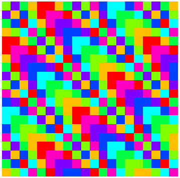
On the amortized cost of an odometer

V. Berthé (LIRMM/CNRS), C. Frougny (LIAFA/Univ. Paris 8),
M. Rigo (Univ. Liège), J. Sakarovitch (ENST/CNRS)

Work in progress...
Journées de Numération – TU Graz – 20th April 2007



So far, different aspects of odometers have been studied :

Combinatorial, Metrical, Topological, Dynamics,
Sequential properties, ...

G. Barat, T. Downarowicz, C. Frougny, P. Grabner, P. Liardet,
R. Tichy, A. M. Vershik, ...

OUR MAIN QUESTION

What is the cost / complexity in average for computing the odometer (i.e., successor map) on finite words, e.g. on integer representations ?

$$\begin{array}{ccccc} n & \longrightarrow & \mathrm{rep}(n) & \in \Sigma^* \\ \downarrow & & \downarrow & \\ n+1 & \longrightarrow & \mathrm{rep}(n+1) & \in \Sigma^* \end{array}$$

WHERE DOES IT COME FROM ?

WORDS'05

E. Barcucci, R. Pinzani, M. Poneti, *Exhaustive generation of some regular languages by using numeration systems*.

For numeration systems built on some linear recurrent sequences of order 2, the "amortized cost" for computing $\mathrm{rep}(n+1)$ from $\mathrm{rep}(n)$ is bounded by a constant (CAT).

J. SAKAROVITCH, ELTS. DE THÉORIE DES AUTOMATES'03

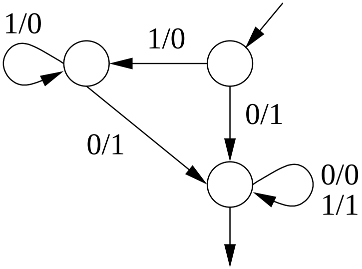
For any rational set R of A^*, the odometer on R is a synchronized function.

i.e., letter-to-letter (left or right) finite transducer with a terminal function appending values of the form (u, ε) or (ε, v)

More than synchronized functions, we will often assume that we have a (right) sequential transducer to do the computation.

A transducer T is sequential if

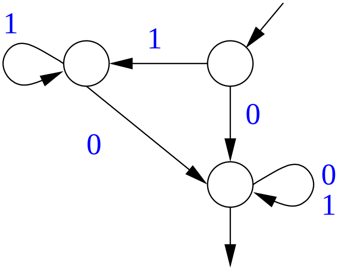
- T has a unique initial state,
- the underlying input automaton is deterministic.



More than synchronized functions, we will often assume that we have a (right) sequential transducer to do the computation.

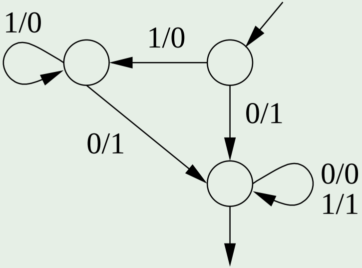
A transducer T is sequential if

- T has a unique initial state,
- the underlying input automaton is deterministic.



Usual binary system

A (TRIVIAL) SEQUENTIAL FUNCTION



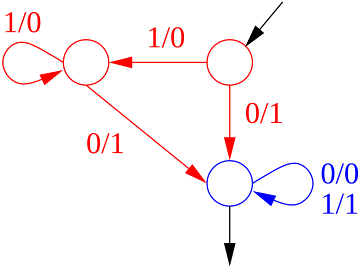
10100111
10101000

DEFINITION (COST)

We define the cost for computing rep$(n+1)$ from rep(n) as

- the position up to where the carry propagates, or
- the length of the path lying in the "transient part",
- for an integer base system, the number of changed digits.

