

第5讲 中间代码生成

学习的主要内容和目标

➤ 学习的主要内容

- ✧ 中间代码形式;
- ✧ 赋值语句;

➤ 学习的目标

- ✧ 理解程序设计语言典型语法结构的单遍扫描翻译方法。

中间代码形式

➤ 中间代码

◇ 源程序的一种内部表示形式

◇ 作用

- ✓ 源语言和目标语言之间的桥梁，避开二者之间较大的语义跨度；
- ✓ 提高编译器的可移植性；
- ✓ 利于进行与目标机无关的优化。

属性文法

➤ 中间代码

✧ 三地址码

✓ $A := (B + C) * (B + D)$

(1) $T1 := B + C$

(2) $T2 := B + D$

(3) $T3 := T1 * T2$

(4) $A := T3$

赋值语句的翻译

➤ 赋值语句所期待的中间代码

✧ $X := (A + B) * C$

$T1 := A + B$

$T2 := T1 * C$

$X := T2$

如何设计语
法制导定义？



赋值语句的翻译

➤ 赋值语句的翻译

✧ $X := (A + B) * C$

$T1 := A + B$

$T2 := T1 * C$

$X := T2$

$S \rightarrow id := E$

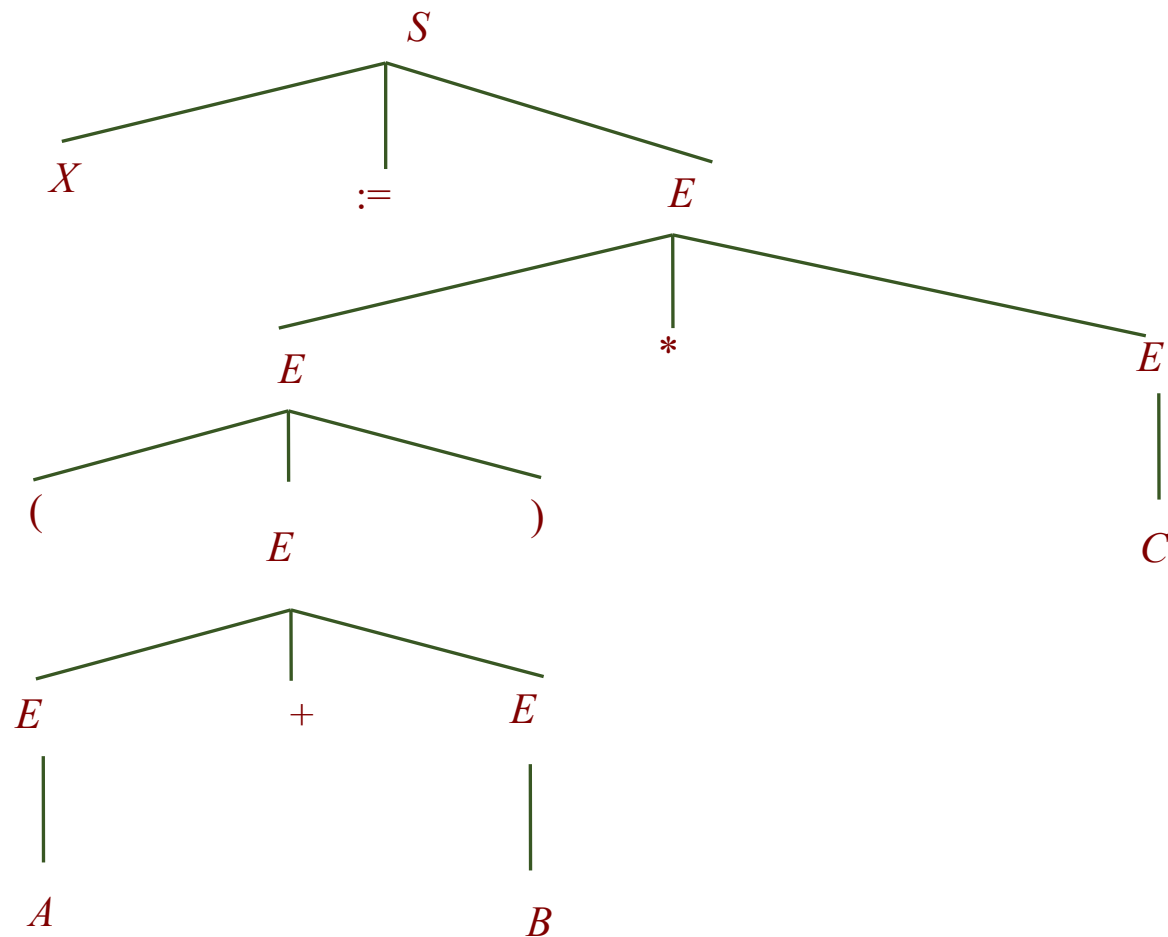
$E \rightarrow E_1 + E_2$

$E \rightarrow E_1 * E_2$

$E \rightarrow -E_1$

$E \rightarrow (E_1)$

$E \rightarrow id$



赋值语句的翻译

➤ 语义属性

- ✧ **id.name** : id 的词法名字（符号表中的名字）
- ✧ 综合属性 **S.code** : 表示语句 S 翻译得到的代码序列；
- ✧ 综合属性 **E.addr** : 表示一个临时变量；

赋值语句的翻译

➤ 语义函数/过程

- ✧ 函数lookup: 在符号表中查看是否存在id.name对应的项, 如果有则返回序号, 不存在则返回0;
