

第 6 讲书面作业包括两部分。第一部分为 Lecture06.pdf 中课后作业题目中的

2, 4, 8, 9 以及 11。第二部分为以下题目：

A1 如下是以 $G[S]$ 为基础文法的一个 L 翻译模式：

$$\begin{aligned} S &\rightarrow \{ P.i := 0 \} P \{ \text{print}(P.s) \} \\ P &\rightarrow \wedge \{ P_1.i := P.i \} P_1 \{ P_2.i := P_1.s \} P_2 \{ P.s := P_2.s + 1 \} \\ P &\rightarrow \vee \{ P_1.i := P.i \} P_1 \{ P_2.i := P.i \} P_2 \{ P.s := P_1.s + P_2.s \} \\ P &\rightarrow \neg \{ P_1.i := P.i \} P_1 \{ P.s := P.i + P_1.s \} \\ P &\rightarrow \underline{id} \{ P.s := 0 \} \end{aligned}$$

试针对该 L 翻译模式构造一个自上而下的递归下降（预测）翻译程序

```
void ParseS ( )                // 主函数
{
    pi := 0;
    ps := ParseP(pi);
    print (ps);
}

int ParseP ( int i )           // 主函数
{
    switch (lookahead) {       // lookahead 为下一个输入符号
        case '^':
            MatchToken ('^');
            p1i := ①;
            p1s := ParseP (p1i);
            p2i := p1s;
            p2s := ParseP (p2i);
            ps := p2s + 1;
            break;
        case 'v':
            MatchToken ('v');
            p1i := ②;
            p1s := ParseP (p1i);
            ③;
            p2s := ParseP (p2i);
            ④;
            break;
        case '¬':
            MatchToken ('¬');
            ⑤;
            p1s := ParseP (p1i);
```

```

        _____⑥_____ ;
        break;
    case id:
        MatchToken(id);
        ps := 0;
        break;
    default:
        printf("syntax error \n")
        exit(0);
}
return ps;
}

```

其中使用了与课程中所给的 `MatchToken` 函数。

该翻译程序中 ①~⑥ 的部分未给出，试填写之。

A2. 给定用来描述某种命题逻辑公式的文法 $G[S]$:

- (1) $S \rightarrow P$
- (2) $P \rightarrow P P \wedge$
- (3) $P \rightarrow P P \vee$
- (4) $P \rightarrow P \neg$
- (5) $P \rightarrow id$

其中，终结符 \wedge 、 \vee 、 \neg 分别代表三种逻辑连接词，id 是标识符（在计算逻辑公式真值的时候，代表一个命题变元）。

如下是以 $G[S]$ 为基础文法的一个 S 翻译模式：

- (1) $S \rightarrow P$ $\{ print(P.s) \}$
- (2) $P \rightarrow P_1 P_2 \wedge$ $\{ P.s := f_1(P_1.s, P_2.s) \}$
- (3) $P \rightarrow P_1 P_2 \vee$ $\{ P.s := f_2(P_1.s, P_2.s) \}$
- (4) $P \rightarrow P_1 \neg$ $\{ P.s := f_3(P_1.s) \}$
- (5) $P \rightarrow id$ $\{ P.s := g(id) \}$

其中，`print` 为打印函数， f_1 ， f_2 ， f_3 ， g 为其他语义函数。

(a) 如果在 LR 分析过程中根据这一翻译模式进行自下而上语义计算，试写出在按每个产生式归约时语义处理的一个代码片断（设语义栈由向量 `val` 表示，归约前栈顶位置为 `top`，终结符不对应语义值，而每个非终结符的综合属性都只对应一个语义值，本题中可用 `val[i].s` 表示；不用考虑对 `top` 的维护）。

(b) 指定语义函数

$$f_1(x, y) = \text{if } (x = 1 \text{ and } y = 1) \text{ then } 1$$

$$\quad \text{else if } (x = 0 \text{ and } y = 0) \text{ then } 1$$

$$\quad \text{else } 0$$

$$f_2(x, y) = \text{if } (x = 0 \text{ and } y = 1) \text{ then } 1$$

$$\quad \text{else if } (x = 1 \text{ and } y = 0) \text{ then } 1$$

$$\quad \text{else } 0$$

$$f_3(x) = \text{if } x = 0 \text{ then } 1 \text{ else } 0$$

$$g(a) = 1 \quad g(b) = 1 \quad g(c) = 0$$

那么对于合法输入串 $a b \neg \wedge a b c \wedge \vee a \neg c b \wedge \vee \vee \vee$ ，翻译模式的语义计算结果是什么？（即 *print* 的打印结果，你无需给出计算过程）