Dombrova Research

Na twierdzenie Collatza

Krótkie sprawozdanie – 22 lipca 2021

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

Andrzej Salwicki
Dombrova Research
Partyzantów 19
05-092 Łomianki, POLAND
salwicki@gmail.com

Streszczenie. Przedstawiamy dowód twierdzenia Collatza.

Abstract. Collatz conjecture has a proof.
We present a couple of observations. 1. The problem is of algorithmic nature, The conjectute 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 GCl 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 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. Wprowadzenie

Hipoteza sformułowana przez Lothara Collatza w r. 1937 jest problemem algorytmicznym[1]. Należy wykazać, że poniższy program Cl wykonywany w standardowej strukturze $\mathfrak{N}$ liczb naturalnych z dodawaniem, ma dla każdego argumentu $n \neq 0$ obliczenie skończone.
Będziemy badać własność stopu programu Cl.

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

2. Drzewo Collatza

Ten rozdział zostanie rozwinięty.

3. Formuła stopu

Poniższa formuła halt wyraża własność stopu programu Cl. Należy wykazać,że jest ona prawdziwa w strukturze $\mathfrak{N}$ liczb naturalnych z dodawaniem

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

Formuła **LC** stwierdzająca, że istnieje iteracja programu zawartego pomiędzy nawiasami { }, także jest formułą stopu programu Cl.

$$\bigcup \left\{ \begin{array}{l} \mathbf{if}\ n \neq 1\ \mathbf{then} \\ \quad \left[\begin{array}{l} \mathbf{if}\ n\ \underline{\mathrm{mod}}\ 2 = 1 \\ \quad \mathbf{then}\ n \leftarrow 3n+1 \\ \quad \mathbf{else}\ n \leftarrow n\ \underline{\mathrm{div}}\ 2 \\ \mathbf{fi} \end{array} \right] \\ \mathbf{fi} \end{array} \right\} \big(n = 1\big) \tag{\textbf{LC}}$$

Należy wykazać jej prawdziwość w standardowej strukturze $\mathfrak{N}$ liczb naturalnych lub przeprowadzić jej dowód w algorytmicznej teorii liczb naturalnych $\mathcal{ATN}$, ponieważ każdy model tej teorii jest izomorficzny z strukturą $\mathfrak{N}$. Zobacz [MS87].

[1] Może dziwić, że niektórzy umieszczają ten problem w teorii układów dynamicznych.

Teoria $\mathcal{ATN}$ ma trzy aksjomaty $\left\{\begin{array}{l} \forall_x\, S(x) \neq 0 \\ \forall_{x,y}\, S(x) = S(y) \implies x = y \\ \forall_x \{y := 0; \mathbf{while}\ y \neq x\ \mathbf{do}\ y := y+1\ \mathbf{od}\}\,(y = x) \end{array}\right\}$.

Teorię $\mathcal{ATN}$ możemy wzbogacić dodając definicje pożytecznych operacji $+, \cdot 3, \div 2$, oraz predykatu parzystości. Dla oszczędności miejsca przyjmiemy, że napis $e(n)$ oznacza *n jest liczbą parzystą*, zaś napis $o(n)$ czytamy *n jest liczbą nieparzystą.*

Posługując się aksjomatami rachunku programów i aksjomatami teorii $\mathcal{ATN}$ możemy wypisać kilka formuł równoważnych formule stopu. Poniżej prezentujemy jedną z nich

$$\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} \mathbf{if}\ n \neq 1\ \mathbf{then} \\ \quad \left[\begin{array}{l} \mathbf{if}\ n\ \underline{\mathrm{mod}}\ 2 = 1 \\ \quad \mathbf{then}\ n \leftarrow 3n+1 \\ \quad \mathbf{else}\ n \leftarrow n\ \underline{\mathrm{div}}\ 2 \\ \mathbf{fi} \end{array}\right] \\ \mathbf{fi} \end{array}\right\}^8 \bigcup \left\{\begin{array}{l} \mathbf{if}\ n \neq 1\ \mathbf{then} \\ \quad \left[\begin{array}{l} \mathbf{if}\ n\ \underline{\mathrm{mod}}\ 2 = 1 \\ \quad \mathbf{then}\ n \leftarrow 3n+1 \\ \quad \mathbf{else}\ n \leftarrow n\ \underline{\mathrm{div}}\ 2 \\ \mathbf{fi} \end{array}\right] \\ \mathbf{fi} \end{array}\right\} \big(n = 1\big)
\end{array}\right\}$$

