



编译原理复习讲座 (1)

——绪论到语法分析



计算学部 金牌讲师团 尹博
2024年5月



说明：

本次复习讲座主要包括绪论、语言及文法、词法分析和语法分析的内容。



每一部分包括知识点回顾和相关题目的理解。讲座内容偏向**基础**：基础的知识点和一些类型题目。

本人水平十分有限，讲解如有疏漏、错误，恳请各位批评指正！多多包涵，感谢😊

主要内容:

知识点回顾

一、绪论

编译的概念

将高级语言翻译成汇编语言或机器语言的过程

编译系统的组成

8大部分

T形图

二、语言及其文法

一些基本概念

文法的定义

语言的定义

文法分类

CFG的语法分析树

三、词法分析

单词的描述

单词的识别

词法分析阶段的错误处理

四、语法分析

自顶向下的分析

预测分析法

自底向上的分析

LR分析法



主要内容:

各个章节相关习题:

- 绪论: T 1.1^[1], T 1.2, T 1.3以及SPOC中一些题目
- 语言及其文法: SPOC中的一些题目
- 词法分析: T 3.1, 2020(B)-一
- 语法分析: T 4.2(1), T 4.3, T 5.1(3), T 6.1(2), T6.2(1)
T 7.1(1), T 7.2, T 7.5(3), T 7.6, T 7.7 , SPOC

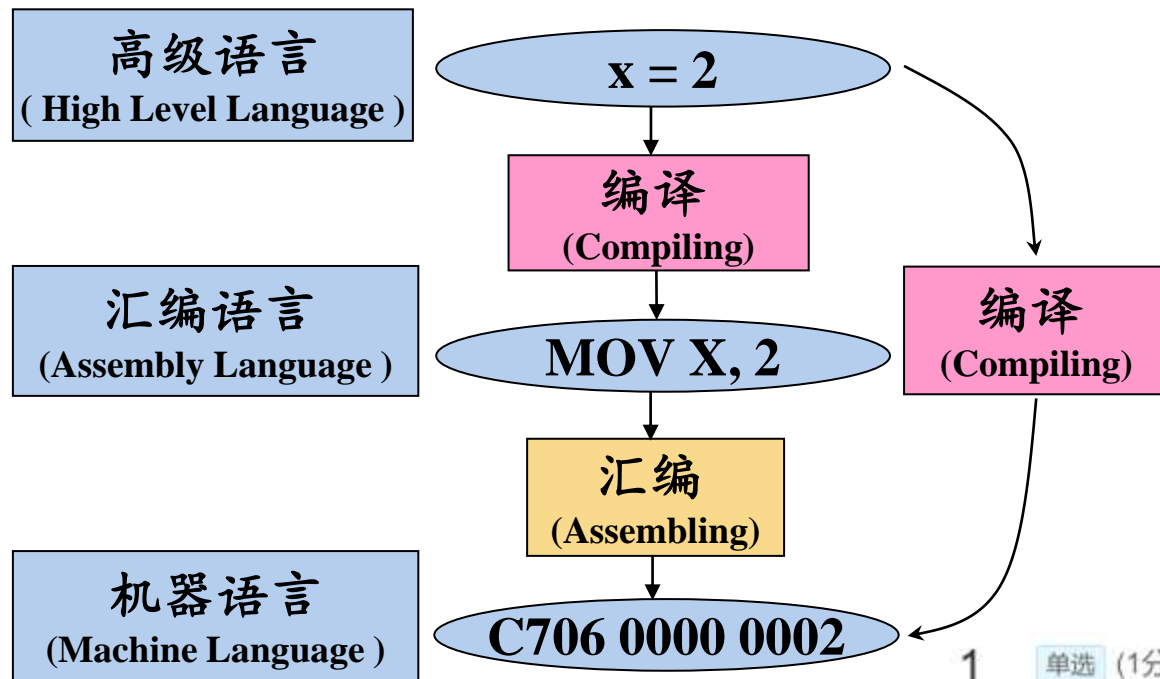
[1] T 指取自《编译原理》习题(带红色星号的为选做).pdf`中的题目, 题号相对应



第一章： 绪论



- 知识点回顾
- 编译的概念



编译：将 高级语言 翻译成 汇编语言或机器语言 的过程
源语言 目标语言

1 单选 (1分) 把汇编语言程序翻译成机器可执行的目标程序的工作是由()完成的。

- A. 解释器
- B. 预处理器
- C. 编译器
- D. 汇编器

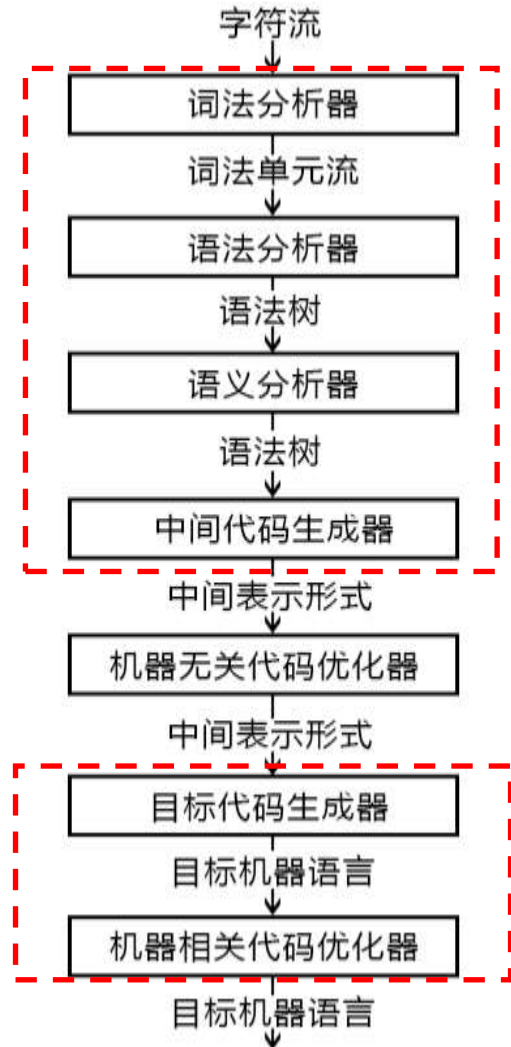
- 知识点回顾 

- 编译系统的八大组成部分

分析部分/
前端(front end):
与源语言相关

- 表格管理
- 出错处理

综合部分/
后端(back end):
与目标语言相关





• 习题

• SPOC – 第1讲

3 **单选** (1分) 通常一个编译程序中，不仅包含词法分析，语法分析，语义分析，中间代码生成，代码优化，目标代码生成等六个部分，还应包括()。

- A. 解释器
- B. 模拟执行器
- C. 表格处理和出错处理
- D. 符号执行器

2.编译程序绝大多数时间花在（表格管理）上。

6 **单选** (1分) 将编译程序分成若干“遍”，是为了()。

得分/总分

- A. 利用有限的机器内存并提高机器的执行效率
- B. 利用有限的机器内存但降低了机器的执行效率
- C. 使程序的结构更为清晰
- D. 提高程序的执行效率

