Article

Not peer-reviewed version

A Short Computer Program That Computes in the Limit a Non-Computable Function from N to N

Apoloniusz Tyszka *

Posted Date: 15 August 2025

doi: 10.20944/preprints202508.0363.v2

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

Article

A Short Computer Program That Computes in the Limit a Non-Computable Function from $\mathbb{N}$ to $\mathbb{N}$

Apoloniusz Tyszka

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

Abstract

For $n \in \mathbb{N}$, $f(n)$ denotes the smallest $b \in \mathbb{N}$ such that if a system of equations

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

has a solution in $\mathbb{N}^{n+1}$, then $\mathcal{S}$ has a solution in $\{0, \dots, b\}^{n+1}$. 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 $f(n)$. The previously known computer programs by other authors do not compute in the limit non-computable functions from $\mathbb{N}$ to $\mathbb{N}$. We present a short program in *MuPAD* that computes in the limit a 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.

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

MSC: 03D20

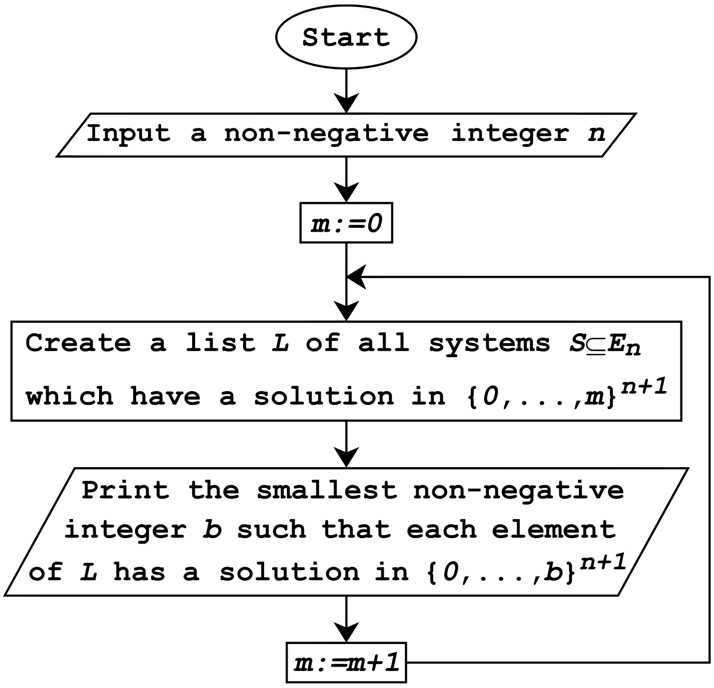
For $n \in \mathbb{N}$, let

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

1. A program that computes in the limit a function $f: \mathbb{N} \to \mathbb{N}$ which eventually dominates every computable function $g: \mathbb{N} \to \mathbb{N}$

Theorem 1. *([5, p. 118]). There exists a limit-computable function $f: \mathbb{N} \to \mathbb{N}$ which eventually dominates every computable function $g: \mathbb{N} \to \mathbb{N}$.*