1010 0111
1010 1000

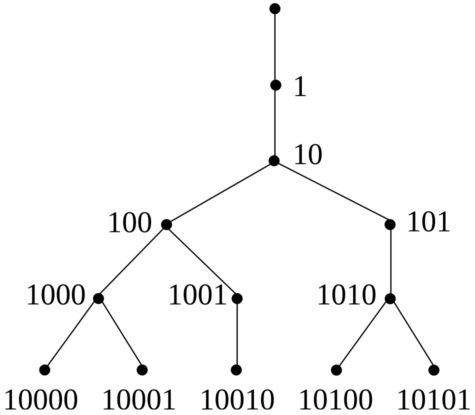


1
10
11
100
101
110
111
1000
1001
1010
1011

Alternative definition (Cost)

Another interpretation for the cost in the lexicographic tree :

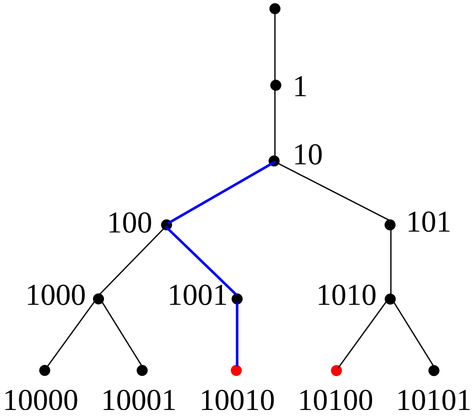
- half of the distance between rep(n) and rep($n+1$)
- distance to the common ancestor of rep(n) and rep($n+1$)



Alternative definition (Cost)

Another interpretation for the cost in the lexicographic tree :

- half of the distance between rep(n) and rep($n+1$)
- distance to the common ancestor of rep(n) and rep($n+1$)



So, cost can be expressed mainly on words

$$uav \longrightarrow ubv', \quad a \neq b, \quad |v| = |v'|$$

$$\text{cost} = |av|$$

Let us introduce a different notion (computational aspects)

DEFINITION (COMPLEXITY)

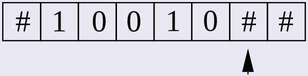
The (algorithmic) complexity for computing rep($n+1$) from rep(n) is the minimum number of operations required to perform this computation (in the sense of a Turing machine).

Remark

Consider a numeration system such that the odometer can be realized by a letter-to-letter (right) sequential transducer.

In that case, the cost is equal to the (algorithmic) complexity.

Indeed, it is not possible to do less computations,
the Turing machine at least has to read the digits up to where the carry propagates



"cost $\leq$ complexity"

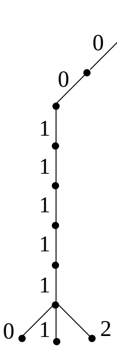
COST ≠ COMPLEXITY

$X^2 - 3X + 1$, $\beta = \frac{3+\sqrt{5}}{2}$, $d_\beta(1) = 21^\omega$, $(U_n)_{n\geq 0} = 1, 3, 8, 21, \ldots$

$\mathrm{rep}(\mathbb{N}) = \{\varepsilon, 1, 2, 10, 11, 12, 20, 21, 100, 101, 102, \ldots\}$

forbidden factors : $2\,1^*\,2$

100111111 → 100111112 but 102111111 → 110000000



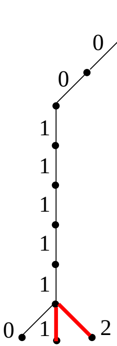
Cost $\neq$ Complexity

$X^2 - 3X + 1$, $\beta = \frac{3+\sqrt{5}}{2}$, $d_\beta(1) = 21^\omega$, $(U_n)_{n\geq 0} = 1, 3, 8, 21, \ldots$

$\text{rep}(\mathbb{N}) = \{\varepsilon, 1, 2, 10, 11, 12, 20, 21, 100, 101, 102, \ldots\}$

forbidden factors : $2\,1^*\,2$

100111111 → 100111112 but 102111111 → 110000000



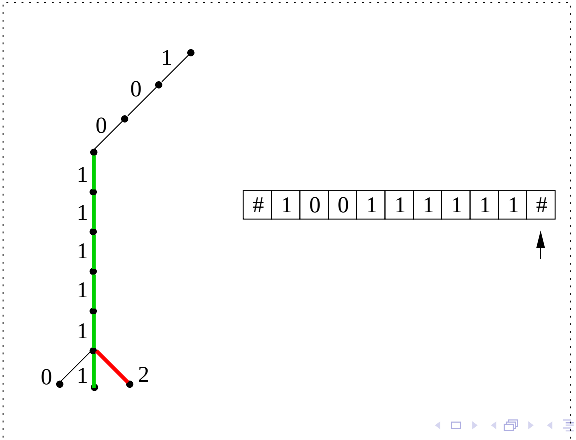
COST $\neq$ COMPLEXITY

$X^2 - 3X + 1$, $\beta = \frac{3+\sqrt{5}}{2}$, $d_\beta(1) = 21^\omega$, $(U_n)_{n\geq 0} = 1, 3, 8, 21, \ldots$

