Preprints.org

Article

Not peer-reviewed version

# A Short Program in MuPAD that Computes in the Limit a Function $f: \mathbb{N} \to \mathbb{N}$ Which Eventually Dominates Every Computable Function $g: \mathbb{N} \to \mathbb{N}$

Apoloniusz Tyszka *

Posted Date: 2 October 2025

doi: 10.20944/preprints202508.0363.v4

Keywords: computable function; eventual domination; limit-computable function; non-computable function; single-fold Diophantine representation

Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 October 2025 doi:10.20944/preprints202508.0363.v4

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

*Article*

# A Short Program in *MuPAD* That Computes in the Limit a Function $f : \mathbb{N} \to \mathbb{N}$ Which Eventually Dominates Every Computable Function $g : \mathbb{N} \to \mathbb{N}$

**Apoloniusz Tyszka**

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

**Abstract**

It is known that there exists a limit-computable function $f : \mathbb{N} \to \mathbb{N}$ which is not computable. Every known proof of this fact does not lead to the existence of a short computer program that computes $f$ in the limit. 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 $\mathcal{S} \subseteq E_n$ has a solution in $\mathbb{N}^{n+1}$, then $\mathcal{S}$ 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 $\mathcal{S} \subseteq E_n$ 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 $\beta$ 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:** computable function; eventual domination; limit-computable function; non-computable function; single-fold Diophantine representation

**MSC:** 03D20

---

## 1. Introduction

**Definition 1.** *(cf. [9][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 $\lim_{m \to \infty} \xi(n, m) = f(n)$.*

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. In particular, this observation applies to the proof of Theorem 1 in [8], see also Observation 1.

**Observation 1.** *Let $\varphi$ be a computable bijection from $\mathbb{N}$ to the set of all Diophantine equations. For $n \in \mathbb{N}$, let*

$$\theta(n) = \begin{cases} 1, & \text{if } \varphi(n) \text{ is solvable in non-negative integers} \\ 0, & \text{otherwise} \end{cases}$$

*The function $\theta : \mathbb{N} \to \mathbb{N}$ is computable in the limit. A negative solution to Hilbert's 10th problem implies that the function $\theta$ is not computable. There is no known $\varphi$ for which there exists a short computer program that computes $\theta$ in the limit.*

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 October 2025 doi:10.20944/preprints202508.0363.v4



*MuPAD* is a part of the Symbolic Math Toolbox in MATLAB R2019b. In this article, we present a short program in *MuPAD* that computes in the limit a function $f : \mathbb{N} \to \mathbb{N}$ which eventually dominates every computable function $g : \mathbb{N} \to \mathbb{N}$.

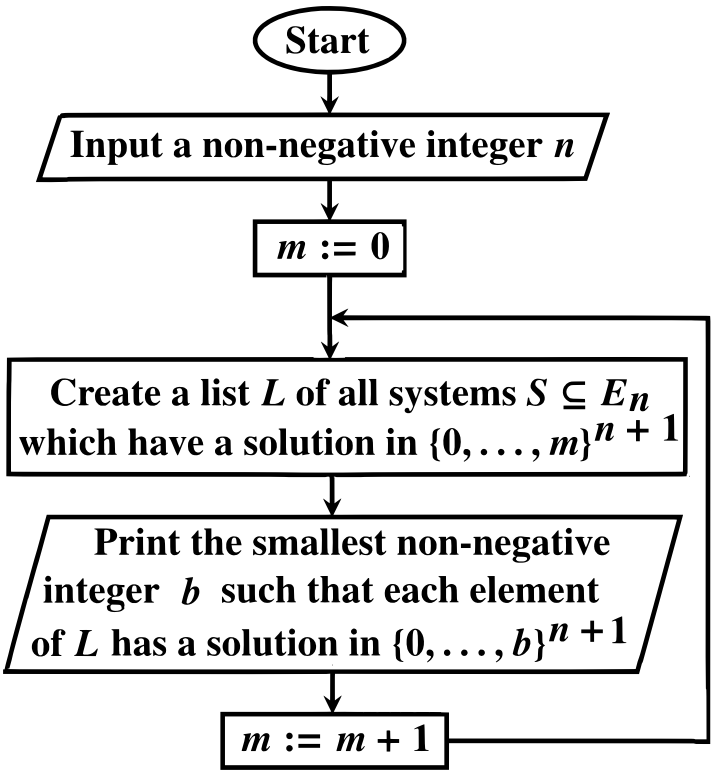
## 2. A Limit-Computable Function $f : \mathbb{N} \to \mathbb{N}$ Which Eventually Dominates Every Computable Function $g : \mathbb{N} \to \mathbb{N}$

