Kolmogorov complexity in multiplicative arithmetic

Armando B. Matos

Technical Report Series: –

Departamento de Ciência de Computadores – Faculdade de Ciências

&

Laboratório de Inteligência Artificial e Ciência de Computadores

Universidade do Porto

Rua do Campo Alegre, 823 4150 Porto, Portugal

Tel: +351+2+6001672 – Fax: +351+2+6003654

http://www.ncc.up.pt/fcup/DCC/Pubs/treports.html

Kolmogorov complexity in multiplicative arithmetic

Armando B. Matos

July 2005

Abstract

Kolmogorov complexity can be made computable by restricting the underlying computational model, namely by using only total functions. In this work we consider some arithmetic computation models where the (total) functions are arithmetic expressions and study the question: given an integer n, what is the minimum length of an expression with value n? If integers are represented in binary (or in any other basis of numeration) arithmetic expressions involving only "+" and "×" are never shorter than the integer itself. However, if "unary" representation is used (as for instance in the Peano formalization of arithmetic), compression is possible and some interesting problems arise. In this paper we study the following arithmetic systems: $(\mathbb{N}, \times)$, $(\mathbb{N}, S, \times)$ and $(\mathbb{N}, +, \times)$ where "S" denotes the successor operator. The last two systems are equivalent in the sense that, for every integer n, $C(n)$ is $O(\log n)$. In the system$(\mathbb{N}, \times)$ numbers can be either compressible or incompressible, see Theorem 1.

1 Introduction

Let us consider some arithmetic formal system such as Presburger or Peano arithmetic. Given an integer n there may be many (possibly an infinite number of) expressions with value n and it is natural to ask which is the minimum length of such an expression, that is, what is the Kolmogorov complexity of n relatively to the arithmetic we are considering. We study more deeply this problem for the case of a multiplicative fragment of arithmetic with the integers represented in "unary". Other two fragments are briefly mentioned.

We show that many properties of the arithmetic and classic Kolmogorov complexities are quite different. Among these properties we study computability, the form of the function $C(n)$ and a weak form of the incompressibility Theorem (Theorem 5).

This paper is organized as follows. After some preliminaries in Section 2, we will study in Section 3 a particular unary multiplicative arithmetic system $(\mathbb{N}, \times)$; this study is generalized in Section 4. Some results relative to two other arithmetic systems are given in Section 5. Finally in Section 6 we summarize the main contributions of this work and mention some areas for future research.

2 Preliminaries

Notation

The notation used in this work is fairly standard. By $\ln n$ and $\log n$ we denote respectively the natural and the base 2 logarithm of n; $f^2(x)$ denotes $(f(x))^2$ (not $f(f(x))$!). The representation in some formal arithmetic system of the integer n is denoted by $\overline{n}$; we may have for instance $\overline{3} = SSS0$. An integer representation is *unary* if the length of the representation of n is $\alpha n + \beta$ where α and β are constants and $\alpha > 0$. If E is an expression, $|E|$ denotes its length. For instance if the length of an expression is defined as the number of symbols it contains and n is represented as S^n0, we have $|\overline{3}| = 4$ and $|\overline{3} \times \overline{2}| = |SSS0 \times SS0| = 8$.

A note on computable versions of Kolmogorov complexity

The Kolmogorov complexity $C(n)$ of the integer (or string) n is the length of a minimum program p such that the standard universal Turing machine with input p halts and outputs n, see [6].

Kolmogorov complexity is in general incomputable; this fact can be seen as a consequence of the undecidability of the halting problem. There are two ways of making Kolmogorov complexity computable:

- by bounding the execution time of standard universal Turing machine.
- by using total models of computation.

It is well known that no model of computation characterizes exactly the set of total functions. There are however some "natural" models of computation that define only total functions, namely

- Primitive recursive functions
- Arithmetics functions such as products of integers[1].

Let $\mathcal{A} = (\mathbb{N}, \cdots)$ be an arithmetical structure such as $(\mathbb{N}, +, \times)$. We define the Kolmogorov complexity of n relative to $\mathcal{A}$ as

$$C_{\mathcal{A}}(n) = \min\{|E| \,:\, \text{the expression } E \text{ has value } n\}$$

In the rest of this paper the expression "Kolmogorov complexity" is assumed to be relative to an arithmetic formal system.