- ✧ 函数newtemp: 生成一个临时变量名;
- ✧ 全局变量nextquad: 表示即将生成的下一条三地址代码的地址;
- ✧ 函数gen: 每调用一次gen生成一条三地址代码, 且将记录下一条三地址代码的序号的nextquad加1。

赋值语句的翻译

➤ 赋值语句的翻译

$S \rightarrow id := E$ { $p := \text{lookup}(id.name)$; if $(p \neq \text{nil})$ then $S.code := E.code \parallel \text{gen}(p := E.addr)$ else error }

$E \rightarrow E_1 + E_2$ { $E.addr := \text{newtemp}$; $E.code := E_1.code \parallel E_2.code \parallel \text{gen}(E.addr := E_1.addr + E_2.addr)$ }

$E \rightarrow E_1 * E_2$ { $E.addr := \text{newtemp}$; $E.code := E_1.code \parallel E_2.code \parallel \text{gen}(E.addr := E_1.addr * E_2.addr)$ }

$E \rightarrow -E_1$ { $E.addr := \text{newtemp}$; $E.code := E_1.code \parallel \text{gen}(E.addr := \text{'uminus'} E_1.addr)$ }

$E \rightarrow (E_1)$ { $E.addr := E_1.addr$; $E.code := E_1.code$ }

$E \rightarrow \mathbf{id}$ { $p := \text{lookup}(id.name)$; if $(p \neq \text{nil})$ then $E.addr := p$ else error; }