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.** *([8, 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 $\mathcal{S} \subseteq E_n$ has a solution in $\mathbb{N}^{n+1}$, then $\mathcal{S}$ 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 [11]. The term *"dominated"* in the title of [11] means *"eventually dominated"*. Flowchart 1 shows a semi-algorithm which computes $f(n)$ in the limit, see [11].




**Flowchart 1**

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

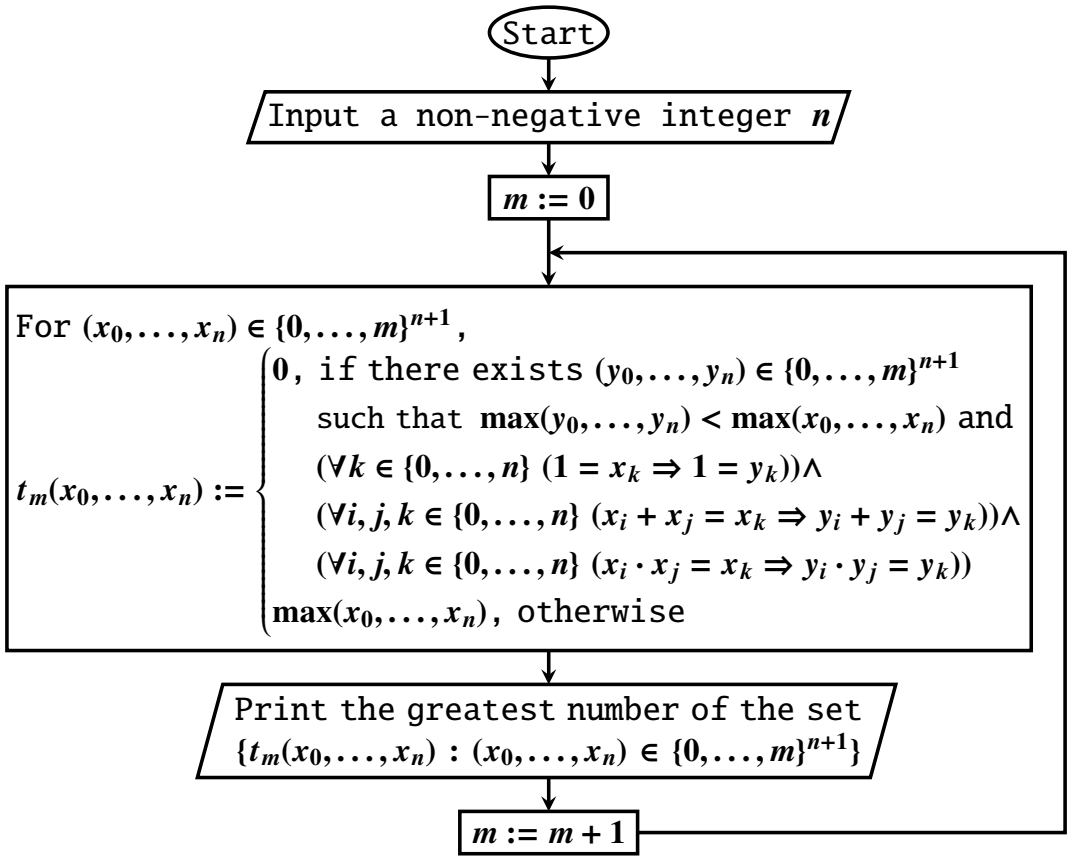
© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 October 2025 doi:10.20944/preprints202508.0363.v4



## 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 1.** *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 $(a_0,\dots,a_n) \in \{0,\dots,m\}^{n+1}$,

