БЪЛГАРСКА АКАДЕМИЯ НА НАУКИТЕ ACADÉMIE BULGARE DES SCIENCES
ИЗВЕСТИЯ НА МАТЕМАТИЧЕСКИЯ ИНСТИТУТ
BULLETIN DE L'INSTITUT DE MATHÉMATIQUES
Том (Tome) XIII

ПРЕОБРАЗУВАНЕ НА СИНТАКСИСА НА АЛГОРИТМИЧНИ ЕЗИЦИ ВЪВ ВРЪЗКА С ПОСТРОЯВАНЕ НА СИНТАКСИЧЕСКИ УПРАВЛЯЕМИ ТРАНСЛАТОРИ

Петър Бърнев, Валентин Томов и Маргарита Бърнева

Разработката на всеки транслатор изисква точно описание на синтаксиса на съответния алгоритмичен език. В тази статия се счита, че синтаксическото описание се извършва чрез метаезика на Bakus, който се използва в съобщенията на редица алгоритмични езици, като АЛГОЛ-60 [1], АЛГЭК [2], АЛГАМС [3].

Непосредственото използване на синтаксическите описания на алгоритмичните езици при създаване на транслатори е неудобно, тъй като тези описания са предназначени главно за детайлно запознаване с езика. Поради това се налага преобразуването на синтаксическото описание във вид на таблици, диаграми и т. н.

Особено важен е въпросът за преобразуване на синтаксическите описания при конструкцията на синтаксически управляеми транслатори. В този случай синтаксисът на даден алгоритмичен език играе ролята на входна информация за транслатора и следователно е необходимо да се фиксира стандартна форма за неговото представяне, която да бъде удобна за използуване от транслатора.

С цел да се опрости структурата на транслатора синтаксическото описание се разделя на две части чрез въвеждането на система от базисни понятия и символи. За базисни понятия се избират такива понятия, които могат да се синтезират лесно в началния стадий (редакторската фаза) на транслацията. По-нататък е необходима само тази част от синтаксическото описание, която изгражда целевото понятие *програма* чрез базисните понятия и символи вместо чрез основните символи на алгоритмичния език. Именно тази част на синтаксическото описание трябва да бъде преобразувана в подходяща за използване форма.

Проверката дали дадена система образува базис и преобразуването на синтаксическото описание в подходящ за транслатора вид е трудоемна работа, в която лесно могат да се допуснат грешки. Тези грешки се откриват и отстраняват трудно в по-нататъшната разработка на транслатора. Поради това от значение е тези дейности да се алгоритмизират и да се извършват чрез автоматична сметачна машина. Настоящата работа е посветена на този въпрос. Като изходна форма на синтаксиче-

ското описание се използва записването му чрез метаезика на Bakus, а за преобразувана форма се използва описаното от L. Bolliet в [4] представяне на синтаксиса в разпознавателна форма във вид на списък със създаване на съответна матрица на наследниците. След уточняването на проблема и на някои негови особености се описват разработените алгоритми и програми за изследване на избрания базис и за преобразуване на синтаксическото описание. Избран е базис и се привеждат подробно окончателните резултати за езика АЛГОЛ-60, като във връзка с това се отбелязват някои особености в синтаксическото описание на този език.

1. ВЪВЕЖДАНЕ НА СИСТЕМА ОТ БАЗИСНИ ПОНЯТИЯ В СИНТАКСИЧЕСКОТО ОПИСАНИЕ

Синтаксическото описание определя целевото понятие γ (програма) чрез съвкупността A от основните символи $\alpha_1, \alpha_2, \ldots, \alpha_p$, като се използва съвкупност B от междинни понятия $\beta_1, \beta_2, \ldots, \beta_{m-1}$.

Синтаксическото описание съдържа съвкупност F от металингвистични формули φ_i, $i=1, 2, \ldots, m$. Всяка металингвистична формула определя понятие δ ($\delta=\gamma$ или $\delta \in B$) чрез елементи от множествата A и B. Множеството F трябва да е пълно в смисъл, че съдържа формули, чрез които понятието програма в крайна сметка може да се определи само чрез елементи от множеството A.

Всяка металингвистична формула има следната структура:

$$\delta ::= x_1 \mid x_2 \mid x_3 \mid \ldots \mid x_n,$$

където знакът има смисъл на „или“, а x_i са един или няколко присъединени елемента от множествата A и B.

Под система от базисни елементи (базис) ще разбираме всяко множество P от елементи, принадлежащи на A и B, за които съществува подмножество F' от F така, че понятието γ да може да бъде изразено чрез елементите на P, като се използват формулите от F'. С други думи, базисът P и множеството F' играят ролята съответно на A и F в случая, когато синтаксическото описание е изградено чрез основните символи. Множеството F' се получава от F, като се премахнат тези металингвистични формули, които служат за определяне на базисните елементи — понятия — чрез основните символи.

Изборът на подходящ базис се извършва въз основа на редица съображения. Поради сложността на връзките и синтаксическото описание установяването дали дадена система от елементи образува базис е трудоемна дейност. Създаването на алгоритъм, чрез който може да се провери дали дадена система елементи образува базис, и ако не, по какви причини, улеснява експериментирането за намиране на необходимия базис.

2. РАЗПОЗНАВАТЕЛНА ФОРМА НА СИНТАКСИСА И КОДИРАНЕТО Й ВЪВ ВИД НА СПИСЪК. МАТРИЦА НА НАСЛЕДНИЦИТЕ

След въвеждането на базис и синтезирането на базисните понятия в редакторската фаза на транслацията по-нататък транслацията изисква извършването на синтаксически анализ.

Удобна форма за представяне на синтаксическото описание на даден алгоритмичен език за ползване от синтаксически управляем транслатор се описва в [4]. Там е изтъкнато като целесъобразно синтаксисът да се представи в така наречената разпознавателна форма, записана във вид на списък, като освен това се състави специална матрица, наречена матрица на наследниците.

Разпознавателната форма на синтаксическото описание се получава, като всички металингвистични формули

$$\delta ::= x_1 \mid x_2 \mid \ldots \mid x_n$$

се представят във вида

$$x_1 \sqsupset \delta,$$
$$x_2 \sqsupset \delta,$$
$$\cdot$$
$$x_n \sqsupset \delta$$

и след това се групират в отделни групи формулите, имащи един и същ най-ляв елемент в съответното x_i.

Вътре в дадена група формулите се подреждат по някакъв признак. За такъв признак е избран броят на елементите във формулата, като подреждането се извършва в растящ ред.

С цел да се съкрати необходимото място за записването на отделните групи в оперативната памет и да се улесни работата при синтаксическия анализ всяка група от разпознавателната форма се записва във вид на дърво. За удобно записване в оперативната памет дървото се представя във вид на списък.

При този начин на кодиране се избягва повторното записване на еднаквите начални части на формулите и се улеснява обработката на информацията по дървото в низходяща и възходяща посока.

Така например групата металингвистични формули в разпознавателен вид

$$\alpha\beta \sqsupset \delta,$$
$$\alpha\beta\gamma \sqsupset \beta,$$
$$\alpha\beta\alpha\gamma \sqsupset \gamma,$$
$$\alpha\beta\alpha\gamma\beta\alpha \sqsupset \beta$$

се записва като дървовидна структура в списък, както е показано в табл. 1, като знакът * означава, че няма съответно предшественик или наследник, т. е. начало или край на формула.

Таблица 1

Пореден номер	E	Предшественик	Наследник
1	α	*	
2	β	1	4, 6
3	δ	2	*
4	γ	2	
5	β	4	
6	α	2	
7	γ	6	9
8	γ	7	*
9	β	7	
10	α	9	
11	β	10	*

За редовете, които не са край на формула, първият наследник е елементът от следващия ред и това не се отбелязва в стълба за наследниците.