赋值语句的翻译

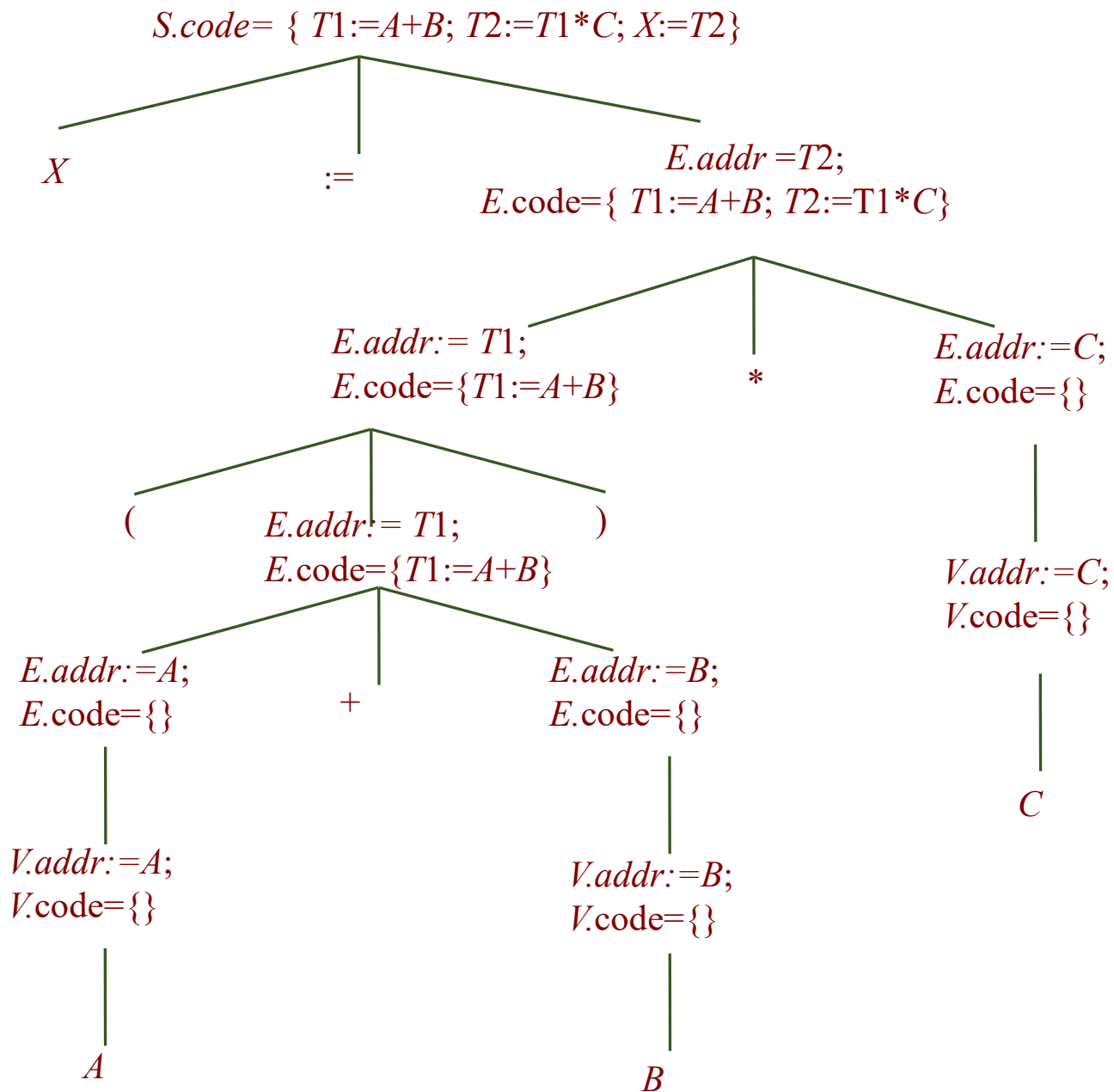
➤ 赋值语句的翻译

✧ $X := (A+B)*C$

$T1 := A+B$

$T2 := T1 * C$

$X := T2$



赋值语句的翻译

➤ 赋值语句的翻译

$S \rightarrow id := E \{ p := \text{lookup}(id.name); \text{ if } (p \neq \text{nil}) \text{ then gen}(p := E.addr) \text{ else error} \}$

$E \rightarrow E_1 + E_2 \{ E.addr := \text{newtemp}; \text{ gen}(E.addr := E_1.addr + E_2.addr) \}$

$E \rightarrow E_1 * E_2 \{ E.addr := \text{newtemp}; \text{ gen}(E.addr := E_1.addr * E_2.addr) \}$

$E \rightarrow -E_1 \{ E.addr := \text{newtemp}; \text{ gen}(E.addr := 'uminus' E_1.addr) \}$

$E \rightarrow (E_1) \{ E.addr := E_1.addr ; E.code := E_1.code \}$

$E \rightarrow \mathbf{id} \{ p := \text{lookup}(id.name); \text{ if } (p \neq \text{nil}) \text{ then } E.addr := p \text{ else error}; \}$

