

ЕДНО ПРИЛОЖЕНИЕ НА ЕКВИВАЛЕНТНИТЕ ПРЕОБРАЗУВАНИЯ НА ПРОГРАМИ

Маргарита Н. Спирилонова, Валентина Р. Пелова

Проблемът за съвместимостта между елементи на математическото осигуряване, създадени за различни автоматични сметачни машини, има важно практическо значение. Типичен пример е използването на програми, написани на входния език на даден транслатор от някакъв алгоритмичен език с друг транслатор от същия алгоритмичен език. В такива случаи се налага предварително преобразуване на използваниите програми в съответствие с входния език на наличния транслатор. При това ръчното преобразуване е източник на грешки и е свързано със загуба на време от страна на програмиста. Естествен е стремежът това да се извърши автоматично, т. е. от програма-преобразувател, която да се използува при необходимост от подобни преобразувания. В такъв случай на транслация се подава преобразуваната от нея програма.

Този проблем може да се разглежда и в друг аспект. Ако при създаден транслатор от някой алгоритмичен език се окаже желателно да се разшири апаратът на входния език, внасянето на съответните допълнения и изменения в транслатора е често сложна и трудоемка задача. Друга възможност за разширяване на изразните средства на входния език е предварителното (преди транслация) преобразуване на програмите, съдържащи „добавените“ конструкции на езика, в програми, еквивалентни на съответните изходни, но съдържащи само конструкции на входния език на наличния транслатор.

В настоящата работа се разглежда конкретен пример на програма-преобразувател с описаното по-горе предназначение и се привеждат някои общи съображения, свързани с подобни реализации.

В сектор „Математическо осигуряване“ на ЕЦНПКММ е създаден транслатор от езика ФОРТРАН за машината „Минск-32“. Входният език на транслатора, наречен ФОР32 [1], е много близък до базисния ФОРТРАН и може да се разглежда като подмножество на ФОРТРАН IV [2], най-разпространения вариант на езика ФОРТРАН. Тъй като в много изчислителни центрове в света са създадени програми на ФОРТРАН IV с практическа стойност, които могат да се използват, има смисъл да се обсъди дали с една програма-преобразувател в горния смисъл може да се извърши лесен преход от ФОРТРАН IV към ФОР32.

В приложението в края на статията са дадени различията между двата езика. Преобразуванията на някои от тях са тривиални (например 1, 2, 4, 9, 10 и др.), а на други — доста трудоемки и практически безсмислени (например 24 и 25). Следователно създаването на програма-

преобразувател за пълния списък от различия е нецелесъобразно. Ясно е, че има смисъл да се осъществява преход между два транслатора с помощта на такава програма, ако различията между входните езици са в разумни граници. А в някои случаи е подходящо такава програма да обработва подсписък на пълния списък от различия, като се изключат някои рядко срещащи се конструкции или такива, за които отнапред е ясно, че няма да се срещат в програми за определен клас задачи.

Но една такава програма-преобразувател може да се направи и просто, за да се добавят някои конструкции към входния език на даден транслатор, като се цели разширение на езика, а не непременно осигуряване на съвместимост (пълна или частична) между транслатори от различни версии на един и същ език.

За илюстрация на описаната идея беше разработена програма, която преобразува логическите елементи, разрешени във ФОРТРАН IV, в конструкции, допустими във ФОР32 (езикът ФОР32 не съдържа логически данни и оператори). По такъв начин програма, написана на разширенето на ФОР32 с логическите елементи на ФОРТРАН IV, може да се преобразува в програма на ФОР32.

Програмата-преобразувател е написана на езика за символично кодиране на машината „Минск-32“ [3]. Тя преобразува логическите елементи по следния начин:

Логически елементи в изходната програма

I. Логически константи:

.TRUE.
.FALSE.

II. Спецификация:

LOGICAL

III. Операции за отношение:

A.GT.B
A.GE.B
A.LT.B
A.LE.B
A.EQ.B
A.NE.B

IV. Логически операции:

.NOT.A
A.AND.B

Еквивалентни конструкции в преобразуваната програма

Числови константи:

1
0

Спецификация:

INTEGER

Отношението изцяло се заменя с една цяла променлива, а преди оператора, съдържащ това отношение, се записва обръщение към подпрограма, която дава на цялата променлива стойност единица в случай, че отношението има стойност „истина“, и нула — в противен случай. Целите променливи се генерират в програмата и имат вида Ib111, Ib112, . . . Ib999

Аритметични операции:

1-A
A * B

$$\begin{aligned} A \text{.OR.} B = \\ .\text{NOT}.(\text{.NOT.} A \text{.AND..} \text{NOT.} B) \end{aligned}$$

V. Логически оператор за присвояване:

$$U := b,$$

където U е логическа променлива, b — логически израз. За по-лесно разпознаване програмата-преобразувател изисква да се използува знак за присвояване „ $: =$ “.