$\text{rep}(\mathbb{N}) = \{\varepsilon, 1, 2, 10, 11, 12, 20, 21, 100, 101, 102, \ldots\}$

forbidden factors : $2\,1^*\,2$

$100111111 \rightarrow 100111112$ but $102111111 \rightarrow 110000000$



All this work on cost/complexity can be done in a general setting

Definition

An abstract numeration system is a triple $S = (L, A, <)$ where L is a infinite (rational) language over a totally ordered alphabet $(A, <)$.

The representation of $n \in \mathbb{N}$ is the $(n+1)$-st word in the genealogically (i.e., radix) ordered language L.

Example

$L = \{(ab), (ac)\}^*$, $a < b < c$

0	1	2	3	4	5	6	7	$\cdots$
ε	ab	ac	$abab$	$abac$	$acab$	$acac$	$ababab$	$\cdots$

Example continues... Amortized cost / complexity

$$\varepsilon \underset{1}{\rightarrow} 1 \underset{2}{\rightarrow} 10 \underset{1}{\rightarrow} 11 \underset{3}{\rightarrow} 100 \underset{1}{\rightarrow} 101 \underset{2}{\rightarrow} 110 \underset{1}{\rightarrow} 111 \underset{4}{\rightarrow}$$

In base k, k^n words from ε to $\overbrace{(k-1)\cdots(k-1)}^{n}$,

$$\frac{k^n + k^{n-1} + \cdots + 1}{k^n} = \frac{k - k^{-n}}{k-1} \rightarrow \frac{k}{k-1}, \text{ as } n \rightarrow \infty$$

Definition (Amortized cost)

$$\lim_{n\to\infty} \left(\sum_{w \in \mathcal{L}, |w| \leq n} \text{cost}(w) \right) \Big/ \#\{w \in \mathcal{L} : |w| \leq n\}$$

Same for amortized complexity

Example continues... Amortized cost / complexity

$$\varepsilon \;\underset{1}{\to}\; 1 \;\underset{2}{\to}\; 10 \;\underset{1}{\to}\; 11 \;\underset{3}{\to}\; 100 \;\underset{1}{\to}\; 101 \;\underset{2}{\to}\; 110 \;\underset{1}{\to}\; 111 \;\underset{4}{\to}$$

In base k, k^n words from ε to $\overbrace{(k-1)\cdots(k-1)}^{n}$,

$$\frac{k^n + k^{n-1} + \cdots + 1}{k^n} = \frac{k - k^{-n}}{k-1} \to \frac{k}{k-1}, \text{ as } n \to \infty$$

Definition (Amortized cost)

$$\lim_{n\to\infty} \left(\sum_{w \in \mathcal{L}, |w| \le n} \text{cost}(w) \right) \Big/ \#\{w \in \mathcal{L} : |w| \le n\}$$

Same for amortized complexity

First exercise...

For Fibonacci system...

$$\varepsilon \underset{1}{\rightarrow} 1 \underset{2}{\rightarrow} 10 \underset{3}{\rightarrow} 100 \underset{1}{\rightarrow} 101 \underset{4}{\rightarrow} 1000 \underset{2}{\rightarrow} 1010 \underset{5}{\rightarrow}$$

$$\text{amortized cost} = \text{amortized complexity} \rightarrow \frac{\tau}{\tau - 1} \simeq 2.618$$

First exercise...

For Fibonacci system...

$$\begin{array}{cccccccc} \varepsilon & \rightarrow 1 & \rightarrow 10 & \rightarrow 100 & \rightarrow 101 & \rightarrow 1000 & \rightarrow 1010 & \rightarrow \\ & 1 & 2 & 3 & 1 & 4 & 2 & 5 \end{array}$$

$$\text{amortized cost} = \text{amortized complexity} \rightarrow \frac{\tau}{\tau - 1} \simeq 2.618$$

Theorem

Let L be a rational language having $\mathcal{M}$ as trim minimal automaton.