Rachunki te można kontynuować. (I tak zrobiliśmy.)
Już teraz wyłaniają się pewne prawidłowości:

- wyraźnie widać odpowiedniość pomiędzy poziomami drzewa Collatza i składnikami formuły stopu,
- na wyższych poziomach drzewa Collatza pojawiają się coraz liczniej liczby nieparzyste – a kolejne składniki formuły stopu są alternatywami coraz dłuższych koniunkcji czynników postaci $o(.)$ lub $e(.)$,

- Zauważ, alternatywę $(n = 128 \vee n = 21 \vee n = 20 \vee n = 3)$ można zapisac tak $(n \cdot 3^0 + 0 = 2^7 \vee n \cdot 3^1 + 1 = 2^6 \vee n \cdot 3^1 + 4 = 2^6 \vee n \cdot 3^2 + 5 = 2^5)$.

W rozdziale 4 postaramy się wykorzystać tę obserwację.

4. Trójki

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

Fakt 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 $\langle x, y, z \rangle \asymp n$.

Dowód:
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 $y = 2^z - k$. □

Przykład 2. 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.

Twierdzenie 3. 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

Definicja 1. *Dwie trójki $\langle x, y, z\rangle$ i $\langle u, v, t\rangle$ są <u>równoważne</u> gdy reprezentują tę samą liczbę naturalną*

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

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

Definicja 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 1 wynika, że dla kazdej liczby naturalnej n można wskazać kodującą ją trójkę, która ma liczbę y większą od dowolnie wybranej liczby naturalnej k.

Definicja 3. *Nieparzystość* trójek wyznaczona jest przez nieparzystość liczby y

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

Kolejne spostrzeżenie

Fakt 4. *Liczba n jest nieparzysta wttw gdy reprezentująca ją trójka $\langle x, y, z\rangle$ jest nieparzysta.*

Definicja 4. Jedynka

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

Definicja 5. Klasa trójek równowaznych jest zbiorem trójek reprezentujących tę samą liczbę n.

Jedynka jest reprezentowana przez wiele trójek, np. $\langle 0, 0, 0\rangle$, $\langle 0, 1, 1\rangle$, $\langle 1, 1, 2\rangle$, $\langle 1, 5, 3\rangle$, $\langle 2, 7, 4\rangle$, ...

Na trójkach możemy określić dwie operacje

Definicja 6. Operacja $div2$ przeprowadza trójkę parzystą $\langle x, y, z\rangle$ w trójkę $\langle x, y \div 2, z-1\rangle$

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

Definicja 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

Fakt 5. *Trójka $\langle x, y, z\rangle$ reprezentuje liczbę n wttw gdy trójka $\langle x, y \div 2, z-1\rangle$ reprezentuje liczbę $n \div 2$.*
Ponadto $\langle x, y \div 2, z-1\rangle \prec \langle x, y, z\rangle$.
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

Lemat 6. *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 $\langle x, y, z\rangle$.*

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

Dowód:
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.

Fakt 7. *Działanie na trójkach prowadzi od trójki kodującej liczbę n do kolejnej trójki, która reprezentuje wynik m następującej instrukcji warunkowej K :* $\{$if $odd(n)$ then $n := 3*n+1$ else $n := ndiv2$ fi$\}$.

$$\begin{tikzcd}
n \arrow[r, "\{\textbf{if } odd(n) \textbf{ then } m:=3n+1 \textbf{ else } m:=n/2 \textbf{ fi}\}_{\mathfrak{N}}"] & m \\
\langle x, y, z\rangle \arrow[u, "n\cdot 3^x+y=2^z"] \arrow[r, "\{\textbf{if } odd(y) \textbf{ then } u,v,t:=x-1,y-3^{x-1},z \textbf{ else } u,v,t:=x,y/2,z-1 \textbf{ fi}\}_{\mathfrak{T}}"] & \langle u, v, t\rangle \arrow[u, "m\cdot 3^u+v=2^t"']
\end{tikzcd}$$

