On the sequentiality of the successor function

Christiane Frougny

Université Paris 8
and
L.I.A.F.A. (LITP)
4 place Jussieu, 75252 Paris Cedex 05, France
Email: `Christiane.Frougny@litp.ibp.fr`

Running head: On the successor function.

Proofs to be sent to:
Prof. Christiane Frougny
L.I.A.F.A. (LITP)
4 place Jussieu
75252 Paris Cedex 05, France.

Abstract

Let U be a strictly increasing sequence of integers. By a greedy algorithm, every nonnegative integer has a greedy U-representation. The successor function maps the greedy U-representation of N onto the greedy U-representation of $N+1$. We characterize the sequences U such that the successor function associated to U is a left, resp. a right sequential function. We also show that the odometer associated to U is continuous if and only if the successor function is right sequential.

1 Introduction

It is well known that, in the classical K-ary number system, where K is an integer ≥ 2, the successor function, which maps the K-representation of N onto that of $N+1$, is computable by a sequential finite 2-tape automaton (that is to say, deterministic on inputs) working from right to left but not from left to right (there is a carry which propagates from right to left). In Computer Arithmetic, on-line arithmetic consists in performing operations in Most Significant Digit First mode (i.e. from left to right), digit serially after a certain delay of latency (see [Er84]). This mode of doing allows pipelining different operations such as addition, multiplication and division. To be able to perform on-line addition in integer base K, it is necessary to use a redundant number system such as the Avizienis signed-digit representation [Av61], which consists in changing the digit set. Instead of taking digits from the canonical set $\{0, \cdots, K-1\}$, they are taken from a balanced set of the form $\{\bar{a}, \cdots, a\}$, where $\bar{a}$ denotes the digit $-a$, a being an integer such that $a+1 \leq K \leq 2a$.

On the other hand, non-standard numeration systems have been widely studied. Given a strictly increasing sequence of integers U, every nonnegative integer N can be represented with respect to the system U, that is to say, N has a representation $d_k \cdots d_0$ such that $N = \sum_{i=0}^{k} d_i u_i$. A classical way to obtain such a representation is to use a greedy algorithm ([Fr85]), which gives the greatest representation for the lexicographical ordering. The digits d_i are then elements of a canonical alphabet A_U, denoted by A for short. The set of greedy representations of all the nonnegative integers is denoted by $L(U)$. For instance, taking $U = \{K^n \mid n \geq 0,\ K \text{ integer } \geq 2\}$ gives the standard K-ary number system with $A = \{0, \cdots, K-1\}$. The Fibonacci numeration system is defined from the sequence of Fibonacci numbers with $u_0 = 1$, $u_1 = 2$, $u_n = u_{n-1} + u_{n-2}$ for $n \geq 2$, and $A = \{0, 1\}$ (see [K88]). One of the interests in non-standard numeration systems relies in the fact that they are naturally redundant.

The successor function in the numeration system associated to U is the function Succ : $A^* \longrightarrow A^*$ that maps the greedy U-representation of the integer N onto the greedy U-representation of $N+1$. In [F96] we have proved that the successor function is computable by a finite 2-tape automaton if and only if $L(U)$ is recognizable by a finite automaton (the proof is given in Theorem 4 below). When the set $L(U)$ is recognizable by a finite automaton then U must be a linear recurrent sequence with integral coefficients [Sh92].

These questions are linked to the representation of real numbers in non-integral base $\beta > 1$, and particularly to what is known as the β-expansion of 1, denoted by $d(1, \beta)$ (see Section 2.2). In [Ho95] are given conditions on the β-expansion of 1 and on associated sequences U which imply that the set $L(U)$ is recognizable by a finite automaton.

In this paper we focus on the sequentiality of the finite 2-tape automaton computing the successor function. We first study the left sequentiality (the sequentiality from left to right) of the successor function in non-standard numeration systems. We show that the successor function associated to U is a left subsequential function if and only if U is one of the following sequences :

Case 1. Let $\beta > 1$ be a number such that $d(1, \beta) = d_1 \cdots d_m$, and let $U = (u_n)_{n \geq 0}$ be defined by

$$u_n = d_1 u_{n-1} + \cdots + d_m u_{n-m} + 1, \text{ for } n \geq n_0 \geq m$$

with $1 = u_0 < u_1 < u_2 < \cdots < u_{n_0-1}$.

Case 2. U is the set of positive integers.

In [FSa97] we have written an algorithm which, given a left subsequential 2-tape automaton computing a relation such that the difference between the length of input words and the length of output words is bounded, constructs an equivalent *on-line* finite automaton, that is to say, a left subsequential finite 2-tape automaton which is letter-to-letter after an initial period where it reads the input and outputs nothing. As a corollary, we obtain that, for the above systems U, it is possible to design an on-line finite 2-tape automaton which computes the successor function.

We then consider right sequentiality and prove that the successor function associated to a sequence U is a right subsequential function (on $0^*L(U)$) if and only if $L(U)$ is recognizable by a finite automaton and if the set M of lexicographically maximum words of $L(U)$ is of the form :

$$M = \bigcup_{i=1}^{i=p} y_i^* z_i \cup M_0$$

where M_0 is finite, $|y_i| = p$ and the union is disjoint.

A case which is frequently met is the following one : U is an integral linear recurrent sequence with characteristic polynomial P having a dominant root $\beta > 1$. Then the successor function associated to U is right subsequential if and only if the following conditions are satisfied :

1) the β-expansion of 1 is finite : $d(1, \beta) = d_1 \cdots d_m$,

2) U is defined by

$$u_n = d_1 u_{n-1} + \cdots + d_m u_{n-m} \text{ for } n \geq n_0 \geq m$$

and $1 = u_0 < u_1 < \cdots < u_{n_0-1}$ (Theorem 3).

In a dynamical context, the successor function is extended to what is called *odometer* or *adding machine* (see [GLT95]). We make a connection with a result of [GLT95], showing that : Let U such that $L(U)$ is recognizable by a finite automaton. Then the odometer associated to U is continuous if and only if the successor function is right subsequential on $0^*L(U)$.

Recall that the *normalization function* on an alphabet of integers C is the function $\nu_C : C^* \longrightarrow A^*$ which maps any U-representation on C^* of a nonnegative integer onto the greedy U-representation of that integer (see [FSo96]). Addition of nonnegative integers represented with respect to U is a particular case of normalization : let $A = \{0, \cdots, a\}$ be the canonical alphabet associated to U, then addition is the normalization $\{0, \cdots, 2a\}^* \longrightarrow \{0, \cdots, a\}^*$. Here we give an example (Example 1) where the function Succ is left subsequential, although normalization is never computable by a finite 2-tape automaton, and an other one (Example 3) where Succ is right subsequential, and such that for any alphabet $C \supset A$, normalization on C is not computable by a finite 2-tape automaton.

2 Definitions

2.1 Representation of integers

Let $U = (u_n)_{n \geq 0}$ be a strictly increasing sequence of integers with $u_0 = 1$. A *representation in the system U* — or a *U-representation* — of a nonnegative integer N is a finite sequence of integers $(d_i)_{0 \leq i \leq k}$ such that

$$N = \sum_{i=0}^{k} d_i u_i.$$

Such a representation will be written $d_k \cdots d_0$, most significant digit first.

A word $d = d_k \cdots d_0$ is said to be *lexicographically greater* than a word $f = f_k \cdots f_0$, and this will be denoted by $d >_{lex} f$, if there exists an index $0 \leq i \leq k$ such that $d_k = f_k, \ldots, d_{i+1} = f_{i+1}$ and $d_i > f_i$. Among all possible U-representations $d_k \cdots d_0$ of a given integer N one is distinguished and called the *greedy* (or the *normal) U-representation* of N: it is the greatest in the lexicographical ordering. It is obtained by the following greedy algorithm (see [Fr85]):

Given integers m and p let us denote by $q(m, p)$ and $r(m, p)$ the quotient and the remainder of the Euclidean division of m by p.

Let $k \geq 0$ such that $u_k \leq N < u_{k+1}$ and let $d_k = q(N, u_k)$ and $r_k = r(N, u_k)$, $d_i = q(r_{i+1}, u_i)$ and $r_i = r(r_{i+1}, u_i)$ for $i = k-1, \cdots, 0$. Then $N = d_k u_k + \cdots + d_0 u_0$.

The greedy representation of N will be denoted by $\langle N \rangle$. By convention the greedy representation of 0 is the empty word ε. Under the hypothesis that the ratio u_{n+1}/u_n is bounded by a constant as n tends to infinity (that we will assume in this paper), the integers d_i of the greedy U-representation of any integer N are bounded and contained in a *canonical* finite alphabet A_U associated to U. The set of greedy U-representations of all the nonnegative integers is a subset of the free monoid A_U^*, and is denoted by $L(U)$. The sequence U together with the alphabet A_U defines a *numeration system* associated to U. In the sequel we denote A_U by A. The *numerical value* of a word $w = d_k \cdots d_0$, is given by $\pi(w) = \sum_{i=0}^{k} d_i u_i$.

The successor function in the numeration system associated to U is the function Succ : $A^* \longrightarrow A^*$ that maps the greedy U-representation of the integer N onto the greedy U-representation of $N + 1$.

2.2 Representation of real numbers

Let $\beta > 1$ be a real number. A *representation in base* β (or a β-representation) of a real number $x \in [0, 1]$ is an infinite sequence $(x_i)_{i \geq 1}$ such that $x = \sum_{i \geq 1} x_i \beta^{-i}$.

A particular β-representation of x — called the *β-expansion* — can be computed by the "greedy algorithm" [R57] : Denote by $[y]$ and $\{y\}$ the integer part and the fractional part of a number y. Let $x_1 = [\beta x]$, $r_1 = \{\beta x\}$, and, for $i \geq 2$, $x_i = [\beta r_{i-1}]$, and $r_i = \{\beta r_{i-1}\}$. Then $x = \sum_{i \geq 1} x_i \beta^{-i}$. When β is not an integer, the digits x_i obtained by this algorithm are elements of the set $\{0, \ldots, [\beta]\}$; when β is an integer, the digits x_i of the β-expansion of a number $x \in [0, 1[$ are in $\{0, \ldots, \beta - 1\}$, and the β-expansion of 1 is just $d(1, \beta) = \beta$. If an expansion ends in infinitely many zeros, it is said to be *finite*, and the ending zeros are omitted.

