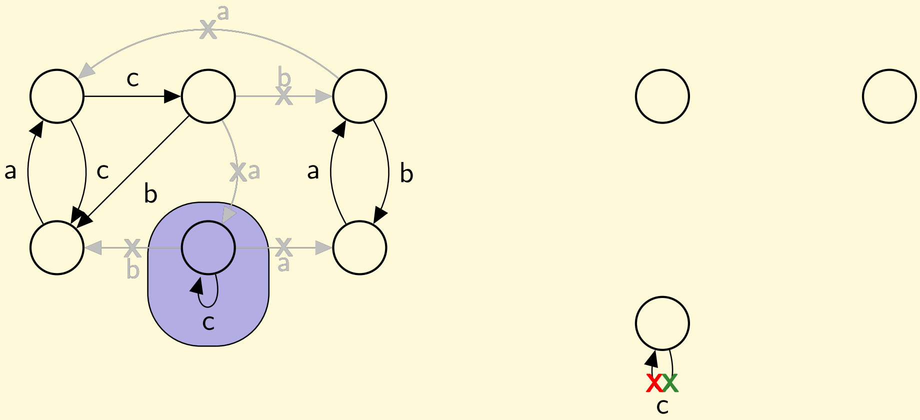
Step 2: Merge safe components

Suppose the language is *prefix-independent*



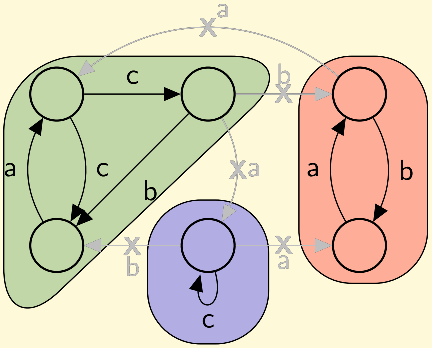
Step 2: Merge safe components

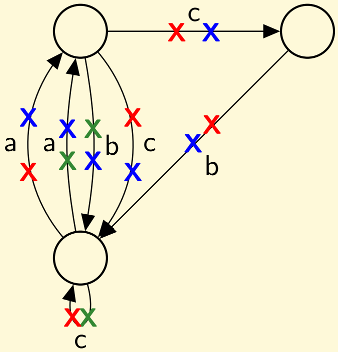
Suppose the language is *prefix-independent*



Step 2: Merge safe components

Suppose the language is *prefix-independent*





+ all other transitions, with X X X

Why it is minimal

[Abu Radi, Kupferman 2019]

For all equivalent HD co-Büchi automaton $\mathcal{C}$ there is an injection $\eta : \mathit{SafeComp}(\mathcal{A}^L_{min}) \to \mathit{SafeComp}(\mathcal{C})$ such that $|\eta(S)| \geq |S|$ for all $S \in \mathit{SafeComp}(\mathcal{C})$.

Why it is minimal

[Abu Radi, Kupferman 2019]

For all equivalent HD co-Büchi automaton $\mathcal{C}$ there is an injection $\eta : \mathit{SafeComp}(\mathcal{A}^L_{min}) \to \mathit{SafeComp}(\mathcal{C})$ such that $|\eta(S)| \geq |S|$ for all $S \in \mathit{SafeComp}(\mathcal{C})$.

- We have an HD gen. co-Büchi of size $m =$ size of the largest safe component of $\mathcal{A}^L_{min}$

Why it is minimal

[Abu Radi, Kupferman 2019]

For all equivalent HD co-Büchi automaton $\mathcal{C}$ there is an injection $\eta : SafeComp(\mathcal{A}^L_{min}) \to SafeComp(\mathcal{C})$ such that $|\eta(S)| \geq |S|$ for all $S \in SafeComp(\mathcal{C})$.

