Arithmetics of eventually periodic tau-adic expansions

P. Ambrož*†

1 Introduction

Let τ be the golden mean, i.e. an algebraic integer with minimal polynomial x^2-x-1, and let τ' be its algebraic conjugate. We consider eventually periodic τ-adic expansions of real numbers, that is, left infinite eventually periodic representations of real numbers in the positional numeration system with the base $\{(\tau')^n\}$, whose coefficients form sequences admissible in the usual τ-numeration system. Let $\mathcal{F}_{\text{ep}}(\tau')$ denote the set of all real numbers which have their tau-adic expansions eventually periodic to the left. It has been proved [2] that for α, a conjugate of a Pisot number β, a number x is an element of $\mathcal{F}_{\text{ep}}(\alpha)$ if and only if it is an element of $\mathbb{Q}(\alpha)$. The golden mean is a Pisot number, hence $\mathcal{F}_{\text{ep}}(\tau')$ is a ring. In this paper we give algorithms to perform ring arithmetic operations in $\mathcal{F}_{\text{ep}}(\tau')$. More precisely, we construct a transducer with a countable number of states to perform addition, subtraction is reduced to two additions, and for multiplication, we give an algorithm which uses additions and subtractions.

2 Representation of numbers

Tau-expansions. A representation in base τ (or simply *τ-representation*) of a real number $x \in \mathbb{R}_+$ is an infinite sequence $(x_i)_{k\geq i}$, $x_i \in \mathbb{Z}$ such that

$$x = x_k\tau^k + x_{k-1}\tau^{k-1} + \cdots + x_1\tau + x_0 + x_{-1}\tau^{-1} + \cdots$$

It is denoted by $(x)_\tau = x_kx_{k-1}\cdots x_1x_0\bullet x_{-1}x_{-2}\cdots$, most significant digit first.

Let $x_kx_{k-1}\ldots x_1x_0\bullet x_{-1}x_{-2}\ldots$ be a τ-representation. The *τ-value* is the function $\pi_\tau : \mathbb{Z}^{\mathbb{N}} \to \mathbb{R}$ defined by $\pi_\tau(x_kx_{k-1}\cdots) := \sum_{k\geq i} x_i\tau^i$.

Among all τ-representations of x there is one particular, called *τ-expansion*, for which the coefficients x_i are non-negative integers and $\sum_{i=-\infty}^{N} x_i\tau^i < \tau^{N+1}$ for all $-\infty < N < k$. Every $x \in \mathbb{R}_+$ has a unique τ-expansion which is found by the greedy algorithm [6]. The digits x_i obtained by the greedy algorithm are elements of the alphabet $A_\tau = \{0,1\}$, called the *canonical alphabet*. The τ-expansion of a number x is denoted by $\langle x\rangle_\tau$.

Let C be a finite alphabet of integers. The *normalization function* on $C^{\mathbb{N}}$ is the function $\nu_C : C^{\mathbb{N}} \to A_\tau^{\mathbb{N}}$ that maps a word w of $C^{\mathbb{N}}$ to $\langle x\rangle_\tau$, where $x = \pi_\tau(w)$. It is known that the normalization is computable by a finite state automaton when the base is a Pisot number [4]. Recall that a *Pisot number* is an algebraic integer $\beta > 1$ whose algebraic conjugates are in modulus less than one.

A sequence of coefficients which corresponds to some τ-expansion is usually called *admissible* in the τ-numeration system. A general condition was given by Parry [5] for the characterization of admissible sequences of coefficients in a numeration system with base $\beta > 1$. The reformulation of this condition in our simple case is as follows. Admissible τ-expansions are sequences over the alphabet A_τ not containing the word 11 as a factor.

*Doppler Institute for Mathematical Physics and Applied Mathematics and Department of Mathematics, FNSPE, Czech Technical University; LIAFA, CNRS UMR 7089, Université Paris VII. Email: ampy@linux.fjfi.cvut.cz

†The author acknowledge financial support by Czech Science Foundation GA ČR 201/05/0169, and by the grant LC00602 of the Ministry of Education, Youth and Sports of the Czech Republic.

