# An efficient algorithm to decide periodicity of *b*-recognisable sets using MSDF convention


Victor Marsault

joint work with Bernard Boigelot, Isabelle Mainz and Michel Rigo

Montefiore Institute and Department of Mathematics,
Université de Liège, Belgium


Bruxelles
2017–02–24

# Plan

# Integer base numeration systems



- $b > 1$
- Alphabet used to represent numbers: $[\![ b ]\!] = \{0, 1, \ldots, b-1\}$

- $$\begin{array}{rcccl} \mathrm{VAL} & : & [\![ b ]\!]^* & \longrightarrow & \mathbb{N} \\ & & a_n \cdots a_1 a_0 & \longmapsto & a^n b^n + \cdots + a_1 b^1 + a_0 b^0 = \sum_i^n a_i b^i \end{array}$$

In base $b = 2$, $\mathrm{VAL}(010011) = 0 + 2^3 + 0 + 0 + 2^1 + 2^0 = 19$ .

- $$\begin{array}{rcccl} \mathrm{REP} & : & \mathbb{N} & \longrightarrow & [\![ b ]\!]^* \\ & & 0 & \longmapsto & \varepsilon \\ & & n > 0 & \longmapsto & \mathrm{REP}(m)\, a\,, \quad \text{where } (m, a) \text{ is the} \\ & & & & \qquad\qquad \text{Eucl. div of } n \text{ by } b. \end{array}$$

In base $b = 2$, $\mathrm{REP}(19) = \mathrm{REP}(9)\,1 = \mathrm{REP}(4)\,11 = \cdots = 10011$ .

## $b$-recognisable sets



### Definition

*$X$: a set of integers.*

*$X$ is $b$-**recognisable** if $\mathrm{REP}(X)$ is a regular language.*

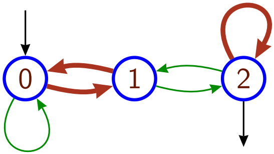
## Definition

*X: a set of integers.*

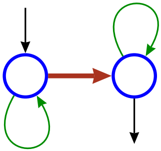
*X is b-**recognisable** if* $\mathrm{REP}(X)$ *is a regular language.*

## Theorem (folklore)

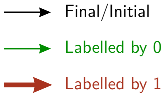
- *Each eventually-periodic set is b-recognisable.*
- *Some sets are 2-recognisable but not 3-recognisable.*




Automaton accepting
$0^*\mathrm{REP}(2+3\mathbb{N})$



Automaton accepting
$0^*\mathrm{REP}\left(\{2^i | i \in \mathbb{N}\}\right)$




Legend

# $b$-recognisable sets (2)



## Theorem (Cobham, 1969)

*$b, c$: two integer bases, multiplicatively independent*[†]*.*
*$X$: a set of integers.*

$$\left.\begin{array}{l} X \textit{ is } b\textit{-recognisable} \\ X \textit{ is } c\textit{-recognisable} \end{array}\right\} \implies X \textit{ is eventually periodic}$$

[†]such that $b^i \neq c^j$ for all $i, j > 0$.

## Corollary

$$\left\{ \text{Eventually periodic sets} \right\} = \left\{ \text{Sets } b\text{-recognisable for all } b \right\}$$

## Periodicity problem

### $\textsc{Periodicity}$ problem

- ***Parameter***: *an integer base $b > 1$.*
- ***Input***: *a deterministic finite automaton $\mathcal{A}$*
  *(hence the $b$-recognisable set $X$ accepted by $\mathcal{A}$).*
- ***Question***: *is $X$ eventually periodic ?*

### Theorem (Honkala, 1986)

$\textsc{Periodicity}$ *is decidable.*

# Periodicity problem



## Best decision algorithms

*Least Significant Digit First (LSDF)* convention: the input automaton reads its entry "from right to left".

**Theorem (Leroux, 2005)**

*With LSDF convention,* PERIODICITY *is decidable in polynomial time.*

**Theorem (M.-Sakarovitch, 2013)**

*With LSDF convention,* PERIODICITY *is decidable in linear time if the input automaton is minimal.*

# Periodicity problem

## Our contribution

**Theorem (Boigelot–Mainz–M.–Rigo, submitted)**

PERIODICITY *is decidable in* $0(b\,n\,log(n))$ *time*

*(where $n$ is the state-set cardinal.)*

# Plan

# Pseudo-morphism



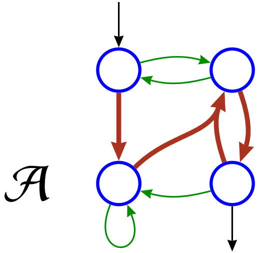
## Definition

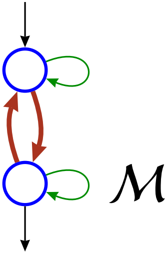
*$\mathcal{A}, \mathcal{M}$: two complete DFA*
*$\varphi$: a function {states of $\mathcal{A}$} $\to$ {states of $\mathcal{M}$}*

*$\varphi$ is a* ***pseudo-morphism*** *$\mathcal{A} \to \mathcal{M}$ if*

- *$\varphi$ maps the initial state of $\mathcal{A}$ to the initial state of $\mathcal{M}$*
- *$s \xrightarrow{a} s'$ in $\mathcal{A}$ $\iff$ $\varphi(s) \xrightarrow{a} \varphi(s')$ in $\mathcal{M}$*

(A pseudo-morphism is a morphism with no condition on final states.)



