ELSEVIER

Theoretical Computer Science 222 (1999) 55–75

Theoretical Computer Science

www.elsevier.com/locate/tcs

On arithmetical first-order theories allowing encoding and decoding of lists

Patrick Cegielski [a], Denis Richard [b,*]

[a] *L.A.C.L., Université Paris XII-IUT, Route forestière Hurtault, F-77300 Fontainbleau, France*
[b] *Laboratoire de Logique, Algorithmique et Informatique de Clermont I (LLAIC 1), I.U.T. Informatique, B.P. 86, F-63172 Aubière Cedex, France*

Received March 1997; revised November 1997
Communicated by M. Nivat

Abstract

In Computer Science, n-tuples and lists are usual tools; we investigate both notions in the framework of first-order logic within the set of nonnegative integers. Gödel had firstly shown that the objects which can be defined by primitive recursion schema, can also be defined at first-order, using natural order and some coding devices for lists. Second he had proved that this encoding can be defined from addition and multiplication. We show this can be also done with addition and a weaker predicate, namely the coprimeness predicate. The theory of integers equipped with a pairing function can be decidable or not. The theory of decoding of lists (under some natural condition) is always undecidable. We distinguish the notions encoding of n-tuples and encoding of lists via some properties of decidability–undecidability. At last, we prove it is possible in some structure to encode lists although neither addition nor multiplication are definable in this structure. © 1999 Elsevier Science B.V. All rights reserved.

Résumé

On utilise couramment en informatique les n-uplets et les listes sur un ensemble donné; nous étudions ces deux notions dans le cadre de la logique du premier ordre et pour l'ensemble des entiers naturels. Gödel a montré que les objets définis par un schéma de récurrence primitive sont définissables au premier ordre avec la relation d'ordre et le codage des listes, eux-mêmes définissable avec l'addition et la multiplication; nous montrerons que ce codage peut également s'effectuer avec l'addition et un prédicat plus faible que la multiplication, à savoir la coprimarité. On montre aussi que les notions de n-uplets et de listes se distinguent par des arguments de décidabilité-indécidabilité. La théorie des entiers munis d'une fonction de couplage peut-être -ou non- décidable. Par contre la théorie du décodage des listes, soumise à une certaine condition naturelle, est toujours indécidable. On montre enfin qu'il existe des structures dans lesquelles on peut coder les listes sans pour autant que l'addition (et donc l'ordre) et la multiplication ne soient définissables. © 1999 Elsevier Science B.V. All rights reserved.

Keywords: Encoding; Decoding; Pairing function; List; First logic; Decidability; Undecidability

* Corresponding author. E-mail: richard@llaic.u-clermont.fr.

PII: S0304-3975(97)00281-8

0. Motivations

In the 1960s, John MacCarthy decided of using recursion within programming languages, and nowadays it becomes quite natural and even efficient to present objects by using a structurally inductive definition. Thanks to the notion of data structures of stacks, Edsger Dijkstra had realized in 1960 the implementation of recursion within the Algol language. From that time, the usual way of implementing recursion lies on the notion of a stack. At a more general level, Kurt Gödel, in 1931, had already proved the arithmetical definability of certain recursively defined objects within first-order logical structures via some coding of finite sequences of nonnegative integers.

This fundamental method deserves a detailed investigation and quite many questions arise: for instance, ordered pairs being the shortest nontrivial lists, one can ask whether coding lists could be reduced to pairing. The use of known coding devices for lists seems to unavoidably lead to undecidable theories. Hence, a second question consists of knowing whether this is an unavoidable consequence. In case of a positive answer, one may ask whether there exists some pairing coding with decidable associated theory on nonnegative integers. Actually, this happens and we must know whether it is always the case, but we shall show that it does not. Coding finite sequences and other kinds of abstract fundamental data structures which are of frequent use in Computer Science (such as stacks, trees, graphs) so provides a whole problematic.

A first step of this investigation consists of formalizing what we consider to be an actual coding, before looking, as a second step, at above mentioned problems we put in the framework of first-order logic. We restrict our interest only to finite sequences of nonnegative integers.

1. Definitions and notations

1.1. What to code

At first, we need to precise what objects are to be coded, i.e. what notions of ordered pair, finite sequences and what kinds of list we shall make use of.

Definition 1.1. Let A a non-empty set.

(1) *An ordered pair* on A is an element of the cartesian product (or cross product) $A \times A$ or A^2.

(2) More generally, for any nonnegative integer n, we call set of *n-tuples* on A the set A^n. By definition, the set A° only contains the empty tuple $\langle\,\rangle$.

(3) We call *set of lists* or *set of finite sequences* on A, the union $A^* = \bigcup_{n\in\mathbb{N}} A^n$. For a list L, the unique nonnegative integer n such that $L \in A^n$ is called *length* of the list, and is denoted by $lh(L)$.

(4) We call *list with header* or *headed-list* any ordered pair $\langle n, L\rangle$, where L is a list on A and $n = lh(L)$.

Mathematically speaking, there is no difference between the object we call n-tuple and the object we call list of length n. On the contrary, the set of lists is not restricted to a set of n-tuples since $A^* \neq A^n$ for each n. In the programming languages we have to distinguish in the very declaration between a n-tuple which is claimed to be an array (static data structure) and a list considered as a linked-list (dynamical data structure).

1.2. Encoding

Roughly speaking, coding n-tuples is just replacing n informations (as integers for instance) by one from which we can recover the n previous data. Of course, we need a mathematical formalization of it.

1.2.1. Encoding and integer encoding

In set theory, encoding is just one-to-one mapping. In Computer Science, encoding usually deals with integers and is an arithmetical notion. More precisely:

Definition 1.2. (1) Let E and F be sets. An *encoding* of E into F is any one-to-one mapping from E into F.

(2) Let E be a set. An *integer encoding* of E is any encoding from E into $\mathbb{N}$.

Remark. (1) Intuitively, as said above, the interest of an encoding is of summarizing several informations in a sole one, i.e. looking for the situation in which F is less complicated in a sense than E.

(2) The restriction of an encoding for lists to n-tuples is an encoding of n-tuples. On the contrary the union of encodings of n-tuples (or *n-tupling*) for all $n \in \mathbb{N}$ does not provide any encoding for lists, but just to some enumeration of lists by layers (see Example 4).

1.2.2. Integer pairing functions of nonnegative integers

Here, the base set A is $\mathbb{N}^2$.

Definition 1.3. A *pairing function* is any one-to-one mapping from $\mathbb{N}^2$ into $\mathbb{N}$, which is nothing but an integer encoding of ordered pairs.

Example 1. The first pairing function was given by Cantor in 1873:

$$J(x, y) = \frac{(x + y)(x + y + 1)}{2} + y.$$

The function J is not only an injection but is also onto, hence is a bijection.

Example 2. A second pairing function (which is also a bijection) is

$$f(x, y) = 2^x(2y + 1).$$

Remark. This later encoding seems, a priori, to provide largest codes compared to the previously presented ones, which seems to be a reason why it is avoided in practice. But in fact the largest size of this coding $f(x,y)=2^x(2y+1)$ is just an appearance, and does not exist. Obviously, for a given y_0, the mapping associating $f(x,y_0)$ to x increases faster than $J(x,y_0)$.

