

编译原理 - 作业(3): 语法分析 LR、语义分析

截止时间: 2023/5/9 (周二) 课前, 14:19:59

提交方式: <https://easyhpc.net/course/164>

1. 对于如下文法 G

- (1) $E \rightarrow XY$
 (2) $X \rightarrow cXa \mid b$
 (3) $Y \rightarrow d$

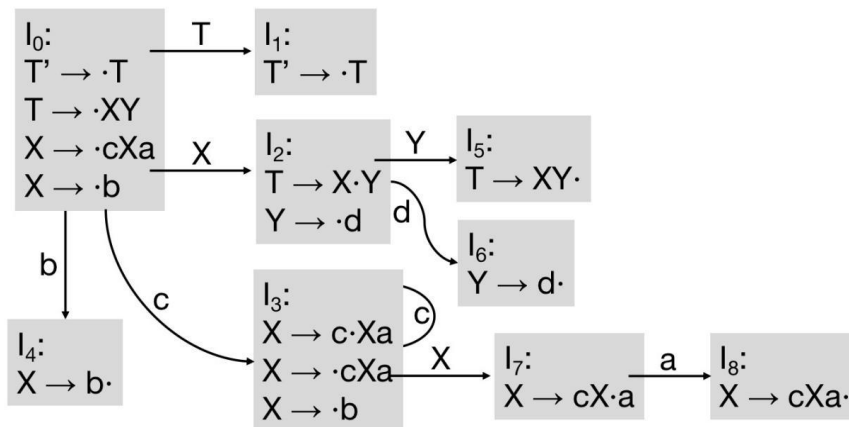
(1) 求文法 G 的增广文法 G' [2 分]

【参考答案】

- (0) $E' \rightarrow E$
 (1) $E \rightarrow XY$
 (2) $X \rightarrow cXa \mid b$
 (3) $Y \rightarrow d$

(2) 对文法 G' 构建 LR(0) 解析的有穷自动机 (FA), 包括状态和转换。提示: 项目集闭包 (closure of item sets) [8 分]。

【参考答案】



【评分标准】 每个状态转换 (包括节点和边) 错误扣 1 分, 扣完为止。

(3) 构建 LR(0) 解析表。注: 如有需要, 请自行添加更多行。[7 分]

State	ACTION					GOTO		
	a	b	c	d	\$	E	X	Y

【参考答案】

State	ACTION					GOTO		
	a	b	c	d	\$	E	X	Y

0		s4	s3			1	2	
1					acc			
2				s6				5
3		s4	s3				7	
4	r3	r3	r3	r3	r3			
5	r1	r1	r1	r1	r1			
6	r4	r4	r4	r4	r4			
7	s8							
8	r2	r2	r2	r2	r2			

【评分标准】每个状态在ACTION表或GOTO表中的行出现错误扣0.5分，扣完为止。

(4) 列出解析输入串 **cbad** 的过程，包括每一步输入串和解析栈变化及采取的具体动作。
注：如有需要，请自行添加更多行。[8 分]

Stack	Input	Action

Stack	Input	Action
0 \$	cbad\$	[0, c]: shift, to state 3
0 3 \$ c	bad\$	[3, d]: shift, to state 4
0 3 4 \$ c b	ad\$	[4, a]: reduce, X -> b
0 3 \$ c A	ad\$	[GOTO: [3, X] => 7
0 3 7 \$ c A	ad\$	[7, a]: shift, to state 8
0 3 7 8 \$ c A a	d\$	8, b]: reduce, X-> cXa
0 \$ A	d\$	GOTO: [0, X] => 2
0 2 \$ A	d\$	[2, b]: shift, to state 6
0 2 6 \$ A d	\$	[6, \$]: reduce, Y -> d
0 2 \$ A B	\$	GOTO: [2, Y] => 5
0 2 5 \$ A B	\$	[5, \$]: reduce, S -> AB
0 \$ S	\$	GOTO: [0, S] => 1
0 1 \$ S	\$	acc

