CS335 (Compiler Design) : Project Milestone 3

Deepak Sangle
200860
Department of Computer Science
`sangleds20@iitk.ac.in`

Shreyasi Mandal
200956
Department of Computer Science
`shreyansi20@iitk.ac.in`

Vartika
201089
Department of Computer Science
`vartikag20@iitk.ac.in`

JAVA17 Compiler

We extended the 3Ac Intermediate Representation (IR) of JAVA 17 (combined) lexer and parser called ASTGenerator, to add runtime support for procedure calls. Our implementation of activation records includes the following fields -

- space for actual parameters,
- space for return value,
- space for old stack pointers to pop an activation,
- space for saved registers,
- space for locals.

Tools Used

The following tools are used for creating **AST Generator**.

- Flex - This is used for the lexer implementation of the java program
- Bison - This is used to generate the parser
- Graphviz - Graph visualisation tool to visualize the AST

Language Features Supported

- Basic Operators: $+$, $-$, $*$, $/$, $++$, $--$,$\hat{}$, !, $\sim$
- Assignment Operators: $+=$, $-=$, $*=$, $/=$, $\&=$, $|=$,$\hat{}=$, $\%=$
- Conditional Operators: $==$, $!=$, $>=$, $<=$, $>$, $<$, $||$, $\&\&$
- Bitwise Operators: $<<$, $>>$, $>>>$
- Control Flow: if-then, if-then-else, for and while
- Primitive data types: int, byte, short, long, float, double
- 1D arrays with primitive data types
- Methods and method calls
- Recursion

Preprint. Under review.

- Library function println()
- Classes and objects

My 3AC Convention

Following is my 3AC calling convention for stack manipulation

- The pushparam keyword pushes the temporary register onto the stack and is used appropriately by popparam keyword where this parameter is popped.
- The %rax register stores the return value of a function call. This resistor's value is then assigned to any temporary register in the caller function.
- The %sp register denotes the stack frame pointer. It is created after every function is called.
- The stack pointer is incremented every time a local variable, array, or class is initialized. I have used the "push x" operation to denote that we are pushing the variable x onto the stack. After that, we are also incrementing the value of the stack frame pointer according to the size allocated to the variable x (in the case of class objects, I have allocated a size equal to the sum of sizes of all field members of it).
- I have not completely assumed the number of temporaries to be infinity. The number of local variables present in any function limits it.
- The call keyword calls the appropriate function and automatically adjusts the stack frame pointer along with the return values.

Note: I have not dealt with statements like System.out.println. The reason is that I have not added any Class named System for it to reference. There are many other functionalities like this; I will add all these classes in the next milestone.