More generally, we want to prove that

Proposition 1.1. *Any pairing bijective function cannot be ultimately greater than another.*

Proof. In a first step, we show that there is no bijection h from $\mathbb{N}$ onto $\mathbb{N}$ satisfying the following condition (C):

$$\exists n_0 \in \mathbb{N}\ \forall n \in \mathbb{N}\ (n > n_0 \Rightarrow h(n) > n).$$

Suppose such an h exists. For $n > n_0$, we have $h(n) > n_0$. Consequently, any integer belonging to the interval $[0, n_0+1]$ would be the range of some integer of the interval $[0, n_0]$ in order to respect the surjective character of h. But this contradicts the one-to-one character of h.

Now we reduce the existence a pair $\{f,g\}$ of bijections from $\mathbb{N}^2$ onto $\mathbb{N}$, satisfying

$$\exists (x_0,y_0) \in \mathbb{N}^2\ \forall (x,y) \in \mathbb{N}^2 [(x > x_0 \text{ or } y > y_0) \Rightarrow f(x,y) < g(x,y)]$$

to the existence of a bijection h from $\mathbb{N}$ into $\mathbb{N}$ verifying condition (C) above. It suffices to take $h = g_o f^{-1}$, and $n_0 = \mathrm{Max}\{f([0,x_0] \times [0,y_0])\}$. □

Example 3. There are other encodings which were introduced for various reasons. For instance, in [21] the coding defined by $f(x,y)=(p_x)^y$, where p_x is the $(x+1)$th prime integer, is used for constructing an inner $\langle M, \oplus, \otimes \rangle$ model of Peano isomorphic to $\langle \mathbb{N}, +, \times \rangle$ for which $DEF(\mathbb{N}, +, \times) \neq DEF(\mathbb{N}, \oplus, \otimes)$. (For a given language $\mathscr{L}$, $DEF(\mathbb{N}, \mathscr{L})$ is the set of relations on $\mathbb{N}$ which are $\mathscr{L}$-definable.)

1.2.3. Integer encodings of n-tuples of nonnegative integers

The set A we are considering is $\mathbb{N}^n$ for a given integer $n > 0$.

Example 4. For a given pairing function C, one can easily define by induction the family $(C_n)_{n \geqslant 2}$ of n-tuplings as follows:

$$C_2(a,b) = C(a,b),$$

$$C_{n+1}(a_1, \ldots, a_n, a_{n+1}) = C(C_n(a_1, \ldots, a_n), a_{n+1}).$$

This has been immediately noted by Cantor in 1873.

But there is another way to encode n-tuples (for instance, a restriction of some encoding of lists, as Gödel's one in Example 5 below).

Example 5. The Cantor pairing is based on the idea of counting anti-diagonals $x+y=k$ and then of counting within a given diagonal by increasing ordinates. This geometrical device can be generalized to n-tupling, which we call Cantor n-tupling function K_n which is a bijection from $\mathbb{N}^n$ onto $\mathbb{N}$. At first, we use the level k of the hyperplane H_k of the equation $x_1+x_2+\cdots+x_n=k$ and then the level h in the hyperplane H_k, having in turn for equation $x_1+\cdots+x_{n-1}=h$, and so and up to obtaining the line $x_1+x_2=$ constant. One can check that Cantor n-tupling K_n is expressed via binomial coefficients as follows:

$$K_n(x_1,\ldots,x_n)=\binom{x_1+\cdots+x_n+n-1}{n}+\binom{x_1+\cdots+x_{n-1}+n-2}{n-1}+\cdots$$
$$+\binom{x_1+x_2+1}{2}+\binom{x_1}{1}.$$

Remark. The Cantor pairing function C (which does encode n-tuples for a given integer n) is polynomial, whose variables are the coordinates of n-tuples. Moreover, as Skolem has already noticed, this pairing function is a polynomial of degree 2 and the n-tupling function obtained from the previous, is a polynomial of degree n in Example 5 but 2^n in Example 4.

More precisely, we get two Cantor pairing functions C and C', defined by $C'(x,y)=C(y,x)$, and by generalizing, by composition, this situation there are $n!$ polynomial n-tupling functions of degree 2^n which can be constructed using permutations of coordinates. Actually, they are the only n-tupling polynomials we know. Fueter proposed, in 1923, four conjectures about the set of polynomial pairing functions (see [28, p. 24]).

1.2.4. Integer encodings of nonnegative integer lists

The base set A is $(\mathbb{N})^*$ of nonnegative integer lists, namely the finite sequences of nonnegative integers.

Example 6. The first encoding was given by Gödel in 1931:

$$f(\langle\,\rangle)=1,$$
$$f(\langle a_0,\ldots,a_n\rangle)=p_0^{a_0+1}\ldots p_n^{a_n+1},$$

where p_i denotes the $(i+1)$th prime and $p_0=2$.

Example 7. An encoding which results from Ackermann set-theoretical interpretation of $\langle\mathbb{N},+,\times\rangle$ given in 1937 (see [1]) is the following:

$$f(\langle\,\rangle)=1,$$
$$f(\langle a_0,\ldots,a_n\rangle)=\sum_{i=0}^{n}2^{\sum_{j=0}^{i}(a_j+1)}.$$

Example 8. In 1946 (cf. [19]), Quine was encoding lists of integers represented in unary expression. This encoding uses binary expansion of integers to represent, say, the sequence $(1,2,3,4,5)$ by $1101110111101111110111111$.

The aim of Quine was in particular to prove that the theory of words with concatenation is undecidable.

We have already seen that there exist n-tupling functions for a given integer n (namely the functions constructed from the pairing function C) which are polynomial. A question arises to know whether there is an encoding function of lists with header (respectively of lists) of nonnegative integers which is polynomial. To be more precise, is there a coding f of lists and a polynomial $p(X)$ of the sole variable X such that, for all nonnegative integers n and all n-tuple $\langle a_1,\dots,a_n\rangle$, we have

$$f(\langle a_1,\dots,a_n\rangle)\leqslant p(a_1+\cdots+a_n).$$

Actually, we prove below that polynomials encoding for lists do not exist:

Proposition 1.2. *There is no encoding function for lists with headers, or for lists, which is polynomial.*

Proof. Let p be such a function. For a given integer $n\geqslant 2$, there exist $(n+1)^n$ lists with length n such that all their components are not greater than n. Since f is one-to-one, we have

$$(n+1)^n\geqslant p(n^2)$$

which is impossible for any polynomial p. □

1.3. Decoding

We begin by defining an intrinsic decoding notion without explicitely refering to a given encoding function.

1.3.1. The case of ordered pairs

Definition 1.4. Let J be a given pairing function. We call the *associated depairing function*, or *associated projections*, the mappings K and L, which satisfy in $\mathbb{N}$,

$$\forall x\ \forall y\ (K(J(x,y))=x \quad \text{and} \quad L(J(x,y))=y).$$

These notations J, K, and L are conventional concerning pairing functions and associated projections since they were introduced by Robinson in 1949.

Example 9. The projections associated to the Cantor pairing function $J = C$ are easily expressed by just using an existential quantifier:

$$K(z) = x \Leftrightarrow \exists y\ [J(x,y) = z] \Leftrightarrow \exists y\ [(x+y)(x+y+1)+2y = 2z],$$
$$L(z) = y \Leftrightarrow \exists x\ [J(x,y) = z] \Leftrightarrow \exists x\ [(x+y)(x+y+1)+2y = 2z].$$

Of course, we have a more explicit way for depairing this function C: let us put $d = x + y$. We have

$$d(d+1) \leqslant 2z < (d+1)(d+2).$$

There exists a unique d satisfying the previous conditions so that $2y = 2z - d(d+1)$ providing y; then $x = d - y$.