### Definition

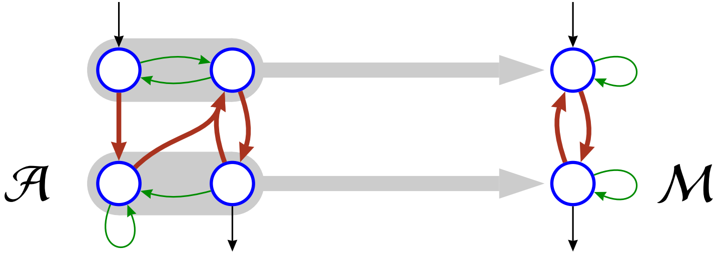
*$\mathcal{A}, \mathcal{M}$: two complete DFA*

*$\varphi$: a function $\{$states of $\mathcal{A}\} \to \{$states of $\mathcal{M}\}$*

*$\varphi$ is a* ***pseudo-morphism*** *$\mathcal{A} \to \mathcal{M}$ if*

- *$\varphi$ maps the initial state of $\mathcal{A}$ to the initial state of $\mathcal{M}$*
- *$s \xrightarrow{a} s'$ in $\mathcal{A} \iff \varphi(s) \xrightarrow{a} \varphi(s')$ in $\mathcal{M}$*

(A pseudo-morphism is a morphism with no condition on final states.)



## Definition

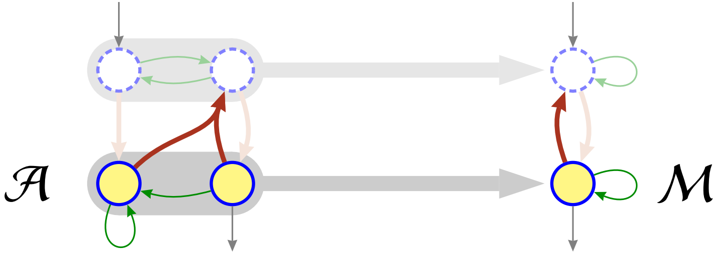
*$\mathcal{A}, \mathcal{M}$: two complete DFA*

*$\varphi$: a function {states of $\mathcal{A}$} $\rightarrow$ {states of $\mathcal{M}$}*

*$\varphi$ is a **pseudo-morphism** $\mathcal{A} \rightarrow \mathcal{M}$ if*

- *$\varphi$ maps the initial state of $\mathcal{A}$ to the initial state of $\mathcal{M}$*
- $s \xrightarrow{a} s'$ *in* $\mathcal{A} \iff \varphi(s) \xrightarrow{a} \varphi(s')$ *in* $\mathcal{M}$

(A pseudo-morphism is a morphism with no condition on final states.)



## Definition

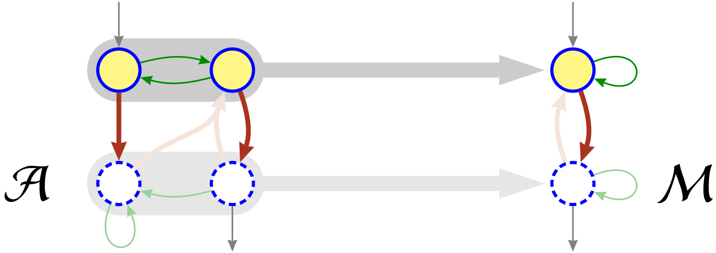
$\mathcal{A}, \mathcal{M}$: *two complete DFA*

$\varphi$: *a function* $\{$*states of* $\mathcal{A}\} \to \{$*states of* $\mathcal{M}\}$

$\varphi$ *is a* ***pseudo-morphism*** $\mathcal{A} \to \mathcal{M}$ *if*

- $\varphi$ *maps the initial state of* $\mathcal{A}$ *to the initial state of* $\mathcal{M}$
- $s \xrightarrow{a} s'$ *in* $\mathcal{A}$ $\iff$ $\varphi(s) \xrightarrow{a} \varphi(s')$ *in* $\mathcal{M}$

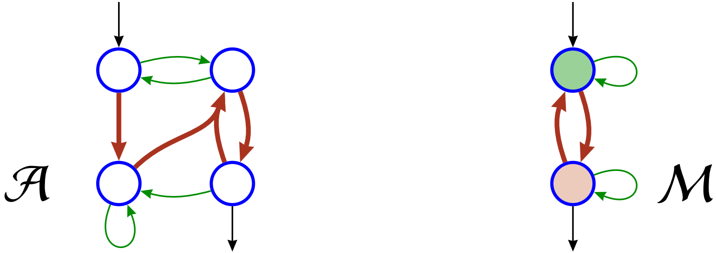
(A pseudo-morphism is a morphism with no condition on final states.)



