SpecVer & LEM'12 2018 1–33
Dombrova Research

On Collatz theorem

Report of – September 3, 2021

Grażyna Mirkowska
Faculty of Mathematics and Natural Sciences
UKSW Wôycickiego 1/3
01-938 Warszawa Poland
G.Mirkowska@uksw.edu.pl

Andrzej Salwicki
Dombrova Research
Partyzantôw 19
05-092 Łomianki, Poland
salwicki@gmail.com

Abstract. Collatz conjecture has a proof.
We present a couple of observations. 1. The problem is of algorithmic nature, The conjecture states that Collatz algorithm Cl has the halting property. 2. There is an evidence of infinite execution of the program in a non-standard model of Peano arithmetic. 3. For every natural number n there exists numbers x, y, z such that the equation $n \cdot 3^x + y = 2^z$ holds. 4. Another algorithm IC computes on triples x, y, z. The consecutivve states of memory of any computation form monotone, descending sequences. 5. Hence, if a computation on triples is finite and successful then the corresponding computation of Collatz algorithm is finite too. 6. We construct an infinite set Z of elementary sentences that express the negation of halting proprty of Collatz algorithm. 7. The set Ax' of formulas that contains all axioms of elementary theory of addition of natural numbers and the set Z is consistent and has a model. Let $\mathfrak{M}$ denote any structure which is a model of axioms Ax'. 8. We show that the structure $\mathfrak{M}$ is not isomorphic to the standard structure of natural numbers with addition.
From this we infer that execution of Collatz algorithm in standard model of arithmetic is finite.

1. Introduction

The conjecture formulated by Lothar Collatz in 1937 is an algorithmic problem [1]. It should be shown that the following Cl program, executed in the standard structure $\mathfrak{N}$ of natural numbers with addition, has a finite computation for each $n \neq 0$.
We will be investigating the stop property of the following Cl program.

$$Cl: \left\{ \begin{array}{l} \textbf{while } n \neq 1 \textbf{ do} \\ \quad \textbf{if } even(n) \textbf{ then } n := \frac{n}{2} \textbf{ else } n := 3n+1 \textbf{ fi} \\ \textbf{od} \end{array} \right\}$$

In 2004, we noticed that there is a counterexample, see Appendix C (page 28). The two conclusions that can be made out of it, are:

- The formulation of the Collatz hypothesis requires clarification that the calculations are carried out in the standard model of natural numbers with addition (the operations of multiplication by 3 and division by 2 can be defined in Presburger arithmetic by means of addition).
- The Peano axioms, much less the Presburger axioms, are not sufficient to prove the conjecture.

And we found that if the conjecture is true, then it is a theorem of the algorithmic theory of natural numbers $\mathcal{ATN}$, (see the page 4).

2. Collatz tree

It is easy to notice that the set of those natural numbers for which the computation of the Collatz algorithm is finite, forms a tree.

Definition 2.1. ***Collatz tree*** $\mathcal{DC}$ is a subset $D \subset N$ of the set N of natural numbers and the function f defined on the set $D \setminus \{1\}$.

$$\mathcal{DC} = \langle D, f \rangle$$

[1] It may be surprising that some people place this problem in the theory of dynamical systems.

where $D \subset N, 1 \in D, f\colon D \setminus \{1\} \to D$.
Function f **is determined as follows**

$$f(n) = \begin{cases} n \div 2 & \textbf{when } n \mod 2 = 0 \\ 3n+1 & \textbf{when } n \mod 2 = 1 \end{cases}$$

, the set D **is the least set containing the number 1 and closed with respect to the function** f**,**

$$D = \{n \in N : \exists_{i \in N}\ f^i(n) = 1\ \}\ .$$

As it is easy to see, this definition is highly entangled and the decision whether the set D **contains every natural number is equivalent to the Collatz problem.**

Remark 2.1. Set D has the following properties :

$$x \in D \implies (x+x) \in D \tag{1}$$
$$x \in D \wedge \exists_y x = y + y \implies y \in D \tag{2}$$
$$x \in D \wedge \exists_y x = y + y + 1 \implies (x+x+x+1) \in D \tag{3}$$
$$x \in D \wedge (\exists_e \exists_z e = z + z + 1 \wedge x = e + e + e + 1) \implies e \in D \tag{4}$$

Implications (1) and (4) show left and right son of element x.
Similar, interesting properties has the complement of set D, if it is a non-empty set.

Remark 2.2. If the complement $N \setminus D$ is a non-empty set, then it has similar properties:

$$x \in N \setminus D \implies (x+x) \in N \setminus D \tag{5}$$
$$x \in N \setminus D \wedge \exists_y x = y + y \implies y \in N \setminus D \tag{6}$$
$$x \in N \setminus D \wedge \exists_y x = y + y + 1 \implies (x+x+x+1) \in N \setminus D \tag{7}$$
$$x \in N \setminus D \wedge (\exists_e \exists_z e = z + z + 1 \wedge x = e + e + e + 1) \implies e \in N \setminus D \tag{8}$$

Remark 2.3. ***If Collatz conjecture is not true, then both sets*** D ***and*** $N \setminus D$ ***are infinite.***

From properties (6) and (7) follow the

Fact 2.1. If the x element does not belong to the Collatz tree then the computation of the Collatz algorithm starting with the state $v(n) = x$ **is not finite.**

3. Halting formula

Halting formula of a program K is any formula that expresses the finiteness of a computation of the program K.

The following formula (halt) expresses the halting property of Collatz program Cl. It is to be shown that the formula is valid in the $\mathfrak{N}$ structure with addition natural numbers.

$$\left\{\begin{array}{l}\textbf{while } n \neq 1 \textbf{ do}\\ \quad \textbf{if } even(n) \textbf{ then } n := \frac{n}{2} \textbf{ else } n := 3n+1 \textbf{ fi}\\ \textbf{od}\end{array}\right\}(n=1) \qquad \text{(halt)}$$

Formula (LC) that asserts, there exists an iteration of the program if . . . fi such that the condition $(n = 1)$ holds after is a halting formula of program Cl too.

$$\bigcup\left\{\begin{array}{l}\textbf{if } n \neq 1 \textbf{ then}\\ \quad \left[\begin{array}{l}\textbf{if } n \underline{\bmod} 2 = 1\\ \qquad \textbf{then } n \leftarrow 3n+1\\ \qquad \textbf{else } n \leftarrow n \underline{\operatorname{div}} 2\end{array}\right]\\ \textbf{fi}\end{array}\right\}(n=1) \qquad \text{(LC)}$$

Our goal is to show that the halting formula is valid in the standard structure $\mathfrak{N}$ of natural numbers or to prove the formula in algorithmic theory of natural numbers $\mathcal{ATN}$. For any model of the theory is isomorphic to the structure $\mathfrak{N}$. See [MS87], page 139.

$$\text{Axioms of } \mathcal{ATN}\left\{\begin{array}{l}\forall_x\, x+1 \neq 0\\ \forall_{x,y}\, x+1 = y+1 \implies x = y\\ \forall_x \{y := 0; \textbf{while } y \neq x \textbf{ do } y := y+1 \textbf{ od}\}(y = x)\end{array}\right\} \qquad \text{(ATN)}$$

Theory $\mathcal{ATN}$ can be extended, we can add definitions of useful operations $+, \times 3, \div 2$, and the parity unary relation. We shall use the following notation the atomic formula $e(n)$ reads n ***is even***, and formula $o(n)$ reads n ***is odd***.

Making use of axioms of calculus of programs $\mathcal{AL}$ and axioms of algorithmic theory of natural numbers $\mathcal{ATN}$ we can write a couple of formulas that are equivalent to the halting formula halt. Below

we presnt one of these formulas

$$\left\{\begin{array}{l}
(o(n) \wedge \mathbf{n} = 1) \vee \\
(e(n) \wedge o(\frac{n}{2}) \wedge \mathbf{n} = 2) \vee \\
(e(n) \wedge e(\frac{n}{2}) \wedge o(\frac{n}{4}) \wedge \mathbf{n} = 4) \vee \\
(e(n) \wedge e(\frac{n}{2}) \wedge e(\frac{n}{4}) \wedge o(\frac{n}{8}) \wedge \mathbf{n} = 8) \vee \\
(e(n) \wedge e(\frac{n}{2}) \wedge e(\frac{n}{4}) \wedge e(\frac{n}{8}) \wedge o(\frac{n}{16}) \wedge \mathbf{n} = 16) \vee \\
\left(\begin{array}{l} e(n) \wedge e(\frac{n}{2}) \wedge e(\frac{n}{4}) \wedge e(\frac{n}{8}) \wedge e(\frac{n}{16}) \wedge o(\frac{n}{32}) \wedge \mathbf{n} = 32) \vee \\ o(n) \wedge e(3n+1) \wedge e(\frac{3n+1}{2}) \wedge e(\frac{3n+1}{4}) \wedge e(\frac{3n+1}{8}) \wedge o(\frac{3n+1}{16}) \wedge \mathbf{n} = 5 \end{array}\right) \vee \\
\left(\begin{array}{l} (e(n) \wedge e(\frac{n}{2}) \wedge e(\frac{n}{4}) \wedge e(\frac{n}{8}) \wedge e(\frac{n}{16}) \wedge e(\frac{n}{32}) \wedge o(\frac{n}{64}) \wedge \mathbf{n} = 64) \vee \\ e(n) \wedge o(\frac{n}{2}) \wedge e(\frac{3n+1}{2}) \cdots \wedge \mathbf{n} = 10 \end{array}\right) \vee \\
\left(\begin{array}{l} e(n) \wedge e(\frac{n}{2}) \wedge e(\frac{n}{4}) \wedge e(\frac{n}{8}) \wedge e(\frac{n}{16}) \wedge e(\frac{n}{32}) \wedge e(\frac{n}{64}) \wedge o(\frac{n}{128}) \wedge \mathbf{n} = 128) \vee \\ o(n) \wedge e(3n+1) \wedge e(\frac{3n+1}{2}) \cdots \wedge \mathbf{n} = 21) \\ e(n) \wedge o(\frac{n}{2}) \wedge e(\frac{3n+1}{2}) \cdots \wedge \mathbf{n} = 20 \vee \\ o(n) \wedge e(3n+1) \wedge o(\frac{3n+1}{2}) \cdots \wedge \mathbf{n} = 3 \end{array}\right) \vee \\
\left\{\begin{array}{l} \textbf{if } n \neq 1 \textbf{ then} \\ \quad \left[\begin{array}{l} \textbf{if } n \underline{\bmod} 2 = 1 \\ \quad \textbf{then } n \leftarrow 3n+1 \\ \quad \textbf{else } n \leftarrow n \underline{\operatorname{div}} 2 \end{array}\right] \\ \textbf{fi} \end{array}\right\}^8 \bigcup \left\{\begin{array}{l} \textbf{if } n \neq 1 \textbf{ then} \\ \quad \left[\begin{array}{l} \textbf{if } n \underline{\bmod} 2 = 1 \\ \quad \textbf{then } n \leftarrow 3n+1 \\ \quad \textbf{else } n \leftarrow n \underline{\operatorname{div}} 2 \end{array}\right] \\ \textbf{fi} \end{array}\right\} (n = 1)
\end{array}\right\}$$

Based on these experiments, we can make some observationsi:

- **the correspondence between the levels of the Collatz tree and the components of the halting formula is clearly visible,**
- **at higher levels of the Collatz tree more and more odd numbers appear - and the subsequent components of the halting formula are alternatives of longer and longer conjunctions of factors** $o(\mathbf{.})$ **or** $e(\mathbf{.})$**,**
- **Note, the alternative** $(n = 128 \vee n = 21 \vee n = 20 \vee n = 3)$ **may be written as follows** $(n \cdot 3^0 + 0 = 2^7 \vee n \cdot 3^1 + 1 = 2^6 \vee n \cdot 3^1 + 4 = 2^6 \vee n \cdot 3^2 + 5 = 2^5)$.

