A falsifiable statement Γ of the form "$\exists f : \mathbb{N} \to \mathbb{N}$ of unknown computability such that ..." which strengthens a mathematical theorem

Apoloniusz Tyszka

Abstract

We define a function $\gamma : \mathbb{N} \to \mathbb{N}$ which eventually dominates every computable function $\alpha : \mathbb{N} \to \mathbb{N}$. We show that there is a simple computer program which for $n \in \mathbb{N}$ prints the sequence $\{\gamma_i(n)\}_{i=0}^{\infty}$ of non-negative integers converging to $\gamma(n)$. We define a function $f : \mathbb{N} \to \mathbb{N}$ of unknown computability which eventually dominates every function $\delta : \mathbb{N} \to \mathbb{N}$ with a single-fold Diophantine representation. We show that there is a simple computer program which for $n \in \mathbb{N}$ prints the sequence $\{f_i(n)\}_{i=0}^{\infty}$ of non-negative integers converging to $f(n)$. Let Γ denote the following statement: $\exists f : \mathbb{N} \to \mathbb{N}$ *of unknown computability such that* f *eventually dominates every function* $\delta : \mathbb{N} \to \mathbb{N}$ *with a single-fold Diophantine representation and there is a simple computer program which for* $n \in \mathbb{N}$ *prints the sequence* $\{f_i(n)\}_{i=0}^{\infty}$ *of non-negative integers converging to* $f(n)$. The statement Γ has all properties from the title of the article.

Key words and phrases: eventual domination, finite-fold Diophantine representation, limit-computable function, predicate $\mathcal{K}$ of the current mathematical knowledge, single-fold Diophantine representation, time-dependent truth in mathematics with the predicate $\mathcal{K}$ of the current mathematical knowledge.

2020 Mathematics Subject Classification: 03D20.

1 Predicate $\mathcal{K}$ of the current mathematical knowledge

$\mathcal{K}$ denotes both the predicate satisfied by every currently known theorem and the set of all currently known theorems. Any theorem of any mathematician from past or present belongs to $\mathcal{K}$. The set $\mathcal{K}$ is time-dependent.

Observation 1. *$\mathcal{K}$ contains all written down theorems and their particular cases. For example,*

$$\{0+1=1,\ 1+1=2,\ 2+1=3,\ \ldots\} \subseteq \mathcal{K}$$

Observation 2. *$\mathcal{K}$ contains every particular case of any written down schema of theorems. For example, every axiom of ZFC belongs to $\mathcal{K}$.*

Proposition 1. *If $\mathcal{T}$ denotes the set of twin primes, then the statement*

$$(\neg\mathcal{K}(\mathrm{card}(\mathcal{T})=\omega)) \land (\neg\mathcal{K}(\mathrm{card}(\mathcal{T})<\omega))$$

is true, falsifiable, and expresses what is currently unproved in mathematics.

Statement 1. *There exists a non-zero integer n such that*

$$(\neg\mathcal{K}(n < 0)) \wedge (\neg\mathcal{K}(n > 0)) \tag{1}$$

Proof. It holds for

$$n = \begin{cases} -1, & \text{if } Continuum\ Hypothesis \text{ holds} \\ 1, & \text{otherwise} \end{cases}$$

□

Proposition 2. *Statement 1 holds forever.*

Proof. Since $Continuum\ Hypothesis$ is independent from ZFC, conjunction (1) holds forever for the above n. □

Proposition 3. *Statement 1 does not express what is currently unproved in mathematics.*