**Lemma**

*$\mathcal{A}$: a n-state complete DFA.*
*$\mathcal{M}$: a complete DFA.*

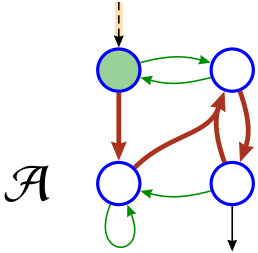
*Computing the pseudo-morphism $\varphi : \mathcal{A} \to \mathcal{M}$, if it exists, may be done in $O(b\,n)$ time.*

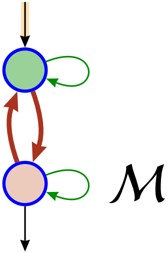


## Lemma

*$\mathcal{A}$: a $n$-state complete DFA.*
*$\mathcal{M}$: a complete DFA.*

*Computing the pseudo-morphism $\varphi : \mathcal{A} \to \mathcal{M}$, if it exists, may be done in $O(b\,n)$ time.*



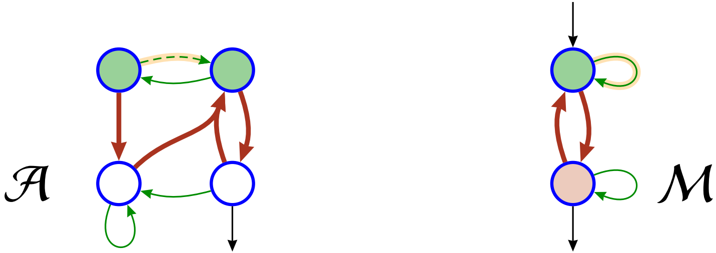
# Computing a pseudo-morphism



## Lemma

*$\mathcal{A}$: a n-state complete DFA.*
*$\mathcal{M}$: a complete DFA.*

*Computing the pseudo-morphism $\varphi : \mathcal{A} \to \mathcal{M}$, if it exists, may be done in $O(b\,n)$ time.*



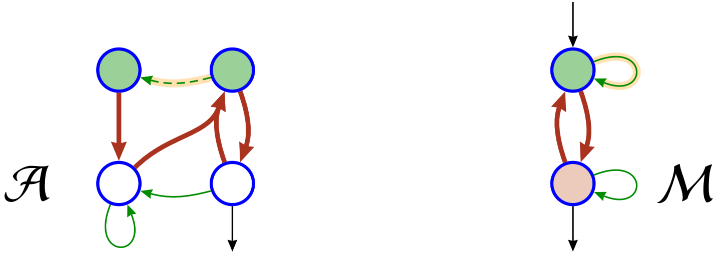
# Computing a pseudo-morphism



### Lemma

*$\mathcal{A}$: a n-state complete DFA.*
*$\mathcal{M}$: a complete DFA.*

*Computing the pseudo-morphism $\varphi : \mathcal{A} \to \mathcal{M}$, if it exists, may be done in $O(b\,n)$ time.*

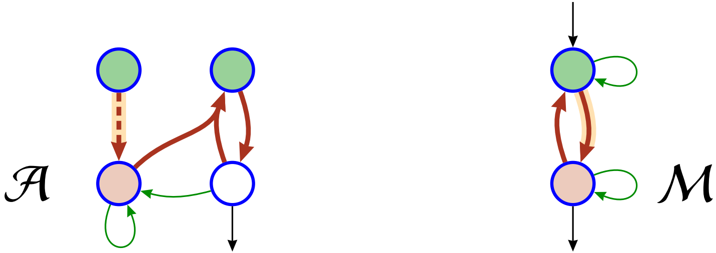


# Computing a pseudo-morphism

**Lemma**

*$\mathcal{A}$: a n-state complete DFA.*
*$\mathcal{M}$: a complete DFA.*

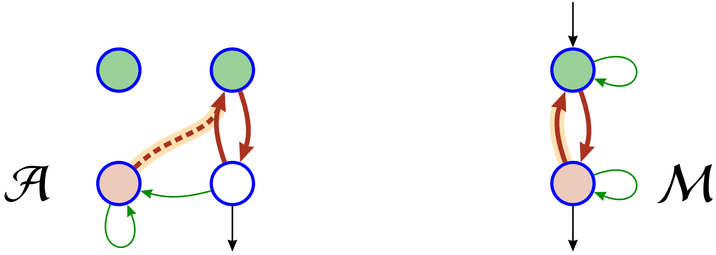
*Computing the pseudo-morphism $\varphi : \mathcal{A} \to \mathcal{M}$, if it exists, may be done in $O(b\,n)$ time.*



**Lemma**

*$\mathcal{A}$: a n-state complete DFA.*
*$\mathcal{M}$: a complete DFA.*

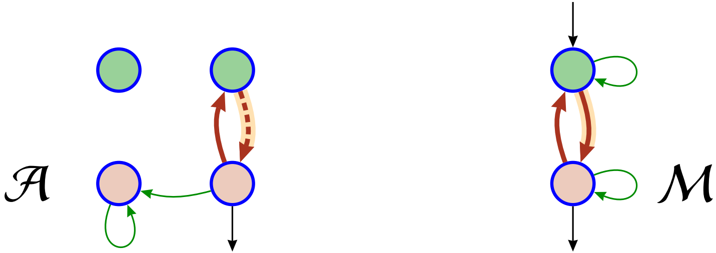
*Computing the pseudo-morphism $\varphi : \mathcal{A} \to \mathcal{M}$, if it exists, may be done in $O(b\, n)$ time.*