Синтаксическият анализ се опростява и ускорява, ако освен с кодираното синтаксическо описание се разполага и с така наречената матрица на наследниците.

Матрицата на наследниците съдържа по един ред за всеки елемент (понятие или символ) от синтаксическото описание и по един стълб за всяко небазисно понятие (наследник).

Елементът a_{ij} на матрицата има стойност 1, ако елементът от i-тия ред има за пряк или косвен наследник понятието от j-ия стълб. В противен случай a_{ij} 0.

Пряк наследник на даден елемент наричаме най-десния елемент във формула от разпознавателната форма на синтаксическото описание, която започва с дадения елемент.

Всеки пряк наследник на наследника на даден елемент се нарича косвен наследник на дадения елемент.

Матрицата на наследниците може да се получи от разпознавателната форма на синтаксическото описание, като най-напред в матрицата се отразят преките наследници и след това по определен алгоритъм се намерят и косвените наследници.

3. АЛГОРИТМИ ЗА АВТОМАТИЧНО ПРЕОБРАЗУВАНЕ НА СИНТАКСИЧЕСКОТО ОПИСАНИЕ ПРИ ДАДЕН БАЗИС И ЗА СЪЗДАВАНЕ НА МАТРИЦАТА НА НАСЛЕДНИЦИТЕ

По-долу се разглеждат разработените алгоритми и съответна програмна система, предназначена за облекчаване на избирането на базис и за извършване на необходимите преобразования на синтаксическото описание, включително за създаване на матрица на наследниците.

3.1. Начално кодиране на синтаксическото описание

Кодирането на синтаксическото описание се извършва с цел да се опрости по-нататъшната машинна обработка, а също и за да може да се получи крайният резултат от обработката в кратък и удобен за ползване вид. Предполага се, че синтаксисът се въвежда в определено конкретно представяне на символите на метаезика (: : =, |, ⟨ ⟩ и край на формула) и на основните символи. На наименованията на понятията се съпоставят началните естествени числа, а на основните символи — последователни естествени числа, започващи от някое число N, надминаващо броя на понятията. Символът се пропуска, символът | се отразява в прекодирания синтаксис, като кодът на следващото по-

Таблица 2

Кодове на понятията

Понятие	Цифра	Цяло без знак
Код	1	2

Таблица 3

Кодове на основните символи

Основен символ	0	1	2	3	4	5	6	7	8	9
Код	10	11	12	13	14	15	16	17	18	19

нятие се вземе с отрицателен знак. Краят на формула се означава с код 0. Кодът +0 означава празен код.

В резултат на прекодирането синтаксисът се преобразува в масив от цели числа и се получават таблици на кодовете на понятията и на кодовете на основните символи.

Например, ако е даден синтаксисът

⟨цифра⟩ : : = 0| 1 2 3| 4| 5| 6| 7 8 9

⟨цяло без знак⟩ : = ⟨цифра⟩|⟨цяло без знак⟩ ⟨цифра⟩,

резултатът от началното кодиране на синтаксиса е масивът от цели числа 1, 10, 11, 12, −13, −14, −15, −16, −17, −18, −19, −0, 2, −1, −2, 1, −0 и табл. 2 и 3.

В случая за N е избрано естественото число 10.

3.2. Обща схема на алгоритъма за преобразуване на синтаксическото описание и създаване матрица на наследниците

Изходна информация за работата на алгоритъма е кодираното синтаксическо описание (масив A от цели числа) и понятията от базиса (масив B от естествени числа).

В резултат от работата на алгоритъма се получава:

а) Ако зададените в масива B понятия могат да участвуват в базис:

— списък на основните символи, които участвуват в базиса, заедно със зададените понятия;

— списък на отстранените понятия;

— синтаксическото описание в разпознавателна форма, построено върху базисните елементи и кодирано във вид на списъци за отделните понятия (целево и междинни);

— матрица на наследниците.

б) ако зададените в масива B понятия не могат да участвуват в базис:

— списък на тези от понятията, които са зададени за базисни, или на междинните понятия, чрез които те се определят посредством основните символи, с указание на металингвистичните формули, в които тези понятия се срещат след елиминирането на определящите ги металингвистични формули.

Структурата на алгоритъма в общи линии е отразена на блок-схемата от фиг. 1. За простота някои части от алгоритъма, носещи спомагателен характер, са пропуснати. Алгоритъмът се състои от три основни части:

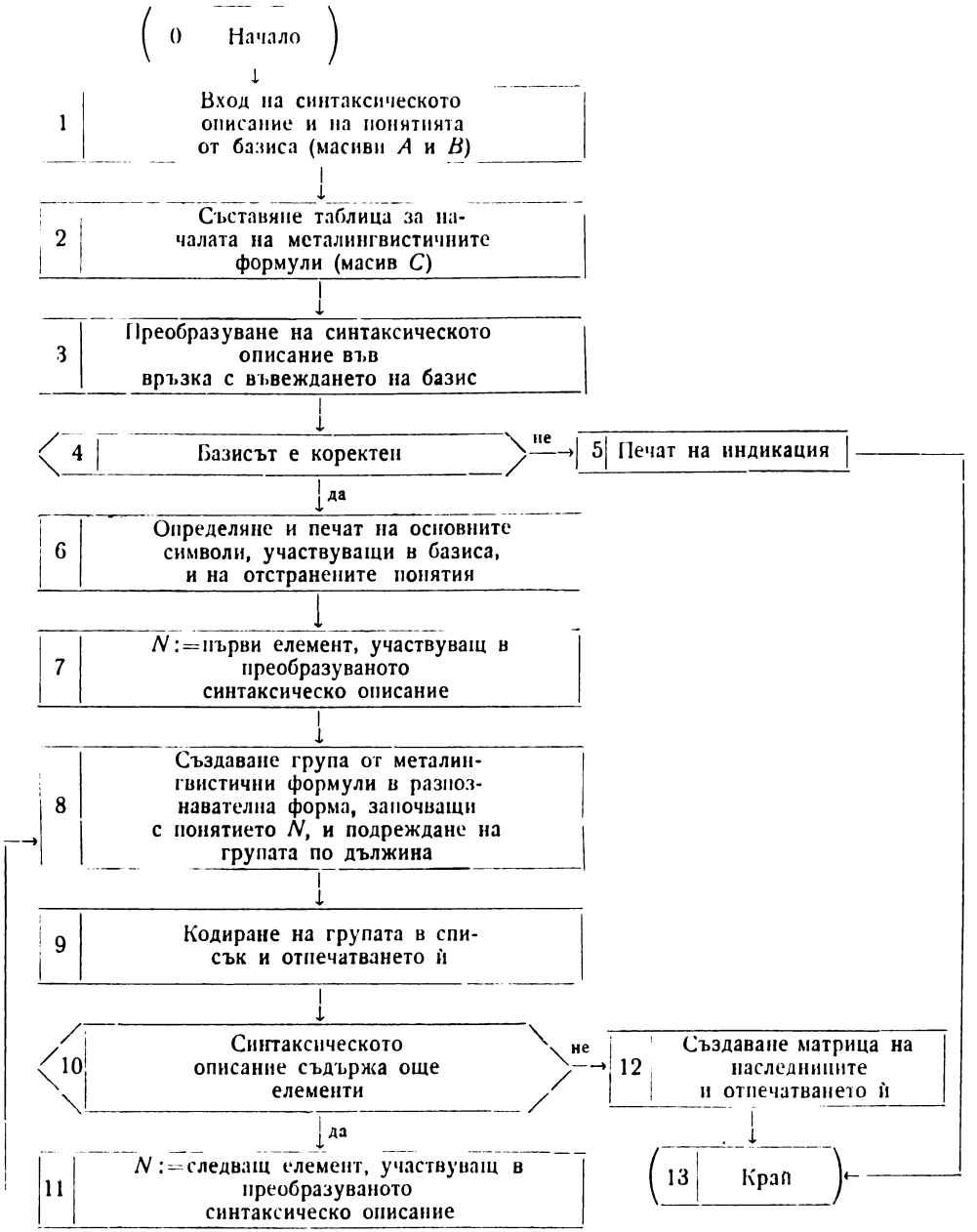
преобразуване на синтаксическото описание във връзка с въвеждането на базис и проверка дали зададените понятия могат да служат за образуването на базис (блокове 3, 4, 5);

преобразуване на синтаксическото описание в разпознавателна форма с представянето на всяка група формули, отнасящи се до дадено понятие, във вид на дърво, кодирано като списък (блокове 7, 8, 9, 10, 11);

създаване на матрица на наследниците (блок 12).

