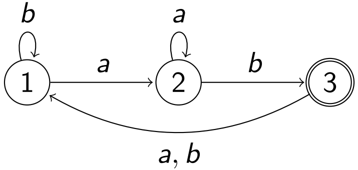
Coinduction based algorithm to decide Büchi automata equivalence

Laureline PINAULT
supervised by Denis KUPERBERG and Damien POUS

M2 internship, ENS de Lyon

February 2017 - June 2017

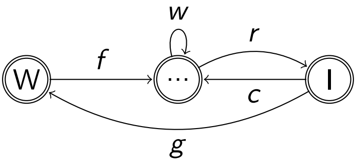
Introduction: Büchi automata



→ Accepts words having infinitely many *a*'s and infinitely many *b*'s

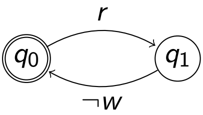
Motivations

PRINTER DRIVER



w: wait, r: request, c: cancel,
g: grant, f: finish

LTL FORMULA



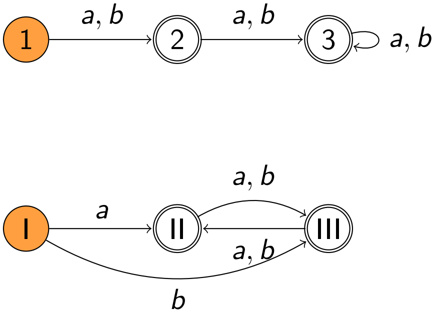
$$\mathbf{G}[r \Rightarrow \mathbf{XF}\ \neg w]$$

Table of Contents

Table of Contents

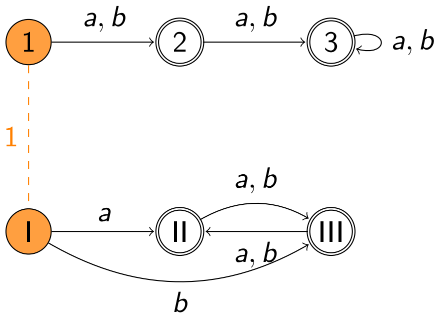
HK algorithm on DFAs

INPUT: 1 DFA, 2 states x and y ; OUTPUT: $x \sim y$?



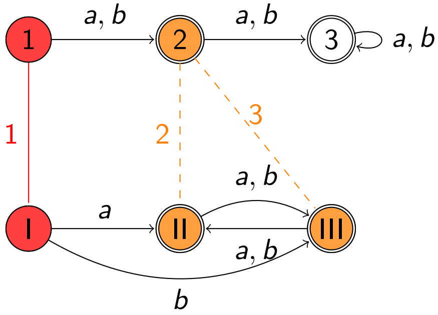
HK algorithm on DFAs

IDEA: Assume $x \sim y$ and see if there is a contradiction



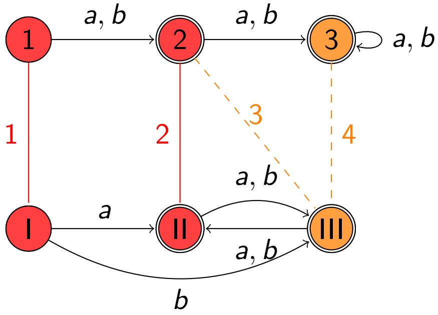
HK algorithm on DFAs

IDEA: Assume $x \sim y$ and see if there is a contradiction



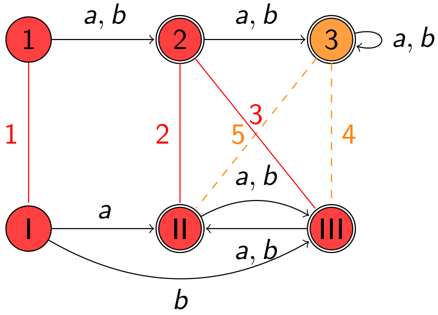
HK algorithm on DFAs

IDEA: Assume $x \sim y$ and see if there is a contradiction



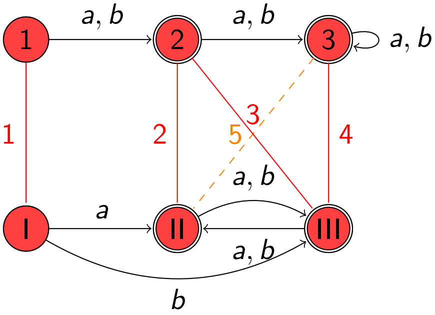
HK algorithm on DFAs

IDEA: Assume $x \sim y$ and see if there is a contradiction



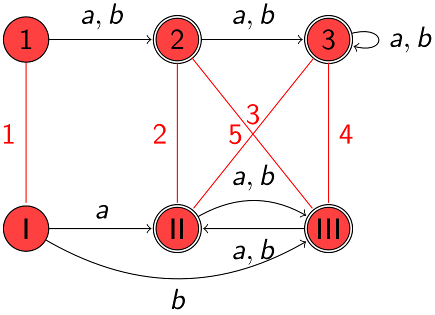
HK algorithm on DFAs

IDEA: Assume $x \sim y$ and see if there is a contradiction



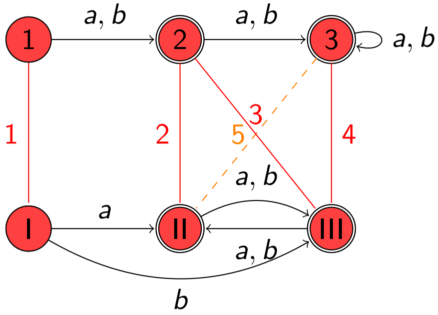
HK algorithm on DFAs

IDEA: Assume $x \sim y$ and see if there is a contradiction



HK algorithm on DFAs

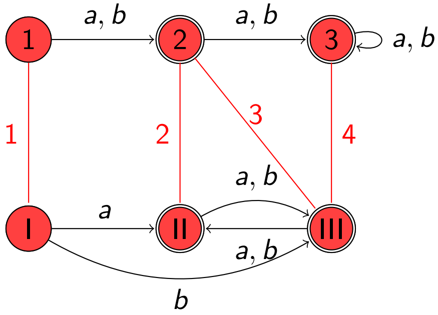
IDEA: Assume $x \sim y$ and see if there is a contradiction