- We have an HD gen. co-Büchi of size $m =$ size of the largest safe component of $\mathcal{A}^L_{min}$
- Every equivalent HD co-Büchi has a safe component of size $\geq m$

Why it is minimal

[Abu Radi, Kupferman 2019]

For all equivalent HD co-Büchi automaton $\mathcal{C}$ there is an injection $\eta : SafeComp(\mathcal{A}^L_{min}) \to SafeComp(\mathcal{C})$ such that $|\eta(S)| \geq |S|$ for all $S \in SafeComp(\mathcal{C})$.

- We have an HD gen. co-Büchi of size $m =$ size of the largest safe component of $\mathcal{A}^L_{min}$
- Every equivalent HD co-Büchi has a safe component of size $\geq m$

If we had a smaller HD gen. co-Büchi we could unfold it to get an HD co-Büchi where all components have size $< m$.

This work

We can minimize generalised HD co-Büchi automata in polynomial time.

Why it is minimal

[Abu Radi, Kupferman 2019]

For all equivalent HD co-Büchi automaton $\mathcal{C}$ there is an injection $\eta : SafeComp(\mathcal{A}^L_{min}) \to SafeComp(\mathcal{C})$ such that $|\eta(S)| \geq |S|$ for all $S \in SafeComp(\mathcal{C})$.

- We have an HD gen. co-Büchi of size $m =$ size of the largest safe component of $\mathcal{A}^L_{min}$
- Every equivalent HD co-Büchi has a safe component of size $\geq m$

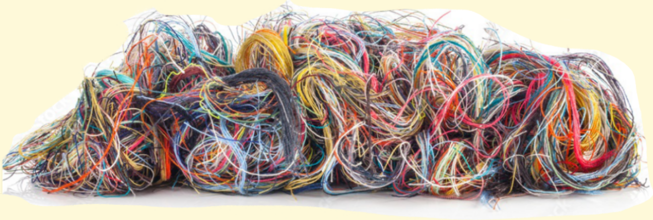
If we had a smaller HD gen. co-Büchi we could unfold it to get an HD co-Büchi where all components have size $< m$.

This work

We can minimize generalised HD co-Büchi automata in polynomial time.

If not prefix-independent $\to \sim$ apply the procedure for each residual.

A sketch of NP-completeness



NP upper bound

$\mathcal{A}$ a (history-)deterministic gen. (co-)Büchi automaton with *n* states and *k* colours.
Is there an automaton of the same type with $\leq m$ states equivalent to $\mathcal{A}$?

NP upper bound

$\mathcal{A}$ a (history-)deterministic gen. (co-)Büchi automaton with n states and k colours.
Is there an automaton of the same type with $\leq m$ states equivalent to $\mathcal{A}$?

- Guess an automaton $\mathcal{B}$ with $\leq m$ states
- Check equivalence between $\mathcal{A}$ and $\mathcal{B}$

NP upper bound

$\mathcal{A}$ a (history-)deterministic gen. (co-)Büchi automaton with n states and k colours.
Is there an automaton of the same type with $\leq m$ states equivalent to $\mathcal{A}$?

- Guess an automaton $\mathcal{B}$ with $\leq m$ states
- Check equivalence between $\mathcal{A}$ and $\mathcal{B}$ ✓

Folklore

Equivalence is decidable in PTIME between all those kinds of automata.

NP upper bound

$\mathcal{A}$ a (history-)deterministic gen. (co-)Büchi automaton with n states and k colours.
Is there an automaton of the same type with $\leq m$ states equivalent to $\mathcal{A}$?

- Guess an automaton $\mathcal{B}$ with $\leq m$ states
- Check equivalence between $\mathcal{A}$ and $\mathcal{B}$ ✓

Folklore

Equivalence is decidable in PTIME between all those kinds of automata.

But $\mathcal{B}$ could have exponentially many colours!

NP upper bound

$\mathcal{A}$ a (history-)deterministic gen. (co-)Büchi automaton with n states and k colours.
Is there an automaton of the same type with $\leq m$ states equivalent to $\mathcal{A}$?

- Guess an automaton $\mathcal{B}$ with $\leq m$ states ✓
- Check equivalence between $\mathcal{A}$ and $\mathcal{B}$ ✓

Folklore

Equivalence is decidable in PTIME between all those kinds of automata.