VI. Логически оператор IF (също за по-лесно разпознаване се изисква името на оператора да бъде IFL):

$$\text{IFL}(b) s.$$

Тук b е логически израз, s — оператор, който за разлика от възприетите ограничения във ФОРТРАН IV може да бъде и DO или IFL.

VII. Форматна логическа спецификация:

$$Lw$$

Пример. Операторът

$$X := B \text{.GT.} C \text{.AND..} \text{NOT.} D$$

ще бъде заменен с

$$\begin{aligned} \text{CALL GT (lb111, B, C),} \\ X = lb111 * (1 - D), \end{aligned}$$

като подпрограмата GT се добавя към основната (преобразувана) програма. Тя има вида

```
SUBROUTINE GT (I, Q, R)
IF (Q=R) 1, 1, 2
1 I=0
    GOTO 3
2 I=1
3 RETURN
END
```

Програмата-преобразувател не извършва синтактичен контрол на изходната програма. Проверява се само балансирането на скобите в логи-

$$\begin{aligned} 1 - (1 - A) * (1 - B) = \\ A + B - AB \end{aligned}$$

Оператор за присвояване:

$$U = a,$$

където U е цяла променлива, а — аритметичният израз, получен от преобразуването на логическия израз b .

Аритметичен оператор IF, последван от етикетирания оператор s от логическия IF:

$$\text{IF}(a=1) \text{ et1, et2, et1}$$

$$\text{et2 s}$$

et1 продължение на програмата, където a е аритметичният израз, получен след преобразуването на логическия израз b . Етикетите et1 и et2 се генерираят в програмата.

Форматна цяла спецификация:

$$Iw$$

ческите изрази. При откриване на грешка се печата съобщение. Пълен контрол се извършва от транслятора ФОР32, на който се подава преобразуваната програма. За конструкции с логически елементи, които са неправилно написани и поради това не са разпознати и преобразувани, също ще се индицира грешка (тъй като са чужди на ФОР32).

Преобразуването се извършва на три етапа, като на всеки етап програмата се преглежда по един път.

На първия етап програмата, подлежаща на преобразуване, се въвежда, изчиства се от интервалите, съставя се таблица на етикетите (която се използва при генерирането на етикети), разпознават се конструкции, които трябва да се пресобразуват (отбелязват се със служебен символ), контролира се балансирането на скобите в логическите изрази и при необходимост се индицират грешки.

На втория етап се извършва преобразуване на вече разпознатите конструкции, като останалият текст се преписва без промяна. Ще се спрем на алгоритъма за преобразуване на логически изрази, който заема съществено място в този етап. Той използва 2 стека (реализирани програмно): ST 1 (стек на операциите) и ST 2 (стек на аргументите), като редовете на ST 2 са с променлива дължина и за отделянето им се използва таблицата УКАЗ, всеки ред на която съдържа началото и края на съответния ред на ST 2. При четене на съмволите на даден логически израз знаците за логически операции и отварящите скоби се внасят в ST 1 в реда на срещането им, при това със своя приоритет: 1 за (, 2 за .OR., 3 за .AND., 4 за .NOT.). В ST 2 се записват аргументите на логическите операции, също по реда на срещането им, като при срещане на отношения вместо тях в ST 2 се записват генерираните цели променливи, с които те се заменят. При срещане на затваряща скоба се спира записването в стековете и се „изпълняват“ операциите, записани в ST 1 до съответната отваряща скоба. Под „изпълнение“ се разбира следното: за поредната операция (намираща се в последната клетка на ST 1) аргументите се вземат от последните една или две клетки (в зависимост от броя на аргументите) на ST 2 и с тях се записва аритметичната операция, с която се замения съответната логическа. Този резултат се записва в ST 2 на мястото на аргументите във вид на аритметичен израз. Аналогични действия се извършват при срещане на логическа операция с приоритет, по-нисък или равен на приоритета на операцията, записана в последния ред на ST 1 (т. е. внасяне в ST 1 става само при срещане на операция с по-висок от текущия приоритет). Когато целият израз е прочетен, се чете отново ST 1 и последователно се изпълняват записаните в него операции, ако все още има такива. Когато ST 1 се изпразни, последният ред на ST 2 съдържа целия преобразуван израз и с помощта на съответния ред от таблицата УКАЗ той се пренася в програмата.

