

What I Hate Most About Scheme And What I'm Doing About It

Alexey Radul

Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology

Boston Lisp User Group, Mar 31st, 2008

Outline

- 1 What I Hate Most About Scheme
- 2 What I'm Doing About It

Outline

- 1 What I Hate Most About Scheme
- 2 What I'm Doing About It

Scheme makes my work unbearable

Scheme is so awesome

Scheme is so awesome
that I can't bear to program
in anything else

... but Scheme's library
support is so awful

... but Scheme's library
support is so awful
that I can't bear to program
in Scheme either!

Scheme > Foo Scheme

Outline

- 1 What I Hate Most About Scheme
- 2 What I'm Doing About It

Here's my chain of reasoning:

Why are there few good libraries?

Because they are
hard to create

Why are good libraries hard to create?

Partly because there is no good
distribution mechanism

Partly because there is no good distribution mechanism

No common module system

Partly because there is no good distribution mechanism

No common module system

No package manager

Partly because portable
code is hard to write

Why is portable code hard to write?

Partly because there are
few good libraries

Partly because there are few good libraries

Yes, this is a recursive problem

Partly because different
implementations will
have different bugs

So, from the bottom up,

A test suite for conformance to R5RS

(Revised⁵ Report on the Algorithmic Language Scheme)

The usual unit testing thing

Open a pipe to a Scheme
Send it a form
Check that it prints what I expect
Lather, rinse, repeat

```
( * 4 5 6 )      ==> 120
(min 4 8)        ==> 4
((lambda (x) (+ x x))
 4)              ==> 8
```

This is not enough

`(sqrt 4)` \implies 2 ?

`(sqrt 4)` \implies 2.0 ?

This is amazingly common

This is amazingly common

Explicit options in the Report
Varied interpretations
Bugs

So I also catalog
implementation choices

```
(sqrt 4) ==>>>  
(choice-cond  
  (exact-sqrt 2)  
  ((not exact-sqrt) 2.0))
```


Status