Tau-adic expansions. A τ*-adic representation* of a real number $x \in \mathbb{R}$ is a left-infinite sequence $(x_i)_{i \geq -k}$, such that $x_i \in \mathbb{Z}$ and

$$x = \cdots + x_2(\tau')^2 + x_1\tau' + x_0 + x_{-1}(\tau')^{-1} + \cdots + x_{-k}(\tau')^{-k} .$$

It is denoted ${}_{\tau'}(x) := \cdots x_1 x_0 \bullet x_{-1} \cdots x_{-k}$. If all finite factors of the sequence $(x_i)_{i \geq -k}$ are admissible in the τ-numeration system, the sequence $(x_i)_{i \geq -k}$ is said to be the τ*-adic expansion* of the number x, and it is denoted ${}_{\tau'}\langle x \rangle$.

Analogous to the case of τ-expansions we define for τ-adic expansions the τ'-value function $\pi_{\tau'} : \mathbb{Z}^{\mathbb{N}} \to \mathbb{R}$ and the normalization function $\nu_C : C^{\mathbb{N}} \to A_\tau^{\mathbb{N}}$.

Example. Number -1 has two τ-adic expansions

$${}^{\omega}(10)0\bullet 1 = {}_{\tau'}\langle -1 \rangle \qquad \text{and} \qquad {}^{\omega}(01)0\bullet = {}_{\tau'}\langle -1 \rangle .$$

Basic properties of the τ-adic expansions of integers and rational numbers were given by the author in [1]. Let us denote by $\mathcal{F}_{\text{ep}}(\tau')$ the set of all real numbers x whose τ-adic expansion is a left-infinite eventually periodic word. It follows from a more general result in [2] that the set $\mathcal{F}_{\text{ep}}(\tau')$ is a ring. The subject of this paper is the study of addition, subtraction and multiplication in $\mathcal{F}_{\text{ep}}(\tau')$.

3 Addition

General principle. All three algorithms (i.e. the algorithm for addition, subtraction and multiplication) are composed of two parts: the first one consist in obtaining of a τ-adic representation of the result (usually non-admissible and not over the canonical alphabet), the second one consists in normalization of such a representation.

The heart of the process used during the first step lies in following two facts.

Fact 3.1. *Let* ${}^{\omega}(u_{m+p} \cdots u_{m+2}u_{m+1})u_m \cdots u_0 \bullet u_{-1} \cdots u_{-k}$ *be a* τ*-adic representation of a real number* u*. Then* ${}^{\omega}(u_{m+1}u_{m+p} \cdots u_{m+2})u_{m+1}u_m \cdots u_0 \bullet u_{-1} \cdots u_{-k}$ *is also a* τ*-adic representation of* u*.*

Fact 3.2. *Let* ${}^{\omega}(u_{m+p} \cdots u_{m+1})u_m \cdots u_0 \bullet u_{-1} \cdots u_{-k}$ *be a* τ*-adic representation of a real number* u*. Then* ${}^{\omega}((u_{m+p} \cdots u_{m+1})^l)u_m \cdots u_0 \bullet u_{-1} \ldots u_{-k}$ *is also a* τ*-adic representation of the number* u*, for any positive integer* $l \in \mathbb{N}$*.*

Let $x, y \in \mathcal{F}_{\text{ep}}(\tau')$. We want to find a representation of $z = x + y$. First we shift period of x or of y so that they start with a coefficient belonging to the same power of τ'. Then we stretch the periods to the length equal to the least common multiple of their original lengths. Finally the result is obtained by a simple digit-wise addition.

Normalization. We describe a transducer $\mathcal{T}$ — a finite non-deterministic transducer performing right to left normalization on alphabet $\{0, 1, 2\}$ in the τ-numeration system (hence also in τ-adic numeration system), with the additional condition on the input word that every coefficient 2 is surrounded by at least one 0 from each side. This condition is equivalent to the input word being a digit wise sum of two expansions in the τ-numeration system.

It is known that the sum of two tau-integers in the τ-numeration system is a number with at most two fractional digits in its τ-expansion [3]. Hence if z_k is the rightmost non-zero coefficient in the τ-adic representation of z, obtained in the previous step, the process of normalization will affect at most coefficients with indices greater than or equal to $k-2$. To avoid technical difficulties we, without loss of generality, request that the input words for $\mathcal{T}$ begin with 00 (once more we recall that the transducer is working from right to left, hence the factor 00 is supposed to be on the right end of the representation of z). In what follows coefficients of the input word will be denoted $\dots a_k a_{k-1} \dots a_1 a_0$ (hence $a_0 = a_1 = 0$) and coefficients of the output word will be $\dots b_k b_{k-1} \dots b_1 b_0$.

