Ultimate periodicity of b-recognisable sets: a quasilinear procedure

Victor MARSAULT,
joint work with Jacques SAKAROVITCH

CNRS / Telecom-ParisTech, Paris, France

Séminaire, Liège
2014-01-15

Outline

Integer base

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

Integer base

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

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

Integer base

- base $b \geq 2$
- alphabet $A_b = \{0, 1, \cdots, b-1\}$
- LSDF representation (Least Significant Digit First)

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

"001" "011"

Integer base

- base $b \geq 2$
- alphabet $A_b = \{0, 1, \cdots, b-1\}$
- LSDF representation (Least Significant Digit First)
- value : $\pi(a_0 a_1 \cdots a_n) = \sum_{i=0}^{n} a_i b^i$

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

"001" "011"

Integer base

- base $b \geq 2$
- alphabet $A_b = \{0, 1, \cdots, b-1\}$
- LSDF representation (Least Significant Digit First)
- value : $\pi(a_0 a_1 \cdots a_n) = \sum_{i=0}^{n} a_i b^i$

Example : binary system - ~~"100"~~ $\overset{base\ 2}{\longleftrightarrow}$ 4; ~~"110"~~ $\overset{base\ 2}{\longleftrightarrow}$ 6;

"001" "011"

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

- automaton $\mathcal{A}$
- $L(\mathcal{A}) \overset{base\ b}{\longleftrightarrow} S$

Ultimately periodic sets

Definition: the class (UP)

An integer set S belongs to (UP) if there exists

- m: preperiod
- p: period
- R: remainder set

such that $\forall\, n > m, \quad n \in S \iff n \bmod p \in R.$

Properties

- (UP) is stable by union and intersection.
- Every finite set belongs to (UP).

State of the art

Theorem

Ultimately Periodic (UP) $\Longrightarrow$ b-rational

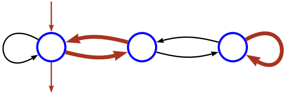


Figure: automaton accepting integers congruent to 0 modulo 3

State of the art

Theorem

Ultimately Periodic (UP) $\Longrightarrow$ b-rational

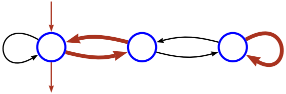


Figure: automaton accepting integers congruent to 0 modulo 3

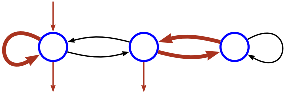


Figure: automaton accepting integers congruent to 0 or 1 modulo 3

State of the art

Theorem

Ultimately Periodic (UP) $\Longrightarrow$ b-rational

Fact

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

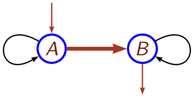


Figure: automaton accepting the powers of 2

State of the art

Theorem

Ultimately Periodic (UP) $\Longrightarrow$ b-rational

Theorem (Cobham, 1969)

$$\left.\begin{array}{l} \bullet\ S\ b_1\text{-rational} \\ \bullet\ S\ b_2\text{-rational} \\ \bullet\ b_1 \text{ and } b_2 \text{ multiplicatively independent} \end{array}\right\} \Longrightarrow S \in (\text{UP})$$

Corollary

$$(UP) = \bigcap_{b \in \mathbb{N}} b\text{-Rat}$$

Problem

ULTIMATE-PERIODICITY

PARAMETER :

- a base b

INPUT :

- a deterministic automaton $\mathcal{A}$

OUTPUT :

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

Problem

ULTIMATE-PERIODICITY

PARAMETER :

- a base b

INPUT :

- a deterministic automaton $\mathcal{A}$

OUTPUT :

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

Theorem (Honkala, 1986)

ULTIMATE-PERIODICITY is decidable.

Two orthogonal generalisations

Theorem (Leroux, 2005)

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

$\rightarrow$ Quadratic complexity

$\rightarrow$ Complicated geometrical algorithm

Two orthogonal generalisations

Theorem (Leroux, 2005)

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

→ Quadratic complexity
→ Complicated geometrical algorithm

Other proof of Honkala's Theorem (ARS, 2009)

- '$+$' is a b-rational relation.
- The class (UP) is definable by a Presburger formula.
- Presburger arithmetic is decidable

→ Exponential complexity

Two orthogonal generalisations

Theorem (Leroux, 2005)

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

$\rightarrow$ Quadratic complexity

$\rightarrow$ Complicated geometrical algorithm

Other proof of Honkala's Theorem (ARS, 2009)

- '$+$' is a b-rational relation.
- The class (UP) is definable by a Presburger formula.
- Presburger arithmetic is decidable

$\rightarrow$ Exponential complexity

Generalisation of this method (CRS, 2012)

ULTIMATE-PERIODICITY is decidable for Pisot U-systems.

Contribution

Theorem

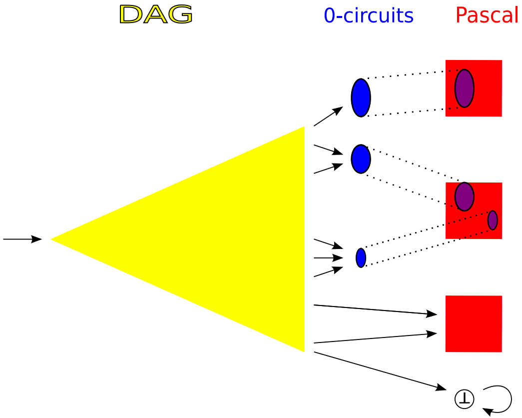
$\mathcal{A}$: a minimal deterministic automaton.
It is decidable in linear time whether $L(\mathcal{A})$ is (UP).

Corollary

ULTIMATE-PERIODICITY is solvable in quasilinear† time.

†In $O(k + n \log n)$, n being the number of states and k the number of transitions.

The UP-criterion



The UP-criterion (2)

Proposition

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

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

Proposition

It is decidable in linear time whether an automaton satisfies the UP-criterion.

Outline

The Pascal automaton $\mathcal{P}_p^R$

Parameters

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

Expected behaviour

$$u \in {A_b}^* \text{ accepted by } \mathcal{P}_p^R \quad \Longleftrightarrow \quad \pi(u) \equiv r\ [p],\ r \in R.$$

The Pascal automaton $\mathcal{P}_p^R$ (2)

LSDF Representation yields:

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

The Pascal automaton $\mathcal{P}_p^R$ (2)

LSDF Representation yields:

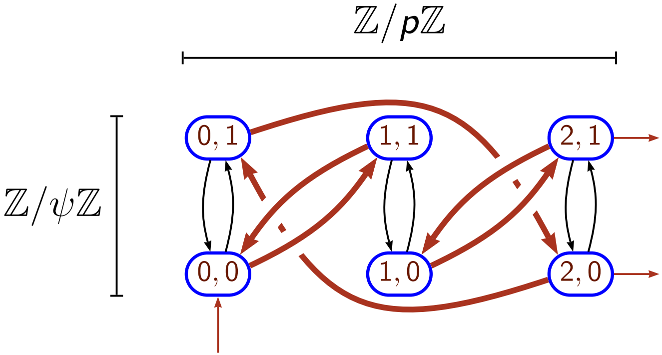
- $\pi(ua) = \pi(u) + a\, b^{|u|}$
- let ψ be the smallest integer s.t. $b^\psi \equiv 1\ [p]$
- $b^k \equiv b^{(k \bmod \psi)} [p]$

- States: $\mathbb{Z}/p\mathbb{Z} \times \mathbb{Z}/\psi\mathbb{Z}$

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

Example: $\mathcal{P}_3^{\{2\}}$

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



Structural properties

Lemma

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

Structural properties

Lemma

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

Lemma (isotropism)

Changing the initial state of a Pascal automaton $\mathcal{P}_p^R$ yields a Pascal automaton $\mathcal{P}_p^S$ with the same period p but a different remainder S.

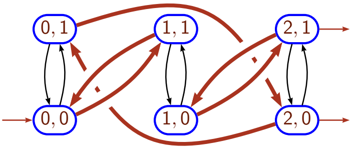
The "good" generator g

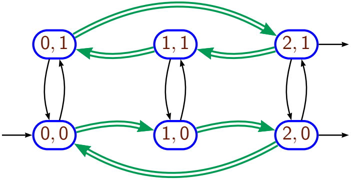
Definition

Fresh letter $g \ = \ 1\,0^{-1}$

Remark

The digit 1 is equivalent to $g0$.





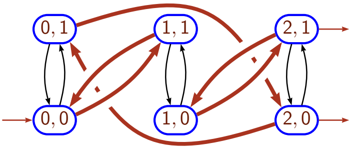
The "good" generator g

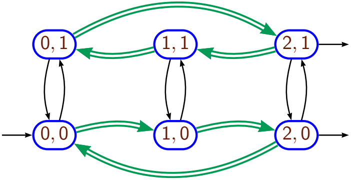
Definition

Fresh letter $g = 1\,0^{-1}$

Remark

A digit d is equivalent to $\underbrace{g \cdots g}_{d \text{ times}} 0\,.$





Theorem

It is decidable in linear time whether
an automaton $\mathcal{A}$ is the quotient of a Pascal automaton.

Pascal Recognition Algorithm

Input: an automaton $\mathcal{A}$.
Output: Is $\mathcal{A}$ the quotient of a Pascal automaton?

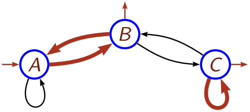
Outline of the algorithm

- Step 1 – Simplifications : changing the alphabet
- Step 2 – Computation of the parameters (p, R, etc)
- Step 3 – Verification

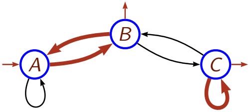
Step 1 – Simplifications

Changing the alphabet of $\mathcal{A}$: $\{0,1\} \longrightarrow \{0,g\}$

1. Remove every transition labelled neither by 0 nor 1.
2. Replace every transition by 1 by one labelled by $g = 10^{-1}$.