[1] The decidability of an arithmetic formal and the computability (totality) of the functions associated with it are two concepts that should not be confused. Let $(\mathbb{N}, +, \times)$ denote the formal system consisting of first order logic with equality together with arithmetic of nonnegative numbers as formalized by Peano axioms. Similarly we have formal systems $(\mathbb{N}, +)$, $(\mathbb{N}, \times)$, $(\mathbb{Z}, +, \times)$, $(\mathbb{R}, +)$ and so on. Some of these systems, such as $(\mathbb{N}, +)$ and $(\mathbb{Z}, +, \times)$, are decidable while other like $(\mathbb{N}, +, \times)$ are not; for a survey of decidable theories see for instance [7]. Here "decidability" refers to the existence of an algorithm that decides if a given well-formed formula is a theorem. Moreover, for some decidable systems the *complexity* of decision algorithms has been studied, see for instance [3, 1, 4].

It should be noticed that even undecidable arithmetics can be used as models of computable Kolmogorov complexity; the only requirements (which are of a very general nature and obviously satisfied by all the formal systems mentioned above) are: (i) the existence of an effective procedure for enumerating the well formed expressions and (ii) the existence of an effective procedure for computing an arithmetic expression.

3 Kolmogorov complexity in a multiplicative formal system

First we study nonnegative integer numbers with multiplication. We use a particular formal system but as mentioned in Section 4 our results are easily generalized.

The number n is represented by $\overline{n} = S^n 0$, a string with length $n+1$.

The well formed expressions are defined by the following syntax which corresponds to what we call *standard* representation.

$$\begin{array}{rcl} \langle \texttt{exp} \rangle & \rightarrow & \langle \texttt{int} \rangle \mid \langle \texttt{exp} \rangle \times \langle \texttt{exp} \rangle \\ \langle \texttt{int} \rangle & \rightarrow & 0 \mid S\langle \texttt{int} \rangle \end{array}$$

Notice that integers are represented in "unary".

The length $|E|$ of an expression E is the number of symbols it contains (later we will use a more general definition of length); for instance $|SSS0 \times SS0| = 8$.

In order to study the compression obtained by factorizing an integer we begin by tabulating a few simple cases; the "gain" of a product is the corresponding (possibly negative) reduction of the expression length.

Product	Integer	Length of integer	Length of product	Gain
2×2	4	5	7	2
2×3	6	7	8	1
2×4	8	9	9	0
2×5	10	11	10	-1
2×6	12	13	11	-2
3×3	9	10	9	-1
3×4	12	13	10	-2

Clearly factorizing an integer does not increase the length of the expression (the gain is not positive) except in the cases 2×2 and 2×3.

We now describe an algorithm for finding an minimum length expression representing a given integer n.

Begin by expressing n as a product of prime factors:

$$n = p_1 \times p_2 \times \cdots \times p_k$$

where $p_1 \leq p_2 \leq ... \leq p_k$.

If $p_1 \geq 3$ the the product has already minimum length.

Otherwise, we can reduce the the length of an expression by replacing 2×2 by 4 or 2×3 by 6. But the question is of course, in what order should these products be done? The following example shows that this order may be important.

Example 1 *Consider the product* $2 \times 2 \times 3$. *This expression can be reduced to the minimal expression* 4×3 *but also to the non minimal expression* 2×6.

Another important example is the expression $2 \times 2 \times 3 \times 3$ which can be reduced to $4 \times 3 \times 3$ or to 6×6, both with length 15.

It follows that we never need more than one 6 and thus the following algorithm computes $C(n)$

```
Input:  an integer $n \geq 2$
Output:  $C_{(\mathbb{N},\times)}(n)$
    1) Factorize $n$ as a product $E$ of primes
    2) While possible replace $2 \times 2$ by 4
    3) If a factor 2 remains (the initial number of 2's is odd)
       and there is at least one factor 3:
         replace $2 \times 3$ by 6.
```

This algorithm runs in polynomial time. This is of course a consequence of using the unary representation of integers.

3.1 Compressible and incompressible integers

We use exponentiation as a convenient notation, i.e. 5^3 means $5 \times 5 \times 5$.

Let the prime factorization of n be partitioned in two factors

$$n = A \times B = (2^{e_2} \times 3^{e_3}) \times B$$

where B denotes the product of primes greater than or equal to 5; e_2 and e_3 can of course be 0. The part B is left unmodified because replacing a product involving at least one factor of B by the corresponding result increases the length of the expression.

The shortest factorization of A depends on e_2 and e_3; we distinguish two cases:

1. e_2 is even, $e_2 = 2f_2$; $A = 4^{f_2} \times 3^{e_3}$.
2. e_2 is odd, $e_2 = 2f_2 + 1$; $A = 2 \times 4^{f_2} \times 3^{e_3}$.

