MATEMATYKA
DYSKRETNA
www.ii.uj.edu.pl/preMD/

Apoloniusz TYSZKA

An infinite loop in MuPAD which implements a limit-computable function $f: \mathbb{N} \setminus \{0\} \to \mathbb{N} \setminus \{0\}$ that cannot be bounded by any computable function

Preprint Nr MD 073
(otrzymany dnia 12.12.2013)

Kraków
2013

Redaktorami serii preprintów Matematyka Dyskretna są:
Wit FORYŚ (Instytut Informatyki UJ)
oraz
Mariusz WOŹNIAK (Katedra Matematyki Dyskretnej AGH)

An infinite loop in *MuPAD* which implements a limit-computable function $f: \mathbb{N} \setminus \{0\} \to \mathbb{N} \setminus \{0\}$ that cannot be bounded by any computable function

Apoloniusz Tyszka

Abstract

For a positive integer n, let $f(n)$ denote the smallest non-negative integer b such that for each system $S \subseteq \{x_i = 1,\ x_i + x_j = x_k,\ x_i \cdot x_j = x_k :\ i, j, k \in \{1, \ldots, n\}\}$ with a solution in non-negative integers $x_1, \ldots, x_n$, there exists a solution of S in $\{0, \ldots, b\}^n$. We prove that the function f is strictly increasing and cannot be bounded by any computable function. We present an infinite loop in *MuPAD* which takes as input a positive integer n and returns a non-negative integer on each iteration. Let $g(n, m)$ denote the number returned on the m-th iteration, if n is taken as input. Then,

$$g(n, m) \leq m - 1,$$

$$0 = g(n, 1) < 1 = g(n, 2) \leq g(n, 3) \leq g(n, 4) \leq \ldots$$

and

$$g(n, f(n)) < f(n) = g(n, f(n) + 1) = g(n, f(n) + 2) = g(n, f(n) + 3) = \ldots$$

Key words: infinite loop, limit-computable function, *MuPAD*, trial-and-error computable function.

2010 Mathematics Subject Classification: 03D25, 11U05, 68Q05.

Limit-computable functions, also known as trial-and-error computable functions, have been thoroughly studied, see [1, pp. 233–235] for the main results. The goal of this article is to present an infinite loop in *MuPAD* which finds the values of a limit-computable function $f: \mathbb{N} \setminus \{0\} \to \mathbb{N} \setminus \{0\}$ by an infinite computation, where f cannot be bounded by any computable function. There are many limit-computable functions $f: \mathbb{N} \setminus \{0\} \to \mathbb{N} \setminus \{0\}$ which cannot be bounded by any computable function. Unfortunately, for all known such functions f, it is difficult to write a suitable computer program. A sophisticated choice of a function f will allow us to do it.

Let

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

For a positive integer n, let $f(n)$ denote the smallest non-negative integer b such that for each system $S \subseteq E_n$ with a solution in non-negative integers $x_1, \ldots, x_n$, there exists a solution of S in $\{0, \ldots, b\}^n$. This definition is correct because there are only finitely many subsets of E_n.

Let $h = \{(1,1)\} \cup \left\{\left(n, 2^{2^{n-2}}\right) :\ n \in \{2,3,4,\ldots\}\right\}$. The system

$$\begin{cases} x_1 &= 1 \\ x_1 + x_1 &= x_2 \\ x_2 \cdot x_2 &= x_3 \\ x_3 \cdot x_3 &= x_4 \\ & \ldots \\ x_{n-1} \cdot x_{n-1} &= x_n \end{cases}$$

has a unique integer solution, namely $\left(1, 2, 4, 16, \ldots, 2^{2^{n-3}}, 2^{2^{n-2}}\right)$. Therefore, $f(n) \geq h(n)$ for any n. Additionally, $f(1) = 1$, $f(2) = 2$, $f(3) = 4$, and $f(n+1) \geq f(n) \cdot f(n) > f(n)$ for any $n \geq 2$.

