编译原理 第一次作业

> 学号:

Exercise 1.1

- Imagine an artificial computer language, which can be utilized to solve a practical problem, i.e. the application of the language.
  - Tips 1. Language is an alternative approach to problem solving.
  - Tips 2. First find a proper problem, then design a language to solve the problem.
- Give an example of a complete piece written in the proposed language.
- Discuss how to define the new language and try your approach.
- Describe the process of changing the thinking of your language to a reality, i.e. how to make the artificial language usable.

解:

1. 预期解决的问题是实现计算器的简单计算功能，包括加、减、乘、除、幂。

   每条语句以运算符开始，由于均为二元计算，因此后面还有两个操作数，即两个待计算的数字。运算符与两个操作数以“`，`”分开，以"`;`"结束，并且字符之间可以有任意数量的空格。

   运算符（不妨设第一个数字为a，第二个为b）：

   - `+ - *`分别代表加减乘除;
   - `^`代表幂运算，第一个操作数为底数，第二个操作数为指数，即 a^b;
   - `/`代表对数运算，第一个操作数为被除数，第二个操作数为除数，即 a/b;
2. 例子:

```
+ ，  1 ，  2;  //1+2
- ，  4 ，  2;  //4-2
* ，  2 ，  3;  //2*3
/ ，  4 ，  2;  //4/2
^ ，  2 ，  3;  //2的3次方
```

3. 对该语言的定义:

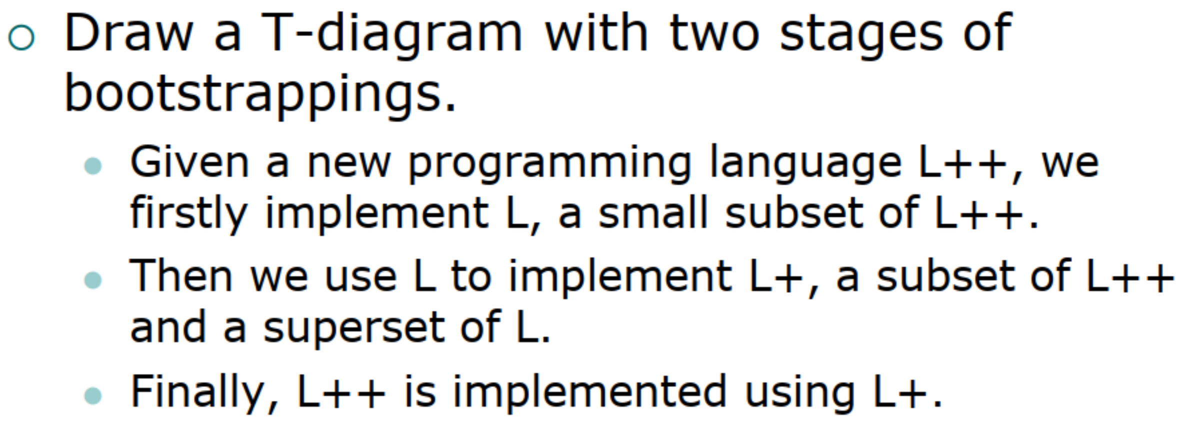
```
字符集：{1,2,3,4,5,6,7,8,9,0,+,-,*,/,^,;}

BNF：
    <计算语句> ::= <运算符> ， integer ， integer ;
    <运算符>   ::= + | - | * | / | ^
    integer   ::= [ symbol ] unsigned
    unsigned  ::= digit { digit }
    symbol    ::= + | -
    digit     ::= 0 | 1 | ... | 9
```

4. 实现这门语言:

   首先对每种计算分别编写相应的计算程序

   在执行的时候，先读取指令，之后按照 “`，`” 进行分词。分词后先读取第一个操作符，再读取后面的两个操作数，最后传给对应的程序进行计算即可。

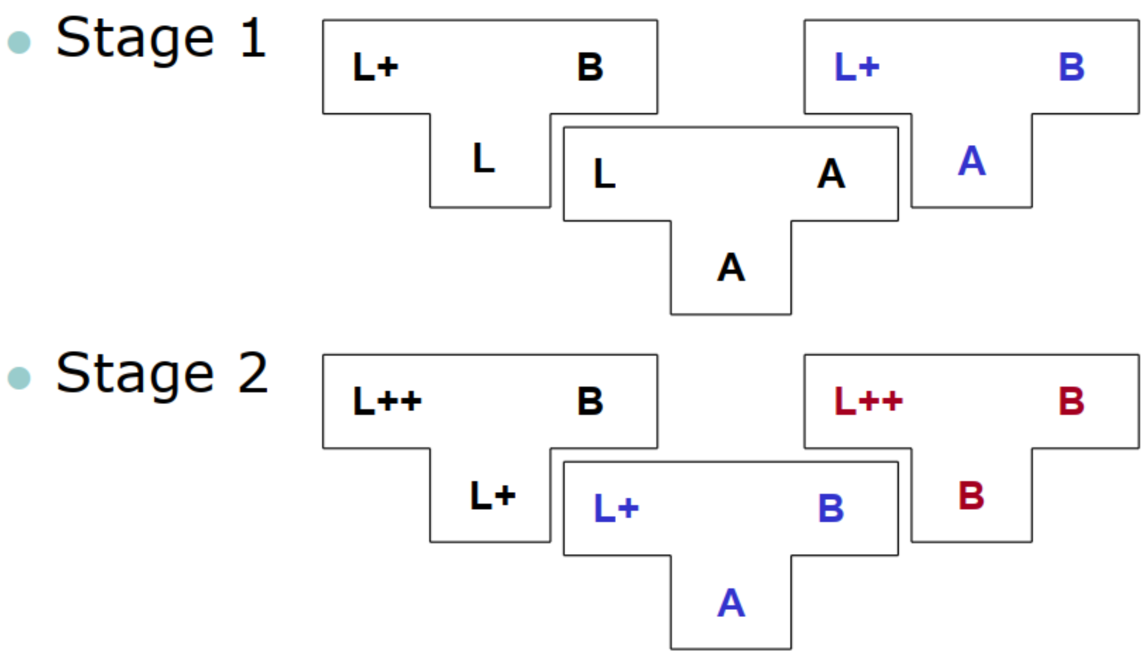
Exercise 1.2



○ Draw a T-diagram with two stages of bootstrappings.

- Given a new programming language L++, we firstly implement L, a small subset of L++.
- Then we use L to implement L+, a subset of L++ and a superset of L.
- Finally, L++ is implemented using L+.

解:



- Stage 1
- Stage 2