Compiler Principle: 语法制导翻译 **Spring 2025**

Homework 7 — April 17

Lecturer: Feng Xiaobing *Completed by: 2022K8009929010 Zhang Jiawei*

7.1

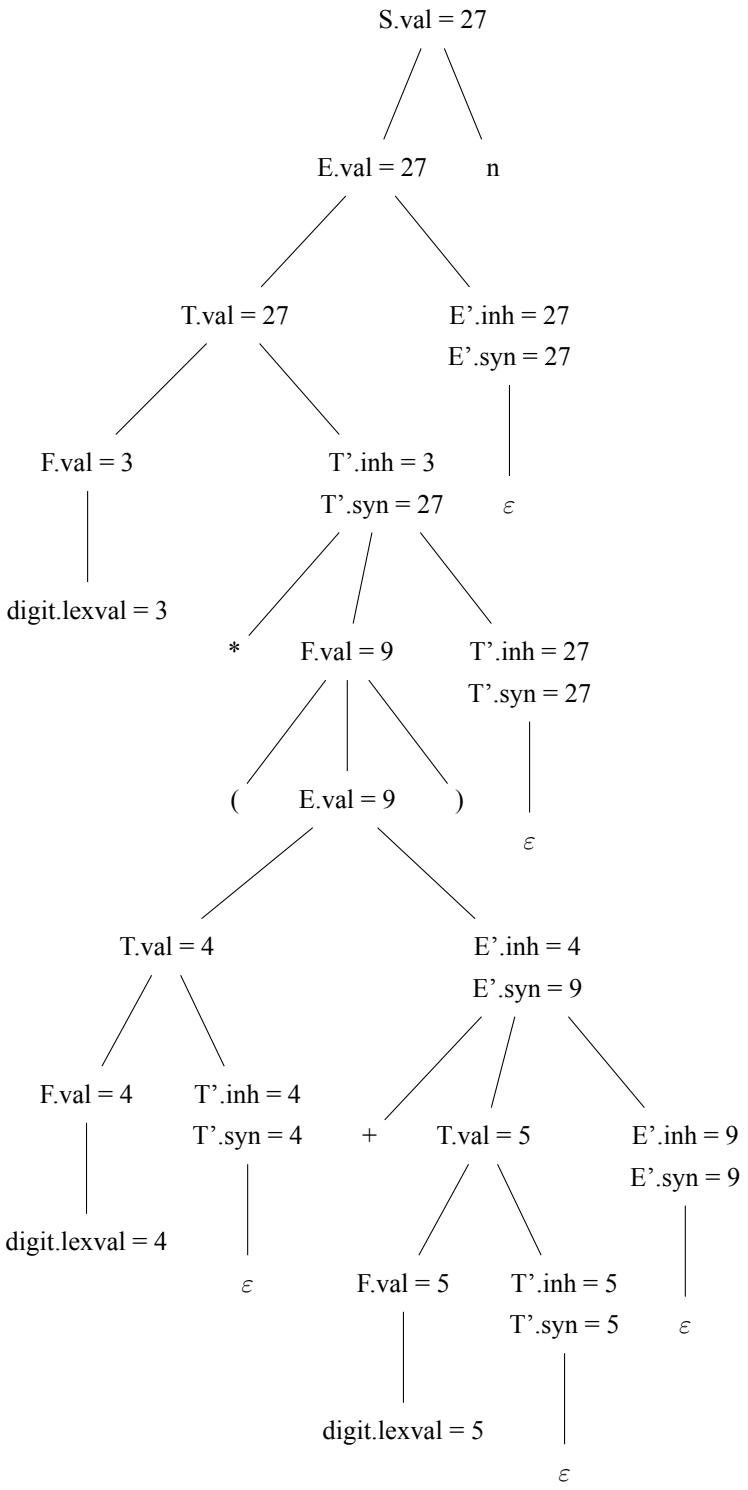
(1) 消除左递归之后的文法如下所示：

$$
\begin{aligned}
S &\to E\,n \\
E &\to T\,E' \\
E' &\to +\,T\,E' \mid \varepsilon \\
T &\to F\,T' \\
T' &\to *\,F\,T' \mid \varepsilon \\
F &\to (\,E\,) \mid digit
\end{aligned}
$$

(2) 所得 SDD 如下表所示：

产生式	语义规则
$S \to E\,n$	$S.val = E.val$
$E \to T\,E'$	$E'.inh = T.val$ $E.val = E'.syn$
$E' \to +\,T\,E'_1$	$E'_1.inh = E'.inh + T.val$ $E'.syn = E'_1.syn$
$E' \to \varepsilon$	$E'.syn = E'.inh$
$T \to F\,T'$	$T'.inh = F.val$ $T.val = T'.syn$
$T' \to *\,F\,T'_1$	$T'_1.inh = T'.inh \times F.val$ $T'.syn = T'_1.syn$
$T' \to \varepsilon$	$T'.syn = T'.inh$
$F \to (\,E\,)$	$F.val = E.val$
$F \to digit$	$F.val = digit.lexval$

(3) 使用上面得到的 SDD，给出表达式 3 * (4 + 5) n 的注释语法分析树：


S.val = 27
E.val = 27
n
T.val = 27
E'.inh = 27
E'.syn = 27
F.val = 3
T'.inh = 3
T'.syn = 27
ε
digit.lexval = 3
*
F.val = 9
T'.inh = 27
T'.syn = 27
(
E.val = 9
)
ε
T.val = 4
E'.inh = 4
E'.syn = 9
F.val = 4
T'.inh = 4
T'.syn = 4
+
T.val = 5
E'.inh = 9
E'.syn = 9
digit.lexval = 4
ε
F.val = 5
T'.inh = 5
T'.syn = 5
ε
digit.lexval = 5
ε