In the following section we shall use this observation.

4. Trójki

Inspirując się obserwacjami poczynionymi podczas prób konstrukcji kolejnych składników formuły stopu wprowadzamy pojęcia: trójki reprezentującej liczbę naturalną i obliczenia na trójkach.
Zacznijmy od następującego spostrzeżenia

Fact 4.1. ***Dla każdej liczby naturalnej*** $n \neq 0$ ***istnieje nieskończenie wiele trójek liczb naturalnych*** x, y, z ***takich, że spełniona jest równość***

$$n \cdot 3^x + y = 2^z$$

Mówimy, że trójka x, y, z reprezentuje (albo, ***koduje***) liczbę n i zapisujemy to $\langle x, y, z\rangle \asymp n$.

Proof:
Dowód tego intuicyjnego faktu wykorzystuje prawo Archimedesa[2].
Niech n będzie dowolnie ustaloną liczbą naturalną.
Wybierzmy liczbę x, może to być dowolnie duża liczba naturalna $x \geq 0$.
Wyznaczymy liczby naturalne y i z w taki sposób, by zachodziła równość $n \cdot 3^x + y = 2^z$. Niech liczba $k = n \cdot 3^x$. Przyjmiemy $z = \mu(2^z \geq k)$ tj. liczba 2^z jest najmniejszą potęgą liczby 2 większą lub równą k. Kładziemy $y = 2^z - k$. □

Example 4.1. Trójka $\langle 1, 7, 6\rangle$ reprezentuje liczbę 19 ponieważ $19 \cdot 3 + 7 = 2^6$.
Liczba 19 jest także reprezentowana przez inne trójki

$\langle 1, 7, 6\rangle$	$19 \cdot 3^{\mathbf{1}} + \mathbf{7} = 2^{\mathbf{6}}$
$\langle 2, 85, 8\rangle$	$19 \cdot 3^{\mathbf{2}} + \mathbf{85} = 2^{\mathbf{8}}$
$\langle 3, 511, 10\rangle$	$19 \cdot 3^{\mathbf{3}} + \mathbf{511} = 2^{\mathbf{10}}$
$\langle 4, 509, 11\rangle$	$19 \cdot 3^{\mathbf{4}} + \mathbf{509} = 2^{\mathbf{11}}$
$\langle 5, 3575, 13\rangle$	$19 \cdot 3^{\mathbf{5}} + \mathbf{3575} = 2^{\mathbf{13}}$
$\langle 6, 2533, 14\rangle$	$19 \cdot 3^{\mathbf{6}} + \mathbf{2533} = 2^{\mathbf{14}}$
$\langle 7, 23983, 16\rangle$	$19 \cdot 3^{\mathbf{7}} + \mathbf{23983} = 2^{\mathbf{16}}$
...	

[2]Przypomnijmy, prawo Archimedesa jest własnością algorytmiczną, prawo to nie jest wyrażalne formułą pierwszego rzędu.

Natomiast, nie każda trójka liczb naturalnych reprezentuje jakąś liczbę naturalną, np. trójki $\langle 2,4,11\rangle$, $\langle 2,4044,11\rangle$.

Fakt 1 można także wyprowadzić formalnie w elementarnej teorii $\mathcal{T}$ dodawania liczb naturalnych.

Theorem 4.1. Zdanie $\forall_n \exists_{,x,y.z}\, n \cdot 3^x + y = 2^z$ jest twierdzeniem teorii $\mathcal{T}$,

Dowód znajdziesz w dodatku B. Wynika stąd, że własność ta jest prawdziwa w każdym modelu tej teorii.

4.1. Własności trójek

Definition 4.1. *Dwie trójki* $\langle x,y,z\rangle$ *i* $\langle u,v,t\rangle$ *są <u>równoważne</u> gdy reprezentują tę samą liczbę naturalną*

$$\langle x,y,z\rangle \equiv \langle u,v,t\rangle \quad \stackrel{\text{df}}{=} \quad \frac{2^z-y}{3^x} \in N \wedge \frac{2^z-y}{3^x} = \frac{2^t-v}{3^u}$$

W zbiorze trójek możemy zdefiniować porządek leksykograficzny $\succ$.

Definition 4.2.

$$\langle u,v,t\rangle \succ \langle x,y,z\rangle \stackrel{df}{\Leftrightarrow} x<u \text{ lub } x=u \text{ i } z<t \text{ lub } x=u \text{ i } z=t \text{ i } y<v$$

Z faktu 4.1 wynika, że dla kazdej liczby naturalnej n można wskazać kodującą ją trójkę, która ma liczbę y większą od dowolnie wybranej liczby naturalnej k.

Definition 4.3. *Nieparzystość* trójek wyznaczona jest przez nieparzystość liczby y

$$odd(\langle x,y,z\rangle) \stackrel{df}{\Leftrightarrow} \text{y jest nieparzyste}$$

Kolejne spostrzeżenie

Fact 4.2. *Liczba* n *jest nieparzysta wttw gdy reprezentująca ją trójka* $\langle x,y,z\rangle$ *jest nieparzysta.*

Definition 4.4. Jedynka

$$equal1(\langle x,y,z\rangle) \stackrel{df}{\Leftrightarrow} (2^z - y = 3^x)$$

Definition 4.5. Klasa trójek równowaznych jest zbiorem trójek reprezentujących tę samą liczbę n.

Jedynka jest reprezentowana przez wiele trójek, np. $\langle 0,0,0\rangle, \langle 0,1,1\rangle, \langle 1,1,2\rangle, \langle 1,5,3\rangle, \langle 2,$

Na trójkach możemy określić dwie operacje

Definition 4.6. Operacja $div2$ przeprowadza trójkę parzystą $\langle x,y,z\rangle$ w trójkę $\langle x, y \div 2, z-1\rangle$

$$\langle x,y,z\rangle \xrightarrow{\{div2\}} \langle x, y \div 2, z-1\rangle$$

Definition 4.7. Operacja $mult3$ określona jest na trójkach $\langle x,y,z\rangle$ takich, że trójka $\langle x,y,z\rangle$ jest nieparzysta i $x > 0$ i $y > 3^{x-1}$. W takim przypadku operacja $mult3$ przeprowadza trójkę $\langle x,y,z\rangle$ w trójkę $\langle x-1, y-3^{x-1}, z\rangle$

$$\langle x,y,z\rangle \xrightarrow{\{mult3\}} \langle x-1, y-3^{x-1}, z\rangle$$

Zauważ

Fact 4.3. ***Trójka*** $\langle x,y,z\rangle$ ***reprezentuje liczbę parzystą*** n ***wttw gdy trójka*** $\langle x, y \div 2, z-1\rangle$ ***reprezentuje liczbę*** $n \div 2$.
Ponadto $\langle x, y \div 2, z-1\rangle \prec \langle x,y,z\rangle$.
Jeśli liczba y ***jest nieparzysta i zachodzi*** $x > 0 \wedge y > 3^{x-1}$ ***to trójka*** $\langle x,y,z\rangle$ ***reprezentuje liczbę*** n ***wttw gdy trójka*** $\langle x-1, y-3^{x-1}, z\rangle$ ***reprezentuje liczbę*** $3n+1$.
Ponadto $\langle x-1, y-3^{x-1}, z\rangle \prec \langle x,y,z\rangle$.

Zauważmy z kolei

Lemma 4.1. ***Jeśli program*** K: {if odd then $mult3$ else $div2$ fi} ***przeprowadza trójkę*** $\langle x,y,z\rangle$ ***w trójkę*** $\langle u,v,t\rangle$, ***to wynikowa trójka*** $\langle u,v,t\rangle$ ***jest mniejsza*** $\prec$ ***od trójki*** $\langle x,y,z\rangle$.

$$\langle x,y,z\rangle \xrightarrow{\{\textbf{if } odd \textbf{ then } mult3 \textbf{ else } div2 \textbf{ fi}\}} \langle u,v,t\rangle \text{ implikuje } \langle u,v,t\rangle \prec \langle x,y,z\rangle$$

Proof:
Jeśli liczba y jest parzysta to od z odejmujemy jedynkę i dzielimy y przez 2. W przeciwnym przypadku zmniejszana jest liczba x i od y odejmowana jest liczba 3^x. □

W każdym kroku obliczenia algorytmu Collatza wyrażenie $x + z$ zmniejsza swą wartość o 1, zmniejszana jest też wartość zmiennej y.

Wynika stąd, że każde obliczenie w strukturze $\mathfrak{T}$ jest skończone. Obserwacje poczynione dotychczas pozwalają stwierdzić, że następujący diagram jest przemienny.

Fact 4.4. ***Działanie na trójkach prowadzi od trójki*** $\langle x, y, z\rangle$ ***kodującej liczbę*** n ***i spełnajacej przy tym warunek*** $\zeta : \big((y \bmod 2 = 1) \Rightarrow (x > 0 \wedge y > 3^{x-1})\big)$, ***do kolejnej trójki*** $\langle u, v, t\rangle$, ***która reprezentuje wynik*** m ***następującej instrukcji warunkowej*** K : **{**if $odd(n)$ then $n := 3 * n + 1$ else $n := ndiv2$ fi**}**.

$$\zeta \Rightarrow \left\{\begin{array}{ccc} n & \xrightarrow{K:\{\textbf{if } odd(n) \textbf{ then } m{:=}3n{+}1 \textbf{ else } m{:=}n/2 \textbf{ fi}\}_{\mathfrak{N}}} & m \\ \uparrow{\scriptstyle n\cdot 3^x+y=2^z} & & {\scriptstyle m\cdot 3^u+v=2^t}\uparrow \\ \langle x,y,z\rangle & \xrightarrow{\bar{K}:\{\textbf{if } odd(y) \textbf{ then } u,v,t{:=}x{-}1,y{-}3^{x-1},z \textbf{ else } u,v,t{:=}x,y/2,z{-}1 \textbf{ fi}\}_{\mathfrak{T}}} & \langle u,v,t\rangle \end{array}\right.$$

Diagram ten jest przemienny pod warunkiem, że można wykonać operację odejmowania tzn. gdy y jest liczbą parzystą lub y jest nieparzyste i ($x > 0$ lub $y > 3^{x-1}$). Semantyczną treść opisaną powyższym diagramem można wyrazic następującą formuła

$$\big((n \cdot 3^x + y = 2^z \wedge odd(y)) \Rightarrow (x > 0 \wedge y > 3^{x-1}))\big) \Rightarrow \{K; \bar{K}\}(n \cdot 3^x + y = 2^z)$$

Pamiętajmy, że programy K oraz $\bar{K}$ są niezależne, a więc przemienne.

$$\big((n \cdot 3^x + y = 2^z \wedge odd(y)) \Rightarrow (x > 0 \wedge y > 3^{x-1}))\big) \Rightarrow \{\bar{K}; K\}(n \cdot 3^x + y = 2^z)$$

Przemienność tego diagramu pozwala nam dostrzec, że każdemu obliczeniu algorytmu Collatza Cl w strukturze $\mathfrak{N}$ liczb naturalnych, odpowiada obliczenie sprzężonego z Cl, algorytmu IC w strukturze trójek $\mathfrak{T}$.

