There exists a non-recursively enumerable subset of $\mathbb{N}$ which has a short definition in terms of arithmetic

Apoloniusz Tyszka
Hugo Kołłątaj University
Balicka 116B, 30-149 Kraków, Poland
E-mail: *rttyszka@cyf-kr.edu.pl*

Abstract

We prove that the sets $\{n \in \mathbb{N}$: n satisfies condition **(1)**$\}$ and $\{n \in \mathbb{N} : n$ does not satisfy condition **(2)**$\}$ are not recursively enumerable. **(1)** $\exists p, q \in \mathbb{N}\ ((n = \frac{1}{2}(p+q)(p+q+1)+q)) \wedge \forall (x_0, \ldots, x_p) \in \mathbb{N}^{p+1}\ \exists (y_0, \ldots, y_p) \in \{0, \ldots, q\}^{p+1}\ \forall i, j, k \in \{0, \ldots, p\}\ (((x_j + 1 = x_k) \Rightarrow (y_j + 1 = y_k)) \wedge ((x_i \cdot x_j = x_k) \Rightarrow (y_i \cdot y_j = y_k))))$. **(2)** $\exists (y_0, \ldots, y_n) \in \mathbb{N}^{n+1}\ \forall i, j, k \in \{0, \ldots, n\}\ (((2^{2^{2^j} \cdot 3^k} + 1 \text{ divides } n) \Rightarrow (y_j + 1 = y_k)) \wedge ((2^{2^{2^i} \cdot 3^j \cdot 5^k + 1} + 1 \text{ divides } n) \Rightarrow (y_i \cdot y_j = y_k)))$. By using Gödel's β function, we can translate the phrase "*a non-negative integer* n *satisfies condition* **(1)**" into a first-order formula of arithmetic. For $n \in \mathbb{N}$, let $E_n = \{1 = x_k,\ x_i + x_j = x_k,\ x_i \cdot x_j = x_k :\ i, j, k \in \{0, \ldots, n\}\}$. For $n \in \mathbb{N}$, $f(n)$ denotes the smallest $b \in \mathbb{N}$ such that if a system of equations $S \subseteq E_n$ has a solution in $\mathbb{N}^{n+1}$, then S has a solution in $\{0, \ldots, b\}^{n+1}$. The author proved earlier that the function $f : \mathbb{N} \to \mathbb{N}$ is computable in the limit and eventually dominates every computable function $g : \mathbb{N} \to \mathbb{N}$. We present a short program in *MuPAD* which for $n \in \mathbb{N}$ prints the sequence $\{f_i(n)\}_{i=0}^{\infty}$ of non-negative integers converging to $f(n)$. For $n \in \mathbb{N}$, $\gamma(n)$ denotes the smallest $b \in \mathbb{N}$ such that if a system of equations $S \subseteq E_n$ has a unique solution in $\mathbb{N}^{n+1}$, then this solution belongs to $\{0, \ldots, b\}^{n+1}$. The author proved earlier that the function $\gamma : \mathbb{N} \to \mathbb{N}$ is computable in the limit and eventually dominates every function $\delta : \mathbb{N} \to \mathbb{N}$ with a single-fold Diophantine representation. We present a short program in *MuPAD* which for $n \in \mathbb{N}$ prints the sequence $\{\gamma_i(n)\}_{i=0}^{\infty}$ of non-negative integers converging to $\gamma(n)$.

2020 Mathematics Subject Classification: 03D25.

Key words and phrases: arithmetic of $\mathbb{N}$, computable function, eventual domination, Gödel's β function, Hilbert's 10th problem, limit-computable function, recursively enumerable set, single-fold Diophantine representation, undecidable decision problem.

1 A program in *MuPAD* which has five lines and computes in the limit a function $\gamma : \mathbb{N} \to \{0, 1\}$ of unknown computability

Semi-algorithms differ from algorithms, as they may not terminate.

Definition 1. *(cf. [13, pp. 233–235]). A computation in the limit of a function $f : \mathbb{N} \to \mathbb{N}$ is a semi-algorithm which takes as input a non-negative integer n and for every $m \in \mathbb{N}$ prints a non-negative integer $\xi(n, m)$ such that $\lim_{m \to \infty} \xi(n, m) = f(n)$.*

By Definition 1, a function $f : \mathbb{N} \to \mathbb{N}$ is computable in the limit when there exists an infinite computation which takes as input a non-negative integer n and prints a non-negative integer on each iteration and prints $f(n)$ on each sufficiently high iteration.

It is known that there exists a limit-computable function $f : \mathbb{N} \to \mathbb{N}$ which is not computable, see Theorem 2. Every known proof of this fact does not lead to the existence of a short computer program that computes f in the limit. So far, short computer programs can only compute in the limit functions from $\mathbb{N}$ to $\mathbb{N}$ whose computability is proven or unknown, see Theorem 1

Lemma 1. *For every* $n \in \mathbb{N}$*,*

$$\frac{sgn(n-1)\cdot(2n+(1-(-1)^n)\cdot(5n+2))}{4} = \begin{cases} 0, & \text{if } n=1 \\ \frac{n}{2}, & \text{if } n \text{ is even} \\ 3n+1, & \text{if } n \text{ is odd and } n \neq 1 \end{cases}$$

MuPAD is a part of the Symbolic Math Toolbox in MATLAB R2019b.

Theorem 1. *The following program in MuPAD computes in the limit a function* $\gamma : \mathbb{N} \to \{0,1\}$*.*

```
input("Input a non-negative integer n",n):
while TRUE do
print(sign(n)):
n:=sign(n-1)*(2*n+(1-(-1)^n)*(5*n+2))/4:
end_while:
```

Proof. It follows from Lemma 1. □

The computability of γ is unknown, see [1, p. 79]. The Collatz conjecture implies that $\gamma(n) = 0$ for every $n \in \mathbb{N}$.

