

P3. Machine Code Generation

Kelley van Evert
s4046854

Not complete

Far from! D: D:

(again... why?)

(Shameless promotion)



Exhibited next year on the ground floor :)

Splb

- Splb = “Simple C-like stuff...”
 - i.e.
 - first-order functions at the toplevel
 - lexical scope, by block (function body, if, while, { ... })
 - follow the standard C-like stack discipline
 - lists and tuples by reference, on the heap
- P4 = Spla: whatever needed to do the language thing
 - i.e.: parse itself (I probably won't get to P2/P3)
 - parser combinators → higher-order functions
 - different abstract machine?

Machine Code Generation

- AST → IR: halfway?
 - It's taking me a while to wrap my head around this translation process
- IR → ASM: simple and stupid, nearly finished
 - Just tying things together in the right way

(~ 300 LOC, definitely won't get bigger than twice this)

Machine Code Generation

- AST → IR
 - My AST features blocks (fn body, if, while, { ... })
 - each with own variable declarations
 - lexical scoping
 - e.g.

```
int main () {  
    int x;  
    x := 4;  
    { int x;  
      x := 5; }  
    return x; // returns 4  
}
```
 - “How to store/address these variables in frame?”

Machine Code Generation

- $AST \rightarrow IR$
 - My AST features blocks (fn body, if, while, { ... })
 - “How to store/address these variables in frame?”
 - Translation encapsulated in state monad

```
type IR a = State IrState a
```

```
transformStmt2IR :: AstStmt → IR IrStmt
```

```
transformExpr2IR :: AstExpr → IR IrExpr
```

- Translation recurses into blocks
 - state includes current environment
 - solves lexical variable problem

Machine Code Generation

- AST \rightarrow IR
 - My AST features blocks (fn body, if, while, { ... })
 - “How to store/address these variables in frame?”
 - Translation encapsulated in state monad
 - other stately things include e.g.
 - fresh labels
 - function code addresses

Machine Code Generation

- AST \rightarrow IR
 - My AST features blocks (fn body, if, while, { ... })
 - “How to store/address these variables in frame?”
 - Translation encapsulated in state monad
 - Currently no optimization whatsoever :P
 - ~~involved register usage etc~~

(Questions)