Lemma 4.2. Jeśli spełniony jest warunek θ

$$\theta : \ (n \cdot 3^x + y = 2^z) \wedge \bigcap \left\{\begin{array}{l} \textbf{if } 3^x + y \neq 2^z \\ \textbf{then} \\ \quad \textbf{if } odd(y) \\ \quad \textbf{then } x, y := x - 1, y - 3^{x-1} \\ \quad \textbf{else } z, y := z - 1, y \div 2 \\ \quad \textbf{fi} \\ \textbf{fi} \end{array}\right\} (odd(y) \implies (x > 0 \wedge y > 3^{x-1}))$$

to następujący diagram jest przemienny

$$\begin{array}{ccc} n & \xrightarrow{\{\textbf{while } n\neq 1 \textbf{ do if } odd(n) \textbf{ then } n{:=}3n{+}1 \textbf{ else } n{:=}n/2 \textbf{ fi od}\}_{\mathfrak{N}}} & 1 \\ \uparrow{\scriptstyle n\cdot 3^x+y=2^z} & & \uparrow{\scriptstyle 3^x+y=2^z} \\ \langle x,y,z\rangle & \xrightarrow{\left\{\begin{array}{l}\textbf{while } 3^x+y\neq 2^z \textbf{ do}\\ \quad \textbf{if } odd(y) \textbf{ then } x,y,z:=x-1,y-3^{x-1},z\\ \quad \textbf{else } x,y,z:=x,y/2,z-1 \textbf{ fi}\\ \textbf{od}\end{array}\right\}_{\mathfrak{T}}} & \langle x,y,z\rangle \end{array}$$

Co wyraża się następującą formułą

$$\theta \Rightarrow \big(\{Cl\}(n=1) \Leftrightarrow \{IC\}(3^x+y=2^z)\big)\ . \tag{RCI}$$

Proof:
gos by induction with respect to the number of iterations. Base of induction reduces to the commutativity of the preceding fact

$$\zeta \Rightarrow \left[\begin{array}{ccc} n & \xrightarrow{\{K\}_{\mathfrak{N}}} & m \\ \uparrow{\scriptstyle n\cdot 3^x+y=2^z} & & \uparrow{\scriptstyle m\cdot 3^u+v=2^t} \\ \langle x,y,z\rangle & \xrightarrow{\bar{K}_{\mathfrak{T}}} & \langle u,v,t\rangle \end{array}\right]\ .$$

If the execution of algorithm is longer, say of length 1, then the following diagram applies

$$(\zeta \wedge \{\bar{K}\}\zeta) \Rightarrow \left[\begin{array}{ccccc} n & \xrightarrow{\{K\}_{\mathfrak{N}}} & n_1 & \xrightarrow{\{K\}_{\mathfrak{N}}} & n_2 \\ \uparrow{\scriptstyle n\cdot 3^x+y=2^z} & & \uparrow{\scriptstyle n_1\cdot 3^{x_1}+y_1=2^{z_1}} & & \uparrow{\scriptstyle n_2\cdot 3^{x_2}+y_2=2^{z_2}} \\ \langle x,y,z\rangle & \xrightarrow{\bar{K}_{\mathfrak{T}}} & \langle x_1,y_1,z_1\rangle & \xrightarrow{\bar{K}_{\mathfrak{T}}} & \langle x_2,y_2,z_2\rangle \end{array}\right]\ .$$

By induction, we prove that for every natural number i the following diagram commutes

$$\bigwedge_{j=0}^{i}\{\bar{K}\}^j\zeta \Rightarrow \left[\begin{array}{ccccccccc} n & \xrightarrow{\{K\}_{\mathfrak{N}}} & n_1 & \xrightarrow{\{K\}_{\mathfrak{N}}} & n_2 & \xrightarrow{\cdots} & & \xrightarrow{\{K\}_{\mathfrak{N}}} & n_i \\ \uparrow{\scriptstyle n\cdot 3^x+y=2^z} & & \uparrow{\scriptstyle n_1\cdot 3^{x_1}+y_1=2^{z_1}} & & \uparrow{\scriptstyle n_2\cdot 3^{x_2}+y_2=2^{z_2}} & & \cdots\uparrow & & \uparrow{\scriptstyle n_i\cdot 3^{x_i}+y_i=2^{z_i}} \\ \langle x,y,z\rangle & \xrightarrow{\bar{K}_{\mathfrak{T}}} & \langle x_1,y_1,z_1\rangle & \xrightarrow{\bar{K}_{\mathfrak{T}}} & \langle x_2,y_2,z_2\rangle & \xrightarrow{\cdots} & & \xrightarrow{\bar{K}_{\mathfrak{T}}} & \langle x_i,y_i,z_i\rangle \end{array}\right]\ .$$

Now, if for some $i \in N$ the number $n_i = 1$, then $\forall_k n_{i+k} = 1$. Thus, the formula RCI has a proof. □

Formuła RCI jest twierdzeniem algorytmicznej teorii $\mathcal{ATN}$ ale nie jest rôwnowazna twierdzeniu Collatza. W rozdziale 5 nawiążemy do tego twierdenia.

4.2. Zbiory T_n

Każdej liczbie n przyporządkowujemy zbiór T_n trójek x, y, z takich, że zachodzi równość $n3^x + y = 2^z$.
Zbiór taki jest niepusty.
Zbiory T_n są parami rozłączne, tj. $n \neq m \implies \mathrm{T}_n \cap \mathrm{T}_m = \emptyset$.
Zbiór T_n jest zamknięty ze względu na operacje $o_1, o_2, o_3, o_4, o_5, o_6$

$$o_1(\langle x,y,z\rangle) = \langle x+1, 3y+2^z, z+2\rangle \tag{9}$$
$$o_2(\langle x,y,z\rangle) = \langle x+1, 3y-2^z, z+1\rangle \quad \textbf{gdy } 3y > 2^z \tag{10}$$
$$o_3(\langle x,y,z\rangle) = \langle x, y+2^z, z+1\rangle \tag{11}$$
$$o_4(\langle x,y,z\rangle) = \langle x, y-2^{z-1}, z-1\rangle \quad \textbf{gdy } y > 2^{z-1} \tag{12}$$
$$o_5(\langle x,y,z\rangle) = \langle x-1, \frac{y-2^{z-2}}{3}, z-2\rangle \quad \textbf{gdy } y > 2^{z-2} \textbf{ i } (y-2^{z-2}) \mathrm{mod} 3 = 0 \tag{13}$$
$$o_6(\langle x,y,z\rangle) = \langle x-1, \frac{y+2^{z-1}}{3}, z-1\rangle \tag{14}$$

Wynika stąd, że zbiór T_n jest nieskończony i nieograniczony.

Zauważ związki

$o_5(o_1(\langle x,y,z\rangle) = \langle x,y,z\rangle$
$o_4(o_3(\langle x,y,z\rangle) = \langle x,y,z\rangle$
$o_6(o_2(\langle x,y,z\rangle) = \langle x,y,z\rangle$
$o_4(o_1(\langle x,y,z\rangle) = \langle x,y,z\rangle$

O trójkach zbędnych (niepotrzebnych) ...
Trójka wieksza od trójki przydatnej nie musi być przydatna, może być zbędna.
PRZYKŁAD.
Ale też od każdej trójki dobrej istnieje większa od niej trójka dobra.

4.3. Program IC

A więc można realizować obliczenia algorytmu Collatza "*na trójkach*".
Rozważmy następujący program. Zakładamy, że program IC startuje z wartościami x, y, z spełniającymi warunel $n \cdot 3^x + y = 2^z \wedge \neg Err$.
Jest to warunek *wstępny* obliczeń.

$$IC: \left\{ \begin{array}{l} \textbf{while } 3^x + y \neq 2^z \ (*\ tj.\ n \neq 1\ *) \textbf{ do} \\ \quad \textbf{if } \big(odd(y) \wedge ((x=0) \textbf{ or } (y < 3^{x-1}))\big) \textbf{ then } Err := true; \textbf{ exit fi}; \\ \quad \textbf{if } odd(y) \textbf{ then } x := x-1;\ y := y-3^x;\ (*n := 3*n+1\ *) \\ \quad \textbf{else } \ z := z-1;\ y := y \,\mathrm{div}\, 2;\ (*\, n := \tfrac{n}{2}\ *) \textbf{ fi} \\ \textbf{od} \end{array} \right\}$$

Obliczenie w strukturze trójek jest zawsze skończone. Wynika to z następującego twierdzenia algorytmicznej teorii liczb naturalnych $\mathcal{ATN}$.

$$\mathcal{ATN} \vdash \forall x\{\textbf{while } x \neq 0 \textbf{ do } x := x - 1 \textbf{ od}\}(x = 0)$$

Co prawda, może się zdarzyć, że obliczenie algorytmu IC zakończy się niepowodzeniem, $Err = true$.

4.4. Własności programu IC

Fakt 1. ***Każde obliczenie programu IC jest skończone i albo jest sukces i osiągnięto jedynkę albo jest błąd i obliczenia algorytmu IC nie można kontynuować.***

$$\mathfrak{N} \models \forall n\,(n \cdot 3^x + y = 2^z) \implies \{IC\}\big(\underbrace{(3^x + y = 2^z)}_{\langle x,y,z\rangle \asymp 1} \vee \underbrace{(odd(y) \wedge (x = 0 \vee y < 3^{x-1}))}_{Error}\big)$$

Błąd o jakim tu mowa to niemożność kontynuowania obliczeń. Jest tak gdy y *jest nieparzyste* i $y < 3^{x-1}$ lub gdy y *jest nieparzyste* i $x = 0$. Pierwszy błąd łatwo wyeliminować, zauważ, że trójka $\langle x, y, z\rangle$ reprezentuje tę samą liczbę co trójka $\langle x, y + 2^z, z + 2\rangle$. Wystarczy więc w sytuacji gdy y *jest nieparzyste* i $y < 3^{x-1}$ zastąpić trójkę $\langle x, y, z\rangle$ przez trójkę $\langle x, y + 2^z, z + 2\rangle$. Zniknie też ryzyko wystąpienia błędu drugiego rodzaju. Upoważnia to nas do sformułowania następującej uwagi.