Before

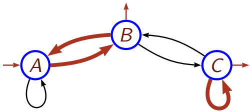


After

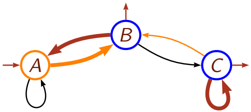
Step 1 – Simplifications

Changing the alphabet of $\mathcal{A}$: $\{0, 1\} \longrightarrow \{0, g\}$

1. Remove every transition labelled neither by 0 nor 1.
2. Replace every transition by 1 by one labelled by $g = 10^{-1}$.



Before

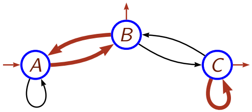


After

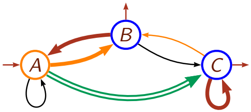
Step 1 – Simplifications

Changing the alphabet of $\mathcal{A}$: $\{0,1\} \longrightarrow \{0,g\}$

1. Remove every transition labelled neither by 0 nor 1.
2. Replace every transition by 1 by one labelled by $g = 10^{-1}$.



Before

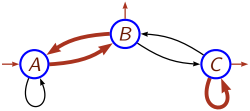


After

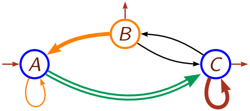
Step 1 – Simplifications

Changing the alphabet of $\mathcal{A}$: $\{0,1\} \longrightarrow \{0,g\}$

1. Remove every transition labelled neither by 0 nor 1.
2. Replace every transition by 1 by one labelled by $g = 10^{-1}$.



Before



After

Changing the alphabet of $\mathcal{A}$: $\{0,1\} \longrightarrow \{0,g\}$

1. Remove every transition labelled neither by 0 nor 1.
2. Replace every transition by 1 by one labelled by $g = 10^{-1}$.

Step 1 – Simplifications

Changing the alphabet of $\mathcal{A}$: $\{0,1\} \longrightarrow \{0,g\}$

1. Remove every transition labelled neither by 0 nor 1.
2. Replace every transition by 1 by one labelled by $g = 10^{-1}$.

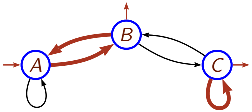
Before

After

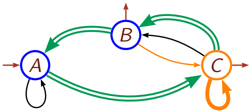
Step 1 – Simplifications

Changing the alphabet of $\mathcal{A}$: $\{0,1\} \longrightarrow \{0,g\}$

1. Remove every transition labelled neither by 0 nor 1.
2. Replace every transition by 1 by one labelled by $g = 10^{-1}$.



Before



After

Step 1 – Simplifications

Changing the alphabet of $\mathcal{A}$: $\{0,1\} \longrightarrow \{0,g\}$

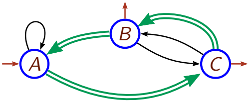
1. Remove every transition labelled neither by 0 nor 1.
2. Replace every transition by 1 by one labelled by $g = 10^{-1}$.

Step 2 – Computation of the parameters

Proposition

If $\mathcal{A}$ is quotient of the Pascal automaton $\mathcal{P}_p^R$; then

- The letter g induces in $\mathcal{A}$ cycles of length p.



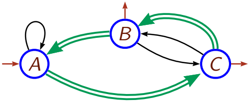
period $p = 3$

Step 2 – Computation of the parameters

Proposition

If $\mathcal{A}$ is quotient of the Pascal automaton $\mathcal{P}_p^R$; then

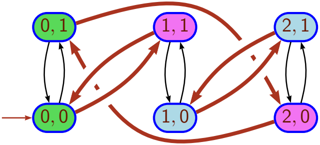
- The letter g induces in $\mathcal{A}$ cycles of length p.
- The remainder set R contains r iff g^r is accepted by $\mathcal{A}$.

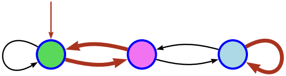


period $p = 3$

remainder set $R = \{1, 2\}$

Step 2 – Computation of the parameters





Definition: Characteristic parameter of a quotient

It is the smallest state merged with the initial state.

In the example, it is $(0, 1)$.

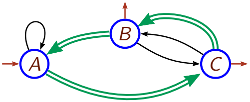
Step 2 – Computation of the parameters

Proposition

If $\mathcal{A}$ is quotient of the Pascal automaton $\mathcal{P}_p^R$; then

- The letter g induces in $\mathcal{A}$ cycles of length p.
- The remainder set R contains r iff g^r is accepted by $\mathcal{A}$.
- The characteristic parameter (s, t) of the quotient is def. by

$$i \xrightarrow{g^s} \xrightarrow{0^t} i$$

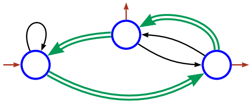


period $p = 3$
remainder set $R = \{1, 2\}$
parameter $(s, t) = (0, 1)$

Use the previous computed parameters to build *the* quotient of $\mathcal{P}_p^R$ associated with (s, t) .

- state set : $\mathbb{Z}/p\mathbb{Z} \times \mathbb{Z}/t\mathbb{Z}$ $3 \times 1 \;=\; 3$ states
- transitions:

$$(x, z) \xrightarrow{0} (x, z+1) \qquad \text{if } z < t-1$$
$$(x, z) \xrightarrow{0} \left(\tfrac{x-s}{b^t}, 0\right) \qquad \text{if } z = t-1$$
$$(x, z) \xrightarrow{g} (x + b^z, z)$$



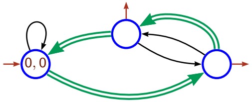
period $p = 3$
remainder set $R = \{1, 2\}$
parameter $(s, t) = (0, 1)$

Use the previous computed parameters to build *the* quotient of $\mathcal{P}_p^R$ associated with (s, t) .

- state set : $\mathbb{Z}/p\mathbb{Z} \times \mathbb{Z}/t\mathbb{Z}$ $\quad 3 \times 1 = 3$ states
- transitions:

$$(x, 0) \xrightarrow{0} (2x, 0)$$

$$(x, 0) \xrightarrow{g} (x + 1, 0)$$



period $p = 3$
remainder set $R = \{1, 2\}$
parameter $(s, t) = (0, 1)$

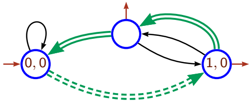
Step 3 – Verifications

Use the previous computed parameters to build *the* quotient of $\mathcal{P}_p^R$ associated with (s, t) .

- state set : $\mathbb{Z}/p\mathbb{Z} \times \mathbb{Z}/t\mathbb{Z}$ $3 \times 1 = 3$ states
- transitions:

$$(x, 0) \xrightarrow{0} (2x, 0)$$

$$(x, 0) \xrightarrow{g} (x + 1, 0)$$



period $p = 3$
remainder set $R = \{1, 2\}$
parameter $(s, t) = (0, 1)$

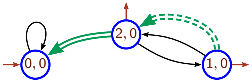
Step 3 – Verifications

Use the previous computed parameters to build *the* quotient of $\mathcal{P}_p^R$ associated with (s, t) .

- state set : $\mathbb{Z}/p\mathbb{Z} \times \mathbb{Z}/t\mathbb{Z}$ $3 \times 1 = 3$ states
- transitions:

$$(x, 0) \xrightarrow{0} (2x, 0)$$

$$(x, 0) \xrightarrow{g} (x + 1, 0)$$



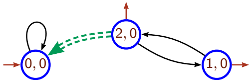
period $p = 3$
remainder set $R = \{1, 2\}$
parameter $(s, t) = (0, 1)$

Use the previous computed parameters to build *the* quotient of $\mathcal{P}_p^R$ associated with (s, t) .

- state set : $\mathbb{Z}/p\mathbb{Z} \times \mathbb{Z}/t\mathbb{Z}$ $\quad 3 \times 1 = 3$ states
- transitions:

$$(x, 0) \xrightarrow{0} (2x, 0)$$

$$(x, 0) \xrightarrow{g} (x + 1, 0)$$



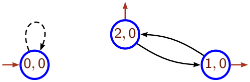
period $p = 3$
remainder set $R = \{1, 2\}$
parameter $(s, t) = (0, 1)$

Use the previous computed parameters to build *the* quotient of $\mathcal{P}_p^R$ associated with (s,t) .

- state set : $\mathbb{Z}/p\mathbb{Z} \times \mathbb{Z}/t\mathbb{Z}$ $3 \times 1 \; = \; 3$ states
- transitions:

$$(x,0) \xrightarrow{0} (2x,0)$$

$$(x,0) \xrightarrow{g} (x+1,0)$$



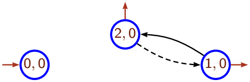
period $p = 3$
remainder set $R = \{1,2\}$
parameter $(s,t) = (0,1)$

Use the previous computed parameters to build *the* quotient of $\mathcal{P}_p^R$ associated with (s, t) .

- state set : $\mathbb{Z}/p\mathbb{Z} \times \mathbb{Z}/t\mathbb{Z}$ $\quad 3 \times 1 = 3$ states
- transitions:

$$(x, 0) \xrightarrow{0} (2x, 0)$$

$$(x, 0) \xrightarrow{g} (x + 1, 0)$$



period $p = 3$
remainder set $R = \{1, 2\}$
parameter $(s, t) = (0, 1)$

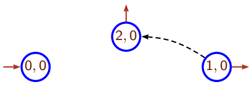
Step 3 – Verifications

Use the previous computed parameters to build *the* quotient of $\mathcal{P}_p^R$ associated with (s, t) .

- state set : $\mathbb{Z}/p\mathbb{Z} \times \mathbb{Z}/t\mathbb{Z}$ $3 \times 1 \ = \ 3$ states
- transitions:

$$(x, 0) \xrightarrow{0} (2x, 0)$$

$$(x, 0) \xrightarrow{g} (x + 1, 0)$$



period $p = 3$
remainder set $R = \{1, 2\}$
parameter $(s, t) = (0, 1)$

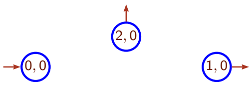
Step 3 – Verifications

Use the previous computed parameters to build *the* quotient of $\mathcal{P}_p^R$ associated with (s, t) .

