Theoretical Informatics and Applications ITA, Vol. 33, N° 1, 1999, p. 79–101
Informatique Théorique et Applications

ON-LINE FINITE AUTOMATA FOR ADDITION IN SOME NUMERATION SYSTEMS

Christiane Frougny[1]

Abstract. We consider numeration systems where the base is a negative integer, or a complex number which is a root of a negative integer. We give parallel algorithms for addition in these numeration systems, from which we derive on-line algorithms realized by finite automata. A general construction relating addition in base β and addition in base β^m is given. Results on addition in base $\beta = \sqrt[m]{b}$, where b is a relative integer, follow. We also show that addition in base the golden ratio is computable by an on-line finite automaton, but is not parallelizable.

1. Introduction

A positional numeration system is given by a base and by a set of digits. In the most usual numeration systems, the base is an integer $b \geq 2$ and the digit set is $\{0, \ldots, b-1\}$. In order to represent complex numbers without separating the real and the imaginary part, one can use a complex base. For instance, it is known that every complex number can be expressed with base $i\sqrt{2}$ and digit set $\{0, 1\}$ (see [20]). For example, $-3/2 - i\sqrt{2}/2 = (101 \cdot 11)_{i\sqrt{2}}$. Recently there have been several contributions to complex arithmetic [2,10,15,18,26,31].

Among the complex bases β that have been considered so far, the most studied ones have the property that there is a power of β which is an integer, namely for base $\beta = i\sqrt{b}$, where $b \geq 2$ is an integer, $\beta^2 = -b$, and for base $\beta = -1 \pm i$, $\beta^4 = -4$ [19,28]. In those systems, the digits are integers. We might also mention that some authors have considered numeration systems with complex digits. For instance, every complex number has a representation in base 2 using digit set $\{0, 1, i, 1+i\}$ [27]. Herreros [18] has studied the representation of complex numbers using base 2 and digit set $\{0, 1, \zeta, \ldots, \zeta^5\}$, where $\zeta^6 = 1$. Robert [30] has considered base $i\sqrt{3}$ and digit set $\{0, 1, (1 + i\sqrt{3})/2\}$.

[1] Université Paris VIII and L.I.A.F.A., Case 7014, 2 place Jussieu, 75251 Paris Cedex 05, France; e-mail: `Christiane.Frougny@liafa.jussieu.fr`

In this work, we do not consider the question of the representability of the complex plane, but we focus on the addition process. Addition of two numbers in the classical b-ary numeration system, where b is an integer ≥ 2, has a linear time complexity. In order to save time, several solutions have been proposed. A celebrated one is the Avizienis signed-digit representation [3], which consists of changing the digit set. Instead of taking digits from the canonical digit set $\{0, \ldots, b-1\}$, they are taken from a balanced set of the form $\{\bar{a}, \ldots, a\}$, where $\bar{a}$ denotes the digit $-a$, a being an integer such that $b/2 < a \leq b-1$ (b has to be ≥ 3). Such a numeration system is *redundant*, that is to say, some numbers may have several representations. This property allows one to perform addition in constant time in parallel, because there is a limited carry propagation. A similar algorithm for base 2 has been devised by Chow and Robertson [8] using digit set $\{\bar{1}, 0, 1\}$. Here addition is realized in parallel with a window of size 3. In terms of automata theory, such functions are called *local*: a function is p-local if the value of an output digit is determined through a window of size p.

On-line arithmetic is the performing of arithmetic operations in Most Significant Digit First (MSDF) mode (that is, from left to right), digit serially after a certain latency delay [12]. This allows the pipelining of different operations such as addition, multiplication and division. It is also appropriate for the processing of real numbers having infinite expansions. It is well known that when multiplying two real numbers, only the left part of the result is significant.

On-line multiplication uses parallel addition, and this allows one to have a linear time algorithm for multiplication. It is then necessary to use a redundant numeration system (see [32]).

In this paper, the finite state automata is our model of computability. A function is computable by a finite automaton if it needs only a finite auxiliary storage memory, independent of the size of the data. In that setting, one knows that addition of two integers in the classical b-ary system is computable by a finite automaton but that squaring is not (see [11]). Actually, the natural finite automaton one designs to perform addition processes numbers in the Least Significant Digit First (LSDF) mode (that is, from right to left), and is called a right subsequential automaton. Moreover, one input digit gives one output digit.

On-line finite automata have been introduced by Muller [25]. They are sequential finite automata processing data in MSDF mode, and such that one input digit gives one output digit, after a certain latency delay. They are a special kind of left subsequential automata. The Avizienis and the Chow-Robertson algorithms for parallel addition in integral base lead to the construction of on-line finite automaton for addition (see [16, 25]). There is a general result which says that if a function is p-local, then it is computable by an on-line finite automaton with delay $p-1$. However, in this paper, we will always give an explicit construction of an on-line finite automaton realizing a local function, having less states than the general one.

Let us recall a result we shall use latter on: a function is said to have a bounded delay if it is realized by a finite automaton such that on every loop, the input and the output have same length. If a function has a bounded delay and if it is left (sub)sequential, then it is computable by an on-line finite automaton [16].

Parallel algorithms for addition in bases -2, $i\sqrt{2}$, $2i$ and $-1+i$ have been given in [26]. Results on addition in bases $-b$, $i\sqrt{b}$ and $-1+i$ in connexion with automata theory have been presented in [15]. Note that in the system defined by Herreros, addition can be performed in parallel [10, 18], and is computable by a right subsequential finite automaton [31]. In the Robert's system, addition is a right subsequential function [31].

In this paper, we first consider addition in negative base, and we show that properties similar to addition in the standard b-ary system are still satisfied. We then show how algorithms for addition in base $i\sqrt{b}$ can be deduced from those in base $-b$. We give the full constructions because they explain the general case. We then present a general result which says that if φ and ψ are two digit set conversions, φ in base β and ψ in base $\gamma = \beta^m$, then if ψ is local, resp. computable by an on-line automaton, resp. letter-to-letter right subsequential, so is φ (Th. 1). Conversely, if φ is computable by a letter-to-letter finite automaton so is ψ, but not on the same digit sets (Prop. 10).

From that we derive that, if b is an integer, $|b| \geq 2$, in base $\beta = \sqrt[m]{b}$, addition on $\{0, \ldots, |b|-1\}$ is a right subsequential function. If $|b| \geq 3$, let $D = \{\bar{a}, \ldots, a\}$ where $a = \lfloor |b|/2 \rfloor + 1$. Then addition in base β on D is a $(m+1)$-local function and it is computable by an on-line finite automaton with delay m. If $|b| \geq 2$ is even, let $a = |b|/2$ and $D' = \{\bar{a}, \ldots, a\}$. Then addition in base β on D' is a $(2m+1)$-local function and it is computable by an on-line finite automaton with delay $2m$. This applies in particular to base $\beta = -1 \pm i$.

We then consider a base which is not a root of an integer, namely base τ, where τ is the golden ratio. We give the explicit on-line finite automaton with delay 3 realizing addition in base τ and digit set $\{0, 1\}$. The same construction is valid for the Fibonacci numeration system. Note that addition in those systems is not computable in parallel.

2. Preliminaries

2.1. Number representations

Let β be a real or complex number such that $|\beta| > 1$, and let A be a finite set of real or complex digits. A *β-representation* of x with digits in A is a finite or a right infinite sequence $(x_k)_{k \leq n}$ with $x_k \in A$ such that $x = \sum_{k=n}^{-\infty} x_k \beta^k$. It is denoted by

$$(x_n \cdots x_0 \cdot x_{-1} x_{-2} \cdots)_\beta.$$

We will present the results for finite words, if the expansions are infinite the constructions are similar. To perform addition in a given numeration system with base β and digit set A, the process will always be the same: take two numbers $x = x_{n-1} \cdots x_0$ and $y = y_{n-1} \cdots y_0$ such that $x = \sum_{k=0}^{n-1} x_k \beta^k$, $y = \sum_{k=0}^{n-1} y_k \beta^k$, with x_k and y_k in A. In parallel, compute $z_k = x_k + y_k$. Then z_k is an element of

$B = \{c+d \mid c, d \in A\}$, and $x+y = \sum_{k=0}^{n-1} z_k \beta^k$. Addition consists of transforming the representation $z_{n-1} \cdots z_0$ of $x+y$ on B into an equivalent one $s_{n-1+l} \cdots s_0$, such that $x+y = \sum_{k=0}^{n-1+l} s_k \beta^k$, with $s_k \in A$.

2.2. Words and automata

Let us recall some definitions. More details can be found in [11]. An *alphabet* A is a finite set. A finite sequence of elements of A is called a *word*, and the set of words on A is the free monoid A^*. The *empty word* is denoted by ε. 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. The prefix (resp. suffix) is *strict* when it is not equal to the entire word w. 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 w n times to itself. The set of words of length n (resp. $\leq n$) of A^* is denoted by A^n (resp. $A^{\leq n}$).

The set of infinite sequences or infinite words on A is denoted by $A^{\mathbb{N}}$. The infinite word $vvv \cdots$ is denoted by v^ω.

An *automaton over* 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 write $p \xrightarrow{a} q$. The automaton is *finite* if Q is finite. 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* 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. 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 [11]).

Let X and Y be two alphabets. A *2-tape automaton* is an automaton over the non-free monoid $X^* \times Y^*$: $\mathcal{A} = (Q, X^* \times Y^*, E, I, T)$ is a directed graph the edges of which are labelled by elements of $X^* \times Y^*$. Words of X^* are referred to as *input words*, words of Y^* are referred to as *output words*. If $(p, (f, g), q) \in E$, we write $p \xrightarrow{f/g} q$. 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 $X^* \times Y^*$ 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. It is equivalent to saying that R is a rational subset of $X^* \times Y^*$. 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 $\mathcal{A}$ is said to be *left sequential* if edges are labelled by elements of $X \times Y^*$, if the *underlying input automaton* obtained by taking the projection over X of the label of every edge is deterministic and if every state is terminal (see [5]). A *left subsequential* 2-tape automaton is a left sequential automaton $\mathcal{A} = (Q, X \times Y^*, E, \{q_0\}, \omega)$, where ω is the *terminal* function ω: $Q \longrightarrow Y^*$, 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 $X \times Y$.