赋值语句的翻译

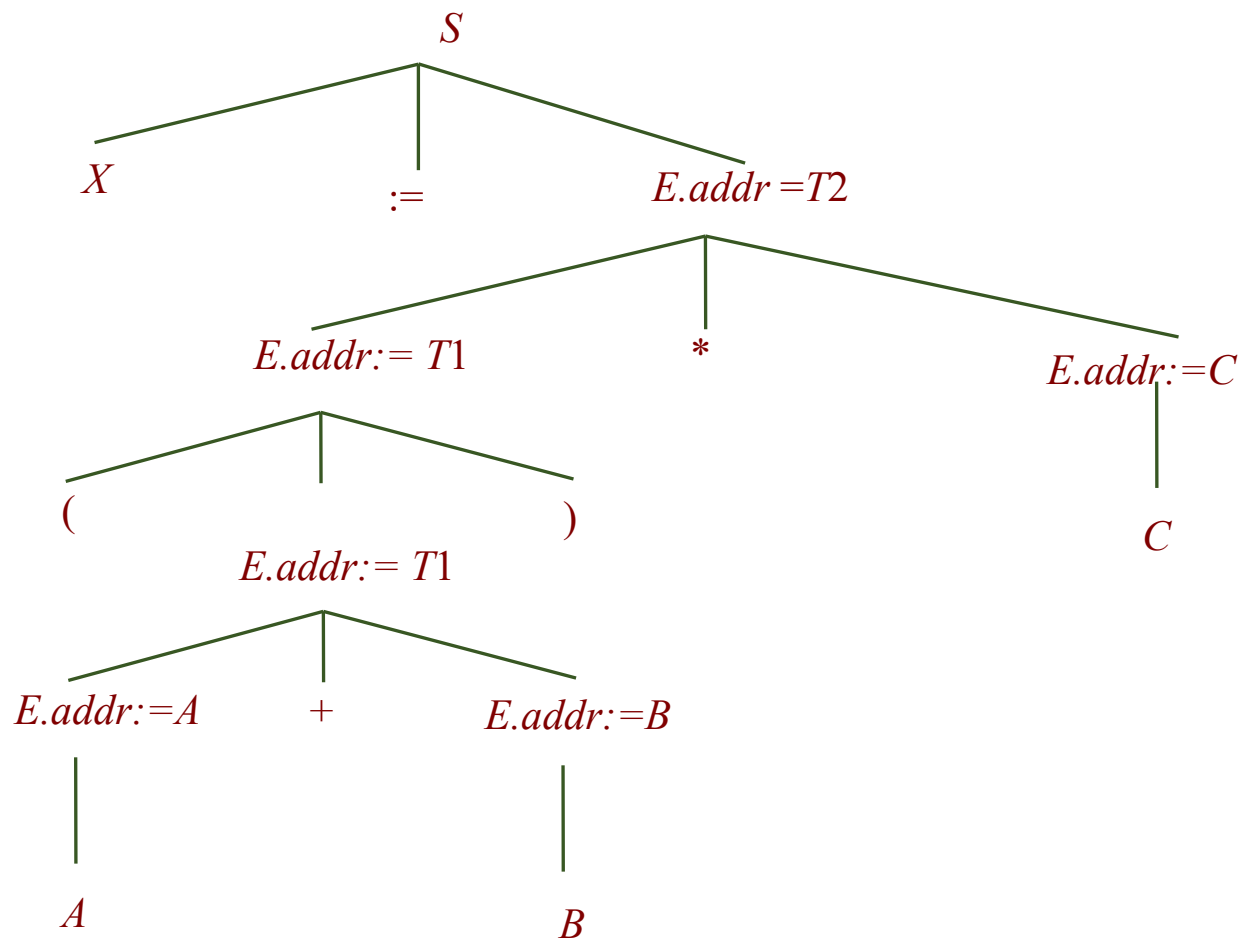
➤ 赋值语句的翻译

✧ $X := (A + B) * C$

$T1 := A + B$

$T2 := T1 * C$

$X := T2$



赋值语句的翻译

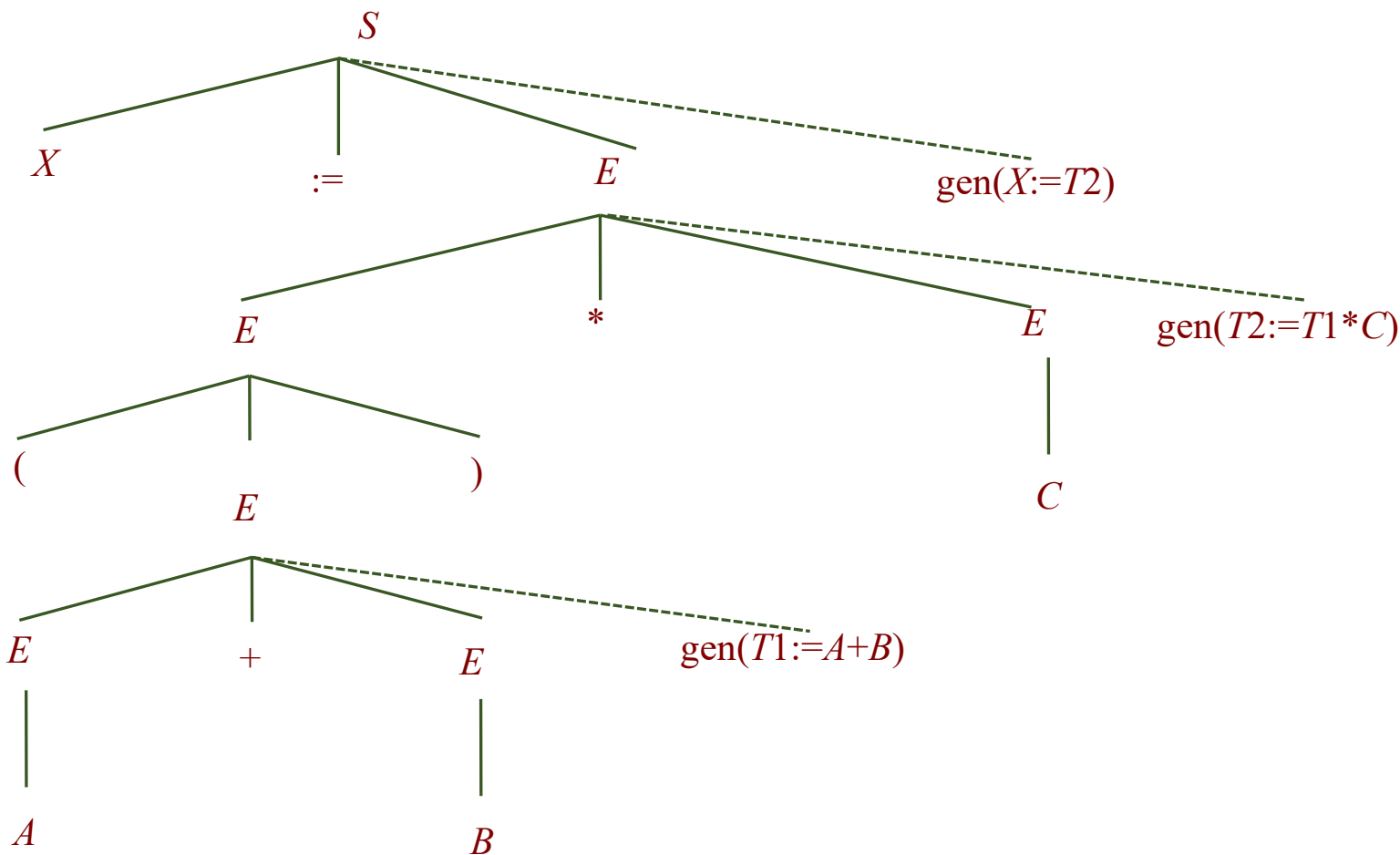
➤ 赋值语句的翻译

✧ $X := (A + B) * C$

$T1 := A + B$

$T2 := T1 * C$

$X := T2$



赋值语句的翻译

➤ 赋值语句的文法

✧ $X := (A+B)*C$

$S \rightarrow id := E$

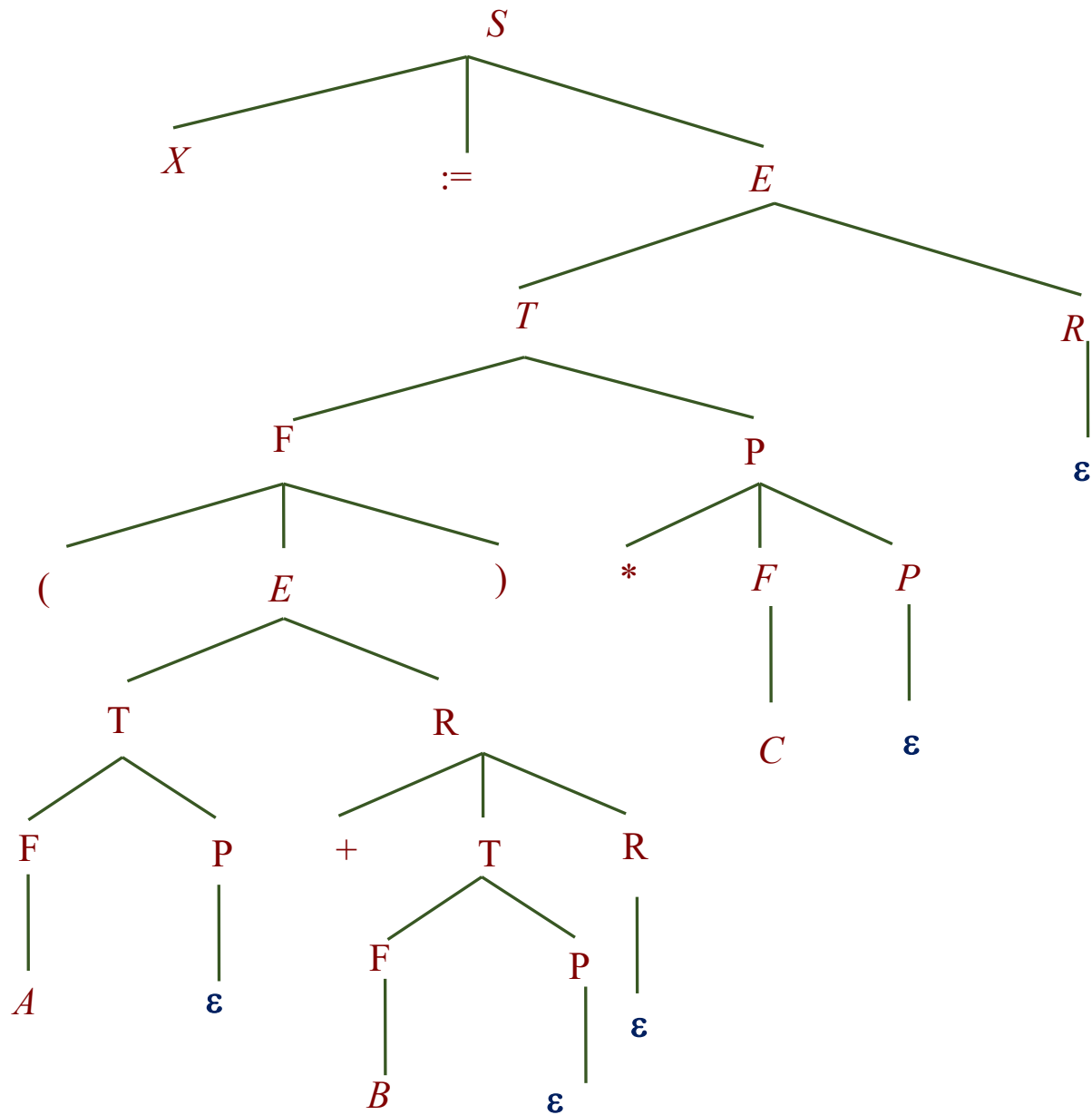
$E \rightarrow TR$

$R \rightarrow + TR \mid \varepsilon$

$T \rightarrow FP$

