

CORECTAREA AUTOMATĂ A DEPĂȘIRII UNITĂȚII ÎNTR-O MAȘINĂ CU VIRGULA FIXĂ

DE

E. MUNTEANU și T. RUS

*Comunicare prezentată de T. POPOVICI, membru corespondent al Academiei R.P.R.,
în ședința din 8 iulie 1961*

În această lucrare vom prezenta un algoritm care realizează corectarea automată, pe baza unui program, a depășirii limitei stabilite de poziția virgulei într-o mașină cu virgula fixă [3].

Algoritmul realizează următorul proces: presupunem că la o instrucțiune K a programului se realizează o depășire a limitei stabilite de poziția virgulei. În acest caz se vor corecta, pentru această instrucțiune, constantele respective în mod convenabil, iar rezultatul R al operației K se va înscris sub forma $R = r \cdot 2^p$, unde r nu depășește limita stabilită de poziția virgulei, iar p se numește ordinul modificării. În celula unde conform programului inițial urma să apară R , va apare r , iar p se înscris într-o celulă aparte. În continuare se analizează dacă instrucțiunea următoare depinde sau nu de modificările făcute anterior (adică de conținutul celulei $\langle r \rangle$). Dacă această instrucțiune depinde de aceste modificări, rezultatului operației pe care ea o realizează i se pune în corespondență un nou ordin. Dacă ea nu depinde de aceste modificări, se execută în mod normal de către program. În felul acesta se pune în evidență un șir de adrese care corespunde constantelor modificate în decursul calculului, până la un moment dat. Fiecare termen al acestui șir este păstrat într-o celulă a memoriei împreună cu ordinul numărului conținut în adresa care reprezintă termenul respectiv al șirului. Menționăm că oricare ar fi doi termeni ai șirului, ei sînt întotdeauna diferiți și ordinul fiecărui termen este diferit de zero.

Vom prezenta în continuare schema programului acestui algoritm în cazul cînd numerele sînt reprezentate în mașină, astfel ca

$$|X| < 1.$$

În acest caz, depășirea unității poate să apară ca rezultat al operațiilor de împărțire, adunare sau scădere.

Schema programului se compune din trei blocuri, pe care le vom nota cu A , B , C . Blocul A este realizat de următoarea schemă operatorială :

$$[K \rightarrow A][K \rightarrow B][K \rightarrow C] p(C_k \neq C_+) \uparrow^1 p(C_k \neq C_-) \uparrow^2 A_k^1 \times \\ \times p(R=0) \uparrow^3 A_k^2 A^3 \downarrow^k A_k^4 \downarrow^4 A^5 \downarrow^5 [r_k \rightarrow (r_k + h)_{10}^0] [(P) \rightarrow (r_k + h)_{10}^{16}] \times \\ \times \uparrow^6 A_k^6 A^7 \uparrow^k A_k^8 \downarrow^4 A^9 \downarrow^4 A_k^9 A^{10} \downarrow^5 A_k^8 \uparrow^5$$

unde am notat cu C_k codul instrucțiunii K , iar cu $(\alpha)_n^p$ celulele restrânse la ordinele de la n la p .

Blocul A realizează, în urma depășirii unității, ca rezultat al executării instrucțiunii K , modificările necesare constantelor (a_k) și (b_k) , trece la executarea instrucțiunii K în program, reface constantele modificate la valoarea lor inițială și formează un termen corespunzător al șirului. Acest bloc predă conducerea calculului instrucțiunii b_1 din blocul B .