An *on-line finite automaton* with delay δ is a particular left subsequential automaton (see [16]): it is composed of a *transient* part, in which every path of length δ starting in the initial state i_0 is of the form

$$i_0 \xrightarrow{a_1/\varepsilon} i_1 \xrightarrow{a_2/\varepsilon} \cdots \xrightarrow{a_\delta/\varepsilon} i_\delta,$$

where $a_i \in X$, for $1 \le i \le \delta$, and the only edge arriving in a state $i_0, \ldots, i_{\delta-1}$ is as above, and of a *synchronous* part where edges are labelled by elements of $X \times Y$. This means that the automaton starts reading words of length $\le \delta$ outputting nothing, and after that delay, outputs serially one digit for each input digit.

The same definition works for functions of infinite words, considering infinite paths in $\mathcal{A}$, but there is no terminal function ω in that case.

All the automata considered so far work implicitly from left to right, that is to say, words are processed from left to right, but one can define similarly *right* automata processing words from right to left.

2.3. Local functions and on-line automata

The notion of local function comes from symbolic dynamics (see [4,23]), where it is defined on biinfinite words and often called a sliding block code. The definition on infinite words is the following one. A function φ: $X^{\mathbb{N}} \to Y^{\mathbb{N}}$ is said to be *p-local* if there exist a positive integer p, a function Φ from X^p to Y such that if $x = (x_i)_{i\ge 0} \in X^{\mathbb{N}}$ and $y = (y_i)_{i \ge 0} \in Y^{\mathbb{N}}$, then $y = \varphi(x)$ if and only if for every $i \ge 0$, $y_i = \Phi(x_i \cdots x_{i+p-1})$. This means that the image of x by φ is obtained through a sliding window of length p. The following result is folklore.

Fact 1. *A p-local function is computable by an on-line finite automaton with delay* $p-1$.

Proof. Let the set of states be $Q = X^{\le p-1}$ and the initial state be ε. Edges are of the form: for $a \in X$, set $\varepsilon \xrightarrow{a/\varepsilon} a$, for $d_1 \cdots d_i \in Q$ with $1 \le i \le p-2$ set $d_1 \cdots d_i \xrightarrow{a/\varepsilon} d_1 \cdots d_i a$, and for $d_1 \cdots d_{p-1} \in Q$, set $d_1 \cdots d_{p-1} \xrightarrow{a/\Phi(d_1 \cdots d_{p-1} a)} d_2 \cdots d_{p-1} a$. □

In this paper the constructions of on-line automata associated with p-local functions we give are different. Using the redundancy of representations, we can construct on-line automata with the same delay $p-1$, but having less states.

It is known that the underlying input automaton of any sequential automaton realizing a p-local function is a *p-local automaton*, that is, the arrival state of any path of length p is entirely determined by the label of the path (see [4]).

One can define local functions of finite words (see [6, 33]). A function φ: $X^* \to Y^*$ is said to be *p-local* if there exist nonnegative integer l and r such that $r+l+1=p$, and a function Φ from X^p to Y such that if $x = x_1 \cdots x_m \in X^*$ and $y = y_1 \cdots y_n \in Y^*$, then $y = \varphi(x)$ if and only if for every $1 \le i \le n$, $y_i = \Phi(x_{i-l} \cdots x_{i+r})$, with the convention that, if at the borders $x_{i-l}, \ldots, x_{k-1}$ are not defined, $\Phi(x_k \cdots x_{i+r}) = \Phi(\varepsilon \cdots \varepsilon x_k \cdots x_{i+r})$, and similarly, if $x_{j+1}, \ldots, x_{i+r}$ are not defined, $\Phi(x_{i-l} \cdots x_j) = \Phi(x_{i-l} \cdots x_j \varepsilon \cdots \varepsilon)$. A p-local function can be computed in parallel with a window of length p. It is both left and right subsequential (see [33]).

Note that, when dealing with representation of numbers, one can always assume that a representation is prefixed or suffixed by an adequate number of zeros. In the sequel, we will always consider functions such that input and output have the same length.

2.4. Standard b-ary number system

Let us recall some results on addition base b, where b is an integer ≥ 2.

Proposition 1. *1) Addition in base $\beta = b$, $b \ge 2$, with digits in $A = \{0, \ldots, b-1\}$, is a letter-to-letter right subsequential function.*
2) Suppose that $b \ge 3$, and let $D = \{\bar{a}, \ldots, a\}$ where $a = \lfloor b/2 \rfloor + 1$. Then base b addition on D is a 2-local function, and is computable by an on-line finite automaton with delay 1.
3) Suppose that $b = 2a$, a being an integer ≥ 1, and let $D = \{\bar{a}, \ldots, a\}$. Then base b addition on D is a 3-local function, and is computable by an on-line finite automaton with delay 2.

1) The fact that addition is a right subsequential function can be found in [11].
2) That addition is a 2-local function is due to Avizienis [3]. For the on-line finite automaton realizing addition in that case, see [25].
3) That addition for $b = 2$ is a 3-local function is in Chow and Robertson [8]. For the construction of the on-line automaton, see [25] and [16].

3. Negative base numeration systems

Let $\beta = -b$, where b is an integer ≥ 2. It is well known (see [20, 21, 24]) that any real number can be represented without a sign in base $-b$ with digits from the canonical digit set $A = \{0, \ldots, b-1\}$. Integers have a unique representation of the form $d_k \cdots d_0$. We show that properties satisfied by base b addition are also valid for base $-b$.

Proposition 2. *Addition in base $\beta = -b$, $b \ge 2$, with digits in $A = \{0, \ldots, b-1\}$, is a letter-to-letter right subsequential function.*

Proof. As explained above in Section 2.1, we have to convert representations over $B = \{0, \ldots, 2b-2\}$ into equivalent representations over A. Number representations are processed from right to left. We construct a right subsequential automaton $\mathcal{A} = (Q, B \times A, E, \{q_0\}, \omega)$ as follows. The set of states is $Q = \{\bar{1}, 0, 1\}$. The name of a state indicates the value of the carry. The initial state is $q_0 = 0$.

Let q be in Q and let z be in B. By the Euclidean division of $q + z$ by $\beta = -b$, there exist unique $s \in A$ and q' such that $q+z = -bq'+s$. Since $-1 \le q+z \le 2b-1$, $-2 < q' = (s - (q + z))/b \le 1$ and thus $q' \in Q$. Hence one defines an edge

$$q \xrightarrow{z/s} q' \in E \iff q + z = \beta q' + s. \tag{1}$$

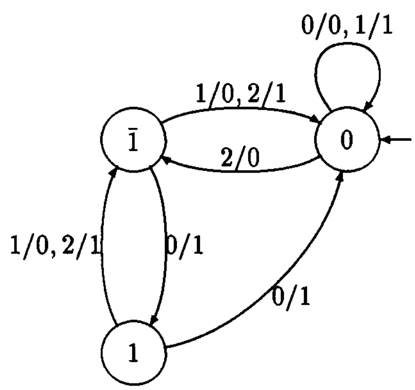
The terminal function ω is defined by $\omega(0) = \varepsilon$, $\omega(1) = 1$ and $\omega(\bar{1}) = 1(b - 1)$.

Let $z_{n-1} \cdots z_0 \in B^*$ and $N = \sum_{k=0}^{n-1} z_k \beta^k$. Starting in initial state $q_0 = 0$, and reading from right to left, we take the unique path

$$0 \xrightarrow{z_0/a_0} q_1 \xrightarrow{z_1/a_1} \cdots q_{n-1} \xrightarrow{z_n/a_n} q_n.$$

Since, for $0 \le k \le n - 1$, $q_k + z_k = \beta q_{k+1} + a_k$, we get $N = a_0 + a_1\beta + \cdots + a_{n-1}\beta^{n-1} + q_n\beta^n$. Thus the β-representation of N is $\omega(q_n)a_{n-1} \cdots a_0 \in A^*$. □

Example 1. Let $\beta = -2$ and $A = \{0, 1\}$. Here is the right subsequential automaton realizing addition in this system[1].



Let $x = 11001$, $y = 11101$, thus $x + y = 22102$. In the automaton, from right to left,

$$0 \xrightarrow{2/0} \bar{1} \xrightarrow{0/1} 1 \xrightarrow{1/0} \bar{1} \xrightarrow{2/1} 0 \xrightarrow{2/0} \bar{1}$$

and $\omega(\bar{1}) = 11$, thus $x + y = 22102 = 1101010$.

Remark 1. Addition in base $-b$ with digits in A is not left subsequential.

Proof. Let us consider $b = 2$ and $A = \{0, 1\}$. Let d be the *left-distance* defined by

$$d(v, w) = |v| + |w| - 2 \mid v \wedge w \mid$$

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

[1] I thank Paul Gastin for his set of macros Autograph.

Let $v = (01)^n 02$ and $w = (01)^{n+1}$. Then $d(v,w) = 4$. The conversion of v on A is $v' = 1(10)^{n+1}$, and that of w is $w' = w$. We have $d(v',w') = 4n+5$, thus the left-distance between v' and w' becomes unbounded when n goes to infinity, as the distance between v and w is bounded. There is a result in [9] which says that, if a function φ is left subsequential, then it has the following property: $\forall k \geq 0,\ \exists K \geq 0,\ d(v,w) \leq k \Rightarrow d(\varphi(v),\varphi(w)) \leq K$. It implies that addition on A cannot be realized by a left subsequential 2-tape automaton. □

We introduce another set of digits in order to obtain a redundant numeration system, analogous to the Avizienis signed-digit representation [3]. Let a such that $b/2 \leq a \leq b-1$ and let $D = \{\bar{a}, \ \ldots, a\}$. Then every real number has a representation in base $-b$ with digits in D. The system is redundant because $|D| = 2a+1 > b$. We consider the smallest balanced digit sets allowing one to perform addition in parallel.

Proposition 3. *Let $\beta = -b$, where b is an integer ≥ 3, and let $D = \{\bar{a}, \ \ldots, a\}$ where $a = \lfloor b/2 \rfloor + 1$. Then base $-b$ addition is a 2-local function. Addition is computable by an on-line finite automaton with delay 1.*

Proof. 1) Let $x + y = \sum_{k=0}^{n-1} z_k \beta^k$, with $z_k \in C = \{\overline{(2a)}, \ \ldots, (2a)\}$. Write z_k on the form $z_k = \beta c_{k+1} + r_k$, with the following rules: if $a \leq z_k \leq 2a$, let $c_{k+1} = \bar{1}$ and $r_k = z_k - b$; if $-2a \leq z_k \leq -a$, let $c_{k+1} = 1$ and $r_k = b + z_k$. If $|z_k| \leq a-1$, let $c_{k+1} = 0$ and $r_k = z_k$. Put $s_k = r_k + c_k$ for $0 \leq k \leq n-1$ and $s_n = c_n$. Thus $x + y = \sum_{k=0}^{n} s_k \beta^k$.

