Article

Not peer-reviewed version

Three Undecidable Decision Problems About a Non-Negative Integer *n* Which Have a Short Description in Terms of Arithmetic

Apoloniusz Tyszka *

Posted Date: 9 February 2026

doi: 10.20944/preprints202508.0363.v7

Keywords: arithmetic of N; computable function; eventual domination; limit-computable function; single-fold Diophantine representation; undecidable decision problem

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 9 February 2026 doi:10.20944/preprints202508.0363.v7

Article

Three Undecidable Decision Problems About a Non-Negative Integer *n* Which Have a Short Description in Terms of Arithmetic

Apoloniusz Tyszka

Hugo Kołłątaj University, Balicka 116B, 30-149 Kraków, Poland; rttyszka@cyf-kr.edu.pl

Abstract

We prove that the following decision problems **(1)** and **(2)** are algorithmically undecidable when $n \in \mathbb{N}$. **(1)** $\exists (y_0, \ldots, y_n) \in \mathbb{N}^{n+1} \, \forall i,j,k \in \{0, \ldots, n\} \, (((2^{2^{2^j \cdot 3^k}} + 1 \text{ divides } n) \Rightarrow (y_j + 1 = y_k)) \wedge ((2^{2^{2^i \cdot 3^j \cdot 5^k + 1}} + 1 \text{ divides } n) \Rightarrow (y_i \cdot y_j = y_k)))$. **(2)** $\exists p,q \in \mathbb{N} \, ((n = 2^p \cdot 3^q) \wedge \forall (x_0, \ldots, x_p) \in \mathbb{N}^{p+1} \, \exists (y_0, \ldots, y_p) \in \{0, \ldots, q\}^{p+1} \, \forall i,j,k \in \{0, \ldots, p\} \, (((x_j + 1 = x_k) \Rightarrow (y_j + 1 = y_k)) \wedge ((x_i \cdot x_j = x_k) \Rightarrow (y_i \cdot y_j = y_k))))$. 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 $n \in \mathbb{N}$, $\beta(n)$ 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 $\beta : \mathbb{N} \to \mathbb{N}$ is computable in the limit and eventually dominates every function $\delta : \mathbb{N} \to \mathbb{N}$ with a single-fold Diophantine representation. The computability of β is unknown. We present a short program in *MuPAD* which for $n \in \mathbb{N}$ prints the sequence $\{\beta_i(n)\}_{i=0}^{\infty}$ of non-negative integers converging to $\beta(n)$.

Keywords: arithmetic of $\mathbb{N}$; computable function; eventual domination; limit-computable function; single-fold diophantine representation; undecidable decision problem

MSC: 03D20; 11U05

---

1. The Collatz Problem Leads to a Short Computer Program That Computes in the Limit a Function $\gamma : \mathbb{N} \to \{0,1\}$ of Unknown Computability

Definition 1. *(cf. [11, 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 $\lim\limits_{m \to \infty} \xi(n,m) = f(n)$.*

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 1. 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.