✓1.00/1.00



• 习题

• SPOC – 第1讲

1 编译是对()。

- A. 机器语言的执行
- B. 汇编语言的翻译
- C. 高级语言的翻译
- D. 高级语言程序的解释执行

正确答案：C 你选对了

2 用高级语言编写的程序经编译后产生的程序叫()。

- A. 源程序
- B. 目标程序
- C. 连接程序
- D. 解释程序

正确答案：B 你选对了

3 ()不是编译程序的组成部分。

- A. 词法分析程序
- B. 代码生成程序
- C. 设备管理程序
- D. 语法分析程序

正确答案：C 你选对了

4 源程序是句子的集合,()可以较好地反映句子的结构。

- A. 线性表
- B. 树
- C. 完全图
- D. 堆栈

正确答案：B 你选对了

- 知识点回顾 

- 词法分析的主要任务

从左向右逐行扫描源程序的字符，识别出各个单词，确定**单词的类型**。

将识别出的单词转换成统一的**机内表示**——词法单元(token)形式

token: < 种别码, 属性值 >

| | 单词类型 | 种别 | 种别码 |
|---|------|--|-------------------|
| 1 | 关键字 | program、if、else、then、... | 一词一码 |
| 2 | 标识符 | 变量名、数组名、记录名、过程名、... | 多词一码 |
| 3 | 常量 | 整型、浮点型、字符型、布尔型、... | 一型一码 |
| 4 | 运算符 | 算术 (+ - * / ++ --) 关系 (> < == != >= <=) 逻辑 (& ~) | 一词一码 或 一型一码 |
| 5 | 界限符 | ; () = { } ... | 一词一码 |



- 知识点回顾 

- 语义分析的主要任务

- 收集标识符的属性信息

- 种属 (Kind)

- 类型 (Type)

- 存储位置、长度

- 值

- 作用域

- 参数和返回值信息

- 语义检查

- 变量 (包括数组、指针、结构体) 或过程 **未经声明就使用**

- 变量 (包括数组、指针、结构体) 或过程名 **重复声明**

- **运算分量**类型不匹配

- **操作符**与**操作数**之间的类型不匹配



• 习题

• SPOC – 第1讲

9 语法分析时所依据的是()

- A. 语法规则
- B. 词法规则
- C. 语义规则
- D. 等价变换规则

正确答案: A 你选对了

6 按逻辑上划分, 编译程序第三步工作是()。

- A. 语义分析
- B. 词法分析
- C. 语法分析
- D. 代码生成

正确答案: A 你选对了

7 编译程序中语法分析器接收以()为单位的输入。

- A. 单词
- B. 表达式
- C. 产生式
- D. 句子


正确答案: A 你选对了

8 编译过程中, 语法分析器的任务就是()。

- A. 分析单词是怎样构成的
- B. 分析单词串是如何构成语句和声明的
- C. 分析语句和声明是如何构成程序的
- D. 分析程序的结构

正确答案: B 你选对了



- 知识点回顾 
- 中间代码生成

➤ 三地址码 (Three-address Code)

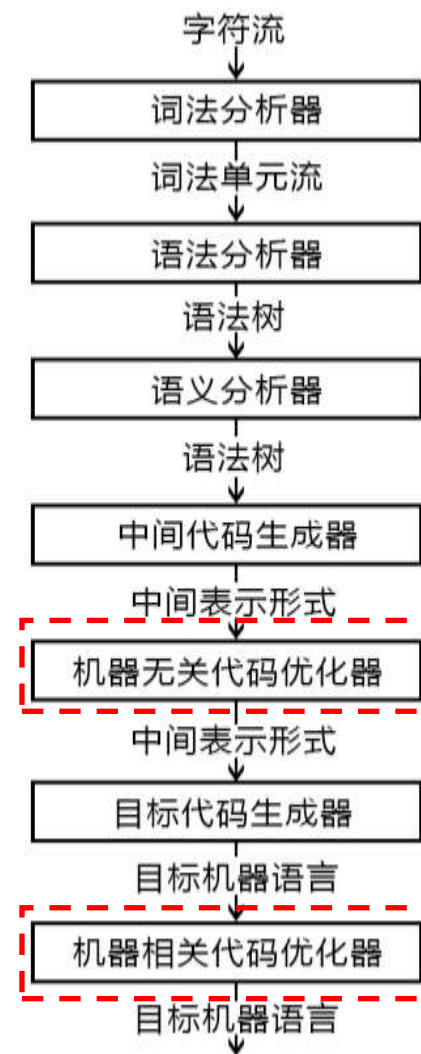
➤ 三地址码由类似于汇编语言的指令序列组成，每个指令最多有三个操作数

➤ 语法结构树/语法树 (Syntax Trees)


- 知识点回顾 


- 代码优化

- 为改进代码所进行的等价程序变换，使其运行得更快一些、占用空间更少一些，或者二者兼顾





- 知识点回顾 
- 目标代码生成
 - 目标代码生成以源程序的**中间表示形式**作为输入，并把它映射到**目标语言**
 - 目标代码生成的一个重要任务是为程序中使用的变量**合理分配寄存器**

- 知识点回顾 
- 编译器的T形图

➤ P: 编译器 (程序)

➤ I: 实现语言

➤ S: 输入的源语言程序

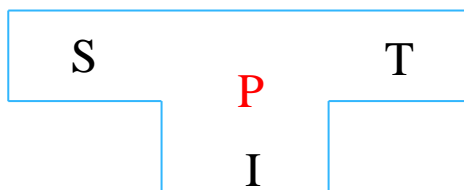
➤ T: 输出的可执行的目标程序

注意:

(1) I表示的是语言, 而S和T表示的是程序

(2) T形图的上端体现了编译器的功能, 即从哪种语言到哪种语言的翻译

(3) T对应于某机器语言





- 习题 

- T 1.1

➤ 将下面的C++程序划分成正确的词素 (token) 序列。
哪些词素应该有相关联的属性值? 应该具有什么值?

```
float limitedSquare(x) {float x ;  
/* returns x-squared, but never more than 100 */  
return (x<=-10.01 ||x>=10.0)?100:x*x;  
}
```



• 习题 

• T 1.1

| | | | |
|---------------|----------------------|---------------------|----------------------|
| float | <FLOAT, _ > | 这里算术、逻辑、关系运算符采用一型一码 | |
| limitedSquare | <IDN, limitedSquare> | | |
| (| <SLP, _ > | |) |
| x | <IDN, x> | | ? <CONP, ?> |
|) | <SRP, _ > | | 100 <CONST_INT, 100> |
| { | <LP, -> | | : |
| float | <FLOAT, _ > | | x <IDN, x> |
| x | <IDN, x> | | * <ARITHMETICAL, * > |
| ; | <SEMI, _ > | | x <IDN, x> |
| return | <RETURN, _ > | | ; |
| (| <SLP, _ > | | <SEMI, _ > |
| x | <IDN, x> | | } |
| <= | <RELOP, <= > | | <RP, _ > |
| - | <ARITHMETICAL, - > | | |
| 10.01 | <CONST_FLOAT, 10.01> | | |
| | <LOGICAL, > | | |
| x | <IDN, x> | | |
| >= | <RELOP, >= > | | |
| 10.0 | <CONST_FLOAT, 10.0> | | |

`CONP` 指条件运算符

• 习题



• T 1.1



若算术、逻辑、关系运算符采用一词一码

| | |
|---------------|----------------------|
| float | <FLOAT, _> |
| limitedSquare | <IDN, limitedSquare> |
| (| <SLP, _> |
| x | <IDN, x> |
|) | <SRP, _> |
| { | <LP, _> |
| float | <FLOAT, _> |
| x | <IDN, x> |
| ; | <SEMI, _> |
| return | <RETURN, _> |
| (| <SLP, _> |
| x | <IDN, x> |
| <= | <LE, _> |
| - | <MINUS, _> |
| 10.01 | <CONST_FLOAT, 10.01> |
| | <OR, _> |
| X | <IDN, x> |
| >= | <GE, _> |
| 10.0 | <CONST_FLOAT, 10.0> |

