文章编号:1007－6492(1999)01－0072－03

产生式外部形态对算术表达式翻译的影响及其解决办法

王文义[1],张行进[1],王若雨[2]

(1.郑州工业大学电气信息工程学院,河南 郑州 450002;2.河南省电力职工大学,河南 郑州 450051)

摘 要:一般地,语法分析技术中的算符优先分析方法适用于对算术表达式的翻译,但在某些文法中产生式的外部形态却往往会影响到这种翻译结果的正确性.对这种情况进行了具体的分析,并提出了解决方法.根据问题产生的原因,采取的主要措施之一就是必需对给定文法中的某些产生式的外部态加以改变.

关键词:算符优先分析;文法;产生式;外部形态

中图分类号:TP 314 **文献标识码**:A

0 引言

在编译技术的语法分析中,有一类文法称为算术表达式文法[1,2],使用算符优先分析方法可以较为方便地完成对该类表达式的前期翻译工作.该方法的基本内容包括对算符优先文法的判定以及相关的算法设计等.整个结构由总控程序、优先关系表、存放待分析对象的输入缓冲区和一个分析栈组成.其各部分的关系如图 1 所示.

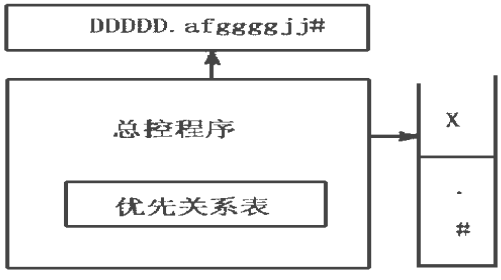

图 1 分析器数据结构

1 First vt ,Last vt 集合和优先关系的算法

为实现计算机对算术表达式的分析,首先应定义两个集合:

$$\text{First vt}(R)=\{a \mid R\overset{+}{\Rightarrow}a\cdots \quad 或 \quad R\overset{+}{\Rightarrow}Qa\cdots\}$$

$$\text{Last vt}(R)=\{b \mid R\overset{+}{\Rightarrow}\cdots b \quad 或 \quad R\overset{+}{\Rightarrow}\cdots bQ\}$$

其中,$a,b\in V_t$,$R,Q\in V_n$.

1.1 First vt 和Last vt 集合的构造算法

可使用下述(1),(2)条构造First vt 集合,使用(3),(4)条构造Last vt 集合:

(1) 若有形如 $R\rightarrow a\cdots$ 或 $R\rightarrow Qa\cdots$ 的产生式,则让 $a\in$First vt(R);

(2) 若有 $a\in$First vt(Q) 且存在形如 $R\rightarrow Q\cdots$ 的产生式,则让 $a\in$First vt(R);

(3) 若有形如 $R\rightarrow\cdots b$ 或 $R\rightarrow\cdots bQ$ 的产生式,则让 $b\in$Last vt(R);

(4) 若有 $b\in$Last vt(Q) 且存在形如 $R\rightarrow\cdots Q$ 的产生式,则让 $b\in$Last vt(R).

1.2 优先关系的构造算法

逐个扫描文法中各个产生式:

(1) 若有形如 $P\rightarrow\cdots ab\cdots$ 或 $P\rightarrow\cdots aRb\cdots$ 的产生式,则让'a'$\doteq$'b';

(2) 若有形如 $P\rightarrow\cdots aR\cdots$ 的产生式,则让'a'$<\cdot$First vt(R) 中的每个终结符号'b';

(3) 若有形如 $P\rightarrow\cdots Rb\cdots$ 的产生式,则让 Last vt(R) 中的每个终结符号'a'都$\cdot>$'b'.

2 一个实例文法及其反映的问题

2.1 实例文法和它的优先关系表

已知文法 $G(E)$:

$E\rightarrow E+T\mid T$

$T\rightarrow T*F\mid F$

$F\rightarrow F\uparrow B\mid B$

收稿日期:1998－06－05;修订日期:1998－09－20

作者简介:王文义(1947—),男,河南省洛阳市人,郑州工业大学副教授,主要从事软件工程数据库应用方面的研究.

$B \rightarrow (E) \mid i$

其中，$V_n = \{E, T, F, B\}$；$V_t = \{+, *, \uparrow, (,), i\}$，'$\uparrow$' 表示乘幂符号.

