Research Article

An Undecidable Decision Problem About a Non-Negative Integer *n* That Has a Short Description in Terms of Arithmetic

Apoloniusz Tyszka[1]

1. Hugo Kołłątaj University, Poland

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 $f(n)$. We prove that no algorithm takes as input a non-negative integer n and decides whether or not $\exists p, q \in \mathbb{N} \left((n = 2^p \cdot 3^q) \wedge \forall (x_0, \ldots, x_p) \in \mathbb{N}^{p+1} \exists (y_0, \ldots, y_p) \in \{0, \ldots, q\}^{p+1} \left((\forall j, k \in \{0, \ldots, p\} \, (x_j + 1 = x_k \Rightarrow y_j + 1 = y_k)\right) \wedge \left(\forall i, j, k \in \{0, \ldots, p\} \, (x_i \cdot x_j = x_k \Rightarrow y_i \cdot y_j = y_k)\right)\right)\right)$. 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 $\beta(n)$.

Corresponding author: Apoloniusz Tyszka, rttyszka@cyf-kr.edu.pl

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. [1]). 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* $n \in \mathbb{N}$,

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

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 [2]. The Collatz conjecture implies that $\gamma(n) = 0$ for every $n \in \mathbb{N}$.

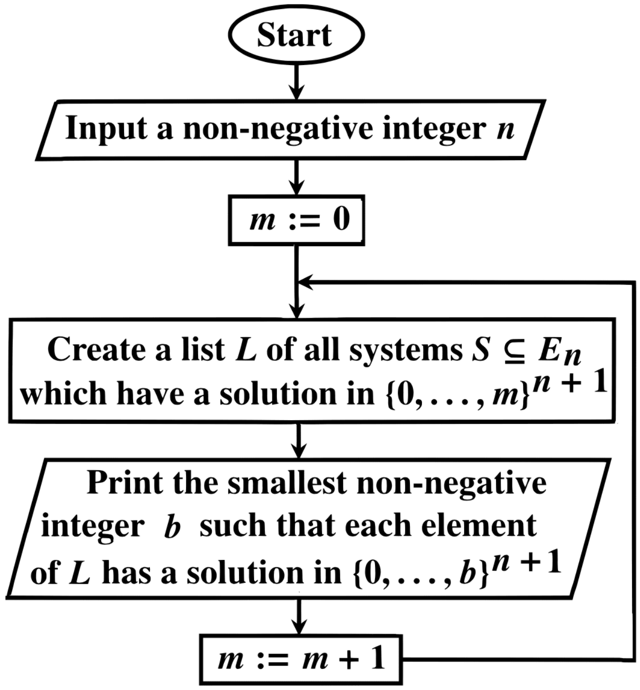
2. A limit-computable function $f : \mathbb{N} \to \mathbb{N}$ which eventually dominates every computable function $g : \mathbb{N} \to \mathbb{N}$

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. [3]. *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 [4]. The term "dominated" in the title of [4] means "eventually dominated". Flowchart 1 shows a semi-algorithm which computes $f(n)$ in the limit, see [4].