An infinite sequence $s = (s_i)_{i \geq 1}$ is said to be greater in the lexicographical ordering than $t = (t_i)_{i \geq 1}$, and it is denoted by $s >_{lex} t$, if there exists an $i \geq 0$ such that $s_1 = t_1, \ldots, s_i = t_i$ and $s_{i+1} > t_{i+1}$. The β-expansion of 1 is denoted by $d(1, \beta) = (d_i)_{i \geq 1}$. Let D_β be the set of β-expansions of numbers of $[0, 1[$. We recall the theorem of Parry [P60]: a sequence $s = (s_n)_{n \geq 1}$ is in D_β if and only if for every $i \geq 1$, $s_i s_{i+1} \cdots$ is smaller in the lexicographical ordering than $d(1, \beta)$ when the latter is infinite, respectively smaller than $d^*(1, \beta) = (d_1 \cdots d_{m-1}(d_m - 1))^\omega$ when $d(1, \beta) = d_1 \cdots d_m$ is finite (where w^ω denotes the infinite word $www \cdots$).

2.3 Finite automata and words

We recall some definitions. More details can be found in [E74] or in [HU79]. An *automaton over a finite alphabet* A, $\mathcal{A} = (Q, A, E, I, T)$ is a directed graph labelled by elements of A; Q is the set of *states*, $I \subset Q$ is the set of *initial* states, $T \subset Q$ is the set of *terminal* states and $E \subset Q \times A \times Q$ is the set of labelled *edges*. If $(p, a, q) \in E$, we note $p \xrightarrow{a} q$. The automaton is *finite* if Q is finite, and this will always be the case in this paper. The automaton $\mathcal{A}$ is *deterministic* if E is the graph of a (partial) function from $Q \times A$ into Q, and if there is a unique initial state. A subset H of A^* is said to be *recognizable by a finite automaton* (or *regular*) if there exists a finite automaton $\mathcal{A}$ such that H is equal to the set of labels of paths starting in an initial state and ending in a terminal state. Let $A^{\mathbf{N}}$ be the set of infinite sequences (or infinite words) on A. A subset K of $A^{\mathbb{N}}$ is said to be *recognizable by a finite automaton* if there exists a finite automaton $\mathcal{A}$ such that K is equal to the set of labels of infinite paths starting in an initial state and going infinitely often through a terminal state (Büchi acceptance condition, see [E74]).

A *2-tape automaton* is an automaton over the non-free monoid $A^* \times B^*$: $\mathcal{A} = (Q, A^* \times B^*, E, I, T)$ is a directed graph the edges of which are labelled by elements of $A^* \times B^*$. Words of A^* are referred as *input words*, as words of B^* are referred as *output words*. If $(p,(f,g),q) \in E$, we note $p \xrightarrow{f/g} q$. The automaton is finite if the set of edges E is finite (and thus Q is finite). These finite 2-tape automata are also known as *transducers*. A relation R of $A^* \times B^*$ is said to be *computable by a finite 2-tape automaton* if there exists a finite 2-tape automaton $\mathcal{A}$ such that R is equal to the set of labels of paths starting in an initial state and ending in a terminal state. A function is computable by a finite 2-tape automaton if its graph is computable by a finite 2-tape automaton. These definitions extend to relations and functions of infinite words as above.

A 2-tape automaton $\mathcal{A}$ with edges labelled by elements of $A \times B^*$ is said to be *left sequential* if the *underlying input automaton* obtained by taking the projection over A of the label of every edge is deterministic (see [Ber79]). A *left subsequential* 2-tape automaton is a left sequential automaton $\mathcal{A} = (Q, A \times B^*, E, \{i\}, \omega)$, where ω is the *terminal* function $\omega : Q \longrightarrow B^*$, whose value is concatenated to the output word corresponding to a computation in $\mathcal{A}$.

A 2-tape automaton $\mathcal{A}$ is said to be *letter-to-letter* if the edges are labelled by couples of letters, that is, by elements of $A \times B$.

All the automata considered so far work implicitly from left to right, that is to say, words are processed from left to right. It is possible to define in a dual way *right* automata, where words are processed from right to left. Usual automata are thus *left* automata.

Let H be a subset of A^*. The *left congruence* modulo H is defined on A^* by

$$f \sim_H g \Leftrightarrow [\forall h \in A^*,\ hf \in H \text{ if and only if } hg \in H].$$

It is known that the set H is recognizable by a finite automaton if and only if the left congruence modulo H has finite index (Myhill-Nerode Theorem, see [E74] or [HU79]). Let us denote by $[f]_H$ the class of f modulo $\sim_H$. Suppose that $\sim_H$ has finite index. One constructs the minimal deterministic right automaton $\mathcal{R}$ recognizing H as follows ([E74]) :

- the set of states of $\mathcal{R}$ is the set $\{[f]_H \mid f \in A^*\}$
- the initial state is $[\varepsilon]_H$
- the set of terminal states is equal to $\{[f]_H \mid f \in H\}$
- for every state $[f]_H$ and every $a \in A$, there is an edge $[f]_H \xrightarrow{a} [af]_H$ (words are processed

from right to left!).

Such a construction implies that there might exist a *sink*, i.e. a non-terminal state s such that, for any letter $a \in A$, there is a loop $s \xrightarrow{a} s$. This happens when $s = [w]_H$, w not in H, and there is no w' such that $w'w$ belongs to H.

A *factor* of a word w is a word f such that there exist words w' and w'' with $w = w'fw''$. When $w' = \varepsilon$, f is said to be a *prefix* of w, and when $w'' = \varepsilon$, f is said to be a *suffix* of w. If H is a subset of A^* we denote by $F(H)$ (resp. $PF(H)$, resp. $SF(H)$) the set of factors (resp. prefixes, resp. suffixes) of words of H. The *length* of a word $w = w_1 \cdots w_n$ with w_i in A for $1 \leq i \leq n$ is denoted by $|w|$ and is equal to n. By w^n is denoted the word obtained by concatenating n times w. The set of words of length n (resp. $\leq n$) of A^* is denoted by A^n (resp. $A^{\leq n}$). By H^+ is denoted $H^* \setminus \varepsilon$. A word f is a factor of an infinite word s if $s = wfs'$, with $s' \in A^{\mathbf{N}}$. The set of factors of a subset K of $A^{\mathbf{N}}$ is denoted by $F(K)$.

3 Main results

3.1 Preliminaries

First, if the successor function associated to U is computable by a 2-tape automaton, then its domain $L = L(U)$ is recognizable by a finite automaton. So in the sequel we assume that L is recognizable by a finite automaton. Then, by [Sh92], U must be a linear recurrent sequence with integral coefficients. Let us recall the following results.

Proposition 1 (folklore, see [Sa83]) *Let H be a subset of A^*, and let $M(H)$ be the union of the lexicographically maximum words of H of each length, as follows :*

$$M(H) = \bigcup_{n \geq 0} \{v \in H \cap A^n \mid \forall w \in H \cap A^n,\ w \leq_{lex} v\}.$$

Then, if H is recognizable by a finite automaton, so is $M(H)$.

Let us denote by M the language $M(L)$ of lexicographically maximum words of L. Let m_n be the word of length n which is maximum in the lexicographical ordering : $m_n = \langle u_n - 1 \rangle$, and $M = \cup_{n \geq 0} \{m_n \mid n \in \mathbf{N}\}$. Notice that the empty word $\varepsilon = m_0$ belongs to M. We have

Proposition 2 [Ho95] *The language L is equal to $L = \cup_{n \geq 0} \{v \in A^n \mid$ every suffix of length $i \leq n$ of v is $\leq_{lex} m_i\}$.*

Proposition 3 [Ho95] *Since $|M \cap A^n| = 1$ for all $n \geq 0$, and M is recognizable by a finite automaton, there exist an integer p, words x_i, y_i, and z_i such that*

$$M = \bigcup_{i=1}^{i=p} x_i y_i^* z_i \cup M_0$$

where M_0 is finite, $|y_i| = p$, and the union is disjoint.

Lemma 1 *The function Succ has the following property : for any word w of L,*

$$0 \leq |Succ\ (w)| - |w| \leq 1.$$

Proof. Let us suppose that $w = w_k \cdots w_0 = \langle N \rangle$. Thus, $N + 1 = 1 + \sum_{i=0}^{i=k} w_i u_i$. As w is greedy, one has $N < u_{k+1}$. Thus $N + 1 \leq u_{k+1}$, and so $|\langle N + 1 \rangle| \leq k + 2$. ■

Thus, it is more convenient to consider words of $0^* L$, denoted by L_0 for short. The function Succ is extended to L_0 in the obvious way. In particular, Succ $(0) =$ Succ $(\varepsilon) = 1$.

Lemma 2 *Let $w = w_k \cdots w_0$ be a word in $0^+ L$. Let $w_{i-1} \cdots w_0$ be the longest suffix of w which belongs to M. Then Succ $(w) = w_k \cdots w_{i+1}(w_i + 1)0^i$.*

3.2 Left sequentiality

We begin giving a proof of the well known fact that, in the classical K-ary number system, where K is an integer ≥ 2, the successor function is not sequentially computable from left to right.

Lemma 3 *In the K-ary number system the successor function cannot be realized by a left subsequential 2-tape automaton.*

Proof. Recall that in the K-ary system $L_0 = \{0, \cdots, K-1\}^*$ and $M = (K-1)^*$. Let d_l be the *left-distance* on A^* defined by

$$d_l(v, w) = |v| + |w| - 2 \mid v \wedge_l w \mid$$

where $v \wedge_l w$ denotes the longest common prefix to v and w.