Uwaga 2. Jeśli obliczenie algorytmu IC rozpoczynające się od pewnej trójki $\langle x, y, z\rangle$ kończy sie po k iteracjach błędem $Err = true$ to obliczenie algorytmu IC rozpoczynającego się od trójki $\langle x + 1, 3y + 2^z, z + 2\rangle$ będzie dłuższe o co najmniej dwie iteracje. Obie te trójki reprezentują tę sama liczbę n.

5. Proof of Collatz theorem

Our plan may be summarized in five points:

(*i*) **We prove that the sentence $\forall_n \exists_{x,y,z}\, n \cdot 3^x + y = 2^z$ is a theorem of the elementary theory Ar of addition of natural numbers (Presburger theory). See Appendix B, page 27. So this sentence holds true for any model of the theory.**

(*ii*) **In Appendix C, page 28, we show infinite computations of Collatz's algorithm in a non-standard (non-Archimedean) model**

of Presburger arithmetic Ar. The model is computable and programmable. This example is not a complete counter-example against Collatz conjecture.

(*iii*) We show that there is a model $\mathfrak{M}$ of the Ar theory in which there is an infinite computation. *We do not assume* that this model contains unreachable elements.

(*iv*) We show that in any model of Ar theory , if for a certain n element, the computation of the Collatz algorithm is infinite, then the model is not isomorphic to the standard model of natural numbers (for it contains unrechable elements).

(*v*) From this we conclude, that if a model has no unreachable elements, then there are no infinite computations.

5.1. Structure in which some element has an infinite Collatz computation.

We have known about the existence of such a structure for several years. It is described in the literature, and in section 9, Appendix C we provide a definition of this structure in a programming language. Now, however, we are not going to assume that there are unreachable elements in the structure.
We construct an elementary theory of Ar' which is an extension of the theory of Ar (Presburger arithmetic) in a way that permits to show an element different from any number that occurs in Cllatz tree.
The modification of axioms of theory Ar is made in three steps

1. (*language*) we add four new constants $\varepsilon, c_x, c_y, c_z$ to the alphabet and correspondingly we extend the sets of term and of formulas of the language of theory Ar.

2. (*logic*) the operation of consequence remains the same, remember the sets of terms and of formulas are bigger,

3. (axioms of data structure) to the set Ax of Presburger's axioms we add the sentence $P3(\varepsilon, c_x) + c_y = P2(c_z)$[3] and two infinite sets of sentences Z and Y

Our intention is to prove two facts.

[3]this sentence expresses the property $(\varepsilon \cdot 3^x + y = 2^z)$

1. An assumption that for some element ε the Collatz computation is infinite, does not lead to a contradiction, and

2. Every algebraic structure $\mathfrak{M}$ that admits infinite Collatz computation is not isomorphic to the standard model of natural numbers with addition.

As a natural reflex, we would like to add the negation of the halting formula for $n = \varepsilon$ to the axioms of Ar theory. However the formula

$$\neg\{n \leftarrow \varepsilon\} \left\{ \begin{array}{l} \textbf{while } n \neq 1 \textbf{ do} \\ \quad \textbf{if } Parz(n) \\ \quad \textbf{then } n \leftarrow n \textbf{ div } 2 \\ \quad \textbf{else } n \leftarrow 3n+1 \\ \quad \textbf{fi} \\ \textbf{od} \end{array} \right\} (n = 1)$$

does not belong to the language of elementary theory Ar. Moreover, the following formula 15 that expresses the same looping property of computation of Collatz algorithm does not belong to the language of first-order theory Ar.

$$\{n \leftarrow \varepsilon\} \bigcap \left\{ \begin{array}{l} \textbf{if } n \neq 1 \textbf{ then} \\ \quad \textbf{if } n \textbf{ mod } 2 = 0 \\ \quad \textbf{then } n \leftarrow n \textbf{ div } 2 \\ \quad \textbf{else } n \leftarrow 3n+1 \\ \quad \textbf{fi} \\ \textbf{fi} \end{array} \right\} (n \neq 1) \qquad (15)$$

However, for every algebraic structure $\mathfrak{M}$ the above formula (15) is valid in $\mathfrak{M}$ if and only if every formula of the following scheme is

valid in $\mathfrak{M}$

$$\{n \leftarrow \varepsilon\}(n \neq 1),$$

$$\{n \leftarrow \varepsilon\} \left\{ \begin{array}{l} \textbf{if } n \neq 1 \textbf{ then} \\ \quad \textbf{if } Parz(n) \textbf{ then } n \leftarrow n \textbf{ div } 2 \\ \quad\quad \textbf{else } n \leftarrow 3n+1 \textbf{ fi} \\ \textbf{fi} \end{array} \right\} (n \neq 1),$$

$$\ldots$$

$$\{n \leftarrow \varepsilon\} \left\{ \begin{array}{l} \textbf{if } n \neq 1 \textbf{ then} \\ \quad \textbf{if } Parz(n) \textbf{ then } n \leftarrow n \textbf{ div } 2 \\ \quad\quad \textbf{else } n \leftarrow 3n+1 \textbf{ fi} \\ \textbf{fi} \end{array} \right\}^{i} (n \neq 1),$$

$$\ldots$$

Każda z tych formuł jest równoważna pewnej formule pierwszego rzędu, która już nie zawiera programów. Łatwo to wykazać posługując się aksjomatami instrukcji przypisana, np. $\{n \leftarrow \varepsilon\}(n > 1) \equiv (\varepsilon > 1)\}$ **i aksjomatami instrukcji złożenia oraz instrukcji warunkowej. Dla przykładu podamy równoważność**

$$\{n \leftarrow \varepsilon\}\{\textbf{if } P(n) \textbf{ then } n \leftarrow n \textbf{ div } 2 \textbf{ else } n \leftarrow 3n+1 \textbf{ fi}\}(n \neq 1) \equiv \Big(\big(P(\varepsilon) \wedge \varepsilon \neq 2\big) \vee \underbrace{\big(\neg P(\varepsilon) \wedge 3\varepsilon + 1 \neq 1\big)}_{false} \Big)$$

Kontynuując otrzymamy m.in. następujące formuły

Formuła
$(o(\varepsilon) \wedge \varepsilon \neq 1)$
$(e(\varepsilon) \wedge o(\frac{\varepsilon}{2}) \wedge \varepsilon \neq 2)$
$(e(\varepsilon) \wedge e(\frac{\varepsilon}{2}) \wedge o(\frac{\varepsilon}{4}) \wedge \varepsilon \neq 4)$
$(e(\varepsilon) \wedge e(\frac{\varepsilon}{2}) \wedge e(\frac{\varepsilon}{4}) \wedge o(\frac{\varepsilon}{8}) \wedge \varepsilon \neq 8)$
$(e(\varepsilon) \wedge e(\frac{\varepsilon}{2}) \wedge e(\frac{\varepsilon}{4}) \wedge e(\frac{\varepsilon}{8}) \wedge o(\frac{\varepsilon}{16}) \wedge \varepsilon \neq 16)$
$\left(\begin{array}{l} e(\varepsilon) \wedge e(\frac{\varepsilon}{2}) \wedge e(\frac{\varepsilon}{4}) \wedge e(\frac{\varepsilon}{8}) \wedge e(\frac{\varepsilon}{16}) \wedge o(\frac{\varepsilon}{32}) \wedge \varepsilon \neq 32) \vee \\ o(\varepsilon) \wedge e(3\varepsilon+1) \wedge e(\frac{3\varepsilon+1}{2}) \wedge e(\frac{3\varepsilon+1}{4}) \wedge e(\frac{3\varepsilon+1}{8}) \wedge o(\frac{3\varepsilon+1}{16}) \wedge \varepsilon \neq 5 \end{array}\right)$
$\left(\begin{array}{l} (e(\varepsilon) \wedge e(\frac{\varepsilon}{2}) \wedge e(\frac{\varepsilon}{4}) \wedge e(\frac{\varepsilon}{8}) \wedge e(\frac{\varepsilon}{16}) \wedge e(\frac{\varepsilon}{32}) \wedge o(\frac{\varepsilon}{64}) \wedge \varepsilon \neq 64 \vee \\ e(\varepsilon) \wedge o(\frac{\varepsilon}{2}) \wedge e(\frac{3\varepsilon+1}{2}) \cdots \wedge \varepsilon \neq 10 \end{array}\right)$
$\left(\begin{array}{l} e(\varepsilon) \wedge e(\frac{\varepsilon}{2}) \wedge e(\frac{\varepsilon}{4}) \wedge e(\frac{\varepsilon}{8}) \wedge e(\frac{\varepsilon}{16}) \wedge e(\frac{\varepsilon}{32}) \wedge e(\frac{\varepsilon}{64}) \wedge o(\frac{\varepsilon}{128}) \wedge \varepsilon \neq 128 \vee \\ o(\varepsilon) \wedge e(3\varepsilon+1) \wedge e(\frac{3\varepsilon+1}{2}) \cdots \wedge \varepsilon \neq 21 \vee \\ e(\varepsilon) \wedge o(\frac{\varepsilon}{2}) \wedge e(\frac{3\varepsilon+1}{2}) \cdots \wedge \varepsilon \neq 20 \vee \\ o(\varepsilon) \wedge e(3\varepsilon+1) \wedge o(\frac{3\varepsilon+1}{2}) \cdots \wedge \varepsilon \neq 3 \end{array}\right)$
$\ldots$

Poziome linie oddzielają formuły odpowiadające poziomom drzewa Collatza.
Dodatkowy zbiór Z zawiera wszystkie formuły $\varepsilon \neq \mathrm{k}$ takie, że wyrazenie $n = \mathrm{k}$ występuje w formule stopu algorytmu Collatza, k jest liczbą. Zbiór Z zawiera więc zdania

$\{\varepsilon \neq 1, \varepsilon \neq 2, \varepsilon \neq 4, \varepsilon \neq 8, \varepsilon \neq 16, \varepsilon \neq 32, \varepsilon \neq 5, \varepsilon \neq 64, \varepsilon \neq 10, \varepsilon \neq 128, \varepsilon \neq 20, \varepsilon \neq 21, \varepsilon \neq 3, \ldots\}$

Zbiór Y zawiera wszystkie zdania pierwszego rzędu równoważne zdaniom postaci

$$\left\{\begin{array}{l} x := c_x; \\ y := c_y; \\ z := c_z; \\ Err := false \end{array}\right\} \left\{\begin{array}{l} \textbf{if } 3^x + y \neq 2^z \textbf{ then} \\ \quad \textbf{if } (odd(y) \wedge (x = 0 \vee y < 3^{x-1})) \textbf{ then } Err := true \\ \quad \textbf{else} \\ \quad\quad \textbf{if } odd(y) \textbf{ then} \\ \quad\quad\quad x := x - 1;\ y := y - 3^x \\ \quad\quad \textbf{else} \\ \quad\quad\quad y := y \div 2;\ z := z - 1 \\ \quad\quad \textbf{fi} \\ \quad \textbf{fi} \\ \textbf{fi} \end{array}\right\}^i (\neg Err)$$

gdzie $i = 0, 1, 2, \ldots$.
Zdania zbioru Y możemy ustawić w ciąg $Y = \{y_0, y_1, y_2, \ldots\}$.