2 A limit-computable function $f : \mathbb{N} \to \mathbb{N}$ which eventually dominates every computable function $g : \mathbb{N} \to \mathbb{N}$

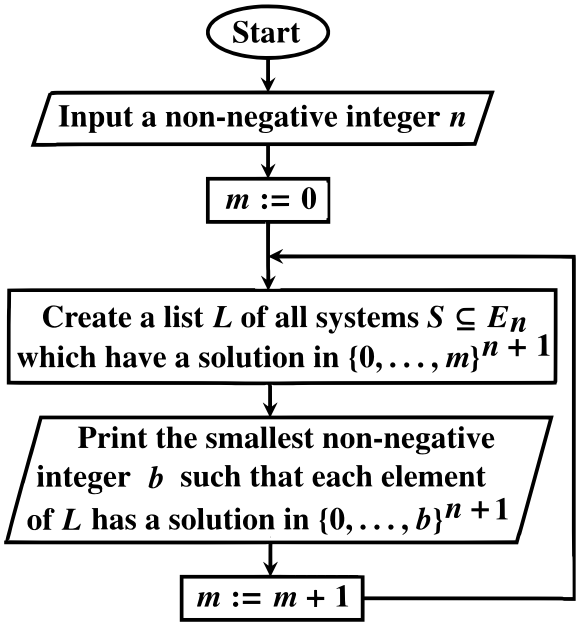
For $n \in \mathbb{N}$, let

$$E_n = \{1 = x_k,\ x_i + x_j = x_k,\ x_i \cdot x_j = x_k :\ i,j,k \in \{0,\dots,n\}\}$$

Theorem 2. *([11, p. 118]). There exists a limit-computable function* $f : \mathbb{N} \to \mathbb{N}$ *which eventually dominates every computable function* $g : \mathbb{N} \to \mathbb{N}$*.*

We present an alternative proof of Theorem 2. For $n \in \mathbb{N}$, $f(n)$ denotes the smallest $b \in \mathbb{N}$ such that if a system of equations $S \subseteq E_n$ has a solution in $\mathbb{N}^{n+1}$, then S has a solution in $\{0,\dots,b\}^{n+1}$. The function $f : \mathbb{N} \to \mathbb{N}$ is computable in the limit and eventually dominates every computable function $g : \mathbb{N} \to \mathbb{N}$, see [15]. The term *"dominated"* in the title of [15] means *"eventually dominated"*.

Theorem 3. *([15]). Flowchart 1 shows a semi-algorithm which computes $f(n)$ in the limit.*



Flowchart 1

A semi-algorithm which computes $f(n)$ in the limit

Definition 2. *An approximation of a tuple $(x_0, \ldots, x_n) \in \mathbb{N}^{n+1}$ is a tuple $(y_0, \ldots, y_n) \in \mathbb{N}^{n+1}$ such that*

$$(\forall k \in \{0, \ldots, n\}\ (1 = x_k \Rightarrow 1 = y_k)) \wedge$$

$$(\forall i, j, k \in \{0, \ldots, n\}\ (x_i + x_j = x_k \Rightarrow y_i + y_j = y_k)) \wedge$$

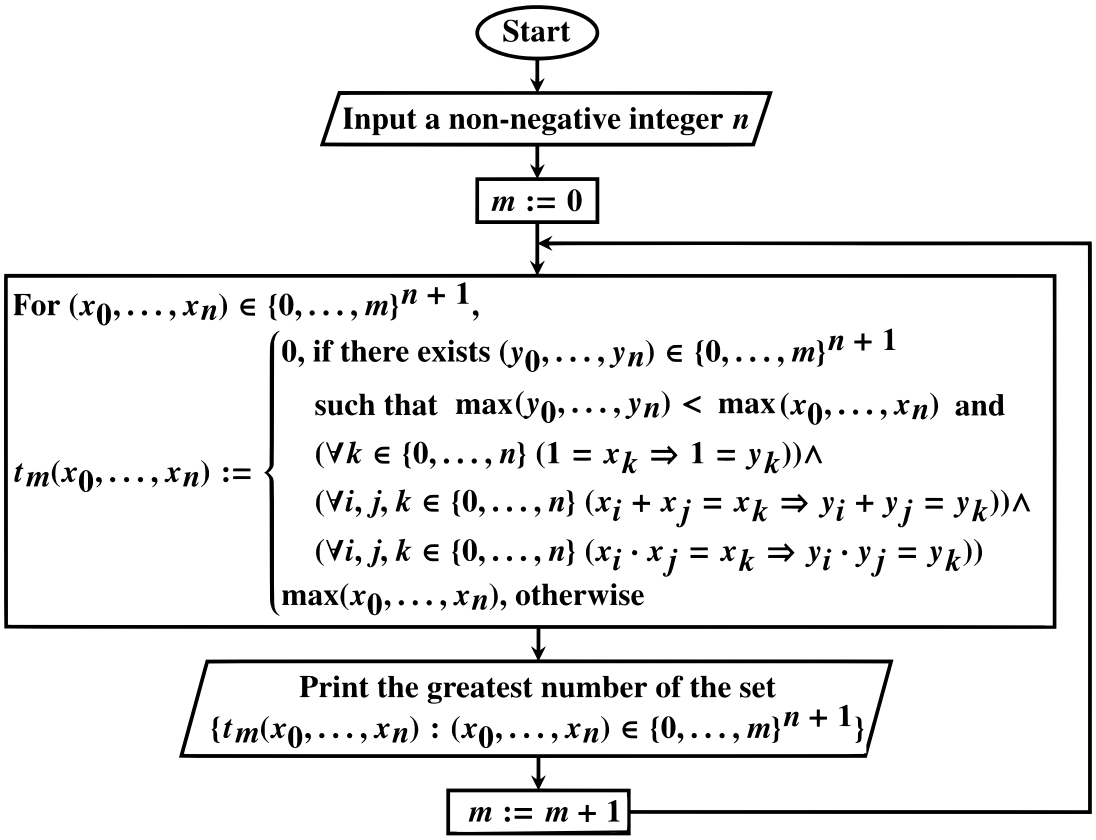
$$(\forall i, j, k \in \{0, \ldots, n\}\ (x_i \cdot x_j = x_k \Rightarrow y_i \cdot y_j = y_k))$$

Observation 1. *For every $n \in \mathbb{N}$, there exists a set $A(n) \subseteq \mathbb{N}^{n+1}$ such that*

$$\mathrm{card}(A(n)) \leqslant 2^{\mathrm{card}(E_n)} = 2^{n+1+2\cdot(n+1)^3}$$

and every tuple $(x_0, \ldots, x_n) \in \mathbb{N}^{n+1}$ possesses an approximation in $A(n)$.