Relation closed under equivalence

HK algorithm on DFAs

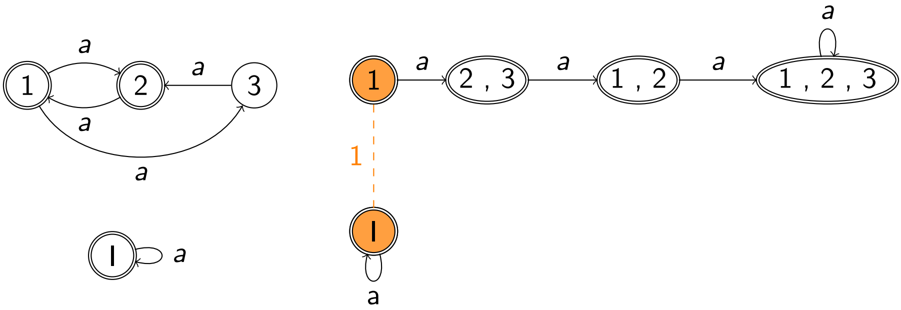
IDEA: Assume $x \sim y$ and see if there is a contradiction



Relation closed under equivalence

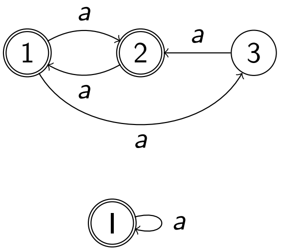
HK algorithm on NFAs

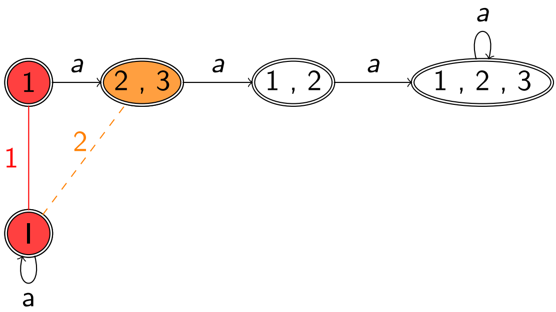
→ Run the algorithm on the powerset



HK algorithm on NFAs

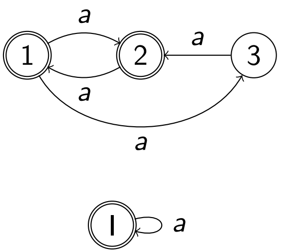
$\rightarrow$ Run the algorithm on the powerset