根据 First vt 和 Last vt 集合的算法可以确定：

(1) First vt(B) = {(, i}；

First vt(F) = {$\uparrow$, (, i}；

First vt(T) = {*, $\uparrow$, (, i}；

First vt(E) = {+, *, $\uparrow$, (, i}；

(2) Last vt(B) = {), i}；

Last vt(F) = {$\uparrow$,), i}；

Last vt(T) = {*, $\uparrow$,), i}；

Last vt(E) = {+, *, $\uparrow$,), i}.

构造优先关系表：

因为产生式中存在形如'(E)'的子串，故有'(' $\doteq$ ')'；

因为产生式中存在形如：'E +'，'T *'，'$F \uparrow$'和'E)'的子串，故分别有：

Last vt(E) 中的每一个终结符号都分别 $\gtrdot$ '+'并且也分别 $\gtrdot$ ')'；

Last vt(T) 中的每一个终结符号都分别 $\gtrdot$ '*'；

Last vt(F) 中的每一个终结符号都分别 $\gtrdot$ '$\uparrow$'；

又因为产生式中存在形如：'+ T'，'* F'，'$\uparrow B$'和'(E'的子串，故分别有：

'+' $\lessdot$ First vt(T) 中的每一个终结符号；

'*' $\lessdot$ First vt(F) 中的每一个终结符号；

'$\uparrow$' $\lessdot$ First vt(B) 中的每一个终结符号；

'(' $\lessdot$ First vt(E) 中的每一个终结符号；

于是可以得到文法 $G(E)$ 的优先关系表 1.

表 1　优先关系表

左符号	右符号					
	+	*	$\uparrow$	(	)	i
+	$\gtrdot$	$\lessdot$	$\lessdot$	$\lessdot$	$\gtrdot$	$\lessdot$
*	$\gtrdot$	$\gtrdot$	$\lessdot$	$\lessdot$	$\gtrdot$	$\lessdot$
$\uparrow$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\lessdot$	$\gtrdot$	$\lessdot$
(	$\lessdot$	$\lessdot$	$\lessdot$	$\lessdot$	$\doteq$	$\lessdot$
)	$\gtrdot$	$\gtrdot$	$\gtrdot$		$\gtrdot$	
i	$\gtrdot$	$\gtrdot$	$\gtrdot$		$\gtrdot$	

2.2　存在问题

由上述实例可以看出，该文法既没有形如 $R \rightarrow \varepsilon$（空字）的产生式，也不存在有两个非终结符号相连的产生式右部[3]. 同时由表 1 也可以看出，文法中任何一对终结符号之间都满足算符优先文法的规定，即每对终结符号之间要么具有 3 种优先关系 $\lessdot$，$\gtrdot$，$\doteq$ 之一，要么就什么也没有（针对出错情况）. 在语法分析中，对算术表达式类的翻译[4]，一个最基本的原则是要满足起码的优先顺序要求，如：

'*' $\gtrdot$ '*'；'+' $\gtrdot$ '+'；'+' $\lessdot$ '*'；'*' $\lessdot$ '$\uparrow$'

即把 $a*b*c$ 解释为 $(a*b)*c$，把 $a+b+c$ 解释为 $(a+b)+c$，把 $a+b*c$ 解释为 $a+(b*c)$，把 $a*b\uparrow c$ 解释为 $a*(b\uparrow c)$. 可见对于这些规则，上述实例是满足的.

应特别注意的是，对于如 $a\uparrow b\uparrow c$ 的乘幂运算，其正确解释应为 $a\uparrow(b\uparrow c)$，但在上述优先关系表中却出现了有悖于实际的'$\uparrow$' $\gtrdot$ '$\uparrow$'的情况，即把乘幂运算解释成了左结合规则. 这也就是说，在翻译诸如 3^{2^3} 时，计算结果将是 729 而不是 6561!

因此可以说，仅对文法形式加以简单约束以及根据由 First vt 和 Last vt 两个集合得到的算符优先关系表在某些情况下尚不能正确地进行算术表达式翻译.