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. By Lemma 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:
```

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

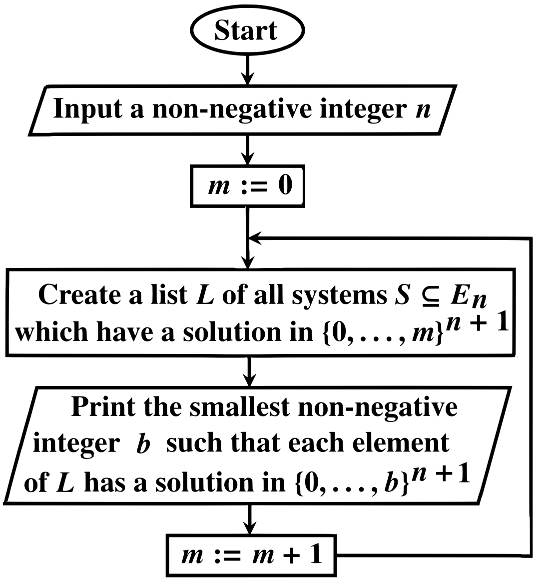
2. A Limit-Computable Function $f : \mathbb{N} \to \mathbb{N}$ Which Eventually Dominates Every Computable Function $g : \mathbb{N} \to \mathbb{N}$

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

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

Theorem 1. *([9, 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 1. 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 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 [13]. The term *"dominated"* in the title of [13] means *"eventually dominated"*. Flowchart 1 shows a semi-algorithm which computes $f(n)$ in the limit, see [13].

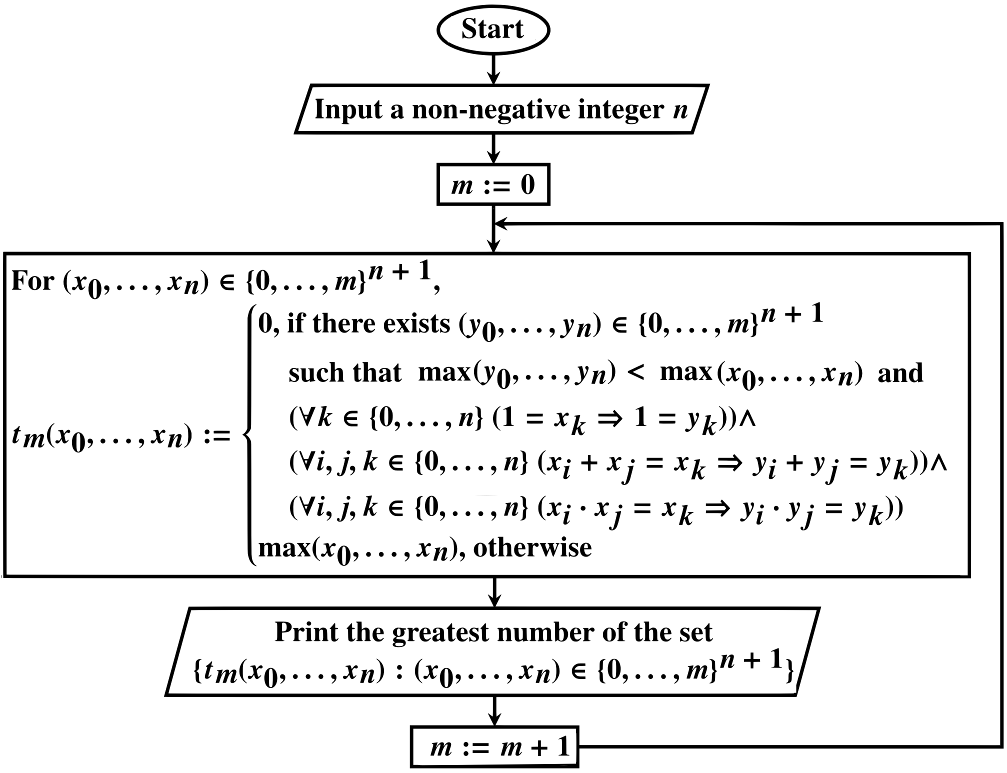


Flowchart 1

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

3. A Short Program in *MuPAD* That Computes f in the Limit

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

Proof. It follows from Lemmas 2 and 3. □

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 $\mathcal{A}(n) \subseteq \mathbb{N}^{n+1}$ such that*

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

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

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

Observation 3. *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 $\{0, \ldots, b\}^{n+1}$.*

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:
```

4. Three Undecidable Decision Problems About a Non-Negative Integer n Which Have a Short Description in Terms of Arithmetic

Theorem 3. *No algorithm takes as input non-negative integers n and m and decides whether or not*

$$\forall(x_0,\dots,x_n)\in\mathbb{N}^{n+1}\ \exists(y_0,\dots,y_n)\in\{0,\dots,m\}^{n+1}$$

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

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

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

Proof. Since the function f is not computable, it follows from Observation 2. □

Lemma 4. *([12, p. 110]). For non-negative integers, the equation $x+y=z$ is equivalent to a system which consists of equations of the forms $\beta+1=\gamma$ and $\alpha\cdot\beta=\gamma$.*