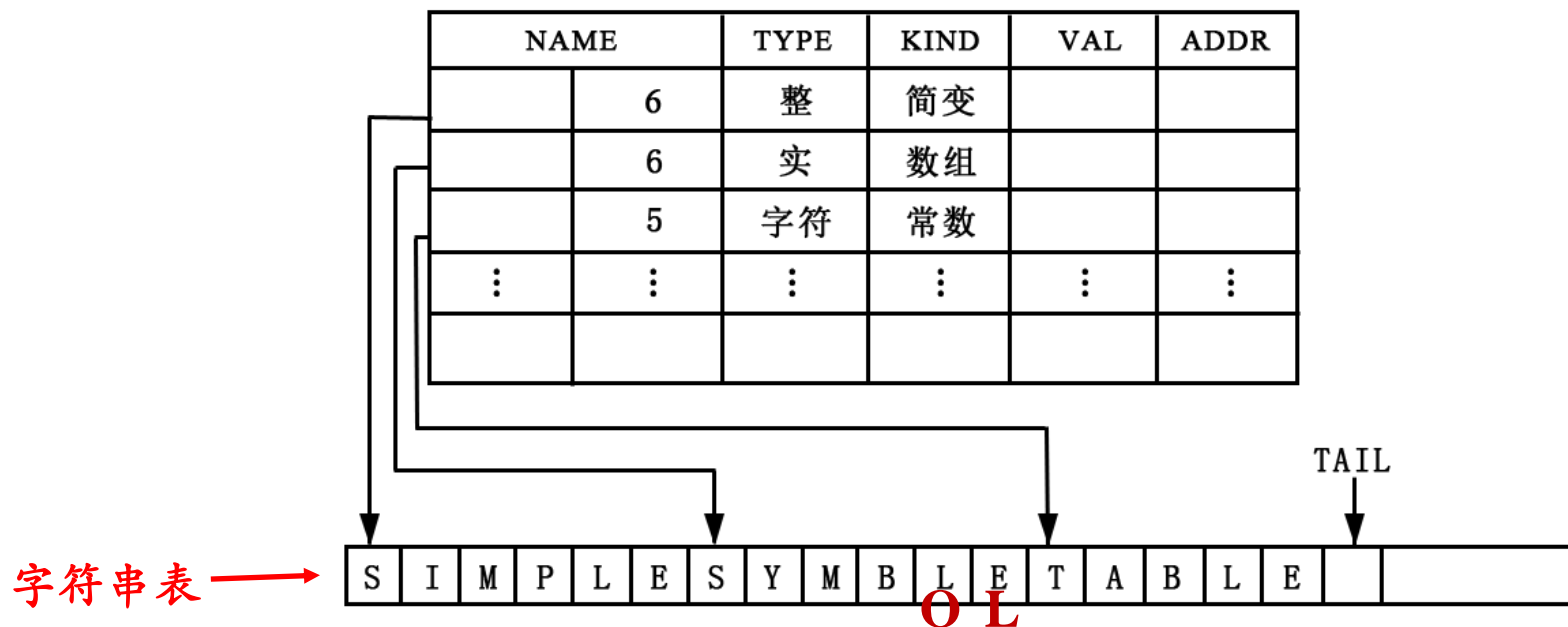
| | |
|-----|------------------|
|) | <SRP, _> |
| ? | <CONP, ?> |
| 100 | <CONST_INT, 100> |
| : | <CONP, :> |
| x | <IDN, x> |
| * | <MUL, _> |
| x | <IDN, x> |
| ; | <SEMI, _> |
| } | <RP, _> |

- 习题


- T 1.2

➤ 符号表中NAME字段为什么要设计字符串表这样一种数据结构？而不是把标识符对应的字符串直接存放到NAME字段？

符号表 (Symbol Table)





- 习题 

- T 1.2

➤ 符号表中NAME字段为什么要设计字符串表这样一种数据结构？而不是把标识符对应的字符串直接存放到NAME字段？

若把标识符对应的字符串直接存放到NAME字段，
考虑：定长OR变长

定长：太长的→存不下，溢出；

太短的→大量空间空闲，造成空间浪费；

变长：查询效率低下。

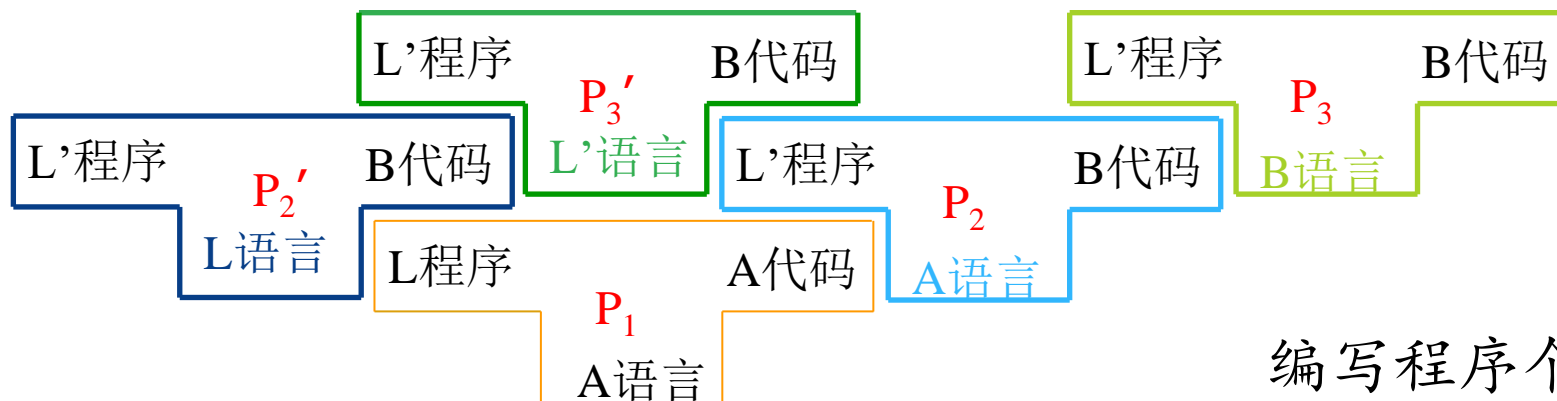
其他考量因素：便于字符串的统一管理

• 习题

• T 1.3

➤ 设A机器上有语言L的编译程序，可以用它来编制B机器上的语言L'的编译程序，试用T形图进行表示（4种方法）

方法一：(目标导向)



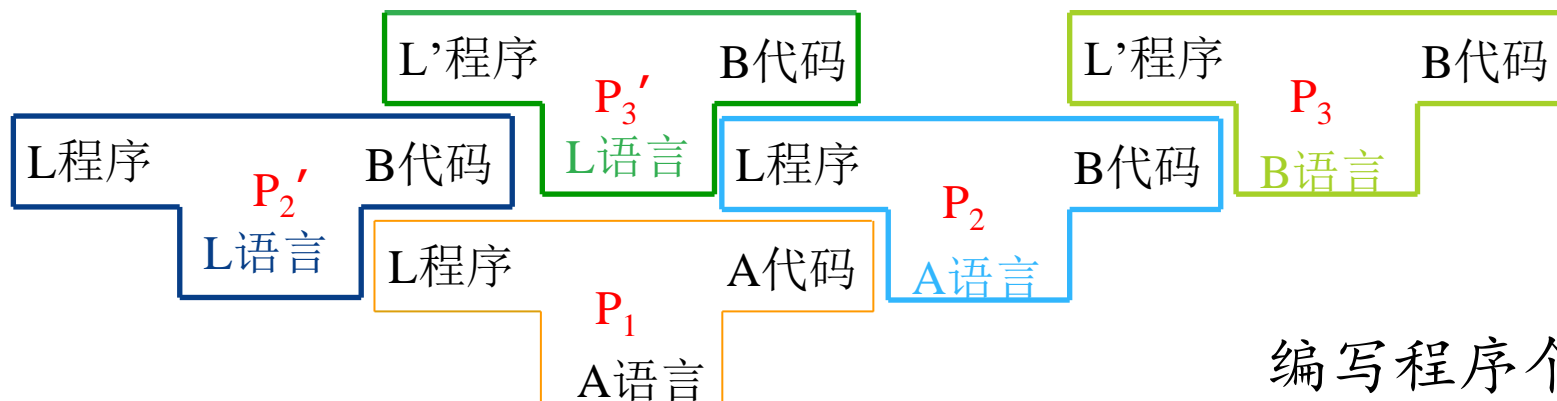
编写程序个数：2个 (P_2' , P_3')
 所使用语言：L、L' (两种)
 编译次数：2

