

1-2 编译系统的结构（前端/后端流图） 总体结构图各部分功能

• 分析部分

前端(front end): 与源语言相关

→

• 综合部分

后端(back end): 与目标语言相关

→

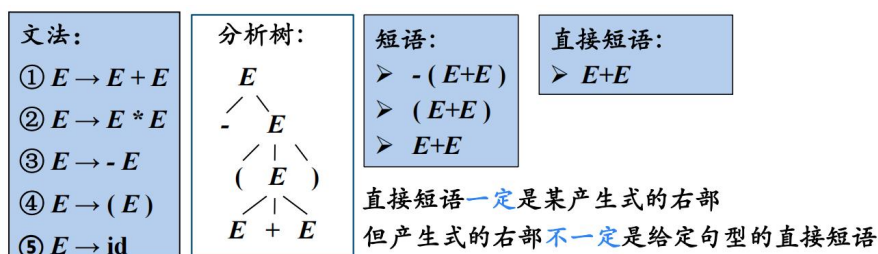


2-5 CFG 的分析树（短语、二义性）

• （句型的）短语

• 给定一个句型，其分析树中的每一棵子树的边缘称为该句型的一个**短语**(phrase)

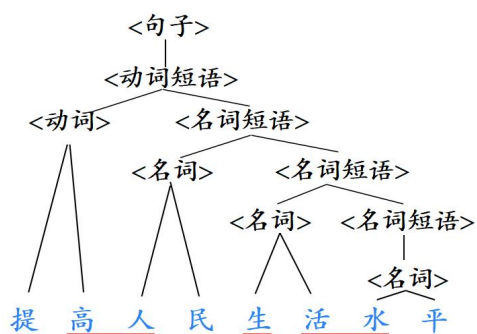
• 如果子树只有父子两代结点，那么这棵子树的边缘称为该句型的一个**直接短语**(immediate phrase)



例题:

文法:

- ① $\langle \text{句子} \rangle \rightarrow \text{动词短语}$
- ② $\langle \text{动词短语} \rangle \rightarrow \langle \text{动词} \rangle \langle \text{名词短语} \rangle$
- ③ $\langle \text{名词短语} \rangle \rightarrow \langle \text{名词} \rangle \langle \text{名词短语} \rangle \mid \langle \text{名词} \rangle$
- ④ $\langle \text{动词} \rangle \rightarrow \text{提 高}$
- ⑤ $\langle \text{名词} \rangle \rightarrow \underline{\text{高 人}} \mid \text{人 民} \mid \underline{\text{民 生}} \mid \text{生 活} \mid \text{活 水} \mid \underline{\text{水 平}}$



4-4 FIRST 集和 FOLLOW 集的计算

1、计算文法符号 X 的 $FIRST(X)$

- $FIRST(X)$: 可以从 X 推导出的所有串首终结符构成的集合
- 如果 $X \rightarrow * \varepsilon$, 那么 $\varepsilon \in FIRST(X)$

· 例

$$\textcircled{1} E \rightarrow TE' \quad FIRST(E) = \{ (\text{ id } \}$$

$$\textcircled{2} E' \rightarrow + TE' \mid \varepsilon \quad FIRST(E') = \{ + \varepsilon \}$$

$$\textcircled{3} T \rightarrow FT' \quad FIRST(T) = \{ (\text{ id } \}$$

$$\textcircled{4} T' \rightarrow * FT' \mid \varepsilon \quad FIRST(T') = \{ * \varepsilon \}$$

$$\textcircled{5} F \rightarrow (E) \mid id \quad FIRST(F) = \{ (\text{ id } \}$$

算法

不断应用下列规则, 直到没有新的终结符或 ε 可以被加入到任何 $FIRST$ 集合中为止

- 如果 X 是一个终结符, 那么 $FIRST(X) = \{ X \}$
- 如果 X 是一个非终结符, 且 $X \rightarrow Y_1 \cdots Y_k \in P$ ($k \geq 1$), 那么如果对于某个 i , a 在 $FIRST(Y_i)$ 中且 ε 在所有的 $FIRST(Y_1), \cdots, FIRST(Y_{i-1})$ 中(即 $Y_1 \cdots Y_{i-1} \rightarrow * \varepsilon$)就把 a 加入到 $FIRST(X)$ 中。如果对于所有的 $j = 1, 2, \dots, k$, ε 在 $FIRST(Y_j)$ 中, 那么将 ε 加入到 $FIRST(X)$
- 如果 $X \rightarrow \varepsilon \in P$, 那么将 ε 加入到 $FIRST(X)$ 中