If the adjacency matrix M of $\mathcal{M}$ is primitive with $\beta > 1$ as dominating Perron eigenvalue and if all states of $\mathcal{M}$ are final, then the amortized cost of the odometer on L is $\frac{\beta}{\beta-1}$.

Remark

- If the corresponding transducer is right sequential, then this is exactly the amortized (algorithmic) complexity.
- Otherwise, we get information on the average position up to where some change can occur. (More ?)

Remark

All states final means L is prefix closed.

PERRON THEORY

Let M be a $d \times d$ primitive matrix having $\beta > 1$ as dominating eigenvalue. The following holds

$$\forall i, j \in \{0, \dots, d-1\}, \exists c_{ij} > 0 \; : \; (M^n)_{ij} = c_{ij}\, \beta^n + o(\beta^n).$$

If $\mathbf{x}$ (resp. $\mathbf{y}$) is a left $1 \times d$ (resp. right $d \times 1$) eigenvector of M of eigenvalue β such that $\mathbf{x}.\mathbf{y} = 1$ then $\forall 0 \leq i, j < d$,

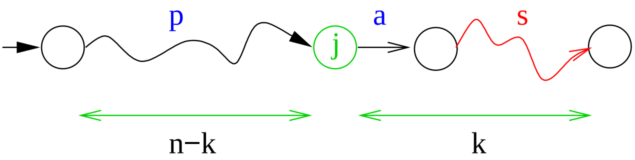
$$c_{ij} = \mathbf{y}_i\, \mathbf{x}_j, \quad i.e., \lim_{n \to \infty} \frac{M^n}{\beta^n} = \mathbf{y}.\mathbf{x}.$$

If $w = pas$ is such that

- $q_0.p = q_j$,
- $a \neq \max A_{q_j}$
- $s \in \max(L_{q_0.pa})$

Fix $q_j \in Q$

$$pas \longrightarrow pbt, \quad |s| = |t|$$



$$\sum_{k=0}^{n-1} (M^{n-k})_{0j} \, (\deg^+(q_j) - 1) \, k .$$

Then sum over Q...

Applications

Result

Let $\beta > 1$ be a Parry number. The amortized cost of the odometer for the canonical linear numeration system associated with β is $\beta/(\beta - 1)$.

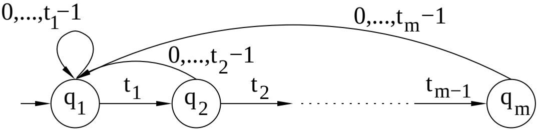
Same remark : cost = complexity when assuming that the odometer is realized with a right sequential transducer.

C. Frougny '97

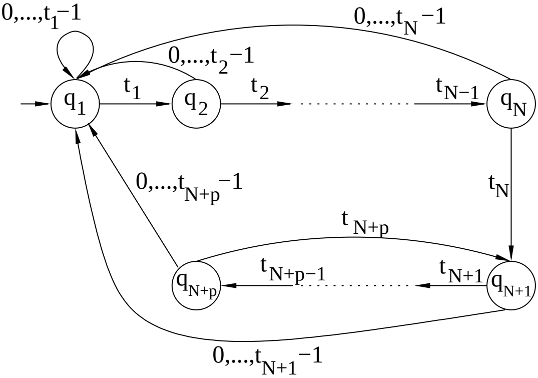
For such β-numeration systems (β being a Parry number), we have

- a right sequential transducer in the finite type,
- but NOT in the sofic case.

simple Parry number



non-simple case



APPLICATIONS

RESULT

Let $S = (L, A, <)$ be an abstract numeration system built on a rational language whose trim minimal automaton $\mathcal{M}$ is primitive and has only final states. If β is the dominating eigenvalue of $\mathcal{M}$ then the amortized cost of the odometer for S is $\beta/(\beta - 1)$.