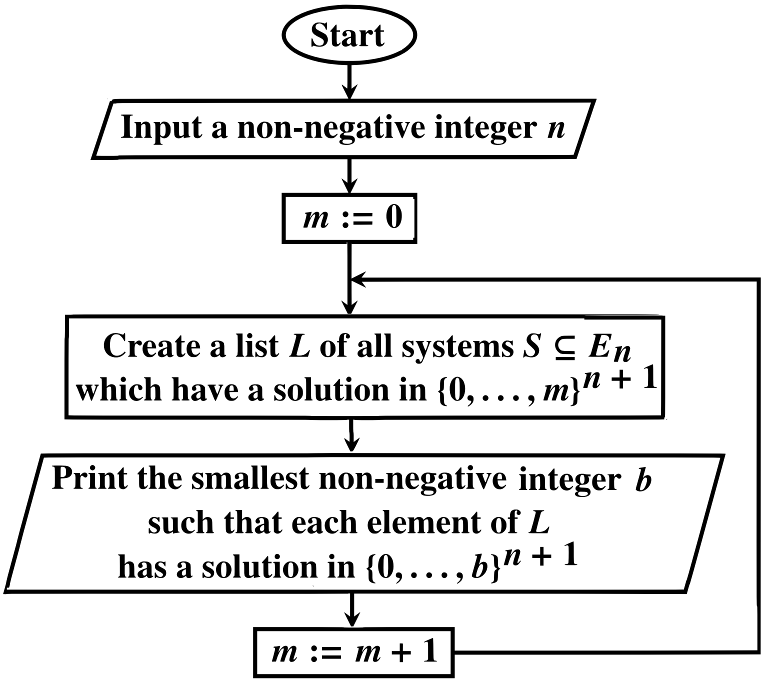
2 Limit-computable functions

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. *([4, p. 118]). There exists a limit-computable function $\gamma : \mathbb{N} \to \mathbb{N}$ which eventually dominates every computable function $\alpha : \mathbb{N} \to \mathbb{N}$.*

We present an alternative proof of Theorem 1. For every $n \in \mathbb{N}$, we define $\gamma(n)$ as 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 this solution belongs to $\{0, \ldots, b\}^{n+1}$. The function $\gamma : \mathbb{N} \to \mathbb{N}$ is computable in the limit and eventually dominates every computable function $\alpha : \mathbb{N} \to \mathbb{N}$, see [5]. Flowchart 1 shows a semi-algorithm which computes $\gamma(n)$ in the limit, see [5].



