

Ultimate periodicity of b-recognisable sets: a quasilinear procedure

Victor MARSAULT,
joint work with Jacques SAKAROVITCH

CNRS / Telecom-ParisTech, Paris, France

17th International Conference on Developments in Language Theory
2013-06-19

- 1 Introduction
- 2 The Pascal automaton
- 3 UP Criterion
- 4 Conclusion and Future work

- base $b \geq 2$
- alphabet $A_b = \{0, 1, \dots, b - 1\}$

- base $b \geq 2$
- alphabet $A_b = \{0, 1, \dots, b - 1\}$

Example : binary system - "100" $\xleftrightarrow{\text{base 2}}$ 4; "110" $\xleftrightarrow{\text{base 2}}$ 6;

- base $b \geq 2$
- alphabet $A_b = \{0, 1, \dots, b - 1\}$
- value : $\pi(a_0 a_1 \dots a_n) = \sum_{i=0}^n a_i b^i$

Example : binary system - ~~"100"~~ $\xleftarrow{\text{base 2}}$ 4; ~~"110"~~ $\xleftarrow{\text{base 2}}$ 6;
"001" "011"

- base $b \geq 2$
- alphabet $A_b = \{0, 1, \dots, b-1\}$
- value : $\pi(a_0 a_1 \dots a_n) = \sum_{i=0}^n a_i b^i$

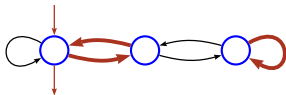
Example : binary system - ~~"100"~~ $\xleftrightarrow{\text{base } 2}$ 4; ~~"110"~~ $\xleftrightarrow{\text{base } 2}$ 6;
"001" "011"

$X \subseteq \mathbb{N}$ is b -rational

- automaton \mathcal{A}
- $L(\mathcal{A}) \xleftrightarrow{\text{base } b} X$

Theorem

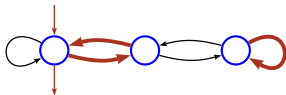
Ultimately Periodic (UP) \implies b -rational



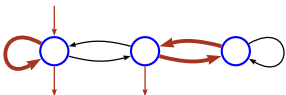
Example: automaton accepting integers congruent to 0 modulo 3

Theorem

Ultimately Periodic (UP) \implies b -rational



Example: automaton accepting integers congruent to 0 modulo 3



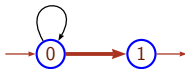
Example: automaton accepting integers congruent to 0 or 1 modulo 3

Theorem

Ultimately Periodic (UP) \implies b -rational

Fact

b -Rat $\not\Rightarrow$ (UP)



Example : accepts the powers of 2

Theorem

Ultimately Periodic (UP) \implies b -rational

Theorem (Cobham, 1969)

- X b_1 -rational
 - X b_2 -rational
 - b_1 and b_2 multiplicatively independent
- } $\implies X \in (\text{UP})$

ULTIMATE-PERIODICITY

PARAMETER :

- a base b

DATA :

- an automaton \mathcal{A}

OUTPUT :

- Does $L(\mathcal{A}) \in (UP)$?

ULTIMATE-PERIODICITY

PARAMETER :

- a base b

DATA :

- an automaton \mathcal{A}

OUTPUT :

- Does $L(\mathcal{A}) \in (UP)$?

Theorem (Honkala, 1986)

ULTIMATE-PERIODICITY is decidable.

Theorem (Leroux, 2005)

Semi-Linear(\mathbb{N}^k) is decidable in b -Rat(\mathbb{N}^k) in P-TIME.

- Quadratic complexity
- Complicated geometrical algorithm

Theorem (Leroux, 2005)

Semi-Linear(\mathbb{N}^k) is decidable in b -Rat(\mathbb{N}^k) in P-TIME.

- Quadratic complexity
- Complicated geometrical algorithm

Note (Allouche Shallit Rampersad, 2009)

- '+' is a b -rational relation.
- The class (UP) is Presburger-definable.

- Exponential complexity
- Generalisation to most common numeration systems

Theorem

\mathcal{A} : a minimal automaton

It is decidable in linear time whether $L(\mathcal{A})$ is (UP).

Corollary

ULTIMATE-PERIODICITY is solvable in $O(n \log_2(n))$ time.