2、计算串 $X_1 X_2 \cdots X_n$ 的 $FIRST$ 集合

- 向 $FIRST(X_1 X_2 \cdots X_n)$ 加入 $FIRST(X_1)$ 中所有的非 ε 符号
- 如果 ε 在 $FIRST(X_1)$ 中, 再加入 $FIRST(X_2)$ 中的所有非 ε 符号;
- 如果 ε 在 $FIRST(X_1)$ 和 $FIRST(X_2)$ 中, 再加入 $FIRST(X_3)$ 中的所有非 ε 符号, 以此类推。
- 最后, 如果对所有的 i , ε 都在 $FIRST(X_i)$ 中, 那么将 ε 加入到 $FIRST(X_1 X_2 \cdots X_n)$ 中。

3、计算非终结符 A 的 $FOLLOW(A)$

$FOLLOW(A)$: 可能在某个句型中紧跟在 A 后边的终结符 a 的集合

$$FOLLOW(A) = \{a \mid S \Rightarrow^* \alpha A a \beta, a \in V_T, \alpha, \beta \in (V_T \cup V_N)^*\}$$

如果 A 是某个句型的的最右符号, 则将结束符 “\$” 添加到 $FOLLOW(A)$ 中

例:

- ① $\underline{E \rightarrow TE'}$ $FIRST(E) = \{ (\text{id} \}$ $FOLLOW(E) = \{ \$) \}$
- ② $\underline{E' \rightarrow +TE'}$ $|\epsilon$ $FIRST(E') = \{ (+ \epsilon \}$ $FOLLOW(E') = \{ \$) \}$
- ③ $\underline{T \rightarrow FT'}$ $FIRST(T) = \{ (\text{id} \}$ $FOLLOW(T) = \{ + \$) \}$
- ④ $\underline{T' \rightarrow *FT'}$ $|\epsilon$ $FIRST(T') = \{ (* \epsilon \}$ $FOLLOW(T') = \{ + \$) \}$
- ⑤ $\underline{F \rightarrow (E)}$ $|\text{id}$ $FIRST(F) = \{ (\text{id} \}$ $FOLLOW(F) = \{ * + \$) \}$

算法

- 不断应用下列规则, 直到没有新的终结符可以被加入到任何 $FOLLOW$ 集合中为止
 - 将 \$ 放入 $FOLLOW(S)$ 中, 其中 S 是开始符号, \$ 是输入右端的结束标记
 - 如果存在一个产生式 $A \rightarrow \alpha B \beta$, 那么 $FIRST(\beta)$ 中除 ϵ 之外的所有符号都在 $FOLLOW(B)$

中

- 如果存在一个产生式 $A \rightarrow \alpha B$, 或存在产生式 $A \rightarrow \alpha B \beta$ 且 $FIRST(\beta)$ 包含 ϵ , 那么 $FOLLOW(A)$ 中的所有符号都在 $FOLLOW(B)$ 中

4-5 递归的预测分析法 (预测分析表)

4-10 LR(0)分析 (增广文法)

右部某位置标有圆点的产生式称为相应文法的一个 **LR(0)项目** (简称为项目)

$$A \rightarrow \alpha_1 \cdot \alpha_2$$

例: $S \rightarrow bBB$

➤ $S \rightarrow \cdot bBB$ ← 移进项目

➤ $S \rightarrow b \cdot BB$ } 待约项目

➤ $S \rightarrow bB \cdot B$ }

➤ $S \rightarrow bBB \cdot$ ← 归约项目

项目描述了句柄识别的状态

产生式 $A \rightarrow \varepsilon$ 只生成一个项目 $A \rightarrow \cdot$

增广文法 (Augmented Grammar)

如果 G 是一个以 S 为开始符号的文法, 则 G 的 **增广文法** G' 就是在 G 中加上新开始符号

S' 和产生式 $S' \rightarrow S$ 而得到的文法

➤ 例

1) $E \rightarrow E + T$
2) $E \rightarrow T$
3) $T \rightarrow T * F$
4) $T \rightarrow F$
5) $F \rightarrow (E)$
6) $F \rightarrow \text{id}$