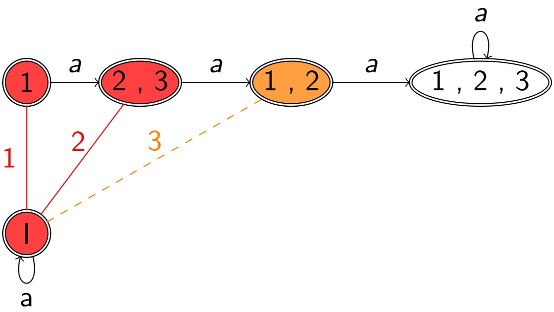




HK algorithm on NFAs

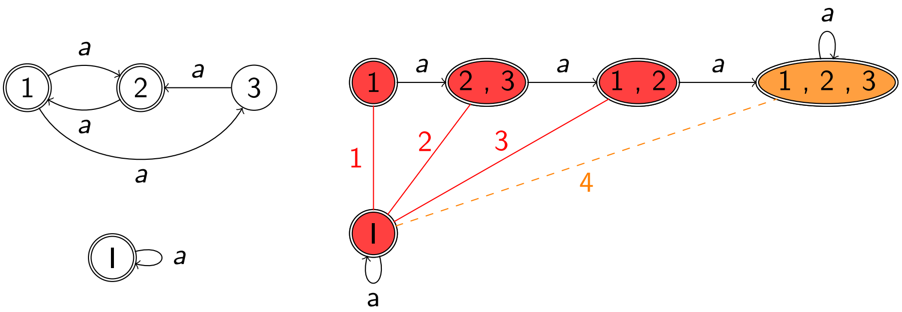
→ Run the algorithm on the powerset





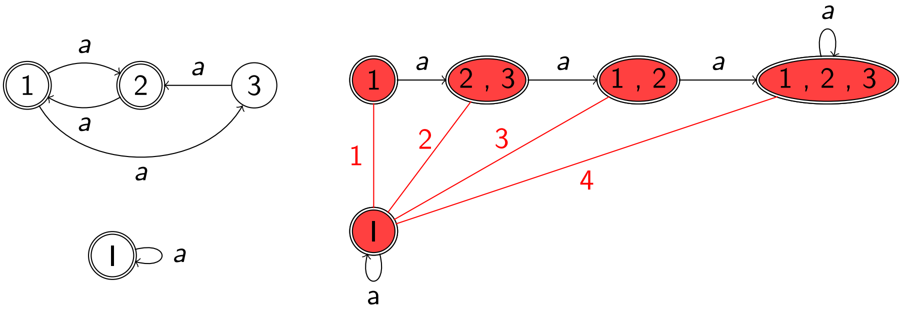
HK algorithm on NFAs

→ Run the algorithm on the powerset

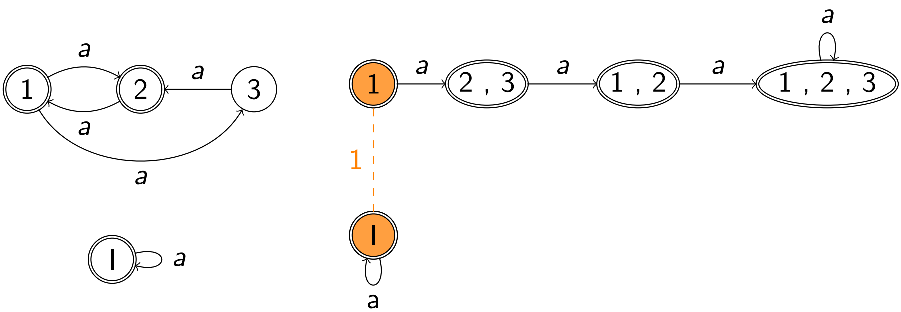


HK algorithm on NFAs

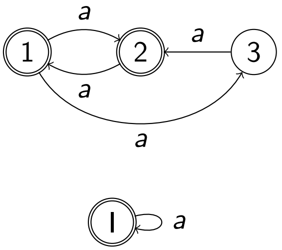
$\rightarrow$ Run the algorithm on the powerset

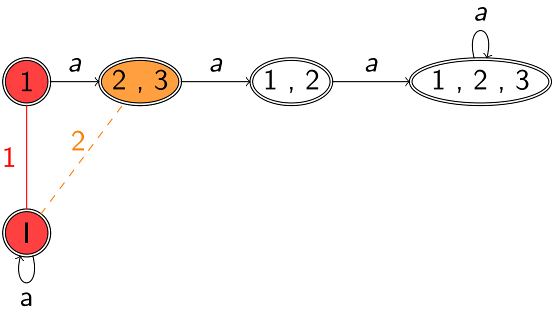


HKC algorithm on NFAs

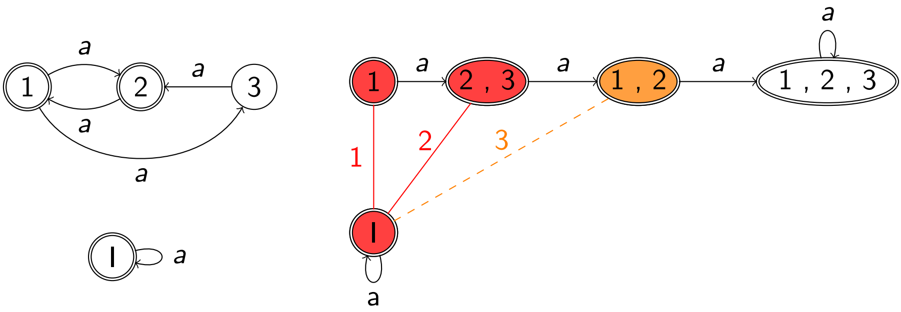


HKC algorithm on NFAs

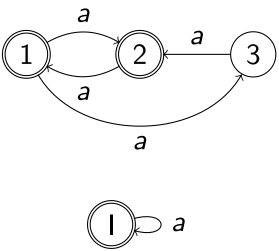


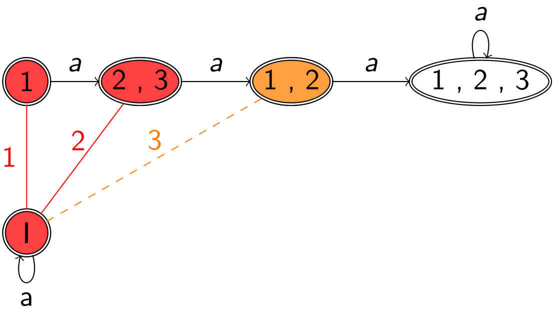


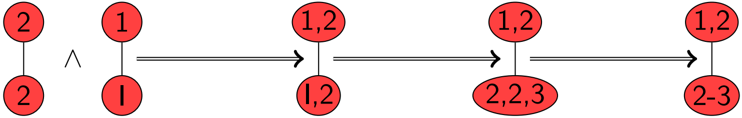
HKC algorithm on NFAs



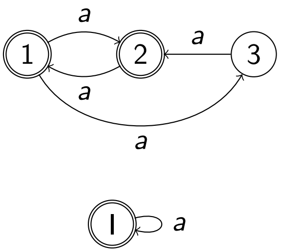
HKC algorithm on NFAs

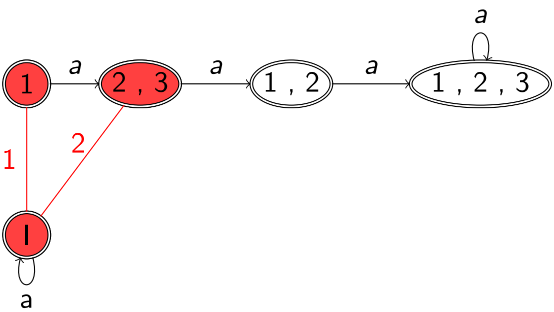


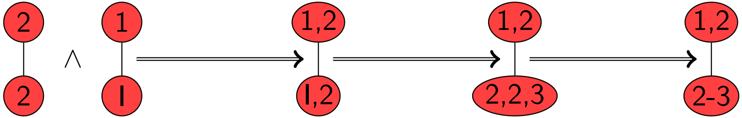




HKC algorithm on NFAs







HKC algorithm on Büchi automata?