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 $y > 3^{x-1}$). Kłopot pojawiający się gdy y jest nieparzyste i nie można zmniejszyć x ani y jest łatwy do usunięcia. Wystarczy zamiast trójki $\langle x, y, z\rangle$ wziąć inną trójkę równoważną i z wartością x większą. Przemienność tego diagramu pozwala nam dostrzec, że każdemu obliczeniu algorytmu Collatza w strukturze $\mathfrak{N}$ liczb naturalnych, odpowiada obliczenie algorytmu sprzężonego w strukturze trójek $\mathfrak{T}$.

Przemienność następującego diagramu

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

nie jest oczywista. Trudniej tez jest sformułować warunek przemienności. Może tak

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

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 $o_1, o_2, o_3, o_4, o_5, o_6$

$$\begin{aligned}
o_1(\langle x,y,z\rangle) &= \langle x+1, 3y+2^z, z+2\rangle && (1)\\
o_2(\langle x,y,z\rangle) &= \langle x+1, 3y-2^z, z+1\rangle \quad \text{gdy } 3y > 2^z && (2)\\
o_3(\langle x,y,z\rangle) &= \langle x, y+2^z, z+1\rangle && (3)\\
o_4(\langle x,y,z\rangle) &= \langle x, y-2^{z-1}, z-1\rangle \qquad \text{gdy } y > 2^{z-1} && (4)\\
o_5(\langle x,y,z\rangle) &= \langle x-1, \frac{y-2^{z-2}}{3}, z-2\rangle \ \text{gdy } y > 2^{z-2} \text{ i } (y-2^{z-2})\mathrm{mod}3 = 0 && (5)\\
o_6(\langle x,y,z\rangle) &= \langle x-1, \frac{y+2^{z-1}}{3}, z-1\rangle && (6)
\end{aligned}$$

Wynika stąd, że zbiór T_n jest nieskończony.

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 tez 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.

$$IC : \left\{ \begin{array}{l} \mathbf{while}\ 3^x + y \neq 2^z\ (*\ tj.\ n \neq 1\ *)\ \mathbf{do} \\ \qquad \mathbf{if}\ \text{y } \textit{nieparzyste} \text{ and } ((x = 0) \text{ or } (y < 3^{x-1}))\ \mathbf{then}\ Err := true;\ \mathbf{exit\ fi}; \\ \qquad \mathbf{if}\ y\ parzyste\ \mathbf{then}\ z := z - 1;\ y := y \,\mathrm{div}\, 2;\ (*\, n := \frac{n}{2}\ *) \\ \qquad \mathbf{else}\ x := x - 1;\ y := y - 3^x;\ (*n := 3 * n + 1\ *)\mathbf{fi} \\ \mathbf{od} \end{array} \right\}$$

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

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

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

4.4. Własności programu IC

Fakt 8. *Każde obliczenie programu IC jest skończone i albo jest sukces i osiągnięto jedynkę albo jest błąd i obliczenia algorytmu IC nie można kontynuować.*

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

5. Zbiory W_x

6. Hotel Collatza

7. Dowód twierdzenia Collatza

Nasz plan

(*i*) Wykazujemy, że zdanie $\forall_n \exists_{x,y,z}\, n \cdot 3^x + y = 2^z$ jest twierdzeniem elementarnej teorii Ar dodawania liczb naturalnych (teoria Presburgera). Por. Dodatek B 17. A więc zdanie to jest prawdziwe w każdym modelu tej teorii.

(*ii*) W Dodatku C 17, pokazaliśmy nieskończone obliczenie algorytmu Collatza w niestandardowym, obliczalnym i programowalnym, modelu teorii Ar.

(*iii*) Wykazujemy, że istnieje model $\mathfrak{M}$ teorii Ar w którym istnieje obliczenie nieskończone. *Nie zakładamy* przy tym, że model zawiera elementy nieosiagalne.

(*iv*) Wykazujemy, że w każdym modelu teorii Ar, jeśli dla pewnego elementu n obliczenie algorytmu Collatza jest nieskończone, to model ten zawiera elementy nieosiagalne.