0) $E' \rightarrow E$
1) $E \rightarrow E + T$
2) $E \rightarrow T$
3) $T \rightarrow T * F$
4) $T \rightarrow F$
5) $F \rightarrow (E)$
6) $F \rightarrow \text{id}$

引入这个新的开始产生式的目的是使得文法开始符号仅出现在一个产生式的左边, 从而使

得分析器只有一个接受状态

文法中的项目

① $S' \rightarrow S$ ② $S \rightarrow vI:T$ ③ $I \rightarrow I,i$ ④ $I \rightarrow i$ ⑤ $T \rightarrow r$

初始项目

(2) $S \rightarrow \cdot vI:T$

(3) $S \rightarrow v \cdot I:T$

(4) $S \rightarrow vI \cdot :T$

(7) $I \rightarrow \cdot I,i$

(8) $I \rightarrow I \cdot ,i$

(9) $I \rightarrow I,i \cdot$

(11) $I \rightarrow \cdot i$

(12) $I \rightarrow i \cdot$

(13) $T \rightarrow \cdot r$

(14) $T \rightarrow r \cdot$

(0) $S' \rightarrow \cdot S$

(1) $S' \rightarrow S \cdot$

(5) $S \rightarrow vI: \cdot T$

(6) $S \rightarrow vI:I \cdot$

(10) $I \rightarrow I:I \cdot$

归约项目

接收项目

① $S' \rightarrow S$	② $S \rightarrow vI:T$	③ $I \rightarrow I, i$	④ $I \rightarrow i$	⑤ $T \rightarrow r$
----------------------	------------------------	------------------------	---------------------	---------------------

<div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">初始项目</div> <div style="border: 1px solid black; padding: 2px;">接收项目</div>	<div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">(2) $S \rightarrow \cdot vI:T$</div> <div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">(3) $S \rightarrow v \cdot I:T$</div> <div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">(4) $S \rightarrow vI \cdot :T$</div> <div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">(5) $S \rightarrow vI: \cdot T$</div> <div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">(6) $S \rightarrow vI:T \cdot$</div>	<div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">(7) $I \rightarrow \cdot I, i$</div> <div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">(8) $I \rightarrow I \cdot , i$</div> <div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">(9) $I \rightarrow I, \cdot i$</div> <div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">(10) $I \rightarrow I, i \cdot$</div>	<div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">(11) $I \rightarrow \cdot i$</div> <div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">(12) $I \rightarrow i \cdot$</div>	<div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">(13) $T \rightarrow \cdot r$</div> <div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">(14) $T \rightarrow r \cdot$</div>
---	--	---	---	---

(0) $S' \rightarrow \cdot S$ (1) $S' \rightarrow S \cdot$

(1) $S' \rightarrow S \cdot$ (2) $S \rightarrow \cdot vI:T$ (3) $S \rightarrow v \cdot I:T$ (4) $S \rightarrow vI \cdot :T$ (5) $S \rightarrow vI: \cdot T$ (6) $S \rightarrow vI:T \cdot$

(7) $I \rightarrow \cdot I, i$ (8) $I \rightarrow I \cdot , i$ (9) $I \rightarrow I, \cdot i$ (10) $I \rightarrow I, i \cdot$

(11) $I \rightarrow \cdot i$ (12) $I \rightarrow i \cdot$ (13) $T \rightarrow \cdot r$ (14) $T \rightarrow r \cdot$

初始项目: (0), (1), (2), (3), (4), (5), (6), (7), (8), (9), (10), (11), (12), (13), (14)
 归约项目: (2), (3), (4), (5), (6), (7), (8), (9), (10), (11), (12), (13), (14)
 接收项目: (1)

后继项目 (Successive Item)

① $S' \rightarrow S$	② $S \rightarrow vI:T$	③ $I \rightarrow I, i$	④ $I \rightarrow i$	⑤ $T \rightarrow r$
----------------------	------------------------	------------------------	---------------------	---------------------