Lemma ([3, p. 291]). *There is an algorithm which accepts as input any computable function $\gamma : \mathbb{N} \to \mathbb{N}$ and returns a positive integer $w(\gamma)$ and a computable function β which to each integer $n \geq w(\gamma)$ assigns a system $S \subseteq E_n$ such that S is satisfiable over non-negative integers and each tuple $(x_1, \ldots, x_n)$ of non-negative integers that solves S satisfies $x_1 = \gamma(n)$.*

Theorem 1. *The function f cannot be bounded by any computable function.*

Proof. Assume, on the contrary, that there exists a computable function $\psi : \mathbb{N} \setminus \{0\} \to \mathbb{N} \setminus \{0\}$ such that $f(n) \leq \psi(n)$ for any positive integer n. By the Lemma, there exist a positive integer n and a system $S \subseteq E_n$ such that S is satisfiable over non-negative integers and each tuple $(x_1, \ldots, x_n)$ of non-negative integers that solves S satisfies $x_1 = \psi(n) + 1$. By this and the definition of f, $f(n) \geq \psi(n) + 1$, a contradiction. □

The following infinite loop in *MuPAD* is also stored in [4]. It takes as input a positive integer n and returns a non-negative integer on each iteration. Unfortunately, on each iteration the program executes a brute force algorithm, which is very time consuming.

```
input("input the value of n",n):
X:=[0]:
while TRUE do
Y:=combinat::cartesianProduct(X $i=1..n):
W:=combinat::cartesianProduct(X $i=1..n):
for s from 1 to nops(Y) do
for t from 1 to nops(Y) do
m:=0:
for i from 1 to n do
if Y[s][i]=1 and Y[t][i]<>1 then m:=1 end_if:
for j from i to n do
for k from 1 to n do
if Y[s][i]+Y[s][j]=Y[s][k] and Y[t][i]+Y[t][j]<>Y[t][k]
then m:=1 end_if:
if Y[s][i]*Y[s][j]=Y[s][k] and Y[t][i]*Y[t][j]<>Y[t][k]
then m:=1 end_if:
end_for:
end_for:
end_for:
if m=0 and max(Y[t][i] $i=1..n)<max(Y[s][i] $i=1..n)
then W:=listlib::setDifference(W,[Y[s]]) end_if:
end_for:
end_for:
print(max(max(W[z][u] $u=1..n) $z=1..nops(W))):
X:=append(X,nops(X)):
end_while:
```

Theorem 2. *Let $g(n,m)$ denote the number returned on the m-th iteration, if n is taken as input. Then,*

$$g(n,m) \le m-1,$$

$$0 = g(n,1) < 1 = g(n,2) \le g(n,3) \le g(n,4) \le \dots$$

and

$$g(n,f(n)) < f(n) = g(n,f(n)+1) = g(n,f(n)+2) = g(n,f(n)+3) = \dots$$

Proof. Let us say that a tuple $y = (y_1, \dots, y_n) \in \mathbb{N}^n$ is a *duplicate* of a tuple $x = (x_1, \dots, x_n) \in \mathbb{N}^n$, if

$$(\forall i \in \{1, \dots, n\}\ (x_i = 1 \Longrightarrow y_i = 1)) \wedge$$
$$(\forall i, j, k \in \{1, \dots, n\}\ (x_i + x_j = x_k \Longrightarrow y_i + y_j = y_k)) \wedge$$
$$(\forall i, j, k \in \{1, \dots, n\}\ (x_i \cdot x_j = x_k \Longrightarrow y_i \cdot y_j = y_k))$$

For each positive integer n, $f(n)$ equals the smallest non-negative integer b such that for each $x \in \mathbb{N}^n$ there exists a duplicate of x in $\{0, \dots, b\}^n$. For each positive integers n and m, $g(n, m)$ equals the smallest non-negative integer b such that for each $x \in \{0, \dots, m-1\}^n$ there exists a duplicate of x in $\{0, \dots, b\}^n$. □

The next theorem is a corollary from Theorems 1 and 2.