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

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

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

□

**Lemma 2.** *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 $\mathcal{S} \subseteq E_n$, if $(a_0,\dots,a_n) \in \{0,\dots,m\}^{n+1}$ and $(a_0,\dots,a_n)$ solves $\mathcal{S}$, then $(a_0,\dots,a_n)$ solves the following system of equations:

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

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

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

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 October 2025 doi:10.20944/preprints202508.0363.v4



□

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

**Proof.** It follows from Lemmas 1 and 2. □

**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 2.** *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 3.** *$f(n)$ 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 4.** *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:
```

For $n \in \mathbb{N}$, $h(n)$ denotes the smallest $b \in \mathbb{N}$ such that if a system of equations $\mathcal{S} \subseteq \{x_i + 1 = x_k,\ x_i \cdot x_j = x_k :\ i, j, k \in \{0, \ldots, n\}\}$ has a solution in $\mathbb{N}^{n+1}$, then $\mathcal{S}$ has a solution in $\{0, \ldots, b\}^{n+1}$.

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 October 2025 doi:10.20944/preprints202508.0363.v4



From [11] and Lemma 3 in [10], 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.

## 4. 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

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

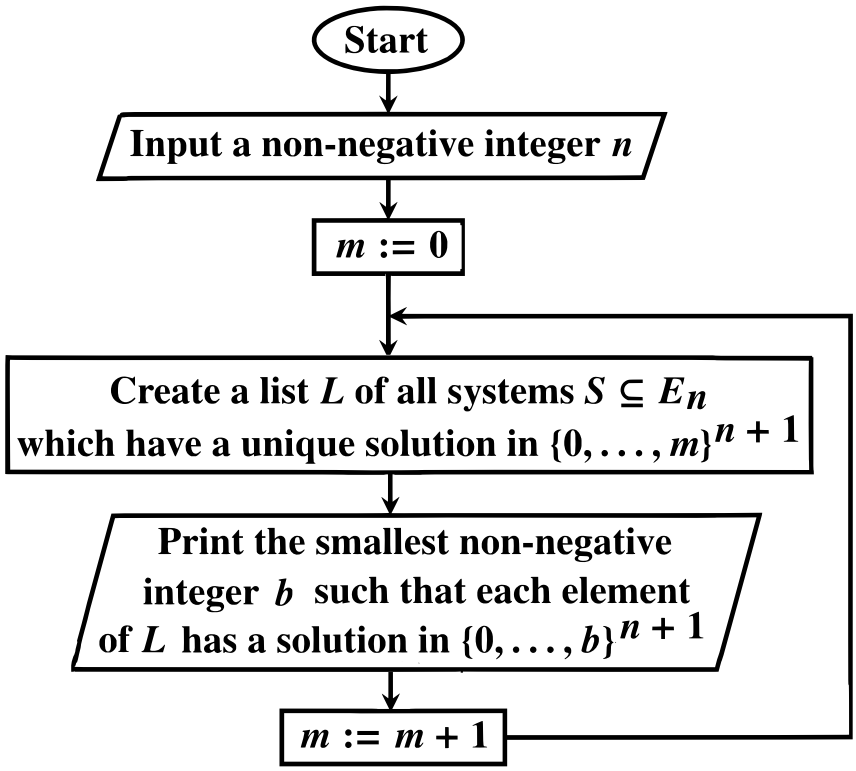
for some polynomial $W$ with integer coefficients, see [5]. The representation (R) is said to be single-fold, if for any $a_1, \ldots, a_n \in \mathbb{N}$ the equation $W(a_1, \ldots, a_n, x_1, \ldots, x_m) = 0$ has at most one solution $(x_1, \ldots, x_m) \in \mathbb{N}^m$.