- 1 Introduction
- 2 The Pascal automaton
 - Definition
 - Properties
- 3 UP Criterion
- 4 Conclusion and Future work

Parameters

- (b : the base)
- p : a period, coprime with b .
- R : a set of remainders modulo p .

Expected behaviour

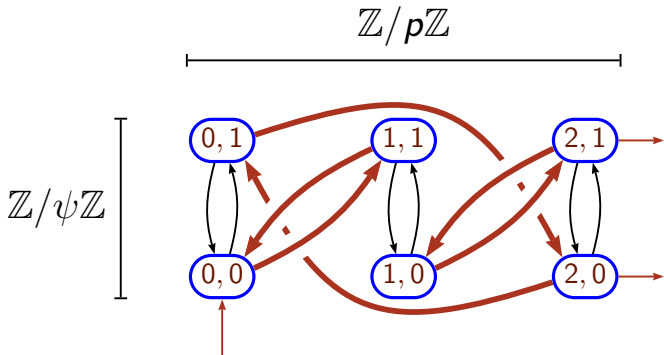
$u \in A_b^*$ accepted $\iff \pi(u) \equiv r [p], r \in R$

- $\pi(ua) = \pi(u) + a b^{|u|}$
- let ψ be the smallest integer s.t. $b^\psi \equiv 1 [p]$
- $b^k \equiv b^{(k \bmod \psi)} [p]$

- $\pi(ua) = \pi(u) + a b^{|u|}$
- let ψ be the smallest integer s.t. $b^\psi \equiv 1 [p]$
- $b^k \equiv b^{(k \bmod \psi)} [p]$

- States: $\mathbb{Z}/p\mathbb{Z} \times \mathbb{Z}/\psi\mathbb{Z}$
 $\pi(u) \bmod p \quad \uparrow \quad \quad \quad \uparrow \quad \quad \quad |u| \bmod \psi$
- Transitions: $(r, s) \xrightarrow{a} (r + ab^s, s + 1)$
- Initial state: $(0, 0)$
- Final states: $R \times \mathbb{Z}/\psi\mathbb{Z}$

- $(b = 2)$
- $p = 3$
- $\psi = 2$ (since $2^2 \equiv 1 [3]$)



Lemma

\mathcal{P}_ρ^R is deterministic and co-deterministic.

→ Each letter induces a permutation of the states

→ The syntactic monoid of \mathcal{P}_ρ^R is a group $(\simeq \mathbb{Z}/p\mathbb{Z} \times \mathbb{Z}/\psi\mathbb{Z})$.

Lemma

\mathcal{P}_ρ^R is deterministic and co-deterministic.

Proposition

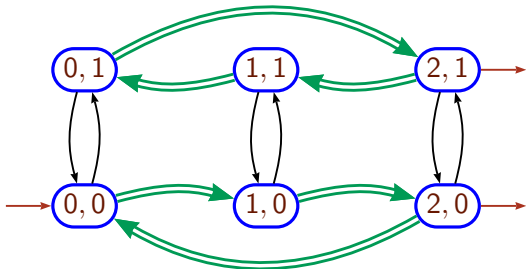
The syntactic monoid of \mathcal{P}_ρ^R is generated by 0 and 1
or 0 and $g = 10^{-1}$

Lemma

\mathcal{P}_p^R is deterministic and co-deterministic.

Proposition

The syntactic monoid of \mathcal{P}_p^R is generated by 0 and 1
or 0 and $g = 10^{-1}$



Lemma

\mathcal{P}_p^R is deterministic and co-deterministic.

Proposition

The syntactic monoid of \mathcal{P}_p^R is generated by 0 and 1
or 0 and $g = 10^{-1}$

Lemma (isotropism)

Changing the initial state of a Pascal automaton \mathcal{P}_p^R yields \mathcal{P}_S^P for some S .

Theorem

Given an automaton \mathcal{A} ,
it is decidable in linear time whether \mathcal{A} is the quotient of a Pascal.

\mathcal{P}_p^R : a Pascal automaton

\mathcal{A} : a quotient of \mathcal{P}_p^R

\sim : the equivalence relation of the quotient

Step 1 – Simplifications

Changing the alphabet of \mathcal{A} : from $\{0, 1, \dots, b-1\}$ to $\{0, g\}$.