1.3.2. The case of n-tuples

Definition 1.5. For any given integer $n \geqslant 2$, a n-tuple f_n of functions f_i^n from $\mathbb{N}$ into $\mathbb{N}$ (namely $f_n = \langle f_1^n, \ldots, f_n^n \rangle$) is called a *decoding function* for n-tuples if and only if

$$\forall \langle a_1, \ldots, a_n \rangle \in \mathbb{N}^n\ \exists c \in \mathbb{N}(f_i^n(c) = a_i)$$

for each i such that $1 \leqslant i \leqslant n$. The integer c is called a *f_n-code* of $\langle a_1, \ldots, a_n \rangle$.

This definition in a sense generalizes n-tuples Definition 1.4 concerning ordered pairs.

Example 10. The depairing functions K and L provide a family $(f_n)_{n \geqslant 1}$ of decoding functions for n-tuples:

$$f_1^1(c) = c,$$

$$f_i^2(c) = \begin{cases} K(c) & \text{if } i = 1, \\ L(c) & \text{if } i = 2, \end{cases}$$

$$f_i^{(n+1)}(c) = \begin{cases} f_i^n(K(c)) & \text{if } 1 \leqslant i \leqslant n, \\ L(c) & \text{if } i = n+1. \end{cases}$$

1.3.3. The case of lists with header (or headed lists)

Definition 1.6. A map f from $\mathbb{N}^3$ into $\mathbb{N}$ is called *a decoding headed-lists function* (abbreviated as dhl-functions) if and only if

$$\forall n \in \mathbb{N}\ \forall \langle a_0, \ldots, a_n \rangle \in \mathbb{N}^{n+1}\ \exists c \in \mathbb{N}\ \forall i \in \mathbb{N}\ (f(c,n,i) = a_i \text{ for } i \leqslant n).$$

The integer c is called an *f-code* of $\langle n+1, a_0, \ldots, a_n \rangle$.

Example 11. Such a decoding function was given at first in 1931 by Gödel, who had considered an auxiliary ternary function β, now called *Gödel β-function*, defined by

$\beta(c,d,i)$ is the remainder of c modulo $[1+(1+i)d]$.

It can be shown that, for any finite sequence $\langle a_0,\dots,a_n\rangle$ of nonnegative integers, there exist $c\in\mathbb{N}$ and $d\in\mathbb{N}$ such that $\forall i\leqslant n\ (\beta(c,d,i)=a_i)$.

We must emphasize the fact that we do not control the values of $\beta(c,d,i)$ for $i\geqslant n+1$. This provides the decoding headed lists function f determined by

$$f(a,n,i)=\beta(K(a),L(a),i).$$

Remark. In the previous definition of a decoding headed lists function, there are two side-effects worth noticing:
- on the one hand, there are integers c which are not f-codes;
- on the other, we do not control the values $f(c,n,i)$ for i greater than n.

But Definition 1.6 suffices for applications dealing with the definability within a first-order language of inductively defined notions, which was at that time the purpose of Gödel. Nevertheless, many decoding functions do have the so-called *compact support property* we define below:

Definition 1.7. A function f from $\mathbb{N}^3$ into $\mathbb{N}$ is said to have a *compact support* (*dhlcs*-function) if this function satisfies

$$\forall c\ \forall n\ \exists M\ \exists a\ \forall i\ [i>M\rightarrow f(c,n,i)=a].$$

Example 12. The decoding headed-lists function $f(c,n,i)=\beta(K(c),L(c),i)$ is a *dhlcs*-function.

Definition 1.8. One says that a *dhl*-function f from $\mathbb{N}^3$ into $\mathbb{N}$ has the *strongly compact support property* if this function satisfies

$$\forall c\ \forall n\ \forall i\ [i>n\rightarrow f(c,n,i)=0].$$

Such a function is called *a dhlScs-function.*

Example 13. The *dhl*-function $\beta(K(c),L(c),i)$ is not a *dhlScs*-function but it is easy to obtain such a function g from β itself by putting

$$g(c,n,i)=\begin{cases} f(c,n,i) & \text{if } i\leqslant n,\\ 0 & \text{if } i>n.\end{cases}$$

1.3.4. The case of nonnegative integer lists

Sometimes, it is more convenient to consider an ordered pair of functions rather than a unique *dhl*-function.

Definition 1.9. An ordered pair $\langle f,g\rangle$ of mappings from $\mathbb{N}^2$ into $\mathbb{N}$ is called an ordered pair of decoding (integer) lists function (*dl*-functional) ordered pair if and only if we have

$$\forall n\in\mathbb{N}\ \forall\langle a_0,\dots,a_n\rangle\in\mathbb{N}^{n+1}\ \exists c\in\mathbb{N}\ \forall i\in\mathbb{N}$$

$$(f(c,i)=a_i)\quad\text{when } i\leqslant n\quad\text{and}$$

$$g(c,i)=\begin{cases}1 & \text{if } i\leqslant n,\\ 0 & \text{otherwise.}\end{cases}$$

The integer c is called an $\langle f,g\rangle$-*code* of $\langle a_0,\dots,a_n\rangle$.

Definition 1.10. A *dl*-functional ordered pair $\langle f,g\rangle$ of functions from $\mathbb{N}^2$ into $\mathbb{N}$ satisfies the *compact support property* (*dlcs-functional ordered pair*) if and only if

$$\forall c\ \exists M\ \exists a\ [i\geqslant M\rightarrow(f(c,i)=a\wedge g(c,i)=0].$$

We have similarly the notion of dh-*functional ordered pair*.

Definition 1.11. A *dl*-functional ordered pair $\langle f,g\rangle$ satisfies the *strongly compact support property* if and only if g verifies

$$\forall c\ \forall i\ \forall j\ [(g(c,i)=0\vee g(c,i)=1)\wedge((g(c,i)=0\wedge j\geqslant i)\rightarrow g(c,j)=0)].$$

We must note that actually, by the very definition, a *dhl*-functional is only useful when the (eventually weak) arithmetical structure M to which f belongs allows simultaneously to define the natural order within M.

At this step, and except for pairing functions, we have separately defined encodings and decodings without using any relationship existing between them. For investigating these links, we must remind the reader of some first-order logical notions.

1.4. Basic notions of first-order logic

The basic notions of first-order logic can be found in any good textbook (as [8] for instance). Let us just recall what concerns definability.

Definition 1.12. Let L be a logical first-order language, S a symbol (symbol of individual constant, function symbol or predicate) and $L^*=L\cup\{S\}$. Let $\mathcal{M}$ be an L-structure with domain M.