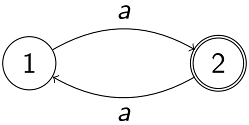


Table of Contents

Ultimately periodic words

Words of shape $u \cdot v^\omega$

$u = u_0 u_1 u_2 u_3 u_4 u_5 u_6 u_7 \ldots$: accepted by a Büchi automaton

$q_0 q_1 q_2 \mathbf{q_f} q_4 q_5 q_6 \mathbf{q_f} q_8 \ldots$: an accepting run

$u_0 u_1 u_2 (u_3 u_4 u_5 u_6)^\omega$ accepted by $q_0 q_1 q_2 \mathbf{q_f} (q_4 q_5 q_6 \mathbf{q_f})^\omega$

$\rightarrow$ Any non-empty rational language has a ultimately periodic word

Equivalence of language equivalence

Corollary

$\mathcal{L}_1 = \mathcal{L}_2$ *iff* $UP(\mathcal{L}_1) = UP(\mathcal{L}_2)$ $\qquad$ $\mathcal{L}_1, \mathcal{L}_2$ *rationals*

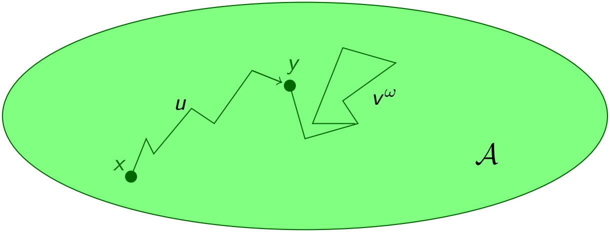
$UP((\mathcal{L}_1 \cup \mathcal{L}_2) \setminus (\mathcal{L}_1 \cap \mathcal{L}_2)) = \varnothing$

$\Rightarrow (\mathcal{L}_1 \cup \mathcal{L}_2) \setminus (\mathcal{L}_1 \cap \mathcal{L}_2) = \varnothing$

$\Rightarrow \mathcal{L}_1 = \mathcal{L}_2$

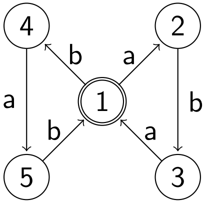
Rationality of ultimately periodic languages

$$UP(\mathcal{L}) \quad - \quad u \cdot v^\omega \quad \rightsquigarrow \quad u \cdot \$ \cdot v \quad - \quad \mathcal{L}_\$$$

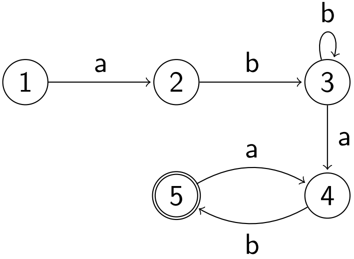


$$\mathcal{L}_\$ = \bigcup_y \mathcal{M}_{x,y} \cdot \$ \cdot \mathcal{N}_y$$

Issues when constructing $\mathcal{A}_{\mathcal{N}_y}$

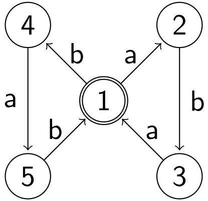


Need to read $(ab)^3$



Need to read *abab* and then *ab*

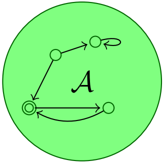
Construction of $\mathcal{A}_{\mathcal{N}_y}$



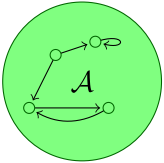
$$\begin{pmatrix} 1,0 \\ 2,0 \\ 3,0 \\ 4,0 \\ 5,0 \end{pmatrix} \xrightarrow{a} \begin{pmatrix} 2,0 \\ \perp \\ 1,1 \\ 5,0 \\ \perp \end{pmatrix} \xrightarrow{b} \begin{pmatrix} 3,0 \\ \perp \\ 4,1 \\ 1,1 \\ \perp \end{pmatrix}$$

$$1 \overset{ab}{\rightsquigarrow} 3 \;,\; 3 \overset{ab}{\rightsquigarrow} 4 \;,\; 4 \overset{ab}{\rightsquigarrow} 1$$