If $a \leq z_k \leq 2a$, then $a - b \leq r_k \leq 2a - b$ and $a - b - 1 \leq s_k \leq 2a - b + 1$. Since $a \leq b-1$, $s_k \leq a$, and since $2a \leq b+1$, $s_k \geq -a$, hence $s_k \in D$. The case $-2a \leq z_k \leq -a$ is symmetric, and the case $|z_k| \leq a-1$ is trivial. Thus $s_k \in D$ for $0 \leq k \leq n$. Hence s_k is a function of $z_k z_{k-1}$, and addition is 2-local.

2) To avoid overflow, we assume that input words begin with a 0. Let $z = z_k \in C$ and let $\rho(z) = (c,r) = (c_{k+1}, r_k)$ as determined in the above algorithm. We construct an on-line automaton $\mathcal{L} = (Q, C \times (D \cup \varepsilon), E, \{q_0\}, \omega)$ with delay 1 realizing addition. Let $K = \{-a+1, \ \ldots, a-1\}$. The set of states of the automaton is $Q = \{\varepsilon\} \cup K$, and the initial state is $q_0 = \varepsilon$. Synchronous edges are defined by: for any $q \in K$ and for any $z \in C$, $q \xrightarrow{z/c+q} r$ in E, with $(c,r) = \rho(z)$. Since $|c| \leq 1$ and $|q| \leq a-1$, $c+q \in D$ and $r \in K$. There is a transient edge $\varepsilon \xrightarrow{0/\varepsilon} 0$.

All edges of $\mathcal{L}$ satisfy the following condition

$$q \xrightarrow{z/d} r \in E \iff \beta q + z = \beta d + r, \tag{2}$$

that is to say, the two words qz and dr have the same numerical value in base β. The terminal function is defined by $\omega(q) = q$ for any $q \in Q$.

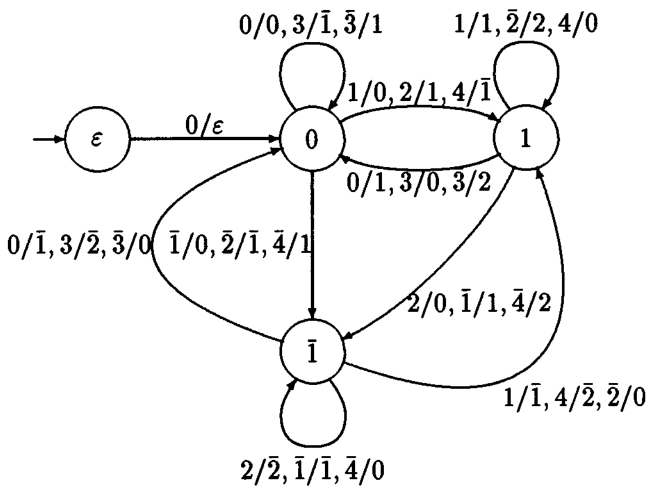
Let $z_{n-1} \cdots z_0 \in C^*$ and $N = \sum_{k=0}^{n-1} z_k \beta^k$. Starting in initial state $q_0 = \varepsilon$, and reading from left to right, we take the unique path

$$\varepsilon \xrightarrow{0/\varepsilon} 0 \xrightarrow{z_{n-1}/a_n} q_1 \xrightarrow{z_{n-2}/a_{n-1}} \cdots q_{n-1} \xrightarrow{z_0/a_1} q_n.$$

Let $\omega(q_n) = a_0$. By (2) we get $\sum_{k=0}^{n-1} z_k\beta^k = \sum_{k=0}^{n} a_k\beta^k$, with $a_k \in D$, and addition is realized by $\mathcal{L}$.

Note that the automaton $\mathcal{L}$ has $2a$ states, compare with the on-line automaton constructed in Fact 1 which has $|C|+1 = 4a+2$ states. □

Example 2. Let $\beta = -3$ and let $D = \{\bar{2}, \ldots, 2\}$. Below is the on-line finite automaton with delay 1 realizing addition in this system.



Take $x = 020\bar{2}$ and $y = 02\bar{1}\bar{2}$. Then $x+y = 04\bar{1}\bar{4}$. We have in the automaton

$$\varepsilon \xrightarrow{0/\varepsilon} 0 \xrightarrow{4/\bar{1}} 1 \xrightarrow{\bar{1}/1} \bar{1} \xrightarrow{\bar{4}/0} \bar{1}$$

and $\omega(\bar{1}) = \bar{1}$, thus $x+y = \bar{1}10\bar{1}$.

In the case that $\beta = -2$, the previous algorithm does not apply. We give an algorithm for that case as well as for any even b, which is analogous to the Chow and Robertson algorithm for base 2.

Proposition 4. *Let $\beta = -b$, where $b = 2a$, a being an integer ≥ 1, and let $D = \{\bar{a}, \ldots, a\}$. Then base $-b$ addition is a 3-local function and is computable by an on-line finite automaton with delay 2.*

Proof. 1) Let $x+y = \sum_{k=0}^{n-1} z_k\beta^k$, with $z_k \in C = \{\bar{b}, \cdots, b\}$, and let $z_k = \beta c_{k+1} + r_k$ be defined by:
If $a+1 \leq z_k \leq b$, let $c_{k+1} = \bar{1}$ and $r_k = z_k - b$; if $-b \leq z_k \leq -a-1$, let $c_{k+1} = 1$ and $r_k = b + z_k$.
If $z_k = a$ and if $z_{k-1} < 0$ then let $c_{k+1} = \bar{1}$ and $r_k = \bar{a}$, else let $c_{k+1} = 0$ and $r_k = a$.
If $z_k = -a$ and if $z_{k-1} > 0$ then let $c_{k+1} = 1$ and $r_k = a$, else let $c_{k+1} = 0$ and $r_k = -a$.
If $|z_k| \leq a-1$, let $c_{k+1} = 0$ and $r_k = z_k$.

Let $s_k = r_k + c_k$ for $0 \le k \le n-1$ and $s_n = c_n$. Clearly $x + y = \sum_{k=0}^{n} s_k \beta^k$. We have to show that $s_k \in D$. When $a+1 \le |z_k| \le b$, whatever the value of z_{k-1} is, we get $|r_k| \le a-1$ and $|c_k| \le 1$, thus $|s_k| \le a$.
If $z_k = a$, and if $z_{k-1} < 0$ then $r_k = -a$ and $c_k = 0$ or 1, thus $s_k = -a$ or $-a+1$ and thus belongs to D. If $z_k = a$ and $z_{k-1} \ge 0$, then $r_k = a$ and $c_k = -1$ or 0, and so $s_k = a-1$ or a. The case $z_k = -a$ is symmetric.
If $|z_k| \le a-1$, $r_k = z_k$ and $|c_k| \le 1$, thus $s_k \in D$. Since s_k is a function of $z_k z_{k-1} z_{k-2}$, addition is a 3-local function.

2) We construct an on-line finite automaton $\mathcal{L} = (Q, C \times (D \cup \varepsilon), E, \{q_0\}, \omega)$ with delay 2 realizing addition. Input words begin with a 0. If $z = z_k \in C$ is such that $a+1 \le |z_k| \le b$ or $|z_k| \le a-1$, we define $\rho(z) = (c, r) = (c_{k+1}, r_k)$ as in the above algorithm. If $|z| = a$ we put $\rho(z) = (c, r) = (0, z)$.

Let $K = \{(d, e) \in D \times D \mid \text{if } d = a \text{ then } e \ge 0 \text{ and if } d = -a \text{ then } e \le 0\} \setminus \{(1, a), (\bar{1}, \bar{a})\}$. These two couples are removed because they are equivalent to $(0, \bar{a})$ and $(0, a)$ respectively, since $b = 2a$. The set of states of the automaton is $Q = \{(\varepsilon, \varepsilon), (\varepsilon, 0)\} \cup K$. The initial state is $q_0 = (\varepsilon, \varepsilon)$. The synchronous part of $\mathcal{L}$ is defined this way: let $(d, e) \in K$.

— If $|e| \le a-1$, then for each $z \in C$, there is an edge $(d, e) \xrightarrow{z/d} (c+e, r)$ where $(c, r) = \rho(z)$. Since $|e| \le a-1$, $|c+e| \le a$. If $c+e = a$, then $e = a-1$ and $c = 1$, thus $r \ge 0$, and $(c+e, r) \in K$ (the symmetric case is similar).

— If $e = a$ and $z < 0$, put $(d, a) \xrightarrow{z/d-1} (c-a, r)$ where $(c, r) = \rho(z)$. Since $z < 0$, $c = 0$ or 1, and $c - a \in D$. We know that $d \ne -a$, thus $d - 1 \in D$. If $c = 0$, then $r = z < 0$, thus $(c-a, r) \in K$.

— If $e = a$ and $z \ge 0$, put $(d, a) \xrightarrow{z/d} (c+a, r)$ where $(c, r) = \rho(z)$. In that case $c = 0$ or -1, and thus $c + a \in D$. If $c = 0$ then $r = z \ge 0$, thus $(c+a, r) \in K$.

— The case $e = -a$ is symmetric: if $z > 0$, put $(d, a) \xrightarrow{z/d+1} (c+a, r)$ where $(c, r) = \rho(z)$. If $z \le 0$, put $(d, a) \xrightarrow{z/d} (c-a, r)$ where $(c, r) = \rho(z)$.
The transient part of $\mathcal{L}$ is defined by:

— $(\varepsilon, \varepsilon) \xrightarrow{0/\varepsilon} (\varepsilon, 0)$, and for $z \in C$, there is an edge $(\varepsilon, 0) \xrightarrow{z/\varepsilon} (c, r)$ where $(c, r) = \rho(z)$.
Hence, for any edge in $\mathcal{L}$

$$(d, f) \xrightarrow{z/x} (e, g) \in E \iff \beta^2 d + \beta f + z = \beta^2 x + \beta e + g \tag{3}$$

i.e. the two words dfz and xeg have the same numerical value in base β. The terminal function is defined by $\omega((d, e)) = de$ for $(d, e) \in Q$.