(1) We say that an element a of M is *definable in the structure* $\mathcal{M}$ if and only if there exists an L-formula ϕ with one free variable such that

$$(\mathcal{M},a)\models x=a\Leftrightarrow\phi(x).$$

(2) We say that an n-ary function f over M is *definable in the structure* $\mathcal{M}$ if and only if there exists an L-formula ϕ with $n+1$ free variables such that

$$(\mathcal{M},f)\models y=f(x_1,\dots,x_n)\Leftrightarrow\phi(x_1,\dots,x_n,y).$$

(3) We say that an n-ary relation R over M is *definable in the structure* $\mathscr{M}$ if and only if there exists an L-formula ϕ with n free variables such that

$$(\mathscr{M}, R) \models R(x_1, \ldots, x_n) \Leftrightarrow \phi(x_1, \ldots, x_n).$$

In this paper we are only interested in *arithmetical structures* since we only deal with integer n-tuples or lists.

Definition 1.13. By *arithmetical structure*, we mean any structure $\langle \mathbb{N}, \mathscr{L} \rangle$ where $\mathscr{L}$ is a set of constants, relations or functions which are definable in $\langle \mathbb{N}, +, . \rangle$.

Also we remind the reader that recursive or recursively enumerable functions are arithmetical. The converse is far from being true since there is a (strict) arithmetical hierarchy. We denote by $DEF(\mathbb{N}, \mathscr{L})$ the set of constants, functions and relations which are first-order definable within the structure $\langle \mathbb{N}, \mathscr{L} \rangle$.

2. Arithmetical encoding of lists

2.1. Gödel's results

One important part of the interest of decoding lists functions stems from the following classical Gödel's result. First, we recall the well-known notion of a primitive recursive "definition" leads to the notion of primitive recursive closure.

Definition 2.1. An arithmetical structure is *closed under primitive recursion* if and only if for any ordered pair $\langle g, h \rangle$ in which g and h are definable in this structure and are, respectively, the $(n+1)$-ary function f defined (or, more precisely, presented by a primitive recursive schema) by

$$f(a_1, \ldots, a_n, 0) = g(a_1, \ldots, a_n),$$

$$f(a_1, \ldots, a_n, k+1) = h(k, a_1, \ldots, a_n, f(a_1, \ldots, a_n, k)),$$

is also first-order definable in this structure.

Proposition 2.1 (Gödel [9]). *Any arithmetical structure within the natural order* $\leqslant$ *and a decoding headed lists function (or a decoding lists function) that are definable is closed under primitive recursion.*

Proof. Let us prove it in the case of a *dhl*-function D. Indeed, we define $f(a_1, \ldots, a_n, k) = r$ by

$$\exists c[D(c,k,0) = g(a_1, \ldots, a_n) \wedge D(c,k,k) = r \wedge \forall i < r \ [D(c,k,i+1) = h(i, a_1, \ldots, a_n, D(c,k,i))]].$$

(Of course, the successor function S is $(\mathbb{N}, \leqslant)$-definable). □

Corollary 2.1. *The theory of a structure* $\langle \mathbb{N}, \leqslant, D\rangle$, *where* D *is a* dhl-*function or a* dl-*function, is undecidable.

Proof. Addition and multiplication are successively defined by the well-known primitive recursive schema:

$$x+0=x \quad \text{and} \quad x+Sy=S(x+y),$$

$$x\cdot 0=0 \quad \text{and} \quad x\cdot Sy=x\cdot y+x,$$

so that the corollary follows from Proposition 2.1 and the undecidability of $\langle \mathbb{N}, +, .\rangle$ [5]. □

The previous Gödel's result takes out its importance from the fact there do exist arithmetical structures which are not closed under primitive recursion. For instance, addition $+$ is not $\langle \mathbb{N}\text{-}S\rangle$-definable ([15]; see for instance [8]) and multiplication . is not $\langle \mathbb{N}, +\rangle$-definable ([18]; see also, for instance, [8]); hence $\langle \mathbb{N}, S\rangle$, $\langle \mathbb{N}, +\rangle$ are not closed under primitive recursion.

Proposition 2.2. *Let* J *be a pairing function (of nonnegative integer ordered pairs). If* J *is* $\langle \mathbb{N}, \mathscr{L}\rangle$*-definable then the associated projections* K *and* L, *and the n-tupling functions and their associated decoding functions for n-tuples are also* $\langle \mathbb{N}, \mathscr{L}\rangle$*-definable.*

Proof. Obvious. □

We have the following partial converse.

Proposition 2.3. *Let* n *be a given nonnegative integer. If the natural order* $\leqslant$ *and a decoding function for n-tuples are* $\langle \mathbb{N}\text{-}\mathscr{L}\rangle$*-definable, then there is a canonically encoding function (we call* associated encoding function), *which is also* $\langle \mathbb{N}, \mathscr{L}\rangle$*-definable.*

Proof. Let us prove it for $n=2$. We have

$$\forall a\ \forall b \in \mathbb{N}, \exists c \in \mathbb{N}\ [K(c)=a \wedge L(c)=b].$$

Let us note that the code c is not necessarily unique. We can $\langle \mathbb{N}, \leqslant, K, L\rangle$-define J by $J(a,b)=\mu c\ [K(c)=a \wedge L(c)=b]$, where μc means (the least c such that), operator which is $(\leqslant)$-definable. □

Problem 1 (*Open*). How to avoid the use of the $\langle \mathbb{N}, \mathscr{L}\rangle$-definability of $\leqslant$ in Proposition 2.3?

Fundamental Remark. An encoding function for lists with header (of nonnegative integers) is a mapping from $\mathbb{N}\times(\mathbb{N})^*$ into $\mathbb{N}$. Therefore, the base set is not a subset of $\mathbb{N}$, so that one cannot define such coding function in any arithmetical structure $\langle \mathbb{N}, \mathscr{L}\rangle$,

except after ... encoding. This has two consequences in what we are concerned:
- there is no possible analogue of Propositions 2.2 and 2.3 about encoding lists and headed lists; the encodings for lists we use in Section 1 are not mappings;
- this also explains why we specially insist on decoding lists and decoding headed lists functions in the frame of first-order logic.

2.2. *Arithmetical encoding in a sublanguage*

We just have seen that within the full structure $\langle \mathbb{N}, +, . \rangle$ we can encode any list. There is no reason for which one can do the same in a given substructure of $\langle \mathbb{N}, +, . \rangle$. Indeed, we have reminded the reader that it is not possible say in $\langle \mathbb{N}, + \rangle$. Nevertheless, we give below an example of a substructure, namely $\langle \mathbb{N}, +, \perp \rangle$, which encodes all lists. It turns out that multiplication is also definable in this structure; the proof of this definability lies on this very encoding. The question whether arithmetical substructure allowing encoding of lists do define both addition and multiplication arises. The negative answer is developed in Section 4.

Denote by $\perp$ the coprimeness predicate defined by $x \perp y$ if and only if $\gcd(x, y) = 1$. To show that the structure $\langle \mathbb{N}, +, \perp \rangle$ is closed under primitive recursion we have to construct an encoding for any list. Actually, we construct at first an encoding for certain lists of primes, each prime being itself the code of some special type of ordered pair. This devices allow us to define multiplication and to apply the results of Gödel for obtaining a general encoding of lists and the primitive recursive closure of $\langle \mathbb{N}, +, \perp \rangle$.

Proposition 2.4 (Richard [23]). *Multiplication is* $\{+, \perp\}$*-definable.*

Proof. We just give the sketch of the proof which is detailed in [23].