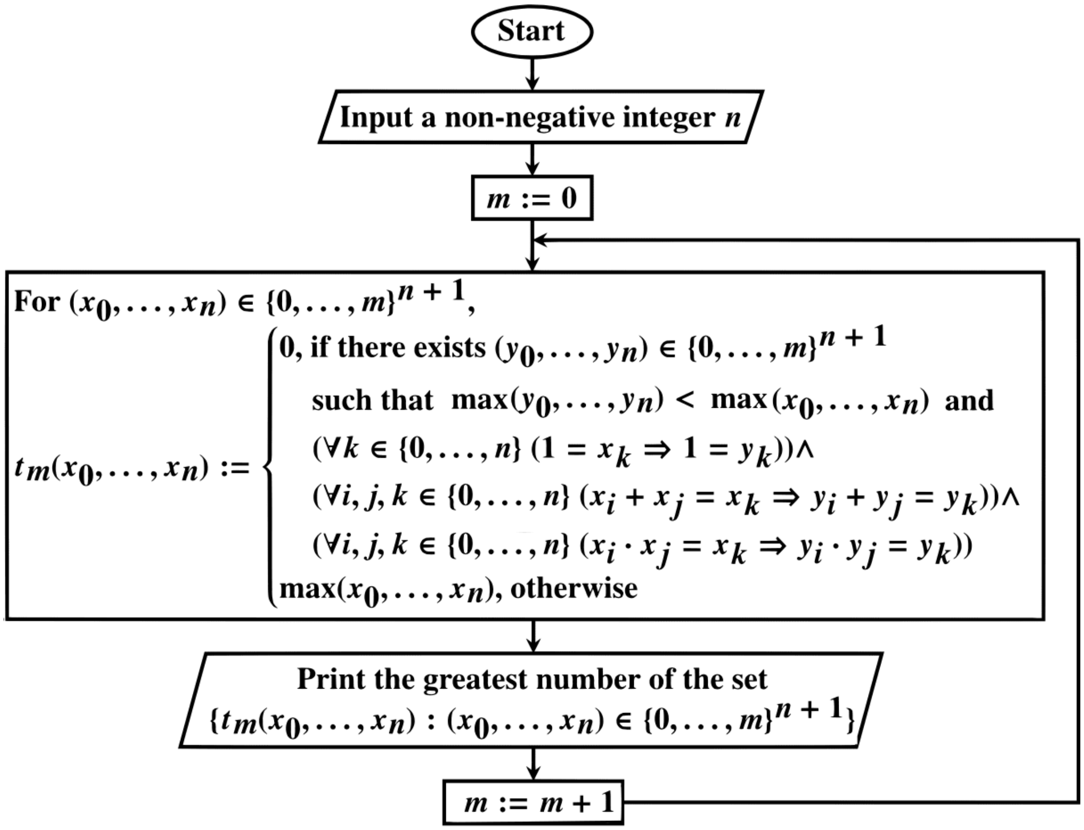


Flowchart 1. A semi-algorithm which computes $f(n)$ in the limit

3. An undecidable decision problem about an ordered pair (n, m) of non-negative integers that has a short description in terms of arithmetic

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

 doi.org/10.32388/JCNAAA.2



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:

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

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

doi.org/10.32388/JCNAAA.2

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

□

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 $(x_0,\dots,x_n) \in \mathbb{N}^{n+1}$ is a tuple $(y_0,\dots,y_n) \in \mathbb{N}^{n+1}$ such that*

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

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

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

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

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

and every tuple $(x_0,\dots,x_n) \in \mathbb{N}^{n+1}$ 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 $(x_0,\dots,x_n) \in \mathbb{N}^{n+1}$ possesses an approximation in $\{0,\dots,b\}^{n+1}$.*

Observation 3. *For every $n, m \in \mathbb{N}$, Flowcharts 1 and 2 print the smallest $b \in \{0,\dots,m\}$ such that every tuple $(x_0,\dots,x_n) \in \{0,\dots,m\}^{n+1}$ possesses an approximation in $\{0,\dots,b\}^{n+1}$.*

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. □

In Theorem 3, $n+1$ is the number of variables. Hence, the formula in Theorem 3 is not a first order formula in the language of arithmetic.