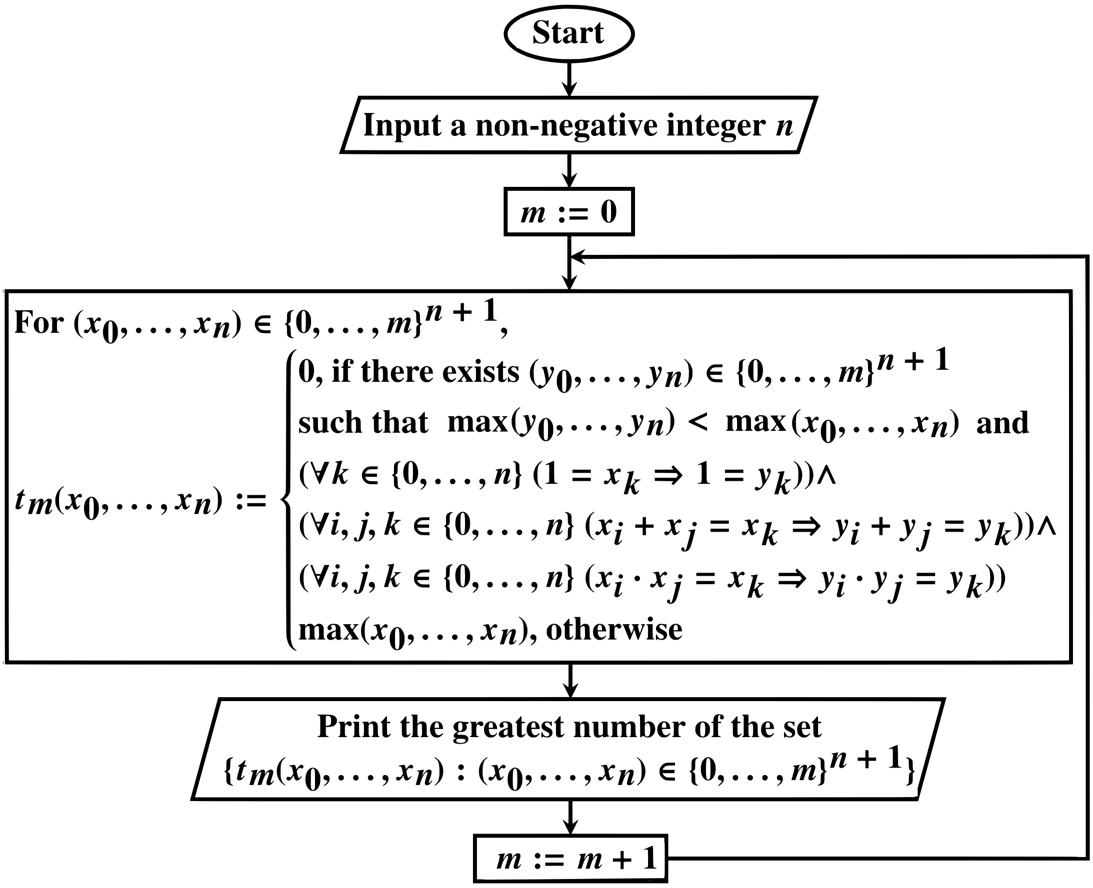
Flowchart 1

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

Proposition 4. *If $k \in \mathbb{N}$, then the statement "the function $\mathbb{N} \ni n \to k + \gamma(n) \in \mathbb{N}$ is uncomputable" belongs to $\mathcal{K}$.*

