Midterm Review

Gilbert Chou
cs3-ta@imail.eecs.berkeley.edu

Navigating Text

C-v – move one screen forward
M-v – move one screen back
C-s/C-r – prompts for text to search. 's' searches forward. 'r' searches back
C-g – cancels a command
C-a – goes to front of the line
C-e – goes to back of the line
M-g – prompts for a line number then goes to it

Cut, Copy, Paste, Undo

C-k – kill the rest of the line
C-w – kill region
'Killing' text is the equivalent of cut

M-w – copy region
C-y – paste
M-y – pressing this after C-y will bring back earlier cuts
C-_ – undo last command

Arithmetic

Addition	+	
Subtraction	-	
Multiplication	*	
Division	/, quotient	*(integer division)*
Remainder	remainder	
Square root	sqrt	

Sample Question #1

Write a procedure that will evaluate expressions like '(1 + 1), '(1 * 1), '(1 / 1), '(1 - 1) '(1 % 1).
Usage: (calc '(1 + 1))

It does not need to evaluate longer expressions like '(1 + 1 + 1).

Words and Sentences

- Joiners: word, sentence
- Accessors: first, last, butfirst, butlast
- Misc: item, count

Input and Output

(last (first (butfirst (butlast '(123 456 789)))))

(butfirst (last (butfirst '(123 456 789))))

(butfirst (butfirst (butfirst '123)))

Quote Confusion

(define w 'hello)

(sentence (quote (sentence w w)) w) → ?

(sentence (word w w))
→ ?

(word unbound-variable)
→ ?

(quote unbound-variable)
→ ?

(quote 'unbound-variable)) → ?

Sample Question #2

Write procedure that given a word returns the piglatin.

Usage:
(piglatin 'hello) → ellohay
(piglatin 'apple) → appleway

Conditionals

- Any value that is not #f is considered true.
- 'and' stops executing when it finds the first false value and returns false. Otherwise it evaluates each argument.
- 'or' stops executing when it finds the first true value then returns it. Otherwise it evaluates each argument.

What do these calls return?

(and 1 2 3) → ?

(or #f 'hello unbound-variable) → ?

Predicates

- equal?
- =, <, >, <=, >=
- member?
- and, or, not

Conventions for user defined predicates

'if' and 'cond' Syntax

(if <predicate>
<execute-when-true> <execute-when-false>)

(cond
(<predicate> <body>)
(<predicate> <body>)
....
(else <body>))

Sample Question #3

```
(define (mystery a b c)
    (cond
        (c (a b c))
        ((and a b c) 'hello)
        ((or a b c) (- (a 3 b)))
        (else
            (and (not (and a b) c))) ) )
```

- Write a procedure call to mystery that will produce the output:

1
-1
#f
hello

Data Abstraction

- Hides implementation from the user
- Provide accessors to retrieve information rather than having the user write it
- Allows the implementation to be changed without changing the use of the data

Case Study

- Differences between version 1 and version 2
- Thought process
- Data abstraction