Let $z_{n-1} \cdots z_0 \in C^*$ and $N = \sum_{k=0}^{n-1} z_k \beta^k$. Starting in initial state $q_0 = (\varepsilon, \varepsilon)$, we take the unique path

$$(\varepsilon, \varepsilon) \xrightarrow{0/\varepsilon} (\varepsilon, 0) \xrightarrow{z_{n-1}/\varepsilon} (d_1, f_1) \xrightarrow{z_{n-2}/a_n} \cdots (d_{n-1}, f_{n-1}) \xrightarrow{z_0/a_2} (d_n, f_n).$$

Let $\omega((d_n, f_n)) = a_1 a_0$. By (3), $\sum_{k=0}^{n-1} z_k \beta^k = \sum_{k=0}^{n} a_k \beta^k$, with $a_k \in D$, and addition in base $-b$ with digit set D is realized by the on-line automaton $\mathcal{L}$, with $4a^2+1$ states. The construction of Fact 1 gives an automaton with $16a^2+12a+3$ states. □

Corollary 1. *The digit set conversion in base $-b$ between numbers written with digits in the canonical digit set $A = \{0, \ldots, b-1\}$ into their representation with digits in $D = \{\bar{a}, \ldots, a\}$, with $a = \lfloor b/2 \rfloor + 1$, or $b = 2a$, is computable in parallel in constant time.*

Proof. Since $A \subset C$, the result follows. □

Remark 2. The inverse conversion, from D to A, cannot be computed on-line, but is right subsequential.

In the same spirit, in [1] it is shown that conversion between numbers written in base b, b integer ≥ 2, with digit set $A = \{0, \ldots, b-1\}$ into their representation in base $-b$ with the same digit set is right subsequential. We now show how to convert directly a classical expansion in base b with digit set $A = \{0, \ldots, b-1\}$ into an equivalent representation in base $-b$ and digit set $D = \{\bar{a}, \ldots, a\}$, where $a+1 \leq b \leq 2a$.

Proposition 5. *Let b be an integer ≥ 2. The conversion from base b and digit set $A = \{0, \ldots, b-1\}$ into base $-b$ and digit set $D = \{\bar{a}, \ldots, a\}$, with $b/2 \leq a \leq b-1$, is a right subsequential function.*

Proof. The set of states of the automaton is $Q = \{\varepsilon, 0, 1, \bar{1}\}$. The initial state is ε. Let $z \in A$. Edges are defined by:
if $0 \leq z \leq a$, let $\varepsilon \xrightarrow{z/z} 0$; if $a+1 \leq z \leq b-1$, let $\varepsilon \xrightarrow{z/z-b} \bar{1}$;
if $0 \leq z \leq a$, let $0 \xrightarrow{z/-z} \varepsilon$; if $a+1 \leq z \leq b-1$, let $0 \xrightarrow{z/b-z} 1$;
if $0 \leq z \leq a-1$, let $1 \xrightarrow{z/z+1} 0$; if $a \leq z \leq b-1$, let $1 \xrightarrow{z/z-b+1} \bar{1}$;
if $0 \leq z \leq a-1$, let $\bar{1} \xrightarrow{z/-z-1} \varepsilon$; if $a \leq z \leq b-1$, let $\bar{1} \xrightarrow{z/-z-1} 1$.
The terminal function ω is given by $\omega(\varepsilon) = \omega(0) = \varepsilon$, $\omega(1) = 1$, and $\omega(\bar{1}) = \bar{1}$. It is straightforward to check that, since $a+1 \leq b \leq 2a$, the output is in D. □

Note that the inverse conversion is also right subsequential.

4. Base $\beta = i\sqrt{b}$

The interest of choosing a complex base and integral digits to represent complex numbers is that computations are handled in a compact way, as when using an integral base for real number computations.

Let $\beta = i\sqrt{b}$, where b is an integer ≥ 2. Any complex number is representable in base β with digits in the canonical digit set $A = \{0, \ldots, b-1\}$ (see [17,19,20]). If $b = c^2$ is a square then every Gaussian integer has a unique finite representation of the form $a_k \cdots a_0 \cdot a_{-1}$, $a_i \in A$.

Let j be an integer ≥ 0, possibly infinite, and let $n \geq 0$. Since $\beta^2 = -b$, we have

$$(a_{2n} \cdots a_0 \cdot a_{-1} \cdots a_{-2j})_\beta = (a_{2n}a_{2n-2} \cdots a_0 \cdot a_{-2} \cdots a_{-2j})_{-b} + i\sqrt{b}(a_{2n-1}a_{2n-3} \cdots a_1 \cdot a_{-1} \cdots a_{-2j+1})_{-b}.$$

Thus, if $z = x + iy \in \mathbb{C}$, x and y in $\mathbb{R}$, the β-representation of z can be obtained by intertwinning the $-b$-representation of x and the $-b$-representation of $y/\sqrt{b}$.

Base $\beta = -i\sqrt{b}$ satisfies the same properties. We treat only the case $\beta = i\sqrt{b}$. Most studied cases are $\beta = 2i$ and $A = \{0, \ldots, 3\}$, strongly related to base -4, and $\beta = i\sqrt{2}$ and $A = \{0, 1\}$ ([15, 20, 21, 26]).

We now show how properties satisfied by base $-b$ addition can be extended to base $i\sqrt{b}$.

Proposition 6. *Addition in base $\beta = i\sqrt{b}$, $b \geq 2$, with digits in $A = \{0, \ldots, b-1\}$ is a letter-to-letter right subsequential function.*

Proof. Since $\beta^2 = -b$, the automaton will be deduced from the right subsequential automaton $\mathcal{A} = (Q, B \times A, E, \{q_0\}, \omega)$ realizing addition in base $-b$ (Prop. 2).

Let $\mathcal{B} = (S, B \times A, F, \{s_0\}, \sigma)$ be defined as follows. The set of states is $S = Q \times Q$ and the initial state is $s_0 = (q_0, q_0)$. The set of edges F is defined by

$$F = \{(p,q) \xrightarrow{z/s} (q', p) \mid q \xrightarrow{z/s} q' \in E,\ p \in Q\}\cdot$$

The terminal function in a state (p, q) is defined by the β-expansion of $\beta p + q$, that is to say, $\sigma((0,0)) = \varepsilon$, $\sigma((0,1)) = 1$, $\sigma((1,0)) = 10$, $\sigma((1,1)) = 11$, $\sigma((1,\bar{1})) = 11(b-1)$, $\sigma((\bar{1},1)) = 10(b-1)1$, $\sigma((\bar{1},\bar{1})) = 11(b-1)(b-1)$.

The automaton $\mathcal{B}$ is right subsequential (and letter-to-letter). Take a word $z_{2n-1} \cdots z_0 \in B^*$ and let $Z = \sum_{k=0}^{2n-1} z_k \beta^k$. There is a path in $\mathcal{B}$

$$(q_0, q_0) \xrightarrow{z_0/a_0} (q_1, q_0) \xrightarrow{z_1/a_1} (p_1, q_1) \xrightarrow{z_2/a_2} \cdots (p_{n-1}, q_{n-1}) \xrightarrow{z_{2n-2}/a_{2n-2}} (q_n, p_{n-1}) \xrightarrow{z_{2n-1}/a_{2n-1}} (p_n, q_n)$$

if and only if there is in $\mathcal{A}$ a path

$$q_0 \xrightarrow{z_0/a_0} q_1 \xrightarrow{z_2/a_2} q_2 \xrightarrow{z_4/a_4} \cdots q_{n-1} \xrightarrow{z_{2n-2}/a_{2n-2}} q_n$$

and a path

$$q_0 \xrightarrow{z_1/a_1} p_1 \xrightarrow{z_3/a_3} p_2 \xrightarrow{z_5/a_5} \cdots p_{n-1} \xrightarrow{z_{2n-1}/a_{2n-1}} p_n.$$

Since $q_n a_{2n-2} a_{2n-4} \cdots a_2 a_0$ is the $-b$-expansion of $\sum_{k=0}^{n-1} z_{2k}(-b)^k$ and $p_n a_{2n-1} a_{2n-3} \cdots a_3 a_1$ is the $-b$-expansion of $\sum_{k=0}^{n-1} z_{2k+1}(-b)^k$, and $\sigma((p_n, q_n)) = \beta p_n + q_n$, we get that $p_n q_n a_{2n-1} a_{2n-2} \cdots a_1 a_0$ is the β-expansion of Z. Thus the right subsequential automaton $\mathcal{B}$ realizes addition in base $\beta = i\sqrt{b}$. □

Addition in base $i\sqrt{b}$ and digit set A cannot be computed on-line: consider $(0001)^n0002$ and $(0001)^n0001$ (see Rem. 1). Similarly to negative base $-b$, we consider digit sets for which addition can be parallelizable.

Proposition 7. *Let $\beta = i\sqrt{b}$, where b is an integer ≥ 3, let $a = \lfloor b/2 \rfloor + 1$ and let $D = \{\bar{a}, \cdots, a\}$. Then base $\beta = i\sqrt{b}$ addition is a 3-local function. Addition is computable by an on-line finite automaton with delay 2.*

Proof. 1) Let $z_k \in C = \{\overline{(2a)}, \cdots, (2a)\}$ and write $z_k = \beta^2 c_{k+2} + r_k = (-b)c_{k+2} + r_k$, as in Proposition 3:
if $a \leq z_k \leq 2a$, let $c_{k+2} = \bar{1}$ and $r_k = z_k - b$, if $-2a \leq z_k \leq -a$, let $c_{k+2} = 1$ and $r_k = b + z_k$, if $|z_k| \leq a-1$, let $c_{k+2} = 0$ and $r_k = z_k$. In any case, $|c_k| \leq 1$ and $|r_k| \leq b - a \leq a - 1$.

Let $s_k = r_k + c_k$ for $0 \leq k \leq n-1$, $s_n = c_n$, $s_{n+1} = c_{n+1}$, and $s_{n+2} = c_{n+2}$. We have $x + y = \sum_{i=0}^{n+2} s_k \beta^k$ with $|s_k| \leq a$. Since s_k is a function of z_k and z_{k-2}, addition is 3-local.

2) To avoid overflow, input words begin with 00. Recall that the on-line automaton $\mathcal{L} = (K \cup \varepsilon, C \times (D \cup \varepsilon), E, \{\varepsilon\}, \omega)$, where $K = \{-a+1, \ldots, a-1\}$, realizes addition in base $-b$ with digit set D, see Proposition 3.