Flowchart 2 shows a simpler semi-algorithm which computes $f(n)$ in the limit.



Flowchart 2

A simpler semi-algorithm which computes $f(n)$ in the limit

Lemma 2. *For every $n, m \in \mathbb{N}$, the number printed by Flowchart 2 does not exceed the number printed by Flowchart 1.*

Proof. For every $(a_0, \ldots, a_n) \in \{0, \ldots, m\}^{n+1}$,

$$E_n \supseteq \{1 = x_k : \ (k \in \{0, \ldots, n\}) \wedge (1 = a_k)\} \cup$$

$$\{x_i + x_j = x_k : \ (i, j, k \in \{0, \ldots, n\}) \wedge (a_i + a_j = a_k)\} \cup$$

$$\{x_i \cdot x_j = x_k : \ (i, j, k \in \{0, \ldots, n\}) \wedge (a_i \cdot a_j = a_k)\}$$

□

Lemma 3. *For every $n, m \in \mathbb{N}$, the number printed by Flowchart 1 does not exceed the number printed by Flowchart 2.*

Proof. Let $n, m \in \mathbb{N}$. For every system of equations $S \subseteq E_n$, if $(a_0, \ldots, a_n) \in \{0, \ldots, m\}^{n+1}$ and $(a_0, \ldots, a_n)$ solves S, then $(a_0, \ldots, a_n)$ solves the following system of equations:

$$\{1 = x_k : \ (k \in \{0, \ldots, n\}) \wedge (1 = a_k)\} \cup$$

$$\{x_i + x_j = x_k : \ (i, j, k \in \{0, \ldots, n\}) \wedge (a_i + a_j = a_k)\} \cup$$

$$\{x_i \cdot x_j = x_k : \ (i, j, k \in \{0, \ldots, n\}) \wedge (a_i \cdot a_j = a_k)\}$$

□

Theorem 4. *For every $n, m \in \mathbb{N}$, Flowcharts 1 and 2 print the same number.*

Proof. It follows from Lemmas 2 and 3. □

Corollary 1. *For every $n, m \in \mathbb{N}$, Flowcharts 1 and 2 print the smallest $b \in \{0, \ldots, m\}$ such that every tuple $(x_0, \ldots, x_n) \in \{0, \ldots, m\}^{n+1}$ possesses an approximation in $\{0, \ldots, b\}^{n+1}$.*

Theorem 5. *For every $n \in \mathbb{N}$, $f(n)$ is the smallest $b \in \mathbb{N}$ such that every tuple $(x_0, \ldots, x_n) \in \mathbb{N}^{n+1}$ possesses an approximation in $\{0, \ldots, b\}^{n+1}$.*

Proof. It follows from Theorem 3 and Corollary 1. □

The following program in *MuPAD* implements the semi-algorithm shown in Flowchart 2.

```
input("Input a non-negative integer n",n):
m:=0:
while TRUE do
X:=combinat::cartesianProduct([s $s=0..m] $t=0..n):
Y:=[max(op(X[u])) $u=1..(m+1)^(n+1)]:
for p from 1 to (m+1)^(n+1) do
for q from 1 to (m+1)^(n+1) do
v:=1:
for k from 1 to n+1 do
if 1=X[p][k] and 1<>X[q][k] then v:=0 end_if:
for i from 1 to n+1 do
for j from i to n+1 do
if X[p][i]+X[p][j]=X[p][k] and X[q][i]+X[q][j]<>X[q][k] then v:=0 end_if:
if X[p][i]*X[p][j]=X[p][k] and X[q][i]*X[q][j]<>X[q][k] then v:=0 end_if:
end_for:
end_for:
end_for:
if max(op(X[q]))<max(op(X[p])) and v=1 then Y[p]:=0 end_if:
end_for:
end_for:
print(max(op(Y))):
m:=m+1:
end_while:
```

3 The first example of a non-recursively enumerable subset of $\mathbb{N}$ which has a short definition in terms of arithmetic

Theorem 6. *No algorithm takes as input non-negative integers n and m and decides whether or not*

$$\forall (x_0, \ldots, x_n) \in \mathbb{N}^{n+1} \ \exists (y_0, \ldots, y_n) \in \{0, \ldots, m\}^{n+1}$$
$$((\forall k \in \{0, \ldots, n\} \ (1 = x_k \Rightarrow 1 = y_k)) \wedge$$
$$(\forall i, j, k \in \{0, \ldots, n\} \ (x_i + x_j = x_k \Rightarrow y_i + y_j = y_k)) \wedge$$
$$(\forall i, j, k \in \{0, \ldots, n\} \ (x_i \cdot x_j = x_k \Rightarrow y_i \cdot y_j = y_k)))$$

Proof. Since the function f is not computable, it follows from Theorem 5. □

Theorem 7. *No algorithm takes as input a non-negative integer n and decides whether or not*

$$\exists p, q \in \mathbb{N}\ ((n+1 = 2^p \cdot (2q+1)) \wedge$$
$$\forall (x_0, \ldots, x_p) \in \mathbb{N}^{p+1}\ \ \exists (y_0, \ldots, y_p) \in \{0, \ldots, q\}^{p+1}$$
$$((\forall k \in \{0, \ldots, p\}\ (1 = x_k \Rightarrow 1 = y_k)) \wedge$$
$$(\forall i, j, k \in \{0, \ldots, p\}\ (x_i + x_j = x_k \Rightarrow y_i + y_j = y_k)) \wedge$$
$$(\forall i, j, k \in \{0, \ldots, p\}\ (x_i \cdot x_j = x_k \Rightarrow y_i \cdot y_j = y_k))))$$