Integers with shortest representation have thus one of the forms

$$3^a \times 4^b, \;\; 2 \times 4^b, \;\; 6 \times 3^a \times 4^b \qquad\qquad (a, b \geq 0)$$

The corresponding length is

$$5a + 6b - 1, \;\; 4 + 6b - 1, \;\; 8 + 5a + 6b - 1$$

For instance $72 = 2^3 \times 3^2 = 6 \times 3^1 \times 4^1$ has the minimum representation length $8 + 5 + 6 - 1 = 18$:

$$|SSSSSS0 \times SSS0 \times SSSS0| = 18$$

For what numbers of the form $2^a \times 3^b$ is the relative representation length shortest? The "asymptotic answer" is easy; we just have to compare two possible forms of n:

- for $n = 4^a$: $|n| = 6a - 1$ so $|n| = 6 \log_4 n - 1$
- for $n = 3^a$: $|n| = 5a - 1$ so $|n| = 5 \log_3 n - 1$

and it can be easily checked that $6 \log_4 n > 5 \log_3 n$ for $n \geq 2$ (although the difference is not large).

And what about incompressible numbers? It should be obvious that a large enough integer is incompressible iff it is prime. We summarize our results in the following theorem.

Theorem 1 *Consider the arithmetic* $(\mathbb{N}, \times)$ *with the standard representation. The greatest relative compression of an integer occurs for integers of the form* $n = 4^a$ *for which* $C(n) = \log_4 n - 1$*. An integer* $n \geq 10$ *is incompressible iff it is prime.*

Theorem 2 *In the arithmetic* $(\mathbb{N}, \times)$ *with the standard unary representation there is an algorithm that computes* $C(n)$ *in polynomial time.*

Proof. Notice that integers are represented by very long strings, namely $|n| = n + 1$. This contrasts with binary representation where $|n| = 1 + \lfloor \log_2 n \rfloor$. As all complexities are relative to the length of the input, the result follows easily. ◇

Let us define the *relative incompressibility* $\mathrm{C_{rel}}(n)$ of an integer n as $C_{\mathcal{A}}(n)/|n|$; for instance, prime numbers p have $\mathrm{C_{rel}}(p) = 1$. Theorem 1 provides upper and lower bounds for the compressibility of an integer. Figure 1 illustrates the intermediate behaviour. In particular, it can be easily seen that the integers can be classified by their compressibility – or equivalently by their Kolmogorov complexity – as follows:

- Prime numbers p; $\mathrm{C_{rel}}(p) = 1$ (incompressible integers). The asymptotic "density" of this line is $1/\ln n$, see 3.2. Figure 1 this numbers correspond to the horizontal line with height 1.
- For i = 2, 3,... numbers of the form $n = ip$ where p is prime and i is much smaller than p. These integers have relative incompressibility $\mathrm{C_{rel}}(n) = 1/i$ and density $1/(i \ln x)$, see 3.2. They correspond in Figure 1 to the horizontal lines with height 1/2, 1/3, 1/4... Notice that for large n there are no integers with compressibility between 1 and 1/2, between 1/2 and 1/3,...
- Numbers near the theoretical lower bound, $(6 \log_4(n) - 1)/n$, see Figure 2.

Figures 2, 3, 4 and 5 further illustrate the properties of the Kolmogorov function distribution.

Similar results hold for the more general case studied in Section 4.

3.2 Density of integers

Consider large values of n in Figure 1. The most incompressible integers are the primes which correspond to the line $y = 1$ in the figure. Their density is $1/\ln n$, an immediate consequence of the famous *Prime Number Theorem*, see for instance [5, 2]. Thus for large a and for n much larger than a, we expect to find about $a/\ln n$ in the interval $[n, n + a - 1]$.

Integers of the form $n = ip$ where p is prime and $i << p$ have a Kolmogorov complexity of about n/i. In fact, the length of the shortest expression representing of i never exceeds $i + 1$ which is much smaller that the length $p + 1$ of the shortest representation of p. In summary

$$C(n) = C(ip) = C(i) + 1 + C(p) \leq i + 2 + C(p) = i + 2 + p + 1 \approx p = n/i$$

For i = 2, 3,... these classes of numbers are asymptoticly over the lines $y = 1/i$, and it is not difficult to see that for each i the density of the corresponding line is

$$\frac{1}{i \ln(n/i)} = \frac{1}{i(\ln n - \ln i)} \approx \frac{1}{i \ln n}$$

where the approximation is valid for $n >> i$.

The region immediately above the (exact) lower bound is also interesting, see Figure 2. For these numbers the "area density" seems to be well defined.

4 Generalizing the unary representations

Although we have considered a a particular unary representation for the integers and defined the length of an expression as the number of symbols it contains, the results obtained in Section 3 can be easily generalized.