Step 1: We define a pairing function g of the following set,

$$A = \{\langle p, x \rangle \mid p \text{ is prime and } 1 < x < p\}.$$

By definition $J(p,x)$ is the smallest prime integer q which satisfies

$$q \equiv 1 [\mathrm{mod}\, r] \text{ for all prime integers } r < p \text{ and } q \equiv x \ [\mathrm{mod}\, p].$$

One can prove J is one-to-one. Consider $c(x) = \prod_{i=1}^{i=x} J(p_i, i+1)$, where p_i is the $(i+1)$th prime and $p_0 = 2$. We see that $c(x)$ is the least integer y such that $J(3,2) = 5$ divides y and for all prime integers p and all integers $k < x+1$, if $J(p,k)$ divides y, then $J(p', k+1)$ also divides y, where p' is the smallest prime greater than p. We notice that all the following notions and relations, $x < y$; $x \equiv y [\mathrm{mod}\, p]$ where p is a prime; p is prime and p divides x; p is a product of a finite set of coprime integers are easily $\{+, \perp\}$-definable.

Step 2: It consists of defining the mapping $x \mapsto p_x$ from J.

Step 3: We introduce $d(x,y) = \prod_{i=1}^{i=y} J(p_{ix}, i+1)$ and we note in order to $\{+, \perp\}$-define this function d that $d(x,y)$ is the smallest t such that $[J(p_x, 2)$ divides t and, for all prime integers p_j and for all $k < y+1$, if $J(p_j, k)$ divides t, then $J(p_{j+x}, k+1)$ also divides $t]$.

This property enables us to express a $\{+,\perp\}$-formula and permits to $\{+,\perp\}$-define the graph of multiplication by using again the $\{+,\perp\}$-definability of $\{+,\perp\}$. □

3. Decidability properties separating *n*-tupling and encoding of lists

3.1. Undecidability of theories defining a decoding of headed lists

Theorem 3.1. *If f is a decoding headed lists with strongly compact support function, then the theory $Th(\mathbb{N}, f)$ is undecidable.*

Proof. We define the constant 0 by $x=0 \leftrightarrow \exists c\ \exists n\ \forall i\ [f(c,n,i)=x]$. Then we define the natural order by $x<y \leftrightarrow \forall c\ (f(c,x,y)=0)$ so that the successor function S is also $\langle \mathbb{N}, f\rangle$-definable. Now we define, following Gödel (1931), addition by $x+y=z \leftrightarrow \exists c\ [f(c,y,0)=x \wedge f(c,y,y)=z \wedge \forall i<y\ (f(c,y,Si)=S(f(c,y,i)))]$.

At last multiplication is similarly presented by primitive recursion in a schema using order and addition. Hence, by Proposition 2.1, addition and multiplication are $\langle \mathbb{N}, f\rangle$-definable and $Th(\mathbb{N}, f)$ is consequently undecidable. □

Remark. Corollary 2.1 is not implied by Theorem 3.1 since we use the strong compactness in this theorem to define natural order. Corollary 2.1 is in a sense more general because D is a decoding headed list function, whose support is not supposed to satisfy the strong compactness property.

Theorem 3.2. *If $\langle f,g\rangle$ is an ordered pair of decoding headed list functions with strong compact support then $Th(\mathbb{N}, \langle f,g\rangle)$ is undecidable.*

Proof. We define the constant 0 by $x=0 \leftrightarrow \exists c\ \forall i\ [g(c,i)=x]$. Then 1 is defined by $x=1 \leftrightarrow \exists c\ \exists i\ [g(c,i)=x \wedge g(c,i)\neq 0]$. The natural order is defined by $x\leqslant y \leftrightarrow \forall c\ [g(c,y)=1 \rightarrow g(c,x)=1]$. Then we achieve the proof just as in Theorem 3.1. □

Corollary 3.1. *No functional ordered pair of decoding headed lists functions with strongly compact support can be defined in the structure $\langle \mathbb{N}, +, n\mapsto 2^n\rangle$.*

Proof. The fact to define some *dhlscs*-functional ordered pair contradicts Semenov's result insuring $Th(\mathbb{N}, +, n\mapsto 2^n)$ is decidable (see [26]). □

Analogous corollaries hold for $\langle \mathbb{N}, +\rangle$, $\langle \mathbb{N}, .\rangle$, $\langle \mathbb{N}, ., \leqslant_P\rangle$ (where $\leqslant_P$ denotes the natural order restricted to primes), due, respectively, to [15, 18, 17].

Problem 2 (*Open*). What happens if we remove the strongly compact property?

Let us notice that, however, we know that $Th(\mathbb{N}, \beta)$ for β-Gödel function is undecidable (cf. [21]).

3.2. Decidability and undecidability of theories defining a pairing function

Having recourse to first-order theories allows us to clearly separate encoding devices for n-tuples and for lists. Indeed, we shall see below there do exist decidable theories within which one can define a pairing function (and consequently also a depairing function and a decoding for n-tupling function for any fixed integer n) and within it is impossible to define any decoding lists function.

Theorem 3.3. *There exists a pairing function J such that $Th(\mathbb{N},J)$ and $Th(\mathbb{N},+,J)$ are decidable.*

Proof. As we said above, in 1983, Semenov has shown $Th(\mathbb{N},+,n\mapsto 2^n)$, namely the theory of $\mathbb{N}$ equipped with addition and exponentiation of basis 2, is decidable. We easily define the pairing function J in the usual Ackermann's way in $\langle\mathbb{N},+,n\mapsto 2^n\rangle$ as follows:

$$J(x,y)=2^{x+1}+2^{x+y+2}$$

what achieves the proof. □

From Theorem 3.3 above, it turns out we must know whether $Th(\mathbb{N},J)$ is decidable for any pairing function J. The answer is proved to be negative in Theorem 3.7 below. However, to introduce the technics of proof, we begin by showing weaker results just needing simple arguments.

Theorem 3.4. *There exists a recursive pairing function J such that $Th(\mathbb{N},J,+)$ is undecidable.*

Proof. Let us remind the reader that there exists a finite cofinite binary recursive relation R on $\mathbb{N}$ such that $Th(\mathbb{N},R)$ is undecidable. (For instance $x_y\neq 0$ which means that the $(y+1)$-prime divides x, is undecidable since it has the so-called isomorphic re-interpretation property (see Definition 4.2 below and [21])). We put

$$A=\{(a,b)\in\mathbb{N}^2/\rceil R(a,b)\},$$
$$B=\{(a,b)\in\mathbb{N}^2/R(a,b)\}.$$

These two sets are infinite and we enumerate them as follows:

$$A=\{c_0,c_1,\ldots,c_n,\ldots\},$$
$$B=\{d_0,d_1,\ldots,d_n,\ldots\}.$$

With this definition A and B are disjoint and $\mathbb{N}^2=A\cup B$. So we can define a pairing function J as follows:

$$J(a,b)=\begin{cases}2i & \text{if } (a,b)=c_i,\\ 2i+1 & \text{if } (a,b)=d_i.\end{cases}$$

The theory $Th(\mathbb{N},+,J)$ is undecidable since we can define the relation R within the structure $\langle\mathbb{N},+,J\rangle$ by using the logical equivalence $R(a,b) \leftrightarrow \exists x\ [J(a,b)=x+x+1]$. □

Theorem 3.5. *There exists a recursive pairing function J such that $Th(\mathbb{N},S,J)$ is undecidable (where S denotes as usual the successor function).*

Proof.