We denote the initial state of $\mathcal{T}$ by q_0. Then we have a starting part of the transducer

$$\begin{array}{ccc}
 & q_0 \xrightarrow{00|\varepsilon} q_1 & \\
q_1 \xrightarrow{0|100} \bar{1}1 & q_1 \xrightarrow{0|000} 00_{(a)} & q_1 \xrightarrow{1|100} 00_{(a)} \\
q_1 \xrightarrow{0|000} 00_{(b)} & q_1 \xrightarrow{1|000} 1\bar{1}_{(a)} & q_1 \xrightarrow{1|001} \bar{1}2 \\
q_1 \xrightarrow{0|010} 1\bar{2} & q_1 \xrightarrow{2|001} 01_{(d)} & q_1 \xrightarrow{2|100} 1\bar{1}_{(c)} \, .
\end{array}$$

We now describe the synchronous part of the transducer. Its states are denoted by words of length two, with $d_1 d_0$ representing the polynomial $d_1\tau + d_0$ and the signed digit -1 being denoted by $\bar{1}$. Transitions are of the form $d_1 d_0 \xrightarrow{a_k|b_k} d_1' d_0'$ if $a_k + (d_0 + d_1\tau) = b_k + (d_0'\tau + d_0'\tau^2)$.

All the states of the transducer are final. The synchronous part of the transducer $\mathcal{T}$ is drawn in Figure 1. Following proposition easily follows.

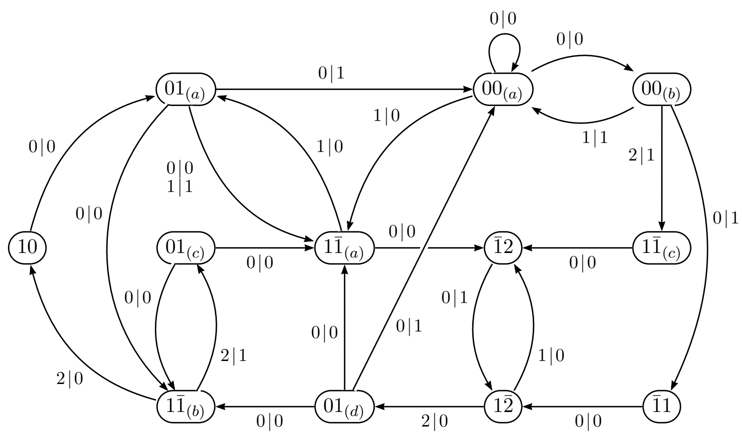


Figure 1: The synchronous part of the transducer $\mathcal{T}$.

Proposition 3.3. *Apart from the starting part the rest of $\mathcal{T}$ is a letter-to-letter transducer with zero delay (i.e. when the transducer reads the input letter a_k it writes the output letter b_k). Moreover, if $d_1' d_0'$ is the state reached after the k-th step (i.e. after writing the coefficient b_k of the output word) we have*

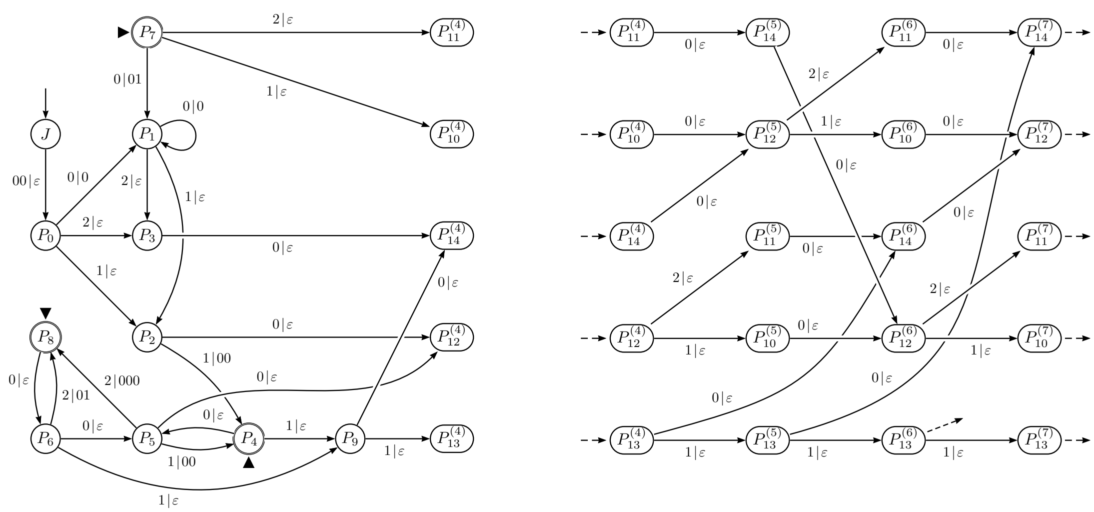
$$\sum_{i=0}^{k} a_i (\tau')^i = \left(d_1'(\tau')^2 + d_0'\tau'\right)(\tau')^k + \sum_{i=0}^{k} b_i (\tau')^i \, .$$