$P \rightarrow * FP \mid \varepsilon$

$F \rightarrow (E) \mid id$



赋值语句的翻译

➤ 赋值语句的文法

$S \rightarrow id := E \{ p := \text{lookup}(id.name); \text{if } (p \neq \text{nil}) \text{ then gen}(p := E.addr) \text{ else error} \}$

$E \rightarrow T \{ R.i := T.addr \} \quad R \{ E.addr := R.addr \}$

$R \rightarrow + T \{ R_1.i := \text{newtemp}; \text{gen}(R_1.i := R.i \text{ '+' } T.addr) \} \quad R_1 \{ R.addr := R_1.addr \}$

$R \rightarrow \varepsilon \{ R.addr := R.i \}$

$T \rightarrow F \{ P.i := F.addr \} \quad P \{ T.addr := P.addr \}$

$P \rightarrow * F \{ P_1.i := \text{newtemp}; \text{gen}(P_1.i := P.i \text{ '*' } F.addr) \} \quad P_1 \{ P.addr := P_1.addr \}$

$P \rightarrow \varepsilon \{ P.addr := P.i \}$

$F \rightarrow (E) \{ F.addr := E.addr \}$

$F \rightarrow id \{ p := \text{lookup}(id.name); \text{if } (p \neq \text{nil}) \text{ then } F.addr := p \text{ else error; } \}$

赋值语句的翻译

➤ 赋值语句的文法