- state set : $\mathbb{Z}/p\mathbb{Z} \times \mathbb{Z}/t\mathbb{Z}$ $3 \times 1 = 3$ states
- transitions:

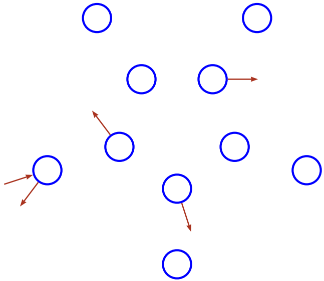
$$(x, 0) \xrightarrow{0} (2x, 0)$$

$$(x, 0) \xrightarrow{g} (x + 1, 0)$$



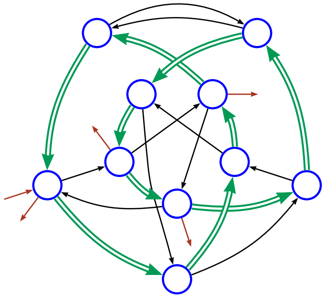
period $p = 3$
remainder set $R = \{1, 2\}$
parameter $(s, t) = (0, 1)$

Another example for base $b = 3$



Changing the alphabet of $\mathcal{A}$: $\{0, 1, 2\} \longrightarrow \{0, g\}$

1. Remove every transition labelled neither by 0 nor 1.
2. Replace every transition by 1 by one labelled by $g = 10^{-1}$.

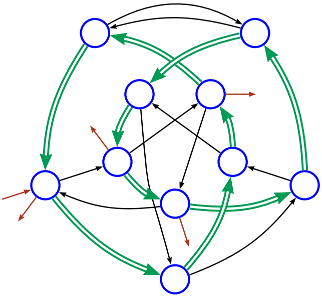


Step 2 – Computation of the parameters

Proposition

If $\mathcal{A}$ is quotient of the Pascal automaton $\mathcal{P}_p^R$; then

- The letter g induces in $\mathcal{A}$ cycles of length p.

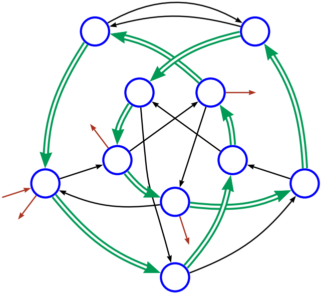


period $p = 5$

Proposition

If $\mathcal{A}$ is quotient of the Pascal automaton $\mathcal{P}_p^R$; then

- The letter g induces in $\mathcal{A}$ cycles of length p.
- The remainder set R contains r iff g^r is accepted by $\mathcal{A}$.



period $p = 5$
remainder set $R = \{0, 3\}$

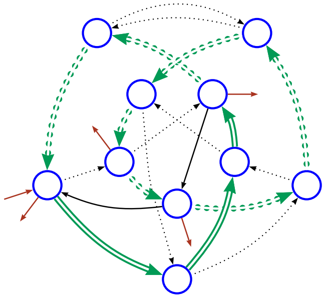
Step 2 – Computation of the parameters

Proposition

If $\mathcal{A}$ is quotient of the Pascal automaton $\mathcal{P}_p^R$; then

- The letter g induces in $\mathcal{A}$ cycles of length p.
- The remainder set R contains r iff g^r is accepted by $\mathcal{A}$.
- The characteristic parameter (s, t) of the quotient is def. by

$$i \xrightarrow{g^s} \xrightarrow{0^t} i$$



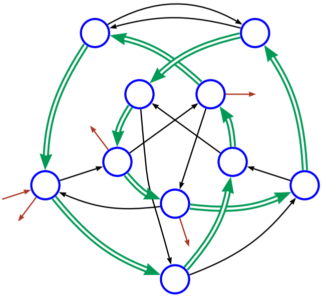
period $p = 5$
remainder set $R = \{0, 3\}$
parameter $(s, t) = (3, 2)$

Step 3 – Verifications

- state set : $\mathbb{Z}/p\mathbb{Z} \times \mathbb{Z}/t\mathbb{Z}$ $\quad 5 \times 2 = 10$ states
- transitions:

$$(x,z) \xrightarrow{0} (x, z+1) \qquad \text{if } z < t-1$$
$$(x,z) \xrightarrow{0} \left(\tfrac{x-s}{b^t}, 0\right) \qquad \text{if } z = t-1$$

$$(x,z) \xrightarrow{g} (x + b^z, z)$$



period $p = 5$

remainder set $R = \{0, 3\}$

parameter $(s, t) = (3, 2)$

$\frac{1}{b^t} \bmod p = \frac{1}{9} \bmod 5 = 4$

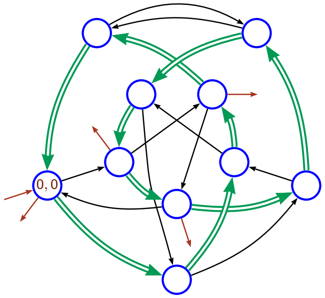
$\frac{x-s}{b^t} = 3 - x$

Step 3 – Verifications

- state set : $\mathbb{Z}/p\mathbb{Z} \times \mathbb{Z}/t\mathbb{Z}$ $\quad 5 \times 2 = 10$ states
- transitions:

$$(x,0) \xrightarrow{0} (x,1)$$
$$(x,1) \xrightarrow{0} (3-x,0)$$

$$(x,0) \xrightarrow{g} (x+1,0)$$
$$(x,1) \xrightarrow{g} (x+3,1)$$



period $p = 5$
remainder set $R = \{0, 3\}$
parameter $(s,t) = (3,2)$

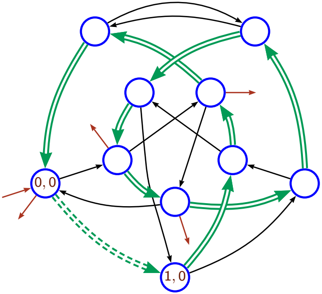
$\frac{1}{b^t} \bmod p = \frac{1}{9} \bmod 5 = 4$
$\frac{x-s}{b^t} = 3 - x$

Step 3 – Verifications

- state set : $\mathbb{Z}/p\mathbb{Z} \times \mathbb{Z}/t\mathbb{Z}$ $\quad 5 \times 2 = 10$ states
- transitions:

$$(x,0) \xrightarrow{0} (x,1)$$
$$(x,1) \xrightarrow{0} (3-x,0)$$

$$(x,0) \xrightarrow{g} (x+1,0)$$
$$(x,1) \xrightarrow{g} (x+3,1)$$



period $p = 5$

remainder set $R = \{0, 3\}$

parameter $(s,t) = (3,2)$

$\frac{1}{b^t} \bmod p = \frac{1}{9} \bmod 5 = 4$

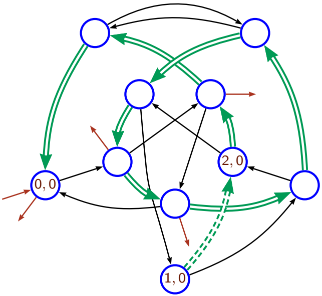
$\frac{x-s}{b^t} = 3 - x$

Step 3 – Verifications

- state set : $\mathbb{Z}/p\mathbb{Z} \times \mathbb{Z}/t\mathbb{Z}$ $\quad 5 \times 2 = 10$ states
- transitions:

$$(x,0) \xrightarrow{0} (x,1)$$
$$(x,1) \xrightarrow{0} (3-x,0)$$

$$(x,0) \xrightarrow{g} (x+1,0)$$
$$(x,1) \xrightarrow{g} (x+3,1)$$



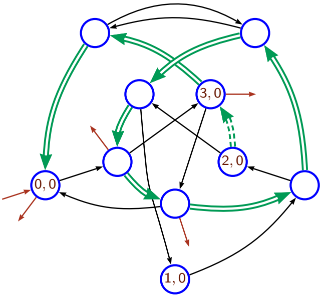
period $p = 5$
remainder set $R = \{0, 3\}$
parameter $(s,t) = (3,2)$

$\frac{1}{b^t} \bmod p = \frac{1}{9} \bmod 5 = 4$
$\frac{x-s}{b^t} = 3 - x$

- state set : $\mathbb{Z}/p\mathbb{Z} \times \mathbb{Z}/t\mathbb{Z}$ $\quad 5 \times 2 = 10$ states
- transitions:

$$(x,0) \xrightarrow{0} (x,1)$$
$$(x,1) \xrightarrow{0} (3-x,0)$$

$$(x,0) \xrightarrow{g} (x+1,0)$$
$$(x,1) \xrightarrow{g} (x+3,1)$$



period $p = 5$
remainder set $R = \{0, 3\}$
parameter $(s, t) = (3, 2)$

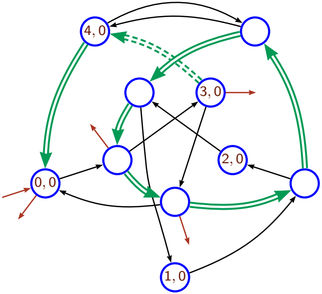
$\frac{1}{b^t} \bmod p = \frac{1}{9} \bmod 5 = 4$
$\frac{x-s}{b^t} = 3 - x$

