

编译原理作业 (5)

姓名: _____ 学号: _____

2024 年 05 月 05 日

请独立完成作业，不得抄袭。

若得到他人帮助，请致谢。

若参考了其它资料，请给出引用。

鼓励讨论，但需独立书写解题过程。

允许并鼓励使用 ChatGPT 等工具，但需明确说明使用方式。

Semantics of Context-Free Languages

by

DONALD E. KNUTH

California Institute of Technology

ABSTRACT

“Meaning” may be assigned to a string in a context-free language by defining “attributes” of the symbols in a derivation tree for that string. The attributes can be defined by functions associated with each production in the grammar. This paper examines the implications of this process when some of the attributes are “synthesized”, i.e., defined solely in terms of attributes of the *descendants* of the corresponding nonterminal symbol, while other attributes are “inherited” i.e., defined in terms of attributes of the *ancestors* of the nonterminal symbol. An algorithm is given which detects when such semantic rules could possibly lead to circular definition of some attributes. An example is given of a simple programming language defined with both inherited and synthesized attributes, and the method of definition is compared to other techniques for formal specification of semantics which have appeared in the literature.

1 作业（必做部分）

题目 1

考虑如下文法 G ,

$$S \rightarrow L.L \mid L$$

$$L \rightarrow LB \mid B$$

$$B \rightarrow 0 \mid 1$$

请设计语法制导的翻译方案，计算 S 产生的二进制小数对应的十进制值。如， $101.101_2 = 5.625_{10}$ 。你需要自行定义合适的属性。

解答:

• 方法一: 仅使用综合属性

* 4.6 用 S 的综合属性 val 给出下面文法中 S 产生的二进制数的值。例如, 输入 101.101 时, $S.val = 5.625$ 。要求仅用综合属性来计算 $S.val$ 。

$$\begin{aligned} S &\rightarrow L.L \mid L \\ L &\rightarrow LB \mid B \\ B &\rightarrow 0 \mid 1 \end{aligned}$$

答 给文法符号 B, L 和 S 以综合属性 val , 另外再给 L 一个表示其长度的综合属性 $length$ 。所求的语法制导定义如下:

$$\begin{aligned} S &\rightarrow L_1.L_2 & S.val &= L_1.val + L_2.val / 2^{L_2.length} \\ S &\rightarrow L & S.val &= L.val \\ L &\rightarrow L_1B & L.val &= L_1.val \times 2 + B.val; \quad L.length = L_1.length + 1 \\ L &\rightarrow B & L.val &= B.val; \quad L.length = 1 \\ B &\rightarrow 0 & B.val &= 0 \\ B &\rightarrow 1 & B.val &= 1 \end{aligned}$$

分析 在这个问题中比较困难的是有小数点的情况。在 $S \rightarrow L_1.L_2$ 中, L_1 和 L_2 分处小数点的两侧, 为了避免使用继承属性, 在此对 L_1 和 L_2 使用同样的计算方法, 而在把它们值相加之前, 将 L_2 的值除以 $2^{L_2.length}$ 。要想避免这样的指数运算和除法, 就要对 L_1 和 L_2 分别使用不同的方式求值。这时需要用到继承属性, 因为如果 L 没有继承属性的话, 到了 L 的产生式中就无法区分该 L 究竟是出现在小数点的左边还是右边。习题 4.19 将给出使用继承属性的解决办法。

• 方法二: 使用综合属性和继承属性

习题 4.19 使用综合属性

* 4.19 像习题 4.6 那样, 用语法制导定义决定 $S.val$ 。在该定义中, B 的唯一综合属性是 c , 它给出由 B 产生的那一位对最终值的贡献。例如, 101.101 的最左一位和最右一位对值的贡献分别是 4 和 0.125。

$$\begin{aligned} S &\rightarrow L.L \mid L \\ L &\rightarrow LB \mid B \\ B &\rightarrow 0 \mid 1 \end{aligned}$$

答 先将文法改写成

$$\begin{aligned} S &\rightarrow L.R \mid L \\ L &\rightarrow BL \mid B \\ R &\rightarrow RB \mid B \\ B &\rightarrow 0 \mid 1 \end{aligned}$$

所求语法制导定义如下, 其中 i 是 B 的继承属性, val 和 c 是综合属性。

$$\begin{aligned} S &\rightarrow L.R & S.val &= L.val + R.val \\ S &\rightarrow L & S.val &= L.val \\ L &\rightarrow BL_1 & B.i &= L_1.c \times 2; \quad L.c = L_1.c \times 2; \quad L.val = L_1.val + B.c \\ L &\rightarrow B & B.i &= 1; \quad L.c = 1; \quad L.val = B.c \\ R &\rightarrow R_1B & B.i &= R_1.c / 2; \quad R.c = R_1.c / 2; \quad R.val = R_1.val + B.c \\ R &\rightarrow B & B.i &= 0.5; \quad R.c = 0.5; \quad R.val = B.c \\ B &\rightarrow 0 & B.c &= 0 \\ B &\rightarrow 1 & B.c &= B.i \end{aligned}$$