Remark. Should the transducer $\mathcal{T}$ be used to to normalize finite words, for simplicity we would suppose that the input and output word are of the same length, i.e. there is enough zeros in the front (recall that we are working from right to left) of each input word for the automaton arrive into the state $00_{(a)}$, this state would be the only one final state.

Now let us assume that we would like to really use the transducer $\mathcal{T}$ to perform the normalization (that is to say to implement the adding machine). A non-deterministic transducer over infinite words does not seem to be the best suitable machine for this task. Unfortunately the usual subset construction used to determinize transducers requires the input automaton to be subsequential, this is not true in our case.

Let us for a while forget that our function is not sequential and apply the determinization algorithm to transducer $\mathcal{T}$. We obtain a transducer, say $\mathcal{T}_{\text{det}}$, with an infinite (but countable) number of states. However, while performing the algorithm it is easy to see that the resulting transducer $\mathcal{T}_{\text{det}}$ is virtually composed of two parts: "non-repeating part" (counting 11 states) and the "repeating part" (the rest of the transducer).

The "non-repeating" part of the transducer $\mathcal{T}_{\text{det}}$ is in Figure 2.1. The states $P_{10}^{(4)}$ to $P_{14}^{(4)}$ are the first states of the repeating part, the state J is the only one initial state and the triangle-marked states P_4, P_7 and P_8 are the states where edges returning from the repeating part reenter the non-repeating part (see below).



2.1: Non-repeating part of $\mathcal{T}_{\text{det}}$

2.2: Repeating part of $\mathcal{T}_{\text{det}}$

Figure 2: Parts of the transducer $\mathcal{T}_{\text{det}}$

The states of the repeating part of the transducer $\mathcal{T}_{\text{det}}$ are denoted by symbols $P_j^{(i)}$. The subscript indicates the type of the state, i.e. the set of states of transducer $\mathcal{T}$ "contained" in state P_j, whereas the superscript indicates the length of memorized words, $i = |w_k|$ for all pairs (q_k, w_k) in $P_j^{(i)}$.

We can organize the repeating part of $\mathcal{T}_{\text{det}}$ into the levels according to the superscripts of the states within, all the transitions between the states inside the repeating part have empty output word. The first four levels of the repeating part are drawn in Figure 2.2.

In addition to the edges inside the repeating part, there are also the so-called "return edges", i.e. the edges aiming back to the non-repeating part (note that for clarity reasons these are not drawn in Figure 2.2).

Transducer $\mathcal{T}_{\text{det}}$ realizes the same function as the transducer $\mathcal{T}$. Unfortunately, since the normalization function ν_C is not sequential, it has an infinite number of states; we have to deal with this fact.

It turns out that if we get rid of a few particular cases, which can be treated separately, only a finite portion of the transducer $\mathcal{T}_{\text{det}}$ will be actually used during the normalization.

Obviously, the cases resulting in the use of the whole transducer $\mathcal{T}_{\text{det}}$ (or strictly speaking use of an infinite number of different states of $\mathcal{T}_{\text{det}}$) are those for which a prefix of the input word is an input label of some path in the repeating part of $\mathcal{T}_{\text{det}}$, which never uses any return edge (i.e. never returns back to the non-repeating part). We will treat these cases first.

Since all the edges inside the repeating part of $\mathcal{T}_{\text{det}}$ have empty output and so the input using infinite part of $\mathcal{T}_{\text{det}}$ will cause the transducer only to read and write nothing from some coefficient on, we will call those cases as "infinite reading" cases. They are simply characterized by the following lemma.