Step 1: There is a set W such that $Th(\mathbb{N},S,W)$ is undecidable. Take any recursively enumerable nonrecursive set W, for instance, the indices of Turing machines which do halt on their own index. In the theory $Th(\mathbb{N},S,W)$, one can ask whether $n\in W$ by a question of the form $W(\bar{n})$, where $\bar{n}$ is $SSSS\ldots S0$ with n occurrences of S.

Step 2: The set W above is necessarily infinite. Let us recursively enumerate W, i.e. let us define a recursive mapping g from $\mathbb{N}$ into W, so that $W=g[\mathbb{N}]$. Let C be the usual Cantor pairing function. We put $J(a,b)=g(C(a,b))$, which determines a new recursive pairing function J from $\mathbb{N}^2$ into $\mathbb{N}$. To ask whether $n\in W$ turns out into the question

$$\exists a\ \exists b\ [J(a,b)=\bar{n}]$$

which is a first-order $\langle\mathbb{N},J,S\rangle$-formula so that $Th(\mathbb{N},J,S)$ is undecidable since W is not recursive. □

Remark. Actually, the J of Theorem 3.5 is even primitive recursive and probably of lower complexity. Now we can obtain a pairing function with undecidable theory, more precisely:

Theorem 3.6. *There exists a recursive pairing function J such that $Th(\mathbb{N},J)$ is undecidable.*

Proof. It can be proved by two steps.

Step 1: There is a bijection f of $\mathbb{N}$ such that $Th(\mathbb{N},f)$ is undecidable. Let us consider a recursive enumeration g of some recursively enumerable nonrecursive set W:

$$W=\{g(0),g(1),\ldots,g(n),\ldots\}.$$

We determine a (recursive) bijection f as follows:

- considering the $(g(0)+2)$ first integers of $\mathbb{N}$, we define the restriction of f to these elements by a cycle

$$f(0)=1,\quad f(1)=2,\ldots,f(g(0))=g(0)+1,\quad f(g(0)+1)=0;$$

- considering the $(g(1)+2)$ next integers, we define the restriction of f to these elements by a cycle

$$f(g(0)+2)=g(0)+3, \quad f(g(0)+3)=g(0)+4,\ldots,$$
$$f(g(0)+2+g(1))=g(0)+2+g(1)+1,$$
$$f(g(0)+2+g(1)+1)=g(0)+2;$$

- then we determine similarly $f(x)$ for the $g(2)+2$ next integers x, then for the $g(3)+2$ next elements, and so and ...

The theory $Th(\mathbb{N},f)$ is undecidable since to know whether $n\in W$, it suffices to ask the following question expressed by a first-order $\langle\mathbb{N},f\rangle$-formula:

$$\exists x_0\ \exists x_1\ \ldots\exists x_{n+1}\ [f(x_0)=x_1\wedge f(x_1)=x_2\wedge\cdots\wedge f(x_n)=x_{n+1}\wedge f(x_{n+1})=x_0].$$

Step 2: Using any recursive pairing function J_0, we obtain a second recursive pairing function J_1 defined by

$$J_1(2n+1,2n+1)=4n+1; \quad J_1(2n,2n)=2n$$

and $J_1(a,b)=4J_0(a,b)+3$ if $a\neq b$. So we get a recursive pairing function J such that $J(a,b)=J_1(a,b)$ for any ordered pair which is not of the form $\langle 2n,2n\rangle$ and such that $J(2n,2n)=2f(n)$, where f is the recursive bijection f we have exhibited in the first step.

Such a pairing function J is, by its very construction, recursive. Consequently, $Th(\mathbb{N},J)$ is undecidable since to know whether $n\in W$, it suffices to ask the following question in a first-order form, namely, for instance, by the question of the truth-value of the sentence

$$\exists x_0\exists x_1\ldots\exists x_{n+1}[J(x_0,x_0)=x_1\wedge J(x_1,x_1)=x_2\wedge\cdots\wedge J(x_n,x_n)=x_{n+1}$$
$$\wedge J(x_{n+1},x_{n+1})=x_0],$$

noting that the x_i's are necessarily even when the answer is positive. □

The pairing functions being arithmetical, they are $\langle\mathbb{N},+,\times\rangle$-definable. One knows multiplication is not $(+)$-definable. Hence, it is natural to ask whether there exists a recursive pairing function J such that multiplication is $\langle\mathbb{N},+,J\rangle$-definable. The answer yet is positive:

Theorem 3.7. *There exists a primitive recursive pairing function J, namely the Cantor pairing function C, such that multiplication is* $\langle\mathbb{N},+,J\rangle$*-definable.*

Proof. We have $C(x,x+1)=2(x+1)^2$. Hence, the mapping which associates to an integer n its square n^2 is $\langle\mathbb{N},+,J\rangle$-definable. As already noted by Tarski, the definition of multiplication in this language follows from the formula $(x+y)^2=x^2+y^2+xy+xy$. □

4. Undecidability levels of structures with decoding lists functions

A somewhat more difficult question is to know whether arithmetical substructure allowing encoding of lists do define both addition and multiplication. A negative answer is given in the present section.

In Section 3, we have seen that every arithmetical structure in which a decoding headed lists with compact support function is definable has an undecidable theory on $\mathbb{N}$. However, one can distinguish several levels of undecidability according to several possible criteria. The first criterium coming to mind is certainly the level within the arithmetical hierarchy. Actually, this is not the one we shall deal in the present paper; we deal more with a criterium which is linked to the means used to prove an arithmetical theory is undecidable.

The Church–Turing's result on the undecidability of $Th(\mathbb{N}, +, \times)$ has permitted to show that lot of arithmetical structures have undecidable theories by proving $DEF(\mathbb{N}, \mathscr{L}) = DEF(\mathbb{N}, +, \times)$. We can call it the level of arithmetical substructure having the *complete definability property.*

Nevertheless, this method is not convenient to conclude for certain arithmetical structures. The second author has introduced the method called *isomorphic re-interpretation property* (*IRP*). Generalisation of this within computer science is nothing but simulation. For more details, one can see [4]. Let us call IRP the level of arithmetical substructures having the *isomorphic re-interpretation.*

4.1. Definitions and problems

Definition 4.1. An arithmetical structure $\langle \mathbb{N}, \mathscr{L} \rangle$ is said *def-complete* if and only if $\mathrm{DEF}(\mathbb{N}, \mathscr{L}) = \mathrm{DEF}(N, +, \times)$.

We know lot of def-complete arithmetical structures. For a detailed survey about them, one can consult the Korec's monograph (see [14]). In the framework of decoding functions, one necessarily comes to the question of knowing whether the natural order relation (and consequently addition and multiplication by Corollary 2.1) is $\langle \mathbb{N}, \mathscr{L} \rangle$-definable in any arithmetical structure within which a decoding lists function is definable. We shall prove that the answer is negative.

Definition 4.2. (i) A structure $\langle A, c_1, \ldots, c_n \rangle$ is *emulatable* in a structure $\mathscr{M}$, with domain M, iff there exists a structure $\langle B, d_1, \ldots, d_n, \equiv \rangle$ isomorphic to $\langle A, c_1, \ldots, c_n, = \rangle$, where B is a subset of M, $\equiv$ is a binary relation on A, such that $B, d_1, \ldots, d_n, \equiv$ are $\mathscr{M}$-definable.