Zdanie y_i ze zbioru Y wyraża następującą własność: ***jeśli w obliczeniu trójkowym wykonano*** i ***kroków i osiągnięto stan*** $\langle x, y, z\rangle$, ***to jeśli aktualna wartośc trójki*** $\langle x, y, z\rangle$ ***nie reprezentuje jedynkę to można będzie wykonać krok*** $i+1$***-szy. Inaczej mówiąc zdanie*** y_i ***wyklucza ryzyko błędu*** Err ***w*** $i+1$***-szym kroku obliczenia*** .

diagram

$$y_0: \ \neg odd(c_y) \vee (c_x > 0 \wedge c_y > 3^{c_x - 1})$$

$$y_1: \ \left\{\begin{array}{l} x := c_x; \\ y := c_y; \\ z := c_z \end{array}\right\} \left\{\begin{array}{l} \textbf{if } 3^x + y \neq 2^z \textbf{ then} \\ \quad \textbf{if } odd(y) \textbf{ then} \\ \quad\quad x := x - 1;\ y := y - 3^x \\ \quad \textbf{else} \\ \quad\quad y := y \div 2;\ z := z - 1 \\ \quad \textbf{fi} \\ \textbf{fi} \end{array}\right\} (odd(y) \implies (x > 0 \wedge y > 3^{x-1}))$$

Na zbiór Ax' aksjomatów nowej teorii składają się: aksjomaty Ax teorii Presburgera (por. Dodatek A, strona 25), zbiór Df definicji pomocniczych, zdanie U ustalające odpowiedniość elementu ε i trójki $\langle c_x, c_y, c_z\rangle$ oraz nieskończony zbiór Z i nieskończony zbiór zdań Y.

$$Ax' = Ax \cup Df \cup U \cup Z \cup Y$$

Df to skończony zbiór zawierający definicje funkcji $P2$, $P3$, predykatu parzystości,
$U = \{P3(\varepsilon, c_x) + c_y = P2(c_z)\}$.

Udowodnimy, że zbiór Ax' jest niesprzeczny. Zaczniemy od wykazania, że każdy skończony podzbiór Ax_0 zbioru Ax' jest niesprzeczny . Wykażemy mianowicie, że zbiór Ax_0 posiada model w standardowej strukturze $\mathfrak{N}$ liczb naturalnych z dodawaniem. Nasze zadanie ogranicza się do podania właściwej interpretacji czterech stałych: $\varepsilon, c_x, c_y, c_z$.
Zbiór $Ax \cup Df \cup U$ jest niesprzeczny, por. rozdział 4 i dodatki A strona 24 i B strona 27, modelem dla tego zbioru jest standardowa struktura liczb naturalnych z dodawaniem.
Niech Z_0 będzie dowolnym skończonym podzbiorem zbioru Z, podobnie, niech Y_0 będzie dowolnym skończonym podzbiorem zbioru Y.
Zbiór zdań $Ax \cup Df \cup U \cup Z_0 \cup Y_0$ jest niesprzeczny. Aby to wykazać weźmy jako wartość stałej ε jakąś liczbę l większą od każdej liczby k występującej w zbiorze zdań Z_0. Taki wybór zapewnia prawdziwośc każdego zdania ze zbioru Z_0. Z nieskończonego zbioru trójek T_l reprezentujących element l, należy wybrać taką by każde zdanie ze zbioru Y_0 było prawdziwe. Niech j będzie najwiekszą liczbą o

której pewne zdanie ze zbioru Y_0 powiada *jeśli po wykonaniu j iteracji programu trójkowego będzie potrzeba wykonan kolejnej iteracji to aktualne wartości liczb x i y będą wystrczająco duż* . Do osiągnięcia tego celu wystarczy by początkowa wartość zmiennej $x = c_x$ była większa od liczby j. Wystarczy, na przykład, przyjąć $c_x = j + 1 \& c_z = j + 2 \& c_y = 2^{j+2} - l \cdot 3^{j+1}$. W ten sposób zapewniamy prawdziwośc zdania U bo $l \cdot 3^{c_x} + c_y = 2^{c_z}$ i prawdziwość każdego zdania ze zbioru Y_0.

We proved that every finite subset of the set Ax' is consistent. By the compactness theorem on first-order logic[4], we obtain

Lemma 5.1. The set Ax' is a consistentset of formulas.

Now, we apply the model existence theorem, which reads: *for every consistent set S of first-order formula there exists an algebraic structure $\mathfrak{A}$ which is a model of the set S, i.e. for every formula $\sigma \in S$ the formula is valid in the structure $\mathfrak{A}$.*
In this way we proved the following lemma.

Lemma 5.2. There is an algebraic structure $\mathfrak{M}$, such that every sentence of the set Ax' is valid in it.

Corollary 5.1. The execution of Collatz algorithm starting with value of variable n equal ε, $v(n) = \varepsilon$ is infinite.

For the structure $\mathfrak{M}$ is a model of sets Z and Y of sentences.

Corollary 5.2. Element ε of structure $\mathfrak{M}$ does not belong to the Collatz tree DC.

Corollary 5.3. The execution of program IC starting with the triple $\langle c_x, c_y, c_z \rangle$ is infinite .

Proof:

[4]Theorem *If every finite subset of a set S of formulas is consistent, then the set S is consistent too.*

It suffices to show that the computation of the following program

$$\left\{\begin{array}{l} x := c_x;\ y := c_y;\ z := c_z;\ Err := false; \\ \textbf{while } 3^x + y \neq 2^z \textbf{ do} \\ \quad \textbf{if } (odd(y) \wedge (x = 0 \vee y < 3^{x-1})) \textbf{ then } Err := true; exit \\ \quad \textbf{else} \\ \quad\quad \textbf{if } odd(y) \textbf{ then} \\ \quad\quad\quad x := x - 1;\ y := y - 3^x \\ \quad\quad \textbf{else} \\ \quad\quad\quad y := y \div 2;\ z := z - 1 \\ \quad\quad \textbf{fi} \\ \quad \textbf{fi} \\ \textbf{od} \end{array}\right\}.$$

is infinite, i.e. it does not terminate with *Err=true*, nor with $3^x + y = 2^z$. We accept the following denotations: the symbol t denotes the triple $\langle c_x, c_y, c_z\rangle$. and the symbol $\bar{K}$ denotes the following program

$$\bar{K}: \left\{\begin{array}{l} \textbf{if } 3^x + y \neq 2^z \textbf{ then} \\ \quad \textbf{if } (odd(y) \wedge (x = 0 \vee y < 3^{x-1})) \textbf{ then } Err := true \\ \quad \textbf{else} \\ \quad\quad \textbf{if } odd(y) \textbf{ then} \\ \quad\quad\quad x := x - 1;\ y := y - 3^x \\ \quad\quad \textbf{else} \\ \quad\quad\quad y := y \div 2;\ z := z - 1 \\ \quad\quad \textbf{fi} \\ \quad \textbf{fi} \\ \textbf{fi} \end{array}\right\}.$$

We know that the triple t represents the element ε. This is illustrated by the diagram

$$\begin{array}{ccccccccccc} \varepsilon & \xrightarrow{K_{\mathfrak{M}}} & \varepsilon_1 & \xrightarrow{K_{\mathfrak{M}}} & \varepsilon_2 & \xrightarrow{K_{\mathfrak{M}}} & \ldots \varepsilon_k & \xrightarrow{K_{\mathfrak{M}}} & \varepsilon_{k+1} & \xrightarrow{K_{\mathfrak{M}}} & \ldots \\ \uparrow & & & & & & & & & & \\ t & & & & & & & & & & \end{array} \qquad \text{(Dia1)}$$

The upward arrow from t to ε means that the equality $\varepsilon \cdot 3^{c_x} + c_y = 2^{c_z}$ holds.
The sentence y_0 from the set Y is true. Hence there exists the triple $t_1 = \{\bar{K}\}^1_{T_{\mathfrak{M}}}(t)$.

$$\begin{array}{ccccccccccc} \varepsilon & \xrightarrow{K_{\mathfrak{M}}} & \varepsilon_1 & \xrightarrow{K_{\mathfrak{M}}} & \varepsilon_2 & \xrightarrow{K_{\mathfrak{M}}} & \ldots \varepsilon_k & \xrightarrow{K_{\mathfrak{M}}} & \varepsilon_{k+1} & \xrightarrow{K_{\mathfrak{M}}} & \ldots \\ \uparrow & & & & & & & & & & \\ t & \xrightarrow[\succ]{\bar{K}_{T_{\mathfrak{M}}}} & t_1 & & & & & & & & \end{array} \qquad \text{(Dia2)}$$

We can add the arrow from t_1 to ε_1. It is justified by tthe Fact 4.4, page 9.

$$\begin{array}{ccccccccccc} \varepsilon & \xrightarrow{K_{\mathfrak{M}}} & \varepsilon_1 & \xrightarrow{K_{\mathfrak{M}}} & \varepsilon_2 & \xrightarrow{K_{\mathfrak{M}}} & \ldots \varepsilon_k & \xrightarrow{K_{\mathfrak{M}}} & \varepsilon_{k+1} & \xrightarrow{K_{\mathfrak{M}}} & \ldots \\ \uparrow & & \uparrow & & & & & & & & \\ t & \xrightarrow[\succ]{\bar{K}_{T_{\mathfrak{M}}}} & t_1 & & & & & & & & \end{array} \tag{Dia3}$$

By induction, we obtain that *for every natural number* k *the following diagram is comutative*.

$$\begin{array}{ccccccccccc} \varepsilon & \xrightarrow{K_{\mathfrak{M}}} & \varepsilon_1 & \xrightarrow{K_{\mathfrak{M}}} & \varepsilon_2 & \xrightarrow{K_{\mathfrak{M}}} & \ldots \varepsilon_k & \xrightarrow{K_{\mathfrak{M}}} & \varepsilon_{k+1} & \xrightarrow{K_{\mathfrak{M}}} & \ldots \\ \uparrow & & \uparrow & & \uparrow & & \uparrow & & \uparrow & & \\ t & \xrightarrow[\succ]{\bar{K}_{T_{\mathfrak{M}}}} & t_1 & \xrightarrow[\succ]{\bar{K}_{T_{\mathfrak{M}}}} & t_2 & \xrightarrow[\succ]{\bar{K}_{T_{\mathfrak{M}}}} & \ldots t_k & \xrightarrow[\succ]{\bar{K}_{T_{\mathfrak{M}}}} & t_{k+1} & \xrightarrow[\succ]{\bar{K}_{T_{\mathfrak{M}}}} & \ldots \end{array} \tag{Dia}$$

The structure $\mathfrak{M}$ models the set Y of sentences, hence, after k-fold execution of assignment instruction {x:=x-1;y:=y-3^x} one can safely execute the same instruction again, without risk of *Err* error. Moreover, for every natural number i the equality $\varepsilon_i \cdot 3^{x_i} + y_i = 2^{z_i}$ holds, here the elements x_i, y_i, z_i are constituents of the triple t_i. Hence the computation of the program IC in the structure $\mathfrak{M}$ can be prolongued to any desired length.

