Week 15 day 2 (Final Info)

Table of Contents

[CS 403 - 001 Spring 2016](/CS403-001/)

[About](/CS403-001/about/)

# Week 15 day 2 (Final Info)

Apr 28, 2016

Download Word (docx):

## Navigate using audio

# Assignment 3 # 2 Solution

(define (cache f)  
 (define locals (get 'parameters f))  
 (define denv (get' \_\_context f))  
 (define (replace code)  
 (cond  
 ((null? code) nil)  
 ((and (symbol? code) (member? code locals))  
 code  
 )  
 ((symbol? code)  
 (get code denv)  
 )  
 ((and (pair? code) (eq? (car code) 'define) (pair? (cadr code)))  
 (set! locals (cons (car (cadr code)) locals))  
 code  
 )  
 ((and (pair? code) (eq? (car code) 'define))  
 (set! locals (cons (cadr code) locals))  
 (cons 'define (cons (car code) (replace (cddr code))))  
 )  
 ((pair? code)  
 (cons (replace (car code)) (replace (cdr code)))  
 )  
 (else  
 code  
 )  
 )  
 )  
 (set 'code (replace (get 'code f)) f)  
)

# Final Info

## Ch 1: + we’re going to have to convert from recursive to iterative + aka how many formal parameters? + which use stores vs sources? + normal vs applicative order + normal order is delayed evaluation + converting from lets to lambdas and local defines and vice versa ## Ch 2: + Church numerals and data structures / data directed programming ## Ch 3: + Evaluation + How environments arise and what they look like (major part of exam) + How many links go here? + What gets garbage collected? + Audio 0:01:52.576666

# Drawing Environment example

(define x 3)  
(define b 4)  
(define (max x b)  
 (if (< x b)  
 b  
 x  
 )  
)  
(define )

|  |  |
| --- | --- |
| VARS | VALUES |
| b | 4 |
| max | CLOSURE: {signature, body, calling\_env} |

^

|

|  |  |
| --- | --- |
| VARS | VALUES |
| x | 4 |
| b | 3 |

eval if… return 4 (Audio 0:05:34.761196 )

* Audio 0:13:14.443867
  + lost in the jibber jabber
  + Programs are stupid and so are we

(define (max x b)  
 (define (debug)  
 (inspect b)  
 (inspect x)  
 )  
 debug  
)  
  
(define x 3)  
(define b 4)  
(define (max x b)  
 (if (< x b)  
 b  
 x  
 )  
)  
(define c (max b x))  
(c) ; needs an env too

* Audio 0:15:11.146399
* now we have to extend our second table again

|  |  |
| --- | --- |
| VARS | VALUES |
| b | 4 |
| max | CLOSURE: {signature, body, calling\_env} |

^

|

|  |  |
| --- | --- |
| VARS | VALUES |
| x | 4 |
| b | 3 |
| debug | CLOSURE: {sig,body,env} |

^

|

|  |  |
| --- | --- |
| VARS | VALUES |

…

(inspect b) b is 3

* Audio 0:19:33.761706
* We have to hook C up to E1 even though it has no formal parameters
* What if we changed debug to a lambda?
  + The closure in E1 would just have a different signature
* What if max returned 5?
  + C would just get a value instantly and E1’s closure gets garbage collected
* Audio 0:26:29.924377
  + Maybe he’ll ask one where every variable is named x
* [Sign up for demo](http://troll.cs.ua.edu/cgi-bin/appointments.cgi?lusth-demo)

## CS 403 - 001 Spring 2016

* CS 403 - 001 Spring 2016
* [jmbeach1@crimson.ua.edu](mailto:jmbeach1@crimson.ua.edu)
* jmbeach

Website for notes and study material for CS 403 (Programming Languages) at The University of Alabama Spring 2016