ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 1979

Seria: MATEMATYKA-FIZYKA z. 34 Nr kol. 623

Marek BALCER
Andrzej MIKA
Tomasz PIZOŃ

ALGORYTM ROZSTRZYGANIA PRAWDZIWOŚCI RÓWNOŚCI DWÓCH DOWOLNYCH WYRAŻEŃ ARYTMETYCZNYCH

Streszczenie. Autorzy podają algorytm rozstrzygania prawdziwości równości dwóch dowolnych wyrażeń arytmetycznych, traktowanych jako słowa dopuszczalne pewnego alfabetu. Podstawą konstrukcji tego algorytmu jest zbudowanie kilku operatorów dokonujących przekształceń słów dopuszczalnych w myśl praw arytmetyki.

1. WSTĘP

W niniejszej pracy podajemy algorytm rozstrzygania prawdziwości równości dwóch wyrażeń arytmetycznych. Podstawą jego budowy jest definicja słowa dopuszczalnego (SD) i definicja termowego wyrażenia arytmetycznego (TWA) szczegółowo omówione w pracy [1], oraz równoważność tych dwóch definicji podana w postaci twierdzenia w pracy [2]. Przypomnimy tylko krótko, że słowo dopuszczalne zbudowane jest z alfabetów: cyfrowego, literowego i znaków (nawiasy, działania arytmetyczne) przez taką konkatenację elementów tych alfabetów, by otrzymane słowo można było interpretować jako wyrażenie arytmetyczne.

Zbiór termowych wyrażeń arytmetycznych składa się z termów zbudowanych z elementów zbioru słów literowych i cyfrowych za pomocą operatorów mających interpretację działań arytmetycznych na tych słowach. Podstawową metodą konstruowania termowego wyrażenia arytmetycznego jest budowa odpowiedniego drzewa (mówią o tym tw. 1 i tw. 2 zamieszczone wraz z dowodami w pracy [1]). Chcąc przeprowadzić na pewnym TWA przekształcenie w myśl praw arytmetyki spostrzegamy, że struktura drzewa staje się do tego celu niewygodna, ponieważ dokonanie tego przekształcenia pociąga najczęściej duże zmiany w budowie tego drzewa. Opieranie programowania maszyn cyfrowych na strukturze drzewa, zwłaszcza, że ulega ono dużym przeobrażeniom, jest bardzo niewygodne.

Sposób w jaki wprowadzimy przekształcenia słowa dopuszczalnego (SD) wydaje się być łatwiejszym w realizacji.

2. KONSTRUKCJA OPERATORÓW

W pierwszej kolejności opiszemy oprratory, które przekształcają SD oraz ich równości w myśl praw arytmetyki. Oznaczmy przez D zbiór słów dopuszczalnych.

Niech $D_1 = \left\{ C : C = A * =_{\mathcal{A}} * B * ;_{\mathcal{A}} ;\ A, B \in D \right\}$

oraz $D_2 = \left\{ C : C = A * ;_{\mathcal{A}} ;\ A \in D \right\}$.

Elementy zbioru D_2 zapisujemy jako konkatenację poszczególnych znaków z konkatenantów typu L i traktowanych jako niepodzielne konkatenantów typu M. Zatem, jeżeli $A \in D_2$, to

$$A = K_1 * \ldots * K_m * ;_{\mathcal{A}} \quad i(1 \leq i \leq m)\ K_i \in \mathcal{A}_3 \cup \mathcal{A}_4 \cup \mathcal{A}_2 - \left\{ =_{\mathcal{A}}, ;_{\mathcal{A}} \right\}$$

Uwaga! Przez φ_r^s oznaczamy φ dla $K_r \ldots K_s$ (zob. [1]).

Oto operator F_1.

$$F_1 : D_1 \longrightarrow D_2$$

$$\forall C \quad (C \in D_1,\ C = A * =_{\mathcal{A}} * B * ;_{\mathcal{A}})\ F_1(C) = A * -_{\mathcal{A}} * (_{\mathcal{A}} * B *)_{\mathcal{A}} * ;_{\mathcal{A}}$$

A oto kilka innych operatorów:

$$F_6 : D_2 \longrightarrow D_2$$

jeżeli $(\exists_i\ (1 \leq i \leq m-2) : K_i = \ulcorner (_{\mathcal{A}} \urcorner ;$

$\exists_j\ (i+1 \leq j \leq m-1) : K_{i+1} * \ldots * K_j \in D \wedge (\forall_s\ (i+1 \leq s \leq j)$

$\text{na}\ K_s\ \varphi_{i+1}^{j} = 0)$;

