SpecVer & LEM'12 projects ColThm 2018 1–??
Dombrova Research

Dowód twierdzenia Collatza
DRAFT - please do not distribute

Grażyna Mirkowska
Dombrova Research
Partyzantów 19
05-092 Łomianki, POLAND
G.Mirkowska@uksw.edu.pl

Andrzej Salwicki
Dombrova Research
salwicki@mimuw.edu.pl

Streszczenie. Dowodzimy, że dla każdej liczby naturalnej n algorytm Collatza ma obliczenie skończone.

15 listopada 2018

1. Wprowadzenie

Ten szkic jest raportem z aktualnego stanu pracy nad artykułem o twierdzeniu Collatza. Idea dowodu jest bardzo prosta. Pełny dowód wymaga uzupełnienia szczegółów.
Algorytmu Collatza przedstawiać nie trzeba.

$$\{\mathbf{while}\ n \neq 1\ \mathbf{do\ if}\ odd(n)\ \mathbf{then}\ n := 3n+1\ \mathbf{else}\ n := n/2\ \mathbf{fi\ od}\}$$

Mamy zamiar opisać jego obliczenia w dwu, na pozór, różnych strukturach danych: $\mathfrak{N}_0$ oraz $\mathcal{ZT}$. Pierwsza struktura to standardowy zbiór liczb naturalnych $\mathfrak{N}_0$ ze zwykłymi działaniami następnika i dodawania. To wystarcza by dobrze (tj. efektywnie) zdefiniować polecenia n:=3$\cdot$ n +1 oraz n:= n $\div$ 2. Dalej zapisywać je będziemy jako polecenie $mult3$ i polecenie $div2$. W algorytmie Collatza występują też dwie formuły $n \neq 0$ oraz $odd(n)$. Zastąpimy je przez $equal1$ i krótkie odd. W algorytmie Collatza występuje tylko jedna zmienna, nie musimy jej wymieniać z nazwy.

Druga struktura to zbiór trójek liczb naturalnych $\mathcal{ZT}$ z odpowiednio dobranymi poleceniami, zobacz poniżej. Stąd konieczność zapisania tego algorytmu w abstrakcyjny sposób.

$$\{\mathbf{while}\ \mathrm{not}\ equal1\ \mathbf{do\ if}\ odd\ \mathbf{then}\ mult3\ \mathbf{else}\ div2\ \mathbf{fi\ od}\}$$

Jedyna zmienna n występująca w tym programie przyjmuje bądź wartości będące liczbami naturalnymi(realizacja obliczeń w strukturze $\mathfrak{N}_0$) bądź trójkami liczb naturalnych. Polecenia $mult3$ i $div2$ to, odpowiednio: pomnóż n przez 3 i dodaj 1 oraz podziel x przez 2, bądź gdy działania wykonywane są na trójkach polecenia te realizowane sa w inny sposób.

2. Trójki

Zacznijmy od następującego spostrzeżenia

Fakt 1. Dla każdej liczby naturalnej $n \neq 0$ istnieje nieskończenie wiele trójek liczb naturalnych x, y, z takich, ze spełniona jest równość

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

Dowód tego intuicyjnego faktu wykorzystuje prawo Archimedesa. Mówimy, że trójka x, y, z *reprezentuje* (albo, *koduje*) liczbę n.

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

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

Natomiast nie każda trójka liczb naturalnych reprezentuje jakąś liczbę naturalną, np. $\langle 2, 4, 11\rangle$, $\langle 2, 4044, 11\rangle$ ...
W zbiorze trójek możemy zdefiniować porządek leksykograficzny $\prec$.

Definicja 3.

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

Definicja 4. Operacje wykonywane na trójkach:
$div2 : \langle x, y, z\rangle \to \langle x, y \div 2, z - 1\rangle$
$mult3 : \langle x, y, z\rangle \to \langle x - 1, y - 3^{x-1}, z\rangle$

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 $n/2$.
Trójka $\langle x, y, z\rangle$ reprezentuje liczbę n wttw gdy trójka $\langle x - 1, y - 3^{x-1}, z\rangle$ reprezentuje liczbę $3n + 1$.

Definicja 6. *Nieparzystość* trójek

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

Kolejne spostrzeżenie

Fakt 7. Liczba n jest nieparzysta wttw gdy reprezentująca ją trójka $\langle x, y, z \rangle$ jest nieparzysta.

Definicja 8. Jedynka

$$equal1(\langle x, y, z \rangle) \overset{df}{\Leftrightarrow} \frac{2^z - y}{3^x} = 1$$

Jedynka jest reprezentowana przez wiele trójek $\langle 0, 0, 0 \rangle$, $\langle 0, 1, 1 \rangle$, $\langle 1, 1, 2 \rangle$, $\langle 1, 5, 3 \rangle$, ...
Łącząc te obserwacje stwierdzamy, że następujący diagram jest przemienny

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

A więc można realizować obliczenia algorytmu Collatza "*na trójkach*".
Jest oczywiste, że

Fakt 9. *Jeśli obliczenie algorytmu w strukturze trójek jest skończone i wolne od błędu, to algorytm wykonywany w standardowym modelu liczb naturalnych zatrzymuje się po skończonej liczbie kroków.*

Może się jednak zdarzyć, że obliczenie algorytmu wykonywane w strukturze trójek będzie przerwane ponieważ nie da się wyznaczyć kolejnej trójki.