**Lemma**

*$\mathcal{A}$: a n-state complete DFA.*
*$\mathcal{M}$: a complete DFA.*

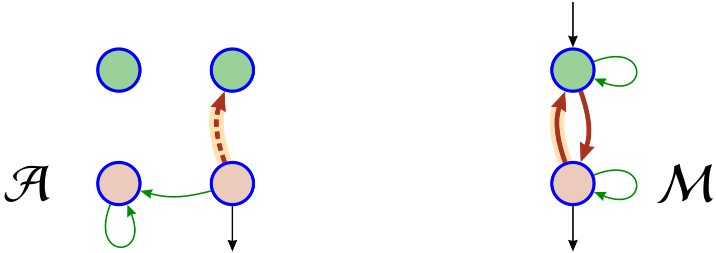
*Computing the pseudo-morphism $\varphi : \mathcal{A} \to \mathcal{M}$, if it exists, may be done in $O(b\,n)$ time.*



## Lemma

*$\mathcal{A}$: a n-state complete DFA.*
*$\mathcal{M}$: a complete DFA.*

*Computing the pseudo-morphism $\varphi : \mathcal{A} \to \mathcal{M}$, if it exists, may be done in $O(b\, n)$ time.*



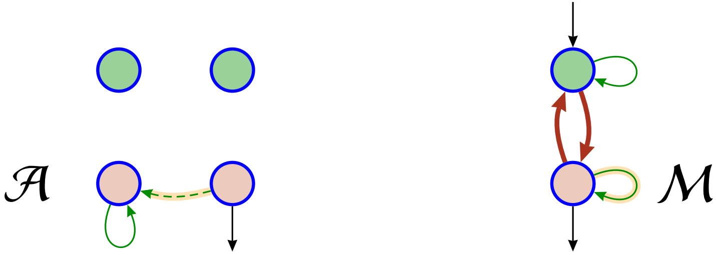
# Computing a pseudo-morphism



## Lemma

*$\mathcal{A}$: a n-state complete DFA.*
*$\mathcal{M}$: a complete DFA.*

*Computing the pseudo-morphism $\varphi : \mathcal{A} \to \mathcal{M}$, if it exists, may be done in $O(b\,n)$ time.*



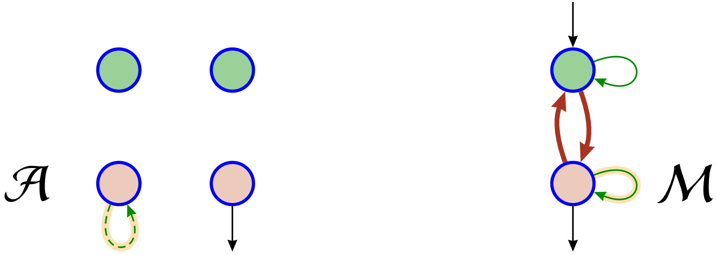
# Computing a pseudo-morphism



**Lemma**

*$\mathcal{A}$: a n-state complete DFA.*
*$\mathcal{M}$: a complete DFA.*

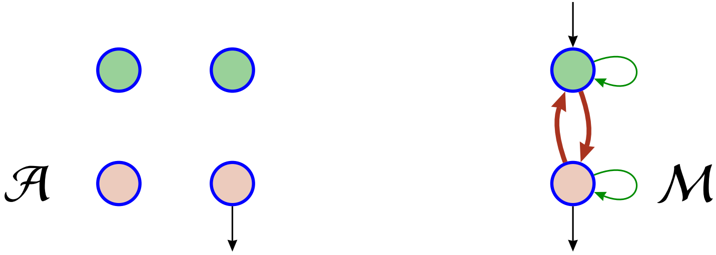
*Computing the pseudo-morphism $\varphi : \mathcal{A} \to \mathcal{M}$, if it exists, may be done in $O(b\,n)$ time.*



**Lemma**

*$\mathcal{A}$: a n-state complete DFA.*
*$\mathcal{M}$: a complete DFA.*

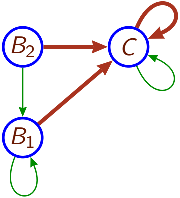
*Computing the pseudo-morphism $\varphi : \mathcal{A} \to \mathcal{M}$, if it exists, may be done in $O(b\,n)$ time.*



## Definition

*$\mathcal{A}$: a complete DFA.*
*$s, s'$: states of $\mathcal{A}$.*
*$m$: an integer.*

*$s$ and $s'$ are $m$-ultimately-equivalent (w.r.t. $\mathcal{A}$),*
*if $\forall$word $u$ of length $m$, $[\ s \xrightarrow{u} t$ and $s' \xrightarrow{u} t'$ implies $t = t'\ ]$.*




- $B_1$ and $B_2$ are 1-ult.-equiv.
- All others pairs are not ult.-equiv., as witnessed by the family $0^*$.