Suppose that the integer n is represented by a string with length $\alpha n + \beta$ with $\alpha > 0$ and that the representation of "$\times$" has length γ. This generalization includes the following examples:

- The standard representation, $\alpha = \beta = \gamma = 1$.
- A representation where the integer n is represented by $\overbrace{S(S(\cdots S(0)\cdots))}^{n\ S\text{'s}}$. In this case $\alpha = 3$ and $\beta = 1$.
- The standard representation where the symbols "0", "S" and "$\times$" are represented by the 2-bit codes `00`, `01` and `10` respectively. Here we have $\alpha = \beta = \gamma = 2$.

We now define a few concepts used in this Section.

Definition 1 *Let E be a product of integers. A* computation step *is the replacement of two integers m and n of E by the result of their product; we say that those integers are* used *in the step and refer to the it as the "step $m \times n$".*

Definition 2 *A* valid computation *or simply a* computation *is a sequence of expressions*

$$E_1 \to E_2 \to \cdots \to E_m$$

where E_1 contain only prime factors, E_m is a shortest length representation and $E_i \to E_{i+1}$ is a computation step for $1 \le i \le m-1$. For $1 \le i \le m$ the sequence

$$E_i \to E_{i+1} \to \cdots \to E_m$$

is called a final part *of a computation. Associated with every computation and with every final part of a computation there is a computation graph as illustrated in Figure 6.*

Definition 3 *An expression E is a* local minimum *if no computation step decreases its length; in particular integers are local minimums.*

Definition 4 *The* gain *associated with the computation step $m \times n$ is the corresponding length reduction:*

$$g(m,n) = (\alpha(m+n) + 2\beta + \gamma) - (\alpha mn + \beta) = \alpha(m + n - mn) + \beta + \gamma$$

Figure 7 shows the value of the gain $g(m,n)$ for particular values of α, β, γ, m and n.

4.1 Complexity of prime numbers

The asymptotic complexity (shortest representation length) of a prime number p is αp; and, more generally a number $n = ip$ with $i << p$ has an asymptotic complexity $\alpha n/i$. Thus the height of horizontal lines in Figures 1 and 3 should be multiplied by α.

4.2 Complexity lower bound

We will now study the general form of the exact lower bound. Assuming that the most compressible integers have the form $n = a^k$, we look for the value of a that minimizes the length

$$l(n) = \alpha n + \beta = k(\alpha a + \beta) + (k-1)\gamma = k(\alpha a + \beta + \gamma) - \gamma$$

for n constant. Express the length as a function of a

$$l(a) = log_a(n)(\alpha a + \beta + \gamma) - \gamma$$

The value of a that minimizes $l(a)$ is the root of

$$\alpha a(\ln^2 a - 1) = \beta + \gamma$$

which is independ of n. Figure 8 shows the solutions of this equation for $0.5 \leq \alpha \leq 2.5$, $0.5 \leq \beta+\gamma \leq 4.5$. We can see for instance that, for $\alpha = \beta = \gamma = 1$ the most compressible integers are of the form 4^k. For these numbers we have (see also Figures 1 and 2)

$$C_{\mathcal{A}}(n) = (\alpha a + \beta + \gamma)\log_a(n) - \gamma = 6\log_a(n) - 1$$

4.3 Algorithms for finding the shortest representation

To find a shortest expression with value n begin as before by writing n as a product of prime factors

$$n = p_1 \times p_2 \times \cdots \times p_m$$

A shortest expression for n can be found by multiplying successively two appropriate factors until no further multiplication reduces the length of the expression.

We begin with two negative results.
An important question is: can we select arbitrarily a pair of factors provided that the length is reduced, and find a shortest expression? The answer is negative. As we have already seen in Example 1 the steps of a computation must in general be performed in the correct order[2].

We may also ask if the following is an algorithm for finding the shortest expression denoting a integer: after expressing the integer as a product of prime factors, successively perform a multiplication that corresponds to the greatest length reduction until no multiplication reduces the length of the expression. The answer is again negative as the following example shows.

Example 2 *Let $\alpha = \beta = 1$, $\gamma = 2$ and consider the following prime factorizing*

$$2 \times 2 \times 3 \times 3$$

[2] Thus there is no result analogous to the "Church-Rosser" of λ-calculus.

The multiplication corresponding to the largest length reduction is 2×2 *but the shortest length is obtained with*

$$2 \times 2 \times 3 \times 3 \;\rightarrow\; 2 \times 3 \times 6 \;\rightarrow\; 6 \times 6 \;\; \textit{(length 16)}$$

If we begin with 2×2*, the final expression is not optimum*

$$2 \times 2 \times 3 \times 3 \;\rightarrow\; 4 \times 3 \times 3 \;\; \textit{(length 17)}$$

Let us now present two positive results.