Fakt 3. Execution of program IC starting with the state $v : \frac{x|y\ |z}{c_x|c_y|c_z}$ is infinite.

Note, the sequence of triples in the lower row of the last diagram is decreasing! and infinite.
By earlier considerations we arrive to the conclusion that the structure $\mathfrak{M}$ contains unreachable elements.

□

From this immediately follows the following lemma.

Lemma 5.3. The structure $\mathfrak{M}$ contains unreachable elements.

We have proved that there exists an algebraic structure $\mathfrak{M}$ with the following properties:

- the structure is a model of Presburger arithmetic Ar,

- the structure contains an element ε such, that the computation of Collatz algorithm starting with ε is infinite,
- the structure contains unreachable elements.

To complete our goal it suffices to show that if a model of Presburger arithmec admits an infinite computation of Collatz algorithm then it contains unreachable elements.

5.2. Infinite computation of Collatz algorithm requires unreachable elements

Dnia 2 Septembri 2021

Do zakończenia dowodu należy wykazać, że jeśli dla elementu ε_0 obliczenie algorytmu Collatza jest nieskończone (krótko: ε_0 jest nie-Collatzowy), to jest nieosiągalny.

Oznaczmy przez c najmniejszy element z taki, że 2^z jest większe od ε_0, $c \stackrel{df}{=} (\mu z)(2^z > \varepsilon_0)$.

Udowodnimy lemat mocniejszy.

Lemma 5.4. *Niech obliczenie Collatza dla elementu ε_0 będzie nieskończone.*

$$\varepsilon_0 \xrightarrow{K_{\mathfrak{M}}} \varepsilon_1 \xrightarrow{K_{\mathfrak{M}}} \varepsilon_2 \xrightarrow{K_{\mathfrak{M}}} \dots \varepsilon_k \xrightarrow{K_{\mathfrak{M}}} \varepsilon_{k+1} \xrightarrow{K_{\mathfrak{M}}} \dots \tag{Dia0s}$$

Wykażemy, że istnieje trójka elementów τ_0, dla której obliczenie algorytmu IC jest także nieskończone, i która reprezentuje element ε_0 .

Wybieramy początkową trójkę $\tau_0 = \langle \varepsilon_0, 2^{2\varepsilon_0+c} - \varepsilon_0 \cdot 3^{\varepsilon_0}, 2\varepsilon_0 + c \rangle$.
Łatwo sprawdzić, że zachodzi równość $\varepsilon_0 \cdot 3_0^{\varepsilon} + (2^{2\varepsilon_0+c} - \varepsilon_0 \cdot 3_0^{\varepsilon}) = 2^{2\varepsilon_0+c}$.
Na poniższym diagramie fakt ten obrazujemy pionową strzałką łączącą τ_0 i ε_0.

$$\begin{array}{ccccccccccc} \varepsilon_0 & \xrightarrow{K_{\mathfrak{M}}} & \varepsilon_1 & \xrightarrow{K_{\mathfrak{M}}} & \varepsilon_2 & \xrightarrow{K_{\mathfrak{M}}} & \dots & \varepsilon_k & \xrightarrow{K_{\mathfrak{M}}} & \varepsilon_{k+1} & \xrightarrow{K_{\mathfrak{M}}} & \dots \\ \uparrow & & & & & & & & & & & \\ \tau_0 & & & & & & & & & & & \end{array} \tag{Dia1s}$$

Wykażemy ponadto, że w każdym kroku iteracji spełnione są następujące warunki:

(1) dla każdej liczby naturalnej $i = 0, 1, 2, \dots$ trójka τ_i reprezentuje element ε_i,

(2) **dla każdej liczby naturalnej** $i = 0, 1, 2, \ldots$ **trójka** τ_{i+1} **jest wynikiem wykonania jednej iteracji w programie** IC, **tj.** $\tau_i \xrightarrow{\bar{K}} \tau_{i+1}$,

(3) **Niech** w **będzie liczbą elementów nieparzystych napotkanych po drodze od elementu** ε_0 **do elementu** ε_k, $k = w + v$. **Liczba** v **napotkanych elementów parzystych jest sumą** $v = \sum_{j=1}^{w} c_j$, **gdzie** c_j **jest liczbą elementów parzystych oddzielających kolejne elementy nieparzyste** $j-1$**-pierwszy i** j**-ty kolejny element nieparzysty lub ostatni element** ε_k. **Trójka** τ_k **jest postaci**

$$\tau_{w+v}: \ \langle \varepsilon_0 - w, \boxed{2^{2\varepsilon_0+c-v} - \frac{3^{\varepsilon_0-w}}{2^v}\left(\varepsilon_0 \cdot 3^w + \sum_{j=1}^{w} 3^{w-j} \cdot 2^{\sum_{l=1}^{j} c_l}\right)}, 2\varepsilon_0 + c - v\rangle.$$

(4) **w obliczeniu algorytmu** IC **nie wystąpi błąd** Err, **tj. dla każdej liczby naturalnej** $i = 0, 1, 2, \ldots$ **po wykonaniu** i**-tej iteracji algorytmu** IC **stan zmiennych** x, y, z **spełnia implikację** $odd(y) \Rightarrow (x > 0 \wedge y > 3^{x-1})$, **czyli dla każdej liczby** i, **zachodzi warunek**

$$(x = \varepsilon_0 \wedge z = 2\varepsilon_0 + c \wedge y = 2^{2\varepsilon_0+c} - \varepsilon_0 \cdot 3^{\varepsilon_0}) \Rightarrow \bigwedge_{j=0}^{i} \{\bar{K}^j\}\big(odd(y) \Rightarrow (x > 0 \wedge y > 3x - 1)\big).$$

Dowód lematu przeprowadzimy przez indukcję ze względu na liczbę i **kroków iteracji . W każdym kroku sprawdzimy warunki (1) –(4).**

Baza. $i = 0$ **Zauważmy, że punkty (1) – (3) są spełnione.**

Fact 5.1. Zachodzi nierównooś $2^{2\varepsilon_0+c} - \varepsilon_0 \cdot 3^{\varepsilon_0} - 3^{\varepsilon_0-1} > 0$.

Jest tak, ponieważ $2^{2\varepsilon_0+c} = 4^{\varepsilon_0-1} \cdot (4 \cdot 2^c)$ **i** $\varepsilon_0 \cdot 3^{\varepsilon_0} - 3^{\varepsilon_0-1} = (3 \cdot \varepsilon_0 + 1) \cdot 3^{\varepsilon_0-1}$.
A więc punkt (4) jest także spełniony. Mamy taki obraz

$$\begin{array}{ccccccccccc}
\varepsilon_0 & \xrightarrow{K_{\mathfrak{M}}} & \varepsilon_1 & \xrightarrow{K_{\mathfrak{M}}} & \varepsilon_2 & \xrightarrow{K_{\mathfrak{M}}} & \ldots \varepsilon_k & \xrightarrow{K_{\mathfrak{M}}} & \varepsilon_{k+1} & \xrightarrow{K_{\mathfrak{M}}} & \ldots \\
\uparrow & & & & & & & & & & \\
\tau_0 & \xrightarrow[(4)]{} & & & & & & & & &
\end{array} \qquad \text{(Dia0)}$$

Strzałka pozioma o początku τ_k, wskazuje, gotowość do wykonania instrukcji K – *bez* ryzyka błędu.

Krok indukcyjny. Zakładamy, że punkty (1) –(4) są spełnione dla $j \leq k$. **co można zilustrować następującym diagramem**

$$\begin{array}{ccccccccccc}
\varepsilon & \xrightarrow{K_{\mathfrak{M}}} & \varepsilon_1 & \xrightarrow{K_{\mathfrak{M}}} & \varepsilon_2 & \xrightarrow{K_{\mathfrak{M}}} & \ldots & \varepsilon_k & \xrightarrow{K_{\mathfrak{M}}} & \varepsilon_{k+1} & \xrightarrow{K_{\mathfrak{M}}} & \ldots \\
\uparrow & & \uparrow & & \uparrow & & & \uparrow & & & & \\
\tau & \xrightarrow{\bar{K}_{\mathfrak{M}}} & \tau_1 & \xrightarrow{\bar{K}_{\mathfrak{M}}} & \tau_2 & \xrightarrow{\bar{K}_{\mathfrak{M}}} & \ldots & \tau_k & \xrightarrow[(4)]{} & & &
\end{array} \qquad \text{(Diak)}$$

Trzeba wykazać, że powyższy diagram można uzupełnić trójką τ_{k+1} i, że punkty (1)–(4) spełnione są także dla $k+1$.

$$\begin{array}{ccccccccccc} \varepsilon & \xrightarrow{K_{\mathfrak{M}}} & \varepsilon_1 & \xrightarrow{K_{\mathfrak{M}}} & \varepsilon_2 & \xrightarrow{K_{\mathfrak{M}}} & \dots\ \varepsilon_k & \xrightarrow{K_{\mathfrak{M}}} & \varepsilon_{k+1} & \xrightarrow{K_{\mathfrak{M}}} & \dots \\ \uparrow & & \uparrow & & \uparrow & & \dots\ \uparrow & & \uparrow & & \\ \tau & \xrightarrow{\bar{K}_{\mathfrak{M}}} & \tau_1 & \xrightarrow{\bar{K}_{\mathfrak{M}}} & \tau_2 & \xrightarrow{\bar{K}_{\mathfrak{M}}} & \dots\ \tau_k & \xrightarrow{\bar{K}_{\mathfrak{M}}} & \tau_{k+1} & \xrightarrow[(4)]{} & \dots \end{array} \qquad \text{(Diak+1)}$$

Należy rozpatrzyć dwa przypadki:
A) element ε_k jest nieparzysty, lub B) element ε_k jest parzysty.
przypadek A)
Drugi element trójki τ_k jest nieparzysty. Z założenia indukcyjnego (4) wykonanie odejmowań y:=y-3^{x-1} oraz x:=x-1 nie prowadzi do błędu..
Trójka $\tau_{k+1} = K(\tau_k)$ jest postaci

$$\tau_{(w+1)+v}: \ \langle \varepsilon_0 - (w+1), \boxed{2^{2\varepsilon_0+c-v} - \frac{3^{\varepsilon_0-(w+1)}}{2^v}\Big(\varepsilon_0 \cdot 3^{w+1} + \sum_{j=1}^{w+1} 3^{w+1-j} \cdot 2^{\sum_{l=1}^{j} c_l}\Big)}, 2\varepsilon_0 + c - v\rangle.$$