(*v*) Wnioskujemy stąd, że jeśli model nie ma elementów nieosiągalnych to nie istnieją+ obliczenia nieskończone.

7.1. Struktura w której program Collatza ma obliczenie nieskończone

Konstruujemy rozszerzenie teorii Ar (arytmetyki Presburgera) w taki sposób by wykazać istnienie struktury w której istnieją obliczenia niesko ńczone algorytmu Collatza. Modyfikacja polega na tym, że 1° do alfabetu dodajemy cztery nowe stałe $\varepsilon, c_x, c_y, c_z$ i odpowiednio modyfikujemy zbiór formuł, oraz 2° dodajemy dwa nieskończone zbiory formuł Z i Y. Jak tworzymy zbiór Z?
Naszym zamiarem jest wykazanie, że

1. Przypuszczenie, że dla pewnego elementu ε oblicenie Collatza jest nieskończone nie prowadzi do sprzeczności, i

2. każda struktuta algebraiczna $\mathfrak{M}$ w której takie obliczenie istnieje, nie jest standardowym modelem liczb naturalnych.

Naturalnym odruchem chcielibyśmy dodać negację formułý stopu dla $n = \varepsilon$. Ale formuła ta

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

nie należy do języka pierwszego rzędu. Podobnie nie należy do języka pierwszego rzędu inna formuła wyrażająca tę sama własność

$$\{n \leftarrow \varepsilon\} \bigcap \left\{ \begin{array}{l} \textbf{if } n \neq 1 \textbf{ then} \\ \quad \textbf{if } Parz(n) \\ \quad \textbf{then } n \leftarrow n \textbf{ div } 2 \\ \quad \textbf{else } n \leftarrow 3n+1 \\ \quad \textbf{fi} \\ \textbf{fi} \end{array} \right\} (n \neq 1) \tag{7}$$

Natomiast, w każdej strukturze $\mathfrak{M}$ powyższa formuła (7) jest prawdziwa wtedy i tylko wtedy gdy prawdziwe są w tej strukturze wszystkie formuły następującej postaci

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

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

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

Kontynuując otrzymamy m.in. następujące formuły

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

Poziome linie oddzielają formuły odpowiadające poziomom drzewa Collatza.
Dodatkowy zbiór aksjomatów Z składa się z wszystkich takich formuł.

Zbiór Y zawiera wszystkie zdania pierwszego rzędu równoważne zdaniom postaci

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

gdzie $i = 0, 1, 2, \ldots$.

Zdania ze zbioru Y mówią: *jeśli w obliczeniu trójkowym wykonano i kroków, to można będzie wykonać krok $i+$-szy, jesli zajdzie taka potrzeba.*

Na zbiór Ax' aksjomatów nowej teorii składają się: aksjomaty Ax teorii Presburgera (por. Dodatek A, strona 14), zbiór Z, definicje funkcji $P2$ i $P3$(por. Dodatek B, strona 17), zdanie $P3(\varepsilon, c_x) + c_y = P2(c_z)$, oraz zbiór zdań Y.

Nietrudno zauważyć, że zbiór Ax' jest niesprzeczny. Niesprzeczny bowiem jest każdy skończony podzbiór zbioru Ax'. Dla każdego skończonego podzbioru S_0 zbioru Ax', $S_0 \subset Ax'$, wybierzemy jako stałą ε liczbę większą od jakiejkolwiek liczby występującej w zbiorze S_0. Posłużymy się twierdzeniem o zwartości logiki pierwszego rzędu. Głosi ono: *jeśli każdy skończony podzbiór pewnego zbioru formuł S jest niesprzeczny , to niesprzeczny jest cały zbiór S.* Wykorzystując twierdzenie o istnieniu modelu i uzyskujemy

Twierdzenie 9. Istnieje struktura algebraiczna $\mathfrak{M}$, w której każda formuła ze zbioru Ax' jest prawdziwa.

Wniosek 1. Obliczenie programu Collatza Cl w strukturze $\mathfrak{M}$ dla wrtościowania początkowego $v(n) = \varepsilon$ można dowolnie przedłużać, tj. jest ono nieskończone.

Niech ciąg
$\{\varepsilon, \varepsilon_1, \varepsilon_2, \ldots, \varepsilon_k, \ldots\}$ będzie tym obliczeniem.

