A little bit of Lisp

Introduction to Artificial Intelligence
CSCE 476-876, Spring 2002
www.cse.unl.edu/~S02-476-876

Read LWH: Chapters 1, 2, 3, and 4.
Friday, Jan 25 2002: ask your questions on Lisp/xemacs.

Berthe Y. Choueiry (Shu-we-ri)
choueiry@cse.unl.edu, (402)472-5444

Features of Lisp

1. Interactive: interpreted and compiled
2. Symbolic
3. Functional
4. Second oldest language but still 'widely' used (Emacs, AutoCad, Yahoo Store, Orbitz, etc.)

Software/Hardware

- We have Allegro Common Lisp (by Franc Inc.): alisp and mlisp
- There are many old and new dialects (CormanLisp, Kyoto CL, LeLisp, CMU CL, SBCL, ECL, OpenMCL, CLISP, etc.)
- There have also been Lisp machines (Symbolics, Connection Machine, IT Explorer, others?)

Lisp as a functional language

`(function-name arg1 arg2` *`etc`*`)`

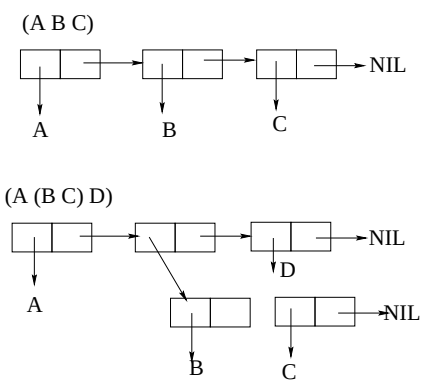
1. Evaluate arguments
2. evaluate function with arguments
3. return the result

Functions as arguments to other functions:
`(name2 (name1 arg1 arg2` *`etc`*`) arg3 arg2` *`etc`*`)`

Symbolic language

- Atoms: numeric atoms (numbers), symbolic atoms (symbols)
  Each symbol has: print-name, plist, package, symbol-value, symbol-function

- Lists:



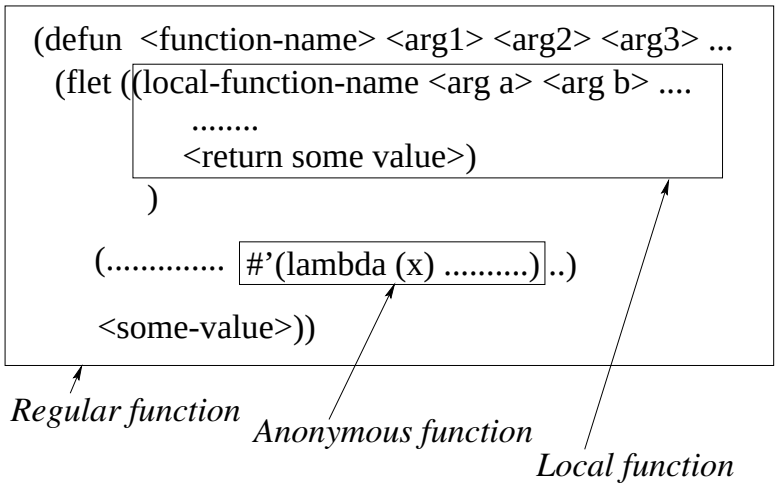
Symbolic expressions: symbols and lists

More constructs

- Data types:
  atoms and lists, packages, strings, structures, vectors,
  bit-vectors, arrays, streams, hash-tables, classes (CLOS), etc.
  NIL, T, numbers, strings: special symbols, evaluate to self
- Basic functions:
  `first (car), rest (cdr), second, tenth`
  `setf`: does not evaluate first argument `cons, append, equal`,
  operations on sets, *etc.*
- Basic macros:
  `defun, defmacro, defstruct, defclass, defmethod,`
  `defvar, defparameter`

- Special forms:
  `let`, `let*`, `flet`, `labels`, `progn`,
- Predicates:
  `listp`, `endp`, `atom`, `numberp`, `symbolp`, `evenp`, `oddp`, *etc.*
- Conditionals:
  `if <test> <then form> <else form>`,
  `when <test> <then form>`,
  `unless <test> <else form>`,
  `cond`,
  `case`
- Looping constructs:
  `dolist`, `dotimes`, `do`, `mapcar`, `loop`,
- Lambda functions

A really functional language



`defun, flet/labels, lambda`

What makes Lisp different?

Paradigms of AI Programming, Norvig

- Built-in support for lists
- Dynamic storage management (garbage collection!)
- Dynamic typing
- First-class functions (dynamically created, anonymous)
- Uniform syntax
- Interactive environment
- Extensibility