

# 第11讲(中间代码生成 1)要点

▶本章是第5章 (语法制导翻译) 的延伸

实现机制 (第5章) **SDT** (第6章)

- 如何扩展语法分析栈
- · 如何将SDT中的抽象定义式改写成具体 . 执行的栈操作 • 如何扩展递归下降过程

  - 如何修改L属性定义的SDT,使之适合 bottom\_up翻译
  - 声明语句的SDT
- 赋值语句的SDT 控制语句的SDT
  - 过程调用语句的SDT

# 声明语句的翻译

#### 户主要任务

分析所声明id的种属、类型和地址,在符号表中为id建立一条记录(kind)(type)

```
第6章 简变 int array(n,T) 数组 real T_1 \times T_2 \times ... \times T_n \to R 过程 char pointer (T) 指针 bool record((N_1 \times T_1) \times ... \times (N_n \times T_n)) 结构体 第7章
```

类型表达式(T)

② 
$$D \rightarrow T \text{ id}; D$$

$$\bigcirc D \rightarrow \varepsilon$$

$$\textcircled{4} T \rightarrow BC$$

$$\bigcirc$$
  $T \rightarrow \uparrow T_1$ 

$$\bigcirc B \rightarrow \text{real} \{ B.type = real; B.width = 8; \}$$

$$\otimes$$
  $C \rightarrow \varepsilon$ 

$$9 C \rightarrow [\text{num}]C_1$$

变量	作用
offset	下一个可用的相对地址

翻译的主要任务: 类型表达式和地址

符号	综合属性	
T	type	width
В	type	width

 $offset_{i+1} = offset_i + T_i.width$ 

翻译的主要任务: 类型表达式和地址

② 
$$D \rightarrow T \text{ id}$$
; { enter(id.lexeme, T.type, offset); offset = offset + T.width; } $D$ 

$$\bigcirc D \rightarrow \varepsilon$$

$$(4)$$
  $T \rightarrow BC$ 

符号	综合属性	
T	type	width
В	type	width

$$\bigcirc T \rightarrow \uparrow T_1$$

$$\bigcirc B \rightarrow \text{real} \{ B.type = real; B.width = 8; \}$$

$$\otimes$$
  $C \rightarrow \varepsilon$ 

$$\bigcirc$$
  $C \rightarrow [\text{num}]C_1$ 

变量	作用
offset	下一个可用的相对地址

$$offset_{i+1} = offset_i + T_i.width$$

```
 (1) P \rightarrow \{ offset = 0 \} D
```

翻译的主要任务: 类型表达式和地址

- ②  $D \rightarrow T \text{ id}$ ; { enter(id.lexeme, T.type, offset); offset = offset + T.width; } D
- $(4) T \rightarrow BC$
- $\bigcirc$   $T \rightarrow \uparrow T_1$
- 7  $B \rightarrow \text{real} \{ B. type = real; B. width = 8; \}$
- $\otimes$   $C \rightarrow \varepsilon$
- $\bigcirc$   $C \rightarrow [\text{num}]C_1$

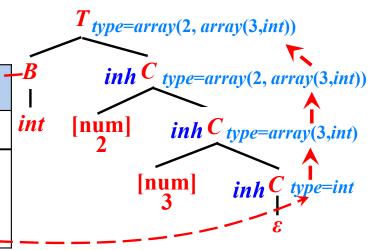
例: int[2][3]

type=array(2, array(3,int))

右结合

变量	作用
offset	下一个可用的相对地址
t.w_	将类型和宽度信息从语法 分析树中的B结点传递到 对应于产生式C→E的结点

符号	综合属性	
T	type	width
В	type	width
C	type	width



翻译的主要任务: 类型表达式和地址

- ②  $D \rightarrow T \text{ id}$ ; { enter(id.lexeme, T.type, offset); offset = offset + T.width; } D
- $\bigcirc D \rightarrow \varepsilon$
- (4)  $T \rightarrow BC$
- $\bigcirc$   $T \rightarrow \uparrow T_1$
- 7  $B \rightarrow \text{real} \{ B. type = real; B. width = 8; \}$
- $\otimes$   $C \rightarrow \varepsilon$
- $\bigcirc$   $C \rightarrow [\text{num}]C_1$

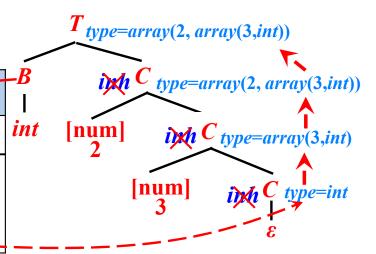
例: int[2][3]

type=array(2, array(3,int))

右结合

变量	作用
offset	下一个可用的相对地址
t.w	将类型和宽度信息从语法 分析树中的B结点传递到 对应于产生式C→E的结点

符号	综合属性	
T	type	width
В	type	width
C	type	width



翻译的主要任务: 类型表达式和地址

- ②  $D \rightarrow T \text{ id}$ ; { enter(id.lexeme, T.type, offset); offset = offset + T.width; } D
- 4  $T \rightarrow B$  { t = B.type; w = B.width;} C { T.type = C.type; T.width = C.width; }
- ⑤  $T \rightarrow \uparrow T_1 \{ T.type = pointer(T_1.type); T.width = 4; \}$
- 6  $B \rightarrow int \{ B.type = int; B.width = 4; \}$
- $\bigcirc B \rightarrow \text{real} \{ B.type = real; B.width = 8; \}$
- ⊗ C → ε { C.type = t; C.width = w; }
- ⑨  $C \rightarrow [\text{num}]C_1 \{ C.type = array(num.val, C_1.type);$  $C.width = num.val * C_1.width; \}$

符号	综合属性	
T	type	width
В	type	width
C	type	width