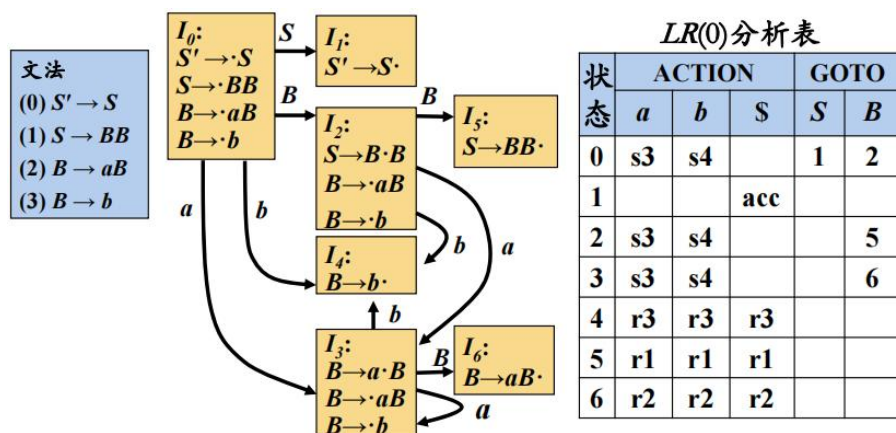
这15个项目中是否会有某些项目是等价的？

可以把等价的项目组成一个项目集(I)，称为**项目集闭包** (Closure of Item Sets)，每个项

例：LR(0)自动机

4-11 LR(0)分析表构造算法（closure 函数、goto 函数、冲突）

例：LR(0)自动机



CLOSURE()函数

计算给定项目集 I 的闭包

$$\text{CLOSURE}(I) = I \cup \{B \rightarrow \cdot \gamma \mid A \rightarrow \alpha \cdot B \beta \in \text{CLOSURE}(I), B \rightarrow \gamma \in P\}$$

```

SetOfItems CLOSURE ( I ) {
    J = I;
    repeat
        for ( J中的每个项  $A \rightarrow \alpha \cdot B \beta$  )
            for ( G的每个产生式  $B \rightarrow \gamma$  )
                if ( 项  $B \rightarrow \cdot \gamma$  不在 J 中 )
                    将  $B \rightarrow \cdot \gamma$  加入 J 中;
    until 在某一轮中没有新的项被加入到 J 中;
    return J;
}
    
```

GOTO ()函数

返回项目集 I 对应于文法符号 X 的后继项目集闭包

$$\text{GOTO}(I, X) = \text{CLOSURE}(\{A \rightarrow \alpha X \cdot \beta \mid A \rightarrow \alpha \cdot X \beta \in I\})$$

```

SetOfItems GOTO ( I, X ) {
    将 J 初始化为空集;
    for ( I 中的每个项  $A \rightarrow \alpha \cdot X \beta$  )
        将项  $A \rightarrow \alpha X \cdot \beta$  加入到集合 J 中;
    return CLOSURE ( J );
}
    
```


构造 $LR(0)$ 自动机的状态集

规范 $LR(0)$ 项集族(*Canonical $LR(0)$ Collection*)

$$C = \{ I_0 \} \cup \{ I \mid \exists J \in C, X \in V_N \cup V_T, I = GOTO(J, X) \}$$

```
void items(  $G'$  ) {  
     $C = \{ \text{CLOSURE}(\{ [ S' \rightarrow \cdot S ] \} ) \}$ ;  
    repeat  
        for ( $C$  中的每个项集  $I$ )  
            for (每个文法符号  $X$ )  
                if (  $GOTO(I, X)$  非空且不在  $C$  中)  
                    将  $GOTO(I, X)$  加入  $C$  中;  
    until 在某一轮中没有新的项集被加入到  $C$  中;  
}
```

$LR(0)$ 分析表构造算法

- 构造 G' 的规范 $LR(0)$ 项集族 $C = \{ I_0, I_1, \dots, I_n \}$
- 令 I_i 对应状态 i 。状态 i 的语法分析动作按照下面的方法决定:

if $A \rightarrow \alpha \cdot a \beta \in I_i$ and $GOTO(I_i, a) = I_j$ *then* $ACTION[i, a] = sj$

if $A \rightarrow \alpha \cdot B \beta \in I_i$ and $GOTO(I_i, B) = I_j$ *then* $GOTO[i, B] = j$

if $A \rightarrow \alpha \cdot \in I_i$ 且 $A \neq S'$ *then for* $a \in VT \cup \{ \$ \}$ *do* $ACTION[i, a] = rj$