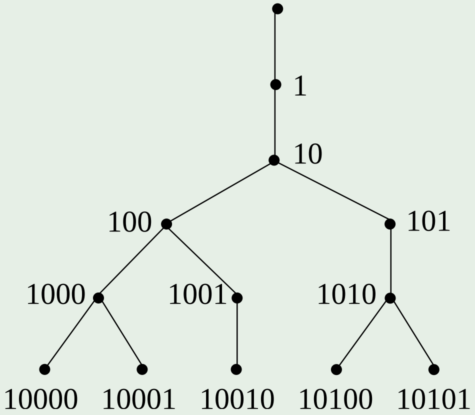
Same remark : cost = complexity when assuming that the odometer is realized with a right sequential transducer.

NEXT STEP, EASY TO HANDLE

Consider several primitive strongly connected components...

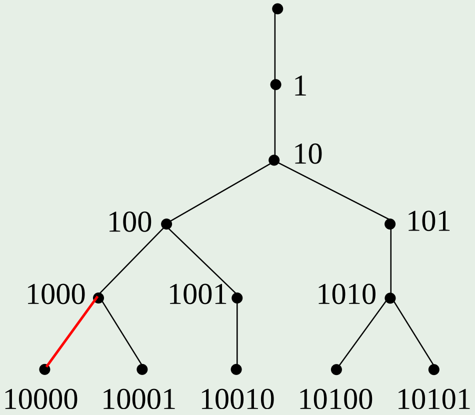
Let's have a look at the lexicographic tree

Fibonacci words of length 5



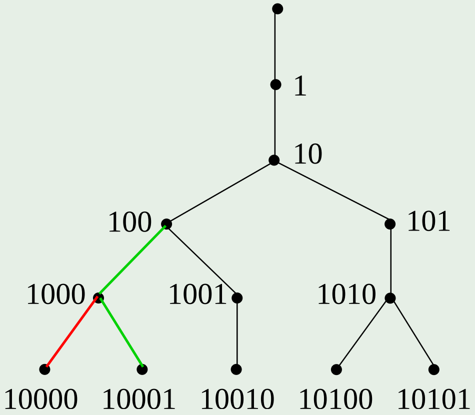
Let's have a look at the lexicographic tree

FIBONACCI WORDS OF LENGTH 5



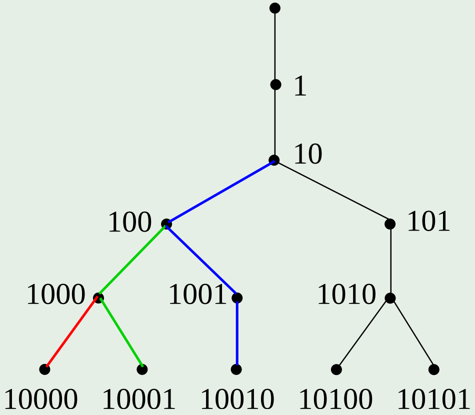
Let's have a look at the lexicographic tree

FIBONACCI WORDS OF LENGTH 5



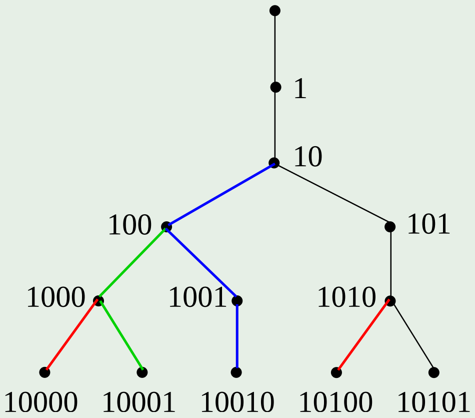
Let's have a look at the lexicographic tree

Fibonacci words of length 5



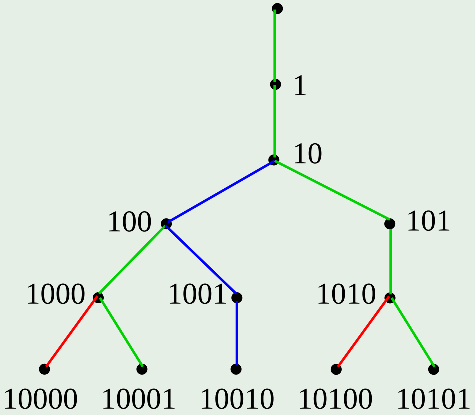
Let's have a look at the lexicographic tree