Lemma 1 *If* $n' > n \geq 2$ *then for any* $m \geq 2$*,* $g(n,m) > g(n',m)$*.*

Proof. As $m \geq 2$ and $n' > n > 0$ we have

$$g(n',m) = \alpha(m - n'(m-1)) + \beta + \gamma < \alpha(m - n(m-1)) + \beta + \gamma = g(n,m)$$

$\diamond$

From this Lemma it follows that if the step $a \times b$ does not decrease the length of an expression and $a' > a$, then the step $a' \times b$ does not also decrease its length.

The following result is a consequence of Theorem 4, but we provide a separate proof.

Theorem 3 *If* $E_1 \rightarrow \cdots \rightarrow E_m$ *is shortest valid computation,* E_m *is a local minimum and for* $1 \leq i \leq m-1$*, no* E_i *is a local minimum.*

Proof. E_m is a local minimum because otherwise there would be a shorter expression representing the same integer.
Now consider a computation step $E_i \rightarrow E_{i+1}$ where say $E_i = f_1 \times \cdots \times f_k$ and $E_{i+1} = (f_1 f_2) \times \cdots \times f_k$. If E_i is a local minimum, no multiplication $f_1 \times f_i$ $(i \geq 3)$ decreases its length so that the same is true (Lemma 1) for multiplications $(f_1 f_2) \times f_i$ in E_{i+1}; and similarly for multiplications $f_2 \times f_i$ $(i \geq 3)$. The other possible multiplications in E_{i+1} are also possible in E_i. It follows that E_{i+1}, E_{i+2},. . . would also be local minimum and have non-decreasing lengths so that either E_m would not be a minimum length expression or there would be a shortest valid computation. $\diamond$

Theorem 4 *No negative gain step is involved in a valid computation.*

Proof. By contradiction. Suppose that valid computations may have negative gain steps and among those computations

$$E_1 \rightarrow E_2 \rightarrow \cdots \rightarrow E_m$$

consider a shortest final part of a computation

$$s: \quad E_i \rightarrow \cdots \rightarrow E_m$$

involving a negative gain step. As this sequence is as short as possible, the negative gain step must be unique and it must be the first one, $E_i \rightarrow E_{i+1}$, say

$$m \times n \times a_1 \times \cdots \times a_l \rightarrow mn \times a_1 \times \cdots \times a_l$$

We consider two cases. If mn is not involved in further steps, a shorter final expression would be obtained if mn in E_l is replaced by $m \times n$.

Otherwise let the first transition involving mn be

$$mn \times a \times b_1 \times \cdots \times b_p \to mna \times b_1 \times \cdots \times b_p$$

where the integers a, b_1,. . . , b_p are products[3] of the a_i By assumption we have $g(m, n) < 0$ and $g(mn, a) \geq 0$. Using Lemma 1 we get

$$g(m, a) > g(nm, a) \geq 0$$

But then we can transform the computation as illustrated in Figure 9. If the computation step $n \times ma$ has a negative gain, then the negative gain step has been pushed down by the transformation, but this contradicts the assumption that the original computation is the shortest one involving a step with negative gain. On the other hand if the computation step $n \times ma$ has a nonnegative gain, a final part of the computation without negative gain steps is possible; this again is a contradiction. $\diamond$

b

4.4 The incompressibility Theorem

As the unary representation is used through this paper, the "incompressibility" Theorem takes a weak form.

Theorem 5 (Incompressibility Theorem) *Consider an arithmetic system $\mathcal{A}$ where the representation of the integers is "unary" and suppose that the representation of n has length $\alpha n + \beta$. One of the integers n with representation length not exceeding L has a Kolmogorov complexity greater than* $\log((L - \beta)/\alpha) - 1$

Proof. For a fixed m there are $N = 2^{m+1} - 1$ binary strings with length at most m. Therefore, as different integers must have different descriptions, there are at most N integers n with $C_{\mathcal{A}}(n) \leq m$. But there are $\lfloor (L - \beta)/\alpha \rfloor$ integers with length at most L. Therefore, for

$$(L - \beta)/\alpha \geq 2^{m+1}$$

at least one of the integers n with representation length not exceeding L must have a Kolmogorov complexity greater than $m = \log((L - \beta)/\alpha) - 1$. $\diamond$

The factorization graph

Let E be a product of factors. As the product is commutative, the expression E can be described as a *multiset* of integers (the factors). Using this description we now define *reduction step* and *factor graph.*

Notice that the length of the expression E can be expressed in terms of the corresponding multiset:

$$|E| = \alpha \Sigma_{x \in E} x + (\beta + \gamma)\#E - \gamma$$

where $\#E$ denotes the cardinality of E (the number of factors).