Lemma 3.4. *The prefix of an eventually periodic input word triggering the infinite reading in $\mathcal{T}_{\text{det}}$ is one of the following words ${}^{\omega}(1)$, ${}^{\omega}(01)$, ${}^{\omega}(002)$, ${}^{\omega}(\{01, 002\}^*)$.*

There are four simple pre-processing transformations that turn any input word triggering infinite reading either directly into its τ-adic expansion or into a finite word, which can be simply normalized.

In all other cases, only a finite portion of the transducer $\mathcal{T}_{\text{det}}$ is used during the normalization. Obviously, the computation can enter the repeating part of the transducer, but since the prefix of the input word is not any of those from Lemma 3.4, every time the computation enters the repeating part it eventually uses some return edge to go back to the non-repeating part. Indeed, we need only a finite portion of $\mathcal{T}_{\text{det}}$ to normalize these representations.

Therefore we, in fact, have a deterministic transducer performing normalization. Since the result is eventually periodic, the computation done by the transducer can be stopped after a finite number of steps.

4 Subtraction

Let $x, y \in \mathcal{F}_{\text{ep}}(\tau')$. We want to find the τ-adic expansion of $x - y$. The first step is the same as for the addition. By simple digit wise subtraction (using Facts 3.1 and 3.2) we find a τ-adic representation of $x - y$, we will denote it by $z = {}_{\tau'}(x - y)$. Obviously, the coefficients of z are from the alphabet $\{-1, 0, 1\}$. Without loss of generality, we can suppose that z has no fractional part. Normalization is then done through the use of following algorithm.

Algorithm. *Let z be a τ-adic representation of $x - y$.*

1. *We define a partition of z into three other representations u, v_{odd} and v_{even}, such that this partition preserves the numerical value $\pi_{\tau'}(z) = \pi_{\tau'}(u) + \pi_{\tau'}(v_{\text{odd}}) + \pi_{\tau'}(v_{\text{even}})$ and*
   - *u is obtained from z by putting all the negative coefficients equal to zero*
   - *v_{odd} is obtained from z by keeping only negative coefficients which belong to the odd powers of τ'*
   - *v_{even} is obtained from z by keeping only negative coefficients which belong to the even powers of τ'*
2. *We modify v_{odd} and v_{even} by transformation*

$$\widehat{v}_{\text{odd}} := v_{\text{odd}} + {}^{\omega}(10)\bullet 11 \qquad \textit{and} \qquad \widehat{v}_{\text{even}} := v_{\text{even}} + {}^{\omega}(01)\bullet 1\,, \tag{1}$$

   which does not change numerical values of the representations, since both added sequences are representations of zero. We have $u, \widehat{v}_{\text{odd}}, \widehat{v}_{\text{even}} \in \{0,1\}^{\mathbb{N}}$ and $\pi_{\tau'}(u), \pi_{\tau'}(\widehat{v}_{\text{odd}}), \pi_{\tau'}(\widehat{v}_{\text{even}}) \in \mathcal{F}_{\text{ep}}(\tau')$.

3. *A τ-adic expansion of $x - y$ is obtained by performing two consecutive additions $(u + \widehat{v}_{\text{odd}}) + \widehat{v}_{\text{even}}$.*

5 Multiplication

The operation of multiplication is different from addition and subtraction. Even though the usual naive way to perform the multiplication — a series of successive additions — seems to be an infinite process, in the case of eventually periodic expansions it can be used, but it needs more careful investigation.

We start with the simplest case of two purely periodic τ-adic expansions, say ${}_{\tau'}\langle x\rangle = {}^{\omega}(x_k \dots x_0)\bullet$ and ${}_{\tau'}\langle y\rangle = {}^{\omega}(y_l \dots y_0)\bullet$.

At first, let us assume that there is only one non-zero coefficient in ${}_{\tau'}\langle y\rangle$, say y_n, $0 \le n \le l$. In this case the multiplication will consist of successive summation of x multiplied by $(\tau')^{n+li}$ for $i \ge 0$ (i.e. summation of copies of ${}_{\tau'}\langle x\rangle$ every time shifted l positions to the left). This process produces a representation with a "re-occurring pattern": after summation of k shifted copies of ${}_{\tau'}\langle x\rangle$, the period of $(k+1)^{\text{st}}$ copy of ${}_{\tau'}\langle x\rangle$ is exactly aligned with the period of the 1^{st} copy, while in the copies in-between it appears in all other possible "shift positions"; the period of the $(k+2)^{\text{nd}}$ copy is aligned with the period of the 2^{nd} copy and so on. Therefore, the sum will be composed of blocks of the length m, where $m = \operatorname{lcm}(k, l)$, such that each coefficient in a block is by ς greater than the coefficient at the same position one block to the right, where $\varsigma = x_k + \dots + x_0$ is the sum of the coefficients in the period of ${}_{\tau'}\langle x\rangle$

