

## 2018 级弘毅班《编译原理》第三次练习答案

注意：2021 年 5 月 24 日交作业！

一、设有 C 语言说明语句文法  $D$  如下所示：

$$\begin{aligned} D &\rightarrow T L \\ T &\rightarrow \text{int} \mid \text{char} \\ L &\rightarrow *L \mid L[] \mid L() \mid (L) \mid \text{id} \end{aligned}$$

- (1) 由于该文法是二义文法，因此其 LR 分析表一定有移进/归约或归约/归约冲突，试用 JFLAP 指出那些状态集有怎样的冲突；

解：在状态  $I_{10} = \{L \rightarrow *L\bullet, L \rightarrow L\bullet(), L \rightarrow L\bullet[]\}$  面对 ( 和 [ 有移进/归约冲突, 选择移进表示后缀运算优先于前缀运算.

- (2) 请正确地对冲突项目选择移进/归约构造其 SLR 分析表，使得按照该分析表分析正表达式时，其运算的优先级别和结合次序与 C 语言说明语句中规定的优先级别和结合次序一致 (仅列出冲突项集合对应的 Action 表)；

|    | action |      |    |     |    |     |   |    |
|----|--------|------|----|-----|----|-----|---|----|
| 状态 | int    | char | id | (   | )  | [   | ] | \$ |
| 10 |        |      |    | s11 | r4 | s14 |   | r4 |

其中标号为 4 的产生式为：  $L \rightarrow *L$ . 完整的分析表截图如下所示：

|    | (   | )   | *  | [   | ]   | a  | c  | i  | \$  | D | L  | T |
|----|-----|-----|----|-----|-----|----|----|----|-----|---|----|---|
| 0  |     |     |    |     |     |    | s3 | s4 |     | 1 |    | 2 |
| 1  |     |     |    |     |     |    |    |    | acc |   |    |   |
| 2  | s5  |     | s6 |     |     | s8 |    |    |     |   | 7  |   |
| 3  | r3  |     | r3 |     |     | r3 |    |    |     |   |    |   |
| 4  | r2  |     | r2 |     |     | r2 |    |    |     |   |    |   |
| 5  | s5  |     | s6 |     |     | s8 |    |    |     |   | 9  |   |
| 6  | s5  |     | s6 |     |     | s8 |    |    |     |   | 10 |   |
| 7  | s11 |     |    | s12 |     |    |    |    | r1  |   |    |   |
| 8  | r8  | r8  |    | r8  |     |    |    |    | r8  |   |    |   |
| 9  | s11 | s13 |    | s12 |     |    |    |    |     |   |    |   |
| 10 | s11 | r4  |    | s12 |     |    |    |    | r4  |   |    |   |
| 11 |     | s14 |    |     |     |    |    |    |     |   |    |   |
| 12 |     |     |    |     | s15 |    |    |    |     |   |    |   |
| 13 | r7  | r7  |    | r7  |     |    |    |    | r7  |   |    |   |
| 14 | r6  | r6  |    | r6  |     |    |    |    | r6  |   |    |   |
| 15 | r5  | r5  |    | r5  |     |    |    |    | r5  |   |    |   |

其中  $a$  表示  $\text{id}$ ,  $c$  表示  $\text{char}$ ,  $i$  表示  $\text{int}$ .

- (3) 利用你的分析表分析输入正规表达式 “ $\text{int} * \text{id} ( ) [ ]$ ” 的分析过程.  
解：

| 剩余串             | 分析栈              | 分析动作                              |
|-----------------|------------------|-----------------------------------|
| int * id() []\$ | 0                | shift                             |
| * id() []\$     | 0int4            | reduce $T \rightarrow \text{int}$ |
| * id() []\$     | 0T2              | shift                             |
| id() []\$       | 0T2 * 6          | shift                             |
| () []\$         | 0T2 * 6id8       | reduce $L \rightarrow \text{id}$  |
| () []\$         | 0T2 * 6L10       | shift                             |
| ) []\$          | 0T2 * 6L10(11    | shift                             |
| []\$            | 0T2 * 6L10(11)14 | reduce $L \rightarrow L()$        |
| []\$            | 0T2 * 6L10       | shift                             |
| ]\$             | 0T2 * 6L10[12    | shift                             |
| \$              | 0T2 * 6L10[12]15 | reduce $L \rightarrow L[]$        |
| \$              | 0T2 * 6L10       | reduce $L \rightarrow *L$         |
| \$              | 0T2L7            | reduce $D \rightarrow TL$         |
| \$              | 0D1              | 分析成功                              |