(ii) A structure $\mathscr{M}$ has the *isomorphic re-interpretation property* iff the structure $\langle \mathbb{N}, +, \times, = \rangle$ is emulatable in $\mathscr{M}$.

Remark. If a structure has only finitely many constants, functions and relations and has an undecidable theory, and is emulatable in a structure $\mathscr{M}$, then $Th(\mathscr{M})$ is undecidable. In particular, $Th(\mathscr{M})$ is undecidable if $\mathscr{M}$ has the isomorphic re-interpretation property.

The isomorphic re-interpretation property was introduced and used to prove that many structures $\mathcal{M}$ with domain $\mathbb{N}$ have an undecidable theory (see [20, 22, 24]). This notion is weaker than *def*-completeness because, in fact, the isomorphic re-interpretation property does not imply definability of $+$ and $\times$, and the converse is obviously true.

Now one can ask whether every arithmetical structure $\langle \mathbb{N}, \mathscr{L} \rangle$ within which a decoding lists function is definable has the isomorphic re-interpretation property. Still the answer is negative.

4.2. *Decoding functions for lists without IRP*

The aim of this section is to prove that there is a level of arithmetical substructure not having the IRP but such that *one can decode any integer list within them.*

Theorem 4.1. *There exists a decoding lists function f such that* IRP *does not hold for* $\langle \mathbb{N}, f \rangle$.

Let us begin by some definitions and classical results.

Definition 4.3. The alphabet of $\mathscr{L}(\mathrm{PA})$ (the language of *Peano* arithmetic) is $A = \{\neg, \vee, \wedge, \forall, \exists, \times, ', (,), =, S, +, .\}$. A variable is nothing but a word of the form $x^{''\ldots'}$. The *Gödel numbering* defined from A is any one-to-one mapping g from A into $\mathbb{N}\backslash\{0\}$, for instance,

$$g(\neg) = 1, \quad g(\vee) = 2, \quad g(1) = 3, \quad g(\forall) = 4, \quad g(\exists) = 5, \quad g(x) = 6, \quad g(') = 7,$$

$$g(() = 8, \quad g()) = 9, \quad g(=) = 10, \quad g(S) = 11, \quad g(+) = 12, \quad g(.) = 13.$$

The *Gödel coding of a word* on A is the injection $ng : A^+ \to \mathbb{N}^*$ defined by

$$ng(w_1 \ldots w_n) = p_1^{g(w_1)} \times \cdots \times p_n^{g(w_n)},$$

where p_i is the $(i+1)$-th prime number with $p_0 = 2$. Due to the fact $\mathscr{L}(\mathrm{PA})$-formulas and $\mathscr{L}(\mathrm{PA})$-sentences are words on the alphabet A this provides us with a *Gödel* numbering (or coding) of $\mathscr{L}(\mathrm{PA})$-formulas and $\mathscr{L}(\mathrm{PA})$-sentences.

Definition 4.4. Let $\langle \mathbb{N}, \oplus, \otimes \rangle$ be an arithmetical structure consisting of the set $\mathbb{N}$ of nonnegative integers with two binary operations. The *satisfaction function* of this structure is the mapping SAT from $\mathbb{N}^2$ into $\{0, 1\}$ satisfying $\mathrm{SAT}(m, n) = 1$ if and only if

(i) $m = ng(\phi(v_1, \ldots, v_k))$ for some $\mathscr{L}(\mathrm{PA})$-formula ϕ;
(ii) $n = ng(\langle a_1, \ldots, a_k \rangle)$ for some n-tuple $\langle a_1, \ldots, a_k \rangle$ of $\mathbb{N}$;
(iii) $(\mathbb{N}, \oplus, \otimes) \models \phi[a_1, \ldots, a_n]$.

Remark. The function SAT may be recursive but this is generally not the case. It can belong to the arithmetical hierarchy (and consequently be Δ_n, Σ_n or Π_n for some

nonnegative integer n) or it may not be arithmetical at all (this is the case for the standard model according to Tarski's theorem we remind the reader below). A discussion of arithmetical hierarchy is given by Enderton in [8].

Definition 4.5. An arithmetical structure $\langle \mathbb{N}, \oplus, \otimes \rangle$ is a *Δ_n-structure* (respectively a *Σ_n-structure* a *Π_n-structure*) if and only if its satisfaction function is Δ_n (respectively Σ_n, Π_n).

Proposition 4.1 (Hilbert-Bernays' Theorem [12]). *The Peano Arithmetic has a Δ_2-model.*

Proof. See, for instance, ([27], p. 860). □

Definition 4.6. The *truth function* of an arithmetical $\mathscr{L}$-structure $\mathscr{M}$ is the function TRUE from $\mathbb{N}$ into $\{0,1\}$ defined by $\mathrm{TRUE}(n)=1$ if and only if;
(i) $n = ng(\sigma)$ for a sentence σ of $\mathscr{L}$;
(ii) $\mathscr{M} \models \sigma$.

Remark. (1) In contrast to the definition of satisfaction function, we do not need to assume the structure is denumerable to define its truth function.

(2) If the satisfaction function SAT of a given stucture is Δ_n (respectively Σ_n, Π_n) then its truth function is also Δ_n (respectively Σ_n, Π_n). We shall make use of the following classical theorem.

Theorem 4.2 (Gödel [10] and Tarski [29]). *The truth function of the standard model $\langle \mathbb{N}, +, . \rangle$ is not arithmetical.*

Proof of Theorem 4.1. The function β of Gödel is defined by

$$f(c,n,i) = \begin{cases} \beta(K(c),L(c),i) & \text{if } i \leqslant n, \\ 0 & \text{if } i > n. \end{cases}$$

This function is a decoding headed lists function with strongly compact support.

This function is $\langle \mathbb{N}, +, \times \rangle$-definable by a formula ϕ with four free variables, so that $f(c,n,i)=d \leftrightarrow \phi(c,n,i,d)$, or in order to deal with sentences $f(c,n,i)=d \leftrightarrow \phi(\overline{c},\overline{n},\overline{i},\overline{d})$, putting as usual $\overline{c}=SSSS\ldots S0$ with n occurrences of symbol S and the constant 0 which are both $\langle \mathbb{N}, +, . \rangle$-definable.

Now consider a structure $\langle \mathbb{N}, \oplus, \otimes \rangle$ which is a Δ_2-model of Peano arithmetic. Let us denote by $\psi(\overline{c},\overline{n},\overline{i},\overline{d})$ the formula obtained by replacing $+$ and ., respectively, by $\oplus$ and $\otimes$ within $\phi(\overline{c},\overline{n},\overline{i},\overline{d})$. Let us denote by g the function of domain $\mathbb{N}$ defined by the logical equivalence

$$g(c,n,i)=d \leftrightarrow \psi(\overline{c},\overline{n},\overline{i},\overline{d}).$$

The PA-provable sentences show that g is actually a function, and a decoding function for headed lists with the strongly compactness property. The structure $\langle \mathbb{N}, g\rangle$ does not have the *IRP*, otherwise one could define an isomorphic model to $\langle \mathbb{N}, +, .\rangle$ within $\langle \mathbb{N}, g\rangle$ and consequently within $\langle \mathbb{N}, \oplus, \otimes\rangle$. Due to the fact the truth function of $\langle \mathbb{N}, g\rangle$ is arithmetical, the truth function of $\langle \mathbb{N}, +, .\rangle$ would be in turn, at least, arithmetical (namely Δ_2) This contradicts Tarski's Theorem. □

Acknowledgements

We thank the referees for pointing out some errors (in particular in Proposition 1.1) and suggesting many improvements.