We construct an on-line automaton with delay 2, $\mathcal{M} = (S, C \times (D \cup \varepsilon), F, \{s_0\}, \sigma)$ as follows. Let the set of states be $S = \{(\varepsilon, \varepsilon), (\varepsilon, 0)\} \cup (K \times K)$, the initial state be $s_0 = (\varepsilon, \varepsilon)$. The synchronous transitions of $\mathcal{M}$ are defined this way: for any p and q in K,

$$(q, p) \stackrel{z/q+c}{\longrightarrow} (p, r) \in F \iff q \stackrel{z/q+c}{\longrightarrow} r \in E.$$

The transient part is $(\varepsilon, \varepsilon) \stackrel{0/\varepsilon}{\longrightarrow} (\varepsilon, 0)$ and $(\varepsilon, 0) \stackrel{0/\varepsilon}{\longrightarrow} (0, 0)$. The terminal function is $\sigma(q, p) = qp$ for $(q, p) \in S$. Note that for any edge in $\mathcal{M}$

$$(q, p) \stackrel{z/x}{\longrightarrow} (q', p') \in F \iff \beta^2 q + \beta p + z = \beta^2 x + \beta q' + p' \tag{4}$$

i.e. the two words qpz and $xq'p'$ have the same numerical value in base β.

Let $z_{2n-1} \cdots z_0 \in C^*$ and $Z = \sum_{k=0}^{2n-1} z_k \beta^k$. There is a path in $\mathcal{M}$

$$(\varepsilon, \varepsilon) \stackrel{0/\varepsilon}{\longrightarrow} (\varepsilon, 0) \stackrel{0/\varepsilon}{\longrightarrow} (0, 0) \stackrel{z_{2n-1}/a_{2n+1}}{\longrightarrow} (0, q_1) \stackrel{z_{2n-2}/a_{2n}}{\longrightarrow} \cdots$$
$$(p_{n-1}, q_{n-1}) \stackrel{z_1/a_3}{\longrightarrow} (q_{n-1}, p_n) \stackrel{z_0/a_2}{\longrightarrow} (p_n, q_n)$$

if and only if there is in $\mathcal{L}$ a path

$$\varepsilon \stackrel{0/\varepsilon}{\longrightarrow} 0 \stackrel{z_{2n-2}/a_{2n}}{\longrightarrow} q_1 \stackrel{z_{2n-4}/a_{2n-2}}{\longrightarrow} \cdots q_{n-1} \stackrel{z_0/a_2}{\longrightarrow} q_n$$

and a path

$$\varepsilon \stackrel{0/\varepsilon}{\longrightarrow} 0 \stackrel{z_{2n-1}/a_{2n+1}}{\longrightarrow} p_1 \stackrel{z_{2n-3}/a_{2n-1}}{\longrightarrow} \cdots p_{n-1} \stackrel{z_1/a_3}{\longrightarrow} p_n.$$

Letting $a_0 = q_n$ and $a_1 = p_n$, we have that $\sum_{k=0}^{n-1} z_{2k}(-b)^k = \sum_{k=0}^{n} a_{2k}(-b)^k$, $\sum_{k=0}^{n-1} z_{2k+1}(-b)^k = \sum_{k=0}^{n} a_{2k+1}(-b)^k$ and thus the representation of Z on the alphabet D is $a_{2n+2}a_{2n+1}\cdots a_1a_0$. $\square$

We now consider the case where b is even.

Proposition 8. *Let $\beta = i\sqrt{b}$, where $b \geq 2$ is even, let $a = b/2$ and let $D = \{\bar{a}, \ldots, a\}$. Then base $\beta = i\sqrt{b}$ addition is a 5-local function. Addition is computable by an on-line finite automaton with delay 4.*

Proof. 1) For $z_k \in C = \{-b, \ldots, b\}$ let $z_k = \beta^2 c_{k+2} + r_k = (-b)c_{k+2} + r_k$, as in Proposition 4:
if $a+1 \leq z_k \leq 2a$, let $c_{k+2} = \bar{1}$ and $r_k = z_k - b$,
if $-2a \leq z_k \leq -a-1$, let $c_{k+2} = 1$ and $r_k = b + z_k$,
if $z_k = a$ and if $z_{k-2} < 0$, let $c_{k+2} = \bar{1}$ and $r_k = \bar{a}$ else let $c_{k+2} = 0$ and $r_k = a$,
if $z_k = \bar{a}$ and if $z_{k-2} > 0$, let $c_{k+2} = 1$ and $r_k = a$ else let $c_{k+2} = 0$ and $r_k = \bar{a}$,
if $|z_k| \leq a-1$, let $c_{k+2} = 0$ and $r_k = z_k$.
Let $s_k = r_k + c_k$ for $0 \leq k \leq n-1$, $s_n = c_n$, and $s_{n+1} = c_{n+1}$. Then $x + y = \sum_{i=0}^{n+1} s_k\beta^k$. That s_k belongs to D is proved in Proposition 4. Since s_k is a function of z_k, z_{k-2} and z_{k-4}, addition is a 5-local function.

2) Consider words with digits in C, beginning with 00. Let $\mathcal{L} = (Q, C \times (D \cup \varepsilon), E, \{(\varepsilon,\varepsilon)\}, \omega)$ be the on-line finite automaton with delay 2 realizing addition on D in base $-b$ with $K = \{(d,e) \in D \times D \mid \text{if } d = a \text{ then } e \geq 0 \text{ and if } d = -a \text{ then } e \leq 0\} \setminus \{(1,a),(\bar{1},\bar{a})\}$ and $Q = \{(\varepsilon,\varepsilon),(\varepsilon,0)\} \cup K$ (Prop. 4). We construct an on-line automaton with delay 4, $\mathcal{M} = (S, C \times (D \cup \varepsilon), F, \{s_0\}, \sigma)$ as follows.

Let us define the *shuffle* of two words by $(d,f) \sqcup\!\sqcup (e,g) = (d,e,f,g)$. Note that this is not the general shuffle product, but the internal shuffle product (see [11]). Let $K \sqcup\!\sqcup K = \{(d,f) \sqcup\!\sqcup (e,g) \mid (d,f) \in K,\ (e,g) \in K\}$. Let the set of states be $S = \{(\varepsilon,\varepsilon) \sqcup\!\sqcup (\varepsilon,\varepsilon); (\varepsilon,\varepsilon) \sqcup\!\sqcup (\varepsilon,0); (\varepsilon,0) \sqcup\!\sqcup (\varepsilon,0)\} \cup \{(\varepsilon,0) \sqcup\!\sqcup (c,r) \mid (c,r) \in \{-1,0,1\} \times D\} \cup (K \sqcup\!\sqcup K)$ and the initial state be $s_0 = (\varepsilon,\varepsilon) \sqcup\!\sqcup (\varepsilon,\varepsilon)$. The synchronous transitions of $\mathcal{M}$ are defined this way: let $(d,f) \sqcup\!\sqcup (e,g) \in K \sqcup\!\sqcup K$,

$$(d,f) \sqcup\!\sqcup (e,g) \xrightarrow{z/y} (e,g) \sqcup\!\sqcup (d',f') \in F \iff (d,f) \xrightarrow{z/y} (d',f') \in E$$

The transient part of $\mathcal{M}$ is: $(\varepsilon,\varepsilon) \sqcup\!\sqcup (\varepsilon,\varepsilon) \xrightarrow{0/\varepsilon} (\varepsilon,\varepsilon) \sqcup\!\sqcup (\varepsilon,0)$ and $(\varepsilon,\varepsilon) \sqcup\!\sqcup (\varepsilon,0) \xrightarrow{0/\varepsilon} (\varepsilon,0) \sqcup\!\sqcup (\varepsilon,0)$. For $z \in C$, let $\rho(z) = (c,r) \in \{-1,0,1\} \times D$ such that $z = \beta^2 c + r$. We define edges $(\varepsilon,0) \sqcup\!\sqcup (\varepsilon,0) \xrightarrow{z/\varepsilon} (\varepsilon,0) \sqcup\!\sqcup (c,r)$ where $(c,r) = \rho(z)$; for $z' \in C$, $(\varepsilon,0) \sqcup\!\sqcup (c,r) \xrightarrow{z'/\varepsilon} (c,r) \sqcup\!\sqcup (c',r')$ where $(c',r') = \rho(z')$.

Note that for any edge in $\mathcal{M}$

$$(d,f) \sqcup\!\sqcup (e,g) \xrightarrow{z/x} (d',f') \sqcup\!\sqcup (e',g') \in F \iff \beta^4 d + \beta^3 e + \beta^2 f + \beta g + z = \beta^4 x + \beta^3 d' + \beta^2 e' + \beta f' + g' \qquad (5)$$

i.e. the two words $defgz$ and $xd'e'f'g'$ have the same numerical value in base β.

The terminal function is $\sigma((d,f) \sqcup\!\sqcup (e,g)) = defg$ for $(d,f) \sqcup\!\sqcup (e,g) \in S$.
Let $z_{2n-1}\cdots z_0 \in C^*$ and $Z = \sum_{k=0}^{2n-1} z_k\beta^k$. There is a path in $\mathcal{M}$

$$(\varepsilon,\varepsilon) \sqcup\!\sqcup (\varepsilon,\varepsilon) \xrightarrow{0/\varepsilon} (\varepsilon,\varepsilon) \sqcup\!\sqcup (\varepsilon,0) \xrightarrow{0/\varepsilon} (\varepsilon,0) \sqcup\!\sqcup (\varepsilon,0) \xrightarrow{z_{2n-1}/\varepsilon} (\varepsilon,0) \sqcup\!\sqcup (d_1,f_1) \xrightarrow{z_{2n-2}/\varepsilon}$$

$$(d_1,f_1) \sqcup\!\sqcup (e_1,g_1) \xrightarrow{z_{2n-3}/a_{2n+1}} (e_1,g_1) \sqcup\!\sqcup (d_2,f_2) \xrightarrow{z_{2n-4}/a_{2n}} \cdots$$

$$(e_{n-1},g_{n-1}) \sqcup\!\sqcup (d_{n-1},f_{n-1}) \xrightarrow{z_1/a_5} (d_{n-1},f_{n-1}) \sqcup\!\sqcup (e_n,g_n) \xrightarrow{z_0/a_4} (e_n,g_n) \sqcup\!\sqcup (d_n,f_n)$$

if and only if there is in $\mathcal{L}$ a path

$$(\varepsilon,\varepsilon) \xrightarrow{0/\varepsilon} (\varepsilon,0) \xrightarrow{z_{2n-2}/\varepsilon} (d_1,f_1) \xrightarrow{z_{2n-4}/a_{2n}} \cdots (d_{n-1},f_{n-1}) \xrightarrow{z_0/a_4} (d_n,f_n)$$

and a path

$$(\varepsilon,\varepsilon) \xrightarrow{0/\varepsilon} (\varepsilon,0) \xrightarrow{z_{2n-1}/\varepsilon} (e_1,g_1) \xrightarrow{z_{2n-3}/a_{2n+1}} \cdots (e_{n-1},g_{n-1}) \xrightarrow{z_1/a_5} (e_n,g_n).$$

Letting $a_3 = e_n$, $a_2 = d_n$, $a_1 = g_n$ and $a_0 = f_n$, we have that $\sum_{k=0}^{n-1} z_{2k}(-b)^k = \sum_{k=0}^{n} a_{2k}(-b)^k$, $\sum_{k=0}^{n-1} z_{2k+1}(-b)^k = \sum_{k=0}^{n} a_{2k+1}(-b)^k$ and thus the representation of Z on the alphabet D is $a_{2n+1}a_{2n}\cdots a_1a_0$. □

Corollary 2. *The digit set conversion in base $\beta = i\sqrt{b}$ between numbers written with digits in the canonical digit set $A = \{0, \ldots, b-1\}$ into their representation with digits in $D = \{\bar{a}, \ldots, a\}$, whith $a = \lfloor b/2 \rfloor + 1$, or $b = 2a$, is computable in parallel in constant time.*

Remark 3. The inverse conversion, from D to A cannot be computed on-line, but it is a right subsequential function.