\mathcal{P}_p^R : a Pascal automaton

\mathcal{A} : a quotient of \mathcal{P}_p^R

\sim : the equivalence relation of the quotient

Step 1 – Simplifications

Changing the alphabet of \mathcal{A} : from $\{0, 1, \dots, b-1\}$ to $\{0, g\}$.

Step 2 – Computation of the parameters

g induces in \mathcal{A} only cycles of length p .

→ Yields p then indirectly R , ψ and

t , the smallest second component among state $\sim (0, 0)$.

Step 3 – Verification

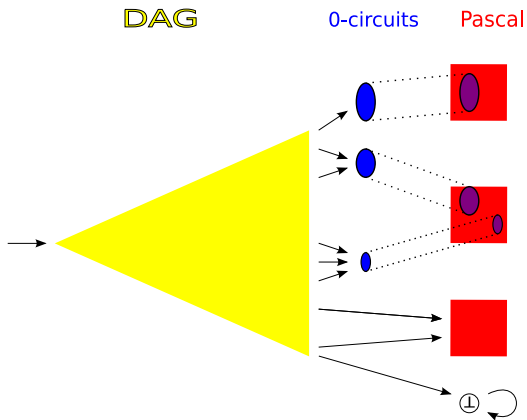
In every \sim -equivalence class, there is exactly one state of the form (s', t') with $t' < t$.

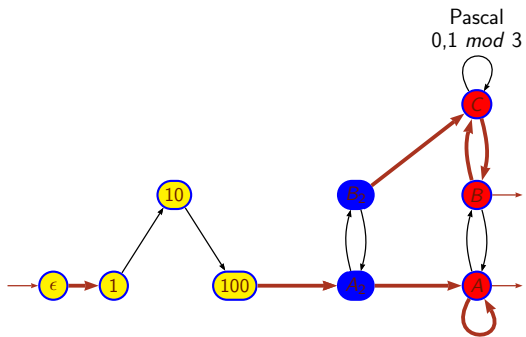
Browse the automaton \mathcal{A} marking the states:

- The initial state is marked as $(0,0)$
- If a state is marked (x,y) , then
 - Its successor by g is marked $(x + b^y, y)$
 - Its successor by 0 is marked $(x, y + 1)$ if $y + 1 < t$
or
otherwise
 - Its successor by 0 must be $(\frac{x-s}{b^t}, 0)$

- 1 Introduction
- 2 The Pascal automaton
- 3 UP Criterion
 - Statement
 - Completeness
 - Correctness
- 4 Conclusion and Future work

SCC: maximal Strongly Connected Component





Theorem

It can be verified in linear time whether a given automaton satisfies the UP-Criterion.

Theorem

It can be verified in linear time whether a given automaton satisfies the UP-Criterion.

Theorem

\mathcal{A} : a *minimal* automaton.

\mathcal{A} satisfies the UP-criterion $\iff L(\mathcal{A})$ is (UP).

Proposition 1

The UP-criterion is stable by quotient.

Proposition 1

The UP-criterion is stable by quotient.

Proposition 2

Every set of (UP) is recognized by an automaton satisfying the UP-criterion.

Hence, since the minimal quotient is unique:

Theorem (Completeness)

The minimal automaton accepting a given (UP) set satisfies the UP-criterion.

a set of (UP) : $\{n \mid n > m \text{ and } n \equiv r [p] \text{ with } r \in R\}$

- preperiod
- period
- remainder set

a set of (UP) : $\{n \mid n > m \text{ and } n \equiv r [p] \text{ with } r \in R\}$

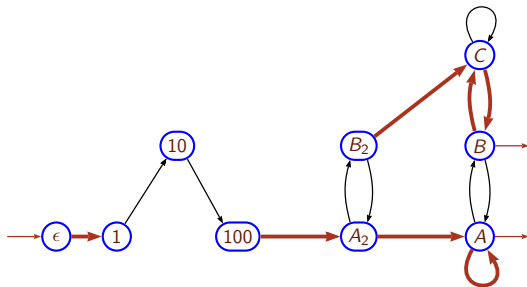
- preperiod
- period \Rightarrow Pascal's period & DAG size.
- remainder set