Blocul B este realizat de următoarea schemă operatorială :

$$b_1 \uparrow^1 \downarrow^k p(a_{k+1} = (a_{k+1} + h)_{10}^0) \uparrow^2 p(b_{k+1} = (b_{k+1} + h)_{10}^0) \uparrow^3 p(C_{k+1} \neq C_+) \downarrow^4 \times \\ \times p(C_{k+1} = C_-) \uparrow^5 \downarrow^8 A_k^1 \uparrow^4 \downarrow^5 [(a_{k+1} + h)_{10}^{16} \rightarrow P_a] [(b_{k+1} + h)_{10}^{16} \rightarrow P_b] \times \\ \times p((P_a) = (P_b)) \uparrow^6 [(P_a) \rightarrow P] \uparrow^6 A_k^2 A^3 [(P_a) \rightarrow P] \uparrow^{k+1} \downarrow^{k+1} A_k^4 \uparrow^{c_1} \downarrow^3 \times \\ \times p(C_{k+1} \neq C_+) \uparrow^7 p(C_{k+1} = C_-) \uparrow^8 \downarrow^7 [(a_{k+1} + h)_{10}^{16} \rightarrow P_a] A_k^5 \uparrow^{k+1} \downarrow^{k+1} \times \\ \times A_k^6 [(P_a) \rightarrow P] \uparrow^{c_1} \downarrow^2 p(b_{k+1} = (b_{k+1} + h)_{10}^0) \uparrow^9 p(C_{k+1} \neq C_+) \uparrow^{10} p(C_{k+1} = C_-) \times \\ \times \downarrow^{10} \downarrow^{10} [(b_{k+1} + h)_{10}^{16} \rightarrow P_b] A_k^7 \uparrow^{k+1} \downarrow^{k+1} A_k^8 [(P_b) \rightarrow P] \uparrow^{c_1} \downarrow^9 F(k) \uparrow^1.$$

Blocul B realizează deci analiza instrucțiunii $K + 1$ din punct de vedere al faptului că a_{k+1} și b_{k+1} sînt sau nu termeni ai șirului considerat. În cazul în care a_{k+1} sau b_{k+1} sînt termeni ai șirului considerat, face modificările necesare pentru executarea instrucțiunii $K + 1$ și reface constantele a_{k+1} și b_{k+1} la valorile lor inițiale. În cazul în care a_{k+1} și b_{k+1} nu sînt termeni ai șirului, trece la executarea instrucțiunii $K + 1$ în program și analizează instrucțiunea următoare. Acest bloc predă conducerea calculului instrucțiunilor c sau c_1 ale blocului C .

Blocul C are următoarea schemă operatorială :

$$c \uparrow^{k+1} \downarrow^{k+1} p((P) \neq 0) \uparrow^{b_1} (r_{k+1} \rightarrow (r_{k+1} + h)_{10}^0) [(P) \rightarrow (r_{k+1} + h)_{10}^{16}] \times F(x) \uparrow^{b_1}.$$

Blocul C realizează deci un termen al șirului în cazul cînd acesta nu provine dintr-o depășire a unității, ci dintr-o modificare făcută de blocul B .

АВТОМАТИЧЕСКОЕ КОРРЕКТИРОВАНИЕ ПРЕВЫШЕНИЯ ЕДИНИЦЫ В МАШИНЕ С НЕПОДВИЖНОЙ ЗАПЯТОЙ

РЕЗЮМЕ

В настоящей работе излагается программа, автоматически осуществляющая изменение в случае, если какой-нибудь промежуточный результат превысит единицу, т.е. тогда, когда операция явилась бы невыполнимой при наличии машины с неподвижной запятой. Изменение вступает в действие только в случае надобности.

CORRECTION AUTOMATIQUE AU CAS OÙ L'UNITÉ SERAIT DÉPASSÉE DANS UNE MACHINE À VIRGULE FIXE

RÉSUMÉ

Les auteurs exposent un programme réalisant automatiquement une modification, lorsqu'un résultat intermédiaire surpasserait l'unité, c'est-à-dire lorsque l'opération ne saurait être réalisée par une machine à virgule fixe. La modification n'entre pas en fonction que si le besoin s'en ferait sentir.

BIBLIOGRAFIE

1. A. A. LIAPUNOV, *Despre schemele logice de programe*. Probleme de cibernetică, Seria tehnică, Biblioteca Analelor Romîno-Sovietice, 69-70.
2. D. D. MC CRACKEN, *Digital Computer Programing*, John Wiley and Sons, Inc. New York, 1957.
3. М. Р. ШИРА БУРА, *Система стандартных подпрограмм*, Москва, 1958.