二、设有文法  $G$  定义如下：

$$\begin{aligned}
 S &\rightarrow AB \\
 A &\rightarrow aAb \mid aA \mid a \\
 B &\rightarrow bBa \mid bB \mid \varepsilon
 \end{aligned}$$

- (1) 试描述文法  $G$  所生成的语言；  
解：文法生成的语言为： $\{a^m b^{n+p} a^n \mid m \geq 1 \wedge n, p \in \mathbb{N}\}$ 。
- (2) 试不用构造分析表直接说明该文法不是 SLR 文法；  
解： $A$  成分有二义性，即：

$$\begin{aligned}
 A &\Rightarrow aAb \Rightarrow_{lm} aaAb \Rightarrow_{lm} aaab \\
 A &\Rightarrow aA \Rightarrow_{lm} aaAb \Rightarrow_{lm} aaab
 \end{aligned}$$

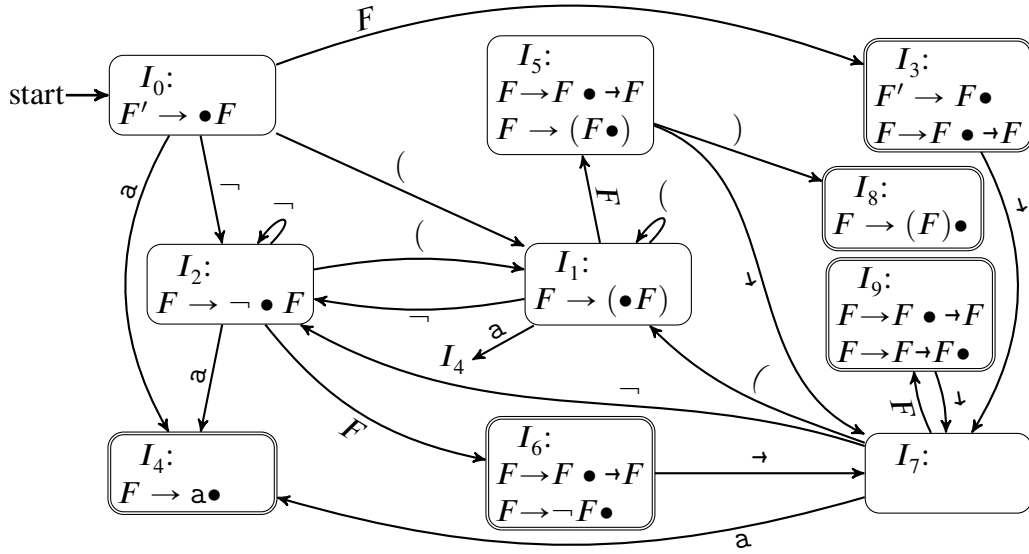
- (3) 试修改文法  $G$  使之成为 SLR 文法。  
解：

$$\begin{aligned}
 S &\rightarrow aA \\
 A &\rightarrow aA \mid bB \mid \varepsilon \\
 B &\rightarrow bB \mid C \\
 C &\rightarrow bCa \mid a
 \end{aligned}$$

三、设  $G(F)$  的拓广文法  $G(F')$  如下所示： (2020 考题)

$$\begin{aligned}
 F' &\rightarrow F & (0) \\
 F &\rightarrow F \rightarrow F & (1) \\
 &| \neg F & (2) \\
 &| (F) & (3) \\
 &| a & (4)
 \end{aligned}$$

