

编译原理第 10 次 (?) 作业

答案不唯一，包含了必要的信息即可。

P166 2

考虑下面的类 ALGOL 程序：

```
BEGIN
  INTEGER I;
  REAL X, Y;
  READ(I, X, Y);
  EXAMPLEPROC(I, X, Y);
  PROCEDURE EXAMPLEPROC(INTEGER J, REAL X, Y)
  BEGIN
    STRING S(8);      ! 点 1
    ...
    BEGIN
      REAL R(J, J);   ! 点 2
      ...
    END
  END
END
END
```

画出执行到点 1 和点 2 时，运行时栈的内容。

答案解析

点 1:

```

1  /*
2  |   数组 S   |
3  |   Y       |
4  |   X       |
5  |   J       |
6  |S 的模板/模块 |
7  | prev abp |
8  | ret addr |
9  | abp(1)   | <-- abp(2)
10 |   Y       |
11 |   X       |
12 |   I       | <-- abp(1)
13 */

```

点 2:

```

1  /*
2  |   数组 R   |
3  |R 的模板/模块 |
4  | prev abp |
5  | ret addr |
6  | abp(2)   |
7  | abp(1)   | <-- abp(3)
8  |   数组 S   |
9  |   Y       |
10 |   X       |
11 |   J       |
12 |S 的模板/模块 |
13 | prev abp |
14 | ret addr |
15 | abp(1)   | <-- abp(2)
16 |   Y       |
17 |   X       |
18 |   I       | <-- abp(1)
19 */

```

P175 1

将条件语句 `if X=Y+2 then Z:=X else Z:=Y+1` 转换成波兰后缀表示

答案解析

答案不唯一。

下面使用 `jez` 代表 `jump if zero`, 使用 `jmp` 代表跳转, 并且地址从 1 开始计数。

源代码:

```
1  if X=Y+2 then
2      Z:=X
3  else Z:=Y+1
```

预留跳转标签的占位, 可以得到如下后缀式:

```
1  // 1 2 3 4 5 6 7
2      X Y 2 + = L1 jez
3
4  // 8 9 10 11 12
5      Z X := L2 jmp
6
7  // L1: 13 14 15 16 17
8          Z Y 1 + :=
9
10 // L2: 18
```

回填地址, 得:

```
1  // 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
2      X Y 2 + = 13 jez Z X := 18 jmp Z Y 1 + :=
```

`jmp L2` 的 `L2` 如果是 17, 视为错误 (如果从 0 开始计数则 16 错误)

P175 2

将语句 $A := (B+C) \uparrow E + (B+C) * F$ 转换成三元式、间接三元式和四元式序列。

答案解析

答案不唯一。

先转换成后缀式： $A \ B \ C \ + \ E \ \uparrow \ B \ C \ + \ F \ * \ + \ :=$ ，可以确认执行的顺序是 $+ \ \uparrow \ + \ * \ + \ :=$

则三元式序列为：

- 1 (1) +, B, C
- 2 (2) \uparrow , (1), E
- 3 (3) +, B, C
- 4 (4) *, (3), F
- 5 (5) +, (2), (4)
- 6 (6) :=, A, (5)

可以进行一定的优化：

- 1 (1) +, B, C
- 2 (2) \uparrow , (1), E
- 3 (3) *, (1), F
- 4 (4) +, (2), (3)
- 5 (5) :=, A, (4)

间接三元式序列：

- 1 // 三元式
- 2 (1) +, B, C
- 3 (2) \uparrow , (1), E
- 4 (3) *, (1), F
- 5 (4) +, (2), (3)
- 6 (5) :=, A, (4)
- 7 // 操作序列
- 8 (1)(2)(1)(3)(4)(5)

四元式序列：

```

1  +, B, C, T1
2  ↑, T1, E, T2
3  *, T1, F, T3
4  +, T2, T3, A

```

如果四元式最后有 $:=$ ，也可以接受。

P175 4

为第 2 题的语句构造抽象语法树

答案解析

由于后缀式是抽象语法树的后序遍历序列，直接根据 $A \ B \ C \ + \ E \ \uparrow \ B \ C \ + \ F \ * \ + \ :=$ 自底向上构造语法树。