Step 3 – Verifications

- state set : $\mathbb{Z}/p\mathbb{Z} \times \mathbb{Z}/t\mathbb{Z}$ $\quad 5 \times 2 = 10$ states
- transitions:

$$(x,0) \xrightarrow{0} (x,1)$$
$$(x,1) \xrightarrow{0} (3-x,0)$$

$$(x,0) \xrightarrow{g} (x+1,0)$$
$$(x,1) \xrightarrow{g} (x+3,1)$$



period $p = 5$

remainder set $R = \{0, 3\}$

parameter $(s, t) = (3, 2)$

$\frac{1}{b^t} \bmod p \;=\; \frac{1}{9} \bmod 5 \;=\; 4$

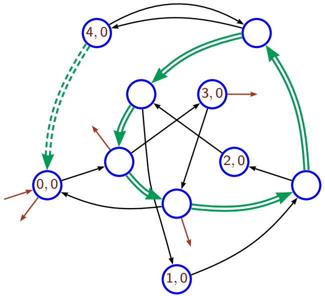
$\frac{x-s}{b^t} = 3 - x$

Step 3 – Verifications

- state set : $\mathbb{Z}/p\mathbb{Z} \times \mathbb{Z}/t\mathbb{Z}$ $\quad 5 \times 2 = 10$ states
- transitions:

$$(x,0) \xrightarrow{0} (x,1)$$
$$(x,1) \xrightarrow{0} (3-x,0)$$

$$(x,0) \xrightarrow{g} (x+1,0)$$
$$(x,1) \xrightarrow{g} (x+3,1)$$



period $p = 5$

remainder set $R = \{0, 3\}$

parameter $(s, t) = (3, 2)$

$\frac{1}{b^t} \bmod p = \frac{1}{9} \bmod 5 = 4$

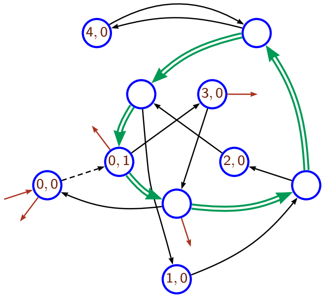
$\frac{x-s}{b^t} = 3 - x$

Step 3 – Verifications

- state set : $\mathbb{Z}/p\mathbb{Z} \times \mathbb{Z}/t\mathbb{Z}$ $\quad 5 \times 2 = 10$ states
- transitions:

$$(x,0) \xrightarrow{0} (x,1)$$
$$(x,1) \xrightarrow{0} (3-x,0)$$

$$(x,0) \xrightarrow{g} (x+1,0)$$
$$(x,1) \xrightarrow{g} (x+3,1)$$



period $p = 5$

remainder set $R = \{0, 3\}$

parameter $(s,t) = (3,2)$

$\frac{1}{b^t} \bmod p = \frac{1}{9} \bmod 5 = 4$

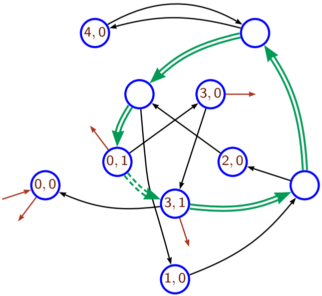
$\frac{x-s}{b^t} = 3 - x$

Step 3 – Verifications

- state set : $\mathbb{Z}/p\mathbb{Z} \times \mathbb{Z}/t\mathbb{Z}$ $\quad 5 \times 2 = 10$ states
- transitions:

$$(x,0) \xrightarrow{0} (x,1)$$
$$(x,1) \xrightarrow{0} (3-x,0)$$

$$(x,0) \xrightarrow{g} (x+1,0)$$
$$(x,1) \xrightarrow{g} (x+3,1)$$



period $p = 5$

remainder set $R = \{0, 3\}$

parameter $(s,t) = (3,2)$

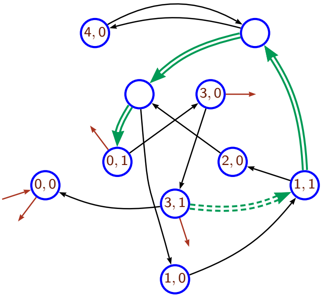
$\frac{1}{b^t} \bmod p = \frac{1}{9} \bmod 5 = 4$

$\frac{x-s}{b^t} = 3 - x$

- state set : $\mathbb{Z}/p\mathbb{Z} \times \mathbb{Z}/t\mathbb{Z}$ $\quad 5 \times 2 = 10$ states
- transitions:

$$(x,0) \xrightarrow{0} (x,1)$$
$$(x,1) \xrightarrow{0} (3-x,0)$$

$$(x,0) \xrightarrow{g} (x+1,0)$$
$$(x,1) \xrightarrow{g} (x+3,1)$$



period $p = 5$
remainder set $R = \{0, 3\}$
parameter $(s, t) = (3, 2)$

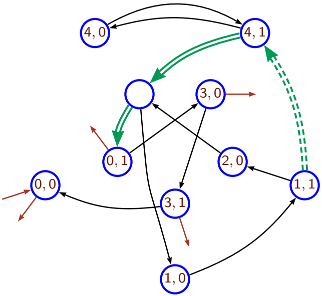
$\frac{1}{b^t} \bmod p = \frac{1}{9} \bmod 5 = 4$
$\frac{x-s}{b^t} = 3 - x$

Step 3 – Verifications

- state set : $\mathbb{Z}/p\mathbb{Z} \times \mathbb{Z}/t\mathbb{Z}$ $\quad 5 \times 2 = 10$ states
- transitions:

$$(x,0) \xrightarrow{0} (x,1)$$
$$(x,1) \xrightarrow{0} (3-x,0)$$

$$(x,0) \xrightarrow{g} (x+1,0)$$
$$(x,1) \xrightarrow{g} (x+3,1)$$



period $p = 5$

remainder set $R = \{0, 3\}$

parameter $(s,t) = (3,2)$

$\frac{1}{b^t} \bmod p = \frac{1}{9} \bmod 5 = 4$

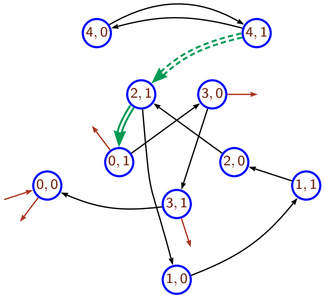
$\frac{x-s}{b^t} = 3 - x$

Step 3 – Verifications

- state set : $\mathbb{Z}/p\mathbb{Z} \times \mathbb{Z}/t\mathbb{Z}$ $\quad 5 \times 2 = 10$ states
- transitions:

$$(x,0) \xrightarrow{0} (x,1)$$
$$(x,1) \xrightarrow{0} (3-x,0)$$

$$(x,0) \xrightarrow{g} (x+1,0)$$
$$(x,1) \xrightarrow{g} (x+3,1)$$



period $p = 5$

remainder set $R = \{0, 3\}$

parameter $(s, t) = (3, 2)$

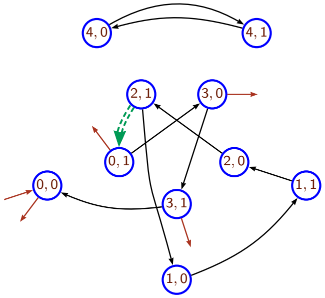
$$\frac{1}{b^t} \bmod p = \frac{1}{9} \bmod 5 = 4$$
$$\frac{x-s}{b^t} = 3 - x$$

Step 3 – Verifications

- state set : $\mathbb{Z}/p\mathbb{Z} \times \mathbb{Z}/t\mathbb{Z}$ $5 \times 2 = 10$ states
- transitions:

$$(x,0) \xrightarrow{0} (x,1)$$
$$(x,1) \xrightarrow{0} (3-x,0)$$

$$(x,0) \xrightarrow{g} (x+1,0)$$
$$(x,1) \xrightarrow{g} (x+3,1)$$



period $p = 5$
remainder set $R = \{0, 3\}$
parameter $(s,t) = (3,2)$

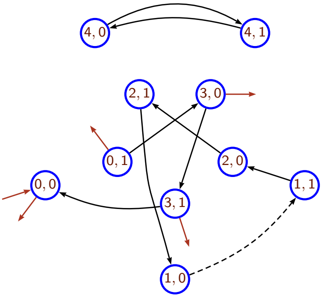
$\frac{1}{b^t} \bmod p = \frac{1}{9} \bmod 5 = 4$
$\frac{x-s}{b^t} = 3 - x$

Step 3 – Verifications

- state set : $\mathbb{Z}/p\mathbb{Z} \times \mathbb{Z}/t\mathbb{Z}$ $\quad 5 \times 2 = 10$ states
- transitions:

$$(x,0) \xrightarrow{0} (x,1)$$
$$(x,1) \xrightarrow{0} (3-x,0)$$

$$(x,0) \xrightarrow{g} (x+1,0)$$
$$(x,1) \xrightarrow{g} (x+3,1)$$



period $p = 5$
remainder set $R = \{0, 3\}$
parameter $(s,t) = (3,2)$

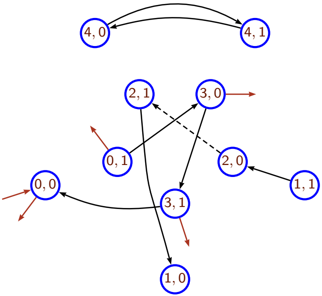
$\frac{1}{b^t} \bmod p = \frac{1}{9} \bmod 5 = 4$
$\frac{x-s}{b^t} = 3 - x$

Step 3 – Verifications

- state set : $\mathbb{Z}/p\mathbb{Z} \times \mathbb{Z}/t\mathbb{Z}$ $\quad 5 \times 2 = 10$ states
- transitions:

$$(x, 0) \xrightarrow{0} (x, 1)$$
$$(x, 1) \xrightarrow{0} (3 - x, 0)$$

$$(x, 0) \xrightarrow{g} (x + 1, 0)$$
$$(x, 1) \xrightarrow{g} (x + 3, 1)$$



period $p = 5$
remainder set $R = \{0, 3\}$
parameter $(s, t) = (3, 2)$

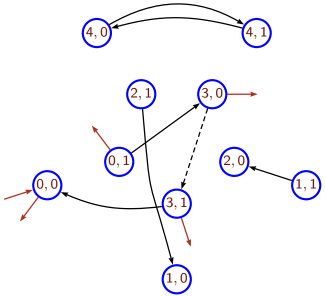
$\frac{1}{b^t} \bmod p = \frac{1}{9} \bmod 5 = 4$
$\frac{x-s}{b^t} = 3 - x$

