

Class 34: Implementing LISP

Held: Monday, 25 April 2011

Summary: Today we continue to explore the techniques used to implement Lisp.

Related Pages:

- EBoard.
- Reading: Graham, Paul. (2002). The Roots of Lisp..

Notes:

- Easter weekend consumed my time. I'm not sure when I'm getting grades to you.
- Those of you who missed one assignment may make up for it by implementing a simple LISP interpreter in C (basically, writing the parser and the N or so key functions). For the parser, you should implement `quote` but need not implement the shorthand of `'`.
- Reading for Wednesday: Trail: The Reflection API, available at <http://download.oracle.com/javase/tutorial/reflect/>.
- Reminder: Projects start Friday.
- Due to prereg and College committee work, I need folks to sign up for appointments. Times at <http://www.cs.grinnell.edu/~rebelsky/Schedule/signup.2011S.txt>. Signup via email.
- CS Picnic: Friday, May 6, 2011.
- EC for Thursday's Bites to Bits.
- EC for Friday's CS table.

Overview:

- Detour: Quoting.
- `eval`, continued.
- Flaws in `eval`.
- Graham's Claims.
- Implementing Lisp.

Quoting

- A few of you asked why we didn't do more quoting in CSC151. Here's the example I tend to give of how quoting confuses students.
- What do you expect for each of the following:
 - `(symbol? (car '('a))`
- `(caar '('a))`

Eval, Continued

- We'll continue our translation of `eval` into pseudo-Scheme.
- The legendary `label`.

Flaws

- Graham mentions two significant flaws in McCarthy's definition.
- What are they?
- How hard will they be to resolve?
- Sam's contrived scoping example

```
((lambda (x)
  ((lambda (f x) (f x))
   (lambda (y) (cons x y))
   a))
 b)
```

vs.

```
((lambda (x)
  ((lambda (f z) (f z))
   (lambda (y) (cons x y))
   a))
 b)
```

Detour: Graham's Claims

- Graham claims that other languages ("median languages") are getting closer to Lisp.
- Graham also claims that `eval` is central to what makes Lisp Lisp.
- What does he mean by these claims.

Implementing Lisp

- What does all of this say about the ease or difficulty of implementing Lisp?
- In particular, what do you need to write (in assembly or C or ...)?