$K_{j+1} = \ulcorner)_{\mathcal{A}} \urcorner$, $K_{j+2} \neq \ulcorner *_{\mathcal{A}} \urcorner$ oraz gdy $i > 1$ to $K_{i-1} \neq \ulcorner -_{\mathcal{A}} \urcorner$

$K_{i-1} \neq \ulcorner *_{\mathcal{A}} \urcorner ;$

wtedy

$$F_6(A) = K_1 * \ldots * K_{i-1} * K_{i+1} * \ldots * K_j * K_{j+2} * \ldots * K_m * ;_{\mathcal{A}}).$$

W przeciwnym razie $F_6(A) = A$.
F_6 likwiduje zbędne nawiasy.

$$F_7 : D_2 \longrightarrow D_2$$

jeżeli

$\exists i \quad (2 \leq i \leq m-2) : K_i = \ulcorner (_{\mathcal{A}} \urcorner \wedge K_{i-1} = \ulcorner -_{\mathcal{A}} \urcorner ;$

$\exists j\ (i+1 \leq j \leq m-1)$: $K_{i+1} * \dots * K_j \in D \wedge (\forall_s\ (i+1 \leq s \leq j)$ na

$$K_s \varphi^{j}_{i+1} = 0);$$

$K_{j+1} = \ulcorner (_{\mathcal{A}} \urcorner \wedge K_{j+2} \neq \ulcorner *_{\mathcal{A}} \urcorner$, to

jeżeli $K_{i+1} = \ulcorner -_{\mathcal{A}} \urcorner$, to budujemy $K'_{i+2} * \dots * K'_j$

w następujący sposób:

$\forall r\ (i+3 \leq r \leq j-i)\ (K_r = \ulcorner -_{\mathcal{A}} \urcorner \to K'_r = \ulcorner +_{\mathcal{A}} \urcorner)$

$\forall r\ (i+3 \leq r \leq j-1)\ (K_r = \ulcorner +_{\mathcal{A}} \urcorner \to K'_r = \ulcorner -_{\mathcal{A}} \urcorner)$

$\forall r\ (i+2 \leq r \leq j)\ ((K_r \neq \ulcorner -_{\mathcal{A}} \urcorner \wedge K_r \neq \ulcorner +_{\mathcal{A}} \urcorner) \to K'_r = K_r)$

Wtedy

$F_7(A) = K_1 * \dots * K_{i-2} * +_{\mathcal{A}} * K'_{i+2} * \dots * K'_j * K_{j+2} * \dots * K_m * \, ;_{\mathcal{A}};$

jeżeli $K_{i+1} \neq \overline{-_{\mathcal{A}}}$, to budujemy $K'_{i+1} * \dots * K'_j$

w następujący sposób:

$\forall r\ (i+2 \leq r \leq j-1)\left(K_r = \ulcorner -_{\mathcal{A}} \urcorner \to K'_r = \ulcorner +_{\mathcal{A}} \urcorner\right)$

$\forall r\ (i+2 \quad r \quad j-1)\left(K_r = \ulcorner +_{\mathcal{A}} \urcorner \to K'_r = \ulcorner -_{\mathcal{A}} \urcorner\right)$

$\forall r\ (i+1 \quad r \quad j)\ \left((K_r \neq \ulcorner -_{\mathcal{A}} \urcorner \wedge K_r \neq \ulcorner +_{\mathcal{A}} \urcorner) \to K'_r = K_r\right)$

wtedy

$F_7(A) = K_1 * \dots * K_{i-1} * K'_{i+1} * \dots * K'_j * K_{j+2} * \dots * K_m \ ;_{\mathcal{A}}).$

Gdy i, j jak wyżej nie istnieją, to $F_7(A) = A$, np. dla A postaci

$a -_{\mathcal{A}} (_{\mathcal{A}} b *_{\mathcal{A}} c -_{\mathcal{A}} a +_{\mathcal{A}} d)_{\mathcal{A}}$ $F_7(A) = a -_{\mathcal{A}} b *_{\mathcal{A}} c +_{\mathcal{A}} a -_{\mathcal{A}} d.$

$F_8 : D_2 \to D_2$

jeżeli

$(\exists i\ (2 \leq i \leq m-1)$: $K_i = \ulcorner *_{\mathcal{A}} \urcorner \wedge K_{i-1} = \ulcorner 0_{\mathcal{A}} \urcorner;$

