Claims

Group 9
Anton Nørgaard (antno16), Bjørn Glue Hansen (bhans09) & Thor Skjold Haagensen (thhaa16)

Language Extensions

- For statement, increment statement, decrement statement, array concantenation, modulo operator, copy statement, copy range statement, sqrt function and read statement.

Optimization

- **Constant Folding:** Constant expressions are calculated during the weeding phase.
- **Liveness:** Liveness is performed on temporaries created in the intermediate language. This helps simplify the intermediate code. Because temporaries are only live within a statement, the flow control graph is split into statement blocks. This splits function into many smaller parts that are all basic blocks. This makes it easy and quick to perform the analysis. It is missing spill in case there are not enough registers available and will abort when running out.
- **Peephole:** Mostly focused on changing arithmetic instructions. Removal of redundant assignments was removed as it messed up comparison expressions.
- **Dynamic Assignment of Static Link:** The r9 register is responsible for references to variables in other scopes and it is updated based on the number of jumps in the symbol table when such a variable is being used. The compiler is able to detect what the number of jumps for a previous variable was and if it actually need to update the r9 register. Because control flow can mess this up, it is always updated after a label by default.
- **Flags:** Can disable runtime checks and set r9 assignment to a more simple variant through the -nc and -ss flags. When using the more simple assignment of r9, the compiler will not guarantee updates of r9 after a label. This more simple approach will result in some programs with nested functions to not work correctly, but all programs without nested functions will still be able to work.

Runtime

- **Runtime Checks:** Unless the -nc flag is used, the compiled kitty program has runtime checks. These checks return a message and status code on an exception. It shouldn't be possible for a program to segfault.
- **Dynamic Heap Allocation:** The heap is allocated by changing the program break, starting at 8KB. This can be increased up to 1GB. If this is not enough, the program throws an exception and terminates. It is not possible for the heap to shrink.
- **Garbage Collection:** We use stop and copy garbage collection. While this results in half of the heap being wasted, we do not have to worry about fragmentation or memory management.