5. A generalization

We now consider two complex bases β and γ with the property that, for some natural $m > 0$, $\beta^m = \gamma$. Let $x = (x_{nm-1}x_{nm-2}\cdots x_0)_\beta$ be a representation in base β on a certain alphabet X of digits of a number $Z = \sum_{0\le k\le nm-1} x_k\beta^k$ (it is always possible to suppose that representations have length a multiple of m by padding with some zeroes). Let, for $0 \le j \le m-1$, $x^{(j)} = (x_{mk+j})_{0\le k\le n-1}$. Then obviously $x = \sqcup\!\sqcup_m (x^{(m-1)}, x^{(m-2)}, \ldots, x^{(1)}, x^{(0)})$ where $\sqcup\!\sqcup_m$ denotes the m-shuffle[2] of m words of same length. Hence $(x_{nm-1}x_{nm-2}\cdots x_0)_\beta = \beta^{m-1}(x^{(m-1)})_\gamma + \beta^{m-2}(x^{(m-2)})_\gamma + \cdots + \beta(x^{(1)})_\gamma + (x^{(0)})_\gamma$.

First we show a result on automata, which is more or less folklore, but we will use latter on the construction given in its proof. Let $L \subset A^*$, and denote by $L^{\sqcup\!\sqcup_m}$

[2]In Section 4, $p \sqcup\!\sqcup q$ stands for $\sqcup\!\sqcup_2 (p,q)$.

the set of m-shuffles of elements of L of same length

$$L^{\sqcup\!\sqcup_m} = \{\sqcup\!\sqcup_m (v_0^0 v_1^0 \cdots v_{n-1}^0, v_0^1 v_1^1 \cdots v_{n-1}^1, \ldots, v_0^{m-1} v_1^{m-1} \cdots v_{n-1}^{m-1}) = v_0^0 v_0^1 \cdots v_0^{m-1} v_1^0 v_1^1 \cdots v_{n-1}^0 v_{n-1}^1 \cdots v_{n-1}^{m-1} \mid n \geq 0,\ v_0^j v_1^j \cdots v_{n-1}^j \in L \text{ for } 0 \leq j \leq m-1\}.$$

Proposition 9. *If L is recognizable by a finite automaton, so is $L^{\sqcup\!\sqcup_m}$.*

Proof. Let $\mathcal{A} = (Q, A, E, I, T)$ be a finite automaton recognizing L, and let $\mathcal{B} = (S, A, F, J, U)$ as follows: $S = Q^{\sqcup\!\sqcup_m}$, $J = I^{\sqcup\!\sqcup_m}$, $U = T^{\sqcup\!\sqcup_m}$, and there is an edge

$$\sqcup\!\sqcup_m (p_0, \ldots, p_{m-1}) \xrightarrow{a} \sqcup\!\sqcup_m (p_1, \ldots, p_{m-1}, q) \in F \iff p_0 \xrightarrow{a} q \in E. \qquad (6)$$

So there exist a path in $\mathcal{B}$

$$\sqcup\!\sqcup_m (q_0^0, \ldots, q_0^{m-1}) \xrightarrow{v_0^0} \sqcup\!\sqcup_m (q_0^1, \ldots, q_0^{m-1}, q_1^0) \xrightarrow{v_0^1} \sqcup\!\sqcup_m (q_0^2, \ldots, q_0^{m-1}, q_1^0, q_1^1)$$

$$\cdots \sqcup\!\sqcup_m (q_{k-1}^j, \ldots, q_{k-1}^{m-1}, q_k^0, \ldots, q_k^{j-1}) \xrightarrow{v_{k-1}^j} \sqcup\!\sqcup_m (q_{k-1}^{j+1}, \ldots, q_{k-1}^{m-1}, q_k^0, \ldots, q_k^j)$$

$$\cdots \sqcup\!\sqcup_m (q_{n-1}^{m-1}, q_n^0, \ldots, q_n^{m-2}) \xrightarrow{v_{n-1}^{m-1}} \sqcup\!\sqcup_m (q_n^0, \ldots, q_n^{m-1})$$

if and only if for each $0 \leq j \leq m-1$ there is a path in $\mathcal{A}$

$$q_0^j \xrightarrow{v_0^j} q_1^j \xrightarrow{v_1^j} \cdots q_{n-1}^j \xrightarrow{v_{n-1}^j} q_n^j.$$

Hence $L^{\sqcup\!\sqcup_m}$ is recognized by $\mathcal{B}$. □

Recall the notations: let $x = x_{nm-1} \cdots x_0$ be a β-representation on X of $Z = \sum_{0 \leq k \leq nm-1} x_k \beta^k$, and let, for $0 \leq j \leq m-1$, $x^{(j)} = (x_{mk+j})_{0 \leq k \leq n-1}$. Let $y = y_{nm-1} \cdots y_0$ be a β-representation on Y of Z, and let, for $0 \leq j \leq m-1$, $y^{(j)} = (y_{mk+j})_{0 \leq k \leq n-1}$.

Theorem 1. *Let X and Y be two finite alphabets of digits. Let φ: $X^* \to Y^*$ be a digit set conversion in base β and let ψ: $X^* \to Y^*$ be a digit set conversion in base $\gamma = \beta^m$ such that*

$$y = \varphi(x) \iff y^{(j)} = \psi(x^{(j)}), \text{ for } 0 \leq j \leq m-1.$$

— *If ψ is p-local then φ is $(p-1)m+1$-local.*
— *If ψ is computable by a letter-to-letter finite automaton, so is φ.*
— *If ψ is computable by an on-line finite automaton with delay δ, then φ is computable by an on-line finite automaton with delay $m\delta$.*
— *If ψ is letter-to-letter right subsequential, so is φ.*

Proof. 1) Suppose that ψ is p-local: there exist l and r such that $p = l + r + 1$ and Ψ: $X^p \to Y$ such that for each $0 \leq j \leq m-1$ and $0 \leq k \leq n-1$, $y_{mk+j} = \Psi(x_{m(k+l)+j}x_{m(k+l+1)+j} \cdots\ x_{m(k-r)+j})$. Hence y_{mk+j} is determined through a window containing $x_{m(k+l)+j} \cdots x_{m(k-r)+j}$ of length $(p-1)m+1$, and φ is a $(p-1)m+1$-local function.
2) Suppose that ψ is computable by a letter-to-letter finite automaton $\mathcal{A} = (Q, X \times Y, E, I, T)$. By the same construction as in the proof of Proposition 9, we define a letter-to-letter finite automaton $\mathcal{B} = (S, X \times Y, F, J, U)$, with $S = Q^{\sqcup\sqcup_m}$, $J = I^{\sqcup\sqcup_m}$, $U = T^{\sqcup\sqcup_m}$, and edges are defined as in equation (6).

Let $x = x_{nm-1} \cdots x_0 \in X^*$ and $y = \varphi(x) = y_{nm-1} \cdots y_0 \in Y^*$. Since $y^{(j)} = \psi(x^{(j)})$ for $0 \leq j \leq m-1$, $(x^{(j)}, y^{(j)})$ is the label of a path recognized by $\mathcal{A}$. That $\mathcal{B}$ recognizes φ is proved as in Proposition 9.
3) Suppose that ψ is computable by an on-line finite automaton $\mathcal{A} = (Q, X \times (Y \cup \varepsilon), E, q_0, \omega)$ with delay δ, where ω: $Q \to Y^*$ is a partial terminal function. By the same construction as above, we define a letter-to-letter finite automaton $\mathcal{B} = (S, X \times (Y \cup \varepsilon), F, s_0, \sigma)$, with $S = Q^{\sqcup\sqcup_m}$, $s_0 = \sqcup\sqcup_m (q_0,\ \ldots, q_0)$, $\sigma(\sqcup\sqcup_m (p_0,\ \ldots, p_{m-1})) = \sqcup\sqcup_m (\omega(p_0),\ \ldots, \omega(p_{m-1}))$ for states $p_j \in Q$ such that $\omega(p_j)$ is defined. Edges are defined as in equation (6). Clearly, $\mathcal{B}$ is left subsequential. Moreover, if $\mathcal{A}$ has delay δ, every path of length $m\delta$ in $\mathcal{B}$ is labelled by couples belonging to $X \times \varepsilon$, so $\mathcal{B}$ is on-line with delay $m\delta$.
4) Suppose now that ψ is recognized by a letter-to-letter right subsequential automaton $\mathcal{A}$. The same construction as in 3) can be used, with the only change on the definition of edges, that is, equation (6) is replaced by

$$\sqcup\sqcup_m (p_0,\ \ldots, p_{m-1}) \xrightarrow{a} \sqcup\sqcup_m (q, p_1,\ \ldots, p_{m-2}) \in F \iff p_{m-1} \xrightarrow{a} q \in E. \quad (7)$$

□

One can ask about the converse problem: are properties satisfied by base β transferable to base $\gamma = \beta^m$? The answer is well-known for standard number systems: digits are to be grouped by blocks of length m. More generally, let $x = x_{nm-1}x_{nm-2} \cdots x_0$ a word of length nm in X^*. It can be written as $x = x^{[n-1]} \cdots x^{[0]}$, where, for $0 \leq k \leq n-1$, $x^{[k]} = x_{(k+1)m-1}x_{(k+1)m-2} \cdots x_{km}$ is the k-th block of length m of x (from the right). Denote by $\pi(x^{[k]})$ the value in base β of this word, *i.e.* $\pi(x^{[k]}) = x_{(k+1)m-1}\beta^{m-1} + \cdots + x_{km}$, and put $\xi_k = \pi(x^{[k]})$. Let $X_m = \{\pi(d_{m-1} \cdots d_0) \mid d_j \in X \text{ for } 0 \leq j \leq m-1\}$. Then $\xi_k \in X_m$. Analogously, put $\chi_k = \pi(y^{[k]}) \in Y_m$.