4. A short program in MuPAD that computes f in the limit

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:
```

5. An undecidable decision problem about a non-negative integer n that has a short description in terms of arithmetic

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 [4] and Lemma 3 in [5], 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*

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

doi.org/10.32388/JCNAAA.2

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

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

Proof. It holds because the function h is not computable. □

Corollary 1. *No algorithm takes as input a non-negative integer n and decides whether or not*

$$\exists p,q \in \mathbb{N}((n = 2^p \cdot 3^q)\wedge$$

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

$$((\forall j,k \in \{0,\dots,p\}(x_j + 1 = x_k \Rightarrow y_j + 1 = y_k))\wedge$$

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

The previously known examples of computably undecidable decision problems about a non-negative integer n do not have a short description in terms of arithmetic.

Example 1. *Let ϕ be a computable bijection from $\mathbb{N}$ to the set of multivariate polynomials with integer coefficients. For $n \in \mathbb{N}$, the problem whether or not the equation $\phi(n) = 0$ has a solution in non-negative integers is computably undecidable. For every known ϕ, the polynomial $\phi(n)$ does not have a short description in terms of arithmetic.*

Example 2. *Let ψ be a computable bijection from $\mathbb{N}$ to the set of sentences of Peano arithmetic. For $n \in \mathbb{N}$, the problem whether or not $\psi(n)$ is a theorem of Peano arithmetic is computably undecidable. For every known ψ, the sentence $\psi(n)$ does not have a short description in terms of arithmetic.*

6. 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

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

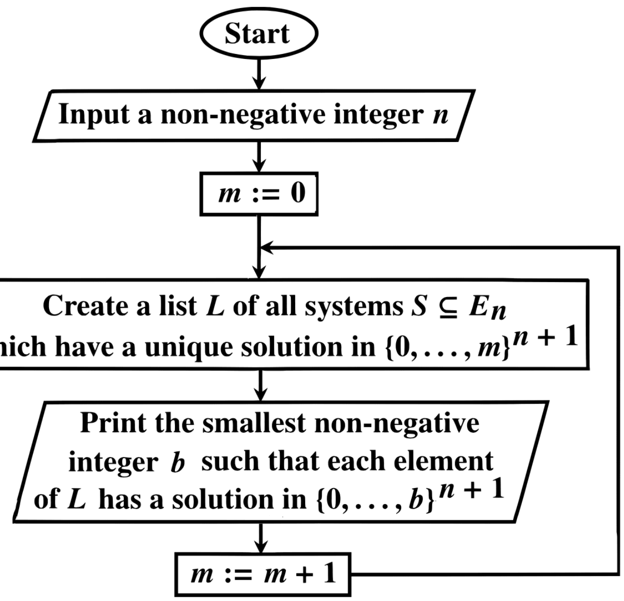
for some polynomial W with integer coefficients, see [6]. The representation (R) is said to be single-fold, if for any $a_1,\dots,a_n \in \mathbb{N}$ the equation $W(a_1,\dots,a_n,x_1,\dots,x_m) = 0$ has at most one solution $(x_1,\dots,x_m) \in \mathbb{N}^m$.

Hypothesis 1. ([7][8][9][10][11][12]). *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 5. *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 [4]. Flowchart 3 shows a semi-algorithm which computes $\beta(n)$ in the limit, see [4].

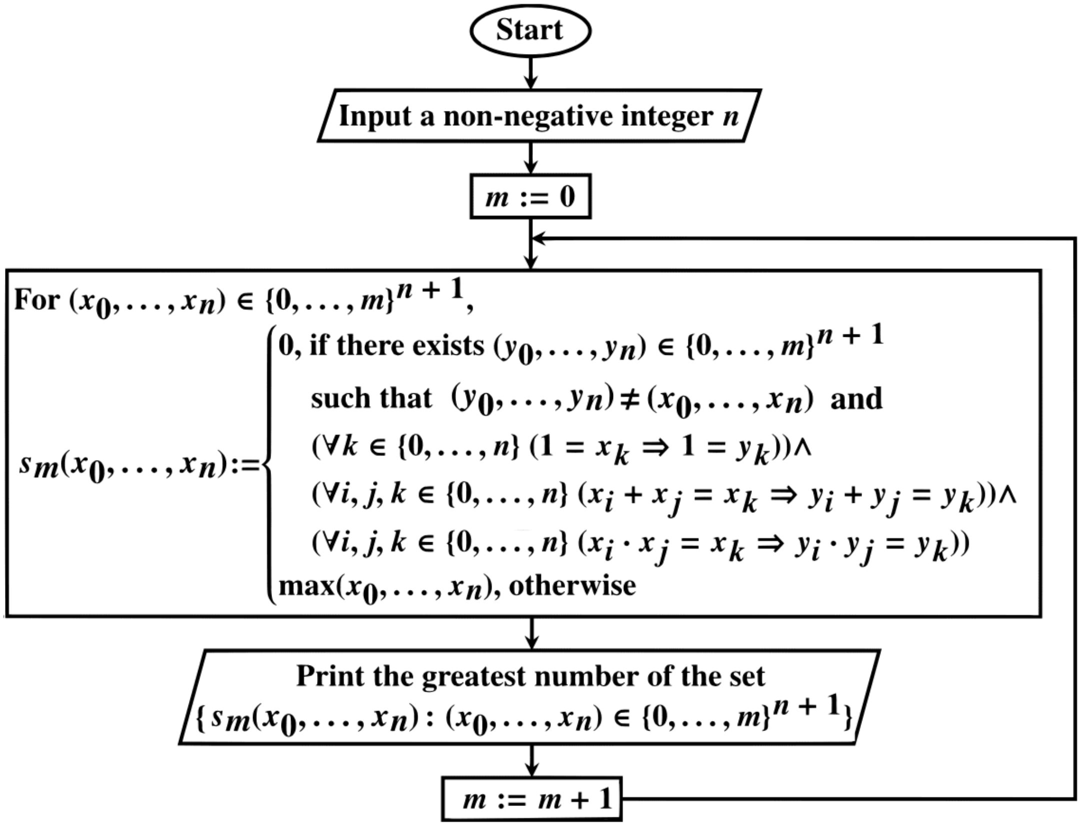


Flowchart 3. A semi-algorithm which computes $\beta(n)$ in the limit

□

7. 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 4. *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 5. *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 $\hat{S}$, where

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

doi.org/10.32388/JCNAAA.2

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

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

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

Theorem 6. *For every $n, m \in \mathbb{N}$, Flowcharts 3 and 4 print the same number.*

Proof. It follows from Lemmas 4 and 5. □

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:
```