Ponadto $c_{w+1} = 0$.
Spełnione są punkty (1) – (3). Sprawdzenie, że zachodzi nierówność $2^{2\varepsilon_0+c-v} - \frac{3^{\varepsilon_0-(w+1)}}{2^v}\big(\varepsilon_0 \cdot 3^{w+1} + \sum_{j=1}^{w+1} 3^{w+1-j} \cdot 2^{\sum_{l=1}^{j} c_l}\big) > 0$ przebiega podobnie jak wyżej.
Zachodzi bowiem równość $2^{2\varepsilon_0+c-v} = \frac{4^{\varepsilon_0-(w+1)}}{2^v} \cdot (4^{w+1} \cdot 2^c)$ oraz nierówności $\frac{4^{\varepsilon_0-(w+1)}}{2^v} > \frac{3^{\varepsilon_0-(w+1)}}{2^v}$ i także $4^{w+1} \cdot 2^c > \big(\varepsilon_0 \cdot 3^{w+1} + \sum_{j=1}^{w+1} 3^{w+1-j} \cdot 2^{\sum_{l=1}^{j} c_l}\big)$
Skąd łatwo otrzymujemy punkt(4).
przypadek B)
W tym przypadku liczba c_w zwiększa się o 1. Trójka τ_{k+1} ma postać

$$\tau_{w+(v+1)}: \ \langle \varepsilon_0 - w, \boxed{2^{2\varepsilon_0+c-(v+1)} - \frac{3^{\varepsilon_0-w}}{2^{v+1}}\left(\varepsilon_0 \cdot 3^w + \sum_{j=1}^{w} 3^{w-j} \cdot 2^{\sum_{l=1}^{j} c_l}\right)}, 2\varepsilon_0 + c - (v+1)\rangle.$$

Sprawdzenie punktów (1)–(4) nie nastęcza trudności. ——————-
W ten sposób zakończyliśmy dowód lematu.
Z powyższych rozważań wynika twierdzenie Collatz.

6. Podsumowanie

Nietrudno zauważyć, że przedstawiony dowód jest okrężny. Nie potrafimy, na razie, przedstawić dowodu wyprowadzonego wprost w

algorytmicznej teorii $\mathcal{ATN}$ z aksjomatów tej teorii lub np. z prawa Archimedesa, które jest twierdzeniem teorii $\mathcal{ATN}$.
Kolejne zadanie to oszacowanie kosztu algorytmu Collatza, wiemy, że obliczenia są skończone, ale nie potrafimy oszacować ich długości. Natomiast koszt odpowiedzi na pytanie czy liczba n jest elementem drzewa Collatza jest stały $O(1)$. Z twierdzenia Collatza potrafimy jednak wyprowadzić kilka ciekaych wniosków, o czym napiszemy osobno.

Podziękowania

Andrzej Szałas znalazł błąd w innej, wcześniejszej pracy na temat algorytmu Collatza. Wiktor Dańko zwrócił naszą uwagę na lukę w analizie obliczeń trójkowych. Ludwik Czaja i Marek Warpechowski przekazali nam swe wątpliwości, dzięki nim powstała obecna wersja pracy.
Wszystkie usterki i błędy są naszego autorstwa.

7. Dodatek A – Kilka faktów o elementarnej teorii dodawania

W rozdziale 4 zaobserwowaliśmy parę pożytecznych faktów o trójkach reprezentujących liczby naturalne.
Rozważać będziemy następującą teorię T_+, por. [Grz71] str. 239 i następne.

Definicja 1. Teoria $T_+ = \langle \mathcal{L}, \mathcal{C}, Ax \rangle$ jest układem trzech przedmiotów:

- $\mathcal{L}$ jest językiem pierwszego rzędu. Na alfabet tego języka składaja się: zbiór V zmiennych, znaki operacji: $0, S, +$, znak relacji równości $=$, znaki funktorów logicznych i kwanrtfikatorów, znaki pomocnicze, m. in nawiasy..
  Zbiór wyrażeń poprawnie zbudowanych to suma teoriomnogościowa zbioru termów T i zbioru formuł F.
  Zbiór termów T jest to najmniejszy zbiór napisów zawierający zbiórzmiennych V i napis 0 i zamknięty ze względu na reguły: jeśli dwa napisy τ_1 oraz τ_2 są termami to termem jest też napis postaci $(\tau_1 + \tau_2)$, jeśli napis τ jest termem to napis $S(\tau)$ jest także termem.

Zbiór formuł jest najmniejszym zbiorem napisów zawierającym równośći tj. napisy postaci $(\tau_1 = \tau_2)$ i zamkniętym ze względu na reguły: jeśli napisy α oraz β są formułami to formułami są tez napisy postaci

$$(\alpha \vee \beta),\ (\alpha \wedge \beta),\ (\alpha \implies \beta),\ \neg\alpha$$

formułami sąteż napisy postaci

$$\forall_x \alpha,\ \exists_x \alpha$$

gdzie x jest zmienną, a α jest formuła.

$\mathcal{C}$ jest operacją konsekwencji zdeterminowaną przez przyjęcie aksjomatów logiki pierwszego rzędu (rachunku predykatów) i reguł wnioskowania logiki pierwszego rzędu

Ax jest zbiorem formuł wyliczonych poniżej.

$$\forall_x\ x+1 \neq 0 \tag{a}$$
$$\forall_x \forall_y\ x+1 = y+1 \implies x = y \tag{b}$$
$$\forall_x\ x+0 = x \tag{c}$$
$$\forall_{x,y}\ (y+1)+x = (y+x)+1 \tag{d}$$
$$\{\Phi(0) \wedge \forall_x [\Phi(x) \implies \Phi(x+1)] \implies \forall_x \Phi(x) \tag{I}$$
(16)

Tu napis $\Phi(x)$ należy zastąpić jakąkolwiek formułą. Ostatni wiersz jest więc schematem indukcji.
Do tego zbioru dodajemy aksjomaty definiujące dodatkowe pojęcia

$$Parz(x) \stackrel{df}{\equiv} \exists_y\, x = y+y \tag{p}$$
$$x\, div\, 2 = y \equiv (x = y+y \vee x = y+y+1) \tag{D2}$$
$$3x \stackrel{df}{=} x+x+x \tag{3x}$$

W teorii T_+ przeprowadzimy dowody paru faktów znanych z rozdziału 4. Dzięki temu upewnimy się, że fakty te prawdziwe ą także w każdym modelu teorii . Natomiast w rozdziale 5 posłużymy się teorią Presburgera

Definicja 2. Teoria $Ar = \langle \mathcal{L}, \mathcal{C}, Ax \rangle$ **jest układem trzech przedmiotów:**

$\mathcal{L}$ **jest językiem pierwszego rzędu. Na alfabet tego języka składaja się: zbiór** V **zmiennych, znaki operacji:** $0, +$**, znak relacji równości** $=$**.**
Zbiór wyrażeń poprawnie zbudowanych to unia zbioru termów T **i zbioru formuł** F**. Zbiór termów** T **jest to najmniejszy zbiór napisów zawierający zbiórzmiennych** V **i napis** 0 **i zamknięty ze względu na reguły: jeśli dwa napisy** τ_1 **oraz** τ_2 **są termami to termem jest też napis postaci** $(\tau_1 + \tau_2)$**, jeśli napis** τ **jesttermem to napis** $S(\tau)$ **jest także termem.**

$\mathcal{C}$ **jest operacją konsekwencji zdeterminowaną przez przyjęcie aksjomatów rachunku predykatów i reguł wnioskowania logiki pierwszego rzędu**

Ax **jest zbiorem formuł wyliczonych poniżej.**

$$\forall_x \ x + 1 \neq 0 \tag{A}$$
$$\forall_x \ x \neq 0 \implies \exists_y x = y + 1 \tag{B}$$
$$\forall_{x,y} \ x + y = y + x \tag{C}$$
$$\forall_{x,y,z} \ x + (y + z) = (x + y) + z \tag{D}$$
$$\forall_{x,y,z} \ x + z = y + z \implies x = y \tag{E}$$
$$\forall_x \ x + 0 = x \tag{F}$$
$$\forall_{x,z} \ \exists_y \ (x = y + z \vee z = y + x) \tag{G}$$
$$\forall_x \exists_y \ (x = y + y \vee x = y + y + 1) \tag{H2}$$
$$\forall_x \exists_y \ (x = y + y + y \vee x = y + y + y + 1 \vee x = y + y + y + 1 + 1) \tag{H3}$$

. . . . . .

$$\forall_x \exists_y \left(\begin{array}{l} x = \underbrace{y + y + \cdots + y}_{k} \vee \\ \quad x = \underbrace{y + y + \cdots + y}_{k} + 1 \vee \\ \quad\quad x = \underbrace{y + y + \cdots + y}_{k} + \underbrace{1 + 1}_{2} \vee \\ \quad\quad \cdots \\ \quad\quad\quad x = \underbrace{y + y + \cdots + y}_{k} + \underbrace{1 + 1 + \cdots + 1}_{k-2} \vee \\ \quad\quad\quad\quad x = \underbrace{y + y + \cdots + y}_{k} + \underbrace{1 + 1 + \cdots + 1}_{k-1} \end{array} \right) \tag{Hk}$$

. . .

Przypomnijmy parę faktów
F1. Teoria T_+ jest elementarnie równoważna teorii Ar.[Pre29, Sta84]
F2. Teoria Ar jest rozstrzygalna. [Pre29].
F3. Złożoność teorii Ar, czyli koszt udowodnienia, że dane zdanie α jest twierdzeniem (lub jego negacją) jest rzędu podwójnie wykładniczego $O(2^{2^n})$. [FR79].
F4. Teorie T_+ oraz Ar mają modele niestandardowe, tj. modele zawierające elementy nieosiągalne, zob Dodatek C, strona 28.

8. Dodatek B– dowód twierdzenia 4.1

W tym rozdziale wykażemy, że zdanie *dla każdego* n *istnieją* x, y, z *takie, że* $n \cdot 3^x + y - 2^z$ jest twierdzeniem teorii T_+ dodawania. Elementarna teoria $\mathcal{T}_+$ dodawania liczb naturalnych została przypomniana w dodatku A.
Działania mnożenia i potęgowania są niedostępne w teorii T_+. Ale nie są niezbędne do osiągnięcia naszego celu.
Teorię T_+ wzbogacamy o dwie funkcje $P2(\cdot)$ oraz $P3(\cdot,\cdot)$. zdefiniowane w ten sposób

$$\begin{array}{l|l} P2(0) = 1 & P3(y,0) = y \\ P2(x+1) = P2(x) + P2(x) & P3(y,x+1) = P3(y,x) + P3(y,x) + P3(y,x) \end{array}$$

Lemat 4. Powyższe definicje są poprawne, tzn. twierdzeniami teorii T_+ wzbogaconej w ten sposób są zdania $\forall_x \exists_y P2(x) = y$ i $\forall_{x,y,z} P2(x) = y \wedge P2(x) = z \implies y = z$.

$$T_+ \vdash \forall_x \exists_y P2(x) = y \quad \text{i}$$

$$T_+ \vdash \forall_{x,y,z} P2(x) = y \wedge P2(x) = z \implies y = z.$$

Podobnie twierdzeniami wzbogaconej teorii T_+ są zdania $\forall_{y,x} \exists_z P3(y,x) = z$ i $\forall_{y,x,z,u} P3(y,x) = z \wedge P3(y,x) = u \implies z = u$.

