

Assignment 1

练习1.1.1: 编译器和解释器之间的区别是什么?

答: 编译器可以阅读用源语言编写的程序, 并把该程序翻译成为一个等价的、用目标语言编写的程序。

解释器是一种另一种常见的语言处理器。它不通过翻译的方式生成目标程序。从用户的角度看, 解释器直接利用用户提供的输入执行源程序中指定的操作。

练习1.1.2: 编译器相对于解释器的优点是什么? 解释器相对于编译器的优点是什么?

答: 在把用户输入映射成为输出的过程中, 编译器产生的机器语言目标程序执行得比解释器快很多。但是解释器的错误诊断效果通常比编译器更好, 因为它逐个语句地执行源程序。

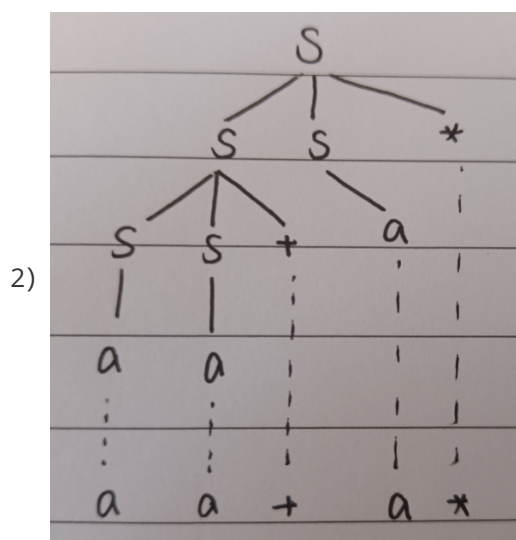
练习2.2.1: 考虑下面的上下文无关文法:

$$S \rightarrow SS+ | SS* | a$$

- 1) 试说明如何使用该文法生成串 $aa+a^*$ 。
- 2) 试为这个串构造一棵语法分析树。
- 3) 该文法生成的语言是什么? 证明你的答案。

解:

1) $S \rightarrow SS* \rightarrow SS+S* \rightarrow aS+S* \rightarrow aa+S* \rightarrow aa+a^*$



3) $L = \{\text{由数字、加号、乘号组成的后缀表达式}\}$

证明: 该文法的终结符号包括 $a + *$; 且加号和乘号均为二元运算符, 在文法中写在二元末尾, 说明生成的是后缀表达式。

练习2.3.1: 构建一个语法制导翻译方案，该方案把算术表达式从中缀表示方式翻译成运算符在运算分量之前的前缀表示方式。例如， $-xy$ 是表达式 $x-y$ 的前缀表示法。给出输入 $9-5+2$ 和 $9-5*2$ 的注释分析树。

解:

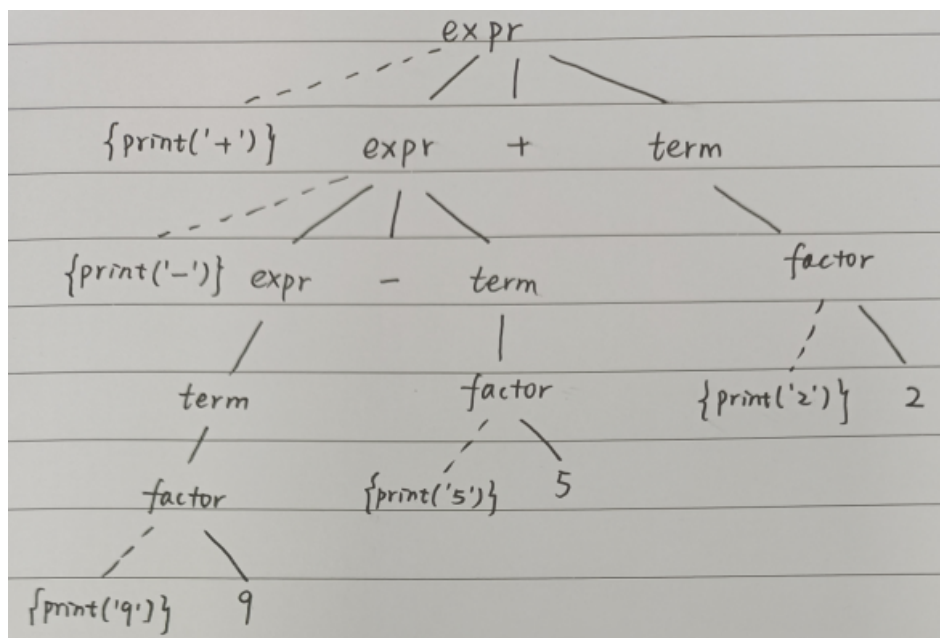
语法翻译方案为:

```

expr → {print('+')} expr + term
      | {print('-')} expr - term
      | term
term  → {print('*')} term * factor
      | {print('/')} term / factor
      | factor
factor → {print(digit)} digit
  
```

注释分析树如下:

9-5+2



9-5*2