$\exists j\ (i+1 \leq j \leq m)$: $K_{i+1} * \dots * K_j \in D$ przy czym

$$\forall s\ (i+2 \leq s \leq j-1)\Big(\big(K_s \in \{\ulcorner +_{\mathcal{A}} \urcorner, \ulcorner -_{\mathcal{A}} \urcorner\}\big) \longrightarrow \big(\text{na } K_s\ \varphi_{i+2}^{j-1} > 1\big)\Big)$$

oraz $K_{j+1} \neq \ulcorner *_{\mathcal{A}} \urcorner$; to wtedy

$$F_8(A) = K_1 * \ldots * K_{i-1} * K_{j+1} * \ldots * K_m * ;_{\mathcal{A}});$$

w przeciwnym razie $F_8(A) = A$.
F_8 odpowiada własności $0.a = 0$.

$$F_9 : D_2 \longrightarrow D_2$$

jeżeli

$$(\exists i\ (2 \leq i \leq m-1) : K_i = \ulcorner *_{\mathcal{A}} \urcorner \wedge K_{i+1} = \ulcorner 0_{\mathcal{A}} \urcorner;$$

$$\exists j\ (1 \leq j \leq i-1) : K_j * \ldots * K_{i-1} \in D_2 \text{ przy czym}$$

$$\forall s\ (j \leq s \leq i-1)\Big(\big(K_s \in \{\ulcorner +_{\mathcal{A}} \urcorner, \ulcorner -_{\mathcal{A}} \urcorner\}\big) \longrightarrow \big(\text{na } K_s\ \varphi_{j}^{i-2} > 1\big)\Big)$$

oraz gdy $j > 1$, to $K_{j-1} \neq \ulcorner *_{\mathcal{A}} \urcorner$. Wtedy

$$F_9(A) = K_1 * \ldots * K_{j-1} * 0_{\mathcal{A}} * K_{i+2} * \ldots * K_m * ;_{\mathcal{A}}).$$

W przeciwnym razie $F_9(A) = A$.
F_9 odpowiada własności $a.0 = 0$.

$$F_{10} : D_2 \longrightarrow D_2$$

jeżeli

$$(\exists i\ (1 \leq i \leq m-1)\ \exists j\ (i+2 \leq j \leq m) : K_i * \ldots * K_j \in D \text{ przy czym}$$

$$\forall s\ (i \leq s \leq j)\ K_s \in \mathcal{A}_3 \cup \mathcal{A}_4 \cup \{\ulcorner *_{\mathcal{A}} \urcorner\} \text{ i gdy } i > 1, \text{ to } K_{i-1} \neq \ulcorner *_{\mathcal{A}} \urcorner;$$

$$K_{j+1} \neq \ulcorner *_{\mathcal{A}} \urcorner;$$

jeśli $r_1 < r_2 < \ldots\ldots < r_p$ są wszystkimi liczbami takimi, że $i+2 \leq r_s \leq j$ oraz $K_{r_s} \in \mathcal{A}_4$ dla $1 \leq s \leq p$, to budujemy

$$\bar{K} = Kr_1 * Kr_2 * \ldots * Kr_p$$

Konstruujemy $\tilde{K}$ w następujący sposób:

gdy $K_i \in \mathcal{A}_4$ to $\tilde{K} = K_i . \bar{K}$,
gdy $K_i \notin \mathcal{A}_4$ to $\tilde{K} = \bar{K}$.

Jeśli $l_1 < l_2 < \ldots < l_t$ są wszystkimi liczbami spośród $\{i, \ldots, j\}$ takimi, że $K_{l_\delta} \in {}_3$ dla $1 \leq \delta \leq t$ to konstruujemy (gdy $t > 0$)

$$\tilde{K}' = K_{l_1} * \, *_{\mathcal{A}} * K_{l_2} * \, *_{\mathcal{A}} * \ldots * \, *_{\mathcal{A}} * K_{l_t}$$

Wtedy, gdy $t > 0$

$$F_{10}(A) = K_1 * \ldots * K_{i-1} * \tilde{K} * \tilde{K}' * K_{j+1} * \ldots * K_m * \mathfrak{z}_{\mathcal{A}};$$

gdy $t = 0$ $F_{10}(A) = K_1 * \ldots * K_{i-1} * \tilde{K} * K_{j+1} * \ldots * K_m * \mathfrak{z}_{\mathcal{A}})$.

W przeciwnym razie $F_{10}(A) = A$.

F_{10} przekształca iloczyny w sensie arytmetyki do postaci "współczynnik razy pozostała część iloczynu".

Z uwagi na przejrzystość pracy zaprezentowano jedynie 6 spośród 12 operatorów. Operator F_2 realizuje prawo rozdzielności prawostronnej mnożenia względem dodawania albo odejmowania; F_3 prawo rozdzielności lewostronnej mnożenia względem dodawania albo odejmowania; F_5 i F_4 redukuje nawiasy przy iloczynach; F_{11} dokonuje dopuszczalnej redukcji liczb naturalnych zaś F_{12} dokonuje dopuszczalnej redukcji jednomianów z uwzględnieniem przemienności mnożenia.

