CMSC 430: Introduction to Compilers

Con: conditional execution

Growing a Language

- Write examples
- Extend concrete syntax
- Extend abstract syntax
- Extend parser
- Revise interpreter to specify semantics
- Revise compiler & run-time system to implement semantics
- Test against examples

Conditional execution

- Con adds **conditionals** to our target language.
- Concrete syntax: `(if (zero? e0) e1 e2)`
  - `(if (zero? 0) (add1 2) 4)`
  - `(if (zero? 1) (add1 2) 4)`

- AST: `IfZero e0 e1 e2`
  - `(if (zero? 0) (add1 2) 4)`

  parsed as
    - `(IfZero 0 (add1 2) 4)`

Con Operational Semantics

```
(if (zero? 0) (add1 2) 4) means 3
(if (zero? 1) (add1 2) 4) means 4
```

Not con programs:

```
(zero? 0)
(if #t 5 10)
(if 1 10 20)
```

Con: Let us implement it

- Ast: ast.rkt
- Parser: parse.rkt
- Interpreter: interp.rkt
- Compiler: compile.rkt
- Randomized testing: random.rkt