**Hypothesis 1.** *([1–4][pp. 341–342], [6][p. 42], [7][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 $\mathcal{S} \subseteq E_n$ has a unique solution in $\mathbb{N}^{n+1}$, then this solution belongs to $\{0, \ldots, b\}^{n+1}$. The computability of $\beta$ is unknown.

**Theorem 3.** *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 [11]. Flowchart 3 shows a semi-algorithm which computes $\beta(n)$ in the limit, see [11].




**Flowchart 3**

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

□

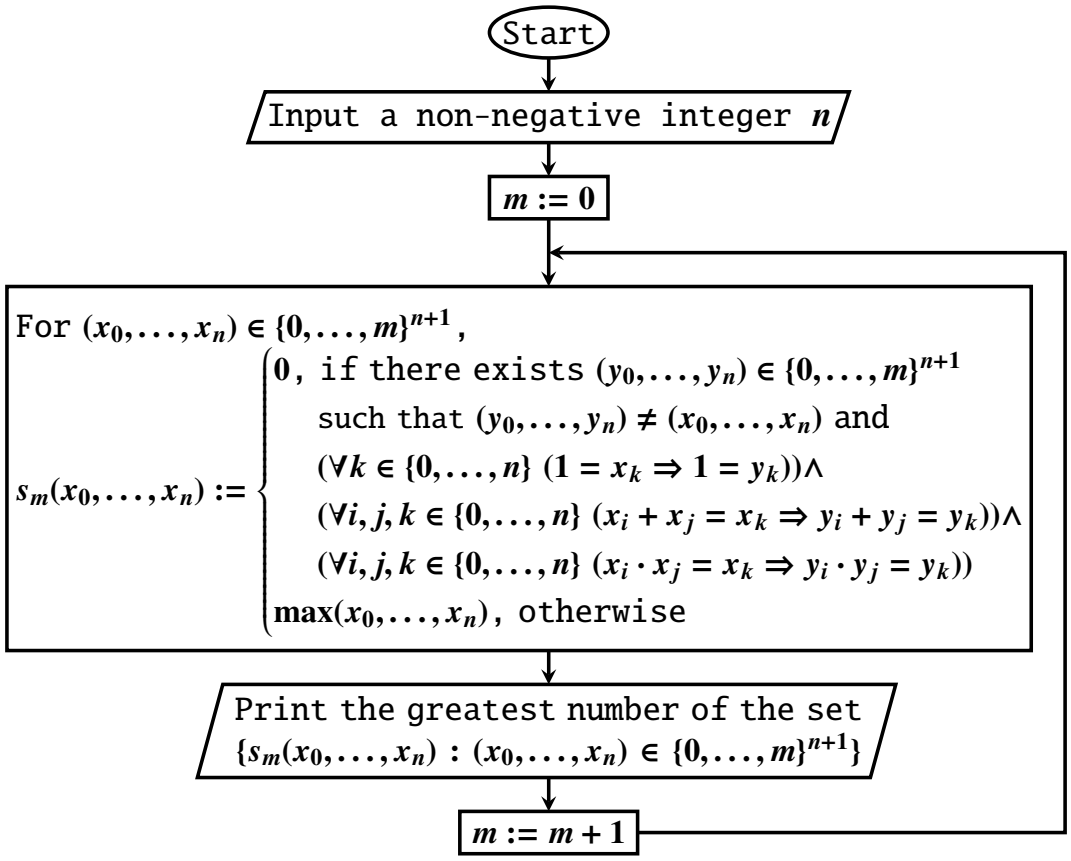
© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 October 2025 doi:10.20944/preprints202508.0363.v4



## 5. A Short Program in *MuPAD* That Computes $\beta$ 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 3.** *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 4.** *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 $\mathcal{S} \subseteq E_n$, if $(a_0,\dots,a_n) \in \{0,\dots,m\}^{n+1}$ is a unique solution of $\mathcal{S}$ in $\{0,\dots,m\}^{n+1}$, then $(a_0,\dots,a_n)$ solves the system $\widehat{\mathcal{S}}$, where

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

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

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

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 October 2025 doi:10.20944/preprints202508.0363.v4



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

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

**Proof.** It follows from Lemmas 3 and 4. □

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. 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.
2. 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.
3. 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.
4. 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.
5. Yu. Matiyasevich, *Hilbert's tenth problem,* MIT Press, Cambridge, MA, 1993.
6. 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.
7. 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.
8. J. S. Royer and J. Case, *Subrecursive Programming Systems: Complexity and Succinctness,* Birkhäuser, Boston, 1994.


© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 October 2025 doi:10.20944/preprints202508.0363.v4



9. 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.
10. 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.
11. 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.

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.