For $n\in\mathbb{N}$, $h(n)$ denotes the smallest $b\in\mathbb{N}$ such that if a system of equations $S\subseteq\{x_j+1=x_k,\ x_i\cdot x_j=x_k:\ i,j,k\in\{0,\dots,n\}\}$ has a solution in $\mathbb{N}^{n+1}$, then S has a solution in $\{0,\dots,b\}^{n+1}$. From Lemma 4 and [13], it follows that the function $h:\mathbb{N}\to\mathbb{N}$ is computable in the limit and eventually dominates every computable function $g:\mathbb{N}\to\mathbb{N}$. A bit shorter program in *MuPAD* computes h in the limit.

Theorem 4. *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. □

Theorem 5. *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 4. □

Corollary 1. *For some non-negative integer n, the formal statement in Theorem 5 is logically undecidable.*

Let $[\cdot]$ denote the integer part function.

Lemma 5. *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 6. *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 4 and Lemma 5. □

Corollary 2. *For some non-negative integer n, the formal statement in Theorem 6 is logically undecidable.*

Lemma 6. *([10, 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.*

Lemma 7. *([12, p. 110]). 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 7. *No algorithm takes as input a non-negative integer n and decides whether or not*

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

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

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 6, 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 7 and a negative solution to Hilbert's 10th problem. □

Corollary 3. *For some non-negative integer n, the formal statement in Theorem 7 is logically undecidable.*

5. A Limit-Computable Function $\beta:\mathbb{N}\to\mathbb{N}$ 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 listable set $\mathcal{M}\subseteq\mathbb{N}^n$ $(n\in\mathbb{N}\setminus\{0\})$ has a Diophantine representation, that is

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

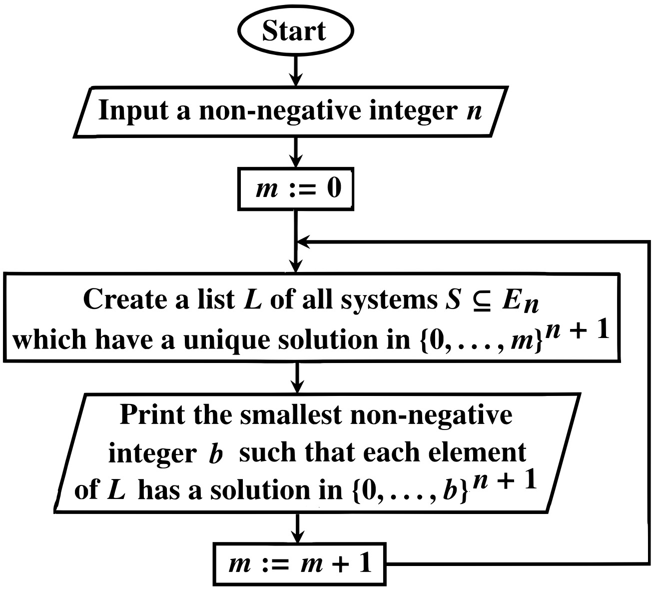
for some polynomial W with integer coefficients, see [6]. The representation (R) is said to be single-fold, if for any $a_1,\ldots,a_n\in\mathbb{N}$ the equation $W(a_1,\ldots,a_n,x_1,\ldots,x_m)=0$ has at most one solution $(x_1,\ldots,x_m)\in\mathbb{N}^m$.

Hypothesis 1. *([2], [3], [4], [5, pp. 341–342], [7, p. 42], [8, p. 745]). Every listable set $\mathcal{X}\subseteq\mathbb{N}^k$ $(k\in\mathbb{N}\setminus\{0\})$ has a single-fold Diophantine representation.*

For $n\in\mathbb{N}$, $\beta(n)$ 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 β is unknown.

Theorem 8. *The function $\beta:\mathbb{N}\to\mathbb{N}$ 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 [13]. Flowchart 3 shows a semi-algorithm which computes $\beta(n)$ in the limit, see [13].