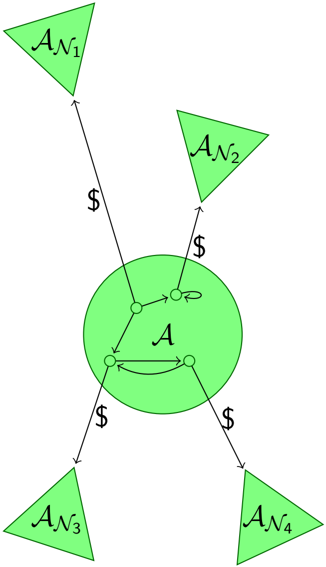
Construction of $\mathcal{A}_\$$



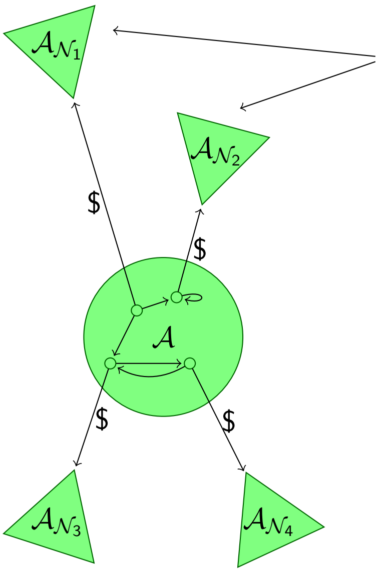
Construction of $\mathcal{A}_\$$



Construction of $\mathcal{A}_\$$



Construction of $\mathcal{A}_\$$



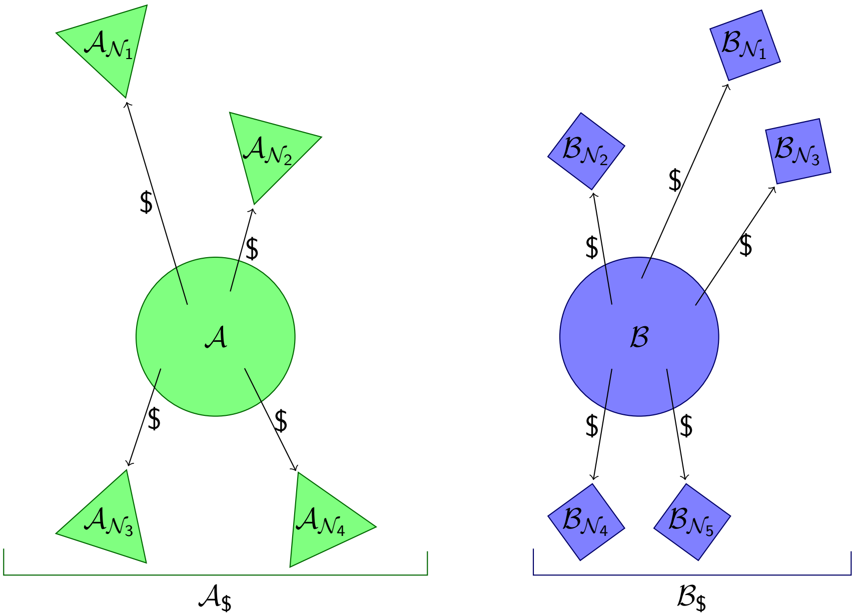
Same structure but
different accepting conditions

Table of Contents

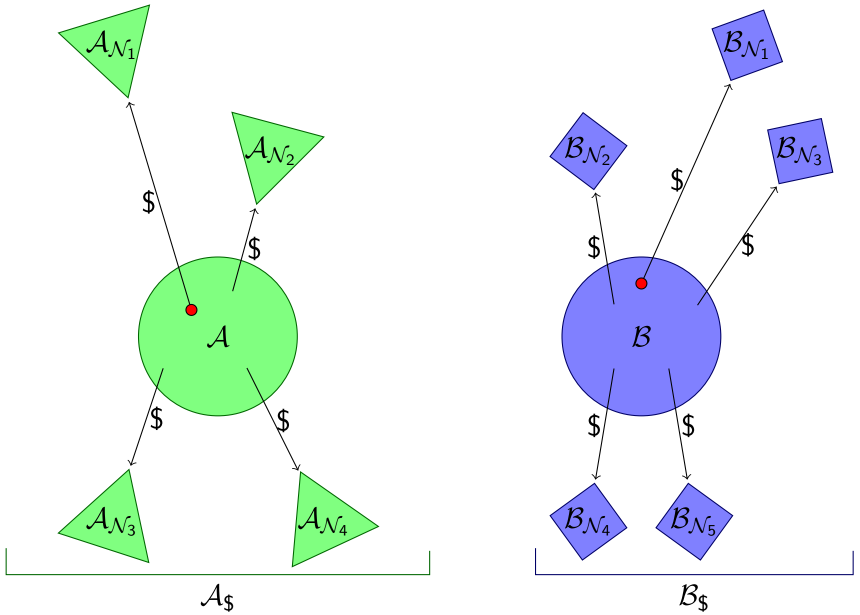
How the algorithm works

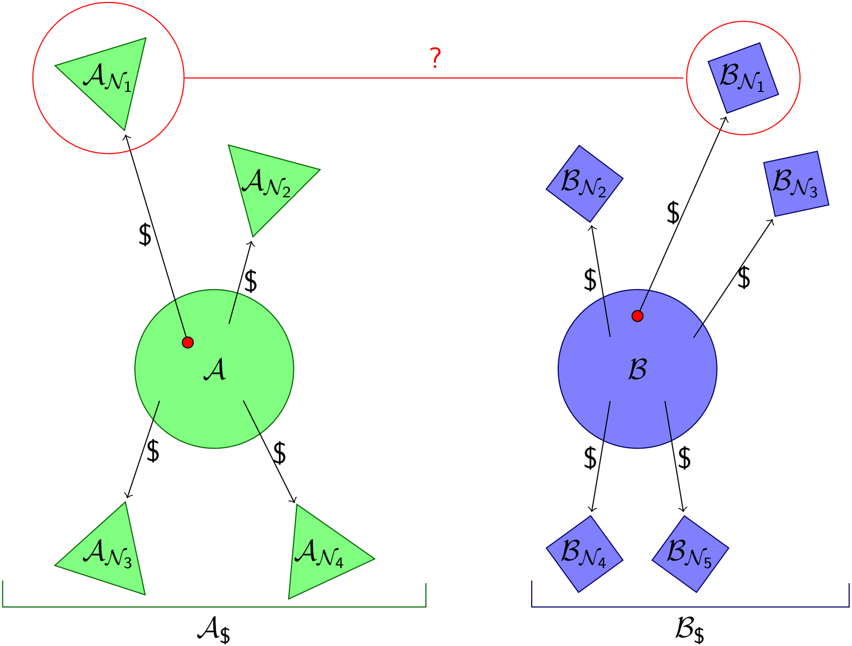
$\rightarrow$ We can run HKC on $\mathcal{A}_{\$}$