分析 首先,从 B 推出的字符是不可能知道这个 B 在串中的具体位置的,因此也不可能计算它的贡献。这样必须为 B 设置继承属性,以便计算 B 的贡献。对于一个二进制数,如 101.101,对于小数点前的 1,需要知道它处在从右向左数的第几位上;而对于小数点后的 1,需要知道它处在从左向右数的第几位上。考查原来的文法,从下面图 4.5 的分析树可以看出,不管数字 1 位于小数点前还是小数点后,它都是一种自左向右的归约,适合于自左向右地计数。这样,小数点左边就需要用继承属性来计数,从而提高了难度。而且,由于小数点左右都使用非终结符 L ,那么, L 就既要有综合属性用于从左向右地计数,又要有继承属性用于自右向左地计数,显然语法制导定义的难度会大大增加。所以这里采用改写文法的办法,让分析树是图 4.6 的形式。这样,小数点两边的归约方式不一样,适合于不同方向的计数,使得语法制导定义能简洁得多。

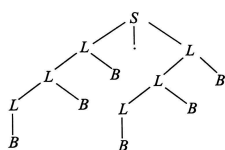


图 4.5 都是自左向右归约的分析树

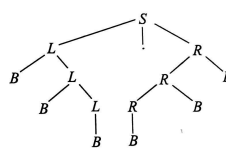


图 4.6 不同归约方式的分析树

由于在分析树中一个 B 对应一个 L (小数点左边) 或一个 R (小数点右边), 那么用 L 和 R 的 c 属性来表示对每一位的贡献, 然后传给 B 的继承属性 i 。如果 B 是 1, 那么 B 的 c 属性就等于它的 i 属性, 否则就等于 0。

如果不限制 B 的综合属性的个数及其含义, 那么可以写出更简单的语法制导定义, 无须继承属性 (见 4.20 题)。另外, 为了使语义规则简明, 本题使用了非 L 属性的语法规则, 它出现在产生式 $L \rightarrow B L$ 的规则集中。

从本习题可以再次看出, 掌握改写文法的技巧对设计语法制导定义是很有用的。

题目 2

考虑如下文法 G ,

$$\begin{aligned} S &\rightarrow (L) \mid a \\ L &\rightarrow L, S \mid S \end{aligned}$$

请设计语法制导的翻译方案, 完成下列任务。你需要自行定义合适的属性。

- (1) 计算每个 a 的嵌套深度。例如, 在句子 $(a, (a, a))$ 中, 每个 a 的嵌套深度分别为 1, 2, 2。
- (2) 计算每个 a 的位置。例如, 在句子 $(a, (a, (a, a), (a)))$ 中, 每个 a 的位置分别为 2, 5, 8, 10, 14。

解答:

4.13 为文法

$$\begin{aligned} S &\rightarrow (L) \mid a \\ L &\rightarrow L, S \mid S \end{aligned}$$

(a) 写一个翻译方案, 它输出每个 a 的嵌套深度。例如, 对于句子 $(a, (a, a))$, 输出的结果是 1 2 2。

(b) 写一个翻译方案, 它打印出每个 a 在句子中是第几个字符。例如, 当句子是 $(a, (a, (a, a), (a)))$ 时, 打印的结果是 2 5 8 10 14。

答 (a) 用继承属性 $depth$ 表示嵌套深度, 所求的翻译方案如下:

$$\begin{aligned} S' &\rightarrow \{ S.depth = 0; \} S \\ S &\rightarrow \{ L.depth = S.depth + 1; \} (L) \\ S &\rightarrow a \{ print (S.depth); \} \\ L &\rightarrow \{ L_1.depth = L.depth; \} L_1, \{ S.depth = L.depth; \} S \\ L &\rightarrow \{ S.depth = L.depth; \} S \end{aligned}$$

(b) 给文法符号 S 和 L 一个继承属性 in 和一个综合属性 out , 分别表示在句子中, 该文法符号推出的字符序列的前面已经有多少个字符, 以及该文法符号推出的字符序列的最后一个字符在句子中是第几个字符。那么所求的翻译方案如下:

$$\begin{aligned} S' &\rightarrow \{ S.in = 0; \} S \\ S &\rightarrow \{ L.in = S.in + 1; \} (L) \{ S.out = L.out + 1; \} \\ S &\rightarrow a \{ S.out = S.in + 1; print (S.out); \} \\ L &\rightarrow \{ L_1.in = L.in; \} L_1, \{ S.in = L_1.out + 1; \} S \{ L.out = S.out; \} \\ L &\rightarrow \{ S.in = L.in; \} S \{ L.out = S.out; \} \end{aligned}$$