• 习题

• T 1.3

➤ 设A机器上有语言L的编译程序，可以用它来编制B机器上的语言L'的编译程序，试用T形图进行表示（4种方法）

方法二：（非目标导向） 综合最优



编写程序个数：2个 (P_2' , P_3')
 所使用语言：L (一种)
 编译次数：2

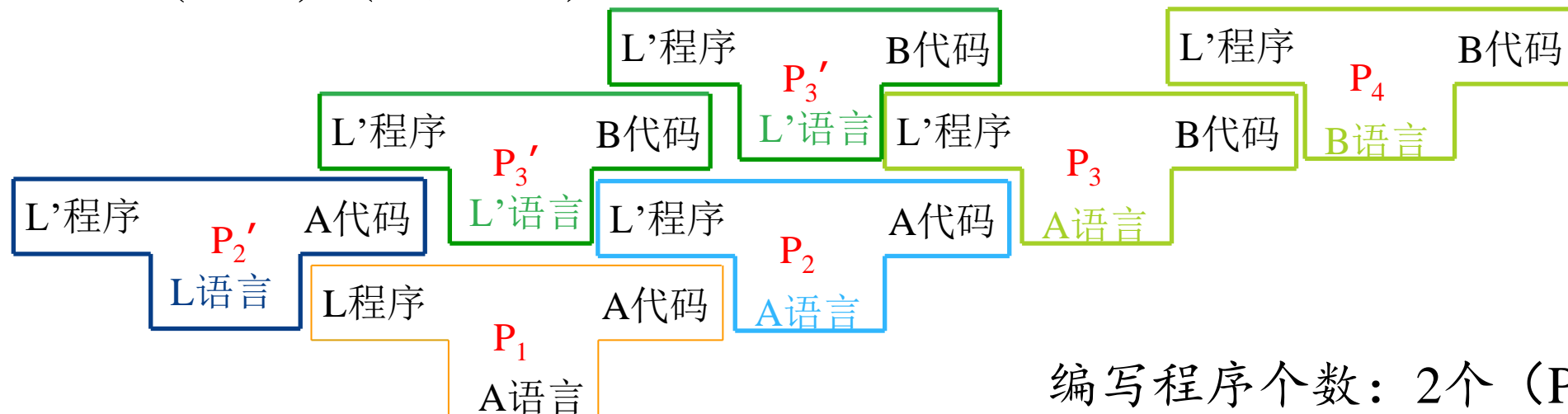
• 习题

• T 1.3

➤ 设A机器上有语言L的编译程序，可以用它来编制B机器上的语言L'的编译程序，试用T形图进行表示（4种方法）

方法三：先自展后移植

(L→L') (机器A→B)



编写程序个数：2个 (P₂' , P₃')
 所使用语言：L、L' (两种)
 编译次数：3

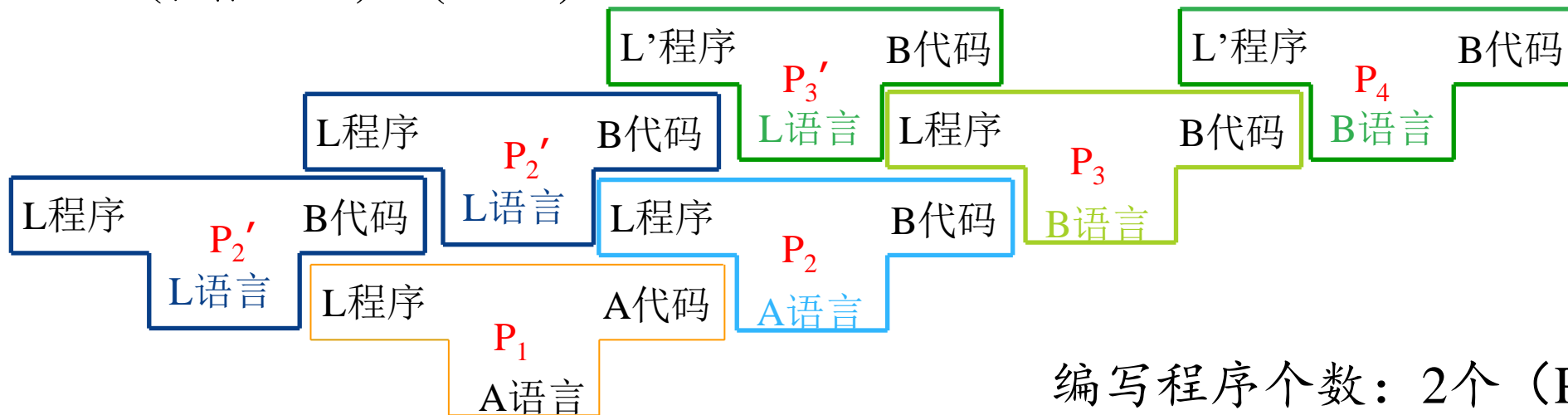
• 习题

• T 1.3

➤ 设A机器上有语言L的编译程序，可以用它来编制B机器上的语言L'的编译程序，试用T形图进行表示（4种方法）

方法四：先移植后自展

(机器A→B) (L→L')



编写程序个数：2个 (P₂' , P₃')

所使用语言：L (一种)

编译次数：3



第二章： 语言及其文法

- 知识点回顾 

- 一些基本概念

- 串 (*String*)

- 语言上的运算

- 串上的运算

- 连接、幂运算


- 字母表

- 字母表上的运算

| 运算 | 定义和表示 |
|---------------|---|
| L 和 M 的并 | $LUM = \{s \mid s \text{属于} L \text{或者} s \text{属于} M\}$ |
| L 和 M 的连接 | $LM = \{st \mid s \text{属于} L \text{且} t \text{属于} M\}$ |
| L 的幂 | $\begin{cases} L^0 = \{\varepsilon\} \\ L^n = L^{n-1}L, n \geq 1 \end{cases}$ |
| L 的Kleene闭包 | $L^* = \bigcup_{i=0}^{\infty} L^i$ |
| L 的正闭包 | $L^+ = \bigcup_{i=1}^{\infty} L^i$ |

推导 (*Derivations*) 和 归约 (*Reductions*)



- 知识点回顾 

- 语言、文法

$$G = (V_T, V_N, P, S)$$

➤ V_T : 终结符集合

➤ V_N : 非终结符集合

➤ P : 产生式集合

➤ S : 开始符号

符号约定

| | | | |
|------|-----------|-------|-------------------------|
| 终结符 | a, b, c | 终结符号串 | u, v, \dots, z |
| 非终结符 | A, B, C | | |
| 文法符号 | X, Y, Z | 文法符号串 | α, β, γ |