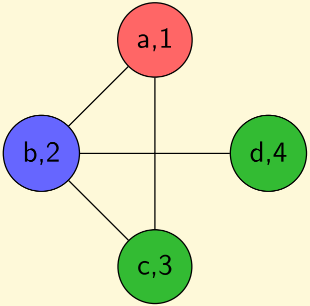
But $\mathcal{B}$ could have exponentially many colours!

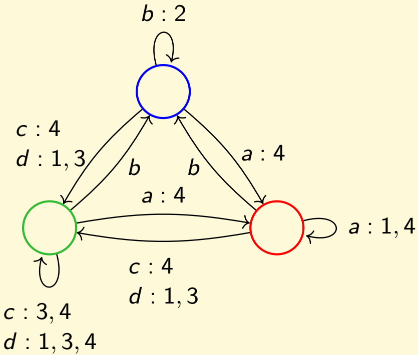
This work

If $\mathcal{B}$ exists then it can be recoloured to use $\leq \mathcal{O}(|\mathcal{A}|km)$ colours.

NP lower bound

From graph 3-colouring.
Suitable language: $L_G = \bigcap_{v \in V} (V^* vv)^\omega \cup V^*(V \setminus N(v))^\omega$.





NP lower bound

- Every k-colouring of G induces a det. gen. Büchi automaton with k states for L_G.
- A 3-state gen. Büchi automaton for L_G induces a 3-colouring of G.

NP lower bound

- Every k-colouring of G induces a det. gen. Büchi automaton with k states for L_G.
- A 3-state gen. Büchi automaton for L_G induces a 3-colouring of G.

This work

This problem is NP-complete:
Given a (history-)deterministic Büchi automaton $\mathcal{A}$ and $k \in \mathbb{N}$,
is there an equivalent one with $\leq k$ states?

NP lower bound

- Every k-colouring of G induces a det. gen. Büchi automaton with k states for L_G.
- A 3-state gen. Büchi automaton for L_G induces a 3-colouring of G.

This work

This problem is NP-complete:
Given a (history-)deterministic Büchi automaton $\mathcal{A}$ with 4 states,
is there an equivalent one with $\leq$ 3 states?

Minimising colours

This work

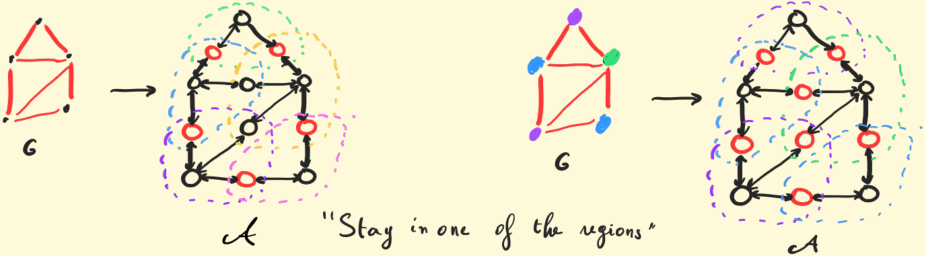
It is NP-complete to minimise both the number of states and colours for (history-)deterministic gen. (co-)Büchi automata.

Minimising colours

This work

It is NP-complete to minimise both the number of states and colours for (history-)deterministic gen. (co-)Büchi automata.

Proof idea:



Minimising colours

This work

It is NP-complete to minimise both the number of states and colours for (history-)deterministic gen. (co-)Büchi automata.

[Casares, M. 2024] $\rightarrow$ study of the complexity of simplifying conditions on ω-automata.

What is left to do

- Minimisation of Büchi, parity automata
- Are HD gen. Büchi automata more succinct than deterministic ones?
- (In-)approximability of minimisation?

What is left to do

- Minimisation of Büchi, parity automata
- Are HD gen. Büchi automata more succinct than deterministic ones?
- (In-)approximability of minimisation?

Thanks!