[3] In fact they are exactly the initial factors a_i's and not products of two or more of them, for otherwise there would be a shorter final part of a computation with a negative gain step. But this fact is not important in the proof.

Definition 5 *We say that E reduces to E' using x, $y \in E$ and write $E \to E'$ if*

$$E' = E \setminus \{x, y\} \cup \{xy\}$$

Notce that $(\#E') = (\#E) - 1$.

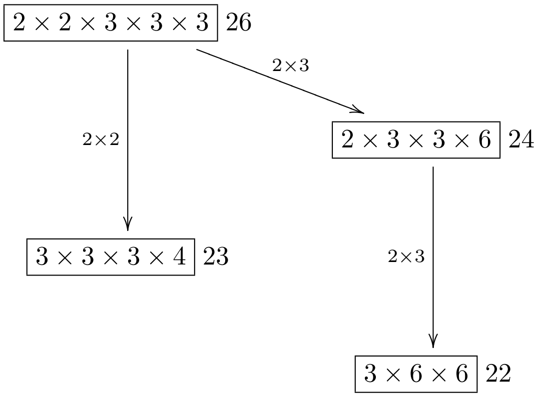
Definition 6 *Let E be a product of factors (multiset of integers). The* factor graph *associated to E is the directed graph defined by the set of nodes*

$$V = \{F \ :\ E \to^\star E'\}$$

and set of edges $\{(F, G) \ :\ F \to G\}$.

The factor graph is clearly acyclic (because $E \to E'$ implies $(\#E') = (\#E) - 1$) and has E as the only node without ancestor. Two observations further elucidate the structure of E. First, the possibility of "performing" reductions (products) of disjoints pairs of nodes in any order and the associativity of multiplication.

In the following example $E = 2 \times 2 \times 3 \times 3 \times 3$, $\alpha = \beta = 1$, $\gamma = 2$. We get $g(2, 2) = 3$, $g(2, 3) = 2$, and $g(x, y) \leq 0$ for $x \geq 3$ or $y \geq 3$. Only the reductions with positive gain are represented.


$2 \times 2 \times 3 \times 3 \times 3$ 26
2×3
2×2
$2 \times 3 \times 3 \times 6$ 24
$3 \times 3 \times 3 \times 4$ 23
2×3
$3 \times 6 \times 6$ 22

5 A note on other arithmetic systems

We briefly comment on two other arithmetic formal systems obtained by adding respectively the successor functor and the "+" operator to the multiplicative unary system. In terms of order of magnitude, the Kolmogorov complexity of an integer relative to these systems is the same (logarithmic on the length of the input integer).

5.1 A note on other the arithmetic system $(\mathbb{N}, S, \times)$

Adding the unary successor function S changes dramaticly the results obtained before. As an example, a prime with the form $p = 2^{2n} + 1$ which in the multiplicative arithmetic is incompressible, can now be highly compressed, $p = S(4^n)$. In fact every integer n has a Kolmogorov complexity $O(\log n)$, see Theorem 6.

The well formed expressions are now defined by the following syntax

$$\langle \texttt{exp} \rangle \quad \rightarrow \quad 0 \mid S(\langle \texttt{exp} \rangle) \mid \langle \texttt{exp} \rangle \times \langle \texttt{exp} \rangle$$

A typical expression is

$$S(S(\overline{4} \times \overline{4} \times \overline{4}) \times S(S(\overline{3} \times \overline{4})))$$

where $\overline{3}$ and $\overline{4}$ denote $S(S(S(0)))$ and $S(S(S(S(0))))$ respectively.

Theorem 6 *In* $(\mathbb{N}, S, \times)$, *for every integer* n, $C_{\mathcal{A}} \in \Theta(\log n)$.

