

第三讲书面作业包括两部分。第一部分为 Lecture03.pdf 中课后作业题目中的

5(2), 5(3), 6(2), 7(2), 7(3), 以及 11 。第二部分为以下题目:

A1 根据你的判断, 针对以下文法是否可以设计一个自顶向下预测分析过程? 如果可以, 需要向前察看多少个输入符号?

(1) 文法 $G_1[S]$:

$$\begin{aligned} S &\rightarrow A \mid B \\ A &\rightarrow aAb \mid c \\ B &\rightarrow aBbb \mid d \end{aligned}$$

(2) 文法 $G_2[S]$:

$$\begin{aligned} S &\rightarrow \varepsilon \mid abA \\ A &\rightarrow Saa \mid b \end{aligned}$$

A2 给定某类表达式文法 $G[E]$:

$$\begin{aligned} E &\rightarrow +ER \mid -ER \mid \underline{positive} R \\ R &\rightarrow *ER \mid \varepsilon \end{aligned}$$

其中, $+$ 和 $-$ 分别代表一元正和一元负运算, $*$ 代表普通的二元乘法运算, positive 为代表正整数 (非0) 的单词。

(1) 针对文法 $G[E]$, 下表给出各产生式右部文法符号串的 *First* 集合, 各产生式左部非终结符的 *Follow* 集合, 以及各产生式的预测集合 *PS*。试填充其中空白表项 (共3处) 的内容:

$G[E]$ 的规则 r	$First(rhs(r))$	$Follow(lhs(r))$	$PS(r)$
$E \rightarrow +ER$	+		+
$E \rightarrow -ER$	-	此处不填	-
$E \rightarrow \underline{positive} R$	<u>positive</u>	此处不填	<u>positive</u>
$R \rightarrow *ER$	*		*
$R \rightarrow \varepsilon$	ε	此处不填	

表中, $rhs(r)$ 为产生式 r 右部的文法符号串, $lhs(r)$ 为产生式 r 左部的非终结符。

(2) $G[E]$ 不是 LL(1) 文法, 试解释为什么?

(3) 虽然 $G[E]$ 不是 LL(1) 文法, 但可以采用一种强制措施, 使得常规的 LL(1) 分析算法仍然可用。针对含5个单词的输入串 $+ - 20 * 18$, 以下基于这一措施以及上述各产生式的预测集合 (或预测分析表) 的一个

表驱动 LL(1) 分析过程：

步骤	下推栈	余留符号串	下一步动作
1	# E	+ - 20 * <u>18</u> #	应用产生式 $E \rightarrow + E R$
2	# $R E$ +	+ - <u>20</u> * <u>18</u> #	匹配栈顶和当前输入符号
3	# $R E$	- <u>20</u> * <u>18</u> #	应用产生式 $E \rightarrow - E R$
4	# $R R E$ -	- <u>20</u> * <u>18</u> #	匹配栈顶和当前输入符号
5	# $R R E$	<u>20</u> * <u>18</u> #	应用产生式 $E \rightarrow \underline{positive} R$
6	# $R R R \underline{positive}$	<u>20</u> * <u>18</u> #	匹配栈顶和当前输入符号
7	# $R R R$	* <u>18</u> #	
8			
9			
10			
11			
12	# $R R R$	#	应用产生式 $R \rightarrow \epsilon$
13	# $R R$	#	应用产生式 $R \rightarrow \epsilon$
14	# R	#	应用产生式 $R \rightarrow \epsilon$
15	#	#	结束

试填写上述分析过程中第7步时使用的产生式，以及第 8~11 步的分析过程，共计13处空白；并指出采用了什么样的强制措施。

A3 给定命题表达式文法 $G[S]$ ：

$$S \rightarrow P$$

$$P \rightarrow \wedge P P \mid \vee P P \mid \neg P \mid \underline{id}$$

其中， \wedge 、 \vee 、 \neg 分别代表命题逻辑与、或、非等运算符单词， \underline{id} 代表标识符单词。

容易得出： $G[S]$ 是 $LL(1)$ 文法。基于 $G[S]$ 的预测分析表和一个分析栈，课程中介绍了一种表驱动的 $LL(1)$ 分析过程。假设有输入符号串： $\vee \vee a \wedge b c \vee \neg a \wedge c b \#$ 。试问，在分析过程中，分析栈中最多会出现几个 S ？几个 P ？若因误操作使输入串多了一个符号，变为 $\vee \vee a \wedge b c c \vee \neg a \wedge c b$ ，当分析过程中发生错误时，关于报错信息，你认为最不可能的选择是（4选1）：
（1）缺运算数；（2）多运算数；（3）缺运算符；（4）多运算符。如果想要从该出错位置恢复分析，可以进行什么操作？

A4 分析表达式文法 (Parsing Expression Grammar, PEG) 是一个四元组 (N, Σ, P, S) , 其中

- N 为非终结符 (nonterminal) 集合;
- Σ 为终结符 (terminal) 集合;
- P 为产生式 (parsing rules) 集合;
- S 为开始符号 (start symbol)。

各产生式形如 $A \rightarrow e$, 其中 A 为非终结符, e 为分析表达式 (parsing expression)。每个分析表达式都会用来匹配输入串的一个前缀。

本题考虑一种简单的 PEG, 其分析表达式定义如下:

$$e ::= t \mid n \mid e_1 e_2 \mid e_1 > e_2 \mid e^* \mid (e)$$

其中 $t \in \Sigma$ 只能匹配终结符 t 自身, $n \in N$ 只能匹配非终结符 n 自身。其他复合的表达式及分析过程为 (分析失败则表明输入串不能被该文法识别):

- 序列 “ $e_1 e_2$ ” 表示: 先用输入串匹配 e_1 , 若成功再用剩下的串匹配 e_2 ;
- 有序选择 “ $e_1 > e_2$ ” 表示: 先尝试用输入串匹配 e_1 , 若成功则分析结束 (忽略 e_2); 若失败, 重新用输入串匹配 e_2 ;
- 星闭包 “ e^* ” 表示: 用输入串匹配 e 零次或多次, 直至匹配失败 (即消耗掉输入串中尽可能多的符号);
- 特别地, 括号用于显式指定优先级。

可以发现, PEG 由于其分析过程允许回溯, 因而无需像传统的 LL 分析那样要向后查看符号。

根据以上设定, 回答下列问题。

(1) 什么样的输入串可以匹配 (指完全匹配, 即匹配完以后没有余留的串) 分析表达式 “ $a^* a b$ ”, 其中 a 和 b 为终结符?

(2) 类似于 “ e^* ”, 我们可以扩展出 “ e^+ ” —— 它表示用输入串匹配 e 一次或多次, 直至匹配失败。请利用以上给出的这些分析表达式, 写出 “ e^+ ” 的定义。

(3) 利用 PEG 可以消除悬挂 else 的二义性。请简要说明如何消除?

(4) 考查下列 CFG (其中 x 为终结符):

$$S \rightarrow x S x \mid x$$

它对应的语言记作 L 。问:

(a) 是否可以直接对上述文法采用递归下降方法来分析? 请说明理由。

(b) 是否存在某个能做递归下降分析的 PEG, 其对应的语言也为 L ? 若存在, 请给出此 PEG; 否则, 请说明理由。