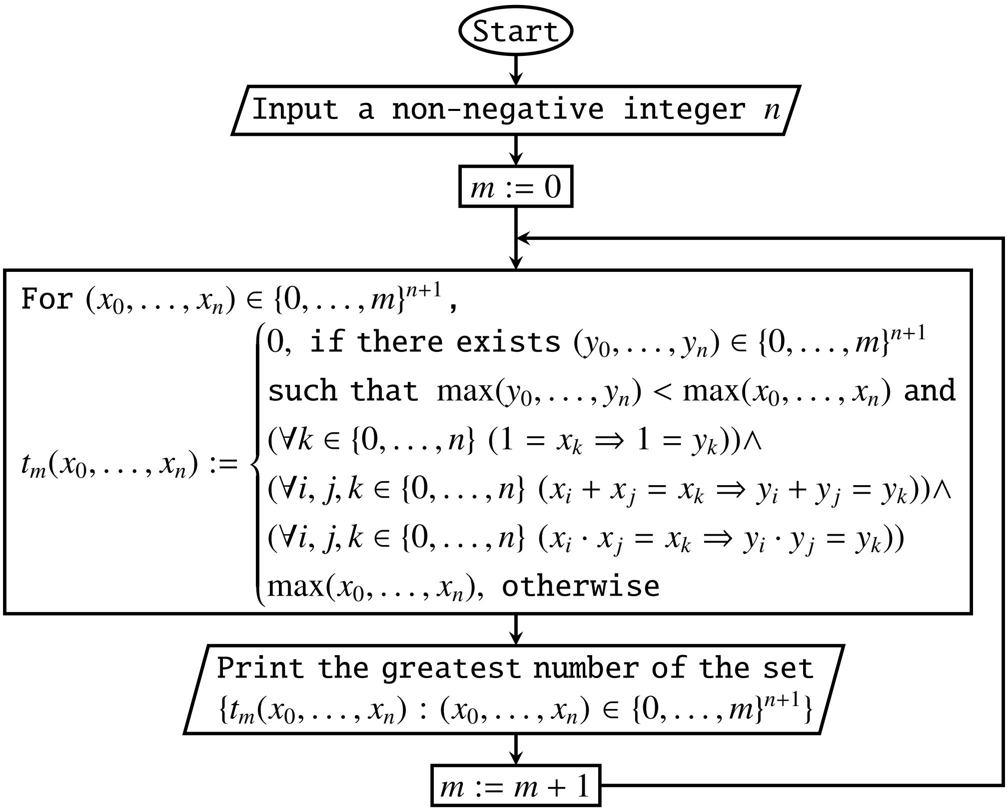
We present an alternative proof of Theorem 1. For $n \in \mathbb{N}$, $f(n)$ denotes the smallest $b \in \mathbb{N}$ such that if a system of equations $\mathcal{S} \subseteq E_n$ has a solution in $\mathbb{N}^{n+1}$, then $\mathcal{S}$ has a solution in $\{0, \dots, b\}^{n+1}$. The function $f: \mathbb{N} \to \mathbb{N}$ is computable in the limit and eventually dominates every computable function $g: \mathbb{N} \to \mathbb{N}$, see [7]. The term *"dominated"* in the title of [7] means *"eventually dominated"*. Flowchart 1 shows a semi-algorithm which computes $f(n)$ in the limit, see [7].



Flowchart 1

A semi-algorithm which computes $f(n)$ 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

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 August 2025 doi:10.20944/preprints202508.0363.v2

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

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

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

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

□

Lemma 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 $(a_0, \ldots, a_n) \in \{0, \ldots, m\}^{n+1}$ and $(a_0, \ldots, a_n)$ solves $\mathcal{S}$, then $(a_0, \ldots, a_n)$ solves the system of equations

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

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

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

□

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. □

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

doi:10.20944/preprints202508.0363.v2


```
m:=m+1:
end_while:
```

The author is not aware about computer programs by other authors which compute in the limit non-computable functions from $\mathbb{N}$ to $\mathbb{N}$.

For $n \in \mathbb{N}$, $h(n)$ denotes the smallest $b \in \mathbb{N}$ such that if a system of equations $\mathcal{S} \subseteq \{x_i + 1 = x_k,\ x_i \cdot x_j = x_k :\ i, j, k \in \{0, \ldots$ has a solution in $\mathbb{N}^{n+1}$, then $\mathcal{S}$ has a solution in $\{0, \ldots, b\}^{n+1}$. From [7] and Lemma 3 in [6], 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.

2. A program that computes in the limit a 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

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

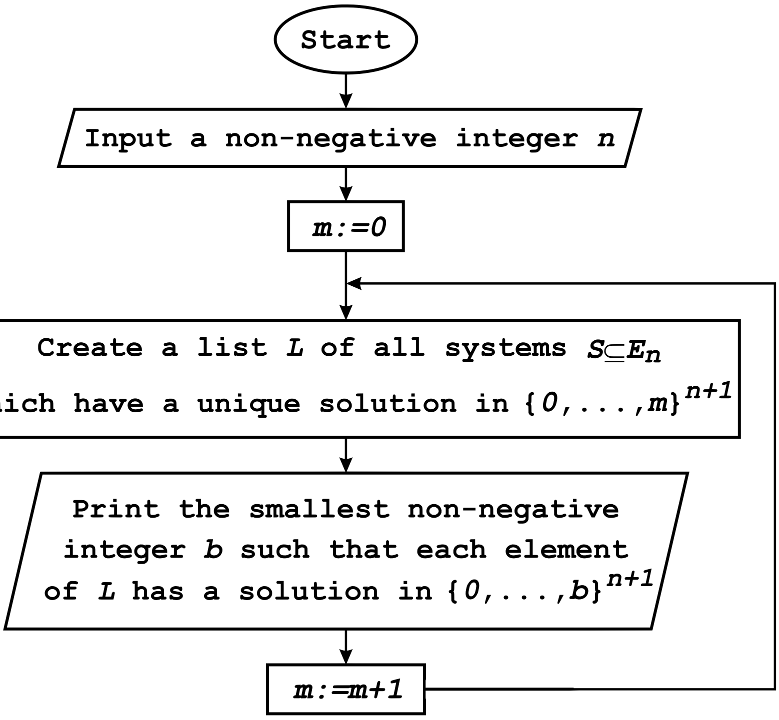
for some polynomial W with integer coefficients, see [2]. The representation (R) is said to be single-fold, if for any $a_1, \ldots, a_n \in \mathbb{N}$ the equation $W(a_1, \ldots, a_n, x_1, \ldots, x_m) = 0$ has at most one solution $(x_1, \ldots, x_m) \in \mathbb{N}^m$.

