

作业9

2020K8009929017 侯昱帆

练习5.2.1: 考虑文法

$E \rightarrow E + T \mid T$

$T \rightarrow T * F \mid F$

$F \rightarrow (E) \mid \text{num} \mid \text{var}$

其中 num 表示数字字面量, 如 123; var 表示变量, 如 x。

如果一个项或表达式中不存在变量, 它可以在编译期进行求值, 例如 $1 + 2 * 3$ 。

设计一个 SDD 来判断各个项 T 和表达式 E 能否在编译期确定它的值。

(备注: 可以用 $E.\text{isconst} = \text{true}$ 表示它可以在编译期求值)

答:

产生式	语义规则
$E \rightarrow E1 + T$	$E.\text{isconst} = E1.\text{isconst} \text{ and } T.\text{isconst}$
$E \rightarrow T$	$E.\text{isconst} = T.\text{isconst}$
$T \rightarrow T1 * F$	$T.\text{isconst} = T1.\text{isconst} \text{ and } F.\text{isconst}$
$T \rightarrow F$	$T.\text{isconst} = F.\text{isconst}$
$F \rightarrow (E)$	$F.\text{isconst} = E.\text{isconst}$
$F \rightarrow \text{num}$	$F.\text{isconst} = \text{true}$
$F \rightarrow \text{var}$	$F.\text{isconst} = \text{false}$

练习5.2.2: 改写下面的 SDT, 消除左递归。其中a、b、c、d为语义动作, 不涉及属性计算。0和1是终结符。

$A \rightarrow A \{a\} B \mid A B \{b\} \mid 0$

$B \rightarrow B \{c\} A \mid B A \{d\} \mid 1$

答:

$A \rightarrow 0 A'$

$A' \rightarrow \{a\} B A' \mid B \{b\} A' \mid \epsilon$

$B \rightarrow 1 B'$

$B' \rightarrow \{c\} A B' \mid A \{d\} B' \mid \epsilon$

练习5.2.3: 为下面的产生式写出一个和例 5.19 类似的 L 属性 SDT。这里的每个产生式表示一个常见的 C 语言中那样的控制流结构。你可能需要生成一个三地址语句来跳转到某个标号 L, 此时你可以生成语句 goto L

1. $S \rightarrow \text{if} (C) S1 \text{ else } S2$

2. $S \rightarrow \text{do } S1 \text{ while} (C)$

3. $S \rightarrow \{ L \}$ 和 $L \rightarrow L S \mid \epsilon$

注1: 列表中的任何语句都可能包含一条从它的内部跳转到下一条语句的跳转指令, 故简单地各个语句按顺序生成代码是不够的。

注2: 可以先写出 SDD, 然后按照 5.4.5 节方法转换为 SDT。

答:

(1) $S \rightarrow \text{if } (C) S1 \text{ else } S2$

SDD

```
S → if (C) S1 else S2    L1 = new();
                          L2 = new();
                          C.true = L1;
                          C.false = L2;
                          S1.next = S.next;
                          S2.next = S.next;
                          S.code = C.code || label || L1 || S1.code || label || L2 ||
S2.code
```

SDT

```
S → if (    {L1 = new(); L2 = new(); C.true = L1; C.false = L2;}
C)         {S1.next = S.next;}
S1 else {S2.next = S.next;}
S2       {S.code = C.code || label || L1 || S1.code || label || L2 || S2.code;}
```

(2) $S \rightarrow \text{do } S1 \text{ while } (C)$

SDD

```
S → do S1 while (C)    L1 = new();
                        L2 = new();
                        S1.next = L2;
                        C.true = L1;
                        C.false = S.next;
                        S.code = label || L1 || S1.code || label || L2 || C.code
```

SDT

```
S → do          {L1 = new(); L2 = new(); S1.next = L2;}
S1 while (      {C.true = L1; C.false = S.next;}
c)             {S.code = label || L1 || S1.code || label || L2 || C.code;}
```

(3) $S \rightarrow \{L\}$ 和 $L \rightarrow LS \mid \epsilon$

SDD

$S \rightarrow \{L\}$	$S.code = L.code;$ $L.next = S.next$
$L \rightarrow L'S$	$L1 = new();$ $L'.next = L1;$ $S.next = L.next;$ $L.code = L'.code \ \ label \ \ L1 \ \ S.code$
$L \rightarrow \epsilon$	$L.code = \epsilon$

SDT

$S \rightarrow \{$	$\{L.next = S.next;\}$
$L\}$	$\{S.code = L.code;\}$
$L \rightarrow$	$\{L1 = new(); L'.next = L1;\}$
L'	$\{S.next = L.next;\}$
S	$\{L.code = L'.code \ \ label \ \ L1 \ \ S.code;\}$
$L \rightarrow \epsilon$	$\{L.code = \epsilon;\}$