Step 3 – Verifications

- state set : $\mathbb{Z}/p\mathbb{Z} \times \mathbb{Z}/t\mathbb{Z}$ $5 \times 2 = 10$ states
- transitions:

$$(x,0) \xrightarrow{0} (x,1)$$
$$(x,1) \xrightarrow{0} (3-x,0)$$

$$(x,0) \xrightarrow{g} (x+1,0)$$
$$(x,1) \xrightarrow{g} (x+3,1)$$



period $p = 5$
remainder set $R = \{0, 3\}$
parameter $(s, t) = (3, 2)$

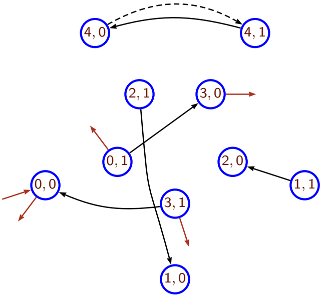
$\frac{1}{b^t} \bmod p = \frac{1}{9} \bmod 5 = 4$
$\frac{x-s}{b^t} = 3 - x$

Step 3 – Verifications

- state set : $\mathbb{Z}/p\mathbb{Z} \times \mathbb{Z}/t\mathbb{Z}$ $\quad 5 \times 2 = 10$ states
- transitions:

$$(x, 0) \xrightarrow{0} (x, 1)$$
$$(x, 1) \xrightarrow{0} (3 - x, 0)$$

$$(x, 0) \xrightarrow{g} (x + 1, 0)$$
$$(x, 1) \xrightarrow{g} (x + 3, 1)$$



period $p = 5$
remainder set $R = \{0, 3\}$
parameter $(s, t) = (3, 2)$

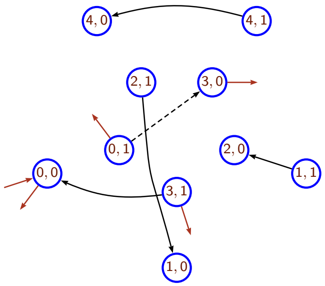
$\frac{1}{b^t} \bmod p = \frac{1}{9} \bmod 5 = 4$
$\frac{x-s}{b^t} = 3 - x$

Step 3 – Verifications

- state set : $\mathbb{Z}/p\mathbb{Z} \times \mathbb{Z}/t\mathbb{Z}$ $\quad 5 \times 2 = 10$ states
- transitions:

$$(x,0) \xrightarrow{0} (x,1)$$
$$(x,1) \xrightarrow{0} (3-x,0)$$

$$(x,0) \xrightarrow{g} (x+1,0)$$
$$(x,1) \xrightarrow{g} (x+3,1)$$



period $p = 5$
remainder set $R = \{0, 3\}$
parameter $(s, t) = (3, 2)$

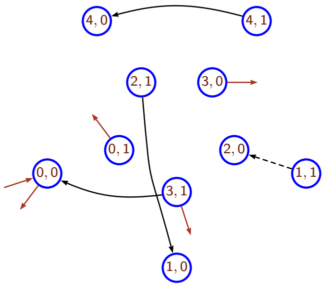
$$\frac{1}{b^t} \bmod p = \frac{1}{9} \bmod 5 = 4$$
$$\frac{x-s}{b^t} = 3 - x$$

Step 3 – Verifications

- state set : $\mathbb{Z}/p\mathbb{Z} \times \mathbb{Z}/t\mathbb{Z}$ $\quad 5 \times 2 \; = \; 10$ states
- transitions:

$$(x,0) \xrightarrow{0} (x,1)$$
$$(x,1) \xrightarrow{0} (3-x,0)$$

$$(x,0) \xrightarrow{g} (x+1,0)$$
$$(x,1) \xrightarrow{g} (x+3,1)$$



period $p = 5$
remainder set $R = \{0, 3\}$
parameter $(s,t) = (3,2)$

$\frac{1}{b^t} \bmod p \; = \; \frac{1}{9} \bmod 5 \; = \; 4$

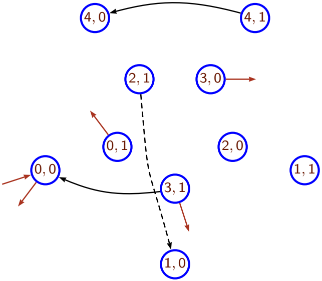
$\frac{x-s}{b^t} = 3 - x$

Step 3 – Verifications

- state set : $\mathbb{Z}/p\mathbb{Z} \times \mathbb{Z}/t\mathbb{Z}$ $\quad 5 \times 2 \; = \; 10$ states
- transitions:

$$(x,0) \xrightarrow{0} (x,1)$$
$$(x,1) \xrightarrow{0} (3-x,0)$$

$$(x,0) \xrightarrow{g} (x+1,0)$$
$$(x,1) \xrightarrow{g} (x+3,1)$$



period $p = 5$

remainder set $R = \{0, 3\}$

parameter $(s, t) = (3, 2)$

$\frac{1}{b^t} \bmod p \; = \; \frac{1}{9} \bmod 5 \; = \; 4$

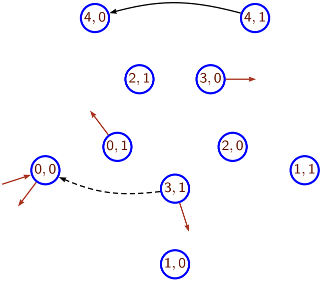
$\frac{x-s}{b^t} = 3 - x$

Step 3 – Verifications

- state set : $\mathbb{Z}/p\mathbb{Z} \times \mathbb{Z}/t\mathbb{Z}$ $5 \times 2 = 10$ states
- transitions:

$$(x,0) \xrightarrow{0} (x,1)$$
$$(x,1) \xrightarrow{0} (3-x,0)$$

$$(x,0) \xrightarrow{g} (x+1,0)$$
$$(x,1) \xrightarrow{g} (x+3,1)$$



period $p = 5$
remainder set $R = \{0, 3\}$
parameter $(s,t) = (3,2)$

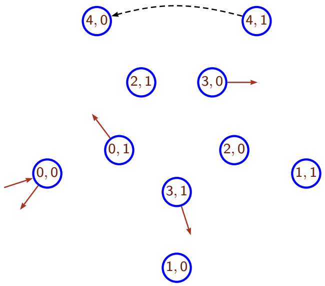
$\frac{1}{b^t} \bmod p = \frac{1}{9} \bmod 5 = 4$
$\frac{x-s}{b^t} = 3 - x$

Step 3 – Verifications

- state set : $\mathbb{Z}/p\mathbb{Z} \times \mathbb{Z}/t\mathbb{Z}$ $\quad 5 \times 2 = 10$ states
- transitions:

$$(x,0) \xrightarrow{0} (x,1)$$
$$(x,1) \xrightarrow{0} (3-x,0)$$

$$(x,0) \xrightarrow{g} (x+1,0)$$
$$(x,1) \xrightarrow{g} (x+3,1)$$



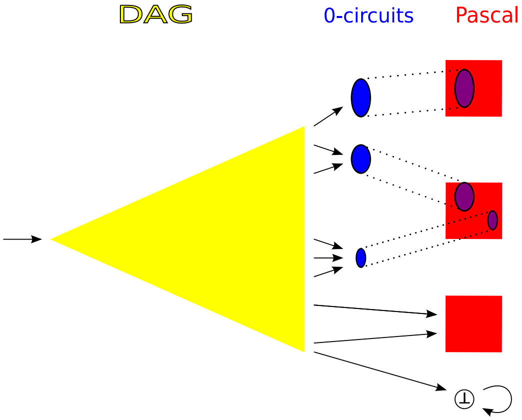
period $p = 5$

remainder set $R = \{0, 3\}$

parameter $(s, t) = (3, 2)$

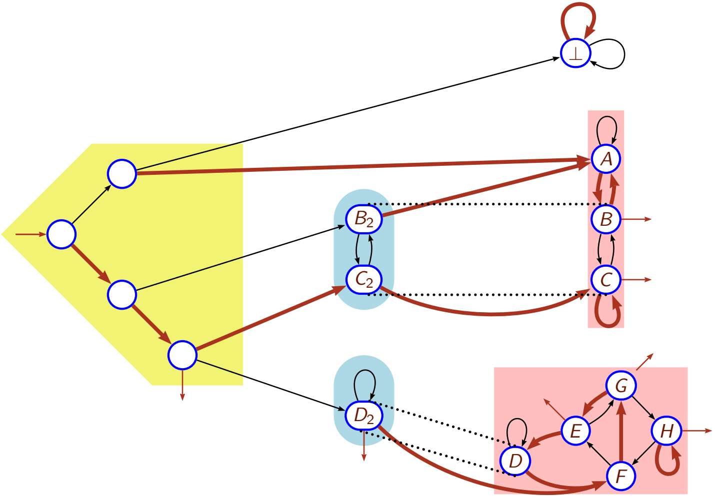
$$\frac{1}{b^t} \bmod p = \frac{1}{9} \bmod 5 = 4$$
$$\frac{x-s}{b^t} = 3 - x$$

Outline



DAG
0-circuits
Pascal
⊥

An automaton satisfying the UP-criterion



Theorem (Linearity)

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

Theorem (Linearity)

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

Theorem (Correctness)

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

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

Theorem (Linearity)

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

Theorem (Correctness)

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

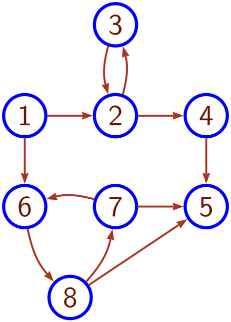
$$\mathcal{A} \text{ satisfies the UP-criterion} \implies L(\mathcal{A}) \text{ is (UP)}.$$

Theorem (Completeness)

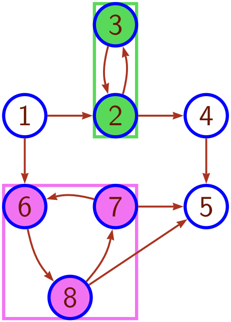
$\mathcal{A}$: a *minimal* DFA.