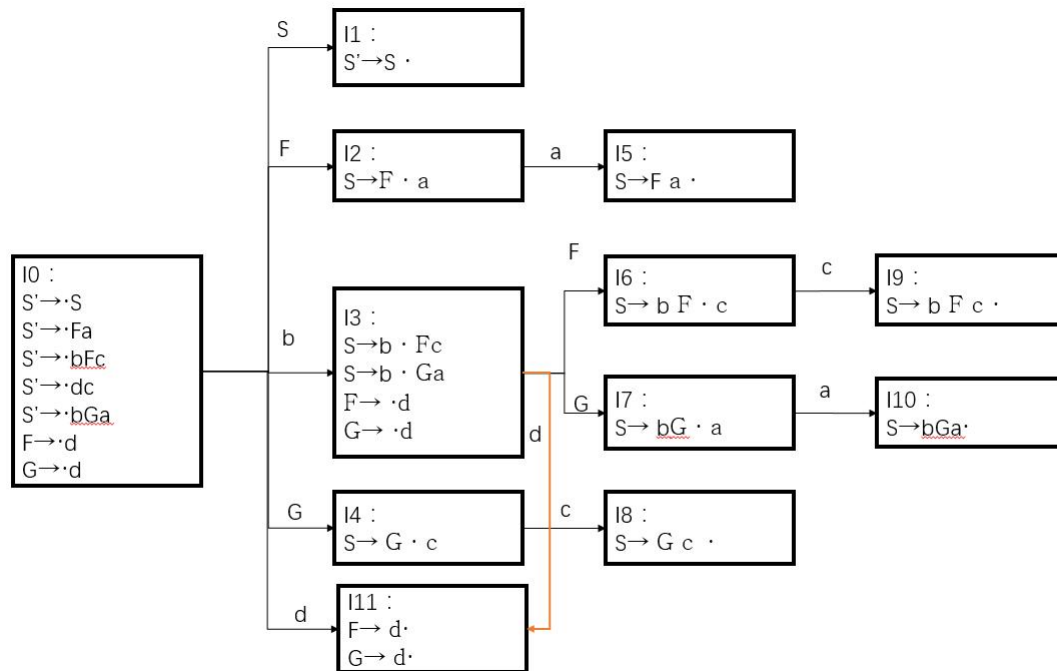
【评分标准】栈状态可不列出，关键动作为 shift 和 reduce。每个 shift/reduce 错误扣 1 分，扣完为止。

2. 对于如下文法 G

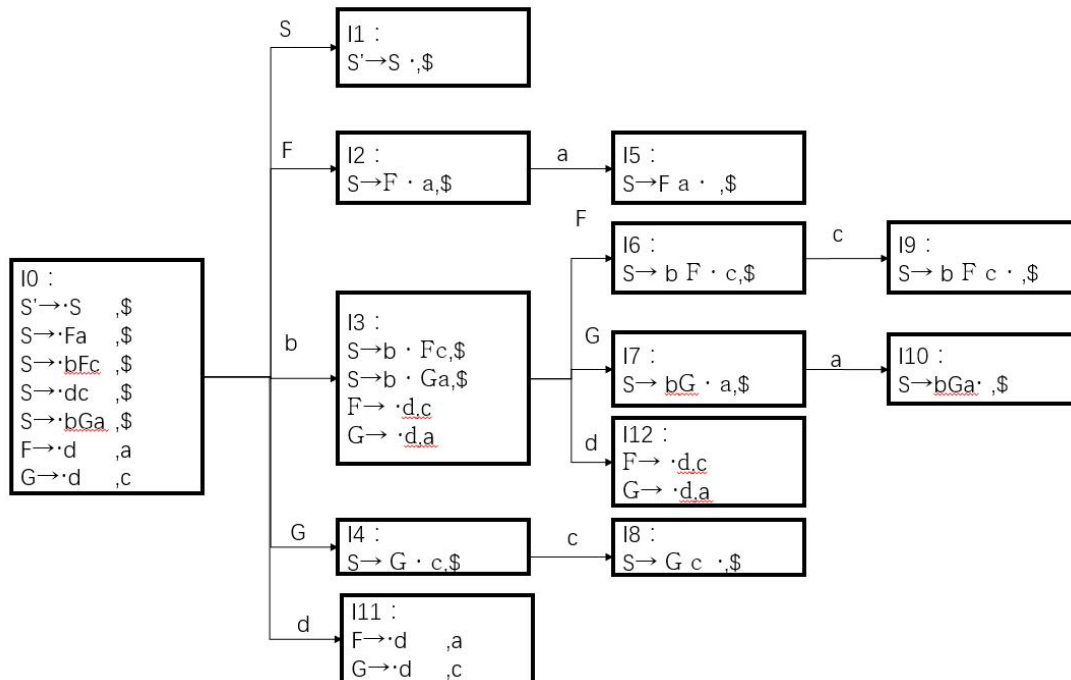
$$\begin{aligned}
 S &\rightarrow F a \mid b F c \mid G c \mid b G a \\
 F &\rightarrow d \\
 G &\rightarrow d
 \end{aligned}$$

