CSE 341: Programming Languages

Autumn 2008
Oct 24 — Macros

Topics

- What are macros and what do they mean?
  - Why do they have a bad reputation?
- Scheme’s macro system and hygiene
  - Free variables in macros
  - Bound variables in macros
  - Why hygiene is usually what you want
- What macros are good and not good for

### Macros

To oversimplify, a macro is just a rule for rewriting programs prepass to evaluation. So it's very syntactic.

The "level" at which macros are defined affects their usefulness:

- “Sublexical” e.g.: Replace \texttt{car} with \texttt{hd} would turn \texttt{car t}
  - No macro system does this; so macro-expander must how to break programs into tokens.
- “Pre-parsing” e.g.: Replace \texttt{add}(x, y) with \( x + y \) (where stand for expressions) would turn \texttt{add}(x, y) \* z into \( x \)
  - Some macro systems are this “dumb” (i.e., token-based)
  - Macro writers use more parens than Schemers.
- “Pre-binding” e.g.: Replace \texttt{car} with \texttt{hd} would turn \texttt{(1 0)} \texttt{[car 1]} \texttt{hd} into \texttt{(let* ((hd 0) (hd 1)) hd)}
  - Few macro systems let bindings shadow macros; Sch

### The bad news

- Macros are very hard to use well.
- Most macro systems are so impoverished they make it hard to ameliorate shortcomings in the underlying language.

But:

- Macros have some good uses
- Scheme has a very sensible, integrated macro system
- So let’s do macros justice for the day.
Hygiene

A “hygienic” macro system:
- Gives fresh names to local variables in macros at each u
  macro
- Binds free variables in macros where the macro is define

Without hygiene, macro programmers:
- Get very creative with local-variable names
- Get creative with helper-function names too
- Try to avoid local variables, which conflicts with predict:

Why macros

Non-reasons:
- Anything where an ordinary binding would work just as well
- Including manual control of inlining.

Reasons:
- Cosmetics
- “Compiling” a domain-specific language
  - But error messages a tough issue
- Changing evaluation-order rules
  - Function application will not do here
- Introducing binding constructs
  - A function here makes no sense