Theorem 3. *There exists a computable function* $\varphi : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ *which satisfies the following conditions:*

1) *For each non-negative integers n and l,*

$$\varphi(n, l) \leq l$$

2) *For each non-negative integer n,*

$$0 = \varphi(n, 0) < 1 = \varphi(n, 1) \leq \varphi(n, 2) \leq \varphi(n, 3) \leq \dots$$

3) *For each non-negative integer n, the sequence* $\{\varphi(n, l)\}_{l \in \mathbb{N}}$ *is bounded from above.*

4) *The function*

$$\mathbb{N} \ni n \xrightarrow{\theta} \theta(n) = \lim_{l \to \infty} \varphi(n, l) \in \mathbb{N} \setminus \{0\}$$

is strictly increasing and cannot be bounded by any computable function.

5) *For each non-negative integer n,*

$$\varphi(n, \theta(n) - 1) < \theta(n) = \varphi(n, \theta(n)) = \varphi(n, \theta(n) + 1) = \varphi(n, \theta(n) + 2) = \dots$$

Proof. Let $\varphi(n, l) = g(n + 1, l + 1)$. The following *MuPAD* code, which is also stored in [2], computes the values of $\varphi(n, l)$.

```
input("input the value of n",n):
input("input the value of l",l):
n:=n+1:
X:=[i $ i=0..l]:
Y:=combinat::cartesianProduct(X $i=1..n):
W:=combinat::cartesianProduct(X $i=1..n):
for s from 1 to nops(Y) do
for t from 1 to nops(Y) do
m:=0:
for i from 1 to n do
if Y[s][i]=1 and Y[t][i]<>1 then m:=1 end_if:
for j from i to n do
for k from 1 to n do
if Y[s][i]+Y[s][j]=Y[s][k] and Y[t][i]+Y[t][j]<>Y[t][k]
then m:=1 end_if:
if Y[s][i]*Y[s][j]=Y[s][k] and Y[t][i]*Y[t][j]<>Y[t][k]
then m:=1 end_if:
end_for:
end_for:
end_for:
if m=0 and max(Y[t][i] $i=1..n)<max(Y[s][i] $i=1..n)
then W:=listlib::setDifference(W,[Y[s]]) end_if:
end_for:
end_for:
print(max(max(W[z][u] $u=1..n) $z=1..nops(W))):
```

□

The commercial version of *MuPAD* is no longer available as a stand-alone product, but only as the *Symbolic Math Toolbox* of *MATLAB*. Fortunately, the presented codes can be executed by *MuPAD Light*, which was and is free, see [5].

References

[1] R. I. Soare, *Interactive computing and relativized computability,* in: *Computability: Turing, Gödel, Church, and beyond* (eds. B. J. Copeland, C. J. Posy, and O. Shagrir), MIT Press, Cambridge, MA, 2013, 203–260.

[2] A. Tyszka, *A MuPAD code that computes the values of* $\varphi(n,l)$, *http://www.cyf-kr.edu.pl/~rttyszka/phi.txt*.

[3] A. Tyszka, *A new characterization of computable functions,* An. Stiint. Univ. Ovidius Constanta Ser. Mat. 21 (2013), no. 3, 289–293, *http://www.emis.de/journals/ASUO/mathematics_/pdf30/Tyszka_A..pdf*, *http://www.anstuocmath.ro/mathematics//pdf30/Tyszka_A..pdf*.

[4] A. Tyszka, *An infinite loop in MuPAD,* *http://www.cyf-kr.edu.pl/~rttyszka/loop.txt*.

[5] A. Tyszka, *Links to an installation file for MuPAD Light,* *http://www.ts.mah.se/utbild/ma7005/mupad_light_scilab_253.exe*, *http://caronte.dma.unive.it/info/materiale/mupad_light_scilab_253.exe*, *http://www.cyf-kr.edu.pl/~rttyszka/mupad_light_scilab_253.exe*, *http://www.projetos.unijui.edu.br/matematica/amem/mupad/mupad_light_253.exe*, *http://www.cyf-kr.edu.pl/~rttyszka/mupad_light_253.exe*.

Apoloniusz Tyszka
University of Agriculture
Faculty of Production and Power Engineering
Balicka 116B, 30-149 Kraków, Poland
E-mail address: *rttyszka@cyf-kr.edu.pl*