$\mathcal{A}$
$\mathcal{B}$

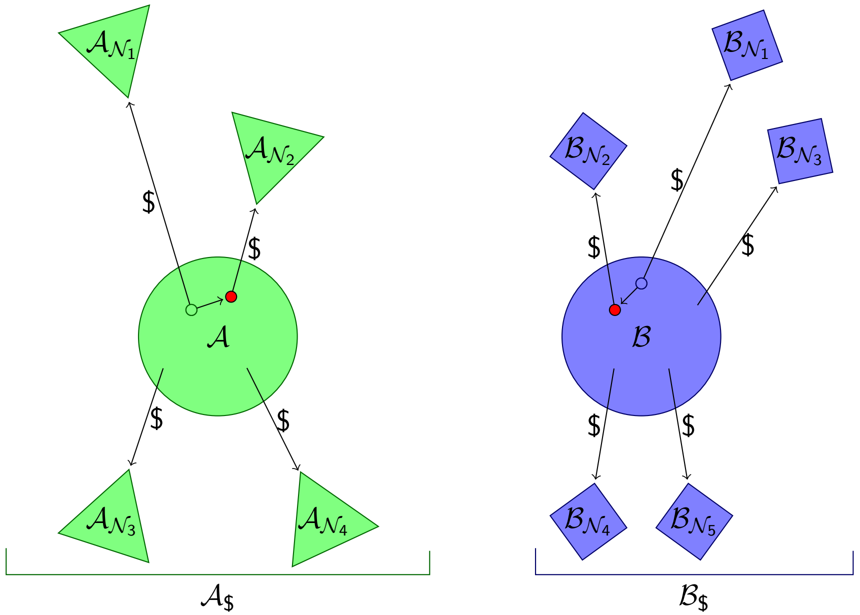


$\mathcal{A}_{\mathcal{N}_1}$
$\mathcal{A}_{\mathcal{N}_2}$
$\mathcal{A}$
$\mathcal{A}_{\mathcal{N}_3}$
$\mathcal{A}_{\mathcal{N}_4}$
$\mathcal{A}_{\$}$
$\mathcal{B}_{\mathcal{N}_1}$
$\mathcal{B}_{\mathcal{N}_2}$
$\mathcal{B}_{\mathcal{N}_3}$
$\mathcal{B}$
$\mathcal{B}_{\mathcal{N}_4}$
$\mathcal{B}_{\mathcal{N}_5}$
$\mathcal{B}_{\$}$

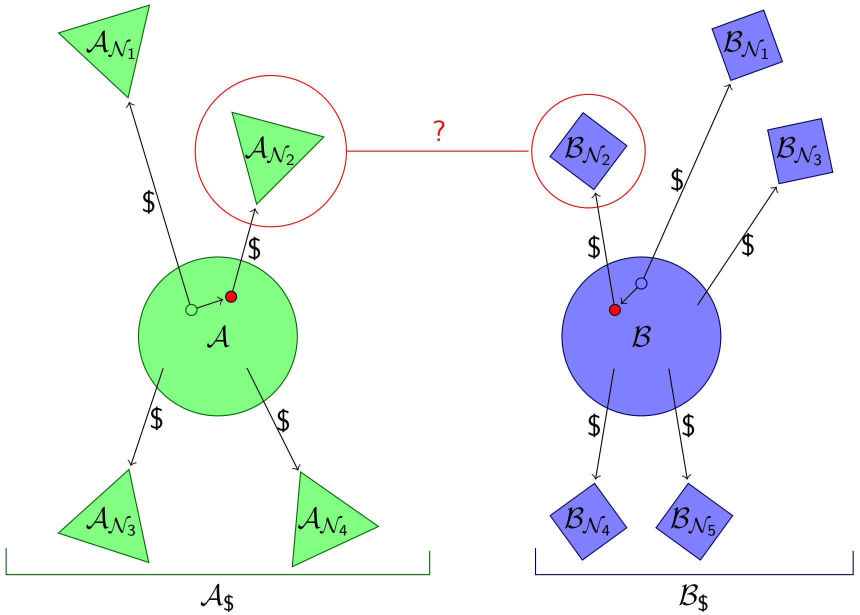

$\mathcal{A}_{\mathcal{N}_1}$
$\mathcal{A}_{\mathcal{N}_2}$
$\mathcal{A}$
$\mathcal{A}_{\mathcal{N}_3}$
$\mathcal{A}_{\mathcal{N}_4}$
$\mathcal{A}_{\$}$
$\mathcal{B}_{\mathcal{N}_1}$
$\mathcal{B}_{\mathcal{N}_2}$
$\mathcal{B}_{\mathcal{N}_3}$
$\mathcal{B}$
$\mathcal{B}_{\mathcal{N}_4}$
$\mathcal{B}_{\mathcal{N}_5}$
$\mathcal{B}_{\$}$



$\mathcal{A}_{\mathcal{N}_1}$
?
$\mathcal{B}_{\mathcal{N}_1}$
$\mathcal{A}_{\mathcal{N}_2}$
$\mathcal{B}_{\mathcal{N}_2}$
$\mathcal{B}_{\mathcal{N}_3}$
$
$
$
$
$
$
$
$
$\mathcal{A}$
$\mathcal{B}$
$\mathcal{A}_{\mathcal{N}_3}$
$\mathcal{A}_{\mathcal{N}_4}$
$\mathcal{B}_{\mathcal{N}_4}$
$\mathcal{B}_{\mathcal{N}_5}$
$\mathcal{A}_{\$}$
$\mathcal{B}_{\$}$