Hypothesis 1. *([1, pp. 341–342], [3, p. 42], [4, 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 β 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 [7]. Flowchart 3 shows a semi-algorithm which computes $\beta(n)$ in the limit, see [7].

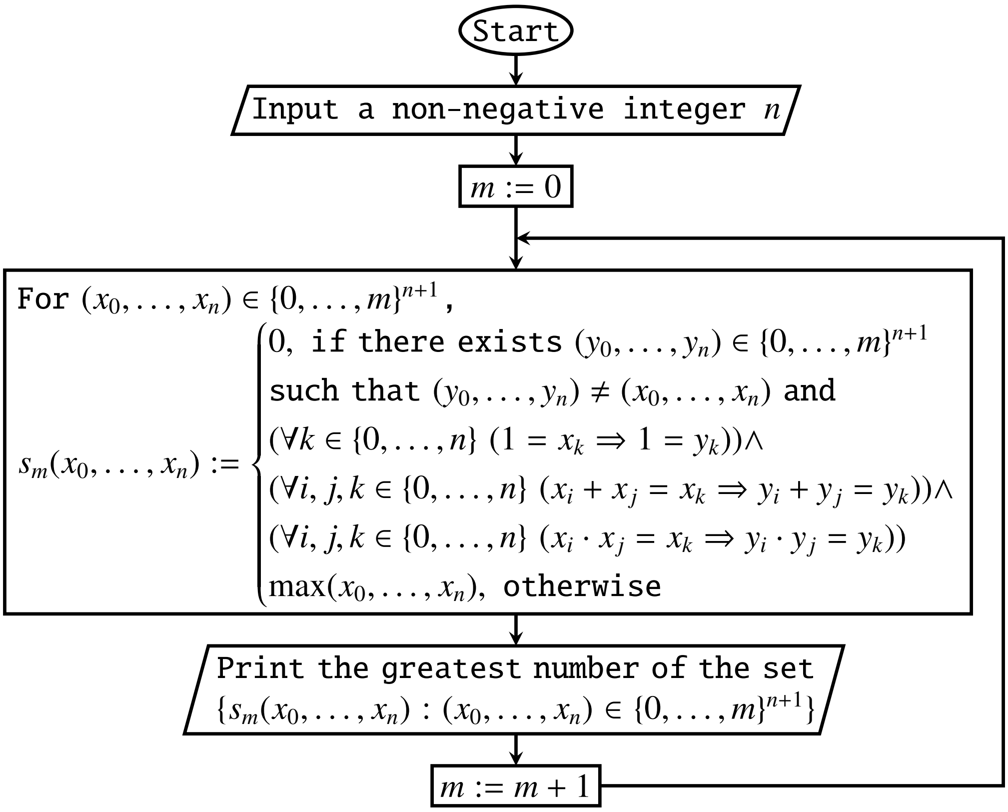


Flowchart 3

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

 doi:10.20944/preprints202508.0363.v2

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

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

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

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

□

Lemma 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 $(a_0, \ldots, a_n) \in \{0, \ldots, m\}^{n+1}$ is a unique solution of $\mathcal{S}$ in $\{0, \ldots, m\}^{n+1}$, then $(a_0, \ldots, a_n)$ solves the system of equations

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

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

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

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

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:
```

doi:10.20944/preprints202508.0363.v2


```
end_while:
```

References

1. 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.
2. Yu. Matiyasevich, *Hilbert's tenth problem,* MIT Press, Cambridge, MA, 1993.
3. 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.
4. 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.
5. J. S. Royer and J. Case, *Subrecursive Programming Systems: Complexity and Succinctness,* Birkhäuser, Boston, 1994.
6. 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://dx.doi.org/10.1515/comp-2018-0012.
7. 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.