Dowód przebiega przez indukcję względem zmiennej x.

Potrzebna nam będzie następująca definicja relacji mniejszości $a < b \stackrel{df}{=} \exists_c\, a + S(c) = b$. Wykorzystując definicje funkcji $P2$ oraz $P3$ napiszemy wyrazenie $P3(n,x) + y = P2(z)$.

Lemma 8.1. Następujące zdanie jest twierdzeniem wzbogaconej teorii T_+

$$\forall_n \exists_{x,y,z} P3(n,x) + y = P2(z)$$

Najpierw przez indukcję dowodzimy, że $T_+ \vdash \forall_n\, n < 2^n$. **A dokładniej,** $T_+ \vdash \forall_n\ n < P2(n)$. **Łatwo sprawdzić, że** $T_+ \vdash 0 < P2(0)$. **Załóżmy, że** $T_+ \vdash \forall_n\{n < P2(n)\}$. **Nierówność** $n + 1 < P2(n + 1)$ **wynika z dwu nierówności** $T_+ \vdash n < P2(n)$ **i** $\mathcal{T} \vdash 1 < P2(n)$.

W podobny sposób uzyskamy $T_+ \vdash P3(n,x) < P2(z) \wedge (z = n + x + x)$
Wynika stąd, że $T_+ \vdash \forall_n \exists_{x,y,z}\ P3(n,x) + y = P2(z)$.
Właściwie wykazaliśmy, że $\forall_{n,x} \exists_{y,z} P3(n,x) + y = P2(z)$

Lemma 8.2. Niech $\mathfrak{M}$ **będzie jakimkolwiek modelem arytmetyki Presburgera. Element** n **jest osiągalny wtedy i tylko wtedy gdy istnieje taka trójka reprezentująca element** n**, tj. taka, że zachodzi równość** $P3(n,x) + y = P2(z)$ **czyli** $n \cdot 3^x + y = 2^z$ **i liczby** x, y, z **są osiągalne.**

Proof:
Jeśli spełnione są formuły
$\{q := 0;\ while\ q \neq x\ do\ q := q + 1\ od\}(x = q)$,
$\{q := 0;\ while\ q \neq y\ do\ q := q + 1\ od\}(y = q)$,
$\{q := 0;\ while\ q \neq z\ do\ q := q + 1\ od\}(z = q)$
i ponadto zachodzi równość
$P3(n,x) + y = P2(z)$ **to łatwo sprawdzic,że spełniona jest formuła** $\{t := 0;\ while\ n \neq t\ do\ t := t + 1\ od\}(t = n)$ □

9. Appendix C - an example of an infinite computation

At this point, we'll remind you of a few facts that are less known to the IT community. In Appendix A we described the Ar **theory of addition of natural numbers. The only functor in the language of this theory is** $+$**, we also have two constants 0 and 1 and the predicate of equality** $=$**. Now, we will program the algebraic structure** $\mathfrak{M}$**, which is a model of this theory, i.e. all axioms of theory** Ar **are true in the structure** $\mathfrak{M}$**. First we will describe this structure as mathematicians do, then we will write a class (ie a program module) implementing this structure. medskip**

9.1. Mathematical description of the structure

$\mathfrak{M}$ is an algebraic structure

$$\mathfrak{M} = \langle M; \mathbf{0}, \mathbf{1}, \oplus; = \rangle \qquad \text{(NonStandard)}$$

such that M is a set of pairs $\langle k, w \rangle$ where element $k \in Z$ is an integer, element w is a rational, non-negative number and the following requirements are lsatisfied:

(*i*) for each element $\langle k, w \rangle$ if $w = 0$ then $k \geq 0$,

(*ii*) the meaning of the constant $\mathbf{0}$ is $\langle 0.0 \rangle$,

(*iii*) the meaning of constant $\mathbf{1}$ is $\langle 1.0 \rangle$,

(*iv*) the operation $\oplus$ of addition is determined as follows

$$\langle k, w \rangle \oplus \langle k', w' \rangle \stackrel{df}{=} \langle k + k', w + w' \rangle.$$

Lemma 9.1. The algebraic structure $\mathfrak{M}$ is a model of Ar theory.

The reader will check that each axiom of the Ar theory is a sentence true in the structure $\mathfrak{M}$.
The structure $\mathfrak{M}$ is not a model of the $\mathcal{ATN}$, algorithmic theory of natural numbers. Elements of the structure $\langle k, w \rangle$. such as $w \neq 0$ are *unreachable*. i.e. for each element $x_0 = \langle k, w \rangle$ such that $w \neq 0$ the following condition holds

$$\neg\{y := 0; \textbf{while } y \neq x_0 \textbf{ do } y := y + 1 \textbf{ od}\}(y = x_0)$$

The subset $\mathfrak{N} \subset \mathfrak{M}$ composed of only those elements for which $w = 0$ is a model of the theory $\mathcal{ATN}$. The elements of the structure $\mathfrak{N}$ are called *reachable*. A very important theorem of the foundations of mathematics is

Fact 9.1. The structures $\mathfrak{N}$ and $\mathfrak{M}$ are not isomorphic. See [Grz71], p. 256.

As we will see in a moment, this fact is also important for IT specialists.

9.2. Definition in programming language

Perhaps you have already noticed that the $\mathfrak{M}$ is computable. The following is a class that implements the structure $\mathfrak{M}$. The implementation uses the integer type, we do not introduce rationalNumbers explicitly.

```
unit StrukturaM: class;
  unit Elm: class(k,li,mia: integer);
  begin
    if mia=0 then raise Error fi;
    if li * mia <0 then raise Error fi;
    if li=0 and k<0 then raise Error fi;
  end Elm;
  add: function(x,y:Elm): Elm;
  begin
    result := new Elm(x.k+y.k, x.li*y.mia+x.mia*y.li, x.mia*y.mia )
  end add;
  unit one : function:Elm;  begin   result:= new Elm(1,0,2)  end one;
  unit zero : function:Elm;  begin   result:= new Elm(0,0,2)  end zero;
  unit eq: function(x,y:Elm): Boolean;
  begin
    result := (x.k=y.k) and (x.li*y.mia=x.mia*y.li )
  end eq;
end StrukturaM
```

The following lemma expresses the correctness of the implementation

Lemma 9.2. The set of Elm objects with the *add* operation is a model of the Ar theory

9.3. Infinite Collatz algorithm computation

How to execute the Collatz algorithm in StructureM? It's easy.

```
pref StrukturaM block
  var n: Elm;
  unit odd: function(x:Elm): Boolean; ... result:=(x.k mod 2)=1 ... end odd;
  unit div2: function(x:elm): Elm; ...
begin
  n:= new Elm(8,1,2);
  while  not eq(n,one)  do
    if  odd(n) then
      n:=add(n,add(n,add(n,one)))  else n:= div2(n)
    fi
  od
end block;
```

Below we present the computation of Collatz algorithm for $n = \langle 8, \frac{1}{2}\rangle$.

$$\langle 8, \frac{1}{2}\rangle, \langle 4, \frac{1}{4}\rangle, \langle 2, \frac{1}{8}\rangle, \langle 1, \frac{1}{16}\rangle, \langle 4, \frac{3}{16}\rangle, \langle 2, \frac{3}{32}\rangle, \langle 1, \frac{3}{64}\rangle, \langle 4, \frac{9}{64}\rangle, \langle 2, \frac{9}{128}\rangle, \dots$$

None of the elements of the above sequence is a standard natural number. Each of them is unreachable. It is worth looking at an example of another calculation. Will something change when we assign n a different object? e.g. n: = new Elm (19,2,10)?

$\langle 19, \frac{10}{2}\rangle, \langle 58, \frac{30}{2}\rangle, \langle 29, \frac{30}{4}\rangle, \langle 88, \frac{90}{4}\rangle, \langle 44, \frac{90}{8}\rangle, \langle 22, \frac{90}{16}\rangle, \langle 11, \frac{90}{32}\rangle, \langle 34, \frac{270}{32}\rangle, \langle 17, \frac{270}{64}\rangle,$
$\langle 52, \frac{810}{64}\rangle, \langle 26, \frac{405}{64}\rangle, \langle 13, \frac{405}{128}\rangle, \langle 40, \frac{1215}{128}\rangle, \langle 20, \frac{1215}{256}\rangle, \langle 10, \frac{1215}{256}\rangle, \langle 5, \frac{1215}{512}\rangle, \langle 16, \frac{3645}{512}\rangle, \langle 8, \frac{3645}{1024}\rangle,$
$\langle 4, \frac{3645}{2048}\rangle, \langle 2, \frac{3645}{4096}\rangle, \langle 1, \frac{3645}{8192}\rangle, \langle 4, \frac{3*3645}{8192}\rangle, \langle 2, \frac{3645*3}{2*8192}\rangle, \langle 1, \frac{3*3645}{4*8192}\rangle, \langle 4, \frac{9*3645}{4*8192}\rangle, \dots$

And one more computation.

$\langle 19, 0\rangle, \langle 58, 0\rangle, \langle 29, 0\rangle, \langle 88, 0\rangle, \langle 44, 0\rangle, \langle 22, 0\rangle, \langle 11, 0\rangle, \langle 34, 0\rangle, \langle 17, 0\rangle, \langle 52, 0\rangle, \langle 26, 0\rangle,$
$\langle 13, 0\rangle, \langle 40, 0\rangle, \langle 20, 0\rangle, \langle 10, 0\rangle, \langle 5, 0\rangle, \langle 16, 0\rangle, \langle 8, 0\rangle, \ \langle 4, 0\rangle, \langle 2, 0\rangle, \langle 1, 0\rangle$

Corollary 9.1. The structure $\mathfrak{M}$, which we have described in two different ways, is the model of the T_+ theory (you can also say that this structure implements the specification given by the axioms of the Ar theory), with the non-obvious presence of unreachable elements in it.

Another observation

Corollary 9.2. The halting property of the Collatz algorithm cannot be proved from the axioms of the T_+ theory, nor from the Ar theory.

The reader may wish to construct the computation that starts with $\langle 8, \frac{1}{7} \rangle$.

References

[FR79] J. Ferrante and C.W. Rackoff. The Computational Complexity of Logical Theories" Springer Verlag, Heidelberg, 1979.

[Grz71] Andrzej Grzegorczyk. Zarys Arytmetyki Teoretycznej. PWN, Warszawa, 1971.

[MS87] Grażyna Mirkowska and Andrzej Salwicki. Algorithmic Logic. http://lem12.uksw.edu.pl/wiki/Algorithmic_Logic, 1987. [Online; accessed 7-August-2017].

[Pre29] Über die Vollständigkeit eines gewissen Systems der Arithmetik ganzer Zahle Warsaw Iniversity, 1929.

[Sta84] Ryan Stansifer. Presburger's Article on Integer Arithmetic: Remarks and Translation. Technical Report TR84-639, Cornell University, 1984.