

Review

Geoffrey Matthews

March 6, 2023

Chapter 1, 2: AE

- AE, p. 7
- parse, p. 8
- calc, p. 13

Chapter 3: WAE

- WAE, p. 16
- subst, p. 22
- calc, p. 20
- Recap: p. 26

Chapter 4: F1WAE

- F1WAE, p. 28
- interp, p. 29
- Recap: p. 31

Chapter 5: Environments

- DefrdSub, p. 34
- lookup, p. 34
- Recap: p. 38

Chapter 6: First-Class Functions

- FAE
- FAE-Value
- closureV
- interp
- Recap: p. 54

Chapter 7: Lazy programming

- Haskell
- Shell scripting
- Lazy Scheme consing

Chapter 8: Implementing Laziness

- CFAE/L-Value, p. 75
- closureV
 - exprV
 - strict, p. 76
- Caching
- CFAE/L-Value, p. 78
- boxed-boolean/CFAL-Value?
- strict, p. 79
- Recap: p. 83

Chapter 9, 10: Recursion

- Recap: p. 101

```
(define (cyclically-bind-and-interp
        bound-id
        named-expr
        env)

(local (
  [define value-holder (box (numV 1729))]
  [define new-env (aRecSub bound-id value-holder env)]
  [define named-expr-val (interp named-expr new-env)])
(begin
  (set-box! value-holder named-expr-val)
  new-env)))
```

Continuations

- Convert function calls to stack/while loop
- Convert function nesting to continuation-passing style, CPS
- `call/cc` and `let/cc`