

# The Patterns You Can't {See, Refactor}

Lisp Macros: Dizzying Heights

# Kyle Burton

## Relay Network

### We are Hiring!

# My Own Personal Blub Paradox

# Code That Writes Code

## Compiler that targets Lisp

### In Lisp

### Without 'eval'

# !Refactorable

If, for, while

class, function, var, package

try, catch, throw

# Special Powers

Introduce Bindings

Defer Execution

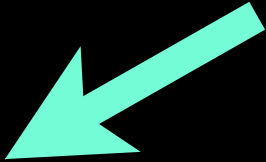
Re-order computation

```
(defmacro when [test & body]  
  `(if ~test  
      (do  
        ~@body)))
```

```
(when (enemy-in-sight)  
  (launch-missiles)  
  (disallow-mineshaft-gap))
```

# Clojure: ..

```
java.util.Calendar.getInstance()  
  .getTimeZone()  
  .getDisplayName();
```



```
(.. (java.util.Calendar/getInstance)  
   getTimeZone  
   getDisplayName)
```



# Closure: $\rightarrow$

$x(y(z(5)))$ ;

$(\rightarrow 5\ z\ y\ x)$

# Clojure: $\rightarrow$

( $\rightarrow$

```
["sort" "us" "by" "length" "please"]  
(map (fn [s] [(count s) s]))  
(sort (fn [[l1 _] [l2 _]] (- l1 l2)))  
(map second))
```

# Common Lisp

```
(defun ptest (names)
  (pcond:pcond
   ((and (:pat ((?first ?last) . ?rest)
              names)
          (:re "(.+)"o" ?first (before-o)))
    (format t "before-o: ~a~&" before-o)
    (ptest ?rest))))
```

```
(ptest '(("George" "Washington")
          ("John" "Adams")))
```

=> before-o: Ge

=> before-o: J

# Scheme: FSM

```
(define m
  (automation init
    (init : (c -> more))
    (more : (a -> more)
           (d -> more)
           (r -> end))
    (end : )))
```

```

(define-syntax automation
  (syntax-rules (: ->)
    ((_ init-state
      (state : (label -> target) ...)
      ...))
    (letrec
      ((state
        (lambda (stream)
          (cond
            ((empty? stream) #t)
            (else
             (case (first stream)
               ((label) (target (rest stream)))
               ...
               (else #f)))))))
      ...))
    init-state))))

```

Power! Power! Power!

Both Ends Are Sharp

Macros Change Semantics

(Do Not Feed After Midnight)

# Thank You!

[kyle.burton@gmail.com](mailto:kyle.burton@gmail.com)

[kburton@relaynetwork.com](mailto:kburton@relaynetwork.com)

[@kyleburton](#)

<http://relaynetwork.com>

<http://github.com/kyleburton/redsnake2013>



# More Info

[http://en.wikipedia.org/wiki/Schwartzian\\_transform](http://en.wikipedia.org/wiki/Schwartzian_transform)

<http://c2.com/cgi/wiki?BlubParadox>

<http://paulgraham.com/avg.html>

<http://stackoverflow.com/questions/267862/what-makes-lisp-macros-so-special>

<http://www.gigamonkeys.com/book/>

<http://hipster.home.xs4all.nl/lib/scheme/gauche/define-syntax-primer.txt>