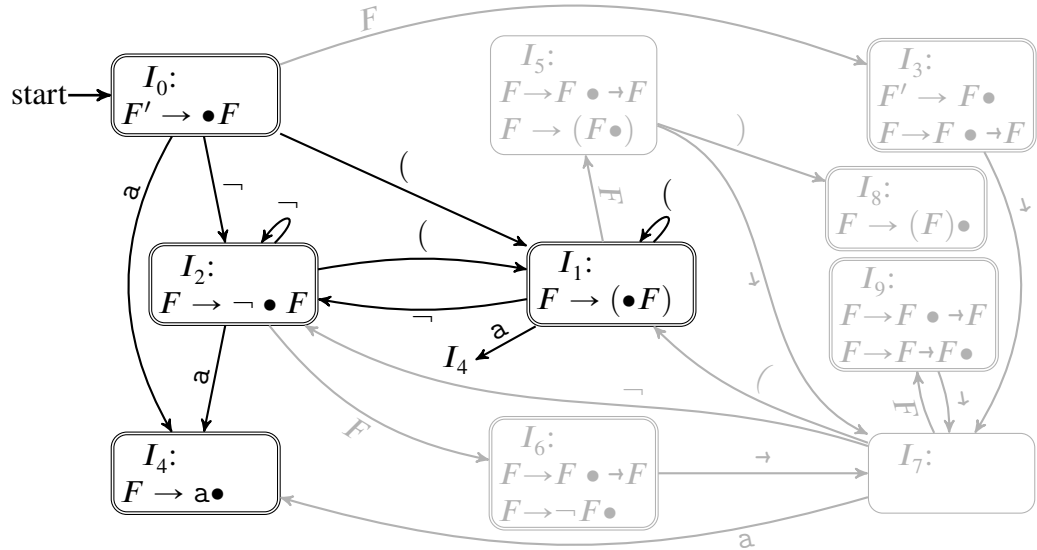
文法  $G(F')$  的识别活前缀 LR(0) 项目自动机  $M$  如下图所示 (注意每个状态仅列出了核心项目, 状态  $I_7$  除外):



- (1) 试求状态  $I_7$  所对应的 LR(0) 项目集;  
状态  $I_7$  的 LR(0) 项目集为

$$\begin{aligned} & \overline{\{F \rightarrow F \rightarrow \bullet F\}} \\ & = \{F \rightarrow F \rightarrow \bullet F, F \rightarrow \bullet F \rightarrow F, F \rightarrow \bullet \neg F, F \rightarrow \bullet (F), F \rightarrow \bullet a\}. \end{aligned}$$

- (2) 试求仅由终结符号组成的活前缀对应的正则表达式;  
仅由终结符号组成的活前缀对应的前缀 DFA 为:



其对应的正则表达式为:  $(\neg | ()^* a ?$ .

- (3) 试构造该文法的 SLR 分析表, 并对分析表中的移进/归约和归约/归约冲突选择正确的移进或归约动作, 使得文法  $G(F)$  的所有语句能被正确地分析

且运算的优先级与结合次序与题三所规定的一致;  
 $\text{Follow}(F) = \{ \rightarrow, ), \$ \}$ . 状态  $I_6$  和状态  $I_9$  面对 ‘ $\rightarrow$ ’ 有移进/归约冲突. 分析表如下所示:

| 状态 | action |        |               |    |    |     | goto |
|----|--------|--------|---------------|----|----|-----|------|
|    | a      | $\neg$ | $\rightarrow$ | (  | )  | \$  | $F$  |
| 0  | s4     | s2     |               | s1 |    |     | 3    |
| 1  | s4     | s2     |               | s1 |    |     | 5    |
| 2  | s4     | s2     |               | s1 |    |     | 6    |
| 3  |        |        | s7            |    |    | acc |      |
| 4  |        |        | r4            |    | r4 | r4  |      |
| 5  |        |        | s7            |    | s8 |     |      |
| 6  |        |        | <b>r2</b>     |    | r2 | r2  |      |
| 7  | s4     | s2     |               | s1 |    |     | 9    |
| 8  |        |        | r3            |    | r3 | r3  |      |
| 9  |        |        | <b>s7</b>     |    | r1 | r1  |      |

(4) 试利用你的分析表写出语句 “ $\neg a \rightarrow a$ ” 的分析过程.  
 语句 “ $\neg a \rightarrow a$ ” 的分析过程如下所示:

| 剩余串                       | 分析栈                   | 分析动作                                   |
|---------------------------|-----------------------|--|
| $\neg a \rightarrow a \$$ | 0                     | shift                                  |
| $a \rightarrow a \$$      | 0 $\neg$ 2            | shift                                  |
| $\rightarrow a \$$        | 0 $\neg$ 2a4          | reduce $F \rightarrow a$               |
| $\rightarrow a \$$        | 0 $\neg$ 2F6          | reduce $F \rightarrow \neg F$          |
| $\rightarrow a \$$        | 0F3                   | shift                                  |
| a\$                       | 0F3 $\rightarrow$ 7   | shift                                  |
| \$                        | 0F3 $\rightarrow$ 7a4 | reduce $F \rightarrow a$               |
| \$                        | 0F3 $\rightarrow$ 7F9 | reduce $F \rightarrow F \rightarrow F$ |
| \$                        | 0F3                   | reduce 分析成功                            |