3. KONSTRUKCJA ALGORYTMU

Dla operatorów F_i $(i=2, \ldots, 10)$ i pewnego $A \in D_2$ wprowadzamy charakterystyki odpowiednio $\delta_2, \ldots, \delta_{10}$, takie, że:

$$\delta_i = \begin{cases} 0 & \text{gdy } F_i(A) = A \\ 1 & \text{gdy } F_i(A) = B \wedge B \neq A \end{cases} \qquad i=2, \ldots, 10$$

Zbudujemy obecnie algorytm rozstrzygania prawdziwości równości typu $A * =_{\mathcal{A}} * B$, gdzie $A, B \in D$. Działa on następująco:

1° Działamy na $A * =_{\mathcal{A}} * B * \mathfrak{z}_{\mathcal{A}}$ operatorem F_1 otrzymując

$$A_1 = F_1(A * =_{\mathcal{A}} * B * \mathfrak{z}_{\mathcal{A}}), \quad A_1 \in D_2;$$

$$\tau := 0; \; \forall j \; (2 < j < 10) \; \delta j := 0. \text{ Przejście do } 2^\circ.$$

2^o $i := 2$. Przejście do 3^o.

3^o Działamy na A_{i-1} operatorem F_i otrzymując

$$A_i = F_i\,(A_{i-1});$$

jeżeli $\delta_i = 1$, to $\tau := 1$; $\delta_i := 0$; $A_{i-1} := A_i$; i powrót do 3^o;

jeżeli $\delta_i = 0$, to $i := i+1$. Przejście do 4^o.

4^o Jeżeli $i \leqslant 10$, to powrót do 3^o. Jeżeli $i > 10$, to przejście do 5^o.

5^o Tworzymy $A_{11} = F_{11}(A_{10})$ i przejście do 6^o.

6^o Tworzymy $A_{12} = F_{12}(A_{11})$; $A_1 := A_{12}$. Przejście do 7^o.

7^o Jeżeli $\tau = 1$, to $\tau := 0$; powrót do 2^o. Jeżeli $\tau = 0$ to koniec.

Jeżeli po zakończeniu pracy algorytmu

1) $A_{12} = \ulcorner ; \mathfrak{A} \urcorner$, to równość $A * = \mathfrak{A} * B$ jest prawdziwa;

2) $A_{12} \neq \ulcorner ; \mathfrak{A} \urcorner$, to równość $A * = \mathfrak{A} * B$ nie jest prawdziwa, przy czym jeżeli

2.1) $A_{12} = K_1 * ; \mathfrak{A} \wedge K_1 \in \mathfrak{A}_4$, to równość ta jest sprzeczna,

2.2) $A_{12} \neq K_1 * ; \mathfrak{A} \wedge K_1 \in \mathfrak{A}_4$, to równość $A * = \mathfrak{A} * B$ jest równaniem.

LITERATURA

[1] Balcer M., Mika A., Pizoń T.: Konstrukcja słów dopuszczalnych pewnego alfabetu jako pojęć równoważnych termowym wyrażeniom arytmetycznym. Zeszyty Naukowe Politechniki Śląskiej.

[2] Balcer M., Mika A., Pizoń T.: Twierdzenie o równoważności definicji słów dopuszczalnych pewnego alfabetu i termowych wyrażeń arytmetycznych. Zeszyty Naukowe Politechniki Śląskiej.

АЛГОРИТМ РЕШЕНИЯ ПРОБЛЕМЫ ИСТИННОСТИ РАВЕНСТВА ДВУХ ПРОИЗВОЛЬНЫХ АРИФМЕТИЧЕСКИХ ВЫРАЖЕНИЙ

Р е з ю м е

В работе приводится алгоритм, разрешающий проблему истинности равенства двух арифметических выражений, рассматриваемых, как допустимые слова некоторого алфавита. В основу конструкции этого алгоритма положены несколько операторов, преобразующих допустимые слова согласно законам арифметики.

L'ALGORITHME DÉCIDANT DE LA VÉRITÉ DE L'EGALITÉ DE DEUX ÉXPRESSIONS ARYTHMÉTIQUES QUELCONQUES

R è s u m é

Les auteurs presentent l'algorithme decidant de la vérité de l'égalité de deux éxpressions arythmétiques, traités comme deux mots admisiblés d'un alphabet. La base de cet algorithme est la construction de quelques opérateurs qui transforment les mots admissibles d'aprés les principes d'arythmétique.