Блоковете 1, 2, 6 играят спомагателна роля. Действието на основните части от алгоритъма е разгледано по-долу.

За описания алгоритъм е създадена програма на езика за символично програмиране в системата МИД-2 за машината „Минск-2“ на Ма-

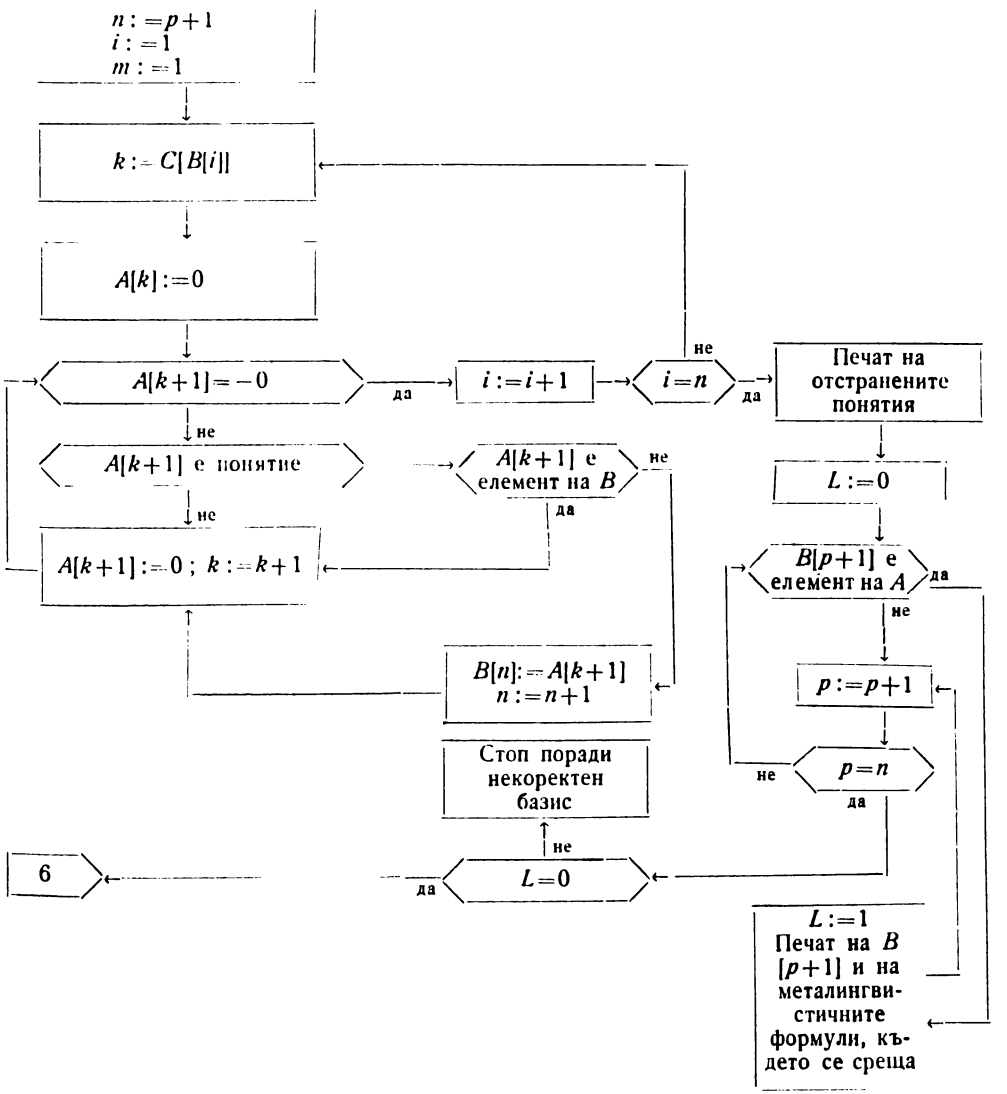


Фиг. 1. Обща блок-схема на алгоритъма

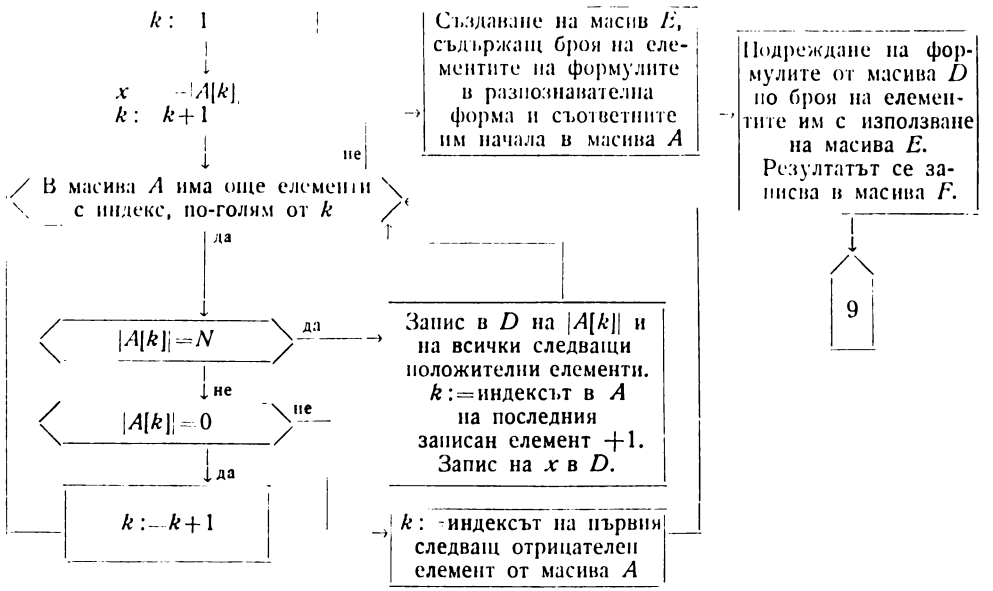
тематическия институт с Изчислителен център на БАН. Програмата се състои от две глави и има общо около 800 оператора. Тя допуска обработката на синтаксическо описание с до 450 понятия и основни символи, като общият брой на елементите на металингвистичните формули не надвишава 2000. За обработката на синтаксическото описание на езика АЛГОЛ-60 са необходими около 15—20 минути машинно време.