7.2

(1) B.i = A.i; A.s = B.i + C.s

(a) A 的综合属性由其子节点定义，符合 S 属性要求；

(b) B 的继承属性由其父节点定义，符合 L 属性要求；

(c) 没有循环依赖，故存在一致的求值过程。

(2) B.i = A.i; A.s = B.i + C.s; D.i = A.i + B.s

(a) A 的综合属性由其子节点定义，符合 S 属性要求；

(b) B 的继承属性由其父节点定义，D 的继承属性由其父节点和前方兄弟节点定义，符合 L 属性要求；

(c) 没有循环依赖，故存在一致的求值过程。

(3) A.s = B.s + C.s

(a) A 的综合属性由其子节点定义，符合 S 属性要求；

(b) 无继承属性语义规则，符合 L 属性要求；

(c) 没有循环依赖，故存在一致的求值过程。

7.3

仍然先消除左递归：

$$
\begin{aligned}
E &\to T\,E' \\
E' &\to +\,T\,E' \mid \varepsilon \\
T &\to F\,T' \\
T' &\to *\,F\,T' \mid \varepsilon \\
F &\to (\,E\,) \mid num \mid var
\end{aligned}
$$

设计 SDD 如下表所示：

产生式	语义规则
$E \to T\, E'$	$E'.inh = T.isconst$ $E.isconst = E'.syn$
$E' \to +\, T\, E'_1$	$E'_1.inh = E'.inh + T.isconst$ $E'.syn = E'_1.syn$
$E' \to \varepsilon$	$E'.syn = E'.inh$
$T \to F\, T'$	$T'.inh = F.isconst$ $T.isconst = T'.syn$
$T' \to *\, F\, T'_1$	$T'_1.inh = T'.inh \times F.isconst$ $T'.syn = T'_1.syn$
$T' \to \varepsilon$	$T'.syn = T'.inh$
$F \to (\, E\,)$	$F.isconst = E.isconst$
$F \to num$	$F.isconst = true$
$F \to var$	$F.isconst = false$

7.4

先消除文法左递归:

$$B \to 1\, B'$$
$$B' \to 0\, B' \,|\, 1\, B' \,|\, \varepsilon$$

然后添加语义动作,写出 SDT:

$$\begin{aligned}
B \to & 1\, B' && \{B'.inh = 1\,,\ B.val = B'.syn\} \\
B' \to & 0\, B'_1 && \{B'_1.inh = B'.inh \times 2\,,\ B'.syn = B'_1.syn\} \\
| & 1\, B'_1 && \{B'_1.inh = B'.inh \times 2 + 1\,,\ B'.syn = B'_1.syn\} \\
| & \varepsilon && \{B'.syn = B'.inh\}
\end{aligned}$$

7.5

(1) 先写出 SDD 如下:

$$\begin{aligned}
S \to \text{if}\,(C)\, S_1\, \text{else}\, S_2 \qquad & L_1 = new(); \\
& C.false = L_1; \\
& S_1.next = S.next; \\
& S_2.next = S.next; \\
& S.code = C.code \,\|\, S_1.code \,\|\, \text{label} \|\, L_1 \,\|\, S_2.code
\end{aligned}$$

对应 SDT 如下：

$$
\begin{aligned}
S \to \text{if}\,(\quad & \{L_1 = new()\,,\ C.false = L_1\} \\
C\,) \quad & \{S_1.next = S.next\} \\
S_1\ \text{else} \quad & \{S_2.next = S.next\} \\
S_2 \quad & \{S.code = C.code \parallel S_1.code \parallel \text{label} \parallel L_1 \parallel S_2.code\}
\end{aligned}
$$

(2) 先写出 SDD 如下：

$$
\begin{aligned}
S \to \text{do}\, S_1\, \text{while}\, (C) \quad & L_1 = new(); \\
& C.false = S.next; \\
& C.true = L_1; \\
& S.code = \text{label} \parallel L_1 \parallel S_1.code \parallel C.code
\end{aligned}
$$

对应 SDT 如下：

$$
\begin{aligned}
S \to \text{do} \quad & \{L_1 = new()\} \\
S_1 \quad & \{C.true = L_1\,,\ C.false = S.next\} \\
\text{while}\,(C) \quad & \{S.code = \text{label} \parallel L_1 \parallel S_1.code \parallel C.code\}
\end{aligned}
$$

(3) 先写出 SDD 如下：

$$
\begin{aligned}
S \to \{\, L \,\} \quad & L.next = S.next\,,\ S.code = L.code \\
L \to L_1\, S \quad & S.next = L.next\,,\ L.code = L_1.code \parallel S.code \\
L \to \varepsilon \quad & \text{if}\,(L.next)\{L.code = \text{goto} \parallel L.next\}\ \text{else}\ \{L.code = \varepsilon\}
\end{aligned}
$$

对应 SDT 如下：

$$
\begin{aligned}
S \to \{ \quad & \{L.next = S.next\} \\
L\,\} \quad & \{S.code = L.code\} \\
L \to L_1 \quad & \{S.next = L.next\} \\
S \quad & \{L.code = L_1.code \parallel S.code\} \\
L \to \varepsilon \quad & \{\text{if}\,(L.next)\,\{L.code = \text{goto} \parallel L.next\}\ \text{else}\ \{L.code = \varepsilon\}\}
\end{aligned}
$$