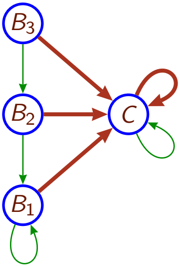
# Ultimate Equivalence



## Definition

*$\mathcal{A}$: a complete DFA.*
*$s, s'$: states of $\mathcal{A}$.*
*$m$: an integer.*

*$s$ and $s'$ are $m$-ultimately-equivalent (w.r.t. $\mathcal{A}$),*
*if $\forall$word $u$ of length $m$, $[\ s \xrightarrow{u} t$ and $s' \xrightarrow{u} t'$ implies $t = t'\ ]$.*



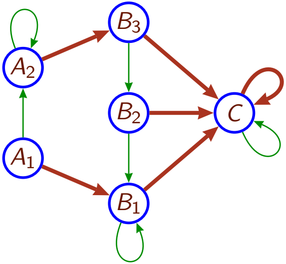

- $B_1$ and $B_2$ are 1-ult.-equiv.
- $B_2$ and $B_3$ are 2-ult.-equiv.
- $B_3$ and $B_1$ are 2-ult.-equiv.

- All others pairs are not ult.-equiv., as witnessed by the family $0^*$.

## Definition

*$\mathcal{A}$: a complete DFA.*
*$s, s'$: states of $\mathcal{A}$.*
*$m$: an integer.*

*$s$ and $s'$ are $m$-ultimately-equivalent (w.r.t. $\mathcal{A}$),*
*if $\forall$word $u$ of length $m$, $[\ s \xrightarrow{u} t$ and $s' \xrightarrow{u} t'$ implies $t = t'\ ]$.*




- $B_1$ and $B_2$ are 1-ult.-equiv.
- $B_2$ and $B_3$ are 2-ult.-equiv.
- $B_3$ and $B_1$ are 2-ult.-equiv.
- $A_1$ and $A_2$ are 3-ult.-equiv.
- All others pairs are not ult.-equiv., as witnessed by the family $0^*$.

# Computing the ultimate-Equivalence relation



$\mathcal{A}$: a DFA.
$n$: the number of states in $\mathcal{A}$.
$b$: the size of the alphabet.

By using the automaton product $\mathcal{A} \times \mathcal{A}$, it is known that:

**Lemma (folklore)**

*Ultimate-equivalence relation of $\mathcal{A}$ can be computed in $O(bn^2)$ time.*

There exists a better algorithm:

**Theorem (Béal-Crochemore, 2007)**

*Ultimate-equivalence relation of $\mathcal{A}$ can be computed in $O(b\, n\, log(n))$ time.*

# The naive automaton $\mathcal{A}_{(p,R)}$ accepting $R + p\mathbb{N}$



$p \in \mathbb{N}$: the period.
$R$: the remainder set.

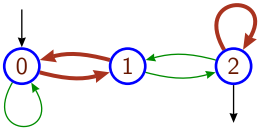
# The naive automaton $\mathcal{A}_{(p,R)}$ accepting $R + p\mathbb{N}$

$p \in \mathbb{N}$: the period.
$R$: the remainder set.

## Definition

$\mathcal{A}_{(p,R)}$:

- *Alph.:* $\{0, \ldots, b-1\}$
- *State set:* $\mathbb{Z}/p\mathbb{Z}$
- *Initial state:* 0
- *Transitions:*
  $\forall$ *state* $s$, $\forall$ *digit* $a$
  $$s \xrightarrow{a} sb + a$$
- *Final-state set:* $R$



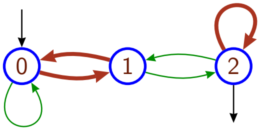

Example 1: $p = 3$ , $R = \{2\}$

$p \in \mathbb{N}$: the period.
$R$: the remainder set.

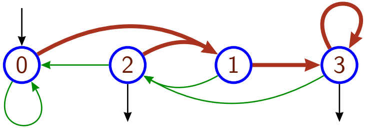
### Definition

$\mathcal{A}_{(p,R)}$:

- *Alph.:* $\{0, \ldots, b{-}1\}$
- *State set:* $\mathbb{Z}/p\mathbb{Z}$
- *Initial state:* 0
- *Transitions:*
  $\forall$ *state* $s$, $\forall$ *digit* $a$
  $$s \xrightarrow{a} sb + a$$
- *Final-state set:* $R$




Example 1: $p = 3$ , $R = \{2\}$



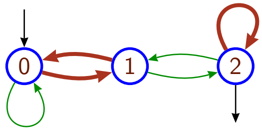

Example 2: $p = 4$ , $R = \{2, 3\}$

$p \in \mathbb{N}$: the period.
$R$: the remainder set.