3.3. Някои подробности и забележки по алгоритмите за преобразуване на синтаксическото описание

Алгоритъмът за проверка дали дадена система задава базис и за преобразуване на синтаксическо описание при въвеждане на базиса е построен така, че изисква задаването само на тези елементи от базиса, които представляват понятия. Основните символи, които трябва да участвуват в базиса, се включват автоматично и техният списък се отпечатва. По такъв начин се улеснява задаването на информацията и експериментирането за намирането на подходящ базис



Фиг. 2. Блок-схема на алгоритъма за преобразуване на синтаксическото описание във връзка с въвеждане на базис (блокове 3, 4 и 5 от фиг. 1)



Фиг. 3. Блок-схема на алгоритъма за създаване на група от металингвистични формули в разпознавателна форма, подредени по броя на елементите и започващи с елемента N (блок 8 от фиг. 1)

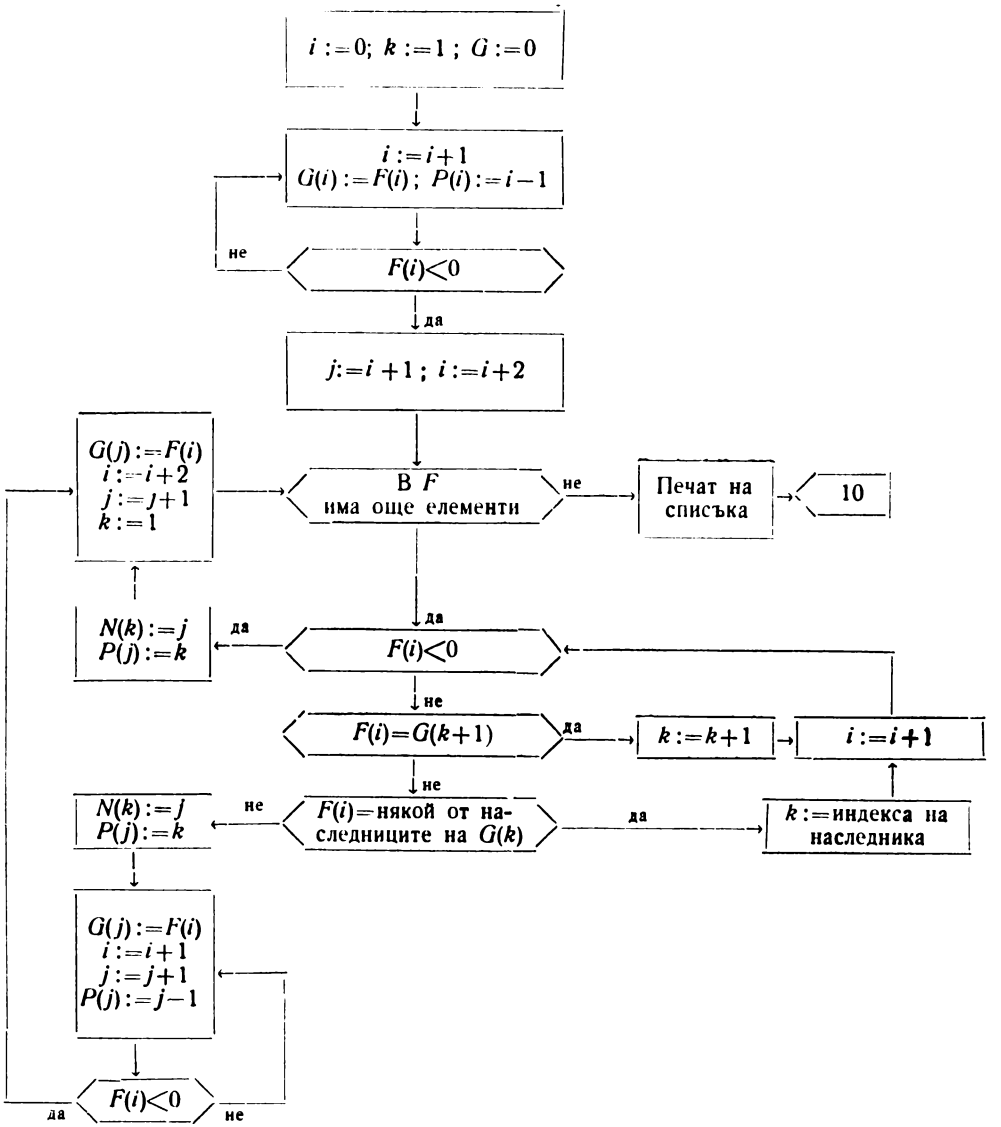
Алгоритъмът използва масивите от цели числа A, B и C. Първоначално масивът A съдържа кодирания синтаксис преди въвеждането на базис; масивът B съдържа понятия (на брой P), с които ще се направи опит да се изгради базис; масивът C съдържа индексите на елементите от масива A, които са начала на металингвистичните формули.

Алгоритъмът действува по следния начин. За всяко понятие от масива B се изследва определящата го металингвистична формула, като участвуващите в нея понятия се записват също в масива B, ако не са вече записани в него. Изследваната металингвистична формула се отстранява от синтаксическото описание. След извършване на тази дейност с всички понятия от масива B (както първоначалните, така и явилите се допълнително) в този масив се съдържат понятията, които (с изключение на базисните) трябва да бъдат отстранени от синтаксическото описание. Техният списък се отпечатва и след това се проверява дали в останалите в масива A металингвистични формули се среща някое от тези понятия. При среща на такова понятие, което означава, че базисът е некоректен, се отпечатва индикация и се продължава изследването, за да се открият всички металингвистични формули, в които се явяват понятия, които не трябва да участвуват в новото синтаксическо описание.

Ако базисът е коректен, към него се присъединяват и се отпечатват всички основни символи, участвуващи в новото синтаксическо описание.

На фиг. 2 е дадена блок-схемата на алгоритъма.

Алгоритъмът за преобразуване на синтаксиса в разпознавателна форма е описан в блок-схемата от фиг. 3. Този алгоритъм използва ма-



Фиг. 4. Блок-схема на алгоритъма за кодиране на група металингвистични формули в разпознавателна форма в списък (блок 9 от фиг. 1)

сива A, съдържащ синтаксическото описание, и записва резултата — групата металингвистични формули в разпознавателна форма, подредени по нарастващ ред на броя на елементите, в масива F. Масивите D и E се използват като работни. Резултатът се кодира, като номерът на последното понятие от всяка металингвистична формула е с отрицателен знак, а номерата на всички останали елементи са с положителни знаци. Алгоритъмът за кодиране на група металингвистични формули в разпознавателна форма във вид на списък използва масива G, в който последователно включва формулите от в масива F. Всяка следваща формула се включва в масива G,

като за начална част се използва някоя от началните части на вече включените формули. Остатъкът от формулата се записва в края на масива G, като се свързва с вече включената от формулата част посредством маркери, наречени предшественик и наследник. Използва се масив N, във всяка компонента на който могат да се запишат индексите на наследниците на съответния елемент на G. Индексите на предшествениците се записват в масива P. Поради дървовидната структура на списъка всеки елемент може да има няколко наследника и единствен предшественик.

На фиг. 4 е дадена блок-схема, която описва основните действия на алгоритъма.

3.4. Алгоритъм за образуване на матрицата на наследниците

В алгоритъма за образуване на матрицата на наследниците е прието, че на началните редове на матрицата се съпоставят небазисните понятия. Броят n на стълбовете на матрицата е равен на броя на тези понятия. Наследникът, който се съпоставя на k-ия стълб, $k = 1, 2, 3, \ldots, n$, е понятието от k-ия ред на матрицата. На всеки стълб в даден ред отговаря по един двоичен разред.

Най-напред, като се използват непосредствено металингвистичните формули, в матрицата се записват преките наследници на елементите на синтаксическото описание.

След това всеки ред на матрицата се обработва отделно, като отначало се определят наследниците на преките наследници, а след това в последователни стъпки се определят само наследниците на новопоявилите се наследници, докато се окаже, че нови наследници не се получават.

