

CS3L: Introduction to Symbolic Programming

Lecture 17:
HOF and Tick-Tack-Toe

Summer 2008

Colleen Lewis
colleenL@berkeley.edu



Today

- `lambda`
- `every`
- `keep`
- `accumulate`
- Tick-tack-toe (ttt)
- Participation grades
- Homework
 - Miniproject 3 starts next week! Due Friday July 25th
 - It is already posted... If you want to get started.



lambda

```
(lambda (arg1 arg2 ...) body... )
```

- Why do we need lambda?
 - To do cool things with `every`, `keep`, `accumulate`



every

```
(every procedure sent )
```

- *procedure*
 - a procedure that takes in **one argument**
 - a procedure that returns a word or a sentence
- *sent*
 - a sentence with 0 or more words
 - a word with 0 or more letters



every example

```
(define (add-suffix suffix sent)  
  (every  
    (lambda (wd) (word wd suffix))  
    sent))  
> (add-suffix 'ing '(eat get let))
```

```
(se
```

```
(lambda ('eat) (word 'eat 'ing))  
'eating
```

```
(lambda ('get) (word 'get 'ing))  
'getting
```

```
(lambda ('let) (word 'let 'ing))  
'leting
```



keep

```
(keep procedure sent )
```

- *procedure*
 - a procedure that takes in **one argument**
 - a procedure that returns #t or #f
- *sent*
 - a sentence with 0 or more words
 - a word with 0 or more letters



keep example

```
(define (alliterations wd sent)
  (keep
    (lambda (one-wd)
      (equal? (first one-wd) (first wd)))
    sent))

> (alliterations 'cat '(car bar con))
```

(se

```
(lambda ('car)
  (equal? (first 'car) (first 'cat)))
'car)
(lambda ('bar)
  (equal? (first 'bar) (first 'cat)))
'())
(lambda ('con)
  (equal? (first 'con) (first 'cat)))
'con)
```

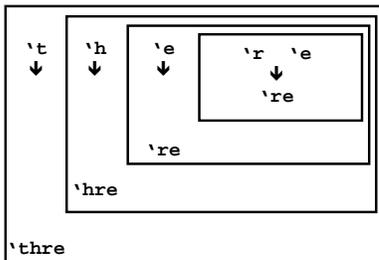
accumulate

(accumulate procedure sent)

- procedure
 - a procedure that takes in **two arguments**
 - a procedure that combines things together
- sent
 - a sentence with 1 or more words
 - a word with 1 or more letters

remove-dupls

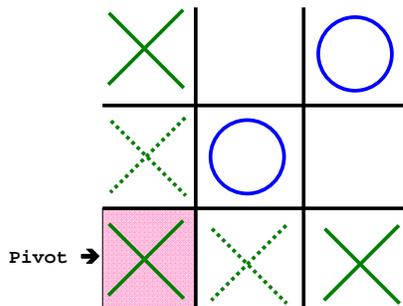
```
> (remove-dupls 'there) → 'thre
```



accumulate example

```
(define (remove-dupls wd)
  (accumulate
    (lambda (char wd-so-far)
      (if (member? char wd-so-far)
          wd-so-far
          (word char wd-so-far)))
    wd))
```

Tick-Tack-Toe (ttt)



Using lambda with define

- These are VERY DIFFERENT:
- How would you call each?

```
(define (adder-1 y)
  (lambda (x) (+ x 1)))
```

```
(define adder-2
  (lambda (x) (+ x 1)))
```

Participation Grades



- I did one of the random participation grading and very few people had done the brainstorm step that I randomly chose.
- http://inst.eecs.berkeley.edu/~cs3/su08/reader_files/geninfo.pdf
- You will also receive a small number of points for participation and for whether you have worked on particular activities in the on-line materials. Which activities we will check on won't be known to you (or to us, beforehand), but will generally be an activity for which you are asked to answer a question or create a file. The purpose of grading this is to encourage you to keep up-to-date on the lab materials; generally, the grades for these activities will consider whether you attempted it, rather than whether you did it well. You will be docked for participation points if you miss (some) lab sessions, although we may decide to additionally consider how well you contributed to the "community" in your lab session.