Fibonacci words of length 5



Let’s have a look at the lexicographic tree

Fibonacci words of length 5



Consequently, the total cost for all words of length n is

$$\mathcal{C}_n := \#\{\text{edges in } T_n\} + 1 = \#\{\text{leaves in } T_n\} = \#(L \cap \Sigma^{\leq n})$$

"Nice" hypothesis :

- L is a prefix closed language ($uv \in L \Rightarrow u \in L$)
- Any branch in the tree is infinite

Remark

If $u_L : n \mapsto \#(L \cap \Sigma^n)$ has a "nice asymptotic behavior", then the amortized cost can be computed...

$$\lim_{n\to\infty} \frac{\sum_{i=0}^{n} \mathcal{C}_i}{\#(L \cap \Sigma^{\leq n})} = \lim_{n\to\infty} \frac{\sum_{i=0}^{n} \sum_{k=0}^{i} u_L(k)}{\sum_{i=0}^{n} u_L(i)}$$

Question

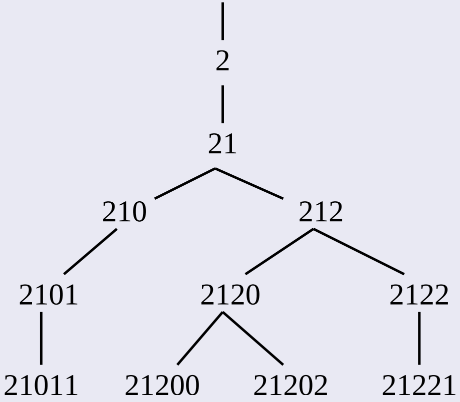
Can we compute the amortized complexity if there is no sequential transducer behind ?

If L is rational, when can the odometer be computed with a right sequential transducer ? (local automaton)

p/q-base (S. Akiyama, C. Frougny, J. Sakarovitch)

$p > q \geq 1$ coprime integers,

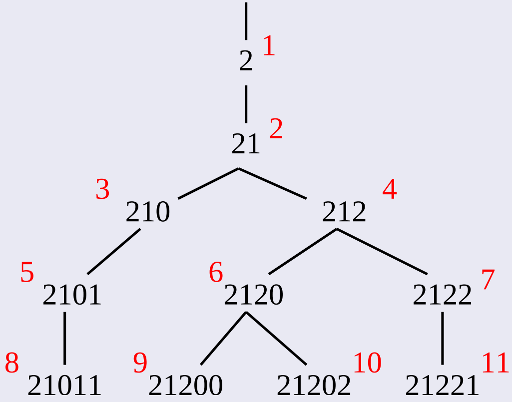
$$N = \sum_{i=0}^{k} \frac{a_i}{q} \left(\frac{p}{q}\right)^i, \quad 0 \leq a_i < p$$



p/q-BASE (S. AKIYAMA, C. FROUGNY, J. SAKAROVITCH)

$p > q \geq 1$ coprime integers,

$$N = \sum_{i=0}^{k} \frac{a_i}{q} \left(\frac{p}{q}\right)^i, \quad 0 \leq a_i < p$$



The language of numeration is “highly” non-rational :
any two sub-trees of the lexicographic tree are non-isomorphic
but it is easy to build a digit-to-digit right sequential transducer that realizes the odometer

PROPOSITION

For the base p/q system, $p > q \geq 1$, the amortized cost (resp. complexity) is

$$\frac{\frac{p}{q}}{\frac{p}{q} - 1}$$

Example of language with zero entropy

a^*b^* is a rational, prefix closed language (and any branch in the lexicographic tree is infinite)

$u(n) = \#(a^*b^* \cap \{a,b\}^n) = n+1$, therefore

$$\lim_{n\to\infty} \frac{\sum_{i=0}^{n} \#(L \cap \Sigma^{\leq i})}{\#(L \cap \Sigma^{\leq n})} = \lim_{n\to\infty} \frac{\frac{1}{6}(n+1)(n+2)(n+3)}{\frac{1}{2}(n+1)(n+2)} = +\infty.$$