➤ 由文法 G 的开始符号 S 推导出的所有句子构成的集合称为**文法 G 生成的语言**，记为 $L(G)$ 。即

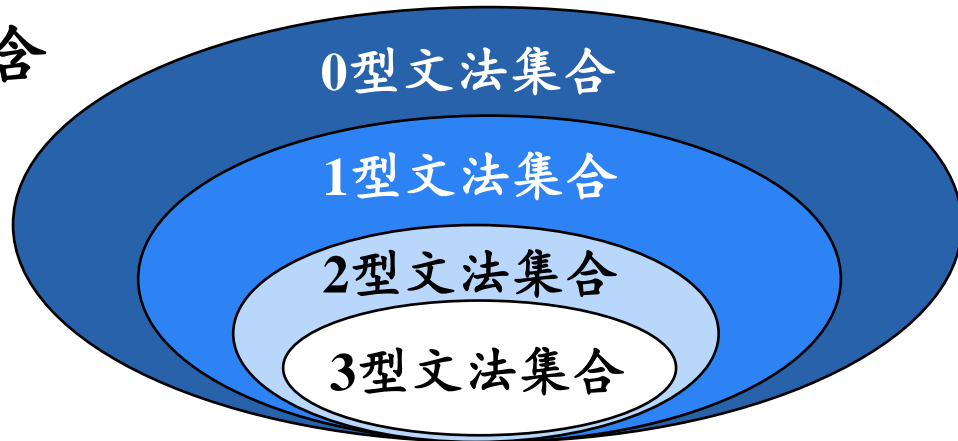
$$L(G) = \{w / S \Rightarrow^* w, w \in V_T^*\}$$

- 知识点回顾 
- 文法的分类 (Chomsky体系)

➤ 逐级限制

- 0型文法: α 中至少包含1个非终结符
- 1型文法 (CSG) : $|\alpha| \leq |\beta|$
- 2型文法 (CFG) : $\alpha \in V_N$
- 3型文法 (RG) : $A \rightarrow wB$ 或 $A \rightarrow w$ ($A \rightarrow Bw$ 或 $A \rightarrow w$)

➤ 逐级包含





- 知识点回顾 

- CFG的分析树

- 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*)

直接短语一定是某产生式的右部

但产生式的右部不一定是给定句型的直接短语



• 习题

• SPOC – 第2讲

5 若文法G定义的语言是无限集,则文法必然是()。

- A. 递归的
- B. 上下文无关的
- C. 二义性的
- D. 无二义性的

正确答案: A 你选对了

8 若一个文法是递归的,则它所产生的语言的句子()。

- A. 是无穷多个
- B. 是有穷多个
- C. 是可枚举的
- D. 个数是常量

正确答案: A 你选对了

6 乔姆斯基(Chomsky)把文法分为四种类型,即0型、1型、2型、3型。其中3型文法是()。

- A. 非限制文法
- B. 正则文法
- C. 上下文有关文法
- D. 上下文无关文法

正确答案: B 你选对了

• 习题 

• SPOC – 第2讲

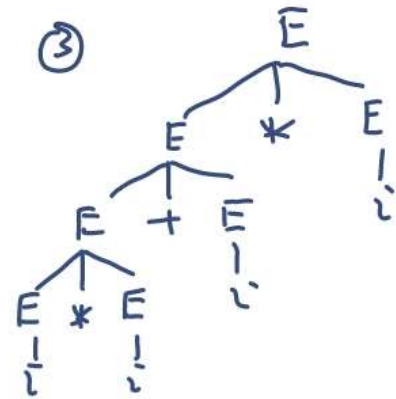
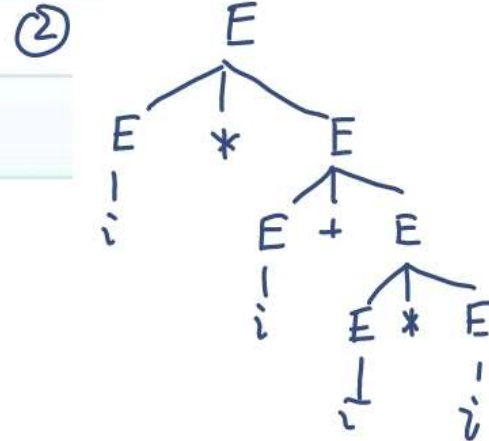
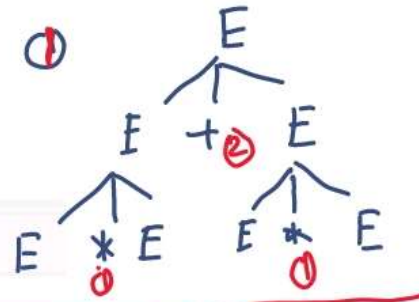
10 文法 $E \rightarrow E+E | E * E | i$ 的句子 $i * i + i * i$ 有 () 棵不同的语法树。



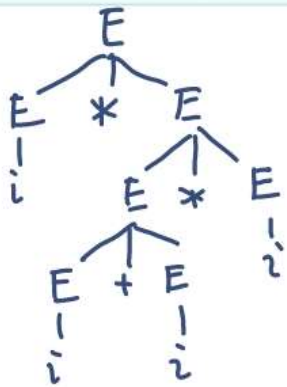
- A. 1
- B. 3
- C. 5
- D. 7

正确答案: C 你选对了

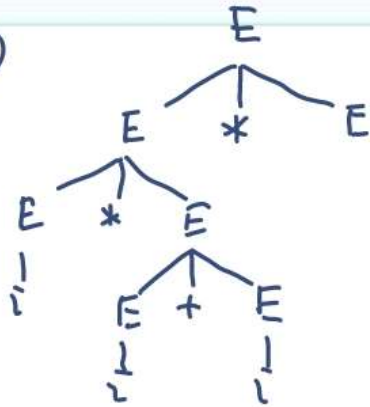
$i * i + i * i$



④




⑤





第三章： 词法分析

- 知识点回顾 
- 单词的描述与识别

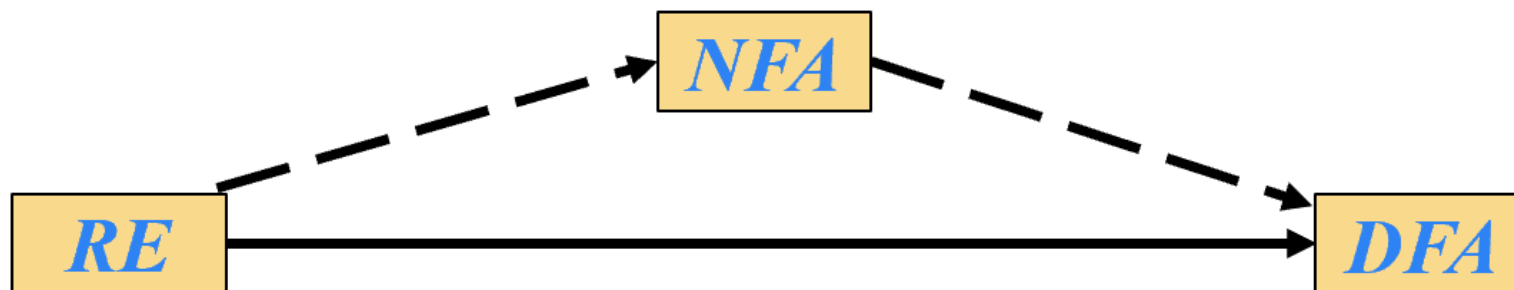
➤ 正则文法(RG)

➤ 正则表达式(*Regular Expression*, *RE*)

➤ 正则定义(RD)

➤ 确定的*FA* (*Deterministic finite automata*, *DFA*)

➤ 非确定的*FA* (*Nondeterministic finite automata*, *NFA*)