Proposition 10. *Let X and Y be two finite alphabets of digits, and let φ: $X^* \to Y^*$ be a digit set conversion in base β. Let ψ: $X_m^* \to Y_m^*$ be a digit set conversion in base $\gamma = \beta^m$ defined by*

$$\chi_{n-1} \cdots \chi_0 = \psi(\xi_{n-1} \cdots \xi_0) \iff y_{nm-1} \cdots y_0 = \varphi(x_{nm-1} \cdots x_0).$$

— *If φ is q-local with $q = (p-1)m+1$ then ψ is p-local.*
— *If φ is computable by a letter-to-letter finite automaton, so is ψ.*

Proof. 1) Let $y = \varphi(x)$, and suppose that φ is q-local: then y_k is determined by a window of length q. Hence a factor of length m, $y_k \cdots y_{k-m+1}$, is determined by a window of length $q+m-1$. It is necessary that $q+m-1$ be a multiple of m to have ψ a p-local function for some p, so $q+m-1 = pm$, and $q = m(p-1)+1$.
2) Suppose that φ is computable by a letter-to-letter finite automaton $\mathcal{B} = (Q, X \times Y, F, I, T)$ We define a letter-to-letter finite automaton $\mathcal{A} = (Q, X_m \times Y_m, E, I, T)$: let $\xi = \pi(x_{m-1} \cdots x_0) \in X_m$ and $\chi = \pi(y_{m-1} \cdots y_0) \in Y_m$. Then

$$q \xrightarrow{\xi/\chi} q' \in E \iff q \xrightarrow{x_{m-1}/y_{m-1}} q_1 \xrightarrow{x_{m-2}/y_{m-2}} \cdots q_{m-1} \xrightarrow{x_0/y_0} q' \in \mathcal{B}.$$

□

Corollary 3. *If φ is computable by an on-line finite automaton with delay $m\delta$ (resp. is letter-to-letter right subsequential) and if every element of X_m has a unique β-representation on X, then ψ is computable by an on-line finite automaton with delay δ (resp. is letter-to-letter right subsequential).*

In general, the representation on X is ambiguous. However, suppose that on Y every element has a unique β-representation, and that φ is a letter-to-letter right subsequential function satisfying a relation like equation (1), then ψ is also letter-to-letter right subsequential: suppose that, in $\mathcal{B}$ there are two paths of length m

$$q \xrightarrow{x_0/y_0} q_1 \xrightarrow{x_1/y_1} \cdots q_{m-1} \xrightarrow{x_{m-1}/y_{m-1}} q'$$

and

$$q \xrightarrow{v_0/w_0} p_1 \xrightarrow{v_1/w_1} \cdots p_{m-1} \xrightarrow{v_{m-1}/w_{m-1}} p'$$

such that $\pi(x_{m-1} \cdots x_0) = \pi(v_{m-1} \cdots v_0) = \xi$. By equation (1), we get that

$$\xi + q = x_{m-1}\beta^{m-1} + \cdots + x_0 + q = \beta^m q' + y_{m-1}\beta^{m-1} + \cdots + y_0$$

and

$$\xi + q = v_{m-1}\beta^{m-1} + \cdots + v_0 + q = \beta^m p' + w_{m-1}\beta^{m-1} + \cdots + w_0.$$

Since the β-representation on Y is unique, $p' = q'$, and $y_0 = w_0, \ldots, y_{m-1} = w_{m-1}$, hence, letting $\chi = \pi(y_{m-1} \cdots y_0)$, there is a unique edge in $\mathcal{A}$ with input label ξ

$$q \xrightarrow{\xi/\chi} q'$$

and $\mathcal{A}$ is right subsequential.

6. Applications

Results on base $\beta = i\sqrt{b}$ presented in Section 4 are of course a corollary of Theorem 1 with $m = 2$ and $\gamma = -b$. The same results hold true for base $-i\sqrt{b}$. Here we consider other roots than square ones.

6.1. Base $\beta = -1 \pm i$

Let us first recall some results on base $\beta = -b \pm i$, where b is an integer ≥ 1. It is known [19, 28] that any complex number is representable in base $\beta = -b \pm i$ with $A = \{0, \ldots, b^2\}$ for canonical digit set. Every Gaussian integer has a unique representation of the form $a_k \cdots a_0$, with $a_i \in A$. We recall the following result [31]: Addition in base $\beta = -b \pm i$, with digits in $A = \{0, \ldots, b^2\}$, is a letter-to-letter right subsequential function.

Remark that $(-1 \pm i)^4 = -4$, but that for any $b \geq 2$, there is no integer $k \neq 0$ such that $(-b \pm i)^k$ is an integer.

Proposition 11. *1) On digit set $D = \{\bar{3}, \ldots, 3\}$, addition in base $\beta = -1 \pm i$ is a 5-local function, and it is computable by an on-line finite automaton with delay 4.*
2) On digit set $D' = \{\bar{2}, \ldots, 2\}$, addition in base $\beta = -1 \pm i$ is a 9-local function, and it is computable by an on-line finite automaton with delay 8.

Proof. 1) It is a consequence of Proposition 3 and of Theorem 1 with $\gamma = -4$ and $m = 4$.
2) It is a consequence of Proposition 4 and of Theorem 1. □

Note that, since in Theorem 1 digit sets in base β and in base γ must be the same, we cannot say anything about addition in base $\beta = -1 \pm i$ on the minimally redundant digit set $\{\bar{1}, 0, 1\}$.

Remark 4. Conversion in base $-1 \pm i$ between digit set $D = \{\bar{3}, \ldots, 3\}$ or $D' = \{\bar{2}, \ldots, 2\}$ and $A = \{0, 1\}$ is not on-line computable, but is computable by a right subsequential automaton.

In reference [1] it is shown how to obtain the $(-1 + i)$-representation of a Gaussian integer from the 2-representation of its real and imaginary part by means of a right subsequential automaton.

6.2. Base $\beta = \sqrt[m]{b}$

Number representation in base $\beta = \sqrt[3]{2}$ has been studied by Körmendi in [22].

More generally, let b be an integer, $|b| \geq 2$, and let m be a positive integer. Then, regardless of the problem of which sets can be represented in base β, the following result is a simple corollary of Propositions 1–4 and Theorem 1.

Proposition 12. *Let b in $\mathbf{Z}$ such that $|b| \geq 2$, and let $\beta = \sqrt[m]{b}$.*
1) Addition in base β on $\{0, \ldots, |b| - 1\}$ is a letter-to-letter right subsequential function.
2) If $|b| \geq 3$, let $D = \{\bar{a}, \ldots, a\}$ where $a = \lfloor |b|/2 \rfloor + 1$. Then addition in base β on D is a $(m+1)$-local function. Addition is computable by an on-line finite automaton with delay m.
3) If $|b| \geq 2$ is even, let $a = |b|/2$ and $D = \{\bar{a}, \ldots, a\}$. Then addition in base β on D is a $(2m+1)$-local function. Addition is computable by an on-line finite automaton with delay $2m$.

7. Golden ratio base

This section presents results on numeration systems which are of a different kind: there is no power of the base which is an integer. Nevertheless, we think they might be of interest, because they give an example where addition is computable by an on-line finite automaton, but is not local.

Let $\beta > 1$ be a real number. Any real number $x \in [0,1]$ can be represented in base β by the following greedy algorithm [29]:
Let $x_1 = \lfloor \beta x \rfloor$ and let $r_1 = \{\beta x\}$ be the fractional part. Then iterate for $k \geq 2$, $x_k = \lfloor \beta r_{k-1} \rfloor$ and $r_k = \{\beta r_{k-1}\}$. Thus $x = \sum_{k \geq 1} x_k \beta^{-k}$, where the digits x_k are elements of the *canonical* alphabet $A = \{0, \ldots, \lfloor \beta \rfloor\}$ if $\beta \notin \mathbb{N}$, $A = \{0, \ldots, \beta - 1\}$ otherwise. The sequence $(x_k)_{k \geq 1}$ is called the β-*expansion* of x. When $\beta \notin \mathbb{N}$, a number x may have several different β-representations on A: this system is naturally redundant. The β-expansion obtained by the greedy algorithm is the greatest one in the lexicographic ordering.

Here we focus on numbers β which are defined as follows: β is the dominant root of an equation of the form

$$X^m - aX^{m-1} - aX^{m-2} - \cdots - aX - b$$

where $a \geq b \geq 1$ are integers, and $m \geq 2$. Such a root is a real number > 1. The numeration systems defined by bases of that kind are called *confluent numeration systems*. The canonical alphabet is then equal to $A = \{0, \ldots, a\}$. The most studied case is the golden ratio $\tau = (1 + \sqrt{5})/2$, with $m = 2$, $a = b = 1$.

We have proved in [13] that addition on $A = \{0, \ldots, a\}$ in a confluent numeration system is left sequential[3]. Moreover it has a bounded delay — it is realized by an automaton having all its loops letter-to-letter [14] — so by the result of [16] quoted in the introduction, it is then computable by an on-line finite automaton. We present here a direct construction of the on-line automaton for base τ.

Proposition 13. *In base* $\tau = (1 + \sqrt{5})/2$*, addition on* $\{0, 1\}$ *is computable by an on-line finite automaton with delay* 3.