四、 现需对题四文法  $G(F')$  所生成的命题公式转换为析取 ( $\vee$ )、合取 ( $\wedge$ ) 和仅有对原子取否的逻辑等价公式, 如:

| 序号 | 原命题公式                             | 转换后的命题公式                     |
|----|-----------------------------------|------------------------------|
| 1  | $A \rightarrow B$                 | $\neg A \vee B$              |
| 2  | $\neg \neg A$                     | $A$                          |
| 3  | $\neg (A \rightarrow B)$          | $(A) \wedge (\neg B)$        |
| 4  | $A \rightarrow (B \rightarrow C)$ | $\neg A \vee \neg B \vee C$  |
| 5  | $(A \rightarrow B) \rightarrow C$ | $(A) \wedge (\neg B) \vee C$ |

为此设计继承属性  $F.is\_neg$ , 其取值为布尔量 `True` 和 `False`; 综合属性  $F.nnf$ , 其取值为  $F$  所表示的语法成分对应的转换后的命题公式 (字符串); `a.lexeme`

取值为 a 所对应的字符串.  $F.is\_neg$  的语义规则如下所示: (2020 年考题)

| 产生式                                 | 语义规则  |
|-------------------------------------|---|
| $F' \rightarrow F$                  | $F.is\_neg = \text{False}$                              |
| $F \rightarrow F_1 \rightarrow F_2$ | $F_1.is\_neg = \neg F.is\_neg; F_2.is\_neg = F.is\_neg$ |
| $F \rightarrow \neg F_1$            | $F_1.is\_neg = \neg F.is\_neg$                          |
| $F \rightarrow (F_1)$               | $F_1.is\_neg = F.is\_neg$                               |

(1) 试写出属性  $F.nnf$  的语法制导定义;

| 产生式                                 | 语义规则  |
|-------------------------------------|---|
| $F \rightarrow F_1 \rightarrow F_2$ | <b>if</b> ( $F.is\_neg == \text{True}$ ) <b>then</b><br>$F.nnf = "(" + F_1.nnf + ")" \wedge "(" + F_2.nnf + ")"$<br><b>else</b><br>$F.nnf = F_1.nnf + "\vee" + F_2.nnf$ |
| $F \rightarrow \neg F_1$            | $F.nnf = F_1.nnf$   |
| $F \rightarrow (F_1)$               | $F.nnf = F_1.nnf$   |
| $F \rightarrow a$                   | <b>if</b> ( $F.is\_neg == \text{True}$ ) <b>then</b><br>$F.nnf = "\neg" + a.lexeme$<br><b>else</b><br>$F.nnf = a.lexeme$  |

(2) 试求 “ $\neg(((A \rightarrow \neg B) \rightarrow C) \rightarrow \neg(D \rightarrow E))$ ” 转换后的命题公式.  
 $((A) \wedge (B) \vee C) \wedge (\neg D \vee E)$

五、 设二进制实数的文法如下所示:

$$\begin{aligned} S &\rightarrow L.L \\ L &\rightarrow LB \mid B \\ B &\rightarrow 0 \mid 1 \end{aligned}$$

现需要设计将二进制实数转换为十进制实数的翻译规程, 分析二进制实数与十进制实数的关系; 设  $x_1x_2 \cdots x_m.y_1y_2 \cdots y_n$  为二进制实数; 则对应的十进制实数的数值为:

$$x_1 * 2^m + x_2 * 2^{m-1} + \cdots + x_m + \frac{y_1}{2} + \frac{y_2}{2^2} + \cdots + \frac{y_n}{2^n}$$

如 101.101 的十进制数值为 5.625。为此对每个非终结符引入属性  $val$ , 则 ‘.’ 号左边的  $L$  与右边的  $L$  的属性  $val$  的计算方法不一致, 因此对  $L$  需要一个表示位置的属性  $side$ , 其取值为  $left$  或  $right$ , 左递归文法  $L \rightarrow LB$  在计算左边  $L$  时很容易找到一个递推公式计算  $val$ , 但分析右边:

$$\underbrace{\frac{y_1}{2} + \frac{y_2}{2^2} + \cdots + \frac{y_{n-1}}{2^{n-1}}}_{L.val} + \underbrace{\frac{y_n}{2^n}}_{B.val}$$