Proof. It follows from Observation 1. □

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



Flowchart 2
A simpler semi-algorithm which computes $\gamma(n)$ in the limit

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..nops(X)]:
for p from 1 to nops(X) do
for q from 1 to nops(X) 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 1 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:
```

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

Let Φ denote the following statement: *the function* $\mathbb{N} \ni n \to 2^n \in \mathbb{N}$ *eventually dominates every function* $\delta : \mathbb{N} \to \mathbb{N}$ *with a single-fold Diophantine representation*. For $n \in \mathbb{N}$, let

$$g(n) = \begin{cases} 2^n, & \text{if } \Phi \text{ holds} \\ \gamma(n), & \text{otherwise} \end{cases}$$

The function $g : \mathbb{N} \to \mathbb{N}$ is computable if and only if Φ holds. Currently,

$$(\neg\mathcal{K}(\Phi)) \wedge (\neg\mathcal{K}(\neg\Phi)) \wedge (\neg\mathcal{K}(g\ is\ computable)) \wedge (\neg\mathcal{K}(g\ is\ uncomputable))$$

Let Ψ denote the following statement: *the function* $\mathbb{N} \ni n \to 2^n \in \mathbb{N}$ *eventually dominates every function* $\delta : \mathbb{N} \to \mathbb{N}$ *with a finite-fold Diophantine representation*. For $n \in \mathbb{N}$, let

$$h(n) = \begin{cases} 2^n, & \text{if } \Psi \text{ holds} \\ \gamma(n), & \text{otherwise} \end{cases}$$

The function $h : \mathbb{N} \to \mathbb{N}$ is computable if and only if Ψ holds. Currently,

$$(\neg\mathcal{K}(\Psi)) \wedge (\neg\mathcal{K}(\neg\Psi)) \wedge (\neg\mathcal{K}(h\ is\ computable)) \wedge (\neg\mathcal{K}(h\ is\ uncomputable))$$

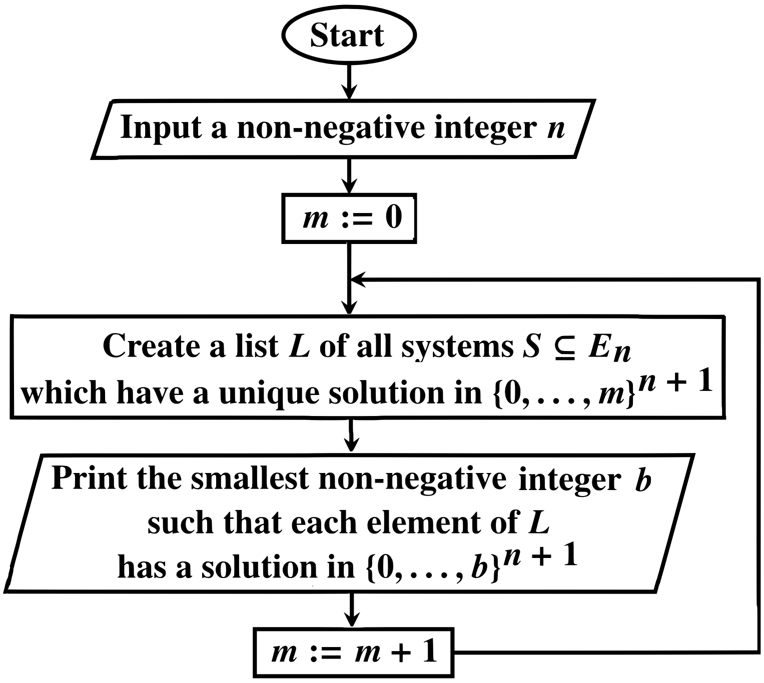
Lemma 1. *The function* g *is computable in the limit and eventually dominates every function* $\delta : \mathbb{N} \to \mathbb{N}$ *with a single-fold Diophantine representation. The function* h *is computable in the limit and eventually dominates every function* $\delta : \mathbb{N} \to \mathbb{N}$ *with a finite-fold Diophantine representation.*

Proof. It follows from Theorem 1. □

For every $n \in \mathbb{N}$, we define $\beta(n)$ as 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 $\{0, \ldots, b\}^{n+1}$.

Theorem 2. *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.*

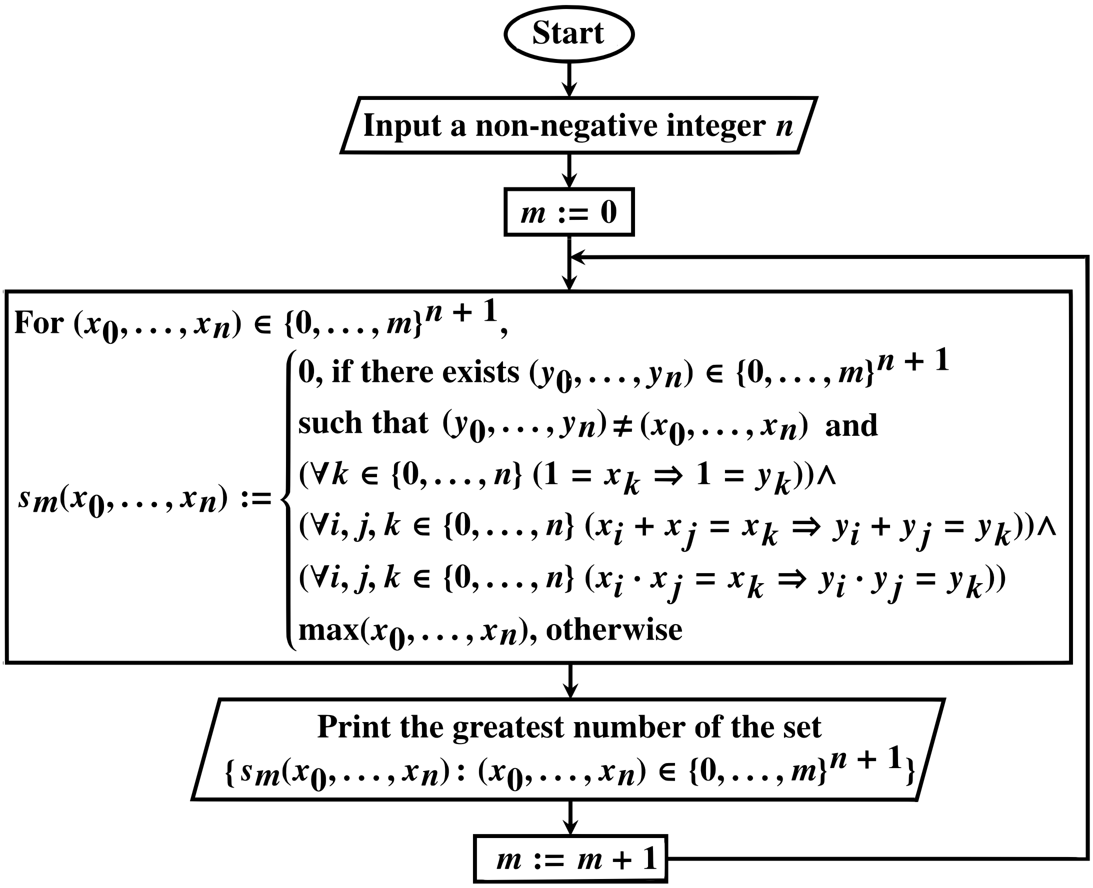
Proof. This is proved in [5]. The term *"dominated"* in the title of [5] means *"eventually dominated"*. Flowchart 3 shows a semi-algorithm which computes $\beta(n)$ in the limit, see [5].



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

□

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..nops(X)]:
for p from 1 to nops(X) do
for q from 1 to nops(X) 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 1 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:
```

Statement 2. *There exists a limit-computable function $f : \mathbb{N} \to \mathbb{N}$ of unknown computability which eventually dominates every function $\delta : \mathbb{N} \to \mathbb{N}$ with a single-fold Diophantine representation.*