Wniosek 2. Struktura $\mathfrak{M}$ zawiera nieskończony zbiór P taki, że każdy element $e \in P$ jest początkiem nieskończonego obliczenia Collatza. Zbiór P zawiera każdy element z nieskończonego ciągu

$$\{\varepsilon, \varepsilon_1, \varepsilon_2, \ldots, \varepsilon_k, \ldots\} \subset P$$

i ponadto spełnia następujące warunki,

$$\begin{aligned}
x \in P &\implies (x+x) \in P \\
x \in P \wedge \exists_y x = y + y &\implies y \in P \\
x \in P \wedge \exists_y x = y + y + 1 &\implies (x+x+x+1) \in P \\
x \in P \wedge (\exists_e \exists_z e = z + z + 1 \wedge x = e + e + e + 1) &\implies e \in P
\end{aligned}$$

.

Fakt 10. Zbiór P i drzewo Collatza są zbiorami rozłącznymi.

———————————-

7.2. Nieskończone obliczenie trójkowe

Struktura $\mathfrak{M}$ jest modelem zbioru zdań Y i zdania $P3(\varepsilon, c_x) + c_y = P2(c_z)$. Każdy początkowy segment następującego diagramu nieskończonego jest przemienny.

Przyjęte oznaczenia:
K: **if** n≠1 **then if** n mod 2=0 **then** n:=n div 2 **else** n:=3n+1 **fi fi**

$$\bar{K} : \left\{ \begin{array}{l}
\textbf{if } 3^x + y \neq 2^z \textbf{ then} \\
\quad \textbf{if} y \div 2 = 0 \\
\quad \textbf{then } y := y\ \underline{div} 2; z := z - 1 \\
\quad \textbf{else} y := y - 3^{x-1}; x := x - 1 \\
\quad \textbf{fi} \\
\textbf{fi}.
\end{array} \right\}$$

t jest trójką $\langle c_x, c_y, c_z \rangle$

Przyjęliśmy także $\varepsilon_i = \{K\}^i_{\mathfrak{M}}(\varepsilon)$ oraz $t_i = \{\bar{K}\}^i_{T_{\mathfrak{M}}}(t)$ dla $i = 1, 2, \ldots$.

$$\begin{tikzcd}
\varepsilon \arrow[r, "K_{\mathfrak{M}}"] & \varepsilon_1 \arrow[r, "K_{\mathfrak{M}}"] & \varepsilon_2 \arrow[r, "K_{\mathfrak{M}}"] & \ldots \varepsilon_k \arrow[r, "K_{\mathfrak{M}}"] & \varepsilon_{k+1} \arrow[r, "K_{\mathfrak{M}}"] & \ldots \\
t \arrow[u] \arrow[r, "\bar{K}_{T_{\mathfrak{M}}}", "\succ"'] & t_1 \arrow[u] \arrow[r, "\bar{K}_{T_{\mathfrak{M}}}", "\succ"'] & t_2 \arrow[u] \arrow[r, "\bar{K}_{T_{\mathfrak{M}}}", "\succ"'] & \ldots t_k \arrow[u] \arrow[r, "\bar{K}_{T_{\mathfrak{M}}}", "\succ"'] & t_{k+1} \arrow[u] \arrow[r, "\bar{K}_{T_{\mathfrak{M}}}", "\succ"'] & \ldots
\end{tikzcd} \tag{Dia}$$

Wynika stąd

Fakt 11. Obliczenie trójkowe rozpoczynające się od stanu $v : \frac{x|y\ |z}{c_x|c_y|c_z}$ jest nieskończone.

Zauważ, ciąg trójek wymieniony w dolnym wierszu diagramu jest malejący(!) i nieskończony. A to z kolei oznacza, że w strukturze tej istnieją elementy nieosiągalne.

Ta obserwcja pozwala udowodnić twierdzenie

Twierdzenie 12. (Collatza)
Program Cl ma własność stopu.
Dla każdej liczby naturalnej n obliczenie programu (w stansardowym modelu liczb naturalnych) jest skończone.