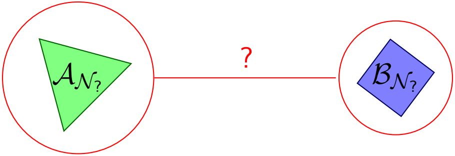


$\mathcal{A}_{\mathcal{N}_1}$
$\mathcal{A}_{\mathcal{N}_2}$
$
$
$\mathcal{A}$
$
$
$\mathcal{A}_{\mathcal{N}_3}$
$\mathcal{A}_{\mathcal{N}_4}$
$\mathcal{A}_\$$
$\mathcal{B}_{\mathcal{N}_1}$
$\mathcal{B}_{\mathcal{N}_2}$
$\mathcal{B}_{\mathcal{N}_3}$
$
$
$
$\mathcal{B}$
$
$
$\mathcal{B}_{\mathcal{N}_4}$
$\mathcal{B}_{\mathcal{N}_5}$
$\mathcal{B}_\$$



$\mathcal{A}_{\mathcal{N}_1}$
$\mathcal{A}_{\mathcal{N}_2}$
?
$\mathcal{B}_{\mathcal{N}_2}$
$\mathcal{B}_{\mathcal{N}_1}$
$\mathcal{B}_{\mathcal{N}_3}$
$
$
$
$
$
$
$\mathcal{A}$
$\mathcal{B}$
$
$
$
$
$\mathcal{A}_{\mathcal{N}_3}$
$\mathcal{A}_{\mathcal{N}_4}$
$\mathcal{B}_{\mathcal{N}_4}$
$\mathcal{B}_{\mathcal{N}_5}$
$\mathcal{A}_\$$
$\mathcal{B}_\$$

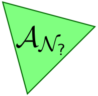
1st improvement: pre-processing



$$\hookrightarrow \{(q_1, r_1), (q_2, r_2), (q_3, r_3), ..., (q_n, r_n)\}$$

$$\forall i, \ \ o(q_i) = o(r_i)?$$

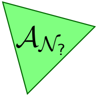
2nd improvement: state compression



: contains less states that we would think

$$\begin{pmatrix}1,0\\2,0\\3,0\\4,0\end{pmatrix} \xrightarrow{a} \begin{pmatrix}4,0\\3,0\\3,1\\1,0\end{pmatrix} \qquad \begin{pmatrix}1,0\\2,0\\3,0\\4,0\end{pmatrix} \xrightarrow{a} \begin{pmatrix}2,1\\3,0\\3,1\\3,1\end{pmatrix}$$

2nd improvement: state compression



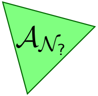
: contains less states that we would think

$$\begin{pmatrix}1,0\\2,0\\3,0\\4,0\end{pmatrix} \xrightarrow{a} \begin{pmatrix}\mathbf{4},\mathbf{0}\\3,0\\3,1\\1,0\end{pmatrix}$$

$$\begin{pmatrix}1,0\\2,0\\3,0\\4,0\end{pmatrix} \xrightarrow{a} \begin{pmatrix}2,1\\3,0\\3,1\\\mathbf{3},\mathbf{1}\end{pmatrix}$$

$$\begin{pmatrix}1,0\\2,0\\3,0\\4,0\end{pmatrix} \xrightarrow{a} \begin{pmatrix}4,0\\3,0\\3,1\\3,1\end{pmatrix}$$

2nd improvement: state compression



: contains less states that we would think

$$\begin{pmatrix}1,0\\2,0\\3,0\\4,0\end{pmatrix} \xrightarrow{a} \begin{pmatrix}4,0\\3,0\\3,1\\\mathbf{1},\mathbf{0}\end{pmatrix}$$

$$\begin{pmatrix}1,0\\2,0\\3,0\\4,0\end{pmatrix} \xrightarrow{a} \begin{pmatrix}\mathbf{2},\mathbf{1}\\3,0\\3,1\\3,1\end{pmatrix}$$

$$\begin{pmatrix}1,0\\2,0\\3,0\\4,0\end{pmatrix} \xrightarrow{a} \begin{pmatrix}2,1\\3,0\\3,1\\1,0\end{pmatrix}$$

2nd improvement: state compression

$\mathcal{A}_{\mathcal{N}_?}$: contains less states that we would think $2^{(2m)^m} \to \left(2^{2m}\right)^m$

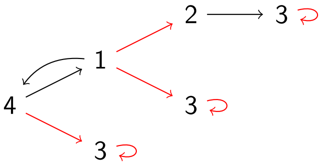
$$\begin{pmatrix} 1,0 \\ 2,0 \\ 3,0 \\ 4,0 \end{pmatrix} \xrightarrow{a} \begin{pmatrix} 4,0 \\ 3,0 \\ 3,1 \\ 1,0 \end{pmatrix}$$

$$\begin{pmatrix} 1,0 \\ 2,0 \\ 3,0 \\ 4,0 \end{pmatrix} \xrightarrow{a} \begin{pmatrix} 2,1 \\ 3,0 \\ 3,1 \\ 3,1 \end{pmatrix} \rightsquigarrow \begin{pmatrix} (4,0),(2,1) \\ (3,0) \\ (3,1) \\ (1,0),(3,1) \end{pmatrix}$$

$$\begin{pmatrix} 1,0 \\ 2,0 \\ 3,0 \\ 4,0 \end{pmatrix} \xrightarrow{a} \begin{pmatrix} 2,1 \\ 3,0 \\ 3,1 \\ 1,0 \end{pmatrix}$$