Przykład 10. Obliczenie zaczynające się od trójki $\langle 3, 511, 10 \rangle$ przebiega następująco

$\langle 3, 511, 10 \rangle$	19
$\langle 2, 502, 10 \rangle$	58
$\langle 2, 251, 9 \rangle$	29
$\langle 1, 248, 9 \rangle$	88
$\langle 1, 124, 8 \rangle$	44
$\langle 1, 62, 7 \rangle$	22
$\langle 1, 31, 6 \rangle$	11
$\langle 0, 30, 6 \rangle$	34
$\langle 0, 15, 5 \rangle$	17
BŁĄD obliczenia	

W obliczeniu wystąpił błąd ponieważ dosłowne wykonanie polecenia $mult3$ prowadzi do $\langle 0-1, 15-1/3, 5\rangle$.
Obliczenie takie daje się naprawić
zastapmy trójkę $\langle 0, 15, 5\rangle$ inną trójką reprezentującą tę sama liczbę 17, np. $\langle 4, 671, 11\rangle$ i wróćmy do obliczania.
Oto kontynuacja przerwanego obliczenia.

$\langle 4, 671, 11\rangle$	17
$\langle 3, 644, 11\rangle$	52
$\langle 3, 322, 10\rangle$	26
$\langle 3, 161, 9\rangle$	13
$\langle 2, 152, 9\rangle$	40
$\langle 2, 76, 8\rangle$	20
$\langle 2, 38, 7\rangle$	10
$\langle 2, 19, 6\rangle$	5
$\langle 1, 16, 6\rangle$	16
$\langle 1, 8, 5\rangle$	8
$\langle 1, 4, 4\rangle$	4
$\langle 1, 2, 3\rangle$	2
$\langle 1, 1, 2\rangle$	1
Sukces!	

2.1. Algorytm poprawiający

Zauważmy, że poprawa może sięgnąć wstecz aż do początku obliczenia.

Przypuśćmy, że trójki $\langle x, y, z\rangle$ oraz $\langle u, v, t\rangle$ reprezentują tę samą liczbę m tj $m = \frac{2^z - y}{3^x} = \frac{2^t - v}{3^u}$. Jeżeli w pewnym obliczeniu, bezpośrednim poprzednikiem trójki $\langle x, y, z\rangle$ jest trójka $\langle x', y', z'\rangle$ to w innym obliczeniu, którym występuje trójka $\langle u, v, t\rangle$ bezpośrednim poprzednikiem tej trójki jest trójka

$$\begin{cases} \langle u, 2v, t+1\rangle & \text{gdy } y' \text{ jest parzyste, lub trójka,} \\ \langle u+1, v+3^{u-1}, t\rangle & \text{gdy } y' \text{ jest nieparzyste.} \end{cases}$$

Wykorzystując to spostrzeżenie możemy cofnąć się w obliczeniu do początku i zacząć od nowa z trójką gwarantującą, że obliczenie będzie dłuższe. A więc po pewnej skończonej liczbie takich poprawek obliczenie w strukturze trójek będzie wolne od błędu.

2.2. Twierdzenie Collatza

Powyższe uwagi pozwalają nam sformułować następujące

Twierdzenie 11. (Collatza)
Dla dowolnej liczby naturalnej n obliczenie algorytmu Collatza jest skończone.

Proof:
Dowód wynika z faktu, że każda kolejna trójka jest mniejsza $\prec$ od poprzedzającej ją trójki.

W celu uniknięcia błędu przerwania obliczeń "*na trójkach*" wystarczy obliczenie rozpocząć od trójki z odpowiednio dużymi liczbami x, y, z. □

3. Wnioski

- Nietrudno zauważyć, że realizacja obliczeń w niestandardowym modelu arytmetyki dodawania może mieć obliczenie nieskończone.
  Jako wniosek z twierdzenia Collatza uzyskujemy następujący

  Fakt 12. Jeśli realizujemy algorytm Collatza w pewnym modelu teorii (Peano) pierwszego rzędu to jeśli w tej strukturze jest obliczenie nieskończone to struktura ta jest modelem niestandardowym.

  .

- Algorytm Collatza wyznacza pewną *permutację* zbioru N liczb naturalnych.

- Można opisać inny algorytm (odwrotny) szukający zadanej liczby n po kolei w warstwach, poczynając od warstwy S_0.

  Fakt 13. Algorytm odwrotny zawsze znajdzie liczbę n.

  Algorytm ten przyporządkowuje liczbie n numer i warstwy oraz położenie j liczby n w warstwie. W ten sposób mamy nową *funkcję pary*. Numerem pary liczb $\langle i, j \rangle$ jest j-ta liczba w warstwie S_i.

4. Nowe pytania

1. Jak określić funkcję kosztu?

2. Czy algorytm Collatza wyznacza najmniejszą trójkę x, y, z taką, że obliczenie zaczynające się od niej, będzie wolne od błędu?

3. Rozumowanie przytoczone powyżej wykazuje *prawdziwość* twierdzenia Collatza. Twierdzenie to nie jest zawarte w zbiorze twierdzeń arytmetyki Peano. (...)
   Jak przeprowadzić *dowód* tego twierdzenia w algorytmicznej teorii liczb naturalnych?