时发现需要记录  $2^n$  才能完成  $LB$  到  $L$  的  $val$  属性计算，为此对  $L$  引入属性  $divisor$ ，设  $L$  所表示的字符串长度为  $n$ ，则  $L.divisor = 2^n$ 。

- (1) 为属性  $val$ ， $side$  和  $divisor$  设计其语法制导定义；
- (2) 分别指出属性  $val$ ， $side$  和  $divisor$  否为综合属性， $L$  属性和继承属性；
- (3) 画出语句 101.101 对应的语法附注树；
- (4) 设计二进制实数转换为十进制实数的翻译规程；
- (5) 为了方便右边  $L$  的  $val$  的计算，将原文法修改为：

$$\begin{aligned} S &\rightarrow L.R \\ L &\rightarrow LB \mid B \\ R &\rightarrow BR \mid B \\ B &\rightarrow 0 \mid 1 \end{aligned}$$

后，仅需  $val$  属性即可完成二进制到十进制的转换，试为该语法设计  $val$  的语法制导定义；并画出语句 101.101 对应的语法附注树。

解：

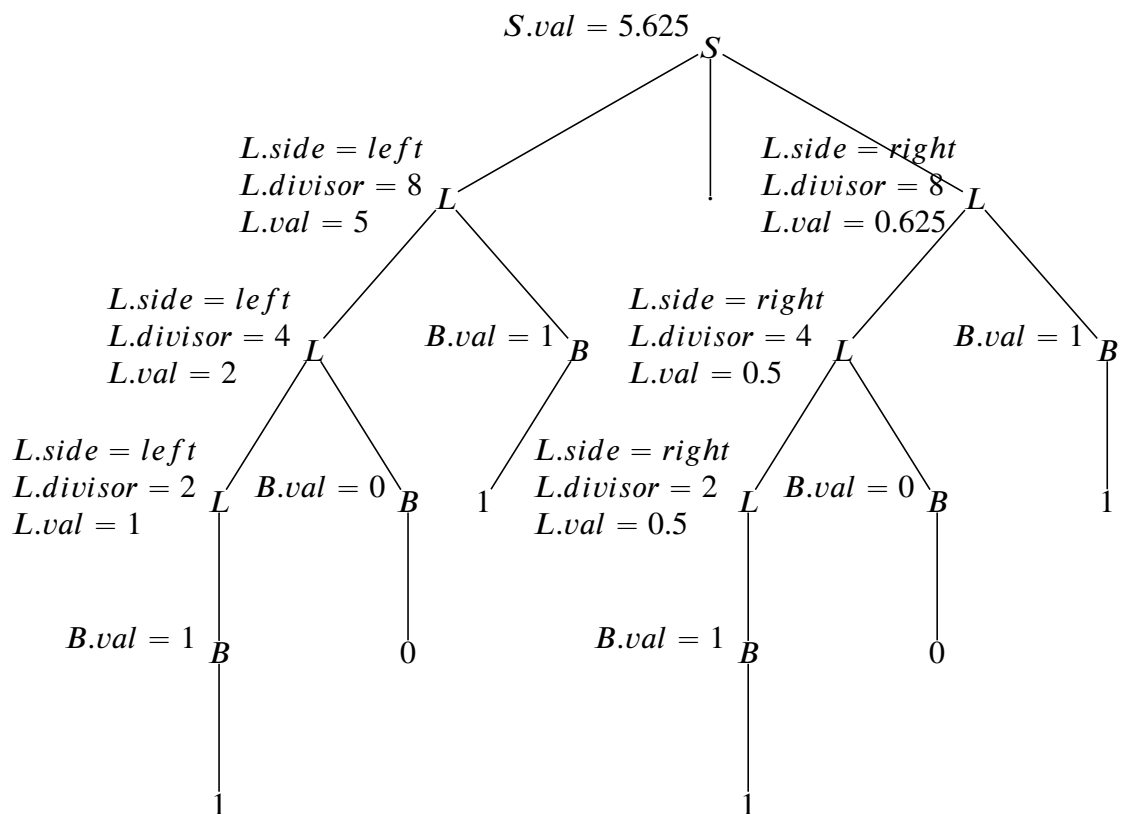
- (1) 语法制导定义如下所示：

| 编号 | 产生式                     | 语义规则   |
|----|-------------------------|--|
| 0  | $S \rightarrow L_1.L_2$ | $S.val := L_1.val + L_2.val$<br>$L_1.side = left$<br>$L_2.side = right$  |
| 1  | $L \rightarrow L_1B$    | $L.val = \text{if } L.side = left \text{ then}$<br>$2 * L_1.val + B.val$<br>$\text{else } L_1.val + B.val / (L_1.divisor * 2)$<br>$L.divisor = L_1.divisor * 2$<br>$L_1.side = L.side$ |
| 2  | $L \rightarrow B$       | $L.val = \text{if } L.side = left \text{ then}$<br>$B.val$<br>$\text{else } B.val / 2$<br>$L.divisor = 2$  |
| 3  | $B \rightarrow 0$       | $B.val = 0$  |
| 4  | $B \rightarrow 1$       | $B.val = 1$  |

- (2) 每个属性的类型如下表所示：

| 属性        | 综合 | L 属性 | 继承 |
|-----------|----|------|----|
| $val$     | n  | y    | y  |
| $side$    | n  | y    | y  |
| $divisor$ | y  | y    | n  |

- (3) 101.101 对应的语法附注树如下所示：



(4) 翻译规程如下所示:

$$\begin{aligned}
 S &\rightarrow \{L_1.side = left; L_2.side = right; \} L_1.L_2 \\
 &\quad \{S.val := L_1.val + L_2.val; \} \\
 L &\rightarrow \{L_1.side = L.side; \} L_1 B \\
 &\quad \{ L.val = \text{if } L.side = left \text{ then} \\
 &\quad \quad 2 * L_1.val + B.val \\
 &\quad \quad \text{else } L_1.val + B.val / (L_1.divisor * 2); \\
 &\quad \quad L.divisor = L_1.divisor * 2; \\
 &\quad \quad L_1.side = L.side; \} \\