1st issue: description of the automaton

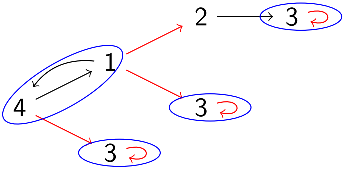
$$\begin{pmatrix} (4,0),(2,1),(3,1) \\ (3,0) \\ (3,1) \\ (1,0),(3,1) \end{pmatrix}$$



1st issue: description of the automaton

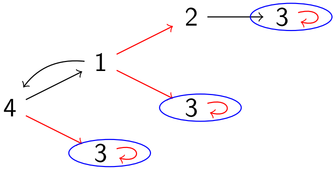
$$\begin{pmatrix} (4,0),(2,1),(3,1) \\ (3,0) \\ (3,1) \\ (1,0),(3,1) \end{pmatrix}$$

1. Compute the strongly connected components



1st issue: description of the automaton

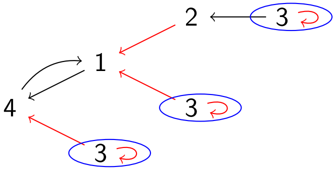
$$\begin{pmatrix} (4,0),(2,1),(3,1) \\ (3,0) \\ (3,1) \\ (1,0),(3,1) \end{pmatrix}$$



1. Compute the strongly connected components
2. Keep the ones having a final edge

1st issue: description of the automaton

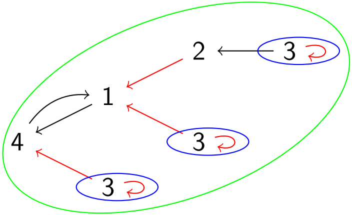
$$\begin{pmatrix} (4,0),(2,1),(3,1) \\ (3,0) \\ (3,1) \\ (1,0),(3,1) \end{pmatrix}$$



1. Compute the strongly connected components
2. Keep the ones having a final edge
3. Reverse the edges

1st issue: description of the automaton

$$\begin{pmatrix} (4,0),(2,1),(3,1) \\ (3,0) \\ (3,1) \\ (1,0),(3,1) \end{pmatrix}$$



1. Compute the strongly connected components
2. Keep the ones having a final edge
3. Reverse the edges
4. Compute the connected components

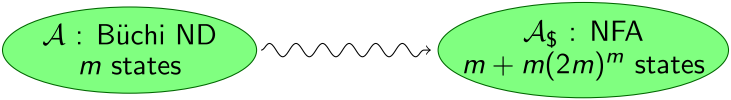
2nd issue: computation of the congruence

$\rightarrow$ The problem becomes NP-Complete

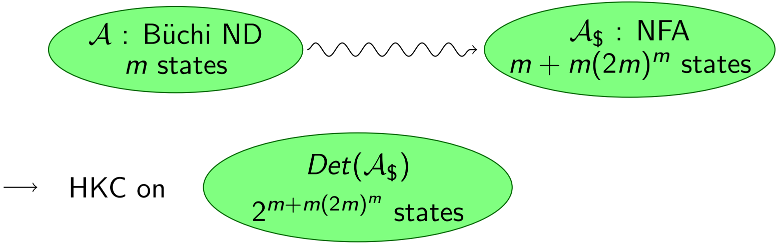
$\rightarrow$ Use of a SAT-Solver

Table of Contents

Summary



Summary


$\mathcal{A}$: Büchi ND
m states
$\mathcal{A}_\$$: NFA
$m + m(2m)^m$ states
$\longrightarrow$ HKC on
$Det(\mathcal{A}_\$)$
$2^{m+m(2m)^m}$ states

Summary

$\mathcal{A}$: Büchi ND, m states $\rightsquigarrow$ $\mathcal{A}_{\$}$: NFA, $m + m(2m)^m$ states

$\longrightarrow$ HKC on $Det(\mathcal{A}_{\$})$, $2^{m+m(2m)^m}$ states

$\longrightarrow$ HKC on $Det(\mathcal{A}_{\mathcal{N}_?})$, $2^{(2m)^m}$ states $+$ HKC on $Det(\mathcal{A})$, 2^m states

Summary

$\mathcal{A}$: Büchi ND, m states $\rightsquigarrow$ $\mathcal{A}_\$$: NFA, $m + m(2m)^m$ states

$\longrightarrow$ HKC on $Det(\mathcal{A}_\$)$, $2^{m+m(2m)^m}$ states

$\longrightarrow$ HKC on $Det(\mathcal{A}_{\mathcal{N}_?})$, $2^{(2m)^m}$ states $+$ HKC on $Det(\mathcal{A})$, 2^m states

$\longrightarrow$ HKC on $\mathcal{A}'_{\mathcal{N}_?}$: DFA, 2^{2m^2} states $+$ HKC on $Det(\mathcal{A})$, 2^m states

Future Work

- Implementation and comparison with existing methods
- Further improvements of the algorithm
- Extension to other automata classes

Thank You

- Filippo Bonchi and Damien Pous.
  Checking NFA equivalence with bisimulations up to congruence.
  In *Principle of Programming Languages (POPL)*, pages 457–468, Roma, Italy, January 2013. ACM.
  16p.

- Hugues Calbrix, Maurice Nivat, and Andreas Podelski.
  Ultimately periodic words of rational ω-languages.
  In *International Conference on Mathematical Foundations of Programming Semantics*, pages 554–566. Springer, 1993.