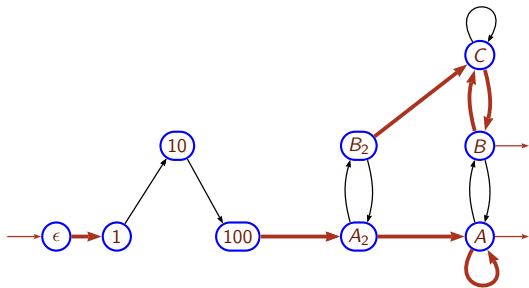
a set of (UP) : $\{n \mid n > m \text{ and } n \equiv r [p] \text{ with } r \in R\}$

- preperiod \Rightarrow 0-circuits & DAG size.
- period \Rightarrow Pascal's period & DAG size.
- remainder set

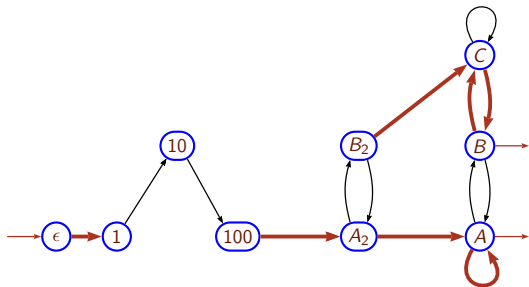
a set of (UP) : $\{n \mid n > m \text{ and } n \equiv r [p] \text{ with } r \in R\}$

- preperiod \Rightarrow 0-circuits & DAG size.
- period \Rightarrow Pascal's period & DAG size.
- remainder set \Rightarrow # of Pascal's & Pascal's remainders

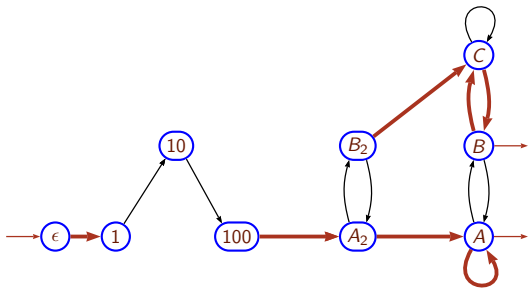




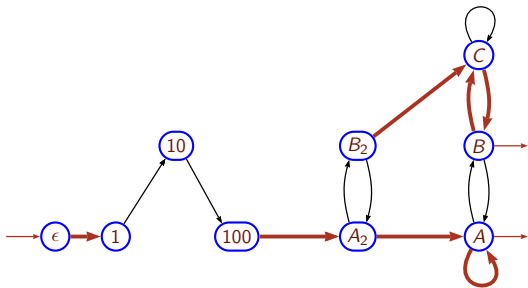
$$L = \{ u \mid u \text{ starts with } 10010^n1 \text{ and } \pi(u) \equiv 0,1 [3] \}$$



$$L = \{ u \mid \underbrace{u \text{ starts with } 10010^n 1}_{\iff \pi(u) \equiv 9 \pmod{16} \text{ and } u > 16} \text{ and } \pi(u) \equiv 0, 1 \pmod{3} \}$$



$$L = \{ u \mid \pi(u) \equiv 9 [16] \text{ and } \pi(u) \equiv 0,1 [3] \text{ and } \pi(u) > 16. \}$$



$$L = \{ u \mid \pi(u) \equiv 9, 25 \pmod{48} \text{ and } \pi(u) > 16. \}$$

- 1 Introduction
- 2 The Pascal automaton
- 3 UP Criterion
- 4 Conclusion and Future work

Conclusion

- Quasilinear algorithm to decide whether a DFA is (UP)
- Structural characterisation of minimal (UP) DFA

Conclusion

- Quasilinear algorithm to decide whether a DFA is (UP)
- Structural characterisation of minimal (UP) DFA

Future work

- Getting rid of the minimality condition
→ Work in progress...
- Getting rid of determinism condition
→ Seems unrealistic with this method.
- Generalising this method to U-Systems
→ The “*isotropism lemma*” is false in the general case
→ Yields an EXP-TIME algorithm (no better than [ASR'09])