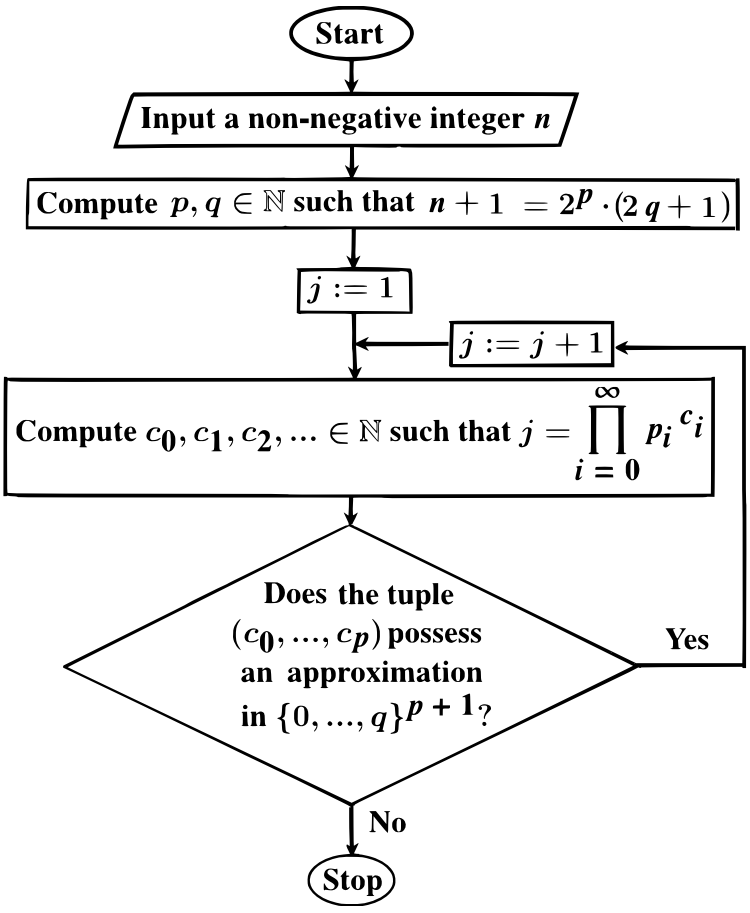
Proof. It follows from Theorem 6. □

Let

$$T_1 = \{n \in \mathbb{N} :\ \exists p, q \in \mathbb{N}\ ((n+1 = 2^p \cdot (2q+1)) \wedge$$
$$\forall (x_0, \ldots, x_p) \in \mathbb{N}^{p+1}\ \ \exists (y_0, \ldots, y_p) \in \{0, \ldots, q\}^{p+1}$$
$$((\forall k \in \{0, \ldots, p\}\ (1 = x_k \Rightarrow 1 = y_k)) \wedge$$
$$(\forall i, j, k \in \{0, \ldots, p\}\ (x_i + x_j = x_k \Rightarrow y_i + y_j = y_k)) \wedge$$
$$(\forall i, j, k \in \{0, \ldots, p\}\ (x_i \cdot x_j = x_k \Rightarrow y_i \cdot y_j = y_k))))\}$$

Theorem 8. *The set $\mathbb{N} \setminus T_1$ is recursively enumerable.*

Proof. For $i \in \mathbb{N}$, let p_i denote the i-th prime number. Flowchart 3 shows a semi-algorithm which takes as input $n \in \mathbb{N}$ and terminates if and only if $n \in \mathbb{N} \setminus T_1$.



Flowchart 3

A semi-algorithm which takes as input $n \in \mathbb{N}$ and terminates if and only if $n \in \mathbb{N} \setminus T_1$

□

Theorem 9. *The set T_1 is not recursively enumerable.*

Proof. It follows from Theorems 7 and 8. □

Lemma 4. *([12, p. 7]). For every $x, y \in \mathbb{N}$, $x \neq y$ implies that the numbers $2^{2^x} + 1$ and $2^{2^y} + 1$ are relatively prime.*

Theorem 10. *The formula that defines the set T_1 can be translated into a first-order formula of arithmetic.*

Proof. For $x \in \mathbb{N} \setminus \{0\}$ and $r \in \mathbb{N}$, let $e(x, r)$ denote the greatest $y \in \mathbb{N}$ such that $(2^{2^r} + 1)^y$ divides x. By Lemma 4, the set T_1 consists of all $n \in \mathbb{N}$ such that

$$\forall w \in \mathbb{N} \setminus \{0\}\ \exists m \in \mathbb{N} \setminus \{0\}\ \exists p, q \in \mathbb{N}\ ((n + 1 = 2^p \cdot (2q + 1)) \wedge$$

$$\forall i, j, k \in \{0, \ldots, p\}\ ((1 = e(w, k) \Rightarrow 1 = \min(e(m, k), q)) \wedge$$

$$(e(w, i) + e(w, j) = e(w, k) \Rightarrow \min(e(m, i), q) + \min(e(m, j), q) = \min(e(m, k), q)) \wedge$$

$$(e(w, i) \cdot e(w, j) = e(w, k) \Rightarrow \min(e(m, i), q) \cdot \min(e(m, j), q) = \min(e(m, k), q))))$$

The above formula can be translated into a first-order formula of arithmetic. □

Lemma 5. *([10]). The function*

$$\mathbb{N}^2 \ni (p, q) \to \frac{1}{2}(p + q)(p + q + 1) + q \in \mathbb{N}$$

is bijective.

