Gazprea Meeting Notes

James, Tianming, Ian & Justin

November 04 2022 - Project Meeting #1

November 07 2022 - Scrum Meeting #1

November 11 2022 - Project Meeting #2

November 14 2022 - Scrum Meeting #2

November 17 2022 - Project Meeting #3

Project Review Meeting #1

Meeting Information:

Meeting date: Nov 4, 2022
Time start: 2:30 PM
Time end: 3:21 PM

Attendees:
Ian (Arrived at 2:30PM)
Justin (Arrived at 2:30PM)
Tianming (Arrived at 2:30PM)
James (Arrived at 2:30PM)

Progress:

Done:

- AST
    - Grammar finished
    - ASTBuilder finished

In progress:

- Definition walk
- Reference walk (James leading the way by following lang patterns)
- Grammar tweaks

Future:

- Type
    - Expression computation
    - Type promotion
- Code generation
- Tianming will focus on runtime, others on walking and building AST, code will meet during codegen walk.

Issues:

- Figuring out a way to resolve type of a typedef

Round table:

- James:
    - Working on the definition and referencing
- Tianming
    - Working on type checking
- Justin

    - Working on typedef resolution
- Ian
    - Working on testing (unit testing), also generates random codes to input

Scrum Meeting #1

Meeting Information:

Meeting date: Nov 7, 2022
Time start: 6:30 PM
Time end: 7:03 PM

Attendees:
Justin (Arrived at 6:30PM)
James (Arrived at 6:30PM)
Tianman (Arrived at 6:30PM)

Progress:

James:

- AST Builder complete
- Def and Ref walks populate the AST with symbols and scopes
- VisitVariableDeclaration, Assignment, TupleLiteral, TupleAccess in progress and going well.
- Moving onto TypeWalk where we will handle Type Promotion, evaluation types and generally populating evaluationType fields of AST nodes in contexts like assignment, declaration, operations etc.

Justin:

- Visit Typedef in Typewalk working.
- Will be handling the binaryOperation visit in typewalk.
- Creating a promotion table from the ANTLR4 Language Implementation patterns book.
- Promote from to Table.
- Will have the ability to enforce error handling for a number of scenarios using typewalk.

Tianming:

- Grammar
    - Changes to visitParameterAtom
    - Grammar is looking good now.
- Runtime code
    - Handling Type and Variable initialization
- Expanding runtime library
    - Binary ops in the runtime. Runtime type checking and computation.
    - Working very well

Issues:

Continually making tweaks to the grammar file creates extra work when refitting existing walks to new rules.

Project Review Meeting #2

Meeting Information:

Meeting date: Nov 11, 2022
Time start: 2:30 PM
Time end: 3:11 PM

Attendees:
Ian (Arrived at 2:30PM)
Justin (Arrived at 2:30PM)
Tianming (Arrived at 2:30PM)
James (Arrived at 2:30PM)

Progress:

Done:

- Set up the codegen walk, reading to start linking in Tianming's runtime and seeing our first program execution happen.
- Everything from def, ref and typewalk lets us start codegen.

In progress:

- Control flow IR, declaring the LLVM function types for all of the runtime interfaces, creating subroutine IR, handle global variables, enforce compile time checks,
- Referencing the runtime interface whenever we can to get full value.

Future:

- Type
    - Expression computation
    - Type promotion
- Code generation

Issues:

- We are entering a large building phase in codegen. Major issues will be found when trying to bridge the gap between runtime and codegen and when we begin extensive testing.

Round table:

- James:
    - Working on visitUnqualifiedType, using the initializeType runtime library, handling for scalars, vectors and tuples.
    - Handling visit character atom and escape sequences

    - Handling visit Tuple access in codegen.

- Tianming
    - Created the testing suite. /tests/helpers/Readme.MD includes valgrind integration which will be useful for part 2
    - Continuing to improve the runtime interface. Variable and Type initialization working for all Part 1 types.
- Justin
    - Handling all control flow in the ir builder, conditional (nested else if will be a challenge) and the variations of loops excluding iterator.
    - Will be starting to add compile time exceptions all over the program.
- Ian
    - Working on testing (unit testing), also generates random codes to input

Scrum Meeting #2

Meeting Information:

Meeting date: Nov 14, 2022
Time start: 6:31 PM
Time end: 6:58 PM

Attendees:
Justin (Arrived at 6:31PM)
James (Arrived at 6:30PM)
Tianman (Arrived at 6:31PM)
Ian (Arrived at 6:30PM)

Progress:

James:

- AST Builder complete
- Def and Ref walks populate the AST with symbols and scopes
- VisitVariableDeclaration, Assignment, TupleLiteral, TupleAccess in progress and going well.
- Moving onto TypeWalk where we will handle Type Promotion, evaluation types and generally populating evaluationType fields of AST nodes in contexts like assignment, declaration, operations etc.

Justin:

- Visit Typedef in Typewalk working.
- Will be handling the binaryOperation visit in typewalk.
- Creating a promotion table from the ANTLR4 Language Implementation patterns book.
- Promote from to Table.
- Will have the ability to enforce error handling for a number of scenarios using typewalk.

Tianming:

- Grammar
    - Changes to visitParameterAtom
    - Grammar is looking good now.
- Runtime code
    - Handling Type and Variable initialization
- Expanding runtime library
    - Binary ops in the runtime. Runtime type checking and computation.
    - Working very well

Ian:

- Working on testing.

Issues:

Continually making tweaks to the grammar file creates extra work when refitting existing tree walks to new rules.

Project Review Meeting #3

Meeting Information:

Meeting date: Nov 17, 2022
Time start: 7:25 PM
Time end: 8:05 PM

Attendees:
Ian (Arrived at 7:23PM)
Justin (Arrived at 7:25PM)
Tianming (Arrived at 7:23PM)
James (Arrived at 7:22PM)

Progress:

Done: -Many test cases are passing successfully

- Most compile time exceptions covered
- Tianming & James fixed a major issue with var qualifier in parameter atom causing segmentation fault if the scalar type was inferred. Justin added the exception for modifying a const parameter via assignment to it.

In progress: -The issues described in issues are being resolved.
Future: - This is the final meeting so everything in progress is our final concern.

Issues:

- Tuple passed as a parameter not preserving aliased labels for indexes when changed. (James fixing this now)
- Min negative Integer Error
    - Problem with stoi possibly needs to be resolved on both the codegen and runtime front (James and Tianming working on this together)
- If statements that contain return statements are causing a terminator in the middle of block error, (Justin working on this)

Round table:

- James:
    - Will implement VariableSwapType in Codegen for tuple issue
    - Modifications to visitCallInSubroutine()
- Tianming
    - Adding runtime capabilities for min negative integer and floating point.

    - Tuple issue fixed on the runtime side.
- Justin
    - Just added const mutability and assignment to unknown type error handling
    - Know how to fix the conditional statement bug with a flag.
- Ian
    - Creating unit tests, merging them into master soon.
    - Found bugs related to tuple and found INT MIN bug.