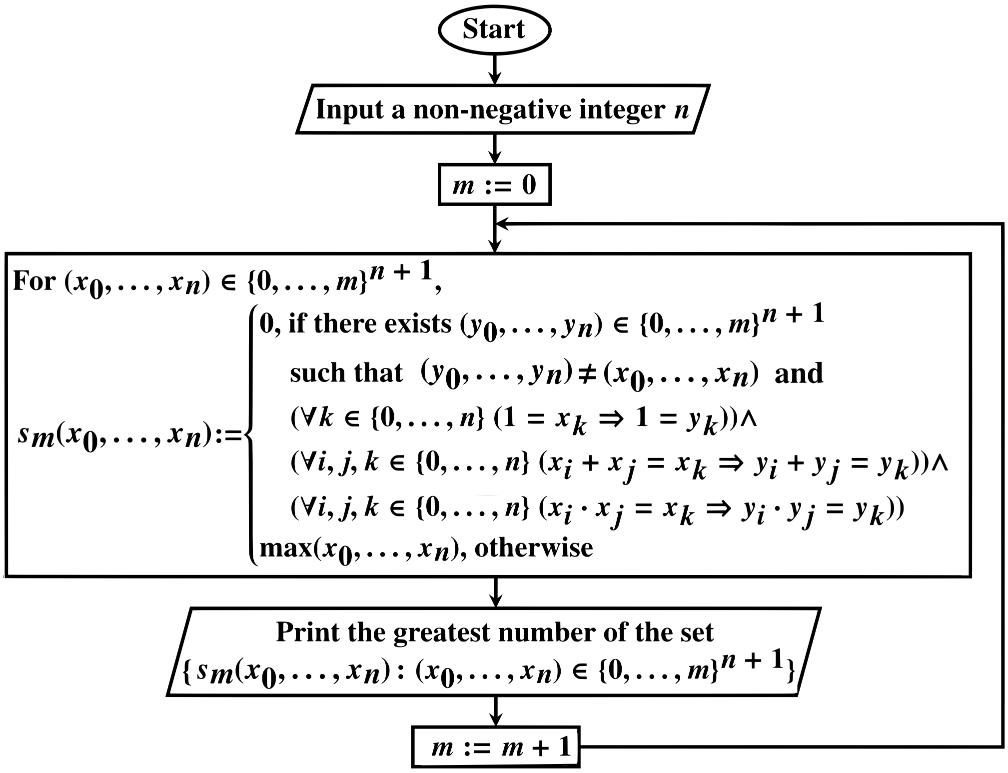
Flowchart 3

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

□

6. A Short Program in *MuPAD* That Computes β in the Limit

Flowchart 4 shows a simpler semi-algorithm which computes $\beta(n)$ in the limit.



Flowchart 4

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

Lemma 8. *For every $n, m \in \mathbb{N}$, the number printed by Flowchart 4 does not exceed the number printed by Flowchart 3.*

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

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

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

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

□

Lemma 9. *For every $n, m \in \mathbb{N}$, the number printed by Flowchart 3 does not exceed the number printed by Flowchart 4.*

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

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

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

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

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

Theorem 9. *For every $n, m \in \mathbb{N}$, Flowcharts 3 and 4 print the same number.*

Proof. It follows from Lemmas 8 and 9. □

The following program in *MuPAD* implements the semi-algorithm shown in Flowchart 4.

```
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-fold-ness,* 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. Yu. Matiyasevich, *Hilbert's tenth problem,* MIT Press, Cambridge, MA, 1993.
7. 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.
8. 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.
9. J. S. Royer and J. Case, *Subrecursive Programming Systems: Complexity and Succinctness,* Birkhäuser, Boston, 1994.
10. W. Sierpiński, *Elementary theory of numbers,* 2nd ed. (ed. A. Schinzel), PWN – Polish Scientific Publishers and North-Holland, Warsaw-Amsterdam, 1987.
11. 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.
12. 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.
13. 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.