Let

$$T = \{n \in \mathbb{N} :\ \exists p, q \in \mathbb{N}\ ((n = \frac{1}{2}(p + q)(p + q + 1) + q) \wedge$$

$$\forall (x_0, \ldots, x_p) \in \mathbb{N}^{p+1}\ \ \exists (y_0, \ldots, y_p) \in \{0, \ldots, q\}^{p+1}$$

$$((\forall k \in \{0, \ldots, p\}\ (1 = x_k \Rightarrow 1 = y_k)) \wedge$$

$$(\forall i, j, k \in \{0, \ldots, p\}\ (x_i + x_j = x_k \Rightarrow y_i + y_j = y_k)) \wedge$$

$$(\forall i, j, k \in \{0, \ldots, p\}\ (x_i \cdot x_j = x_k \Rightarrow y_i \cdot y_j = y_k))))\}$$

Similarly as in Theorem 9, the set T is not recursively enumerable. The proof of it uses Lemma 5.

Definition 3. *([6]). Let $\boldsymbol{\beta} : \mathbb{N}^3 \to \mathbb{N}$ denote Gödel's $\boldsymbol{\beta}$ function. For $x_1, x_2, x_3 \in \mathbb{N}$, $\boldsymbol{\beta}(x_1, x_2, x_3)$ equals the remainder after integer division of x_1 by $1 + (x_3 + 1) \cdot x_2$.*

Lemma 6. *([6]). If $(d_0, \ldots, d_n) \in \mathbb{N}^{n+1}$, then $\exists b, c \in \mathbb{N}\ \forall i \in \{0, \ldots, n\}\ \boldsymbol{\beta}(b, c, i) = d_i$.*

Theorem 11. *The formula that defines the set T can be translated into a first-order formula of arithmetic. It can be proved without using the theorem that exponentiation can be defined by an arithmetical formula.*

Proof. By Lemma 6, the set T consists of all $n \in \mathbb{N}$ such that

$$\forall u, v \in \mathbb{N}\ \exists a, b, p, q \in \mathbb{N}\ ((n = \frac{1}{2}(p + q)(p + q + 1) + q) \wedge$$

$$\forall i, j, k \in \{0, \ldots, p\}\ ((1 = \boldsymbol{\beta}(u, v, k) \Rightarrow 1 = \min(\boldsymbol{\beta}(a, b, k), q)) \wedge$$

$$(\boldsymbol{\beta}(u, v, i) + \boldsymbol{\beta}(u, v, j) = \boldsymbol{\beta}(u, v, k) \Rightarrow \min(\boldsymbol{\beta}(a, b, i), q) + \min(\boldsymbol{\beta}(a, b, j), q) = \min(\boldsymbol{\beta}(a, b, k), q)) \wedge$$

$$(\boldsymbol{\beta}(u, v, i) \cdot \boldsymbol{\beta}(u, v, j) = \boldsymbol{\beta}(u, v, k) \Rightarrow \min(\boldsymbol{\beta}(a, b, i), q) \cdot \min(\boldsymbol{\beta}(a, b, j), q) = \min(\boldsymbol{\beta}(a, b, k), q))))$$

The above formula does not contain the exponentiation function. □

4 The second example of a non-recursively enumerable subset of $\mathbb{N}$ which has a short definition in terms of arithmetic

Lemma 7. *([14, p. 110], cf. Lemma 9). For non-negative integers, the equation $x + y = z$ is equivalent to a system which consists of equations of the forms $v + 1 = w$ and $u \cdot v = w$.*

For $n \in \mathbb{N}$, $h(n)$ denotes the smallest $b \in \mathbb{N}$ such that if a system of equations $S \subseteq \{x_j + 1 = x_k,\ x_i \cdot x_j = x_k :\ i, j, k \in \{0, \ldots, n\}\}$ has a solution in $\mathbb{N}^{n+1}$, then S has a solution in $\{0, \ldots, b\}^{n+1}$. From Lemma 7 and [15], it follows that the function $h : \mathbb{N} \to \mathbb{N}$ is computable in the limit and eventually dominates every computable function $g : \mathbb{N} \to \mathbb{N}$.

Theorem 12. *No algorithm takes as input non-negative integers n and m and decides whether or not*

$$\forall (x_0, \ldots, x_n) \in \mathbb{N}^{n+1}\ \exists (y_0, \ldots, y_n) \in \{0, \ldots, m\}^{n+1}$$
$$((\forall j, k \in \{0, \ldots, n\}\ (x_j + 1 = x_k \Rightarrow y_j + 1 = y_k)) \wedge$$
$$(\forall i, j, k \in \{0, \ldots, n\}\ (x_i \cdot x_j = x_k \Rightarrow y_i \cdot y_j = y_k)))$$

Proof. It holds because the function h is not computable, and for every $n \in \mathbb{N}$, $h(n)$ is the smallest $b \in \mathbb{N}$ such that

$$\forall (x_0, \ldots, x_n) \in \mathbb{N}^{n+1}\ \exists (y_0, \ldots, y_n) \in \{0, \ldots, b\}^{n+1}$$
$$((\forall j, k \in \{0, \ldots, n\}\ (x_j + 1 = x_k \Rightarrow y_j + 1 = y_k)) \wedge$$
$$(\forall i, j, k \in \{0, \ldots, n\}\ (x_i \cdot x_j = x_k \Rightarrow y_i \cdot y_j = y_k)))$$

□

Theorem 13. *No algorithm takes as input a non-negative integer n and decides whether or not*

$$\exists p, q \in \mathbb{N}\ ((n = 2^p \cdot 3^q) \wedge$$
$$\forall (x_0, \ldots, x_p) \in \mathbb{N}^{p+1}\ \exists (y_0, \ldots, y_p) \in \{0, \ldots, q\}^{p+1}$$
$$((\forall j, k \in \{0, \ldots, p\}\ (x_j + 1 = x_k \Rightarrow y_j + 1 = y_k)) \wedge$$
$$(\forall i, j, k \in \{0, \ldots, p\}\ (x_i \cdot x_j = x_k \Rightarrow y_i \cdot y_j = y_k))))$$

Proof. It follows from Theorem 12. □

Let T_2 denote the algorithmically undecidable subset of $\mathbb{N}$ considered in Theorem 13. Similarly as in Theorem 9, the set T_2 is not recursively enumerable. Similarly as in Theorem 10, the formula that defines the set T_2 can be translated into a first-order formula of arithmetic.