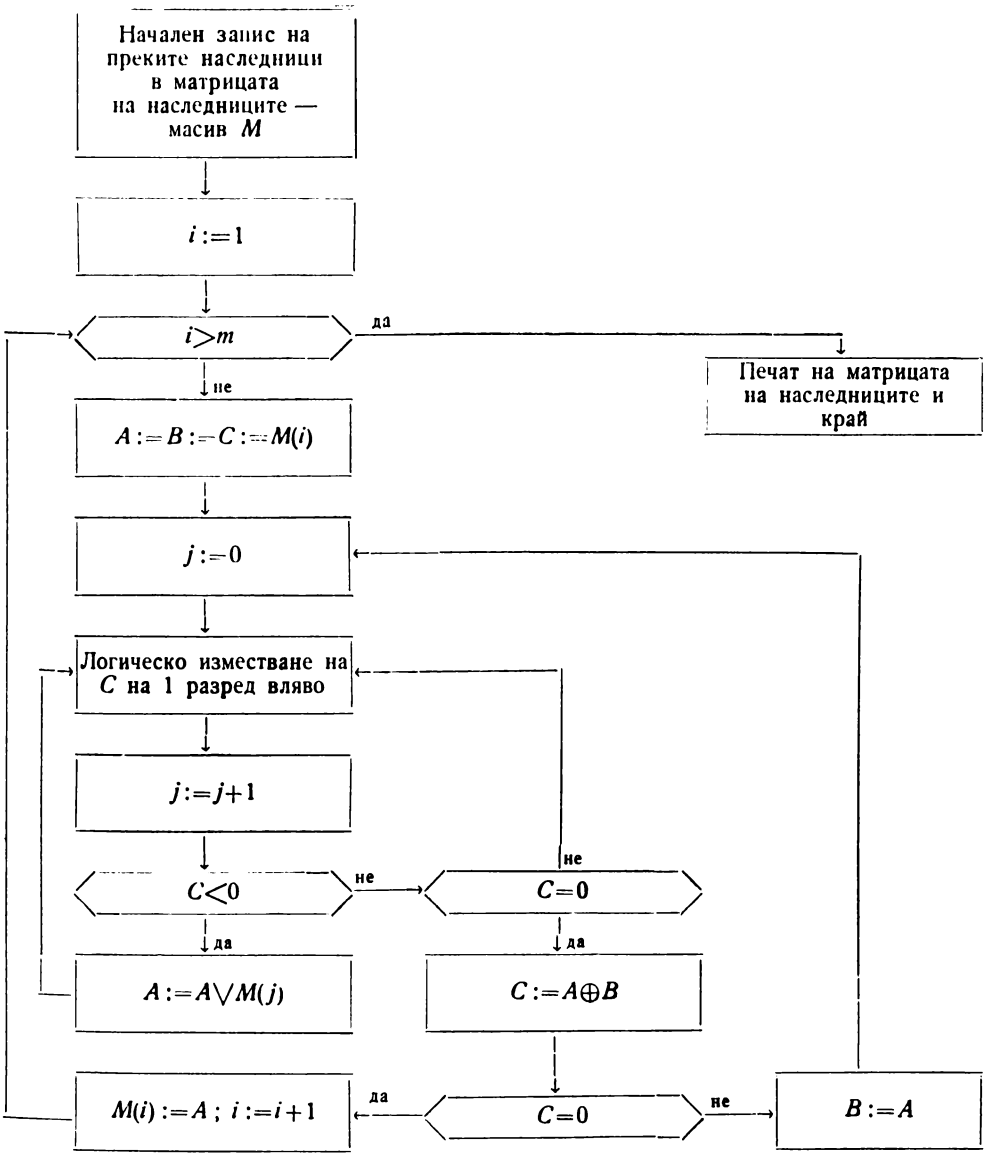
На фиг. 5, където е дадена блок-схема, описваща основните действия на алгоритъма за образуване на матрицата на наследниците, с M е означен масив, на който елементите са редовете на матрицата, а с m — техният брой.

3.5. Кодировка на резултатите

Резултатите — списък на основните символи, включени в базиса списък на отстранените понятия, разпознавателна форма на синтаксическото описание и матрицата на наследниците — се отпечатват в цифрова кодировка, тъй като те са предназначени да служат за входна информация на синтаксически управляем транслатор.

Разпознавателната форма на синтаксическото описание се отпечатва като последователност от списъци, отговарящи на различните групи. Общата структура на списъка бе описана в т. 2.

Всеки списък се състои от отделни редове. Редът съдържа пореден номер, код на елемент от синтаксическото описание, номер на предшественика и номера на наследниците. След предшественика и в края на реда е поставен символът ; Символът -|- означава липса на предшественик или липса на наследници, т. е. съответно начало или край на металингвистична формула. Ако редът не е край на металингвистична формула, се подразбира, че първият наследник е елементът от следващия ред. При по-голяма дължина на реда записът му продължава в полуредове под него.



Фиг. 5. Блок-схема на алгоритъма за образуване на матрицата на наследниците (блок 12 от фиг. 1)

Общата структура на матрицата на наследниците бе описана в т. 2 и 3.4.

Матрицата на наследниците се кодира за краткост в осмична система, т. е. всеки три последователни двоични стълба се обединяват в един осмичен стълб.

4. НЯКОИ ЗАБЕЛЕЖКИ ВЪРХУ СИНТАКСИСА НА АЛГОЛ-60

В различните раздели на синтаксическото описание на АЛГОЛ-60 [1] редица формули се срещат неколкократно, както е показано в табл. 4.

Наред с това в [1] са въведени някои понятия, които не се използват при определянето на други понятия и следователно са излишни. Това естествено не се отнася до понятието програма, което е цел на синтаксическото описание.

Таблица 4

Понятие в езика АЛГОЛ-60	В кои раздели на [1] се среща формула, определяща понятието
Знак на операция за отношение	2.3, 3.4
Фактически параметър	3.2, 4.7
Низ от букви	3.2, 4.7
Ограничител на параметър	3.2, 4.7
Списък от фактически параметри	3.2, 4.7
Съвкупност от фактически параметри	3.2, 4.7
Условие	3.3, 4.5
Безусловен оператор	4.1, 4.5

Накрая, използват се групи понятия, които от гледна точка на синтаксическото описание са синоними, поради което от всяка група е достатъчно да се избере по един представител.

Излишните понятия и групите синтаксически синоними в АЛГОЛ-60 са дадени в табл. 5.

Таблица 5

Излишни понятия в АЛГОЛ-60	Групи от синтаксически си и в АЛГОЛ-60
Основен символ	Идентификатор, идентификатор на променлива, идентификатор на масив, идентификатор на процедура, идентификатор на ключ, формален параметър, проста променлива
Ограничител	
Знак на операция	
Знак на аритметична операция	Аритметичен израз, индексен израз, долна граница, горна граница
Знак на логическа операция	
Знак на операция за следване	Указател на функция, оператор за процедура
Разделител	Празен оператор, празно
Скоба	
Описател	
Спецификатор	
Число	

Приведените в табл. 5 излишни понятия са на базата на синтаксическото описание на АЛГОЛ-60, както то е дадено в [1]. В същност всички нерекурсивно определени понятия са излишни, тъй като те могат да бъдат заместени чрез определящите ги формули във формулите, в които тези

понятия се използват като определящи. В този смисъл например понятието логическа стойност е излишно, тъй като се определя нерекурсивно от металингвистичната формула

⟨логическа стойност⟩ : : = **true** | **false**,

а се среща като определящо понятие само във формулата

⟨първичен логически израз⟩ : : = ⟨логическа стойност⟩ | ⟨променлива⟩ | ⟨указател на функция⟩ | ⟨отношение⟩ | (⟨логически израз⟩).