- (1) 基于 LR(0)和 LR(1)项目构建 DFA;
- (2) 该文法是否为 SLR(1)文法? 请详细说明理由;
- (3) 该文法是否为 LR(1)文法? 请详细说明理由;
- (4) 该文法是否为 LALR(1)文法? 请详细说明理由。

LR(0):



LR(1):



基于 LR(0) 项目的识别活前缀的 DFA 中：在 I4 第一个项目要求面对 c 移进到 8，第二个要求 r5 归约，I7 第一个项目要求面对 a 移进到 10，第二个要求 r5 归约。所以存在冲突，不是 SLR(1) 的！

基于 LR(1) 项目 I11, I12 需要合并，结果和 SLR(1) 一样。

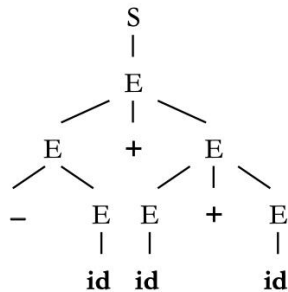
基于 LR(1) 项目的识别活前缀的 DFA 中：没有同心项目集可以合并，是 LR(1) 的。

3. 对于如下文法 G

$$S \rightarrow E$$

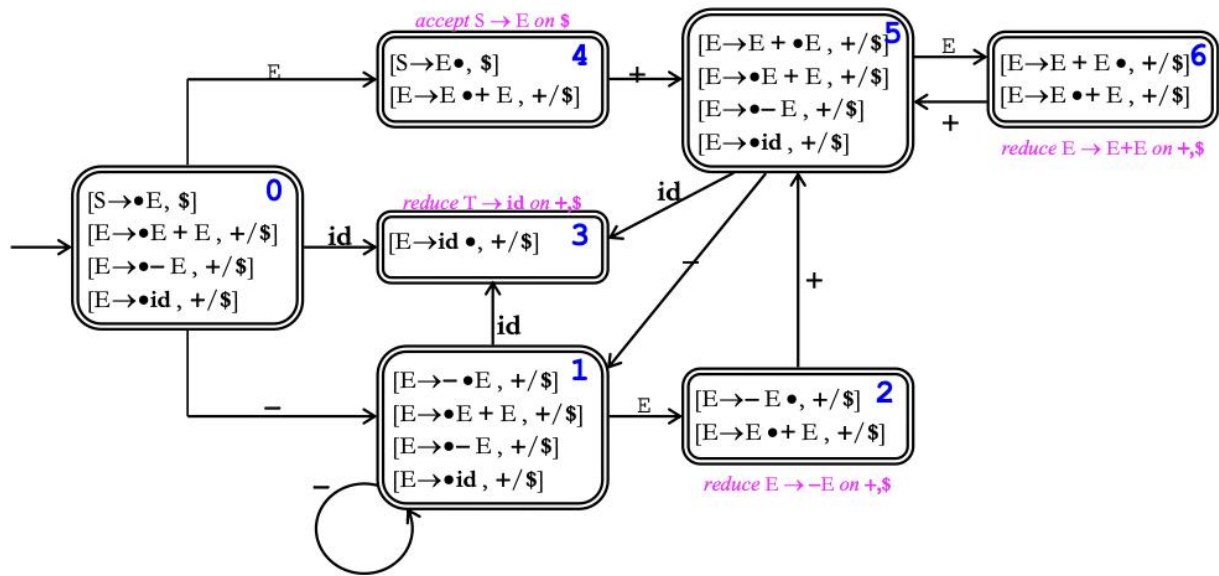
$$E \rightarrow E + E \mid - E \mid id$$