- 知识点回顾 
- 词法分析阶段的错误处理

➤ 词法错误的类型

➤ 单词拼写错误

➤ 例：`int i = 0x3G; float j = 1.05e;`

➤ 非法字符

➤ 例：`~ @`

➤ 词法错误检测

➤ 如果当前状态与当前输入符号在转换表对应项中的信息为空，而当前状态又不是终止状态，则调用错误处理程序

➤ 最简单的错误恢复策略：“恐慌模式 (panic mode)” 恢复

➤ 从剩余的输入中不断删除字符，直到词法分析器能够在剩余输入的开头发现一个正确的字符为止



- 习题



- T 3.1

➤ 为“注释”设计一个DFA。注释是/*和*/之间的串，且串中没有不在双引号(“)中的*/



- 习题 

- 2020年春(B) - 一、

➤ 假设“注释”定义为//和//之间的串，且串中没有不在双引号(“)中的//。设计接收该“注释”的DFA。（转换图形式）



第四章： 语法分析





- 知识点回顾 

- 自顶向下的分析(Top-Down Parsing)

- 从分析树的顶部（根节点）向底部（叶节点）方向构造分析树
- 可以看成是从文法开始符号 S 推导出词串 w 的过程
- 推导的每一步，都需要做两个选择
 - 替换当前句型中的哪个非终结符
 - 用该非终结符的哪个候选式进行替换

➤ 在最左推导中，总是选择每个句型的最左非终结符进行替换

➤ 如果 $S \Rightarrow_{lm}^* \alpha$ ，则称 α 是当前文法的最左句型(leftmost-sentential form)



- 知识点回顾 
- 消除左递归与提取左公因子

$$A \rightarrow A\alpha_1 | A\alpha_2 | \dots | A\alpha_n | \beta_1 | \beta_2 | \dots | \beta_m$$

($\alpha_i \neq \varepsilon$, β_j 不以A开头)

$$A \rightarrow \beta_1 A' | \beta_2 A' | \dots | \beta_m A'$$

$$A' \rightarrow \alpha_1 A' | \alpha_2 A' | \dots | \alpha_n A' | \varepsilon$$



- 知识点回顾 
- 递归下降分析 (Recursive-Descent Parsing)

自顶向下语法分析的通用形式：

- 递归下降分析 (*Recursive-Descent Parsing*)
 - 由一组过程组成，每个过程对应文法的一个非终结符
 - 从文法开始符号 S 对应的过程开始，其中递归调用文法中其它非终结符对应的过程。如果 S 对应的过程体恰好扫描了整个输入串，则成功完成语法分析



- 知识点回顾 

- FIRST集、FOLLOW集、SELECT集

- 非终结符A的FOLLOW集

- 可能在某个句型中紧跟在A后边的终结符a的集合，记为**FOLLOW(A)**

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

- 产生式 $A \rightarrow \beta$ 的**可选集**是指可以选用该产生式进行推导时对应的**输入符号**的集合，记为**SELECT(A → β)**

- $$SELECT(A \rightarrow a\beta) = \{ a \}$$

- $$SELECT(A \rightarrow \varepsilon) = FOLLOW(A)$$

- 给定一个文法符号串 α ， α 的**串首终结符集****FIRST(α)**被定义为可以从 α 推导得到的**串的首终结符**构成的集合。如果 $\alpha \Rightarrow^* \varepsilon$ ，那么 ε 也在**FIRST(α)**中

- 对于 $\forall \alpha \in (V_T \cup V_N)^+$ ， $FIRST(\alpha) = \{ a \mid \alpha \Rightarrow^* a\beta, a \in V_T, \beta \in (V_T \cup V_N)^* \}$;

- 如果 $\alpha \Rightarrow^* \varepsilon$ ，那么 $\varepsilon \in FIRST(\alpha)$



• 习题

• SPOC – 第4讲

4 **判断** (1分) FIRST集中可以含有 ϵ

A. \checkmark

B. \times

2 **单选** (1分) 在编译过程中, 如果遇到错误应该()

A. 当发现错误时, 跳过错误所在的语法单位继续分析下去

B. 当发现错误时立即停止编译, 待用户改正错误后再继续编译

C. 把错误理解成局部的错误

D. 对错误在局部范围内进行纠正, 继续向下分析

1 **单选** (1分) 已知文法G[S]:

$S \rightarrow eT|RT$ $T \rightarrow DR|\epsilon$ $R \rightarrow dR|\epsilon$ $D \rightarrow a|bd$

求FIRST(S)= ()。

A. {e, d}

B. {e,d,a,b}

C. {e, d, a, b, ϵ }

D. {e}



• 习题 

• SPOC – 第4讲

3 单选 (1分) 已知文法G[S]:
 $S \rightarrow eT \mid RT$ $T \rightarrow DR \mid \varepsilon$ $R \rightarrow dR \mid \varepsilon$ $D \rightarrow a \mid bd$
求FOLLOW(D) = ()。

- A. {a,d}
- B. {d,\$}
- C. {d,e}
- D. {d, ε }



- 习题




- T 4.2 (1)

➤ 为下面的每一个文法设计一个预测分析器，并给出预测分析表。你可能先要对文法进行提取左公因子或消除左递归的操作。计算各文法的 **FIRST** 和 **FOLLOW** 集合。

➤ (1) $S \rightarrow 0 S 1 \mid 0 1$



- 习题 

- T 4.3

➤ 为下面的文法设计一个预测分析器，并给出预测分析表。你可能先要对文法进行提取左公因子或消除左递归的操作。计算文法的 **FIRST** 和 **FOLLOW** 集合。

$$S \rightarrow S S + \mid S S * \mid a$$



- 知识点回顾 

- LL(1) 文法

- 产生式 $A \rightarrow \alpha$ 的可选集 *SELECT*

- 如果 $\varepsilon \notin FIRST(\alpha)$, 那么 $SELECT(A \rightarrow \alpha) = FIRST(\alpha)$

- 如果 $\varepsilon \in FIRST(\alpha)$, 那么 $SELECT(A \rightarrow \alpha) = (FIRST(\alpha) - \{\varepsilon\}) \cup FOLLOW(A)$

- 文法 G 是 **LL(1)** 的, 当且仅当 G 的任意两个具有相同左部的产生式 $A \rightarrow \alpha \mid \beta$ 满足下面的条件:

- 不存在终结符 a , 使得 α 和 β 都能够推导出以 a 开头的串

- α 和 β 至多有一个能推导出 ε

- 如果 $\beta \Rightarrow^* \varepsilon$, 则 $FIRST(\alpha) \cap FOLLOW(A) = \Phi$;

- 如果 $\alpha \Rightarrow^* \varepsilon$, 则 $FIRST(\beta) \cap FOLLOW(A) = \Phi$;

- 知识点回顾 
- 递归&非递归实现预测分析

- 递归的方式：基于预测分析表对递归下降分析法进行扩展
- 非递归的方式：显式地维护一个栈结构来模拟最左推导过程

| | 递归的预测分析法 | 非递归的预测分析法 |
|------|--------------------|----------------------------|
| 程序规模 | 程序规模较大， 不需载入分析表 | 主控程序规模较小， 需载入分析表（表较小） 😊 |
| 直观性 | 较好 😊 | 较差 |
| 效率 | 较低 | 分析时间大约正比于待分析程序的长度 😊 |
| 自动生成 | 较难 | 较易 😊 |



- 知识点回顾 
- 预测分析中的错误恢复

➤ 恐慌模式 (Panic Mode)

- 忽略输入中的一些符号，直到输入中出现由设计者选定的**同步词法单元集合**中的某个词法单元
 - 其效果依赖于**同步集合的选取**。集合的选取应该使得语法分析器能从实际遇到的错误中**快速恢复**
 - 例如可以把**FOLLOW(A)**中的所有终结符放入非终结符A的同步记号集合
- 如果终结符在栈顶而不能匹配，一个简单的办法就是弹出此终结符



• 习题



• T 5.1 (3)

