

2019 年秋季学期编译原理课程实践

1 实验目标:

设计实现一个交互式的迷你 python 程序的解释器。

实验文件包中的两个文件 minipy-lab.y 和 minipy-lab.l 分别给出了迷你 python 程序的 yacc 语法描述和 lex 词法描述。

为简单起见, 迷你 python 程序, (1) 仅包含了数值 (INT 和 REAL)、串 (STRING_LITERAL) 列表 (list) 等基本数据类型。(2) 仅包含加减乘除等基本运算, (3) 诸如取列表元素, 取列表片段等操作, (4) 列表的方法和/或内置函数调用等, (5) 赋值运算。

2 实验要求:

使用 yacc 和 lex 来设计并实现迷你 python 解释器。

2.1 阅读迷你 python 程序的语法及词法描述文件;

2.2 参考标准 python 语言约定, 设计合理可行的语义操作描述 (特别是针对列表和串类型的语义检查), 扩展完善上述两个描述文件;

2.3 至少要实现列表 append() 方法, 以及 range()、print()、len()、list(), 等内置函数。输入 "quit()" 后可结束解释器的运行。

2.4 参考的运行结果:

```
[alex@localhost vmshared]$ ./minipy
miniPy>
miniPy>
miniPy> a = [1,2,3,4,]
miniPy> print(a)
[1 , 2 , 3 , 4]
miniPy> b = a * 2
miniPy> print(b)
[1 , 2 , 3 , 4 , 1 , 2 , 3 , 4]
miniPy> print(a[0])
1
miniPy> a[0] = "hello!"
miniPy> print(a)
["hello!" , 2 , 3 , 4]
miniPy> a[2] = [5,6,7]
miniPy> print(a)
```

```
["hello!" , 2 , [5 , 6 , 7] , 4]
miniPy> a[2].append(8)
miniPy> print(a)
["hello!" , 2 , [5 , 6 , 7 , 8] , 4]
miniPy> print(a[2][1])
6
miniPy> c = b + [7,8,9]
miniPy> print(c)
[1 , 2 , 3 , 4 , 1 , 2 , 3 , 4 , 7 , 8 , 9]
miniPy> c[2].append(10)
A non-list found!
miniPy> d = c + 2
Invalid '+' operation for list!
miniPy>
```