- (1) 构建该文法 LR(1) 解析的 DFA；
- (2) 对每一个含有冲突的状态，列出状态的编号、引起冲突的输入符号、以及冲突的类型；
- (3) 画出句子 $id + - id + id$ 的所有分析树，并判断文法 G 是否具有二义性；
- (4) 假设我们想让句子 $- id + id + id$ 仅有如下一棵分析树是合法的(以下将此称为性质 P)。用自然语言描述: 为保证性质 P，相关算符的优先级(Precedence)和结合性质(Associativity) 的规则如何？



(5) 为保证性质 P，根据上述 DFA 构造的 LR(1)分析表中的冲突应如何解析，即在“移进- 归约”冲突中选择移进还是归约、在“归约-归约”冲突中选择以哪一个产生式归约。

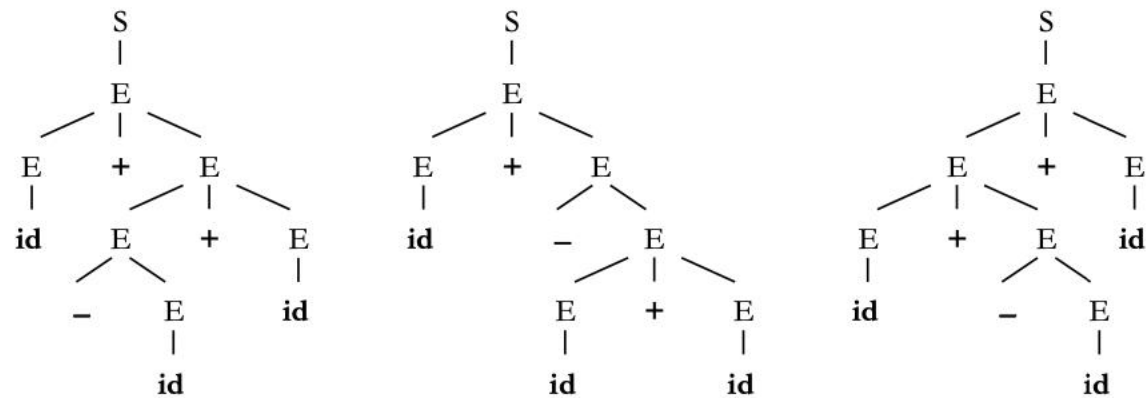
(1) 完整 DFA 如下：



(2) 存在如下冲突

状态	输入符号	冲突类型
2	+	“移入-归约”冲突
6	+	“移入-归约”冲突

(3) 句子 $id + - id + id$ 有 3 棵不同的分析树:



该文法显然是一个二义文法，因为句子 $\text{id} + - \text{id} + \text{id}$ 有多个合法的分析树。

- (4) 为保证性质 **P**，我们需规定一元算符“-”的优先级要高于二元算符“+”，并且算符“+”是右结合的。
- (5) 保证性质 **P**，按以下方式解析冲突：

状态	输入符号	冲突类型	为解决冲突选择
2	+	“移入-归约”冲突	归约
6	+	“移入-归约”冲突	移入

4. 对于如下文法 **G**

$$F \rightarrow N.N$$
$$N \rightarrow ND \mid D$$
$$D \rightarrow 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$$