$$
{}_{\tau'}(xy) = \cdots \underbrace{(z_m + 2\varsigma)\cdots(z_1 + 2\varsigma)}_{3^{\text{rd}}\text{ block}}\underbrace{(z_m + \varsigma)\cdots(z_2+\varsigma)(z_1 + \varsigma)}_{2^{\text{nd}}\text{ block}}\underbrace{z_m \cdots z_2 z_1}_{1^{\text{st}}\text{ block}}\bullet \tag{2}
$$

Example. Let ${}_{\tau'}\langle x\rangle = {}^{\omega}(10010)$ and ${}_{\tau'}\langle y\rangle = {}^{\omega}(01)$. Then $\varsigma = 2$ and the process of computing of a block-representation (2) starts as follows.

...	1 0 0 1 0 1 0 0 1 0	1 0 0 1 0 $\boxed{1\ 0\ 0\ 1\ 0}$	1 0 0 1 0 1 0 0 1 0
...	0 1 0 1 0 0 1 0 1 0	0 1 0 1 0 0 1 0 1 0	0 1 0 1 0 0 1 0
...	0 1 0 0 1 0 1 0 0 1	0 1 0 0 1 0 1 0 0 1	0 1 0 0 1 0
...	0 0 1 0 1 0 0 1 0 1	0 0 1 0 1 0 0 1 0 1	0 0 1 0
...	1 0 1 0 0 1 0 1 0 0	1 0 1 0 0 1 0 1 0 0	1 0
...	1 0 0 1 0 1 0 0 1 0	1 0 0 1 0 $\boxed{1\ 0\ 0\ 1\ 0}$	
...	0 1 0 1 0 0 1 0 1 0	0 1 0 1 0 0 1 0	
...	0 1 0 0 1 0 1 0 0 1	0 1 0 0 1 0	
...	0 0 1 0 1 0 0 1 0 1	0 0 1 0	
...	1 0 1 0 0 1 0 1 0 0	1 0	
	$\iddots$		
...	6 6 5 6 5 5 5 4 5 4	4 4 3 4 3 3 3 2 3 2	2 2 1 2 1 1 1 0 1 0

Boxes emphasize the alignment of periods of the 1^{st} and of the 6^{th} copy of the representation.

Recall that there are two τ-adic representations of zero (obtained by adding 1 to representations of -1), namely ${}^{\omega}(10)\bullet 11$ and ${}^{\omega}(01)\bullet 011$. Hence their digit-wise sum — as well as any multiple of this sum by a constant integer — is also τ-adic representation of zero. Among others ${}_{\tau'}(0) = {}^{\omega}(\varsigma\varsigma)\bullet\varsigma(2\varsigma)\varsigma$.

Let us take the block-shaped representation (2) and successively subtract from it shifted representations $\sigma^{mi}({}^{\omega}(\varsigma\varsigma)\bullet\varsigma(2\varsigma)\varsigma)$ for $i \ge 1$. One can easily see that the fractional point of i^{th} subtracted representation is aligned with the barrier between blocks i and $i+1$, for all $i \ge 1$. After first subtraction, the first block (the right most block) will be changed into $(z_m - \varsigma)(z_{m-1} - 2\varsigma)(z_{m-2} - \varsigma)(z_{m-3})\cdots(z_1)$, the second block into $z_m \cdots z_2 z_1$ (thus into the form of the first block prior to the subtraction), the third block into $(z_m + \varsigma)\cdots(z_2 + \varsigma)(z_1 + \varsigma)$ and so on. The second subtraction will not affect the first block, the second block will be changed into

$(z_m - \varsigma)(z_{m-1} - 2\varsigma)(z_{m-2} - \varsigma)(z_{m-3}) \cdots (z_1)$, the third block into $z_m \cdots z_2 z_1$ and so on. Indeed, after all the subtractions we will have an eventually periodic τ-adic representation

$$ {}_{\tau'}(xy) = {}^{\omega}\big((z_m - \varsigma)(z_{m-1} - 2\varsigma)(z_{m-2} - \varsigma)(z_{m-3}) \cdots (z_2)(z_1)\big)_{\bullet}\,, $$

which obviously still does not have to be an admissible τ-adic expansion. However, such representation can be seen as a serie of finite number of successive additions and subtractions and hence it can be done as described before.