Следователно, ако понятието логическа стойност се замести в последната формула, т. е. ако тази формула се представи във вида

⟨първичен логически израз⟩ : = **true** | **false** | ⟨променлива⟩ | ⟨указател на функция⟩ | ⟨отношение⟩ | (⟨логически израз⟩),

то понятието логическа стойност няма да се среща никъде в синтаксическото описание и ще бъде също излишно.

Използването на повтарящи се формули, на излишни понятия и синтаксически синоними от авторите на [1] вероятно се дължи на липсата на средства за формализиране на семантиката и на желанието да се направи описанието на синтаксиса по-достъпно за разбиране. От друга страна, повторението на формули и използването на излишни понятия и синтаксически синоними затруднява формалната обработка на синтаксическите описания. Поради това още при кодирането на синтаксическото описание трябва да се отстранят дубликатите на повтарящите се металингвистични формули, да се запази само по един представител от синтаксическите синоними, като се отстранят металингвистичните формули, определящи синонимите, и излишните понятия, и се опростят формулите, съдържащи синоними. Такъв е например случаят с металингвистичната формула

⟨фактически параметър⟩ : : = ⟨низ⟩ | ⟨израз⟩ ⟨идентификатор на масив⟩ | ⟨идентификатор на процедура⟩ | ⟨идентификатор на ключ⟩.

Като се вземе пред вид, че последните три елемента на тази формула са синтаксически синоними на понятието идентификатор, тя може да се представи във вида

⟨фактически параметър⟩ : : = ⟨низ⟩ | ⟨израз⟩ | ⟨идентификатор⟩.

След елиминирането на излишните формули и понятия от синтаксическото описание на АЛГОЛ-60, така както то е изложено в [1], кодировката на описанието заема 576 елемента, т. е. по-кратка е с 23% в сравнение с кодировката на оригиналното синтаксическо описание, изискващо 747 елемента. Както бе отбелязано по-горе, синтаксическото описание може да бъде съкратено значително по-силно, ако се премахнат рекурсивно определените понятия.

5. ПРИЛОЖЕНИЕ НА ОПИСАНАТА МЕТОДИКА ЗА ПРЕОБРАЗУВАНЕ НА СИНТАКСИЧЕСКОТО ОПИСАНИЕ НА АЛГОЛ-60

Разгледаната в настоящата работа методика за въвеждане на базис и за преобразуване на синтаксическото описание бе приложена посредством описаните алгоритми и програмна система за синтаксическото описание на АЛГОЛ-60. При това в синтаксическото описание предварително бяха отстранени излишните формули и понятия, както бе описано в т. 4.

Таблица 6

Понятие	Код	Участвували в кодираното синтаксическо описание	Участвували в б зиса	Участвували в синтаксическото описание след въвеждане на базиса
1	2	3	4	5
Празно	1	да	да	да
Основен символ	2	не — излишно	не	не
Буква	3	да	не	не
Цифра	4	да	не	не
Логическа стойност	5	да	да	да
Ограничител	6	не — излишно	не	не
Знак на операция	7	не — излишно	не	не
Знак на аритметична операция	8	не — излишно	не	не
Знак на операция за отношение	9	да	да	да
Знак на логическа операция	10	не — излишно	не	не
Знак на операция за следване	11	не — излишно	не	не
Разделител	12	не — излишно	не	не
Скоба	13	не — излишно	не	не
Описател	14	не — излишно	не	не
Спецификатор	15	не — излишно	не	не
Идентификатор	16	да	да	да
Цяло без знак	17	да	не	не
Цяло	18	да	не	не
Правилна дроб	19	да	не	не
Порядък	20	да	не	не
Десетично число	21	да	не	не
Число без знак	22	да	да	да
Число	23	не — излишно	не	не
Чист низ	24	да	не	не
Отворен низ	25	да	не	не
Низ	26	да	да	да
Израз	27	да	не	да
Идентификатор на променлива	28	не— синоним на 16	не	не
Проста променлива	29	не— синоним на 16	не	не
Индексен израз	30	не— синоним на 49	не	не
Списък от индекси	31	да	не	да
Идентификатор на масив	32	не— синоним на 16	не	не
Променлива с индекси	33	да	не	да
Променлива	34	да	не	да
Идентификатор на процедура	35	не— синоним на 16	не	не
Фактически параметър	36	да	да	да
Низ от букви	37	да	не	да
Ограничител на параметър	38	да	не	да
Списък от фактически параметри	39	да	не	да
Съвкупност от фактически параметри	40	да	не	да
Указател на функция	41	да	не	да
Знак на операция от тип събиране	42	да	да	да
Знак на операция от тип умножение	43	да	да	да
Първичен израз	44	да	не	да
Множител	45	да	не	да
Терм	46	да	не	да
Прост аритметичен израз	47	да	не	да
Условие	48	да	не	да
Аритметичен израз	49	да	не	да
Отношение	50	да	не	да
Първичен логически израз	51	да	не	да

Продължение на таблица 6

1	2	3	4	5
Вторичен логически израз	52	да	не	да
Логически едночлен	53	да	не	да
Логически терм	54	да	не	да
Импликация	55	да	не	да
Прост логически израз	56	да	не	да
Логически израз	57	да	не	да
Етикет	58	да	да	да
Идентификатор на ключ	59	не— синоним на 16	не	не
Указател на ключ	60	да	не	да
Прост именуващ израз	61	да	не	да
Именуващ израз	62	да	не	да
Основен оператор без етикет	63	да	не	да
Основен оператор	64	да	не	да
Безусловен оператор	65	да	не	да
Оператор	66	да	не	да
Край на съставен оператор	67	да	не	да
Начало на блок	68	да	не	да
Съставен оператор без етикет	69	да	не	да
Блок без етикет	70	да	не	да
Съставен оператор	71	да	не	да
Блок	72	да	не	да
Програма	73	да	не	да
Лява част	74	да	не	да
Списък на лява част	75	да	не	да
Оператор за присвояване	76	да	не	да
Оператор за преход	77	да	не	да
Празен оператор	78	не— синоним на 1	не	не
Оператор „ако"	79	да	не	да
Условен оператор	80	да	не	да
Елемент от списъка на цикъл	81	да	не	да
Списък на цикъл	82	да	не	да
Заглавие на цикъл	83	да	не	да
Оператор за цикъл	84	да	не	да
Оператор за процедура	85	не— синоним на 41	не	не
Описание	86	да	не	да
Списък на типа	87	да	не	да
Тип	88	да	да	да
Локален или собствен тип	89	да	не	да
Описание на типа	90	да	не	да
Долна граница	91	не — синоним на 49	не	не
Горна граница	92	не — синоним на 49	не	не
Гранична двойка	93	да	не	да
Списък на гранични двойки	94	да	не	да
Сегмент на масив	95	да	не	да
Списък от масиви	96	да	не	да
Описание на масив	97	да	не	да
Ключов списък	98	да	не	да
Описание на ключ	99	да	не	да
Формален параметър	100	не— синоним на 16	не	не
Списък на формални параметри	101	да	не	да
Съвкупност от формални параметри	102	да	не	да
Списък от идентификатори	103	да	не	да
Списък от стойности	104	да	не	да
Спецификация	105	да	не	да
Съвкупност от спецификации	106	да	не	да
Заглавие на процедура	107	да	не	да
Тяло на процедура	108	да	не	да
Описание на процедура	109	да	не	да