Proof. One of the inequalities implicit in the Θ notation follows from Theorem 5. The other results from the observation that an integer n with a binary representation $b_m b_{m-1} \cdots b_1 b_0$ can be represented by the following expression:

1. Multiply b_m by 2 and if $b_{m-1} = 1$ take the successor of the result. Let E be the resulting expression. For instance

b_m	b_{m-1}	Expression
1	0	$S(0) \times S(S(0))$
1	1	$S(S(0) \times S(S(0)))$

2. Multiply E by 2 and if $b_{m-2} = 1$ take the successor of the result. Assign this new expression to E. For instance

b_m	b_{m-1}	b_{m-2}	Expression
1	0	1	$S((S(0) \times S(S(0))) \times S(S(0)))$
1	1	0	$S(S(0) \times S(S(0))) \times S(S(0))$

3. Continue this construction until b_0 is considered.

The number $m+1$ of bits is $\Theta(\log n)$ and in each step the length of the expression increases at most 11 so that the final length is $\Theta(\log n)$. ◇

5.2 A note on the arithmetic system $(\mathbb{N}, +, \times)$

With addition, the possibilities for expressing a given integer increase enormously. However, in terms of Kolmogorov complexity we have again a result similar to Theorem 6.

Theorem 7 *In* $(\mathbb{N}, +, \times)$, *for every integer* n, $C_{\mathcal{A}} \in \Theta(\log n)$.

Proof. The proof is similar to the proof of Theorem 6. In this case, an integer n with basis 2 representation is $b_m b_{m-1} \cdots b_1 b_0$ can be represented by the expression

$$((\cdots(\overline{b_m} \times \overline{2} + \overline{b_{m-1}}) \times \overline{2} + \cdots) \times \overline{2}) + \overline{b_0}$$

where $\overline{x}$ denotes the representation of n, that is, $\overline{2} = SS0$ and $\overline{b_i}$ is either 0 or $S0$. ◇

6 Conclusion and open problems

In this work we have we have studied the Kolmogorov complexity when the "model of computation" is the unary arithmetic multiplicative system and studied problems related to its computation, see in particular the counter-examples in Section 3 and Theorems 3 and 4.

We also presented a brief study of the multiplicative system with successor or addition, see Theorems 6 and 7. A weak form of the incompressibility Theorem (Theorem 5) was also given.

As far as we know there are no previous work on this study in this area and a lot of work remains to be done, namely a more thorough study of complexity of computing $C(n)$, the study of the "density" of integers near the lower bound (which seems to be well defined, see Figure 2) and the use of arithmetic systems with exponentiation and integers written in binary.

References

[1] L. Berman. Precise bounds for Presburger arithmetic and the reals with addition (preliminary report). In *International Conference of Foundations of Computer Science*, pages 95–99, 1977.

[2] H. Davenport. *Multiplicative Number Theory.* Springer-Verlag, second edition, 1980. (Chapter 18, "Prime Number Theorem").

[3] Jeanne Ferrante and Charles Rackoff. *The Computational Complexity of Logical Theories*, volume 718 of *Lecture Notes in Mathematics.* Springer-Verlag, 1979.

[4] M. Fischer and M. Rabin. Super-exponential complexity of Presburger arithmetic. In *SIAM-AMS Proceedings 7*, pages 27–41, 1974.

[5] A. Ingham. *The distribution of prime numbers.* Cambridge University Press, 1990.

[6] Ming Li and Paul M. B. Vitányi. *An Introduction to Kolmogorov Complexity and its Application.* Texts and Monographs in Computer Science. Springer-Verlag, second edition, 1997.

[7] Michael O. Rabin. Decidable theories. In J. Barwise, editor, *Handbook of Theoretical Computer Science.* Elsevier Science Publishers, 1977.

Figure 1: The value of $C_{\mathcal{A}}(n)/|n|$ for integers n between 100 and 10 000, where $\mathcal{A}$ is the multiplicative algebra mentioned in the text and $|n| = n + 1$ denotes the length of the representation of n. The incompressible integers are the primes (horizontal line $y = 1$); the integers that are the double of primes are 50% compressible (horizontal line $y = 1/2$), etc. The most compressible integers are over the curve $y = (6 \log_4(n) - 1)/n$; here, a more chaotic distribution is apparent.

Figure 2: The value of $C_{\mathcal{A}}(n)/|n|$ for integers n between 20 000 and 25 000 near the lower bound $y = (6\log_4(n) - 1)/x$. This region can be called “chaotic”

Figure 3: The value of $C_{\mathcal{A}}(n)/|n|$ for integers n between 90 000 and 100 000 near the lower bound $y = (6\log_4(n) - 1)/x$. The transition between the "line" and "chaotic" regions is illustrated. The lower bound curve is near the horizontal axis.

Figure 4: The horizontal axis, representing $C_{\mathcal{A}}(n)$ is divided in 2000 equal intervals; for each interval the height of the corresponding bar is the number of integers n between 90 000 and 100 000 with $C_{\mathcal{A}}(n)$ in the corresponding interval. Notice that the height of most bars is 0. The sum of the heights of the 2 rightmost adjacent intervals (in the graph they seem to be just one bar) is the number of prime numbers in that range.

