编译原理作业 10

23.11.08

5.4.4.(1)

产生式	语义规则
S -> if (C) S_1 else S_2	L1 = new() L2 = new() S_1.next = S.next S_2.next = S.next C.true = L1 C.false = L2 S.code = C.code || "label" || L1 || S_1.code || "goto" || S.next || "label" || L2 || S_2.code

5.4.5.(1)

产生式片段	动作
S -> if (	L1 = new() L2 = new() C.true = L1 C.false = L2
C)	S_1.next = S.next
S_1 else	S_2.next = S.next
S_2	S.code = C.code || "label" || L1 || S_1.code || "goto" || S.next || "label" || L2 || S_2.code

5.5.1.(1)

```
string S(label next) {
    if (current input == "if") {
        Advance input;
        Check "(" is the next input and advance;
        label L1 = new();
        label L2 = new();
        string C_code = C(L1, L2);
        Check ")" is the next input and advance;
        string S1_code = S1(next);
        Check "else" is the next input and advance;
        string S2_code = S2(next);
        string S_code = C_code || "label" || L1
            || S1_code || "goto" || next
            || "label" || L2 || S2_code;
```

```
        return S_code;
    }
    else // Other statement type.
}
```

5.5.2.(1)

```
void S(label next) {
    if (current input == "if") {
        Advance input;
        Check "(" is the next input and advance;
        label L1 = new();
        label L2 = new();
        C(L1, L2);
        Check ")" is the next input and advance;
        print("label", L1);
        S1(next);
        print("goto", next);
        Check "else" is the next input and advance;
        print("label", L2);
        S2(next);
    }
    else // Other statement type.
}
```

5.5.5.(1)

产生式	归约时的动作
S -> if (X C) Y S_1 else Z S_2	temp = stack[top-6].code || "label" || stack[top-7].L1 || stack[top-3].code || "goto" || stack[top-10].next || "label" || stack[top-7].L2 || stack[top].code top -= 9 stack[top].code = temp
X -> ε	top += 1 stack[top].L1 = new() stack[top].L2 = new() stack[top].true = stack[top].L1 stack[top].false = stack[top].L2
Y -> ε	top += 1; stack[top].next = stack[top-6].next
Z -> ε	top += 1; stack[top].next = stack[top-9].next

栈结构示意图:

0	1	2	3	4	5	6	7	8	9	10
?	if	(	X	C	)	Y	S_1	else	Z	S_2

0	1	2	3	4	5	6	7	8	9	10
S.next			L1	C.code		S_1.next	S_1.code		S_2.next	S_2.code
			L2							
			C.true							
			C.false							