Таблица 7

Основен символ	Код	Участвува ли в базиса	Основен символ	Код	Участвува ли в базиса	Основен символ	Код	Участвува ли в базиса
a	301	не	1	354	не	own	407	да
b	302	не	2	355	не	Boolean	408	не
c	303	не	3	356	не		409	не
d	304	не	4	357	не	integer	410	не
e	305	не	5	358	не	real	411	да
f	306	не	6	359	не	array	412	да
g	307	не	7	360	не	switch	413	да
h	308	не	8	361	не	procedure	414	да
i	309	не	9	362	не	string	415	да
j	310	не	true	363	не	label	416	да
k	311	не	false	364	не	value	417	не
l	312	не	+	365	не	*)	418	да
m	313	не	−	366	не	код	419	не
n	314	не	×	367	не			
o	315	не	/	368	не			
p	316	не	÷	369	не			
q	317	не	↑	370	да			
r	318	не	<	371	не			
s	319	не	≤	372	не			
t	320	не	=	373	не			
u	321	не	≥	374	не			
v	322	не	>	375	не			
w	323	не	≠	376	не			
x	324	не	≡	377	да			
y	325	не	∪	378	да			
z	326	не	∨	379	да			
A	327	не	∧	380	да			
B	328	не	¬	381	да			
C	329	не	goto	382	да			
D	330	не	if	383	да			
E	331	не	then	384	да			
F	332	не	else	385	да			
G	333	не	for	386	да			
H	334	не	do	387	да			
I	335	не		388	да			
J	336	не		389	не			
K	337	не	10	390	не			
L	338	не	:	391	да			
M	339	не	;	392	да			
N	340	не	:=	393	да			
O	341	не	␣	394	не			
P	342	не	step	395	да			
Q	343	не	until	396	да			
R	344	не	while	387	да			
S	345	не	comment	308	не			
T	346	не	(	399	да			
U	347	не	)	400	да			
V	348	не	[	401	да			
W	349	не	]	402	да			
X	350	не	'	403	не			
Y	351	не	,	404	не			
Z	352	не	begin	405	да			
0	353	не	end	406	да			

*) Произволна последователност от основни символи, несъдържащи символа или ,

1 1+; 3, 4, 5, 6;
2 40 1;+;
3 63 1;+;
4 102 1;+;
5 104 1;+;
6 106 1;+;
1 5+;
2 51 1;+;
1 16+;3, 4, 5, 6, 7, 9, 11, 16, 26;
2 34 1;+;
3 36 1;+;
4 87 1;+;
5 101 1;+;
6 103 1;+;
7 40 1;
8 41 7;+;
9 393 1;
10 74 9;+;
11 388 1;14;
12 87 11;
13 87 12;+;
14 95 11;
15 95 14;+;
16 401 1; 20, 23;
17 31 16;
18 402 17;
19 33 18;+;
20 49 16;
21 402 20;
22 60 21;+;
23 94 16;
24 402 23;
25 95 24;+;
26 102 1;
27 392 26;
28 104 27;
29 106 28;
30 107 29;+;
1 22 +;
2 44 1;+;
1 26 +;
2 36 1;+;
1 27 +;
2 36 1;+;
1 31 +;
2 388 1;
3 49 2;
4 31 3;+;
1 33 +;
2 34 1;+;
1 34+;3, 4;
2 44 1;+;
3 51 1;+;
4 393 1;
5 74 4;+;
1 36+;
2 39 1,+;
1 39+;
2 38 1;
3 36 2;
4 39 3;+;
1 41+;3, 4;
2 44 1;+;
3 51 1;+;
4 63 1;+;
1 42+;
2 46 1;
3 47 2;+;
1 44+;
2 45 1;+;
1 45+; 3;
2 46 1;+;
3 370 1;
4 44 3;
5 45 4;+;
1 46 +;3;
2 47 1;+;
3 43 1;
4 45 3;
5 46 4; +;
1 47+;3, 6;
2 49 1;+;
3 42 1;
4 46 3;
5 47 4;+;
6 9 1;
7 47 6;
8 50 7;+;
1 48+; 4, 6, 10, 14;
2 65 1;
3 79 2;+;
4 84 1;
5 80 4;+;
6 47 1;
7 385 6;
8 49 7;
9 49 8;+;
10 56 1;
11 385 10;
12 57 11;
13 57 12;+;
14 61 1;
15 385 14;
16 62 15;
17 62 16;+;
1 49+;3, 4, 5, 8, 11;
2 27 1;+;
3 31 1;+;
4 81 1;+;
5 397 1;
6 57 5;
7 81 6;+;
8 391 1;
9 49 8;
10 93 9;+;
11 395 1;
12 49 11;
13 396 12;
14 49 13;
15 81 14;+;
1 50+;
2 51 1; +;
1 51+;
2 52 1;+;
1 52+;
2 53 1;+;
1 53+;3;
2 54 1;+;
3 380 1;
4 52 3;
5 53 4;+;
1 54+;3;
2 55 1;+;
3 379 1;
4 53 3;
5 54 4;+;
1 55 +;3;
2 56 1;+;
3 378 1;
4 54 3;
5 55 4; +;
1 56+;3;
2 57 1;+;
3 377 1;
4 55 3;
5 56 4;+;
1 57+;
2 27 1;+;
1 58+;3;
2 61 1;+;
3 391 1;6, 8, 10, 12;
4 64 3;
5 64 4;+;
6 71 3;
7 71 6;+;
8 72 3;
9 72 8;+;
10 80 3;
11 80 10;+;
12 84 3;
13 84 12;+;
1 60+;
2 61 1;+;
1 61+;
2 62 1;+;
1 62+;3;
2 27 1;+;
3 98 1;+;
1 63+;
2 64 1;+;
1 64+;
2 65 1;+;
1 65+;
2 66 1;+;
1 66+;3, 5;
2 108 1;+;
3 406 1;
4 67 3;+;
5 392 1;
6 67 5;
7 67 6;+;
1 68+;
2 392 1; 5;
3 86 2;
4 68 3;+;
5 67 2;
6 70 5;+;
1 69+;
2 71 1;+;
1 70+;
2 72 1;+;
1 71+;3;
2 65 1;+;
3 73 1;+;
1 72+;3;
2 65 1;+;
3 73 1;+;
1 74+;
2 75 1;+;
1 75+;4, 6;
2 74 1;
3 75 2;+;
4 49 1;
5 76 4;+;
6 57 1;
7 76 6;+;
1 76+;
2 63 1;+;
1 77+;
2 631;+;
1 79+;3;
2 80 1;+;
3 385 1;
4 66 3;
5 80 4;+;
1 80+;
2 66 1;+;
1 81+;
2 82 1;+;
1 82+;
2 388 1;
3 81 2;
4 82 3;+;
1 83+;
2 66 1;
3 84 2;+;
1 84+;
2 66 1;+;
1 88+;3, 4, 6;
2 89 1;+;
3 105 1;+;
4 411 1;
5 105 4;+;
6 413 1;8;
7 105 6;+;
8 107 6;

9 108 8;
10 109 9;+;
1 89+;4;
2 87 1;
3 90 2;+;
4 411 1;
5 96 4;
6 97 5;+;
1 90+;
2 86 1;+;
1 93+;
2 94 1;+;
1 94+;
2 388 1;
3 93 2;
4 94 3;+;
1 95+;
2 96 1;+;
1 96+;
2 388 1;
3 95 2;
4 96 3;+;
1 97+;
2 86 1;+;
1 98+;
2 388 1;
3 62 2;
4 98 3;+;
1 99+;
2 86 1;+;
1 101+;

2 38 1;+
3 16 2;
4 101 3;
1 103+;
2 388 1;
3 16 2;
4 103 3;+;
1 105+;
2 103 1;
3 392 2;
4 106 3;+;
1 106+;
2 105 1;
3 103 2;
4 392 3;
5 106 4;+;
1 109+;
2 86 1;+;
1 381+;
2 51 1;
3 52 2;+;
1 382+;
2 62 1;
3 77 2;+;
1 383+;
2 57 1;
3 384 2;
4 48 3;+;
1 386+;
2 34 1;
3 393 2;