➤ 为下列文法构造递归下降语法分析器（参见SPOC讲义
“第4章 语法分析 - 上.pdf” 第42~48页）

➤ (3) $S \rightarrow 0 S 1 \mid 0 1$



• 习题

• SPOC – 第5讲

2 单选 (1分) 采用自上而下分析, 不必()。

- A. 消除回溯
- B. 消除右递归
- C. 消除左递归
- D. 提取公共左因子

4 单选 (1分) ()文法不是LL(1)的

- A. 右递归
- B. 2型
- C. 含有公共左因子的
- D. 递归

5 单选 (1分) 已知文法G是无二义的, 则对G的任意句型 α ()。

- A. 最左推导和最右推导对应的语法树可能相同
- B. 可能存在两个不同的最左推导, 但他们对应的语法树相同
- C. 最左推导和最右推导必定相同
- D. 最左推导和最右推导对应的语法树必定相同

6 单选 (1分) 一个文法G, 若(), 则称它是LL(1)文法。

- A. G的LL(1)分析表中不含多重定义的条目
- B. G中产生式不含左公因子
- C. G中不含左递归
- D. G无二义性



- 知识点回顾 

- 自底向上的语法分析

- 从分析树的底部(叶节点)向顶部(根节点)方向构造分析树

- 可以看成是将输入串 w 归约为文法开始符号 S 的过程

- 自顶向下的语法分析采用最左推导方式 (构造句子的最左推导)

- 自底向上的语法分析采用最左归约方式 (反向构造句子的最右推导)

- 自底向上语法分析的通用框架

- 移入-归约分析(*Shift-Reduce Parsing*)

栈内符号串 + 剩余输入 = “规范句型”



• 知识点回顾 

• LR分析法——LR(0)分析

➤ 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

➤ $S \rightarrow b \cdot BB$

➤ $S \rightarrow bB \cdot B$ } 待约项目

➤ $S \rightarrow bBB \cdot$ ← 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 $A \rightarrow \cdot$

6 单选 (1分) 一个**项目**指明了在LR分析过程中的某个时刻所能看到产生式多大一部分。

A. 项目

- 知识点回顾 

- LR分析法——SLR(1)分析

- SLR分析法的基本思想

已知项目集 I :

$A_1 \rightarrow \alpha_1 \cdot \alpha_1 \beta_1$

$A_2 \rightarrow \alpha_2 \cdot \alpha_2 \beta_2$

...

$A_m \rightarrow \alpha_m \cdot \alpha_m \beta_m$

} m 个移进项目

$B_1 \rightarrow \gamma_1 \cdot$

$B_2 \rightarrow \gamma_2 \cdot$

...

$B_n \rightarrow \gamma_n \cdot$

} n 个归约项目

如果集合 $\{a_1, a_2, \dots, a_m\}$ 和 $FOLLOW(B_1)$, $FOLLOW(B_2)$, \dots , $FOLLOW(B_n)$ 两两不相交, 则项目集 I 中的冲突可以按以下原则解决:

设 a 是下一个输入符号

- 若 $a \in \{a_1, a_2, \dots, a_m\}$, 则移进 a

- 若 $a \in FOLLOW(B_i)$, 则用产生式 $B_i \rightarrow \gamma_i$ 归约

- 此外, 报错

- 知识点回顾 

- LR分析法——LR(1)分析

- *SLR*分析存在的问题

- *SLR*只是简单地考察下一个输入符号 **b** 是否属于与归约项目 **$A \rightarrow \alpha$** 相关联的 **$FOLLOW(A)$** ，但 **$b \in FOLLOW(A)$** 只是归约 **α** 的一个**必要条件**，而非**充分条件**

- 将一般形式为 $[A \rightarrow \alpha \cdot \beta, a]$ 的项称为 **LR(1) 项**，其中 $A \rightarrow \alpha \beta$ 是一个产生式， **a** 是一个**终结符**(这里将 **$\$$** 视为一个特殊的终结符)它表示在当前状态下， A 后面必须紧跟的终结符，称为该项的**展望符**(lookahead)

- 在形如 $[A \rightarrow \alpha \cdot \beta, a]$ 且 $\beta \neq \epsilon$ 的项中，展望符 **a** 没有任何作用

- 但是一个形如 $[A \rightarrow \alpha \cdot, a]$ 的项在**只有在下一个输入符号等于 **a**** 时才可以按照 $A \rightarrow \alpha$ 进行归约



- 知识点回顾 

- LR分析法

➤ *if* $[A \rightarrow \alpha \cdot, c] \in I_i$

$$\left\{ \begin{array}{l} A = S' \quad ACTION [i, \$] = acc \\ A \neq S' \left\{ \begin{array}{l} LR(0) \quad for \forall a \in V_T \cup \{ \$ \} \quad do ACTION [i, a] = r_j \quad (j \text{ 是产生式 } A \rightarrow a \text{ 的编号}) \\ SLR(1) \quad for \forall a \in FOLLOW(A) \quad do ACTION [i, a] = r_j \\ LR(1) \quad 精准归约 \quad \quad \quad ACTION [i, c] = r_j \end{array} \right. \end{array} \right.$$

- 知识点回顾 

- LR分析法——LALR(1)分析

- 寻找具有**相同核心**的LR (1) 项集，并将这些项集合并为一个项集。
所谓项集的核心就是其第一分量的集合
 - 如果分析表中**没有**语法分析动作**冲突**，给定的文法就称为**LALR (1) 文法**，就可以根据该分析表进行语法分析
 - LALR(1)形式上与LR(1)相同、大小上与LR(0)/SLR相当
 - 分析能力介于SLR和LR(1)二者之间
- $$LR(0) < SLR < LALR(1) < LR(1)$$
- 合并后的展望符集合仍为FOLLOW集的子集

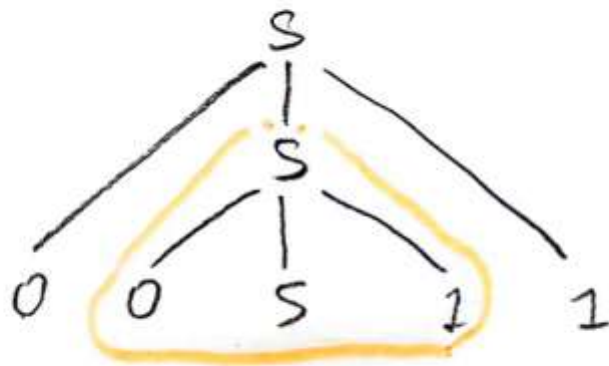
• 习题

• T 6.1 (2)

➤ 对于文法 $S \rightarrow 0S1 \mid 01$ ，指出下面各个最右句型的句柄：

➤ (2) 00S11

T 6.1 (2)



句柄 0S1

• 习题

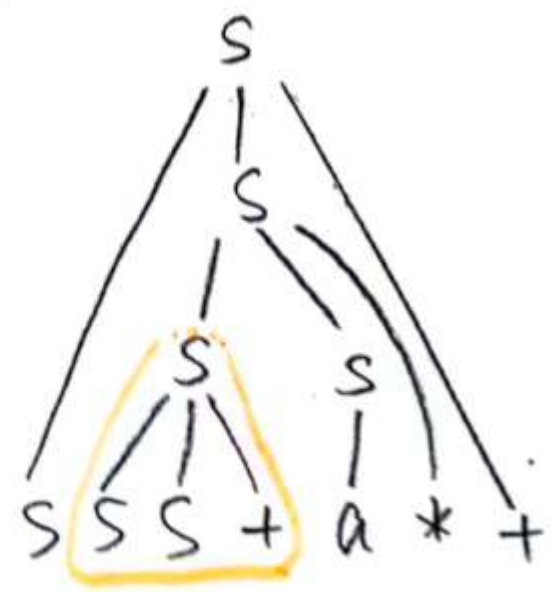
• T 6.2 (1)