Let $v = 0(K-1)^n$ and $w = 0(K-1)^{n-1}0$. Then Succ $(v) = 10^n$, Succ $(w) = 0(K-1)^{n-1}1$. We have $d_l(v, w) = 2$, $d_l(\text{Succ } (v), \text{Succ } (w)) = 2(n + 1)$. Thus the left-distance between Succ (v) and Succ (w) becomes unbounded when n goes to infinity, as the distance between v and w is bounded. By a result of [Ch77], it follows that Succ cannot be realized by a left subsequential 2-tape automaton. ■

Theorem 1 *The successor function associated to U is a left subsequential function if and only if U is one of the following sequences :*

Case 1. *Let $\beta > 1$ be a number such that $d(1,\beta) = d_1 \cdots d_m$, and let U be defined by*

$$u_n = d_1 u_{n-1} + \cdots + d_m u_{n-m} + 1, \; for \; n \geq n_0 \geq m$$

with $1 = u_0 < u_1 < u_2 < \cdots < u_{n_0 - 1}$.

Case 2. *U is the set of positive integers $\mathbf{N} \setminus 0$ (pathological case).*

Proof. We split the proof into several parts.

Proposition 4 *Let $\beta > 1$ be a number such that $d(1,\beta) = d_1 \cdots d_m$, and let U be defined by*

$$u_n = d_1 u_{n-1} + \cdots + d_m u_{n-m} + 1, \; for \; n \geq n_0 \geq m$$

with $1 = u_0 < u_1 < u_2 < \cdots < u_{n_0 - 1}$. Then the set of lexicographically maximum words is equal to

$$M = d_1 \cdots d_m 0^{k+1-m} 0^* \cup M_0$$

where M_0 is a finite set and k is the length of the longest word of M_0, and the successor function associated to U is a left subsequential function.

Proof. Let U be defined as above. Then U satisfies the linear recurrence

$$u_{n+1} = (d_1 + 1) u_n + (d_2 - d_1) u_{n-1} + \cdots + (d_m - d_{m-1}) u_{n+1-m} - d_m u_{n-m}$$

for $n \geq n_0$, and the characteristic polynomial is $P(X) = (X-1)(X^m - d_1 X^{m-1} - \cdots - d_m)$, with β for dominant root.

Let $n \geq n_0 + m - 1$. Since $u_n - 1 = d_1 u_{n-1} + \cdots + d_m u_{n-m}$, we have to show that $d_1 \cdots d_m 0^{n-m}$ is the greedy representation of $u_n - 1$. Suppose the greedy representation of $u_n - 1$ is not that one; since $u_{n-1} \leq u_n - 1 < u_n$, the greedy representation of $u_n - 1$ is $>_{lex} d_1 \cdots d_m 0^{n-m}$, and thus is of the form $d_1 \cdots d_{i-1}(d_i + c) f$, where $1 \leq i \leq m-1$, $c \geq 1$ and $|f| = n - i$. Thus $c u_{n-i} + \pi(f) = d_{i+1} u_{n-i-1} + \cdots + d_m u_n$. Hence $d_{i+1} u_{n-i-1} + \cdots + d_m u_n \geq u_{n-i} = d_1 u_{n-i-1} + \cdots + d_m u_{n-i-m} + 1$, which is impossible because $d_1 \cdots d_m$ is a beta-expansion and thus $d_{i+1} \cdots d_m 0^i <_{lex} d_1 \cdots d_m$ (Theorem of Parry [P60], see Section 2.2).

Now, when $n \leq n_0 + m - 2$, the greedy representation of $u_n - 1$ depends on the choice of the initial conditions $u_1, \ldots, u_{n_0 - 1}$. For instance, if we take for initial conditions the canonical initial conditions associated to β (see [B-M89])

$$u_0 = 1, \; u_i = d_1 u_{i-1} + \cdots + d_i u_0 + 1, \; 1 \leq i \leq m-1$$

and $n_0 = m$, then it is easily checked that, in that case,

$$M = d_1 \cdots d_m 0^* \cup \{\varepsilon, d_1, d_1 d_2, \ldots, d_1 \cdots d_{m-1}\}.$$

We now show that L_0 is recognizable by a finite automaton. Let A be the canonical alphabet associated to U. Let

$$Y = \{0, \ldots, d_1 - 1, d_1 0, \ldots, d_1(d_2 - 1), \ldots, d_1 \cdots d_{m-1} 0, \ldots, d_1 \cdots d_{m-1}(d_m - 1)\}.$$

Then $L_0 = \{f \in Y^* d_1 \cdots d_m 0^{k+1-m} 0^* \cup A^{\leq k} \mid$ every suffix of length $j \leq k$ of f is $\leq_{lex} m_j\}$. This comes from the fact that, if $f \in Y^* d_1 \cdots d_m 0^{k+1-m} 0^*$, $|f| = n \geq k+1$, then $f \leq_{lex} d_1 \cdots d_m 0^{n-m}$ by the theorem of Parry recalled above.

Let $\alpha = k + 1 - m$. So $M = d_1 \cdots d_m 0^\alpha 0^* \cup M_0$. If $\alpha = 0$, then $M = d_1 \cdots d_m 0^* \cup M_0$. Let $\mathcal{M} = (Q, A, E, i_0, T)$ be the following deterministic automaton recognizing $0M$:

• $Q = \{i_0\} \cup PF(0M_0) \cup PF(0 d_1 \cdots d_m 0^\alpha)$, where $PF(H)$ is the set of prefixes of elements of H.

• The set of edges E is defined by : if $q \in Q$, there is an edge $q \xrightarrow{a} qa$ when $a \in A$ and $qa \in Q$. There is an edge $i_0 \xrightarrow{0} 0$. Let us denote by t the state $t = 0 d_1 \cdots d_m 0^\alpha$, there is a loop $t \xrightarrow{0} t$.

• The set of terminal states is $T = 0M_0 \cup \{t\}$.

We consider words beginig with a 0. Remark that if $f \in L_0$, and if $f = f' a d_1 \cdots d_m 0^n$, where $a \in A$ and $n \geq \alpha$, then Succ $(f) = f'(a+1)0^{m+n}$, and if $f = f' a m_i$, where $m_i \in M_0$, $|m_i| = i$, then Succ $(f) = f'(a+1)0^i$ by Lemma 2. So the idea to realize Succ as a left subsequential 2-tape automaton is the following : we construct a 2-tape automaton realizing the identity at the beginning, but with delay one, that is to say, we keep in memory (in the states) the last letter read, until we reach a suffix which is in M, and is transformed as indicated above.

Here is the construction of a left subsequential 2-tape automaton realizing Succ: $\mathcal{S} = (R, A \times A^*, F, i_0, \omega)$. The automaton $\mathcal{M}$ is a subautomaton of the underlying input automaton of $\mathcal{S}$. First, let us denote by X the set $X = PF(M_0) \cup PF(d_1 \cdots d_m 0^{\alpha-1})$ if $\alpha \geq 1$, $X = PF(M_0) \cup PF(d_1 \cdots d_{m-1})$ if $\alpha = 0$.

• The set of states R will be a subset of $AX \cup \{i_0\} \cup \{d_1 \cdots d_m 0^\alpha\}$, containing Q as a subset, and inductively constructed from Q as indicated below. For notation coherence, the state t of $\mathcal{M}$ is here denoted by $d_1 \cdots d_m 0^\alpha$.

• The set of edges F will be defined as follows : first, there is an edge $i_0 \xrightarrow{0/\varepsilon} 0$. Secondly, if $\alpha \geq 1$ (Case 1), and if $r d_1 \cdots d_m 0^{\alpha-1} \in R$, there is an edge $r d_1 \cdots d_m 0^{\alpha-1} \xrightarrow{0/r+1} d_1 \cdots d_m 0^\alpha$; and there is a loop $d_1 \cdots d_m 0^\alpha \xrightarrow{0/0} d_1 \cdots d_m 0^\alpha$. If $\alpha = 0$ (Case 2), we put $r d_1 \cdots d_{m-1} \xrightarrow{d_m/r+1}$

$d_1 \cdots d_m$, and there is a loop $d_1 \cdots d_m \xrightarrow{0/0} d_1 \cdots d_m$.
Now, we give a general rule for defining new edges of the form $q \xrightarrow{a/\lambda(q,a)} \delta(q,a)$ as follows : let $0 \le n \le m+\alpha$, and $q = r_0 \cdots r_n$ be a state of R different from i_0 and from the states of the form $rd_1 \cdots d_m 0^\alpha$ or $rd_1 \cdots d_m 0^{\alpha-1}$ (Case 1), and $rd_1 \cdots d_{m-1}$ (Case 2) just mentioned above. Let $l \ge 0$ be the minimum index such that $r_{l+1} \cdots r_n a$ is in X, then let the next state be $\delta(q,a) = r_l r_{l+1} \cdots r_n a$ and the output be $\lambda(q,a) = r_0 \cdots r_{l-1}$. When $r_1 \cdots r_n a \in X$ ($l = 0$), we get $\delta(q,a) = r_0 \cdots r_n a$ and $\lambda(q,a) = \varepsilon$, and when there is no suffix of $r_0 \cdots r_n a$ belonging to X ($l = n+1$), we get $\delta(q,a) = a$ and $\lambda(q,a) = r_0 \cdots r_n$.

All we have to do now is to determine for which letters $a \in A$ these edges are valid. Let us consider a state of form $q = rd_1 \cdots d_n$, $n \le m-1$ if $\alpha \ge 1$, or $n \le m-2$ when $\alpha = 0$. Then for any letter $a < d_{n+1}$ such that there is no edge labelled by a leaving q in $\mathcal{M}$, we define an edge $q \xrightarrow{a/\lambda(q,a)} \delta(q,a)$.
For any state of the form $q = r_0 r_1 \cdots r_n$ such that $r_1 \cdots r_n$ is in $PF(M_0)$, and for any letter $a < r_{n+1}$ such that $r_1 \cdots r_n r_{n+1}$ belongs to $PF(M_0)$, there is an edge $q \xrightarrow{a/\lambda(q,a)} \delta(q,a)$.
• The terminal function ω is defined by : $\omega(d_1 \cdots d_m 0^\alpha) = 0^{m+\alpha}$. If $r_1 \cdots r_n \in M_0$, then $\omega(r_0 r_1 \cdots r_n) = (r_0+1)0^n$. If a state $q = r_0 \cdots r_n$ is in $L_0 \setminus M$, then $\omega(q) = r_0 \cdots r_{i-1}(r_i+1)r_{i+1} \cdots r_n$ where $r_{i+1} \cdots r_n$ is the longest suffix of q in M_0. ■

Example 1 Let $\beta = 2$, and $u_n = 2u_{n-1} + 1$ for $n \ge 1$, and $u_0 = 1$. Then $u_n = 2^{n+1} - 1$, $M = 20^* \cup \varepsilon$, $L_0 = \{0,1\}^* \cup \{0,1\}^* 20^*$. The sequence U is linearly recurrent, given by $u_n = 3u_{n-1} - 2u_{n-2}$ for $n \ge 2$ and $u_1 = 3$, $u_0 = 1$. Here is the left subsequential 2-tape automaton $\mathcal{S}$ realizing Succ.