4 82 3;
5 387 4;
6 83 5;+;
1 388+;
2 38 1;+;
1 399+;5, 8, 11, 14;
2 39 1;
3 400 2;
4 40 3;+;
5 49 1;
6 400 5;
7 44 6;+;
8 57 1;
9 400 8;
10 51 9;+;
11 62 1;
12 400 11;
13 61 12;+;
14 101 1;
15 400 14;
16 102 15;+;
1 400+;
2 37 1;
3 391 2;
4 399 3;
5 38 4;+;
1 405+;4;
2 86 1;
3 68 2;+;
4 67 1;
5 69 4;+;

1 407+;
2 88 1;
3 89 2;+;
1 411+;3;
2 105 1;+;
3 96 1;
4 97 3;+;
1 412+;3;
2 105 1;+;
3 16 1;
4 393 3;
5 98 4;
6 99 5;+;
1 413+;3;
2 105 1;+;
3 107 1;
4 108 3;
5 109 4;+;
1 414+;
2 105 1;+;
1 415+;
2 105 1;+;
1 416+;
2 103 1;
3 392 2;
4 104 3;+;
1 418+;
2 108 1;+;

Бяха направени експерименти и получени съответни резултати при различни базиси. Приведените резултати се отнасят за базиса с базисни понятия, посочени в табл. 6. В същата таблица са дадени всички понятия на АЛГОЛ-60 с техните кодове и указания, кои от тях са премахнати и по какви причини. В табл. 7 са приведени кодовете на основните символи на АЛГОЛ-60 и са указани кои от тях участвуват в базиса. В табл. 8 е дадена разпознавателната форма на синтаксическото описание на АЛГОЛ-60 при избрания базис, кодирана във вид на списък. Матрицата на наследниците е представена в табл. 9.

Приведените резултати могат да бъдат използвани като входи данни за езика АЛГОЛ-60 в синтаксически управляем транслатор.

1	27	024000000000000000000000
2	31	200000000000000000000000
3	33	66475776174034300600010
4	34	62475776174034300600010
5	36	004000000000000000000000
6	38	000000000000000000000000
7	39	004000000000000000000000
8	40	000000000000000000000000
9	41	62475776174000300600010
10	44	62435776000003006000000
11	45	62435776000000300600000
12	46	62415776000000300600000
13	47	62405776000000300600000
14	48	62401002214001700610010
15	49	62400000000000300600000
16	50	42400376000000000000000
17	51	42400176000000000000000
18	52	42400076000000000000000
19	53	42400076000000000000000
20	54	42400036000000000000000
21	55	42400016000000000000000
22	56	42400006000000000000000
23	57	42400000000000000000000
24	60	42400000600000000010000
25	61	42400000200000000010000
26	62	42400000000000000010000
27	63	00000000074000000000010
28	64	00000000034000000000010
29	65	00000000014000000000010
30	66	00000000004000000000010
31	67	00000000000000000000000
32	68	00000000036540000000010
33	69	00000000034240000000010
34	70	00000000034140000000010
35	71	00000000034040000000010
36	72	00000000034040000000010
37	73	00000000000000000000000
38	74	00000000174014000000010
39	75	00000000174014000000010
40	76	00000000174000000000010
41	77	00000000174000000000010
42	79	00000000014000400000010
43	80	00000000014000000000010
44	81	00000000000000100000000
45	82	00000000000000100000000
46	83	00000000014000020000010
47	84	00000000014000000000010
48	86	00000000000000000000000
49	87	00000000000000000000000
50	89	00000000000000011020000
51	90	00000000000000010000000
52	93	00000000000000000200000
53	94	00000000000000000200000
54	95	00000000000000000040000
55	96	00000000000000000040000
56	97	00000000000000010000000
57	98	00000000000000000010000
58	99	00000000000000010000000
59	101	00000000000000000002000
60	102	00000000000000000000000
61	103	00000000000000000000400
62	104	00000000000000000000000
63	105	00000000000000000000040
64	106	00000000000000000000040
65	107	00000000000000000000000
66	108	00000000000000000000000
67	109	00000000000000010000000
68	370	00000000000000000000000
69	377	00000000000000000000000
70	378	00000000000000000000000
71	379	00000000000000000000000
72	380	00000000000000000000000
73	381	42400176000000000000000
74	382	00000000174002000000010
75	383	62403002214001700610010
76	384	00000000000000000000000
77	385	00000000000000000000000
78	386	00000000014000060000010
79	387	00000000000000000000000
80	388	01000000000000000000000
81	391	00000000000000000000000
82	392	00000000000000000000000
83	393	00000000000000000000000
84	395	00000000000000000000000
85	396	00000000000000000000000
86	397	00000000000000000000000
87	399	62675776600000300611000
88	400	01000000000000000000000
89	401	00000000000000000000000
90	402	00000000000000000000000
91	405	00000000037740000000010
92	406	00000000000000000000000
93	407	00000000000000013020000
94	411	00000000000000010020140
95	412	00000000000000010004140
96	413	00000000000000010000144
97	414	00000000000000000000140
98	415	00000000000000000000140
99	416	00000000000000000000200
100	418	00000000000000000000010
101	1	00200000174000000001250
102	5	42400376000000000000000
103	9	00000000000000000000000
104	16	76575777774034304752430
105	22	62475776000000300600000
106	26	02400000000000000000000
107	37	00000000000000000000000
108	42	62405776000000300600000
109	43	00000000000000000000000
110	58	42400000674340420010010
111	88	00000000000000013020144

ЛИТЕРАТУРА

1. Алгоритмичен език АЛГОЛ-60 (преработено съобщение). С., 1968.
2. Королев, М. А., К. С. Кузьмин, С. С. Лавров, А. А. Летичевский, Г. К. Столяров, М. Р. Шура-Бура. Сообщение об алгоритмическом языке для экономических задач АЛГЭК. М., 1965.
3. Описание языка Алгамс. (Алгоритмы и алгоритмические языки, выпуск 3). М., 1968.
4. Bolliet, L. L'écriture des compilateurs. Rev. franç. traitement inform., 1966, No. 1, 47—73; No. 2, 129—158.

Постъпила на 30. XII. 1969 г.

ПРЕОБРАЗОВАНИЕ СИНТАКСИСА АЛГОРИТМИЧЕСКИХ ЯЗЫКОВ В СВЯЗИ С ПОСТРОЕНИЕМ СИНТАКСИЧЕСКИ УПРАВЛЯЕМЫХ ТРАНСЛЯТОРОВ

Петр Бырнев, Валентин Томов и Маргарита Бырнева

(Резюме)

В работе излагается алгоритм формального преобразования синтаксического описания данного алгоритмического языка в вид, удобный для пользования синтаксически управляемым транслятором. В то же время алгоритм упрощает выбор базисной системы основных понятий и символов.

Алгоритм приложен для языка АЛГОЛ-60. Выбрана система базисных понятий и символов, приведены таблицы преобразования синтаксического описания.

TRANSFORMATION OF THE SYNTAX OF SOME ALGORITHMIC LANGUAGES IN CONNECTION WITH THE CREATION OF SYNTACTICALLY CONTROLLABLE TRANSLATORS

Petâr Bârnev, Valentin Tomov and Margarita Bârneva

(Summary)

The paper contains an algorithm for formal transformation of the syntactic description of a given algorithmic language in order to make it usable by a syntactically controllable translator. At the same time the algorithm under consideration simplifies the choice of a basic system of main concepts and symbols.

The algorithm is applied to ALGOL-60. A system of basic concepts and symbols is chosen and tables containing the transformed syntactic description are given.