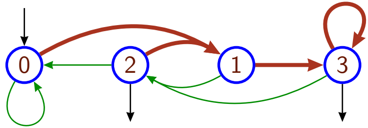
## Definition

$\mathcal{A}_{(p,R)}$:

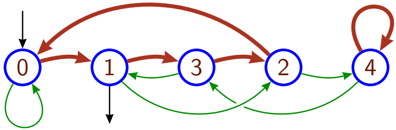
- *Alph.:* $\{0, \ldots, b-1\}$
- *State set:* $\mathbb{Z}/p\mathbb{Z}$
- *Initial state:* 0
- *Transitions:*
  $\forall$ *state* $s$, $\forall$ *digit* $a$
  $$s \xrightarrow{a} sb + a$$
- *Final-state set:* $R$




Example 1: $p = 3$ , $R = \{2\}$




Example 2: $p = 4$ , $R = \{2, 3\}$




Example 3: $p = 5$ , $R = \{1\}$

# Plan

## Theorem (Boigelot–Mainz–M.–Rigo, submitted)

*$\mathcal{A}$: a minimal DFA.*

*$X$: the b-recognisable set accepted by $\mathcal{A}$.*

*$\ell$: the total number of states in 0-circuits.*

*$X$ is purely periodic if and only if*

- *$\exists$ a pseudo-morphism $\varphi : \mathcal{A} \to \mathcal{A}_{(\ell,?)}$;*
- *states $s, s'$ such that $\varphi(s) = \varphi(s')$, are ultimately equivalent;*
- *the initial state of $\mathcal{A}$ bears a 0-loop.*

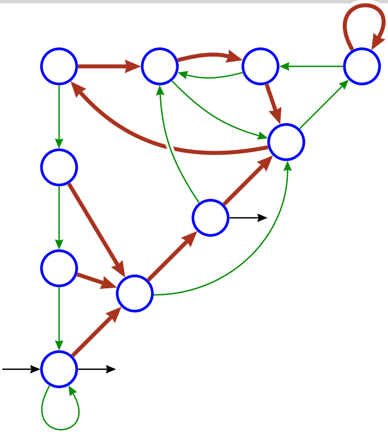
0 Start from a minimal complete DFA $\mathcal{A}$.

1 Count the number $\ell$ of states in the 0-circuits.

2 Build $\mathcal{A}_{(\ell,?)}$.

3 Compute the pseudo-morphism $\varphi : \mathcal{A} \to \mathcal{A}_{(\ell,?)}$.

4 Check that $\varphi$-equivalent states are ultimately-equivalent.



0 Start from a minimal complete DFA $\mathcal{A}$.

1 Count the number $\ell$ of states in the 0-circuits.

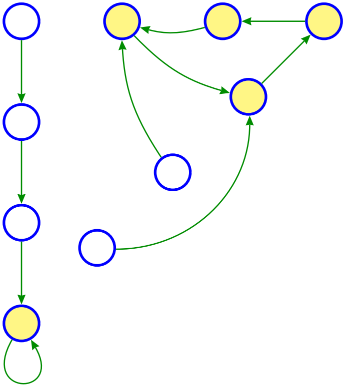
2 Build $\mathcal{A}_{(\ell,?)}$.

3 Compute the pseudo-morphism $\varphi : \mathcal{A} \to \mathcal{A}_{(\ell,?)}$.

4 Check that $\varphi$-equivalent states are ultimately-equivalent.

# Execution on an example





0 Start from a minimal complete DFA $\mathcal{A}$.

1 Count the number $\ell$ of states in the 0-circuits.

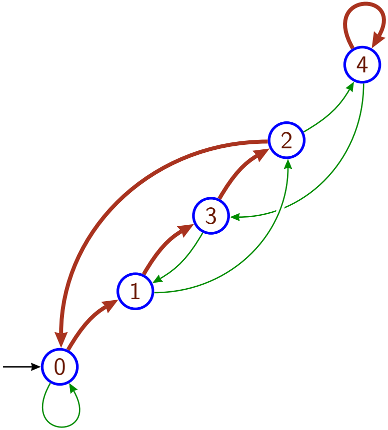
2 Build $\mathcal{A}_{(\ell,?)}$.

3 Compute the pseudo-morphism $\varphi : \mathcal{A} \to \mathcal{A}_{(\ell,?)}$.

4 Check that $\varphi$-equivalent states are ultimately-equivalent.

# Execution on an example






0 Start from a minimal complete DFA $\mathcal{A}$.

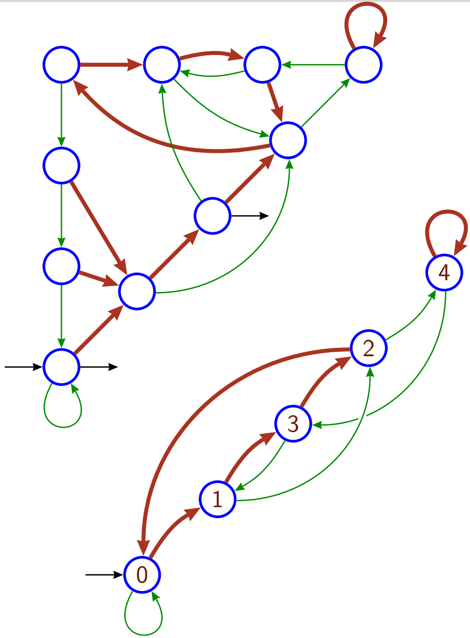
1 Count the number $\ell$ of states in the 0-circuits.