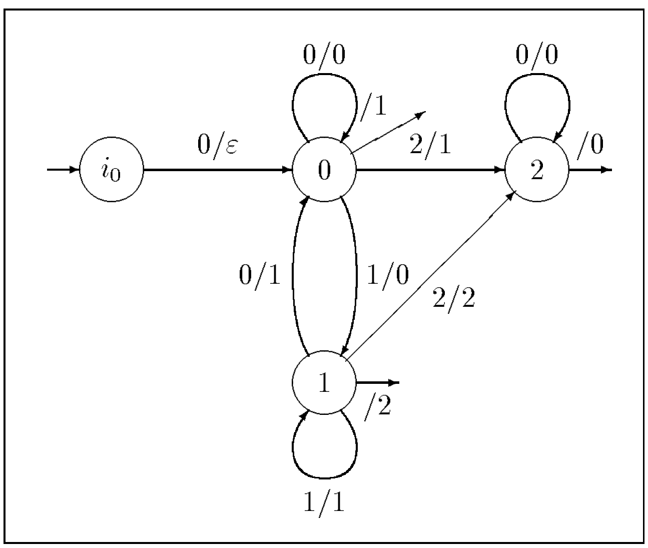

Figure 1. Left-subsequential 2-tape automaton $\mathcal{S}$

It can be shown that, in this system U, normalization on any alphabet (in particular addition) is never computable by a finite automaton. □

Proposition 5 *Let U be the set $\mathbf{N} \setminus 0$. Then $L = M = 10^* \cup \varepsilon$, and the successor function is left subsequential.*

Proof. Since $u_n = n+1$ for $n \geq 0$, $L = M = 10^* \cup \varepsilon$. The characteristic polynomial of U is $P(X) = (X-1)^2$. Below is the left subsequential 2-tape automaton realizing Succ.

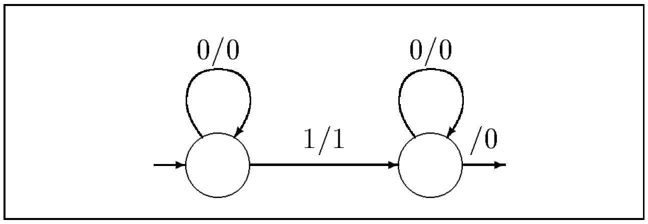

Figure 2. Left-subsequential 2-tape automaton for $\mathbf{N} \setminus 0$

■

Proposition 6 *If U is a sequence such that M is not of the form $s0^* \cup M_0$, where s is a non-empty word not in 0^*, and M_0 is a finite set, then Succ cannot be realized by a left subsequential 2-tape automaton.*

Proof. Let us show that, if $M \neq s0^* \cup M_0$, then there exist words x, y, z and g such that, for every $n \geq 1$, xy^nz is in M and $xy^{n-1}g$ is in $L \setminus M$.
1) Let us suppose that there exists an i, $1 \leq i \leq p$, such that $x_i y_i^* z_i \subseteq M$ with $z_i \neq \varepsilon$, $z_i \notin 0^+$. Then there exists $h <_{lex} z_i$, $|h| = |z_i|$. Thus for every $n \geq 1$, $x_i y_i^n z_i \in M$ and $x_i y_i^n h \in L \setminus M$, by Proposition 2.
2) Otherwise, for every i, $1 \leq i \leq p$, $z_i \in 0^*$. First, let us suppose that there exists an i such that $y_i \notin 0^+$. Then let $h <_{lex} y_i$, $|h| = |y_i|$. Thus for every $n \geq 1$, $x_i y_i^n z_i \in M$ and $x_i y_i^{n-1} h z_i$ is in $L \setminus M$.
Otherwise, suppose that for every i, $1 \leq i \leq p$, $y_i = 0^p$. Then by hypothesis, p must be ≥ 2. For simplicity, suppose $p = 2$. Then $M = x_1(00)^* \cup x_2(00)^* \cup M_0$. Suppose that $x_1 0^\omega <_{lex} x_2 0^\omega$. Then there exists $k \geq 0$ such that $x_1(00)^{n-1}0^k <_{lex} x_2(00)^n$, thus for $n \geq 1$, $x_1(00)^{n-1}0^k \in L \setminus M$ and $x_1(00)^n \in M$.

Now let $v = 0xy^nz$ and $w = 0xy^{n-1}g$ be determined as above. We have $d_l(v, w) = |y| + |z| + |g| - 2 \mid (yz) \wedge_l g \mid = K$, a constant, and Succ $(v) = 10^{n|y|+|x|+|z|}$. Without loss of generality, we can assume that the longest suffix of w belonging to M_0 is a suffix of g. Let $g = g_m \cdots g_i \cdots g_0$, $i \geq 0$, where $g_{i-1} \cdots g_0$ is the longest suffix of g belonging to

M_0. Then Succ $(w) = 0xy^{n-1}g_m \cdots g_{i+1}(g_i+1)g_{i-1}\cdots g_0$. We have $d_l(\text{Succ }(v), \text{Succ }(w)) = (2n-1)|y| + 2|x| + |z| + |g| + 2$. Thus, as in Lemma 3, Succ cannot be realized by a left subsequential 2-tape automaton. ■

Proposition 7 *The only sequences U such that $M = s0^* \cup M_0$, where s is a non-empty word, are given by :*

Case 1. *Let $\beta > 1$ be a number such that $d(1,\beta) = d_1 \cdots d_m$, and let U be defined by*

$$u_n = d_1 u_{n-1} + \cdots + d_m u_{n-m} + 1, \ \text{for } n \geq n_0 \geq m$$

with $1 = u_0 < u_1 < u_2 < \cdots < u_{n_0-1}$.

Case 2. *U is the set of positive integers $\mathbf{N} \setminus 0$.*

Proof. We consider $s \neq 1$, $s = d_1 \cdots d_m 0^\alpha$, with $\alpha \geq 0$ and $d_m \neq 0$. Since $d_1 \cdots d_m 0^j \in L$ for $j \geq \alpha + m - 1$, by Proposition 2 we get

$$d_2 \cdots d_m 0 <_{lex} d_1 \cdots d_m$$

$$d_3 \cdots d_m 00 <_{lex} d_1 \cdots d_m$$

$$\cdots$$

$$d_m 0^{m-1} <_{lex} d_1 \cdots d_m$$

and by a result of Parry [P60], there exists a unique real number $\beta > 1$ such that $d(1,\beta) = d_1 \cdots d_m$. Now, since $d_1 \cdots d_m 0^n \in M$ for $n \geq \alpha$, we have $d_1 \cdots d_m 0^n = \langle u_{m+n} - 1 \rangle$, so $u_{m+n} = d_1 u_{m+n-1} + \cdots + d_m u_{n-1} + 1$, for $n \geq \alpha$. ■

As a corollary of Theorem 1 we get the following.

Corollary 1 *When U is a sequence satisfying the hypothesis of Theorem 1, the successor function is computable by an on-line finite automaton.*

Proof. It is a consequence of Lemma 1 and of [FSa97]. ■

3.3 Right sequentiality

Lemma 4 *There exist sequences U such that the function Succ cannot be realized by a right subsequential 2-tape automaton.*

Proof. Consider the sequence U defined by the following linear recurrent sequence

$$u_n = 3u_{n-1} - u_{n-2} \ ; \ u_0 = 1 \ , \ u_1 = 3.$$

$U = \{1, 3, 8, 21, \cdots\}$ is the sequence of Fibonacci numbers of even index. The canonical alphabet is $A = \{0, 1, 2\}$, $L_0 = \{0,1\}^* \cup \{0,1\}^*21^* \cup (\{0,1\}^*21^*0)^*$ and $M = 21^* \cup \varepsilon$. Since L_0 is recognizable by a finite automaton, Succ is computable by a finite 2-tape automaton.

The *right-distance* on A^* is defined by

$$d_r(v, w) = |v| + |w| - 2 \mid v \wedge_r w \mid$$

where $v \wedge_r w$ denotes the longest common suffix to v and w.

Let $v = 021^n$ and $w = 01^{n+1}$. Then Succ $(v) = 10^{n+1}$ and Succ $(w) = 01^n 2$. We have $d_r(v, w) = 2(n+2) - 2n = 4$, as $d_r(\text{Succ }(v), \text{Succ }(w)) = 2(n+2)$. From [Ch77] it follows that Succ cannot be realized by a right subsequential 2-tape automaton in the numeration system defined by U. ∎

Theorem 2 *The successor function associated to a sequence U is a right subsequential function (on $0^*L(U)$) if and only if $L(U)$ is recognizable by a finite automaton and if the set M of lexicographically maximum words of $L(U)$ is of the form :*

$$M = \bigcup_{i=1}^{i=p} y_i^* z_i \cup M_0$$

where M_0 is finite, $|y_i| = p$ and the union is disjoint.

The proof follows from the following results.