Figure 5: A cumulative version of Figure 4 (integers between 90 000 and 100 000). In contrast with the usual Kolmogorov complexity, it is apparent that most integers are highly compressible.

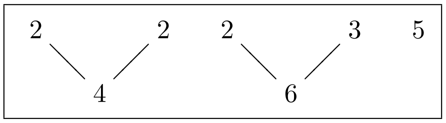


Figure 6: The graph corresponding to the computation $2 \times 2 \times 2 \times 3 \times 5 \to 4 \times 2 \times 3 \times 5 \to 4 \times 6 \times 5$ in an multiplicative system with $\alpha = \beta = \gamma = 1$. From this computation we have $K_{\mathcal{A}}(120) = |\overline{4}| + 1 + |\overline{6}| + 1 + |\overline{5}| = 20$. The same graph may of course correspond to more than one computation.

	2	3	4	5	6	7	8	9	10	11	12	13
2	10	9	8	7	6	5	4	3	2	1	0	-1
3	9	7	5	3	1	-1	-3	-5	-7	-9	-11	-13
4	8	5	2	-1	-4	-7	-10	-13	-16	-19	-22	-25
5	7	3	-1	-5	-9	-13	-17	-21	-25	-29	-33	-37
6	6	1	-4	-9	-14	-19	-24	-29	-34	-39	-44	-49
7	5	-1	-7	-13	-19	-25	-31	-37	-43	-49	-55	-61
8	4	-3	-10	-17	-24	-31	-38	-45	-52	-59	-66	-73
9	3	-5	-13	-21	-29	-37	-45	-53	-61	-69	-77	-85
10	2	-7	-16	-25	-34	-43	-52	-61	-70	-79	-88	-97
11	1	-9	-19	-29	-39	-49	-59	-69	-79	-89	-99	-109
12	0	-11	-22	-33	-44	-55	-66	-77	-88	-99	-110	-121
13	-1	-13	-25	-37	-49	-61	-73	-85	-97	-109	-121	-133

Figure 7: The value of the gain $g(m, n)$ for $\alpha = 1$, $\beta + \gamma = 10$, and m and n between 2 and 13.

Figure 8: The vertical axis represents the integer a for which $n = a^k$ is most compressible. The value of α is in the horizontal axis labeled from 0.5 to 2.5 while the value of $\beta + \gamma$ is in the other horizontal axis (labeled from 0.5 to 4.5). In this region of the (α, $\beta + \gamma$) horizontal plane the optimum values of a are comprised between 3 and 8.

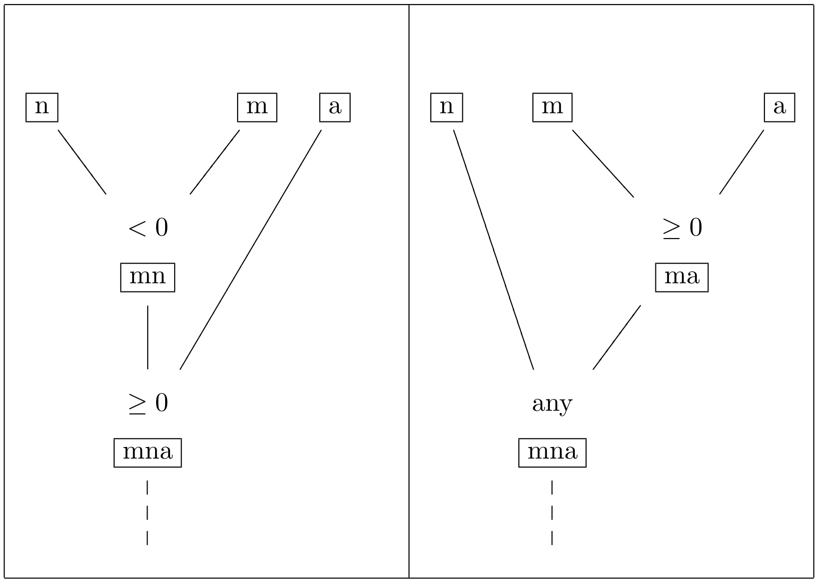


Figure 9: Two computation graphs differing in the way of obtaining mna. The relevant part of the original computation is at left: after the step with negative gain $n \times m$, there is a nonnegative gain step $mn \times a$, where a is is an original factor. At right is the corresponding part of the transformed computation: a step with positive gain $m \times a$ is followed by step $n \times ma$, see text. The sign of step gain – negative, nonnegative or unknown – is written above the resulting integer. The rest of the computation graph is identical on both sides; in particular the final expression (not represented) is the same.