Example (Continuation)**.** Let us take the representation obtained in the previous example and successively subtract $\sigma^{mi}({}^{\omega}(2)_{\bullet}2(4)2)$, $i \geq 1$ from it.

$\ldots$	$6\ 6\ 5\ 6\ 5\ 5\ 5\ 4\ 5\ 4$	$4\ 4\ 3\ 4\ 3\ 3\ 3\ 2\ 3\ 2$	$2\ 2\ 1\ 2\ 1\ 1\ 1\ 0\ 1\ 0$
$\ldots$	$\bar{2}\ \bar{2}\ \bar{2}\ \bar{2}\ \bar{2}\ \bar{2}\ \bar{2}\ \bar{2}\ \bar{2}\ \bar{2}$	$\bar{2}\ \bar{2}\ \bar{2}\ \bar{2}\ \bar{2}\ \bar{2}\ \bar{2}\ \bar{2}\ \bar{2}\ \bar{2}$	$\bar{2}\ \bar{4}\ \bar{2}$
$\ldots$	$4\ 4\ 3\ 4\ 3\ 3\ 3\ 2\ 3\ 2$	$2\ 2\ 1\ 2\ 1\ 1\ 1\ 0\ 1\ 0$	$0\ \bar{2}\ \bar{1}\ 2\ 1\ 1\ 1\ 0\ 1\ 0$
$\ldots$	$\bar{2}\ \bar{2}\ \bar{2}\ \bar{2}\ \bar{2}\ \bar{2}\ \bar{2}\ \bar{2}\ \bar{2}\ \bar{2}$	$\bar{2}\ \bar{4}\ \bar{2}$	
$\ldots$	$2\ 2\ 1\ 2\ 1\ 1\ 1\ 0\ 1\ 0$	$0\ \bar{2}\ \bar{1}\ 2\ 1\ 1\ 1\ 0\ 1\ 0$	$0\ \bar{2}\ \bar{1}\ 2\ 1\ 1\ 1\ 0\ 1\ 0$
$\ldots$	$\bar{2}\ \bar{4}\ \bar{2}$		
$\ldots$	$0\ \bar{2}\ \bar{1}\ 2\ 1\ 1\ 1\ 0\ 1\ 0$	$0\ \bar{2}\ \bar{1}\ 2\ 1\ 1\ 1\ 0\ 1\ 0$	$0\ \bar{2}\ \bar{1}\ 2\ 1\ 1\ 1\ 0\ 1\ 0$
			${}^{\omega}(0\ \bar{2}\ \bar{1}\ 2\ 1\ 1\ 1\ 0\ 1\ 0)$

Now suppose that there are more than one non-zero coefficient in ${}_{\tau'}\langle y \rangle$. Indeed, we can treat them one by one each time pursuing the above described algorithm for the case where ${}_{\tau'}\langle y \rangle$ has only one non-zero coefficient. Doing this we transform a multiplication of two elements form $\mathcal{F}_{\mathrm{ep}}$ into a sum of a finite number of elements from $\mathcal{F}_{\mathrm{ep}}$.

Finally, let us suppose that x or y (or both) are not purely periodic. In the case where y is not purely periodic we just have to add a finite number of shifted copies of x (i.e. of elements of $\mathcal{F}_{\mathrm{ep}}(\tau')$) to the result, whereas in the case where x is not periodic the process will be the same with the small exception that the representation of xy will have non-empty pre-period.

References

[1] P. Ambrož. *On the tau-adic expansions of real numbers.* In 'Words 2005, 5^{th} International Conference on Words, actes', S. Brlek and C. Reutenauer, (eds.), volume 36 of *Publications du LaCIM*, UQÀM (2005), 79–89.

[2] P. Ambrož and C. Frougny. *On alpha-adic expansions in Pisot bases.* Submitted to Theoret. Comp. Sci., (2006).

[3] Č. Burdík, C. Frougny, J.-P. Gazeau, and R. Krejcar. *Beta-integers as natural counting systems for quasicrystals.* J. Phys. A **31** (1998), 6449–6472.

[4] C. Frougny. *Representations of numbers and finite automata.* Math. Systems Theory **25** (1992), 37–60.

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

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