Proposition 8 *If M contains a set of the form xy^*z, with x and y non empty, then Succ cannot be realized by a right subsequential 2-tape automaton.*

Proof. From Proposition 3, we know that

$$M = \bigcup_{i=1}^{i=p} x_i y_i^* z_i \cup M_0$$

where M_0 is finite, $|y_i| = p$, and the union is disjoint. Suppose that there is a set $xy^*z \subseteq M$, with $x \neq \varepsilon$. Let $v = xy^n z \in M$; by assumption on the form of M, there exists $h <_{lex} x$, $|h| = |x|$, and $w = hy^n z \in L_0 \setminus M$. Without loss of generality, one can suppose that the longest suffix of w belonging to M_0 is a suffix of yz, that is to say, $yz = a_j \cdots a_i \cdots a_0$, with $0 \leq i \leq j-1$ being maximal such that $a_{i-1} \cdots a_0 \in M_0$. We have : Succ $(v) = 10^{n|y|+|x|+|z|}$, Succ $(w) = hy^{n-1} a_j \cdots a_{i+1}(a_i + 1)a_{i-1} \cdots a_0$, $d_r(v, w) = 2|x| - 2|x \wedge_r h| = H$, a constant, $d_r(\text{Succ }(v), \text{Succ }(w)) \geq 1 + 2(n|y| + |x| + |z|) - 2i$, hence Succ is not right subsequential by [Ch77]. ∎

Proposition 9 *If M is of the form*

$$M = \bigcup_{i=1}^{i=p} y_i^* z_i \cup M_0$$

where M_0 is finite, $|y_i| = p$ and the union is disjoint, then Succ is a right subsequential function.

Proof. The construction can be followed on Example 2 below. Since $L(U)$ is supposed to be recognizable by a finite automaton, so is L_0. Let $\mathcal{L} = (Q, A, E, r_0, T)$ be the minimal deterministic right automaton recognizing L_0 as in Section 2.3. But, notice that L_0 is *suffix-closed*, that is to say, if fg is in L_0, then g is in L_0 as well (because L is obtained from the greedy algorithm). In particular, for every letter a of A, a is in L_0. Suppose that f is in L_0; then by construction, $[f]_{L_0}$ is a terminal state of $\mathcal{L}$. If af, for $a \in A$, is not in L_0, then for every $b \in A$, baf won't be in L_0 either; thus we can suppress the state $[af]_{L_0}$ which is a sink, and the edge $[f]_{L_0} \xrightarrow{a} [af]_{L_0}$, and we can suppose that every state is terminal. Thus the set of states Q is equal to $T = \{[f]_{L_0} \mid f \in L_0\}$, $r_0 = [\varepsilon]_{L_0}$ is the initial state, every state is terminal, and there is an edge $[f]_{L_0} \xrightarrow{a} [af]_{L_0}$ for every a such that $af \in L_0$.

Let $\mathcal{M} = (Q', A, E', i_0, T')$ be a right deterministic automaton recognizing M, with no sink. Since the empty word is in M, i_0 is also terminal. We say that a set of the form y^*z, where $y \neq \varepsilon$, is a *frying pan*. Let $(b_p \cdots b_1)^* c_n \cdots c_1$ be a frying pan in M. It is recognized in $\mathcal{M}$ as follows : there is a simple path $i_0 \xrightarrow{c_1} q_1 \xrightarrow{c_2} \cdots \xrightarrow{c_n} q_n \xrightarrow{b_1} q_{n+1} \cdots \xrightarrow{b_p} q_{n+p}$, where $q_{n+p} = q_n \in T'$, and there is no other terminal state on the loop between q_n and q_{n+p}.

We construct a right subsequential 2-tape automaton $\mathcal{S} = (Q \cup Q', A \times A^*, F, i_0, \omega)$ containing $\mathcal{M}$ and $\mathcal{L}$ as subautomata of the underlying input automaton of $\mathcal{S}$.

- The set of states of $\mathcal{S}$ is $Q \cup Q'$.
- For every simple path $q_0 \xrightarrow{a_1} q_1 \xrightarrow{a_2} \cdots \xrightarrow{a_{k-1}} q_{k-1} \xrightarrow{a_k} q_k$ in $\mathcal{M}$, where q_k is a terminal state, and either $q_0 = i_0$ or $q_0 = q_k$, we define in $\mathcal{S}$ a path $q_0 \xrightarrow{a_1/\varepsilon} q_1 \xrightarrow{a_2/\varepsilon} \cdots \xrightarrow{a_{k-1}/\varepsilon} q_{k-1} \xrightarrow{a_k/0^k} q_k$. For each $q_j = [a_j \cdots a_1]_M$, $1 \leq j \leq k-1$, the terminal function ω is defined by $\omega(q_j) = a_j \cdots a_2(a_1+1)$, and $\omega(q_0) = \omega(q_k) = 1$. Now, for each q_j, $1 \leq j \leq k-1$, and for every letter d of A such that there is no edge labelled by d leaving q_j in $\mathcal{M}$, we create an edge $q_j \xrightarrow{d/da_j \cdots a_2(a_1+1)} [da_j \cdots a_1]_{L_0}$ if that state exists in $\mathcal{L}$, otherwise the edge is not created. From i_0 and for every letter d of A such that there is no edge labelled by d outgoing from i_0 in $\mathcal{M}$, we create an edge $i_0 \xrightarrow{d/d+1} [d]_{L_0}$ if that state exists. Similarly, from $t = q_k \in T'$, for every letter d of A such that there is no edge labelled by d outgoing from t in $\mathcal{M}$, we create an edge $q_k \xrightarrow{d/d+1} [da_k \cdots a_1]_{L_0}$ if that state exists.
- For every edge $q \xrightarrow{a} s$ in $\mathcal{L}$, q, $s \in Q$, an edge $q \xrightarrow{a/a} s$ is created in $\mathcal{S}$.

For $q \in Q$, the terminal function is given by $\omega(q) = \varepsilon$. ■

There is a funny case where Succ is right subsequential on L but not on 0^*L.

Proposition 10 *The sequence U is an arithmetic progression, defined by*

$$u_{k+p} = c + dp, \text{ for } p \geq 0$$

with $1 = u_0 < u_1 < \cdots < u_k = c$ and $0 < d \leq c$, if and only if the set of lexicographically maximum words is of the form $M = 10^*z \cup M_0$. In that case, the function Succ is right subsequential on L but is not right subsequential on 0^*L.

Proof. 1) Let $u_{k+p} = c + dp$, with $u_k = c$ and $0 < d \leq c$. Let $M_0 = \{\langle u_i - 1\rangle \mid 1 \leq i \leq k\}$. Since $d \leq c = u_k$, we set z to be the word of length k equal to $\langle d-1\rangle$ prefixed by the adequate number of 0's. We have $\langle u_{k+1} - 1\rangle = 1z$ and for $n \geq k+1$, $u_n = u_{n-1} + d$, thus $\langle u_n - 1\rangle = 10^{n-k-1}z$.

2) Let $M = 10^*z \cup M_0$ and $z = z_{k-1}\cdots z_0$. Let $d = z_{k-1}u_{k-1} + \cdots + z_0u_0 + 1$ and let $c = u_k$. Since z is greedy, $0 < d \leq c$, and for $p \geq 0$, $u_{k+p} = u_{k+p-1} + d = u_k + pd$. The characteristic polynomial of U is $P(X) = (X-1)^2$.

3) By Proposition 8 we know that Succ is not right subsequential on 0^*L. We now construct a right subsequential 2-tape automaton realizing Succ on L. First, $L = \{10^nv \mid n \geq 0,\ |v| = |z| = k,\ v \leq_{lex} z\} \cup \{f \mid |f| \leq k,$ such that f does not begin with 0's and every suffix of length i of f is $\leq_{lex} m_i\}$. Let $\mathcal{L} = (Q, A, E, i_0, T)$ be the minimal deterministic right automaton recognizing L_0, with $Q = \{[f]_L \neq \text{sink}\ \}$, $T = \{[f]_L \mid f \in L\}$, $i_0 = [\varepsilon]_L$, and there is an edge $[f]_L \xrightarrow{a} [af]_L$ for every a such that $af \neq$ sink. We write $[f]$ instead of $[f]_L$ for short.

Let $\mathcal{S} = (Q, A \times A^*, F, i_0, \omega)$ be a right subsequential 2-tape automaton having $\mathcal{L}$ as underlying input automaton. F contains a path of the form :
$i_0 \xrightarrow{z_0/\varepsilon} [z_0] \xrightarrow{z_1/\varepsilon} \cdots \xrightarrow{z_{k-1}/\varepsilon} [z]$, there is a loop $[z] \xrightarrow{0/0} [z]$, and an arrow $[z] \xrightarrow{1/10^{k+1}} [1z]$.
The other paths are of the following form, for $v = v_{k-1}\cdots v_0$ a word $<_{lex} z$:
$i_0 \xrightarrow{v_0/\varepsilon} [v_0] \xrightarrow{v_1/\varepsilon} \cdots \xrightarrow{v_{k-2}/\varepsilon} [v_{k-2}\cdots v_0] \xrightarrow{v_{k-1}/v_{k-1}\cdots v_{j+1}(v_j+1)0^j} [v]$ where $0 \leq j \leq k-1$ is maximal such that $v_{j-1}\cdots v_0 \in M_0$. There is a loop $[v] \xrightarrow{0/0} [v]$ and an arrow $[v] \xrightarrow{1/1} [1v]$.
For words $f_i \cdots f_0$, with $i \leq k-2$, the edges are of the form $i_0 \xrightarrow{f_0/\varepsilon} [f_1] \xrightarrow{f_1/\varepsilon} \cdots \xrightarrow{f_i/\varepsilon} [f_i \cdots f_0]$.
The terminal function is defined by $\omega([1z]) = \varepsilon$, $\omega([1v]) = \varepsilon$, for v as above, $\omega(i_0) = 1$, and if $f \in M_0$, $\omega(f) = 10^{|f|}$, and otherwise, for $0 \leq l \leq k-1$, $\omega(f_l \cdots f_0) = f_l \cdots f_{j+1}(f_j+1)0^j$ where $0 \leq j \leq l-1$ is maximal such that $f_{j-1}\cdots f_0 \in M_0$. ■

Remark that if we take $k = 0$, then $c = d = 1$, $u_p = p+1$, and we find again the pathological case, which has thus the property that Succ is both left and right subsequential on L.

We now make an additional hypothesis on the sequence U, which is fulfilled in many cases. Assume that U is an integral linear recurrent sequence with characteristic polynomial P such that P has a *dominant root* $\beta > 1$, that is to say, every other root γ of P is such that $|\gamma| < \beta$. Such a number β is called a *Perron* number.