Proof. Statement 2 follows constructively from Theorem 2 by taking $f = \beta$ because the following conjunction

$$(\neg\mathcal{K}(\beta \text{ is computable})) \ \wedge\ (\neg\mathcal{K}(\beta \text{ is uncomputable}))$$

holds. Statement 2 follows non-constructively from Lemma 1 by taking $f = g$ because the following conjunction

$$(\neg\mathcal{K}(g \text{ is computable})) \ \wedge\ (\neg\mathcal{K}(g \text{ is uncomputable}))$$

holds. □

Since the function γ in Theorem 1 is not computable, Statement 2 does not follow from Theorem 1.

Proposition 5. *Statement 2 strengthens a mathematical theorem. Statement 2 refers to the current mathematical knowledge and may be false in the future. Statement 2 does not express what is currently unproved in mathematics.*

Proof. Statement 2 strengthens Statement 2 without the epistemic condition. The weakened Statement 2 is a theorem which follows from Theorem 1. Statement 2 claims that some mathematically defined function $f : \mathbb{N} \to \mathbb{N}$ satisfies

$$(f \text{ is computable in the limit}) \wedge (\neg\mathcal{K}(f \text{ is computable})) \wedge (\neg\mathcal{K}(f \text{ is uncomputable})) \wedge$$

$$(f \text{ eventually dominates every function } \delta : \mathbb{N} \to \mathbb{N} \text{ with a single-fold Diophantine representation})$$

Conjecture 1 disproves Statement 2. □

Statement 3. *In Statement 2, we can require that there exists a computer program which takes as input a non-negative integer n and prints the sequence $\{f_i(n)\}_{i=0}^{\infty}$ of non-negative integers converging to $f(n)$.*

Proof. Any computer program which implements the semi-algorithm shown in Flowchart 3 or 4 is right. □

Statement 4. *Statement 2 holds for finite-fold Diophantine representations.*

Proof. It follows from Lemma 1 by taking $f = h$ because the following conjunction

$$(\neg\mathcal{K}(h \text{ is computable})) \ \wedge \ (\neg\mathcal{K}(h \text{ is uncomputable}))$$

holds. □

Statement 4 strengthens Statement 2. For Statement 4, there is no known computer program that computes f in the limit.

3 Predicate $\mathcal{K}$ of the written down mathematical knowledge

In this section, $\mathcal{K}$ denotes both the predicate satisfied by every written down theorem and the finite set of all written down theorems. It changes what is taken as known in mathematics.

Proposition 6. *Since $\mathcal{K}$ is finite, there exists $k \in \mathbb{N}$ such that the computability of the function*

$$\mathbb{N} \ni n \to k + \gamma(n) \in \mathbb{N}$$

is unknown. For this k, Statements 2 and 4 hold when $f(n) = k + \gamma(n)$.

Proposition 7 is of little use because Proposition 6 contradicts Proposition 4 with the right definition of $\mathcal{K}$.

Proposition 7. *Statements 2 and 4 can be formulated as mathematical statements. This holds at any time.*

Proof. Let $\mathcal{K} = \{T_1, \ldots, T_n\}$. For $i \in \{1, \ldots, n\}$, let

$$A_i = \begin{cases} (f:\mathbb{N}\to\mathbb{N}) \wedge T_i \wedge (f \neq u_i), & \text{if } T_i \text{ states that a function } u_i:\mathbb{N}\to\mathbb{N} \text{ is computable} \\ (f:\mathbb{N}\to\mathbb{N}) \wedge T_i \wedge (f \neq v_i), & \text{if } T_i \text{ states that a function } v_i:\mathbb{N}\to\mathbb{N} \text{ is uncomputable} \\ f:\mathbb{N}\to\mathbb{N}, & \text{in other cases} \end{cases}$$

The conjunction $A_1 \wedge \ldots \wedge A_n$ expresses that

$$(f:\mathbb{N}\to\mathbb{N}) \wedge (\neg\mathcal{K}(f \text{ is computable})) \ \wedge \ (\neg\mathcal{K}(f \text{ is uncomputable}))$$

□

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, *http://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: 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, *http://doi.org/10.1090/conm/270*.

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

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

[5] 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, *http://doi.org/10.1007/978-3-319-18275-9_24*.

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