5 The third example of a non-recursively enumerable subset of $\mathbb{N}$ which has a short definition in terms of arithmetic

Let $[\cdot]$ denote the integer part of the argument.

Lemma 8. *The function*

$$\mathbb{N} \ni n \xrightarrow{\theta} (n - [\sqrt{n}]^2,\ |n - [\sqrt{n}]^2 - [\sqrt{n}]|) \in \mathbb{N}^2$$

is surjective.

Proof. It holds because $\theta(\{i^2 + j :\ (i, j \in \mathbb{N}) \wedge (j \leqslant i)\}) = \mathbb{N}^2$. □

Theorem 14. *No algorithm takes as input a non-negative integer n and decides whether or not*

$$\forall(x_0,\ldots,x_{n-[\sqrt{n}]^2})\in\mathbb{N}^{n-[\sqrt{n}]^2+1}\ \exists(y_0,\ldots,y_{n-[\sqrt{n}]^2})\in\{0,\ldots,|n-[\sqrt{n}]^2-[\sqrt{n}]|\}^{n-[\sqrt{n}]^2+1}$$

$$((\forall j,k\in\{0,\ldots,n-[\sqrt{n}]^2\}\ (x_j+1=x_k\Rightarrow y_j+1=y_k))\wedge$$
$$(\forall i,j,k\in\{0,\ldots,n-[\sqrt{n}]^2\}\ (x_i\cdot x_j=x_k\Rightarrow y_i\cdot y_j=y_k)))$$

Proof. It follows from Theorem 12 and Lemma 8. □

Let T_3 denote the algorithmically undecidable subset of $\mathbb{N}$ considered in Theorem 14. Similarly as in Theorem 9, the set T_3 is not recursively enumerable. Similarly as in Theorem 10, the formula that defines the set T_3 can be translated into a first-order formula of arithmetic.

6 The fourth example of a non-recursively enumerable subset of $\mathbb{N}$ which has a short definition in terms of arithmetic

Lemma 9. *([14, p. 110], cf. Lemma 7). There exists a constructive algorithm that takes as input a Diophantine equation $D(x_0,\ldots,x_l)=0$ and returns a system S of equations of the forms $y_j+1=y_k$ and $y_i\cdot y_j=y_k$ which is solvable in non-negative integers if and only if the equation $D(x_0,\ldots,x_l)=0$ is solvable in non-negative integers.*

Theorem 15. *No algorithm takes as input a non-negative integer n and decides whether or not*

$$\exists(y_0,\ldots,y_n)\in\mathbb{N}^{n+1}\ \ \forall i,j,k\in\{0,\ldots,n\} \qquad \textbf{(P)}$$

$$((2^{2^{2^j}\cdot 3^k}+1 \text{ divides } n)\Rightarrow(y_j+1=y_k))\wedge((2^{2^{2^i}\cdot 3^j\cdot 5^k+1}+1 \text{ divides } n)\Rightarrow(y_i\cdot y_j=y_k))$$

Proof. If $n>0$, then we can compute a unique $(p,q)\in\mathbb{N}^2$ such that $n=2^p\cdot(2q+1)$. The decision problem **(P)** is algorithmically undecidable because we can obtain undecidability when $n>0$ and for every $i,j,k\in\{0,\ldots,n\}$

$$(\max(j,k)>p)\Rightarrow(2^{2^{2^j}\cdot 3^k}+1 \text{ does not divide } n)$$

and

$$(\max(i,j,k)>p)\Rightarrow(2^{2^{2^i}\cdot 3^j\cdot 5^k+1}+1 \text{ does not divide } n)$$

In this case, by Lemma 4, for every system of equations

$$S\subseteq\{y_j+1=y_k,\ y_i\cdot y_j=y_k:\ i,j,k\in\{0,\ldots,p\}\}$$

the problem of solvability of S in non-negative integers $y_0,\ldots,y_p$ is equivalent to the problem **(P)** for some $n=2^p\cdot(2q+1)$, where n can be computed. Next, we apply Lemma 9 and a negative solution to Hilbert's 10th problem. □

Let T_4 denote the algorithmically undecidable subset of $\mathbb{N}$ considered in Theorem 15.

Theorem 16. *The set T_4 is recursively enumerable.*

Theorem 17. *The set $\mathbb{N}\setminus T_4$ is not recursively enumerable.*

Proof. It follows from Theorems 15 and 16. □

7 A limit-computable function $\gamma : \mathbb{N} \to \mathbb{N}$ of unknown computability which eventually dominates every function $\delta : \mathbb{N} \to \mathbb{N}$ with a single-fold Diophantine representation

The Davis-Putnam-Robinson-Matiyasevich theorem states that every recursively enumerable set $\mathcal{M} \subseteq \mathbb{N}^n$ $(n \in \mathbb{N} \setminus \{0\})$ has a Diophantine representation, that is

$$(a_1, \ldots, a_n) \in \mathcal{M} \Longleftrightarrow \exists x_1, \ldots, x_m \in \mathbb{N} \;\; W(a_1, \ldots, a_n, x_1, \ldots, x_m) = 0 \qquad \texttt{(R)}$$

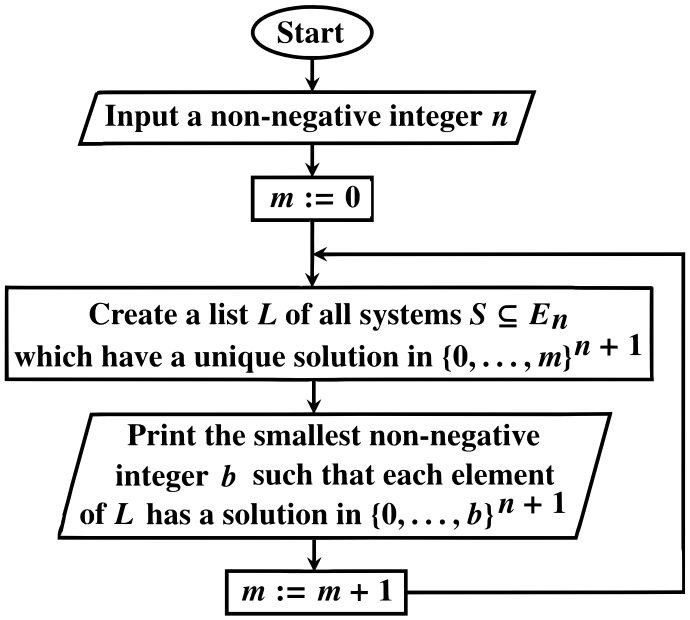
for some polynomial W with integer coefficients, see [7]. The representation (R) is said to be single-fold, if for any $a_1, \ldots, a_n \in \mathbb{N}$ the equation $W(a_1, \ldots, a_n, x_1, \ldots, x_m) = 0$ has at most one solution $(x_1, \ldots, x_m) \in \mathbb{N}^m$.