Dowód:
Niech $\mathfrak{M}$ będzie jakimkolwiek modelem elementarnej teorii dodawania liczb naturalnych. Jeśli dla pewnego elementu n obliczenie programu Cl jest nieskończone to struktura ta nie jest izomorficzna z modelem standardowym liczb naturalnych. □

8. Podsumowanie

Nietrudno zauważyć, że przedstawiony dowód jest okrężny.
Nie potrafimy, na razie, przedstawić dowodu wyprowadzonego w algorytmicznej teorii $\mathcal{ATN}$ z aksjomatów tej teorii lub np. z prawa Archimedesa, które jesttwierdzeniem teorii $\mathcal{ATN}$.
Kolejne zadanie to oszacowanie kosztu algorytmu Collatza, wiemy, że obliczenia są skończone, ale nie potrafimy oszacować ich długości.
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.

9. 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 8. 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.

$$\forall_x\, x + 1 \neq 0 \tag{a}$$
$$\forall_x \forall_y\, x + 1 = y + 1 \implies x = y \tag{b}$$
$$\forall_x\, x + 0 = x \tag{c}$$
$$\forall_{x,y}\, (y + 1) + x = (y + x) + 1 \tag{d}$$
$$\{\Phi(0) \wedge \forall_x\, [\Phi(x) \implies \Phi(x+1)] \implies \forall_x \Phi(x) \tag{I}$$
$$\tag{8}$$

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 7 posłużymy się teorią Presburgera

Definicja 9. 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.

8

$$\forall_x \, x + 1 \neq 0 \tag{A}$$

$$\forall_x \, x \neq 0 \implies \exists_y x = y + 1 \tag{B}$$

$$\forall_{x,y} \, x + y = y + x \tag{C}$$

$$\forall_{x,y,z} \, x + (y + z) = (x + y) + z \tag{D}$$

$$\forall_{x,y,z} \, x + z = y + z \implies x = y \tag{E}$$

$$\forall_x \, x + 0 = x \tag{F}$$

$$\forall_{x,z} \, \exists_y \, (x = y + z \vee z = y + x) \tag{G}$$

$$\forall_x \, \exists_y \, (x = y + y \vee x = y + y + 1) \tag{H2}$$

$$\forall_x \, \exists_y \, (x = y + y + y \vee x = y + y + y + 1 \vee x = y + y + y + 1 + 1) \tag{H3}$$

. . . .

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

$\ldots$

Do tego zbioru dodajemy aksjomaty definiujące dodatkowe pojęcia

$$Parz(x) \equiv \exists_y \, x = y + y \tag{p}$$

$$x \, div \, 2 = y \equiv (x = y + y \ \vee \ x = y + y + 1) \tag{D2}$$

Przypomnijmy parę faktów

F! Teoria T_+ jest równoważna teorii Ar.[Pre29, Sta84]

F2. Teoria Ar jest rozstrzygalna. [Pre29].

F3. Złożoność teorii Ar, czyli koszt udowodnienia [FR79].

F4. Teorie T_+ oraz Ar mają modele niestandardowe, tj. modele zawierające elementy nieosiągalne.

10. Dodatek B– dowód Faktu 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 13. Powyższe definicje są poprawne, tzn. twierdzeniami teorii T_+ wzbogaconej w ten sposób są zdania $\forall_x \exists_y \, P2(x) = y$ i $\forall_{x,y,z} P2(x) = y \wedge P2(x) = z \implies y = z$.

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

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

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 $P3(n,x) + y = P2(z)$.

Lemat 14. 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 $\forall_{n,x} \exists_{y,z} P3(n,x) + y = P2(z)$

11. Dodatek C – przykład obliczenia nieskończonego

Niewielu programistów ...
W tym miejscu przypomnimy kilka faktów mniej znanych społeczeczności informatycznej. W dadatku A opisaliśmy teorię Ar dodawanialiczb naturalnych. Jedynym funktorem w języku tej

teorii jest $+$, mamy tez dwie stałe 0 i 1 oraz predykat $=$ równości. Opiszemy strukturę algebraiczną $\mathfrak{M}$, które jest modelem tej teorii tzn. wszystkie aksjomaty tej teorii są prawdziwe w teorii $\mathfrak{M}$. Najpierw opiszey tę strukturę tak jak to robią matematycy, potem napiszemy klasę (tj. moduł programu) implementujacą tę strukturę.