分析 从本题开始给出一些需要继承属性的翻译方案(或语法制导定义)的例子, 先从简单的抽象例子开始。

(a) 首先判断是否一定要用继承属性。由于 a 的嵌套深度不是由 a 本身能决定的, 所以一定要用继承属性。另外, a 的嵌套深度从 a 左边的部分足以判断, 因此可以用 L 属性定义解决。

(b) 同样, 字符 a 在句子中的位置也不可能是由 a 本身决定的, 所以一定要用继承属性。另外, a 在句子中的位置从 a 左边的部分足以判断, 因此可以用 L 属性定义解决。

可以用另一个综合属性 $total$ 来代替 out , 它的含义是由 S 或 L 推出的字符总数(这样, out 就是 $in + total$)。那么翻译方案改成:

$$\begin{aligned} S' &\rightarrow \{ S.in = 0; \} S \\ S &\rightarrow \{ L.in = S.in + 1; \} (L) \{ S.total = L.total + 2; \} \\ S &\rightarrow a \{ S.total = 1; print (S.in + 1); \} \\ L &\rightarrow \{ L_1.in = L.in; \} L_1, \{ S.in = L.in + L_1.total + 1; \} \\ &\quad S \{ L.total = L_1.total + 1 + S.total; \} \\ L &\rightarrow \{ S.in = L.in; \} S \{ L.total = S.total; \} \end{aligned}$$

题目 3

考虑文法 \mathcal{G}_D (描述了 C 语言的声明语法):

$$D \rightarrow D () \quad (1)$$

$$D \rightarrow D [] \quad (2)$$

$$D \rightarrow * D \quad (3)$$

$$D \rightarrow (D) \quad (4)$$

$$D \rightarrow \text{id} \quad (5)$$

请设计语法制导的翻译方案, 为 **id** 的类型提供英文/中文说明文档 (由于中英文语法结构的不同, 输出英文说明文档可能更容易; 例子见下页)。

你需要自行定义合适的属性, 请**指明**哪些是综合属性, 哪些是继承属性。

注意: 对于处于最外层的 $D[]$, 不需要输出 **array of**, 只需输出 **array**。

解答:

基本结构 (注意: 这里只给出了 SDD, 非 SDT) 如下:

$$D \rightarrow D () \quad (1)$$

print(function returning)

$$D \rightarrow D [] \quad (2)$$

print(array of)

$$D \rightarrow * D \quad (3)$$

print(pointer to)

$$D \rightarrow (D) \quad (4)$$

print()

$$D \rightarrow \text{id} \quad (5)$$

print(declare **id** as)

需要注意的是, 还需要引入一个继承属性 **depth**。对于处于最外层的 $D[]$, 不需要输出 **array of**, 只需输出 **array**。如果没有这部分处理, 酌情扣分。

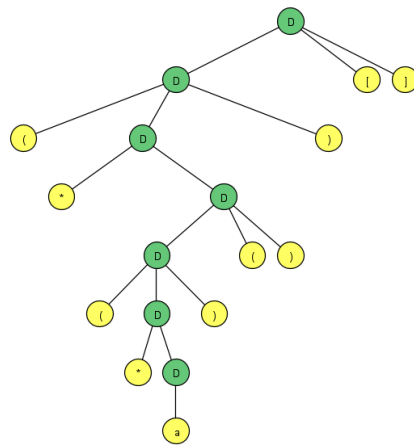


图 1: `(*(*a()))[]`: declare `a` as pointer to function returning pointer to array

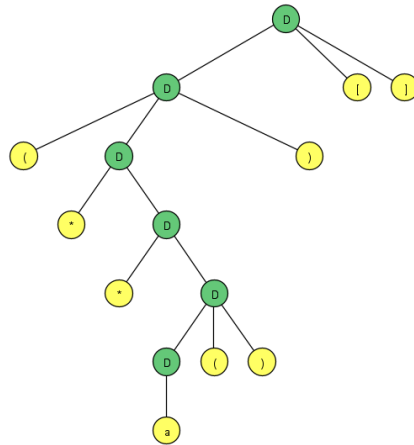


图 2: `(**a())[]`: declare a as function returning pointer to pointer to array

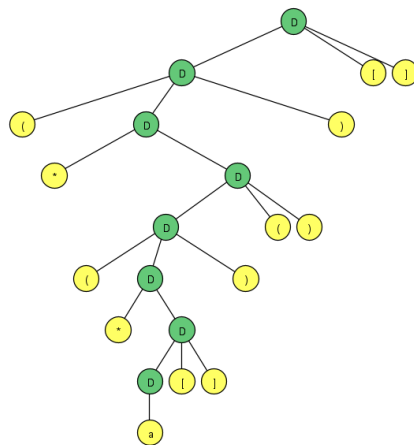


图 3: `(*(*a[]))[]`: declare `a` as array of pointer to function returning pointer to array