Hypothesis 1. *([2], [3], [4], [5, pp. 341–342], [8, p. 42], [9, p. 745]). Every recursively enumerable set $\mathcal{X} \subseteq \mathbb{N}^k$ $(k \in \mathbb{N} \setminus \{0\})$ has a single-fold Diophantine representation.*

For $n \in \mathbb{N}$, $\gamma(n)$ denotes the smallest $b \in \mathbb{N}$ such that if a system of equations $S \subseteq E_n$ has a unique solution in $\mathbb{N}^{n+1}$, then this solution belongs to $\{0, \ldots, b\}^{n+1}$. The computability of γ is unknown.

Theorem 18. *The function $\gamma : \mathbb{N} \to \mathbb{N}$ is computable in the limit and eventually dominates every function $\delta : \mathbb{N} \to \mathbb{N}$ with a single-fold Diophantine representation.*

Proof. This is proved in [15]. Flowchart 4 shows a semi-algorithm which computes $\gamma(n)$ in the limit, see [15].

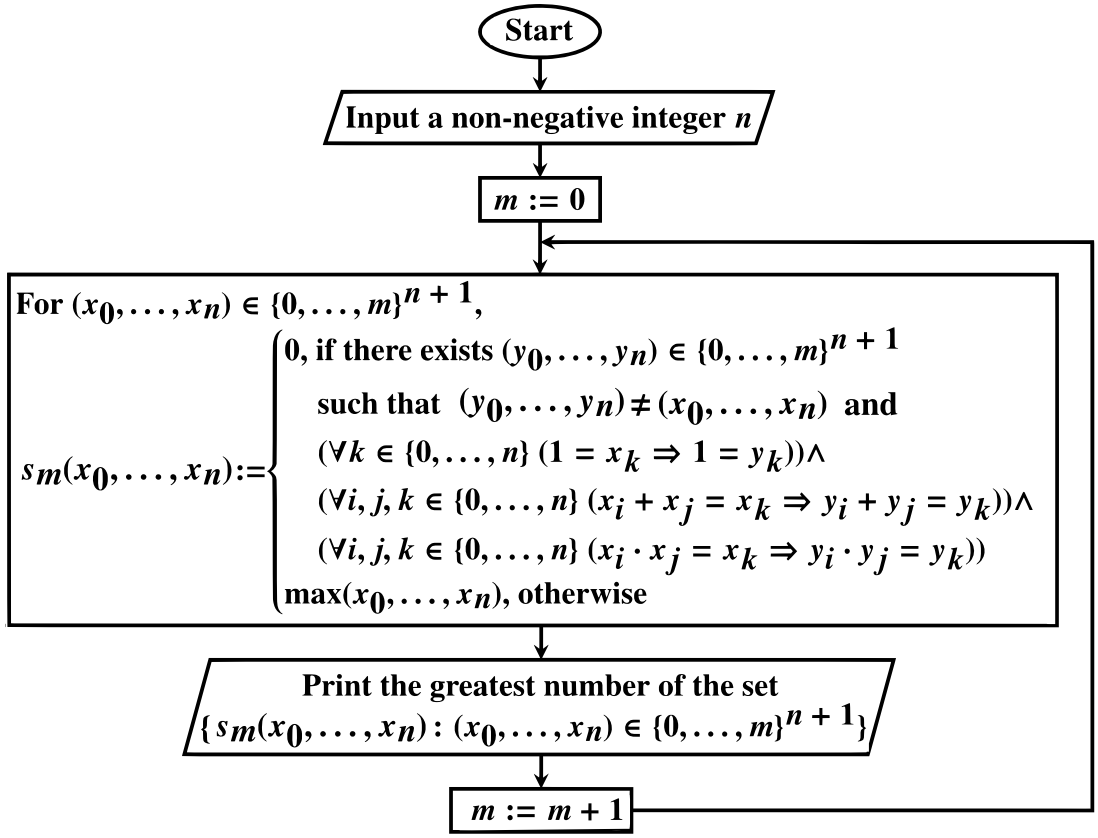


Flowchart 4

A semi-algorithm which computes $\gamma(n)$ in the limit

□

Flowchart 5 shows a simpler semi-algorithm which computes $\gamma(n)$ in the limit.



Flowchart 5

A simpler semi-algorithm which computes $\gamma(n)$ in the limit

Lemma 10. *For every $n, m \in \mathbb{N}$, the number printed by Flowchart 5 does not exceed the number printed by Flowchart 4.*

Proof. For every $(a_0, \ldots, a_n) \in \{0, \ldots, m\}^{n+1}$,

$$E_n \supseteq \{1 = x_k : \ (k \in \{0, \ldots, n\}) \wedge (1 = a_k)\} \cup$$

$$\{x_i + x_j = x_k : \ (i, j, k \in \{0, \ldots, n\}) \wedge (a_i + a_j = a_k)\} \cup$$

$$\{x_i \cdot x_j = x_k : \ (i, j, k \in \{0, \ldots, n\}) \wedge (a_i \cdot a_j = a_k)\}$$

□

Lemma 11. *For every $n, m \in \mathbb{N}$, the number printed by Flowchart 4 does not exceed the number printed by Flowchart 5.*