На третия етап програмата се подготвя за подаване на транслятора ФОР32: пред всеки оператор се поставя името на програмата, номерът на страницата, номерът на реда. Ако някой от добавените оператори е по-дълъг от 80 символа, се формира нов ред — продължение. Програмата се записва на магнитна лента, чието име се задава от пулта за управ-

ление. След програмата на същата магнитна лента се записват подпрограмите за пресмятане на отношения, които са добавени към програмата. Записаната на магнитна лента програма и подпрограмите са готови за работа с транслятора ФОР32.

Преобразуваната програма се извежда и на печат, като след нея се отпечатват добавените подпрограми. Имената на тези подпрограми се отпечатват и на пулта като указание за подготовка на заявки за трансляция.

Програмата-преобразувател поставя следните изисквания:

1. Изходната програма да бъде написана на езика ФОР32, разширен с логическите елементи на ФОРТРАН IV, както са описаны по-горе.
2. В програмата да няма променливи от вида $Ibaaa$, където $aaa \in [111, 999]$, аaaa цяло.

3. Аритметичните изрази, участващи в отношенията, да не съдържат скоби. (Ако се налага използването на скоби, да се извърши присвояване на друга променлива преди оператора, съдържащ съответното отношение. Това ограничение се въвежда за опростяване на алгоритъма, преобразуващ логически изрази, който следи скобите, отнасящи се за реда на изпълнение на логическите операции.)

Описаната програма е съставена в сектор „Математическо осигуряване“ на ЕЦНПКММ като дипломна работа на Валентина Р. Пелова през учебната 1973/74 г. с ръководител и. с. Маргарита Н. Спиридонова. Тази програма е пример за реализация на изложената в началото на статията идея и може да се използува за практическа работа.

ПРИЛОЖЕНИЕ

КОНСТРУКЦИИ НА ЕЗИКА ФОРТРАН IV, НЕРАЗРЕШЕНИ В ЕЗИКА ФОР32

1. Машабен множител
2. Оператор FORMAT с порече от 2 вложения на скоби
3. Масиви с размерност, по-голяма от 3
4. Оператор CALL EXIT
5. Оператор ASSIGN i TO m
6. Оператор GO TO m (x_1, x_2, \dots, x_n)
7. Оператор RETURN i
8. Оператор ENTRY
9. Оператор PAUSE 'текст'
10. Оператори READ b, списък
PUNCH b, списък
PRINT b, списък
11. Логически елементи
12. Параметри END и ERR в оператор READ
13. Оператори CALL DUMP и CALL PDUMP
14. Оператори CALL OVEREL (j), CALL DVCHK (j),
CALL SLITE (j), CALL SLITET (i, j)
15. Оператор NAMELIST
16. Оператор за спецификация IMPLICIT
17. Определяне на спецификации, съдържащи * s
18. Спецификации на формат T и G
19. Предаване на име като параметър на подпрограма

20. Оператори FIND, DEFINE FILE
21. Различия в библиотеките от вградени функции
22. Етикетиран оператор COMMON
23. Задаване на начални стойности: DATA, BLOCK DATA
24. Двойна точност
25. Комплексни числа

ЛИТЕРАТУРА

1. FOR32. Транслатор от ФОРТРАН за машината „Минск-32“. Ръководство за програмиста. Бътрешико издание на Института по математика с изчислителен център при БАН. С. 1973.
2. Джермейн, К.: Программирование на IBM/360. М., Мир, 1971.
3. Кушнерев, Н., Неменман, М., Цагельский, В.: Программирование для ЭВМ „Минск-32“. М., Статистика, 1972.

Постъпила на 14. XII. 1974 г.

AN APPLICATION OF THE EQUIVALENT TRANSFORMATIONS OF PROGRAMS

M. N. Spiridonova, V. R. Pelova

(SUMMARY)

An approach to the problem of compatibility between compilers for different versions of an algorithmic language is considered. An equivalent transformation of the programs before compiling is proposed. A transformation of this kind is suggested as a method for extending a language without modifying the compiler. A program, well illustrating these ideas, is described. It transforms the logical elements of FORTRAN IV into constructions of FOR32 (a version of Basic FORTRAN for "Minsk-32" computer). In such way it permits the programmer to use logical data and operators in FOR32 language. The program (as the compiler for FOR32) is made in the Institute of Mathematics and Mechanics of the Bulgarian Academy of sciences.