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

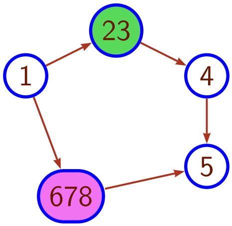
Linear Complexity (1) – Condensation

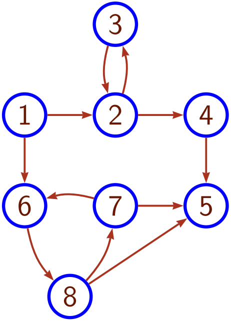


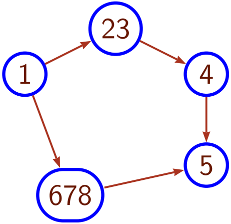
Linear Complexity (1) – Condensation



Linear Complexity (1) – Condensation



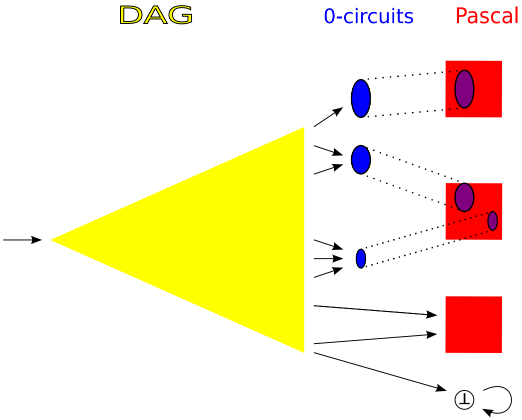




Algorithm (Tarjan, 1972)

Computing the condensation of a graph takes linear time.

Linear Complexity (2)



Verifying that an automaton satisfies the UP-criterion consists in a simple browsing of the automaton and its condensation.

Completeness

Theorem (Completeness)

$\mathcal{A}$: a *minimal* automaton.
$L(\mathcal{A})$ is (UP) $\Rightarrow$ $\mathcal{A}$ satisfies the UP-criterion.

Proof.

Completeness

Theorem (Completeness)

$\mathcal{A}$: a *minimal* automaton.
$L(\mathcal{A})$ is (UP) $\Rightarrow$ $\mathcal{A}$ satisfies the UP-criterion.

Proof.

Proposition 1

Each set of (UP) is accepted by an automaton satisfying the UP-criterion.

Completeness

Theorem (Completeness)

$\mathcal{A}$: a *minimal* automaton.
$L(\mathcal{A})$ is (UP) $\Rightarrow$ $\mathcal{A}$ satisfies the UP-criterion.

Proof.

- Proposition 1 $\Rightarrow$ $\exists\mathcal{B}$ satisfying the UP-crit. and accepting $L(\mathcal{A})$.

Proposition 1

Each set of (UP) is accepted by an automaton satisfying the UP-criterion.

Completeness

Theorem (Completeness)

$\mathcal{A}$: a *minimal* automaton.
$L(\mathcal{A})$ is (UP) $\Rightarrow$ $\mathcal{A}$ satisfies the UP-criterion.

Proof.

- Proposition 1 $\Rightarrow \exists\mathcal{B}$ satisfying the UP-crit. and accepting $L(\mathcal{A})$.
- The minimal automaton is canonical $\Rightarrow$ $\mathcal{A}$ is a quotient of $\mathcal{B}$.

Proposition 1

Each set of (UP) is accepted by an automaton satisfying the UP-criterion.

Completeness

Theorem (Completeness)

$\mathcal{A}$: a *minimal* automaton.
$L(\mathcal{A})$ is (UP) $\Rightarrow$ $\mathcal{A}$ satisfies the UP-criterion.

Proof.

- Proposition 1 $\Rightarrow \exists\mathcal{B}$ satisfying the UP-crit. and accepting $L(\mathcal{A})$.
- The minimal automaton is canonical $\Rightarrow$ $\mathcal{A}$ is a quotient of $\mathcal{B}$.

Proposition 1

Each set of (UP) is accepted by an automaton satisfying the UP-criterion.

Proposition 2

The UP-criterion is stable by quotient.

Completeness

Theorem (Completeness)

$\mathcal{A}$: a *minimal* automaton.
$L(\mathcal{A})$ is (UP) $\Rightarrow$ $\mathcal{A}$ satisfies the UP-criterion.

Proof.

- Proposition 1 $\Rightarrow \exists \mathcal{B}$ satisfying the UP-crit. and accepting $L(\mathcal{A})$.
- The minimal automaton is canonical $\Rightarrow$ $\mathcal{A}$ is a quotient of $\mathcal{B}$.
- Proposition 2 $\Rightarrow$ $\mathcal{A}$ satisfies the UP-criterion.

Proposition 1

Each set of (UP) is accepted by an automaton satisfying the UP-criterion.

Proposition 2

The UP-criterion is stable by quotient.

Completeness – $R \mod b^i$

Expected behaviour

A word u is accepted $\iff \pi(u) \mod b^i$ belongs to R

Completeness – R mod b^i

Expected behaviour

A word u is accepted $\iff \pi(u) \mod b^i$ belongs to R

Remark

In base 10, a word represent a number divisible by 100 if and only if it ends with the factor "00".

Completeness – R mod b^i

Expected behaviour

A word u is accepted $\iff \pi(u) \mod b^i$ belongs to R

Remark

In base 10, a word represent a number divisible by 100 if and only if it starts with the factor "00".

Completeness – R mod b^i

Expected behaviour

A word u is accepted $\iff \pi(u) \mod b^i$ belongs to R

Remark

In base 10, a word represent a number divisible by 100 if and only if it starts with the factor "00".

Lemma

$\pi(u)$ mod b^i belongs to $R \iff u$ have a prefix in $\{\langle r \rangle \mid r \in R\}$

Completeness – R mod b^i

Expected behaviour

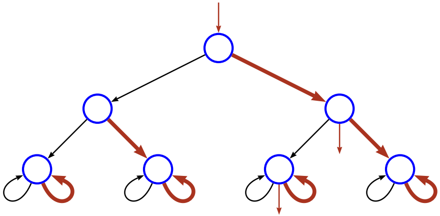
A word u is accepted $\iff$ $\pi(u)$ mod b^i belongs to R

Remark

In base 10, a word represent a number divisible by 100 if and only if it starts with the factor "00".

Lemma

$\pi(u)$ mod b^i belongs to R $\iff$ u have a prefix in $\{\langle r\rangle \mid r \in R\}$



Accepting integers $\equiv 1 \bmod 4$ in base 2.

Completeness – R mod b^i

Expected behaviour

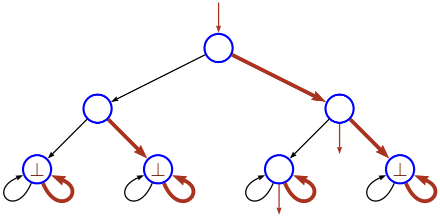
A word u is accepted $\iff$ $\pi(u)$ mod b^i belongs to R

Remark

In base 10, a word represent a number divisible by 100 if and only if it starts with the factor "00".

Lemma

$\pi(u)$ *mod* b^i *belongs to* $R \iff u$ *have a prefix in* $\{\langle r\rangle \mid r \in R\}$



Accepting integers $\equiv 1 \bmod 4$ in base 2.

Completeness – R mod b^i

Expected behaviour

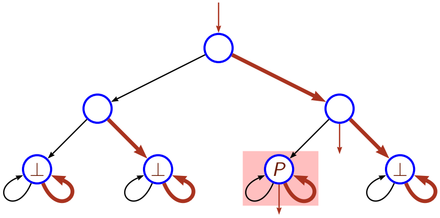
A word u is accepted $\iff$ $\pi(u)$ mod b^i belongs to R

Remark

In base 10, a word represent a number divisible by 100 if and only if it starts with the factor "00".

Lemma

$\pi(u)$ mod b^i belongs to R $\iff$ u have a prefix in $\{\langle r\rangle \mid r \in R\}$



Accepting integers $\equiv 1 \bmod 4$ in base 2.