Proof. Let $n, m \in \mathbb{N}$. For every system of equations $S \subseteq E_n$, if $(a_0, \ldots, a_n) \in \{0, \ldots, m\}^{n+1}$ is a unique solution of S in $\{0, \ldots, m\}^{n+1}$, then $(a_0, \ldots, a_n)$ solves the system $\widehat{S}$, where

$$\widehat{S} = \{1 = x_k : \ (k \in \{0, \ldots, n\}) \wedge (1 = a_k)\} \cup$$

$$\{x_i + x_j = x_k : \ (i, j, k \in \{0, \ldots, n\}) \wedge (a_i + a_j = a_k)\} \cup$$

$$\{x_i \cdot x_j = x_k : \ (i, j, k \in \{0, \ldots, n\}) \wedge (a_i \cdot a_j = a_k)\}$$

By this and the inclusion $\widehat{S} \supseteq S$, $\widehat{S}$ has exactly one solution in $\{0, \ldots, m\}^{n+1}$, namely $(a_0, \ldots, a_n)$. □

Theorem 19. *For every $n, m \in \mathbb{N}$, Flowcharts 4 and 5 print the same number.*

Proof. It follows from Lemmas 10 and 11. □

The following program in *MuPAD* implements the semi-algorithm shown in Flowchart 5.

```
input("Input a non-negative integer n",n):
m:=0:
while TRUE do
X:=combinat::cartesianProduct([s $s=0..m] $t=0..n):
Y:=[max(op(X[u])) $u=1..(m+1)^(n+1)]:
for p from 1 to (m+1)^(n+1) do
for q from 1 to (m+1)^(n+1) do
v:=1:
for k from 1 to n+1 do
if 1=X[p][k] and 1<>X[q][k] then v:=0 end_if:
for i from 1 to n+1 do
for j from i to n+1 do
if X[p][i]+X[p][j]=X[p][k] and X[q][i]+X[q][j]<>X[q][k] then v:=0 end_if:
if X[p][i]*X[p][j]=X[p][k] and X[q][i]*X[q][j]<>X[q][k] then v:=0 end_if:
end_for:
end_for:
end_for:
if q<>p and v=1 then Y[p]:=0 end_if:
end_for:
end_for:
print(max(op(Y))):
m:=m+1:
end_while:
```

References

[1] C. S. Calude, *To halt or not to halt? That is the question,* World Scientific, Singapore, 2024.

[2] D. Cantone, A. Casagrande, F. Fabris, E. Omodeo, *The quest for Diophantine finite-foldness,* Matematiche (Catania) 76 (2021), no. 1, 133–160, *https://doi.org/10.4418/2021.76.1.8*.

[3] D. Cantone, L. Cuzziol, E. G. Omodeo, *On Diophantine singlefold specifications,* Matematiche (Catania) 79 (2024), no. 2, 585–620, *https://lematematiche.dmi.unict.it/index.php/lematematiche/article/view/2703/1218*.

[4] D. Cantone and E. G. Omodeo, *"One equation to rule them all", revisited,* Rend. Istit. Mat. Univ. Trieste 53 (2021), Art. No. 28, 32 pp. (electronic), *https://doi.org/10.13137/2464-8728/33314*.

[5] M. Davis, Yu. Matiyasevich, J. Robinson, *Hilbert's tenth problem, Diophantine equations: positive aspects of a negative solution;* in: Mathematical developments arising from Hilbert problems (ed. F. E. Browder), Proc. Sympos. Pure Math., vol. 28, Part 2, Amer. Math. Soc., Providence, RI, 1976, 323–378, *https://doi.org/10.1090/pspum/028.2*; reprinted in: The collected works of Julia Robinson (ed. S. Feferman), Amer. Math. Soc., Providence, RI, 1996, 269–324.

[6] *Gödel's β function, https://en.wikipedia.org/wiki/G%C3%B6del%27s_%CE%B2_function*.

[7] Yu. Matiyasevich, *Hilbert's tenth problem,* MIT Press, Cambridge, MA, 1993.

[8] Yu. Matiyasevich, *Hilbert's tenth problem: what was done and what is to be done,* in: Proceedings of the Workshop on Hilbert's tenth problem: relations with arithmetic and algebraic geometry (Ghent, 1999), Contemp. Math. 270, Amer. Math. Soc., Providence, RI, 2000, 1–47, *https://doi.org/10.1090/conm/270*.

[9] Yu. Matiyasevich, *Towards finite-fold Diophantine representations,* J. Math. Sci. (N. Y.) vol. 171, no. 6, 2010, 745–752, *https://doi.org/10.1007%2Fs10958-010-0179-4*.

[10] *Pairing function, https://en.wikipedia.org/wiki/Pairing_function*.

[11] J. S. Royer and J. Case, *Subrecursive Programming Systems: Complexity and Succinctness,* Birkhäuser, Boston, 1994.

[12] W. Sierpiński, *Elementary theory of numbers,* 2nd ed. (ed. A. Schinzel), PWN – Polish Scientific Publishers and North-Holland, Warsaw-Amsterdam, 1987.

[13] R. I. Soare, *Interactive computing and relativized computability,* in: B. J. Copeland, C. J. Posy, and O. Shagrir (eds.), Computability: Turing, Gödel, Church and beyond, MIT Press, Cambridge, MA, 2013, 203–260.

[14] A. Tyszka, *A hypothetical upper bound on the heights of the solutions of a Diophantine equation with a finite number of solutions*, Open Comput. Sci. 8 (2018), no. 1, 109–114, *https://doi.org/10.1515/comp-2018-0012*.

[15] A. Tyszka, *All functions $g : \mathbb{N} \to \mathbb{N}$ which have a single-fold Diophantine representation are dominated by a limit-computable function $f : \mathbb{N} \setminus \{0\} \to \mathbb{N}$ which is implemented in MuPAD and whose computability is an open problem*, in: Computation, cryptography, and network security (eds. N. J. Daras, M. Th. Rassias), Springer, Cham, 2015, 577–590, *https://doi.org/10.1007/978-3-319-18275-9_24*.