2.3　出现'$\uparrow$' $\gtrdot$ '$\uparrow$'的原因

已知 $a, b \in V_t$，$R, Q \in V_n$，在构造算符优先关系时，从语法树的构成可以看出，由于语法分析的对象通常是句子，而句子是由叶子组成的，它们具有不可再推导（生成子树）性. 对于文法中的非终结符 R，它代表语法树的某个父节点，具有可再推导性. 从语法树的层次来看，语法分析的顺序是从当前的最左子树向根部前进的，也就是说在空间上是自下而上的. 但对待分析对象的扫描顺序在时间上却是由左至右进行的. 如对图 2，是先扫描 a，再扫描 $Qb\cdots$（或 $b\cdots$），于是才有'a' $\lessdot$ 'b'的优先关系；对图 3，是先扫描 $\cdots Qa$（或 $a\cdots$），再扫描 $b\cdots$，于是才有'a' $\gtrdot$ 'b'的优先关系.

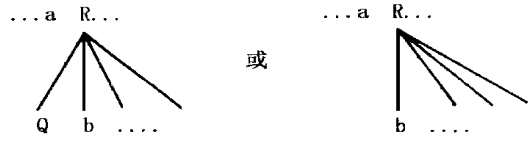

图 2　'a' $\lessdot$ 'b' 的优先关系

图 3 'a'$\gtrdot$'b' 的优先关系

2.4 关于递归产生式 $F \rightarrow F \uparrow B$

对直接左递归的产生式 $F \rightarrow F \uparrow B$，正是由于产生式右部中'$F$'和'$\uparrow$'所处的位置，导致了 Lastvt($F$) 集合中的每一个终结符号'a'都$\gtrdot$'$\uparrow$'.因为 Lastvt($F$) $=\{\uparrow,), i\}$，所以得出了'$\uparrow$'$\gtrdot$'$\uparrow$'的错误结果.可以说文法符号位置的不当是产生这个错误的根本原因.

3 改变产生式外部形态，消除错误源

通过改变产生式外部形态，能够消除上述错误.正确的 $G(E)$ 文法应为如下形式：

$E \rightarrow E + T \mid T$

$T \rightarrow T * F \mid F$

$\boldsymbol{F \rightarrow B \uparrow F \mid B}$

$B \rightarrow (E) \mid i$

注意原来文法中的产生式 $F \rightarrow F \uparrow B$ 已被改为 $F \rightarrow B \uparrow F$（**文法中以粗体显示**）.由算符优先关系的算法知道，上述改动并不影响文法的 Firstvt 集合和 Lastvt 集合[3]，这主要是由于算符优先分析方法忽略了文法中非终结符号之间优先关系的缘故.可以看出，正是由于'F'和'B'互换了位置，才使'$\uparrow$'$\lessdot$Firstvt(F) 中的每一个终结符号，于是原来错误的优先关系'$\uparrow$'$\gtrdot$'$\uparrow$'就变成了符合常规的'$\uparrow$'$\lessdot$'$\uparrow$'.

4 结束语

由上述讨论可知，在算符优先分析方法中，对所给文法仅规定没有右部为 ε 的产生式和右部也不存在两个相继非终结符号的产生式这两条还是不够的，还必须对某些具有右结合性运算符号所在的递归产生式的外部形态作出进一步的限制.本文所给文法就是一个典型的例子.若不加上述限制，将可能会在使用算符优先分析方法分析某些句子时得出错误的结论.

参考文献

[1] 陈火旺，钱家骅，孙永强.编译原理[M].北京：国防工业出版社，1994.

[2] 姜文清.编译技术原理[M].北京：国防工业出版社，1994.

[3] 霍普克罗夫特J E，厄尔曼J D.形式语言及其与自动机的关系[M].莫绍揆，段祥，顾秀芬译.北京：科学出版社，1979.

[4] AHO A V，ULLMAN J D.Principles of Compiler Design[M].New York：Addison—Wesley Publishing Company，1977.

[5] AHO A V，SETHI R，ULLMAN J D.Compilers：Principles，Techniques，and Tools[M].New York：Addison—Wesley Publishing Company，1986.

Influence of Production's External Form on the Translation of Arithmetic Expression and Its Solution

WANG Wen—yi[1]，ZHANG Xing—jin[1]，WANG Ruo—yu[2]

(1.College of Electrical & Information Engineering，Zhengzhou University of Technology，Zhengzhou 450002，China；2.Henan University of Electric Power and Workers，Zhengzhou 450051，China)

Abstract：Generally，the operator precedence parsing is suitable for the translation of arithmetic expressions in syntactic analysis technology，but the production's external form may influence on the translation result correctness in some grammar.This paper discusses the above mentioned problems and gives a solution.According to the reasons causing the above—said problems，one of main method used is that we have to change the external form for some productions in the grammar.

Key words：operator precedence parsing；grammar；production；external form