(j 是产生式 $A \rightarrow \alpha$ 的编号)

if $S' \rightarrow S \cdot \in I_i$ *then* $ACTION[i, \$] = acc$

- 没有定义的所有条目都设置为“error”

LR(0) 自动机的形式化定义

・文法

$$G = (VN, VT, P, S)$$

• LR(0) 自动机

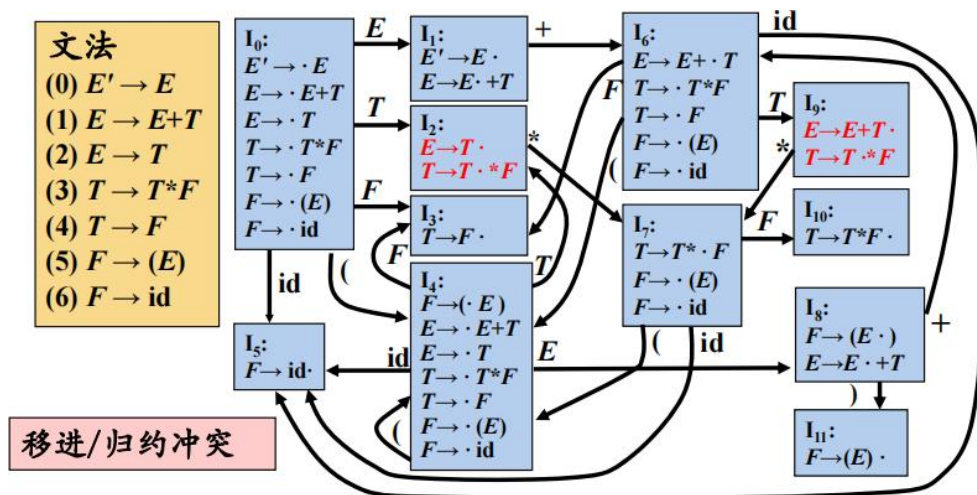
$$M = (C, VN \cup VT, GOTO, I_0, F)$$

$$C = \{I0\} \cup \{I \mid J \in C, X \in VN \cup VT, I = GOTO(J, X)\}$$

- $I0 = \text{CLOSURE}(\{S' \rightarrow \cdot S\})$

- $F = \{ \text{CLOSURE}(\{S' \rightarrow S.\}) \}$

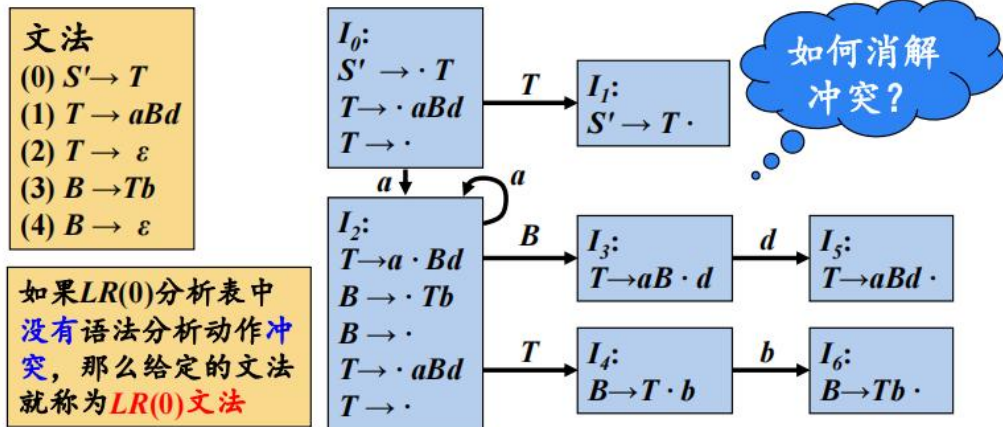
LR(0) 分析过程中的冲突



表达式文法的 $LR(0)$ 分析表含有移进/归约冲突

[illegible]

例：移进/归约冲突和归约/归约冲突



不是所有CFG都能用LR(0)方法进行分析，也就是说，CFG不总是LR(0)文法