Completeness – $r \mod (k \times b^i)$

Chinese remainder theorem

$$n \equiv r \mod (k \times b^i) \iff \begin{cases} n \equiv r_k \mod k \\ n \equiv r_d \mod b^i \end{cases}$$

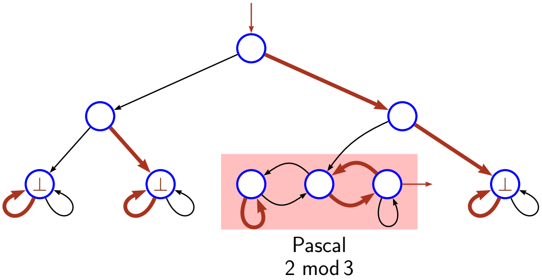
Completeness – $r \bmod (k \times b^i)$

Chinese remainder theorem

$$n \equiv r \bmod (k \times b^i) \iff \begin{cases} n \equiv r_k \bmod k \\ n \equiv r_d \bmod b^i \end{cases}$$

Example: Automaton accepting integers 5 mod 12

$$n \equiv 5 \bmod (3 \times 2^2) \iff \begin{cases} n \equiv 2 \bmod 3 \\ n \equiv 1 \bmod 4 \end{cases}$$



Completeness – $R \bmod (k \times b^i)$

Automaton accepting integers $0, 5, 8 \bmod 12$

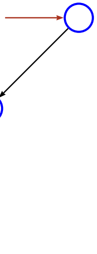
$$
\begin{array}{lclcl}
0 \bmod (3 \times 4) & \iff & 0 \bmod 4 & \text{and} & 0 \bmod 3 \\
5 \bmod (3 \times 4) & \iff & 1 \bmod 4 & \text{and} & 2 \bmod 3 \\
8 \bmod (3 \times 4) & \iff & 0 \bmod 4 & \text{and} & 2 \bmod 3
\end{array}
$$

Completeness – $R \bmod (k \times b^i)$

Automaton accepting integers 0, 5, 8 mod 12

$$0 \bmod (3 \times 4) \iff 0 \bmod 4 \quad \text{and} \quad 0 \bmod 3$$
$$5 \bmod (3 \times 4) \iff 1 \bmod 4 \quad \text{and} \quad 2 \bmod 3$$
$$8 \bmod (3 \times 4) \iff 0 \bmod 4 \quad \text{and} \quad 2 \bmod 3$$

$$0, 5, 8 \bmod 12 \iff \begin{cases} 0 \bmod 4 \quad \text{and} \quad 0,2 \bmod 3 \\ \end{cases}$$

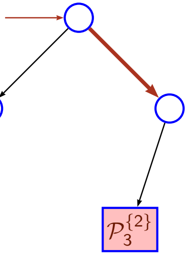


Completeness – $R \bmod (k \times b^i)$

Automaton accepting integers $0, 5, 8 \bmod 12$

$$
\begin{array}{lclcl}
0 \bmod (3 \times 4) & \iff & 0 \bmod 4 & \text{and} & 0 \bmod 3 \\
5 \bmod (3 \times 4) & \iff & 1 \bmod 4 & \text{and} & 2 \bmod 3 \\
8 \bmod (3 \times 4) & \iff & 0 \bmod 4 & \text{and} & 2 \bmod 3
\end{array}
$$

$$
0, 5, 8 \bmod 12 \iff \begin{cases} 0 \bmod 4 & \text{and} \quad 0{,}2 \bmod 3 \\ 1 \bmod 4 & \text{and} \quad 2 \bmod 3 \end{cases}
$$

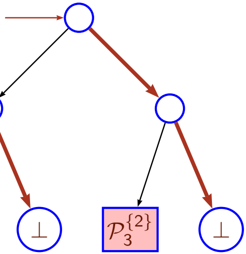


Completeness – $R \bmod (k \times b^i)$

Automaton accepting integers $0, 5, 8 \bmod 12$

$$\begin{array}{llll} 0 \bmod (3 \times 4) & \iff & 0 \bmod 4 & \text{and} & 0 \bmod 3 \\ 5 \bmod (3 \times 4) & \iff & 1 \bmod 4 & \text{and} & 2 \bmod 3 \\ 8 \bmod (3 \times 4) & \iff & 0 \bmod 4 & \text{and} & 2 \bmod 3 \end{array}$$

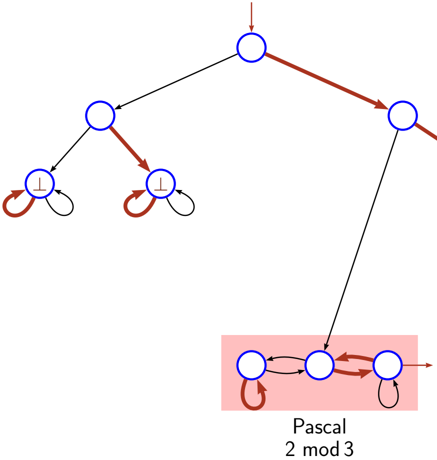
$$0, 5, 8 \bmod 12 \iff \begin{cases} 0 \bmod 4 & \text{and} \quad 0{,}2 \bmod 3 \\ 1 \bmod 4 & \text{and} \quad 2 \bmod 3 \end{cases}$$



Completeness – R mod $(k \times b^i)$ and greater than m

Example

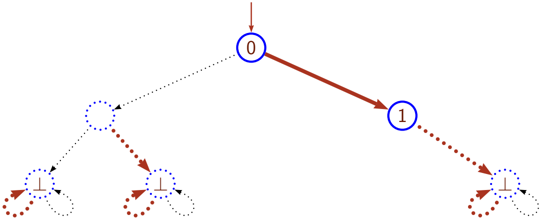
Automaton accepting integers $n > 5$ and $n \equiv 5 \bmod (3 \times 2^2)$

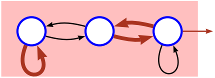


Completeness – R mod $(k \times b^i)$ and greater than m

Example

Automaton accepting integers $n > 5$ and $n \equiv\ 5 \bmod (3 \times 2^2)$



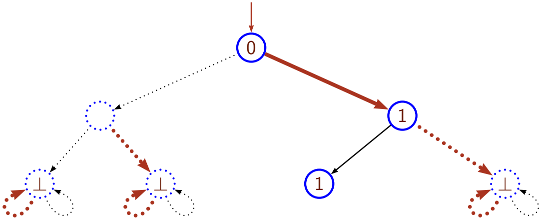


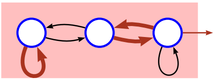
Pascal
2 mod 3

Completeness – R mod $(k \times b^i)$ and greater than m

Example

Automaton accepting integers $n > 5$ and $n \equiv 5 \bmod (3 \times 2^2)$

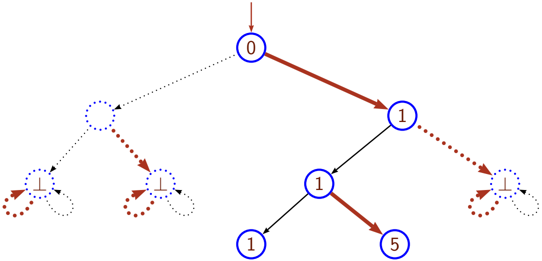


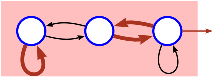


Pascal
2 mod 3

Example

Automaton accepting integers $n > 5$ and $n \equiv 5 \bmod (3 \times 2^2)$



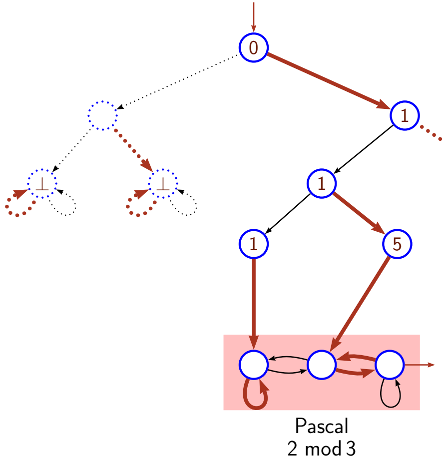


Pascal
2 mod 3

Completeness – R mod $(k \times b^i)$ and greater than m

Example

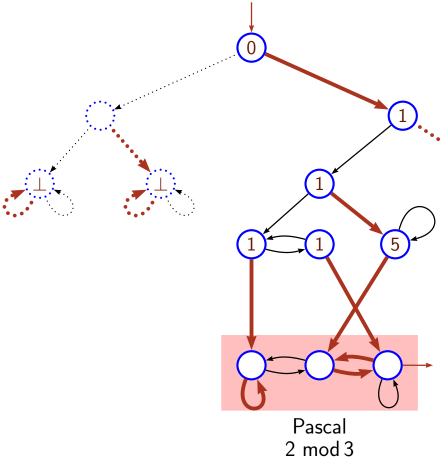
Automaton accepting integers $n > 5$ and $n \equiv 5 \bmod (3 \times 2^2)$



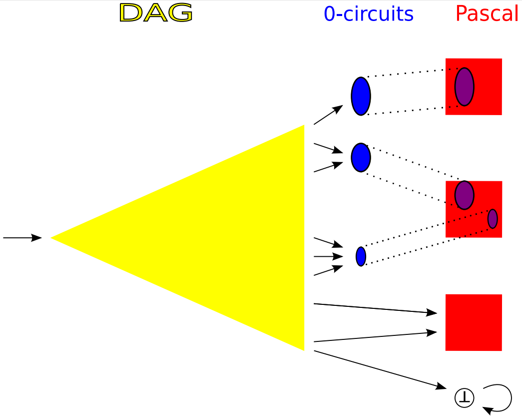
Completeness – R mod $(k \times b^i)$ and greater than m

Example

Automaton accepting integers $n > 5$ and $n \equiv 5 \bmod (3 \times 2^2)$



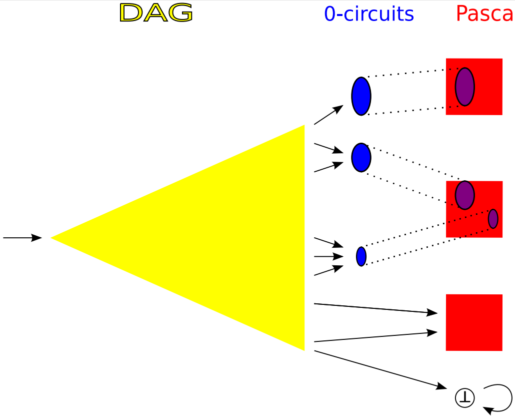
Completeness – Summary



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

- preperiod
- period
- remainder set

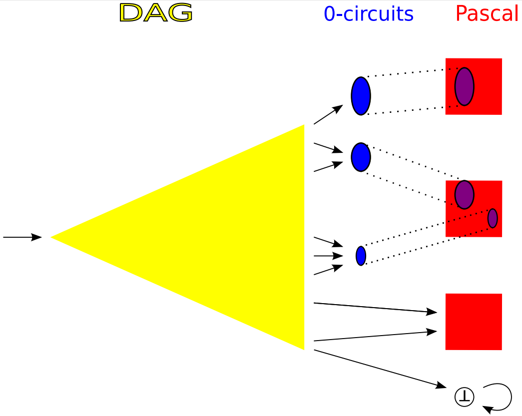
Completeness – Summary



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 depth.
- remainder set

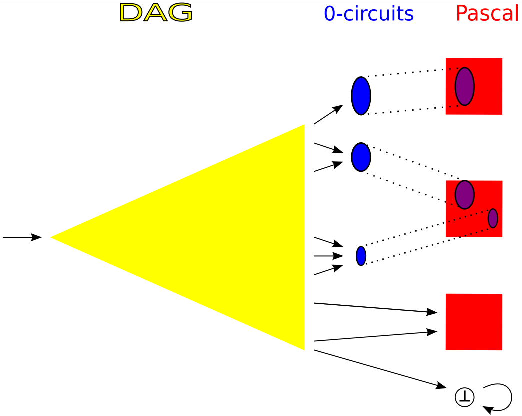
Completeness – Summary



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

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

Completeness – Summary



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

- preperiod ⇒ 0-circuits & DAG depth.
- period ⇒ Pascal's period & DAG depth.
- remainder set ⇒ # of Pascal's & Pascal's remainder sets