Opis matematyczny struktury niestandardowej

Struktura algebraiczna $\mathfrak{M}$

$$\mathfrak{M} = \langle M; 0, 1, \oplus; = \rangle$$

taka, że M jest zbiorem par $\langle k, w \rangle$ gdzie element $k \in Z$ jestliczbą całkowita, element w jest liczbą wymierną nieujemną i spełnione są warunki

(*i*) dla każdego elementu $\langle k, w \rangle$ jeśli $w = 0$ to $k \geq 0$,

(*ii*) znaczeniem stałej 0 jest para $\langle 0, 0 \rangle$,

(*iii*) znaczeniem stałej 1 jest para $\langle 1, 0 \rangle$,

(*iv*) działanie $\oplus$ dodawania określone jest w następujący sposób

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

Czytelnik zechce sprawdzić, że każdy aksjomat teorii Ar jest zdaniem prawdziwym w strukturze $\mathfrak{M}$.
Struktura $\mathfrak{M}$ nie jest modelem algorytmicznej teorii $\mathcal{ATN}$. Elementy struktury $\langle k, w \rangle$ takie. że $w \neq 0$ są *nieosiągalne* .tj. dla każdego elementu $x_0 = \langle k, w \rangle$ prawdziwe jest zdanie

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

Natomiast fragment $\mathfrak{N} \subset \mathfrak{M}$ złożony z tych tylko elementów dla których $w = 0$ jest modelem teorii $\mathcal{ATN}$. Elementy struktury $\mathfrak{N}$ nazywamy *osiągalnymi*. Bardzo ważnym twierdzeniem podstaw matematyki jest

Fakt 15. Struktury $\mathfrak{N}$ i $\mathfrak{M}$ nie są izomorficzne. Zob. [Grz71], str. ...

Jak to zobaczymy za chwilę, fakt ten ma też znaczenie dla informatyków.

Definicja w języku programowania

Być może czytelnik zauważył już, że struktura $\mathfrak{M}$ jest obliczalna. Poniżej podajemy klasę , która implementuje strukturę $\mathfrak{M}$. Implementacja wykorzystuje typ integer, typu wymierane (ang. rationalNumbers) nie wprowadzamy w sposób jawny.

```
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
```

Następujący lemat wyraża poprawność implementacji

Lemat 16. Zbiór obiektów typu Elm z operacja *add* jest modelem teorii Ar

Nieskończone obliczenia algorytmu Collatza

Jak wykonać algorytm Collatza w StrukturzeM? To proste.

```
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;
```

Czy coś sie zmieni gdy zmiennej n przypiszemy inny obiekt? np. n:= new Elm(19,2,10)?
Nie przejmuj się jeśli nie jesteś programistką.
Pniżej możemy obejrzeć obliczenie algorytmu Collatza dla $n = \langle 8, \frac{1}{2}\rangle$.

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

Żaden z elementów powyższego ciągu nie jest standardowa lizba naturalną. Każdy z nich jest nieosiagalny. Warto spojrzeć na przykład innego obliczenia

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

I jescze jednoobliczenie.

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

Wniosek 3. Struktura $\mathfrak{M}$, którą opisaliśmy na dwa rózne sposoby jest modelem teorii T_+ (możesz też powiedzieć, że struktura ta implementuje specyfikację zadaną aksjomatami teorii Ar), zaskakuje nieoczywistą obecnościa w niej elementów nieosiagalnych.

Inne spostrzeżenie

Wniosek 4. Własności stopu algorytmu Collatza nie da się udowodnić z aksjomatów teorii T_+, ani teorii Ar.

Literatura

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

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

[MS87] Grażyna Mirkowska and Andrzej Salwicki. Algorithmic Logic. PWN, Warszawa, 1987. `http://lem12.uksw.edu.pl/wiki/Algorithmic_Logic`. [Online; accessed 7-July-2021].

[Pre29] Mojżesz Presburger. Über die Vollständigkeit eines gewissen Systems der Arithmetik ganzer Zahlen , in welchem die Addition als einzige Operation hervortritt . pages 92–101,395, 1929.

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