2 Build $\mathcal{A}_{(\ell,?)}$.

3 Compute the pseudo-morphism $\varphi : \mathcal{A} \to \mathcal{A}_{(\ell,?)}$.

4 Check that $\varphi$-equivalent states are ultimately-equivalent.

# Execution on an example



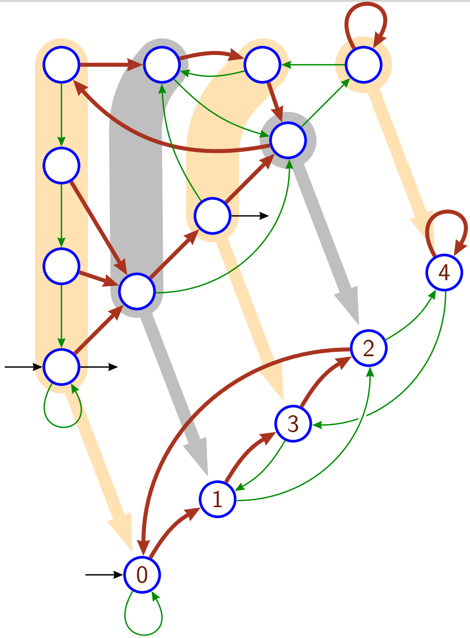

0 Start from a minimal complete DFA $\mathcal{A}$.

1 Count the number $\ell$ of states in the 0-circuits.

2 Build $\mathcal{A}_{(\ell,?)}$.

3 Compute the pseudo-morphism $\varphi : \mathcal{A} \to \mathcal{A}_{(\ell,?)}$.

4 Check that $\varphi$-equivalent states are ultimately-equivalent.



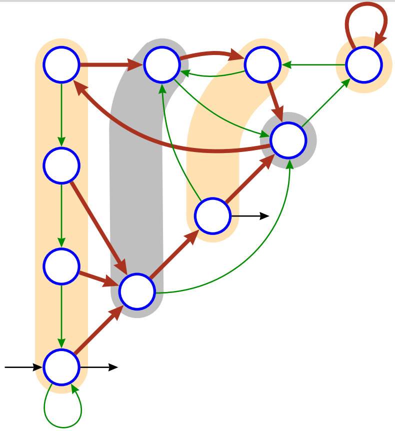

$\boxed{0}$ Start from a minimal complete DFA $\mathcal{A}$.

$\boxed{1}$ Count the number $\ell$ of states in the 0-circuits.

$\boxed{2}$ Build $\mathcal{A}_{(\ell,?)}$.

$\boxed{3}$ Compute the pseudo-morphism $\varphi : \mathcal{A} \to \mathcal{A}_{(\ell,?)}$.

$\boxed{4}$ Check that $\varphi$-equivalent states are ultimately-equivalent.



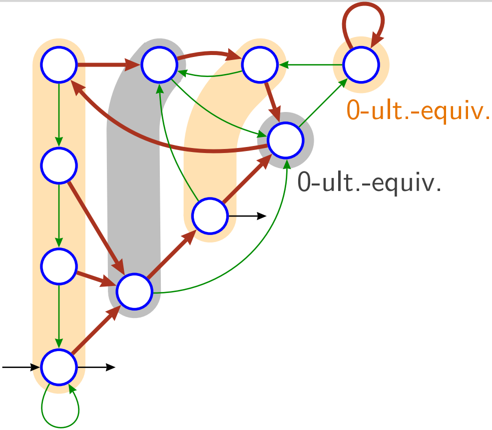
**0** Start from a minimal complete DFA $\mathcal{A}$.

