

# 杂凑法在 ALGOL 121中的应用

巫 张 英\*

( 计算机科学系 )

国产DJS—21型电子计算机(简称为121机)原配有北京大学等单位设计的ALGOL编译系统,1972年该系统投入使用后,各用户分别作了不同程度的改进。后来,四川大学和中山大学合作,为增强系统的功能和提高运行效率而重建了该系统,并于1978年投入运行,定名为ALGOL121。

ALGOL121系统的特点是运行快(运算时间缩短30%以上),功能较强(如查错、改错、库过程等),系统大部分用基本汇编语言写成,可读性好,易于维护。但美中不足的是,该系统的编译时间没有减少。编译时间的相当部分是用在词法分析(也称第一次扫描)中。由于在ALGOL121的词法分析中,对标识符采用了线性查造名字特性表的方法,估计词法分析时间的百分之六十以上用于查造名字特性表。

为了减少编译时间,我们采用“对分层语言应用杂凑法的方案”<sup>[1]</sup>一文所提出的方法,结合ALGOL121编译系统的具体情况,重新设计了ALGOL121第一次扫描的框图。在新的设计中,应用了杂凑法查造名字特性表的技术,并且用基本汇编语言JHY 121(中山大学软件教研室,1979)。编写出新的第一次扫描程序。

ALGOL语言是嵌套结构的,应用杂凑法查造名字特性表比较复杂,在此不作详述。

## 举 例 说 明

在原ALGOL121的第一次扫描中,对标识符是采用线性查表法,所谓线性查表法,就是对于一个标识符,先查保留字,然后查本层的名字特性表,逐个比较。

为了说明问题,请看下面一个很简单的程序的查表情况。

```
‘BEGIN’ ‘REAL’ AI;
```

```
AI: = 5,1;
```

```
...
```

```
‘END’
```

假定其中的AI不是保留字。

当第一次扫描处理到程序第二行应用性出现的标识符AI时,要先查保留字,而保留字有90个,按线性查表法,要逐个进行查询。每查询一个保留字均要执行5条指令,因此,全部保留字查完,就要执行450条指令。另外,对AI查本层的名字特性表又需执

\* 肖金声老师对本工作进行了指导和协助。

行6条指令。故查到AI的词码时,至少要执行456条指令。

在新设计的第一次扫描中,对标识符采用杂凑法查表。对于上述程序而言,当第一次扫描处理到程序第二行应用性出现的标识符AI时,对AI求其杂凑码需要执行10条指令,按AI的杂凑码取其对应的杂凑表元素需要执行3条指令,对取出的杂凑表元素之值进行判断需要执行11条指令,查本层的名字特性表需执行8条指令,因此,查到AI的词码时,只需执行32条指令。

虽然,上述的程序很简单,不能代表所有的情况,但却充分体现出应用杂凑法查表的优越性。

对于其它一般的情形,在新设计的第一次扫描中可能需要多做一些工作。譬如,标识符不同而杂凑码相同时,就需要进行拉链(杂凑同码链),查表时就需要增加沿杂凑同码链查询的一些指令。另外,在退层时,比原第一次扫描增加清名字特性表及对杂凑表进行还原等工作,又需要执行一些指令。尽管如此,应用杂凑法以后,第一次扫描的效率还是提高一倍以上。

### 小 结

由于第一次扫描的重新设计,标识符的内部表示与原来不同,因此,第二次扫描及第三次扫描都需相应作些修改。新改好的ALGOL121编译系统,已于1980年3月开始在中山大学计算中心使用。

通过前一段时间使用的实践证明,应用杂凑法查造名字特性表的技术以后,作词法分析的时间普遍缩短55%以上,效率提高一倍多。

下面介绍几个用户的程序的使用效果,只是对源程序扫描而言。这里所列出的数据,是由本校计算中心组织有关专业人员在现场测试出来的。

| 源程序<br>(行) | 原一扫所需<br>时间(秒) | 新一扫所需<br>时间(秒) | 减少时间比值 | 效率比原来<br>提高的倍数 |
|------------|----------------|----------------|--------|----------------|
| 163        | 23.5           | 9.5            | 60%    | 1.47           |
| 345        | 38.5           | 17             | 56%    | 1.26           |
| 357        | 71             | 28             | 61%    | 1.54           |
| 615        | 121            | 47             | 61%    | 1.57           |
| 678        | 87.5           | 38.5           | 56%    | 1.27           |
| 738        | 95.5           | 42             | 56%    | 1.27           |

### 参 考 文 献

- [1] 肖金山, 对分层语言应用杂凑法的方案, 计算机学报, 1979年第二期。