References

[1] W. Ackermann, Die Widerspruchsfreiheit der allgemunen Mengenlehre, Math. Ann. 114 (1937) 305–315; French translation by F. Gaillard, La non-contradiction de la théorie des ensembles sans axiome d'infini, Séminaire LLAIC1 1 (1989-90) 71–86.

[2] G. Cantor, Uber eine Eigenschaft des Inbegroffes aller reellen zlgebraischen Zahlen, J. Reine Angew. Math. 77 (1874) 258–262. Gesamm. abh., Springer, Berlin, 1930 pp. 15–118, French translation: Acta Math. 2, 305–310.

[3] G. Cantor, R. Dedekind Briefwelrel, Hermann, Paris, 1937; french translation by J. Cavailles, Philosophie mathématique, Hermann, Paris, 1962, pp. 177–249.

[4] P. Cegielski, Definability, decidability and complexity, Ann. Math. Artificial Intelligence 16 (1996) 311–341.

[5] A. Church, An unsolvable problem of elementary number theory, Amer. J. Math. 58 (1936) 345–363; reprinted in M. Davis (1965) 88–107.

[6] M. Davis (Ed.), The Undecidable: Basic Papers on Undecidable Propositions, Unsolvable Problems and Computable Functions, Raven Press, New York, 1965.

[7] E.W. Dijkstra, Recursive programming, Numer. Math. 2 (1960) 312–318; also Report MR33 of the Computation Department of the Mathematical Centre, Amsterdam; reprinted in Rosen, Saul, Programming Systems and Languages, McGraw-Hill, New York, 19XX XV+734 pp., pp. 221–227.

[8] H.B. Enderton A Mathematical Introduction to Logic, Academic Press, New York, 1972, XIII+295 pp.

[9] Gödel, K. Über formal unentscheidbare Sätze der Principia mathematica und verwandter Systeme I, Monatshefte Mathematik Physik 38 (1931) 173–198; English translation in Van Heijenoort (1967) and in Collected Works, Vol. 1, Oxford University Press, Oxford, 1986; French translation in Le théorème de Gödel, Seuil (1989), 184 pp., pp. 105–143.

[10] K. Gödel, Collected Works, Vol. 3, Oxford University Press, Oxford, 1995.

[11] K. Gödel, Letter to Zermelo dated 12 october 1932, in: Grattan-Guinness (Ed.), Historia Mathematica, Vol. 6, 1979, 294–304.

[12] D. Hilbert, P. Bernays, 1979, Grundlagen der Mathematik 1, 2nd ed., Springer, Berlin, 1968; French translation by F. Gaillard, M. Guillaume, Fondements des mathématiques 1, Prépublication LLAIC1, 1997, 394 pp.

[13] D. Hilbert, P. Bernays, Grundlagen der Mathematik vol. 2, 2nd ed., Springer, Berlin, 1970; French translation by F. Gaillard, E. Guillaume, and M. Guillaume, Fondements des mathématiques, vol. 2, Prépublication LLAIC1, 1997, 395 pp.

[14] I. Korec, A list of arithmetical structures strongest with respect to the first order definability, Mathematical Institute Slovak, Academy of Sciences, Bratislava, preprint 33, 1996; Theoret. Comput. Sci., to appear.

[15] C.H. Langford, Some theorems on deducibility, Ann. Math. 28 (1926) 16–40, 459–471.

[16] J. McCarthy, Recursive functions of symbolic expressions and their computation by machine, Part I, Comm. ACM (1960) 184–195; reprinted in Rosen, Saul, Programming Systems and Languages, McGraw-Hill, New York, 1967, XV+734 pp., pp.455–480 and in Horowitz, Ellis, Programming Languages: A Grand Tour, Computer Science Press, Rockville, MD, 1983 2nd ed. 1985, 3rd ed., 1987, IX+512 pp., pp. 203–214.

[17] F. Maurin, The theory of integer multiplication with order restricted to primes is decidable, Symbolic Logic (1994) 184–195.

[18] M. Presburger, Über die Vollständigkeit eines gewissen Systems der Arith-metik ganzer Zahlen, in welchem die Addition als einzige Operation hervortritt, Spra-wozdanie z I Kongresu matematyków krajów slowianskich, Warsawa, Warsaw, 1930, pp. 92–101, 395; English translation and study by J. Zygmunt, History and Philosophy of Logic, vol. 12, 1991, pp.211–233.

[19] W.V.O. Quine, Concatenation as a basis for arithmetics, Symbolic J. Logic 11 (1946) 105–114.

[20] D. Richard, The arithmetics as theories of two orders, Ann. Discrete Math. 23 (1984) 287–312.

[21] D. Richard, Définissabilité en Arithmétique et méthode de codage ZBV appliquée à des langages avec successeur et coprimarité, Thèse d'État, Université Lyon-I, n.85-16, 1985.

[22] D. Richard, Answer to a problem raised by J. Robinson: the arithmetic of positive or negative integers is definable from successor and divisibility, J. Symbolic Logic, 50 (1985) 135–143.

[23] D. Richard, Equivalence of some questions in mathematical logic with some conjectures in number theory, in: Mollin (Ed.), Number Theory and Applications, NATO Asi Series Series C: Mathematical and Physical Sciences, vol. 265, 1988, pp. 529–545.

[24] D. Richard, Definability in terms of the successor function and the coprimeness predicate in the set of arbitrary integers, J. Symbolic Logic, 54 (1989) 1253–1287.

[25] J. Robinson, Definability and decision problems in arithmetic, J. Symbolic Logic vol. 14 (1949) 98–114; reprint in S. Feferman (Ed.), The Collected Works of J. Robinson, American Mathematical Society, Providence, RI, 1996, 338 pp.

[26] A. Semenov, Logical theories of one place functions on the set of natural numbers (in russian), Izv. Acad. Nauka.SSSR Ser. Mat., vol. 47, 1983, pp. 623–658 English translation in Mat. USSR-Izv. 22 (1984) 587–618.

[27] C. Smorynski, The incompleteness theorem, in: J. Barwise (Ed.), Handbook of Mathematical Logic, CHD1 Logic, Studies in Logic, vol. 90, 1977.

[28] C. Smorynski, Logical Number Theory I, Springer, Berlin, 1991, X+405 pp.

[29] A. Tarski, Der Wahrheisbegriff in den formalisierten Sprachen, Studia Philosophica, vol. 1, 1936, pp. 261–405; English translation in: A. Tarski, Logic, Semantics, Metamathematics, Oxford University Press, Oxford, 1956, pp. 157–278; French translation in: A. Tarski, Logique, Sémantique, Métamathématique 1923–1944, Armand Colin, Paris, vol. 1, 1972, vol. 2, 1974, pp. 159–269. [Interesting result for us is Section 5, Theorem 1].

[30] A.M. Turing, On computable numbers, with an application to the Entscheidungsproblem, Proc. London Mathematical Society, vol. 42, 1936, pp. 230–265; correction, ibid., vol. 43, pp. 544–546; reprinted in: Davis, 1965, pp. 116–154; french translation in: A. Turing, J.-Y. Girard, La machine de Turing, Seuil, 1995, pp. 47–102.