- (1) 用自然语言描述上述文法定义的语言；

该文法定义了表示实数的串，一个实数必须由整数和小数两部分组成，整数部分允许以 **0** 开头。

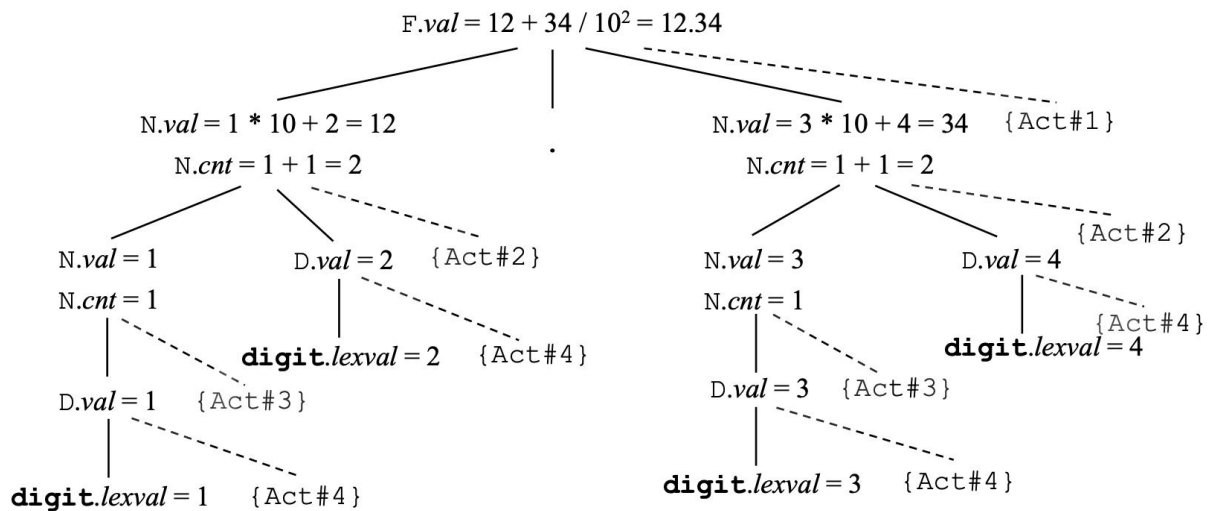
- (2) 给出该文法的一个翻译模式（SDT），其语义为计算一个十进制输入串的实数值（例如，对于输入串 $1\ 2\ 3.4\ 5\ 6$ ，其实数值的计算结果为 123.456 ）；

正确答案有多种。其中一种较为简单、直观的翻译模式如下：

$$\begin{aligned}
 F &\rightarrow N_1 . N_2 & \{ & F.val = N_1.val + N_2.val / 10^{N_2.cnt} \} \\
 N &\rightarrow N_1 D & \{ & N.val = N_1.val * 10 + D.val; \\
 & & & N.cnt = N_1.cnt + 1 \} \\
 &| D & \{ & N.val = D.val; \\
 & & & N.cnt = 1 \} \\
 D &\rightarrow \text{digit} & \{ & D.val = \text{digit.lexval} \}
 \end{aligned}$$

(3) 根据 (2) 中给出的翻译模式, 画出输入串 12.34 的带属性、带动作的标注分析树。

带注释、带动作的分析树如下(其中, **Act#n** 表示上述翻译模式中的第 **n** 个语义动作):



[[评分标准]]

- 1、(2分)描述错误扣2分。
- 2、(8分)未按要求写成翻译模式, 即便写出了一个正确的 SDD, 扣5分。未能分别计算整数部分和 小数部分的值 and 串长, 扣5分; 整数部分和小数部分的值 and 串长计算有误, 扣3分; 边界小错误, 扣1分。
- 3、(6分)分析树的结构有错, 至少扣4分; 未注明属性的名字与值, 至少扣4分; 未画出语义动作, 至少扣3分。内部结点的属性值计算错误, 每个扣1分; 动作遗漏或挂错位置, 每个扣1分。