 L &\rightarrow B \{L.val = \text{if } L.side = left \text{ then} \\
 &\quad \quad B.val \\
 &\quad \quad \text{else } B.val / 2; \\
 &\quad \quad L.divisor = 2; \} \\
 B &\rightarrow 0 \quad \{B.val = 0; \} \\
 B &\rightarrow 1 \quad \{B.val = 1; \}
 \end{aligned}$$

(5) 修改文法后的语法制导定义如下表所示:

| 编号 | 产生式                   | 语义规则                          |
|----|-----------------------|-------------------------------|
| 0  | $S \rightarrow L.R$   | $S.val = L.val + R.val$       |
| 1  | $L \rightarrow L_1 B$ | $L.val = 2 * L_1.val + B.val$ |
| 2  | $L \rightarrow B$     | $L.val = B.val$               |
| 3  | $R \rightarrow BR_1$  | $R.val = R_1.val/2 + B.val/2$ |
| 4  | $R \rightarrow B$     | $R.val = B.val/2$             |
| 5  | $B \rightarrow 0$     | $B.val = 0$                   |
| 6  | $B \rightarrow 1$     | $B.val = 1$                   |

(6) 二进制向十进制转换 YACC 源程序

```
%{
#include <stdio.h>
#define RIGHT 1
#define LEFT 0

typedef struct sem_type {
    double val;
    int side ;
    int divisor;
} SEM_TYPE;      /* Value stack */

void yyerror( const char *);

#define YYSTYPE      SEM_TYPE
#define YYMAXDEPTH 64
#define YYMAXERR    10
#define YYVERBOSE
%}

%%

line : line { $$ .side =LEFT; } 1 '.'
      { $$ .side = RIGHT; } 1 '\n'
      { printf("%lf\n", $3.val + $6.val); }
    |
    ;
1 : 1 b {
    if ($0.side == LEFT) $1.val = 2* $1.val + $2.val;
    else {
        $1.divisor = $1.divisor * 2 ;
        $1.val = $1.val + $2.val / $1.divisor;
    }
    $$ = $1; }
| b {
    $1.divisor = 2;
```



```

        $1.val = $0.side == LEFT? $1.val :
            $1.val/2;
        $$ = $1; }
    ;

b    : '0'  {$$.val = 0; }
    | '1'  {$$.val = 1; }
    ;

%%

int main()
{
    return yyparse();
}

void yyerror (char const *s)
{
    printf("%s\n", s);
}

int yylex (void)
{
    int c;
    if ((c = getchar ()) == EOF)
        return 0;
    return c;
}

```