CS335: Compiler Design

Final Demo

- By Group 6

Python compiler

- The goal of this project was to implement a compilation toolchain, where the input is in a statically typed subset of the Python language and the output is x86_64 code
- This was done in 3 milestones:
  - Milestone 1 - To construct a scanner and a parser and output a graphical representation of the AST of the input program
  - Milestone 2 - To add the symbol table, perform semantic analysis, generate 3AC code and add runtime support for function calls
  - Milestone 3 - To generate x86_64 assembly from the 3AC code and run it via GAS on Linux

Features supported

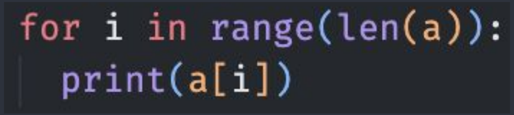
- Primitive data types (e.g., int, float, str, and bool)
- 1D lists
- Basic operators:
    - Arithmetic operators: +,-, *, /, //, %, **
    - Relational operators: ==, !=, >, <, >=, <=
    - Logical operators: and, or, not
    - Bitwise operators: &, |, ˆ, ˜, «, »
    - Assignment operators: =, +=, -=, *=, /=, //=, %=, **=, &=, |=, =ˆ, «=, »=

Features supported

- Control flow via if-elif-else, for, while, break, and continue
- Support iterating over ranges specified using the range() function.
- Support for recursion
- Support for the library function print() for only printing the primitive Python types, one at a time
- Support for classes and objects, including multilevel inheritance and constructors
- Methods and method calls

We have supported all the features that were required.

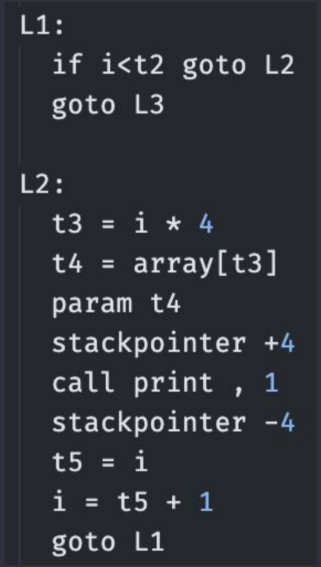
Example



```python
for i in range(len(a)):
  print(a[i])
```

Python

->

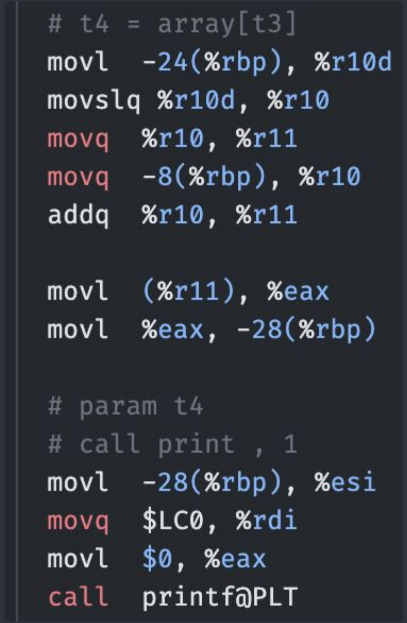


```
L1:
  if i<t2 goto L2
  goto L3

L2:
  t3 = i * 4
  t4 = array[t3]
  param t4
  stackpointer +4
  call print , 1
  stackpointer -4
  t5 = i
  i = t5 + 1
  goto L1
```

3AC

->



```
# t4 = array[t3]
movl   -24(%rbp), %r10d
movslq %r10d, %r10
movq   %r10, %r11
movq   -8(%rbp), %r10
addq   %r10, %r11

movl   (%r11), %eax
movl   %eax, -28(%rbp)

# param t4
# call print , 1
movl   -28(%rbp), %esi
movq   $LC0, %rdi
movl   $0, %eax
call   printf@PLT
```

x86_64

Demo

Overall Effort Sheet

- Yash Gupta (190997) - 33%
- Devesh Shukla (200322) - 33%
- Shivang Pandey (200941) - 33%

All members contributed equally and worked on different tasks.

Thank you