Theorem 3 *Let U be an integral linear recurrent sequence with characteristic polynomial P having a dominant root $\beta > 1$. Then the successor function associated to U is right subsequential if and only if the following conditions are satisfied :*
1) the β-expansion of 1 is finite : $d(1,\beta) = d_1 \cdots d_m$,
2) U is defined by

$$u_n = d_1 u_{n-1} + \cdots + d_m u_{n-m} \text{ for } n \geq n_0 \geq m$$

and $1 = u_0 < u_1 < \cdots < u_{n_0-1}$.

Proof. The following has to be proved : Conditions 1 and 2 of Theorem 3 are satisfied if and only if the set of lexicographically maximum words is of the form $M = \cup_{i=1}^m y_i^* z_i \cup M_0$ and $L = L(U)$ is recognizable by a finite automaton. This will be a consequence of the two following lemmas.

Lemma 5 *Let $\beta > 1$ with $d(1,\beta) = d_1 \cdots d_m$, and let U be defined by*

$$u_n = d_1 u_{n-1} + \cdots + d_m u_{n-m} \quad \text{for } n \geq n_0 \geq m$$

and $1 = u_0 < u_1 < \cdots < u_{n_0-1}$. Then

$$M = \bigcup_{i=1}^{m} (d_1 \cdots d_{m-1}(d_m - 1))^* z_i \cup M_0$$

where M_0 is finite, and L is recognizable by a finite automaton.

Proof. We have to prove that, for n large enough, the greedy representation of $u_n - 1$ is of the form $(d_1 \cdots d_{m-1}(d_m - 1))^k z$, whith $n = mk + |z|$, and k maximum. If not, suppose that the greedy representation is $\langle u_n - 1 \rangle = (d_1 \cdots d_{m-1}(d_m - 1))^j d_1 \cdots d_{i-1}(d_i + c)w$, with $0 \leq j \leq k-1$, $c \geq 1$, $1 \leq i \leq m$, and $|w| = n - mj - i$. Then $cu_{|w|} + \pi(w) = d_{i+1}u_{|w|-1} + \cdots + d_m u_{|w|-m+i} - 1$, and thus $u_{|w|} = d_1 u_{|w|-1} + \cdots + d_m u_{|w|-m} < d_{i+1} u_{|w|-1} + \cdots + d_m u_{|w|-m+i}$, which is impossible because $d_1 \cdots d_m$ is the β-expansion of 1.

Now, for any $q \geq 0$, $\langle u_{n+qm} - 1 \rangle = (d_1 \cdots d_{m-1}(d_m - 1))^{k+q} z$, with the same word z because z is greedy, and thus there exist m different words $z_1, \ldots, z_m$ such that $M = \cup_{i=1}^m (d_1 \cdots d_{m-1}(d_m - 1))^* z_i \cup M_0$.

In [Ho95], it is proved that if $d(1,\beta) = d_1 \cdots d_m$ and $u_n = d_1 u_{n-1} + \cdots + d_m u_{n-m}$, then L is recognizable by a finite automaton. ∎

Lemma 6 *Suppose that P has a dominant root $\beta > 1$, $M = \bigcup_{i=1}^{m} y_i^* z_i \cup M_0$ and L is recognizable by a finite automaton. Then $d(1,\beta) = d_1 \cdots d_m$ and $u_n = d_1 u_{n-1} + \cdots + d_m u_{n-m}$ for $n \geq n_0 \geq m$.*

Proof. From [Ho95], we know that if L is recognizable by a finite automaton then $d(1,\beta)$ must be finite or eventually periodic. First remark that the case where $d(1,\beta)$ could be purely periodic, i.e. $d(1,\beta) = (d_1 \cdots d_m)^\omega$, is impossible, for we would get $1 = d_1 \cdots d_{m-1}(d_m + 1)$, which would be the correct β-expansion of 1.

If $d(1,\beta)$ is eventually periodic, $d(1,\beta) = d_1 \cdots d_l (d_{l+1} \cdots d_{l+m})^\omega$, then in $M = \cup_{i=1}^{m} x_i y_i^* z_i \cup M_0$, words x_i are of the form $x_i = d_1 \cdots d_l$ and words y_i are of the form $y_i = (d_{l+1} \cdots d_{l+m})^k$ (Lemma 7.4 of [Ho95]), a contradiction with the hypothesis.

If $d(1,\beta) = d_1 \cdots d_m$, $M = \cup_{i=1}^{m} x_i y_i^* z_i \cup M_0$ can have two forms (Lemma 7.5 of [Ho95]) :
Case 1. For each i, $x_i = (d_1 \cdots d_{m-1}(d_m - 1))^{k_i}(d_1 \cdots d_m)$ and $y_i = 0^n$. This is in contradiction with our hypothesis.
Case 2. For each i, $x_i = y_i = (d_1 \cdots d_{m-1}(d_m - 1))^k$. Then we get, for n large enough, $\langle u_n - 1 \rangle = (d_1 \cdots d_{m-1}(d_m - 1))^k z$, for some k, $\langle u_{n+m} - 1 \rangle = (d_1 \cdots d_{m-1}(d_m - 1))^{k+1} z$, thus $u_{n+m} = d_1 u_{n+m-1} + \cdots + d_m u_n$. ∎

Example 2 Fibonacci recurrence with non canonical initial conditions.

Let $U = (u_n)_{n \geq 0}$ be the linear recurrent sequence defined by

$$u_n = u_{n-1} + u_{n-2} \text{ for } n \geq 3,\ u_0 = 1,\ u_1 = 4,\ u_2 = 7.$$

The characteristic polynomial of U is $P(X) = X^2 - X - 1$ with $(1+\sqrt{5})/2$ for dominant root. The canonical alphabet is $A = \{0,1,2,3\}$. We have $M = (10)^*3 \cup (10)^*12$ and $L_0 = [\{0,1\}^* \setminus \{0,1\}^*11\{0,1\}^*]\{\varepsilon, 1, 2, 03\}$. On Figure 3, L_0 is recognized by $\mathcal{L}$, M by $\mathcal{M}$, and on Figure 4, Succ is computed by $\mathcal{S}$.

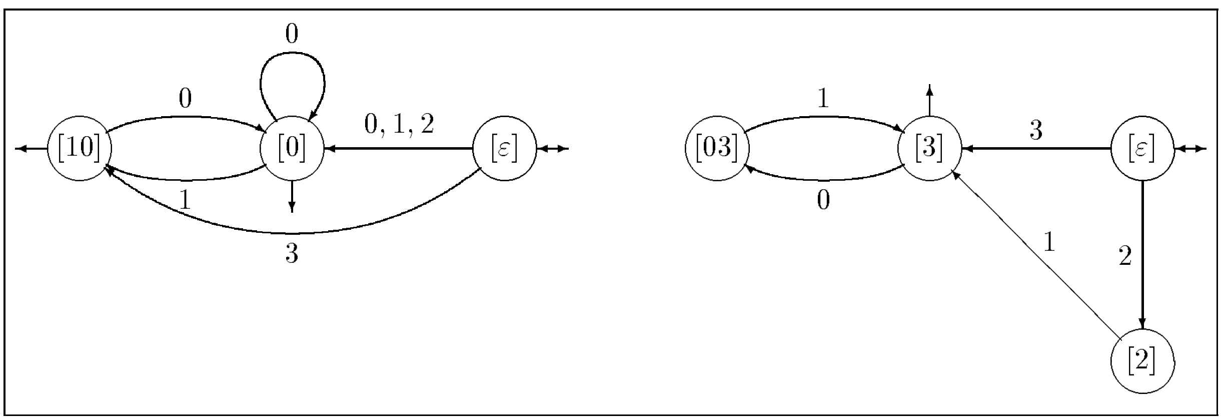

Figure 3. Right automata $\mathcal{L}$ and $\mathcal{M}$.

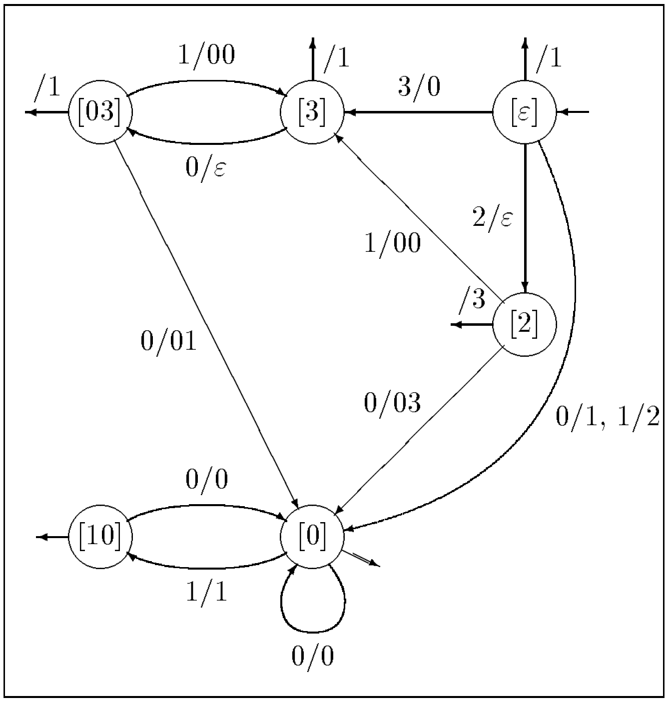

Figure 4. Right subsequential 2-tape automaton $\mathcal{S}$.

□

Recall that an algebraic integer is a root of a polynomial $X^d + a_1 X^{d-1} + \cdots + a_d$ with integral coefficients a_i. A *Pisot* number is an algebraic integer > 1 such that the other roots of its minimal polynomial have modulus < 1.

Example 3 Linear recurrence with a dominant root which is a Perron number but not a Pisot number.

