C->Python编译器说明文档

0 使用方法

pip install antlr4-python3-runtime

cd compiler

python compiler.py [kmp, calc, palindrome, AVLtree]

其中kmp需要输入文本串和模式串，calc需要输入表达式，palindrome需要输入字符串，AVLtree数据已经预定义好，不需要输入。

cd ../test

python [kmp.py, calc.py, palindrome.py, AVLtree.py]

1 支持的C语法特性

* 变量的声明、定义。
* 支持int a = 1, b, c = 3;型的调用
* 支持数组的定义方式: int a[100];

int a[] = {1, 2, 3};

char s[1000];

char s[] = “abcdefg”

struct Node A[1000];

* 函数的定义与调用。支持在表达式中调用函数。
* For循环。由于python语法特性，仅支持for( int i = [表达式]; i [<,<=, >=, >] [表达式]; i[++,--] 或 [++,--]i 或 i = i + [表达式]){}类型。
* While循环。
* Break, continue, return语句。
* If条件判断。支持if/if-else/if-else if\*n/if-else if\*n-else。
* 表达式。
* 逻辑表达式。支持&&、||及其嵌套。
* 算术表达式。支持+、-、\*、/。
* 自增/减。支持++、--。支持前自增/减（即b = ++a）在表达式中使用。
* Struct定义和使用。

2 重点

* 对于struct，考虑到映射到python的class，这样不仅方便定义成员变量，如果进行扩展，也方便后续成员函数的添加。struct的类型标识符统一到了类型标识符中，在对应的语法中通过判断类型名中是否含有struct关键字再进行进一步的处理。另外，对于此标识符用于定义结构体还是定义结构体对象还需要通过判断是否含有定义体进一步区分处理。对于定义体，解析为变量声明/赋值即可。

对于struct数组声明，需要考虑到在映射到python类创建对象数组时的浅拷贝问题，只能通过每个对象单独创建进行解决。我们在这里利用lambda表达式创建列表的方式避免了直接通过\*数目的方式产生的浅拷贝。

* 对于if-else结构，要正确识别出整个结构，需要正确的区分出语法树中以if语句结束、else if语句结束、else语句结束，以及这三者后面还接着响应的选择语句(selectionStatement)的情况。对于if语句结束的情况，和其他两种情况的区分，可以通过判断选择语句的元素个数来区分开，若某一selectionStatement不含else语句，则其就以if语句为结束。其他两种情况的区分需要去查看它们的statement元素中是否含有else，如果不含else，则为else if做结束，否则继续解析。若statement元素的属性不为selectionStatement，说明选择分支就此结束。
* 需要将C中的printf函数转成Python中的print函数。此外print函数默认末尾带换行符，需要将原来的参数中末尾的换行符去除。
* Python中没有自增/减，因此我们将a++/++a转化为 a += 1/ a = a + 1，其中a++不支持在表达式中使用，如b = a++，但是++a支持，如b = ++a可转化为b = a = a + 1。

3 参考资料

语法文件参考了 <https://github.com/antlr/grammars-v4/blob/master/c/C.g4>

测试用例参考了 https://github.com/DingZheng2016/C-to-LLVM-compiler/tree/master/testcases