Proof. Input words start with 00. We define an on-line finite automaton $\mathcal{L} = (Q, \{0,1,2\} \times (\{0,1\} \cup \varepsilon), E, \{\varepsilon\})$. The transient part of $\mathcal{L}$ is of the form
- $\varepsilon \xrightarrow{0/\varepsilon} 0 \xrightarrow{0/\varepsilon} 00$ and $00 \xrightarrow{0/\varepsilon} 000$; $00 \xrightarrow{1/\varepsilon} 001$; $00 \xrightarrow{2/\varepsilon} 002$.
In the synchronous part of $\mathcal{L}$ edges must satisfy the property

$$\begin{aligned} s_1 s_2 s_3 \xrightarrow{d/e} t_1 t_2 t_3 \in E \iff & s_1 \tau^{-1} + s_2 \tau^{-2} + s_3 \tau^{-3} + d\tau^{-4} \\ & = e\tau^{-1} + t_1 \tau^{-2} + t_2 \tau^{-3} + t_3 \tau^{-4} \end{aligned} \tag{8}$$

and for any state $s_1 s_2 s_3 \in Q$, $s_1 \tau^{-1} + s_2 \tau^{-2} + s_3 \tau^{-3} \in [0, 1[$. Edges are the following ones:

[3] The result of addition belongs to the alphabet A, but is *not* the greedy β-expansion.

— for $d \in \{0,1,2\}$, $000 \xrightarrow{d/0} 00d$

— $001 \xrightarrow{0/0} 010$; $001 \xrightarrow{1/0} 100$; $001 \xrightarrow{2/0} 101$

— $002 \xrightarrow{0/0} 11\bar{1}$; $002 \xrightarrow{1/1} 000$; $002 \xrightarrow{2/1} 001$

— for $d \in \{0,1,2\}$, $100 \xrightarrow{d/1} 00d$

— $101 \xrightarrow{0/1} 010$; $101 \xrightarrow{1/1} 100$; $101 \xrightarrow{2/1} 101$

— $010 \xrightarrow{0/0} 100$; $010 \xrightarrow{1/0} 101$; $010 \xrightarrow{2/1} 0\bar{1}2$

— $0\bar{1}2 \xrightarrow{0/0} 01\bar{1}$; $0\bar{1}2 \xrightarrow{1/0} 010$; $0\bar{1}2 \xrightarrow{2/0} 100$

— $1\bar{1}2 \xrightarrow{0/1} 01\bar{1}$; $1\bar{1}2 \xrightarrow{1/1} 010$; $1\bar{1}2 \xrightarrow{2/1} 100$

— $01\bar{1} \xrightarrow{0/0} 001$; $01\bar{1} \xrightarrow{1/0} 002$; $01\bar{1} \xrightarrow{2/0} 1\bar{1}2$

— $11\bar{1} \xrightarrow{0/1} 001$; $11\bar{1} \xrightarrow{1/1} 002$; $11\bar{1} \xrightarrow{2/1} 1\bar{1}2$. □

The on-line automaton $\mathcal{L}$ is not a local automaton, since it has two loops with same input label

$$010 \xrightarrow{1/0} 101 \xrightarrow{0/1} 010$$
$$11\bar{1} \xrightarrow{1/1} 002 \xrightarrow{0/0} 11\bar{1}.$$

In fact, we can prove the following.

Proposition 14. *Addition in base τ on alphabet $\{0,1\}$ is not a local function.*

Proof. Let us suppose that addition φ: $\{0,1,2\}^{\mathbb{N}} \to \{0,1\}^{\mathbb{N}}$ in base τ is a p-local function for some p. Thus there exists a function $\Phi : \{0,1,2\}^p \to \{0,1\}$ such that if $x = (x_i)_{i\geq 1} \in \{0,1,2\}^{\mathbb{N}}$ and $y = (y_i)_{i\geq 1} \in \{0,1\}^{\mathbb{N}}$, then $y = \varphi(x)$ if and only if for every $k \geq 0$, $y_k = \Phi(x_k \cdots x_{k+p-1})$. Since $\Phi(1^p)$ can take only value 0 or 1, for n large enough, the image of a factor containing only n ones is in $\{0,1\}^*0^l\{0,1\}^*$ or in $\{0,1\}^*1^l\{0,1\}^*$, for some large l. Since the word 0001^n0^ω has no equivalent τ-representation containing a large factor of zeroes, $\Phi(1^p)$ must be equal to 1. On the other hand, the word 0021^n20^ω has no equivalent τ-representation on $\{0,1\}$ with a large factor entirely composed of ones. Therefore addition in base τ is not local on $\{0,1\}$. □

Actually, it is possible to show that addition in base τ is 12-local on the alphabet $\{0,1,\ \ldots,12\}$ (see [7]).

These results are also valid for linear numeration systems defined by a linearly recurrent sequence $U = (u_n)_{n\geq 0}$ of the form

$$u_{n+m} = au_{n+m-1} + au_{n+m-2} + \cdots + au_{n+1} + bu_n,\ n \geq 0$$

$$u_0 = 1,\ u_i = (a+1)^i,\ 1 \leq i \leq m-1$$

where $a \geq b \geq 1$ are integers, and $m \geq 2$. Every positive integer N has a representation in this system on the alphabet $A = \{0,\ \ldots,a\}$, meaning that one can write N as $N = d_n u_n + \cdots + d_0 u_0$, with digits $d_k \in A$, using a greedy algorithm: Let n such that $u_n \leq N < u_{n+1}$; let d_n be the quotient of the division of N by u_n, and let r_n be the remainder: $d_n = q(N, u_n)$ and $r_n = r(N, u_n)$. Then iterate

$d_k = q(r_{k+1}, u_k)$ and $r_k = r(r_{k+1}, u_k)$ for $n-1 \leq k \leq 0$. The word $d_n \cdots d_0 \in A^*$ is the *normal* U-representation of N. As above addition is left subsequential [13] and has a bounded delay, so is computable by an on-line finite automaton.

For $m = 2$, $a = b = 1$, we get the *Fibonacci numeration system.*

Corollary 4. *Addition on* $\{0, 1\}$ *in the Fibonacci numeration system is computable by an on-line finite automaton with delay 3, but is not parallelizable.*

Proof. It is the same automaton as in Proposition 13 with a terminal function ω defined by: if $s_1s_2s_3 \in Q$, $\omega(s_1s_2s_3)$ is equal to the Fibonacci representation of the integer $s_1u_2 + s_2u_1 + s_3u_0$. □

References

[1] J.-P. Allouche, E. Cateland, W.J. Gilbert, H.-O. Peitgen, J.O. Shallit and G. Skordev, Automatic maps in exotic numeration systems. *Theory Comput. Syst.* **30** (1997) 285–331.

[2] T. Aoki, H. Amada and T. Higuchi, Real/complex reconfigurable arithmetic using redundant complex number systems, in *Proc. 13th Symposium on Computer Arithmetic* (1997) 200–207.

[3] A. Avizienis, Signed-digit number representations for fast parallel arithmetic. *IEE Trans. Electron. Comput.* **10** (1961) 389–400.

[4] M.P. Béal, *Codage symbolique*, Masson (1993).

[5] J. Berstel, *Transductions and Context-free Languages*, Teubner (1979).

[6] J. Berstel, Fonctions rationnelles et addition. *Actes de l'École de Printemps de Théorie des Langages*, LITP (1982) 177–183.

[7] J. Berstel, Fibonacci words — A survey, *The book of L*, Springer-Verlag (1986) 13–27.

[8] C.Y. Chow and J.E. Robertson, Logical design of a redundant binary adder, in *Proc. 4th Symposium on Computer Arithmetic* (1978) 109–115.

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

[10] J. Duprat, Y. Herreros and S. Kla, New redundant representations of complex numbers and vectors. *IEE Trans. Comput.* **C-42** (1993) 817–824.

[11] S. Eilenberg, *Automata, languages and machines*, Vol. A (Academic Press, 1974).

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

[13] Ch. Frougny, Confluent linear numeration systems. *Theoret. Comput. Sci.* **106** (1992) 183–219.

[14] Ch. Frougny, Representation of numbers and finite automata. *Math. Systems Theory* **25** (1992) 37–60.

[15] Ch. Frougny, Parallel and on-line addition in negative base and some complex number systems, in *Proc. of the Conference Euro-Par 96*, Springer, Lyon, *L.N.C.S.* **1124** (1996) 175–182.

[16] Ch. Frougny and J. Sakarovitch, Synchronisation déterministe des automates à délai borné. *Theoret. Comput. Sci.* **191** (1998) 61–77.

[17] W. Gilbert, Radix representations of quadratic fields. *J. Math. Anal. Appl.* **83** (1981) 264–274.

[18] Y. Herreros, *Contribution à l'arithmétique des ordinateurs*, Ph.D. Dissertation, I.N.P.G., Grenoble, France (1991).

[19] I. Kátai and J. Szabó, Canonical number systems. *Acta Sci. Math.* **37** (1975) 255–280.

[20] D.E. Knuth, An imaginary number system. *CACM* **3** (1960) 245–247.

[21] D.E. Knuth, *The art of computer programming, Seminumerical Algorithms*, Vol. 2, 2nd ed. (Addison-Wesley, 1988).

[22] S. Körmendi, Classical number systems in $\mathbf{Q}[\sqrt[3]{2}]$. *Acta Sci. Math.* **50** (1986) 351–357.
[23] D. Lind and B. Marcus, *An introduction to symbolic dynamics and coding*, Cambridge University Press (1995).
[24] D.W. Matula, Basic digit sets for radix representation. *JACM* **29** (1982) 1131–1143.
[25] J.-M. Muller, Some characterizations of functions computable in on-line arithmetic. *IEE Trans. Comput.* **43** (1994) 752–755.
[26] A.M. Nielsen and J.-M. Muller, Borrow-save adders for real and complex number systems, in *Proc. of the Conference Real Numbers and Computers*, Marseille (1996) 121–137.
[27] K. Pekmestzi, Complex numbers multipliers. *IEE Proc. Computers and Digital Techniques* **136** (1989) 70–75.
[28] W. Penney, A "binary" system for complex numbers. *JACM* **12** (1965) 247–248.
[29] A. Rényi, Representations for real numbers and their ergodic properties. *Acta Math. Acad. Sci. Hungar.* **8** (1957) 477–493.
[30] A. Robert, A good basis for computing with complex numbers. *El. Math.* **49** (1994) 111–117.
[31] T. Safer, Radix representations of algebraic number fields and finite automata, in *Proc. Stacs'98*, *LNCS* **1373** (1998) 356–365.
[32] K.S. Trivedi and M.D. Ercegovac, On-line algorithms for division and multiplication. *IEE Trans. Comput.* **C 26** (1977) 681–687.
[33] O. Vaysse, Addition molle et fonctions p-locales. *Semigroup Forum* **34** (1986) 157–175.

Communicated by Ch. Choffrut.
Received May, 1998. Accepted November, 1998.