Let $u_n = 3u_{n-1} + 2u_{n-2} + 3u_{n-4}$, for $n \geq 4$, with (for instance) $u_0 = 1$, $u_1 = 4$, $u_2 = 15$, $u_3 = 54$. In that case, the canonical alphabet is $A = \{0, \cdots, 3\}$, and $M = (3202)^*(\varepsilon, 3, 32, 320)$. The characteristic polynomial has a dominant root β which is a Perron number, but is not a Pisot number. We have $d(1,\beta) = 3203$, and $L_0 = F(D_\beta)$. By Theorem 3, Succ is a right subsequential function, although for any alphabet $D \supseteq \{0, \cdots, 4\}$ normalization is not computable by a finite 2-tape automaton because β is not a Pisot number [FSo96]. In particular, addition is not computable by a finite 2-tape automaton. □

4 Sequentiality and continuity

4.1 Odometer and automata

In this section we make a connection between right sequentiality of the successor function and continuity of the odometer as defined in [GLT95]. Note that in [GLT95] numbers are written the other way round, that is to say with the least significant digit at the left-end of the representation. We keep on writing numbers with the most significant digit at the left. Let A be an alphabet and denote by ${}^{\mathbf{N}}A$ the set of *left infinite* sequences over A, endowed with the discrete topology. The sequence $\cdots aaa$ is denoted by ${}^{\omega}a$.

As above, U is a strictly increasing sequence of integers with $u_0 = 1$, and A is the canonical alphabet associated to U, $L = L(U)$ is the set of greedy U-representations of the nonnegative integers, and M is the set of lexicographically maximum words of L. Following [GLT95], the *U-compactification* of $\mathbf{N}$ is the set $C = C(U) = \{s = (\cdots s_2 s_1 s_0) \mid \forall j \geq 0,\ s_j \cdots s_0 \in 0^*L\}$. Let $C^0 = \{s \in C \mid \exists N_s\ \forall j \geq N_s,\ s_j \cdots s_0 \notin M\}$. The *odometer* is the function τ defined on ${}^{\mathbf{N}}A$ by :

if $s \in C^0$ and $j \geq N_s$ then $\tau(s) = (\cdots s_{j+2} s_{j+1} \mathrm{Succ}\ (s_j \cdots s_0))$ (this definition does not depend on the choice of j), and if $s \in C \setminus C^0$, then $\tau(s) = {}^{\omega}0$.

Remark that, if we take for U the classical K-ary system, where K is an integer ≥ 2, $u_n = K^n$, and $C(U)$ is the set of K-adic integers.

In the sequel we suppose that L is recognizable by a finite automaton. Let $\mathcal{L} = (Q, A, E, r_0, T)$ be the minimal deterministic right automaton recognizing $L_0 = 0^*L$ and let $\mathcal{M} = (Q_1, A, E_1, i_0, T_1)$ be the minimal deterministic right automaton recognizing M : the set of states is $Q_1 = \{[f]_M \mid f \in A^*\}$ (there might exists a sink σ), $i_0 = \{[\varepsilon]_M\}$, the set of terminal states is $T_1 = \{[f]_M \mid f \in M\}$, and for every a in A, there is an edge $[f]_M \xrightarrow{a} [af]_M$.

Let us denote by $||\mathcal{L}||$ the set of left infinite sequences recognized by $\mathcal{L}$ (Büchi condition, see Section 2.3).

Lemma 7 *$C = ||\mathcal{L}||$ and $C^0 = ||\mathcal{L}|| \setminus ||\mathcal{M}||$.*

Proof. Since $\mathcal{L}$ is right deterministic and every state is terminal, $||\mathcal{L}|| = \{s = (s_j)_{j \geq 0} \mid \forall j\ s_j \cdots s_0 \in 0^*L\} = C$. As $\mathcal{M}$ is right deterministic, $||\mathcal{M}|| = \{s \mid s_j \cdots s_0 \in M$ for infinitely many j's $\} = C \setminus C^0$. ∎

For the sake of completeness, we recall the construction presented in [F96].

Theorem 4 *The successor function in the numeration system associated to U is computable by a letter-to-letter finite 2-tape automaton (which is not right subsequential in general) if and only if the set $L(U)$ is recognizable by a finite automaton.*

Proof. We construct a right (non deterministic) automaton $\mathcal{X}$ as follows. First, $\mathcal{X}$ contains both $\mathcal{L}$ and $\mathcal{M}$ as subautomata.

Now, according to Lemma 2 we distinguish two cases, according to whether the addition of 1 will produce a carry or not.

Case 1. Addition of 1 doesn't produce a carry : it means that we are considering words w having no nonempty suffix in M. The set of such words is $K = L_0 \setminus A^*(M \setminus \varepsilon)$, which is recognizable by a finite automaton. Let $\mathcal{K} = (Q_2, A, E_2, j_0, T_2)$ be the deterministic right automaton of classes mod $\sim_K$ recognizing K. Since every non empty suffix of K is again in K (the empty word is not in K), we can take $T_2 = Q_2 \setminus j_0$. We join $\mathcal{K}$ and $\mathcal{M}$ be taking $j_0 = i_0 = \{[\varepsilon]_M\}$ for initial state of $\mathcal{K}$.

Case 2. Addition of 1 produces a carry. We consider any terminal state $t = [f]_M$ of M, $f \in M$, $t \neq [\varepsilon]_M$.

1) There is no edge outgoing from t. So for any letter $a \in A$, is created a new edge in $\mathcal{X}$: $t = [f]_M \xrightarrow{a} [af]_{L_0}$, if af is in L_0, otherwise the edge is not created.

2) There is an edge outgoing from t in $\mathcal{M}$. If that edge is the first one on a path going from t to the sink σ, there is nothing to do. Otherwise, there is a path going from t to an other terminal state t' (possibly equal to t) in $\mathcal{M}$. From Proposition 3, we know that the label of any path between t and t' is of the form xy^kz, for $k \geq 0$, that is to say, $t = [f]_M$, $t' = [xy^kzf]_M$, for every $k \geq 0$. Let us rename $z = a_{n-1} \cdots a_0$, $y = a_{n+p-1} \cdots a_n$, $x = a_{n+p+m-1} \cdots a_{n+p}$, where the a_i's are in A, and $r_i = [a_{i-1} \cdots a_0 f]_M$. Then $t = r_0$, $t' = r_{n+p+m}$, $r_i \xrightarrow{a_i} r_{i+1}$ in $\mathcal{M}$, for $i \leq 0 \leq n+p+m-1$, and $r_{n+p} = r_n$.

We do the following construction : we duplicate the path from t to $r_{n+p+m-1}$, by creating new states q_i and new edges $q_i \xrightarrow{a_i} q_{i+1}$, for $0 \leq i \leq n+p+m-2$, with $q_0 = t$. Then, for any $0 \leq i \leq n+p+m-1$, for any letter $b_i < a_i$, if $[b_i a_{i-1} \cdots a_0 f]_M = \sigma$ (that is to say if $b_i a_{i-1} \cdots a_0 f$ is not left prolongeable in a word of M), and if $b_i a_{i-1} \cdots a_0 f \in L_0$, we define a new edge $q_i \xrightarrow{b_i} [b_i a_{i-1} \cdots a_0 f]_{L_0}$ which goes into the automaton $\mathcal{L}$, otherwise, we do nothing. Let Q_3 be the set of new states, and E_3 the set of new edges just defined. Now let $\mathcal{X} = (Q \cup Q_1 \cup Q_2 \cup Q_3, A, E \cup E_1 \cup E_2 \cup E_3, q_0 = [\varepsilon]_M, S)$, with the set S of terminal states being defined as follows : every state excepted the states of $\mathcal{M}$ is terminal, $S = Q \cup Q_2 \cup Q_3 \setminus [\varepsilon]_M$.

The right 2-tape automaton $\mathcal{S}$ realizing the function Succ is defined with $\mathcal{X}$ as underlying input automaton :

- any edge $p \xrightarrow{a} q$ of $\mathcal{M}$ becomes $p \xrightarrow{a/0} q$ in $\mathcal{S}$

- any edge $[\varepsilon]_M \xrightarrow{a} q$ in $\mathcal{K}$ becomes $[\varepsilon]_M \xrightarrow{a/a+1} q$ in $\mathcal{S}$
- any new edge in E_3 outgoing from a terminal state t of $\mathcal{M}$, $t \xrightarrow{a} q$, becomes $t \xrightarrow{a/a+1} q$ in $\mathcal{S}$
- all the other edges $p \xrightarrow{a} q$ become $p \xrightarrow{a/a} q$.

We claim that $\mathcal{S}$ realizes the function Succ : let $w/v = (w_k/v_k)\cdots(w_0/v_0)$, be the label of a path in $\mathcal{S}$ going from the initial state $[\varepsilon]_M$ to a terminal state s.
1) s is in Q_2 and w is in K, and thus in L_0; by construction of $\mathcal{S}$, $v_0 = w_0 + 1$, and for $1 \leq i \leq k$, $v_i = w_i$. Hence $v = \text{Succ }(w)$.
2) s is in $Q \cup Q_3$. Let $i \geq 1$ be the greatest index such that $w_{i-1}\cdots w_0$ is in M. Let us denote $m_j = [w_j \cdots w_0]_M$ and $s_j = [w_j \cdots w_0]_{L_0}$.
a) s is in Q and thus w is in L_0. Then

$$[\varepsilon]_M \xrightarrow{w_0/0} m_0 \xrightarrow{w_1/0} \cdots \xrightarrow{w_{i-2}/0} m_{i-2} \xrightarrow{w_{i-1}/0} m_{i-1} \xrightarrow{w_i/w_i+1} s_i \xrightarrow{w_{i+1}/w_{i+1}} \cdots \xrightarrow{w_k/w_k} s_k = s$$

Thus, by Lemma 2, $v = w_k \cdots w_{i+1}(w_i+1)0^i = \text{Succ }(w)$.
b) s is in Q_3; by construction, $w_k \cdots w_i$ is a suffix of a word of M, so w is a suffix of a word of M, every suffix of w is lexicographically smaller than the word of M of same length, and so w is in L_0 by Proposition 1. As above, $v = w_k \cdots w_{i+1}(w_i+1)0^i = \text{Succ }(w)$. ■

With the slight modification that every state of $\mathcal{S}$ excepted the non-terminal states of $\mathcal{M}$ has to be chosen as terminal, it is easy to show the following.