**1** Count the number $\ell$ of states in the $0$-circuits.

**2** Build $\mathcal{A}_{(\ell,?)}$.

**3** Compute the pseudo-morphism $\varphi : \mathcal{A} \to \mathcal{A}_{(\ell,?)}$.

**4** Check that $\varphi$-equivalent states are ultimately-equivalent.



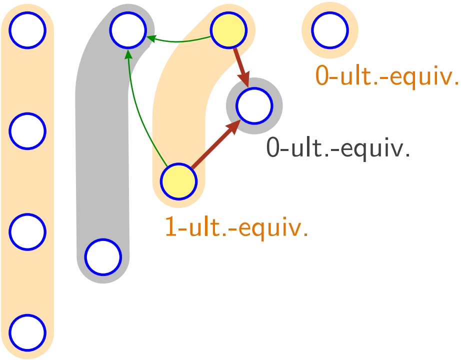

0 Start from a minimal complete DFA $\mathcal{A}$.

1 Count the number $\ell$ of states in the 0-circuits.

2 Build $\mathcal{A}_{(\ell,?)}$.

3 Compute the pseudo-morphism $\varphi : \mathcal{A} \to \mathcal{A}_{(\ell,?)}$.

4 Check that $\varphi$-equivalent states are ultimately-equivalent.



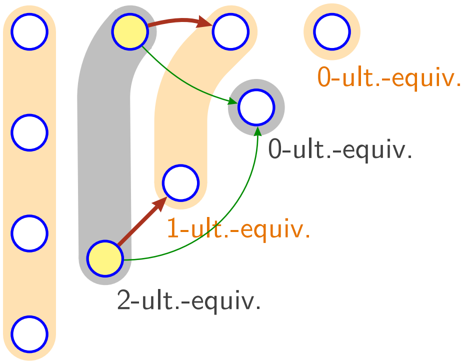

0 Start from a minimal complete DFA $\mathcal{A}$.

1 Count the number $\ell$ of states in the 0-circuits.

2 Build $\mathcal{A}_{(\ell,?)}$.

3 Compute the pseudo-morphism $\varphi : \mathcal{A} \to \mathcal{A}_{(\ell,?)}$.

4 Check that $\varphi$-equivalent states are ultimately-equivalent.




0 Start from a minimal complete DFA $\mathcal{A}$.

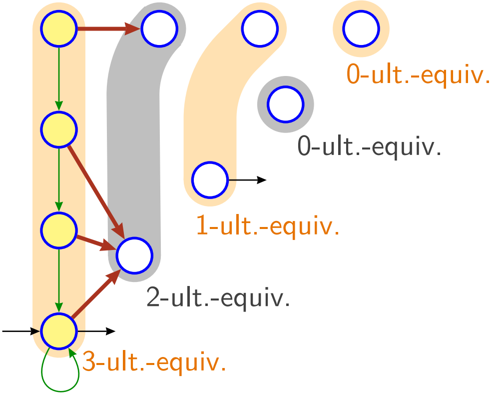
1 Count the number $\ell$ of states in the 0-circuits.

2 Build $\mathcal{A}_{(\ell,?)}$.

3 Compute the pseudo-morphism $\varphi : \mathcal{A} \to \mathcal{A}_{(\ell,?)}$.

4 Check that $\varphi$-equivalent states are ultimately-equivalent.

# Execution on an example




0 Start from a minimal complete DFA $\mathcal{A}$.

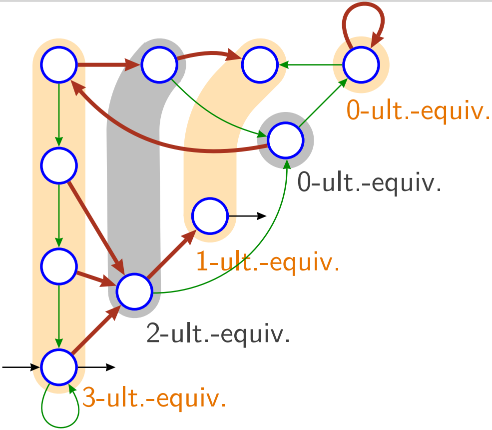
1 Count the number $\ell$ of states in the 0-circuits.

2 Build $\mathcal{A}_{(\ell,?)}$.

3 Compute the pseudo-morphism $\varphi : \mathcal{A} \to \mathcal{A}_{(\ell,?)}$.

4 Check that $\varphi$-equivalent states are ultimately-equivalent.

# Execution on an example




$\left( \begin{array}{c} \text{Then, the period is} \\ b^m \times \ell = 2^3 \times 5 = 40 \end{array} \right)$

0 Start from a minimal complete DFA $\mathcal{A}$.

1 Count the number $\ell$ of states in the 0-circuits.

2 Build $\mathcal{A}_{(\ell,?)}$.

3 Compute the pseudo-morphism $\varphi : \mathcal{A} \to \mathcal{A}_{(\ell,?)}$.

4 Check that $\varphi$-equivalent states are ultimately-equivalent.

# Impurely periodic sets



## Definition

*$S$: an integer set*

*$S$ is **impurely periodic*** $\iff$ *$S$ is eventually periodic but not purely periodic*

## Theorem (Boigelot–Mainz–M.–Rigo, submitted)

*$\mathcal{A}$: a minimal DFA.*

*$S$: the b-recognisable set accepted by $\mathcal{A}$.*

*$\ell$: the total number of states in 0-circuits **minus one**.*

*$S$ is **im**purely periodic if and only if*

- *$\exists$ a pseudo-morphism $\varphi : \mathcal{A} \to \mathcal{A}_{(\ell,?)}$;*
- *every **non-initial** states $s, s'$ such that $\varphi(s) = \varphi(s')$, are ultimately equivalent;*
- *the initial state of $\mathcal{A}$ bears a 0-loop **and has no other incoming transitions**.*

# Conclusion



Since an eventually periodic set is either purely or impurely periodic:

### Theorem (Boigelot–Mainz–M.–Rigo, submitted)

PERIODICITY *is decidable in* $0(b\, n\, log(n))$ *time*

*(where n is the state-set cardinal.)*

### Future work

- Extension to multi-dimensional sets.
- Extension to non-standard numeration systems.

# $\mathcal{A}_{(12,\{5,7\})}$ as the product $\mathcal{A}_{(4,?)} \times \mathcal{A}_{(3,?)}$