类似地，前缀式是前序遍历序列，可以直接根据 $:= \ A \ + \ \uparrow \ + \ B \ C \ E \ * \ + \ B \ C \ F$ 自顶向下构造。

语法树：

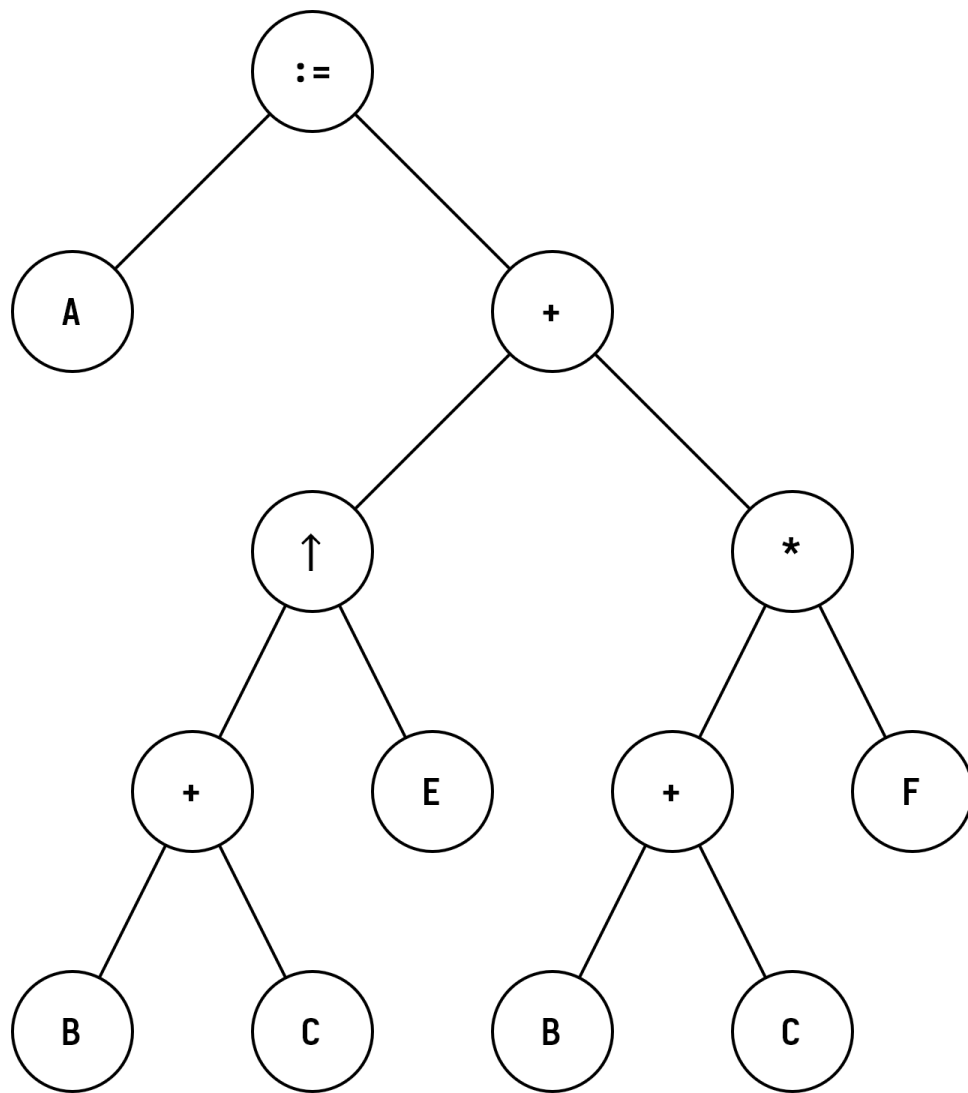


Figure 1: ast