Proposition 11 *The odometer τ is computed by $\mathcal{S}$, that is, $||\mathcal{S}|| = \tau$.*

Proof. Let $\mathcal{S}$ as above, but with set of terminal states taken as $S' = Q \cup Q_2 \cup Q_3 \cup T_1$. Let us consider an infinite path in $\mathcal{S}$ going infinitely often through a terminal state of $\mathcal{M}$. Then the label of this path is of the form $\cdots(0, s_j)\cdots(0, s_0)$ where $s = \cdots s_j \cdots s_0$ and $s_j \cdots s_0 \in M$ for infinitely many j's. Thus $\tau(s) = {}^{\omega}0$. Any other infinite path in $\mathcal{S}$ goes only finitely many times through a terminal state of $\mathcal{M}$, and is of the form $\cdots(s_{j+2}, s_{j+2})(s_{j+1}, s_{j+1} + 1)(s_j, 0)\cdots(s_0, 0)$, where j is the greatest index such that $s_j \cdots s_0 \in M$; since $\text{Succ }(s_j \cdots s_0) = 10^{j+1}$, we get $\cdots s_{j+2}(s_{j+1}+1)0^{j+1} = \tau(s)$. ■

4.2 Continuity

In [GLT95] is proved the following result (without the hypothesis of L being recognizable by a finite automaton). Let $s \in C$, and let $D(s) = \{d \mid s_d \cdots s_0 \in M\}$. Let Δ be the set of finite or empty sequences δ such that there exists $s \in C^0$ with $D(s) = \delta$.

Theorem 5 [GLT95] *The odometer τ is continuous if and only if for all finite or empty sequences $(d_0, \cdots, d_k) \in \Delta$ the set $\{d > d_k \mid (d_0, \cdots, d_k, d) \in \Delta\}$ is finite.*

Here we prove the following.

Theorem 6 *Let U such that L is recognizable by a finite automaton. Then the odometer τ associated to U is continuous if and only if the successor function Succ is right subsequential on 0^*L.*

Proof. By Theorem 2 and Theorem 5, we have to prove that $M = \bigcup_{i=1}^{i=p} y_i^* z_i \cup M_0$ (where M_0 is finite, $|y_i| = p$ and the union is disjoint) if and only if the following condition $(\mathcal{D})$ holds : for all $(d_0, \cdots, d_k) \in \Delta$ the set $\{d > d_k \mid (d_0, \cdots, d_k, d) \in \Delta\}$ is finite.

First let us suppose that M contains a subset xy^*z with $x \neq \varepsilon$, and let $s = {}^{\omega}0$. Then $D(s) = \emptyset$. For every $n \geq 0$, let $t^{(n)} = \cdots 00xy^n z$. Then for each n, $xy^n z \in M$ and $\Delta \supseteq \{|x| + np + |z| \mid n \geq 0\}$, thus condition $(\mathcal{D})$ is not satisfied.

Conversely, let $M = \bigcup_{i=1}^{i=p} y_i^* z_i \cup M_0$. Let us take $t \in C^0$, $t = \cdots t_{j+1} t_j \cdots t_{d_k} \cdots t_{d_0} \cdots t_0$, where $D(t) = (d_0, \cdots, d_k)$. We have that, for each $0 \leq l \leq k$, $t_{d_l} \cdots t_0 \in M$. Because of the form of M and the fact that the union is disjoint, there exists an i, $1 \leq i \leq p$ such that $t_{d_0} \cdots t_0 = z_i$, $t_{d_1} \cdots t_0 = y_i z_i$, $\ldots$, $t_{d_k} \cdots t_0 = y_i^k z_i$, with for $0 \leq l \leq k$, $d_l + 1 = pl + |z_i|$. Thus for all $(d_0, \cdots, d_k) \in \Delta$ the set $\{d > d_k \mid (d_0, \cdots, d_k, d) \in \Delta\} = \{p(k+1) + |z_i| - 1\}$, and thus condition $(\mathcal{D})$ is satisfied. One can remark that, when $M = \bigcup_{i=1}^{i=p} y_i^* z_i \cup M_0$, $C \setminus C^0 = \bigcup_{i=1}^{i=p} {}^{\omega}y_i z_i$. ∎

This result is not surprising since sequential functions of infinite words are continuous in some sense (see [E74]).

Example 4 Let $U = (u_n)_{n \geq 0}$ be the linear recurrent sequence defined by

$$u_n = u_{n-1} + 2u_{n-2}, \quad u_0 = 1, \; u_1 = 3.$$

The characteristic polynomial of U is $P(X) = (X+1)(X-2)$. The canonical alphabet is $A = \{0, 1, 2\}$. The language L is recognizable by a finite automaton, $0^*L = \{0,1\}^* \cup \{0,1\}^* 02(00)^*$, $M = (11)^* \cup 2(00)^*$, $C \setminus C^0 = {}^{\omega}1 = \tau^{-1}(0)$. Since $\tau^{-1}(0) \neq \emptyset$, the odometer τ is surjective. The successor function is not right subsequential from Theorem 2, and thus τ is not continuous (this fact is also easy to prove directly). The successor function is not left subsequential by Theorem 1. It is computable by a finite 2-tape automaton [F96], although on any alphabet normalization is never computable by a finite 2-tape automaton. □

A similar result to Theorem 3, but slightly weaker, is proved in [GLT95], which says : Let $\beta > 1$ and put $(e_i)_{i \geq 1} = d(1, \beta)$ if $d(1, \beta)$ is infinite, $(e_i)_{i \geq 1} = d^*(1, \beta)$ if $d(1, \beta)$ is finite (see Section 2.2). Let U such that for all $n \geq 0$, $u_n = e_1 u_{n-1} + \cdots + e_n u_0 + 1$. Then the odometer associated to U is continuous if and only if $d(1, \beta)$ is finite.

5 Conclusion

Let us recall a result from [FSo96] which says that, if U is a linear recurrent sequence of integers such that its characteristic polynomial is the minimal polynomial of a Pisot number, then normalization is computable by a finite 2-tape automaton on any alphabet of integers, and in particular addition also. This is the case for the sequences given in Lemma 4 and in Example 2.

It should be clear that there is a great difference between addition and the successor function. Of course, if addition is computable by a finite 2-tape automaton (c.f.a. for short), so is Succ. Note that addition in the standard K-ary numeration system is right subsequential (see [E74]), but addition in the Fibonacci numeration system is neither left nor right subsequential, but can be obtained as the composition of a left and of a right subsequential function, explicity given in [Sa81].

Below we summarize the examples considered in this paper. Unless explicitely stated, the results hold for any initial conditions such that $u_0 = 1$ and U is strictly increasing.

• $u_n = 2^{n+1} - 1$ (Example 1). Succ is left subsequential. Addition is not c.f.a.

• $u_n = 3u_{n-1} - u_{n-2}$ (Lemma 4). Succ is c.f.a. but neither left nor right subsequential. Addition is c.f.a.

• $u_n = u_{n-1} + u_{n-2}$ (Example 2). Succ is right subsequential. Addition is c.f.a.

• $u_n = 3u_{n-1} + 2u_{n-2} + 3u_{n-4}$ (Example 3). Succ is right subsequential. Addition is not c.f.a.

• $u_n = u_{n-1} + 2u_{n-2}$, for $n \geq 2$, $u_0 = 1$, $u_1 = 3$ (Example 4). Succ is c.f.a. but neither left nor right subsequential. Addition is not c.f.a.

I thank Paul Gastin for his set of macros “Autograph” for drawing automata.

References

[Av61] Avizienis, A. (1961), Signed-digit number representations for fast parallel arithmetic, *IRE Transactions on electronic computers* **10**, 389–400.

[Ber79] Berstel, J. (1979), *Transductions and context-free languages.* Teubner.

[B-M89] Bertrand-Mathis, A. (1989), Comment écrire les nombres entiers dans une base qui n'est pas entière, *Acta Math. Acad. Sci. Hungar.* **54**, 237–241.

[Ch77] Choffrut, Ch. (1977), Une caractérisation des fonctions séquentielles et des fonctions sous-séquentielles en tant que relations rationnelles, *Theoret. Comput. Sci.* **5**, 325–337.

[E74] Eilenberg, S. (1974), *Automata, Languages and Machines*, vol. A, Academic Press.

[Er84] Ercegovac, M.D. (1984), On-line arithmetic: An overview, *Real time Signal Processing VII* **SPIE 495**, 86–93.

[Fr85] Fraenkel, A.S. (1985), Systems of numeration, *Amer. Math. Monthly* **92(2)**, 105–114.

[F96] Frougny, Ch. (1996), On the successor function in non-classical numeration systems, *Proc. S.T.A.C.S. 96*, Lecture Notes in Computer Science **1046**, 543–553.

[FSo96] Frougny, Ch. and Solomyak B. (1996), On Representation of Integers in Linear Numeration Systems, In *Ergodic theory of* $\mathbf{Z}^d$*-Actions*, edited by M. Pollicott and K. Schmidt, London Mathematical Society Lecture Note Series **228**, Cambridge University Press, 345–368.

[FSa97] Frougny, Ch. et Sakarovitch, J. (1997), Synchronisation déterministe des automates à délai borné, *Theoret. Comput. Sci.*, to appear.

[GLT95] Grabner, P. , Liardet, P. and Tichy, R. (1995), Odometers and systems of numeration, *Acta Arithmetica* **LXXX.2**, 103–123.

[Ho95] Hollander, M. (1995), Greedy Numeration Systems and Recognizability, Preprint.

[HU79] Hopcroft, J.E. and Ullman, J.D. (1979), *Introduction to Automata Theory, Languages, and Computation*, Addison-Wesley.

[K88] Knuth, D.E. (1988), *The Art of Computer Programming, vol. 2: Seminumerical Algorithms*, 2nd ed., Addison-Wesley.

[P60] Parry, W. (1960), On the β-expansions of real numbers, *Acta Math. Acad. Sci. Hungar.* **11**, 401–416.

[R57] Rényi, A. (1957), Representations for real numbers and their ergodic properties, *Acta Math. Acad. Sci. Hungar.* **8**, 477–493.

[Sa81] Sakarovitch, J. (1981), Description des monoïdes de type fini, *E.I.K.* **8/9**, 417–434.

[Sa83] Sakarovitch, J. (1983), Deux remarques sur un théorème d'Eilenberg, *R.A.I.R.O. Informatique Théorique* **17** (1), 23–48.

[Sh92] Shallit, J. (1992), Numeration Systems, Linear Recurrences, and Regular Sets, *Proc. I.C.A.L.P. 92*, Lecture Notes in Computer Science **623**, 89–100.