Proof of Correctness on an example (1)

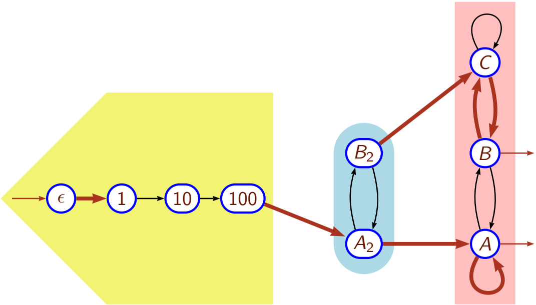
Theorem (Correctness)

$\mathcal{A}$: a *minimal* automaton.
$\mathcal{A}$ satisfies the UP-criterion $\Rightarrow L(\mathcal{A})$ is (UP).

WLOG we can assume that, in $\mathcal{A}$,

- the only final states are in the Pascal automata
  [since other final states accepts only finitely many integer];
- the DAG-part is a simple path
  [since the union of (UP) sets is a (UP) set].

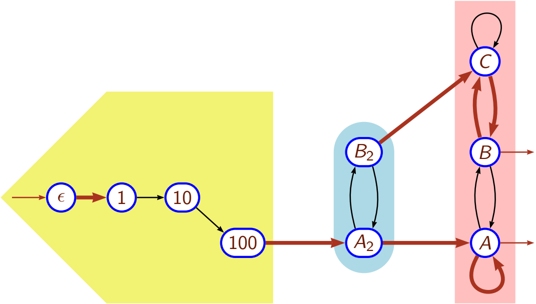
Proof of Correctness on an example (2)



Lemma (isotropism)

Changing the initial state of a Pascal automaton $\mathcal{P}_p^R$ yields a Pascal automaton $\mathcal{P}_p^S$ with the same period p but a different remainder S.

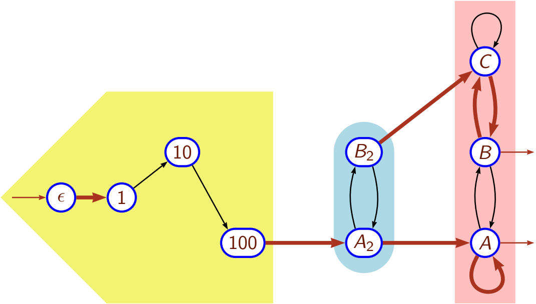
Proof of Correctness on an example (2)



Lemma (isotropism)

Changing the initial state of a Pascal automaton $\mathcal{P}_p^R$ yields a Pascal automaton $\mathcal{P}_p^S$ with the same period p but a different remainder S.

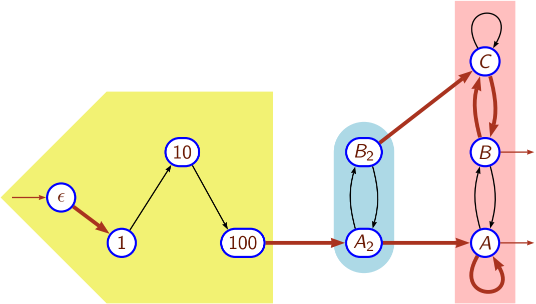
Proof of Correctness on an example (2)



Lemma (isotropism)

Changing the initial state of a Pascal automaton $\mathcal{P}_p^R$ yields a Pascal automaton $\mathcal{P}_p^S$ with the same period p but a different remainder S.

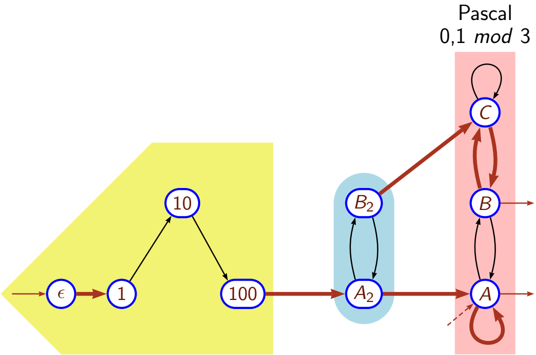
Proof of Correctness on an example (2)



Lemma (isotropism)

Changing the initial state of a Pascal automaton $\mathcal{P}_p^R$ yields a Pascal automaton $\mathcal{P}_p^S$ with the same period p but a different remainder S.

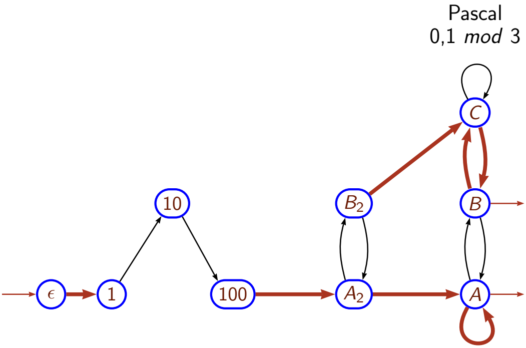
Proof of Correctness on an example (2)



Lemma (isotropism)

Changing the initial state of a Pascal automaton $\mathcal{P}_p^R$ yields a Pascal automaton $\mathcal{P}_p^S$ with the same period p but a different remainder S.

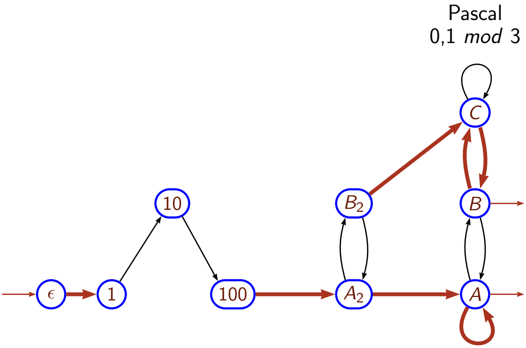
Proof of Correctness on an example (2)



A word u is accepted if and only if

u reaches the Pascal aut. and $\pi(u) \equiv 0{,}1 \bmod 3$.

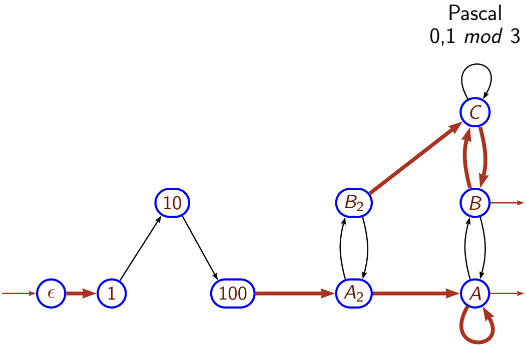
Proof of Correctness on an example (2)



A word u is accepted if and only if

$$u \text{ starts with } 10010^i1 \quad \text{and} \quad \pi(u) \equiv 0,1 \bmod 3.$$

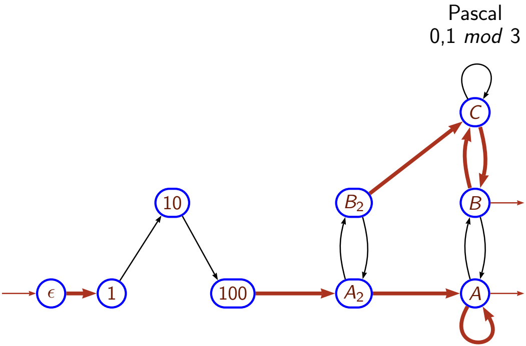
Proof of Correctness on an example (2)



A word u is accepted if and only if

$$\underbrace{u \text{ starts with } 10010^i1}_{\iff\ u > 16 \text{ and } \pi(u) \equiv 9\ (16)} \quad \text{and} \quad \pi(u) \equiv 0{,}1 \bmod 3.$$

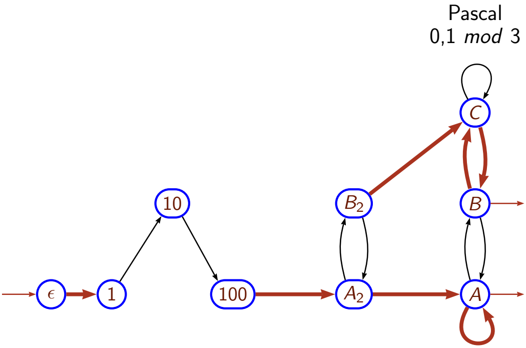
Proof of Correctness on an example (2)



A word u is accepted if and only if

$$\pi(u) > 16 \quad \text{and} \quad \pi(u) \equiv 9 \bmod 16 \quad \text{and} \quad \pi(u) \equiv 0{,}1 \bmod 3.$$

Proof of Correctness on an example (2)



A word u is accepted if and only if

$$\pi(u) > 16 \quad \text{and} \quad \pi(u) \equiv 9, 25 \bmod 48.$$

Outline

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])
  → In nice cases (Fibonacci, Tribonacci, etc), it may yield a P-TIME algorithm.