➤ 对于文法 $S \rightarrow S S + \mid S S * \mid a$, 指出下面各个最右句型的句柄:

➤ (1) $SSS+a*+$

短语、直接短语、句柄

T6.2(1)





• 习题

• SPOC – 第7讲

1 **单选** (1分) 一个LR(1)文法合并同心集后若不是LALR(1)文法()。

- A. 则可能存在移进/归约冲突
- B. 以上说法都不对
- C. 则可能存在移进/归约冲突和归约/归约冲突
- D. 则可能存在归约/归约冲突

2 **单选** (1分) 若状态k含有项目“ $A \rightarrow a \cdot$ ”，且仅当输入符号 $a \in FOLLOW(A)$ 时，才用规则“ $A \rightarrow a$ ”归约的语法分析方法是()。

- A. LALR分析法
- B. LR(0)分析法
- C. LR(1)分析法
- D. SLR(1)分析法

3 **单选** (1分) LR(1)文法都是()。

- A. 可能有二义性但无左递归
- B. 无二义性但可能是左递归
- C. 无二义性且无左递归
- D. 可以既有二义性又有左递归

4 **单选** (1分) 同心集合并可能会产生新的()冲突。

- A. 二义
- B. 移进/移进
- C. 移进/归约
- D. 归约/归约



- 习题 

- T 7.1 (1)

➤ 对于下列各（增广）文法：

① 构造SLR项集和它们的GOTO函数。

② 指出你的项集中有没有动作冲突。

③ 如果存在SLR语法分析表，构造出这个语法分析表。

➤ (1) $S \rightarrow + S S \mid * S S \mid a$ 输入样例：+ * a a a



- 习题 

- T 7.1 (1)

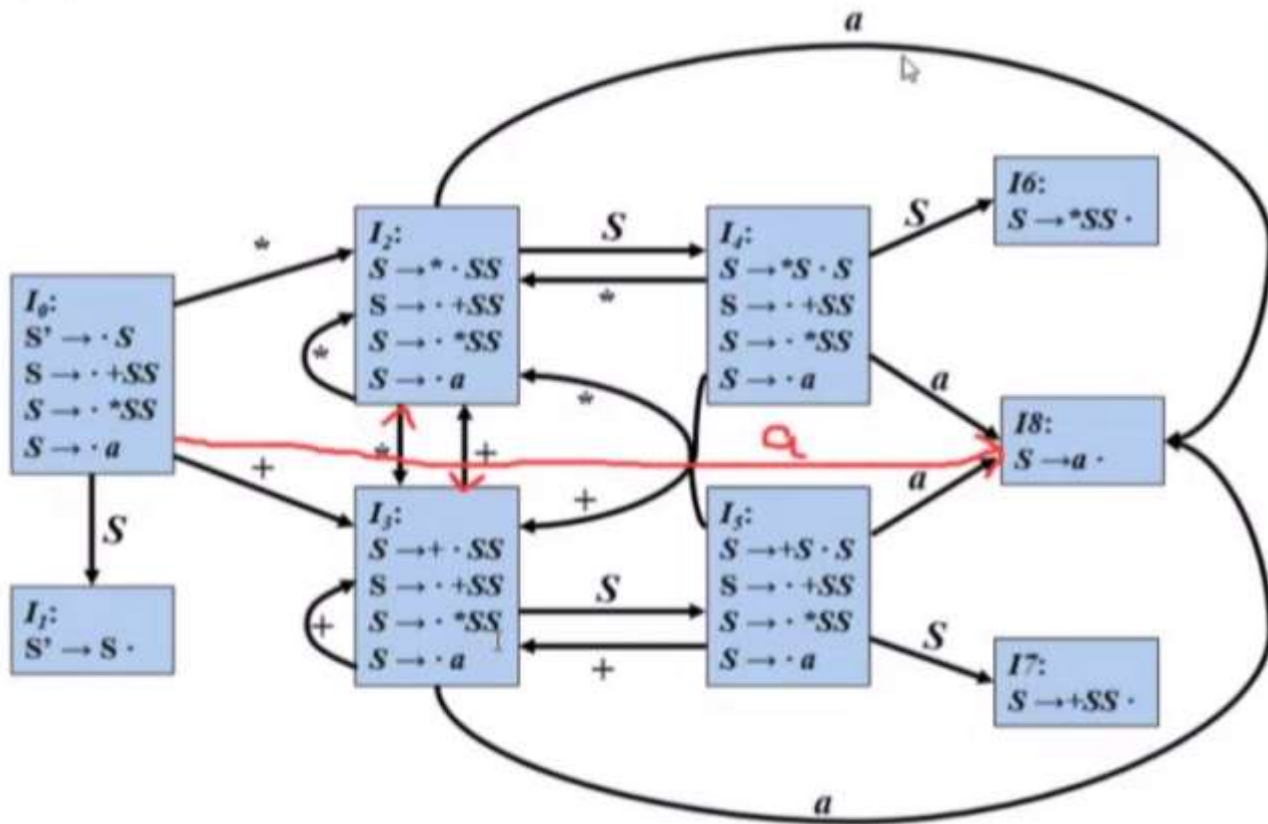
➤ 对于下列各（增广）文法：

① 构造SLR项集和它们的GOTO函数。

• 习题 


• T 7.1 (1)

➤ (1) $S \rightarrow +SS \mid *SS \mid a$ 输入样例: $+*aaa$



| 状态 | ACTION | | | | GOTO |
|----|--------|----|----|-----|------|
| | + | * | a | \$ | |
| 0 | s3 | s2 | s8 | | 1 |
| 1 | | | | acc | |
| 2 | s3 | s2 | s8 | | 4 |
| 3 | s3 | s2 | s8 | | 5 |
| 4 | s3 | s2 | s8 | | 6 |
| 5 | s3 | s2 | s8 | | 7 |
| 6 | r2 | r2 | r2 | r2 | |
| 7 | r1 | r1 | r1 | r1 | |
| 8 | r3 | r3 | r3 | r3 | |



• 习题 

• T 7.2

➤ 说明下面的文法

$$S \rightarrow A a A b \mid B b B a$$
$$A \rightarrow \varepsilon$$
$$B \rightarrow \varepsilon$$

是LL(1)的，但不是SLR(1)的



• 习题 

• T 7.5 (3)

➤ 对于下列各（增广）文法，构造

① 规范LR项集族

② LALR项集族

➤ (3) $S \rightarrow S(S)S \mid \varepsilon$



• 习题



• T 7.6

➤ 说明下面的文法

$$S \rightarrow A a \mid b A c \mid d c \mid b d a$$
$$A \rightarrow d$$

是LALR(1)的，但不是SLR(1)的。



- 习题 

- T 7.7

➤ 说明下面的文法

$$S \rightarrow A a \mid b A c \mid B c \mid b B a$$

$$A \rightarrow d$$

$$B \rightarrow d$$

是LR(1)的，但不是LALR(1)的。



• 习题

• SPOC – 第7讲

5 单选 (1分) 就文法的描述能力来说, 有()。

- A. $LR(1) \subset LR(0)$
- B. $SLR(1) \subset LR(1)$
- C. 无二义文法 $\subset LR(1)$
- D. $SLR(1) \subset LR(0)$

6 单选 (1分) 编译程序的语法分析器必须输出的信息是()。

- A. 语法分析过程
- B. 语法错误信息
- C. 语法规则信息
- D. 语句序列

写在最后：

天之道，损有余而补不足。
人之道则不然，损不足以奉有余。

——《道德经》第七十七章



写在最后：



希望各位同窗复习顺利、
预祝考试顺利！

计算学部-物联网工程-尹博
2024年5月