Notes

2020 Mathematics Subject Classification: 03D20, 11U05.

References

1. ^Soare RI (2013). "Interactive Computing and Relativized Computability." in: Computability: Turing, Gödel, Church and Beyond. Copeland BJ, Posy CJ, Shagrir O (eds.). Cambridge, MA: MIT Press. 203–260.
2. ^Calude CS (2024). To Halt or Not to Halt? That Is the Question. Singapore: World Scientific.
3. ^Royer JS, Case J (1994). Subrecursive Programming Systems: Complexity and Succinctness. Boston: Birkhäuser.
4. a, b, c, d, e, fTyszka A (2015). "All Functions $g:{{\mathbb{N}}\rightarrow{\mathbb{N}}}$ Which Have a Single-Fold Diophantine Representation Are Dominated by a Limit-Computable Function $f:{{{\mathbb{N}} \smallsetminus {\{ 0\}}}\rightarrow{\mathbb{N}}}$ Which Is Implemented in MuPAD and Whose Computability Is an Open Problem." in: Computation, Cryptography, and Network Security. Daras NJ, Rassias MTh (eds.). Cham: Springer. 577–590. doi:10.1007/978-3-319-18275-9_24.
5. ^Tyszka A (2018). "A Hypothetical Upper Bound on the Heights of the Solutions of a Diophantine Equation with a Finite Number of Solutions." Open Comput Sci. **8**(1):109–114. doi:10.1515/comp-2018-0012.
6. ^Matiyasevich Yu (1993). Hilbert's Tenth Problem. Cambridge, MA: MIT Press.
7. ^Cantone D, Casagrande A, Fabris F, Omodeo E (2021). "The Quest for Diophantine Finite-Fold-Ness." Matematiche (Catania). **76**(1):133–160. doi:10.4418/2021.76.1.8.
8. ^Cantone D, Cuzziol L, Omodeo EG (2024). "On Diophantine Singlefold Specifications." Matematiche (Catania). **79**(2):585–620. https://lematematiche.dmi.unict.it/index.php/lematematiche/article/view/2703/1218.
9. ^Cantone D, Omodeo EG (2021). "“One Equation to Rule Them All”, Revisited." Rend Istit Mat Univ Trieste. **53**:Art. No. 28, 32 pp. (electronic). doi:10.13137/2464-8728/33314.
10. ^Davis M, Matiyasevich Yu, Robinson J (1976). "Hilbert's Tenth Problem, Diophantine Equations: Positive Aspects of a Negative Solution." in: Mathematical Developments Arising from Hilbert Problems. Browder FE (ed.). Providence, RI: Amer Math Soc. 323–378. doi:10.1090/pspum/028.2. Reprinted in: The Collected Works of Julia Robinson. Feferman S (ed.). Providence, RI: Amer Math Soc. 1996. 269–324.
11. ^Matiyasevich Yu (2000). "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). Providence, RI: Amer Math Soc. 1–47. doi:10.1090/conm/270.
12. ^Matiyasevich Yu (2010). "Towards Finite-Fold Diophantine Representations." J Math Sci (N Y). **171**(6):745–752. doi:10.1007/s10958-010-0179-4.

Declarations

Funding: No specific funding was received for this work.

Potential competing interests: No potential competing interests to declare.