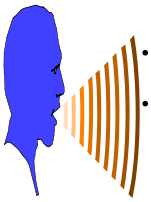
EEL5840: Elements of Machine Intelligence

Subjugator 2006

Announcements

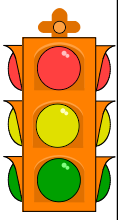


- Today's handouts on www:
  - > Outline Class 5
- Web Site
  - > Today's handouts
  - > www.mil.ufl.edu/eel5840
  - > Software and Notes
  - > XLISP Documentation

EEL5840: Elements of Machine Intelligence

Subjugator 2006

Today's Menu



- Introduction to the AI Language LISP
- LISP
  - > Chapter 2 Basic LISP Primitives
  - > Chapter 3 Procedure Definition & Binding
  - > Chapter 4 Predicates & Conditionals

EEL5840: Elements of Machine Intelligence

Subjugator 2006

Why LISP?

- http://www.paulgraham.com/lisp.html

EEL5840: Elements of Machine Intelligence

Subjugator 2006

The AI Language LISP

- LISP - LISt Processing Invented in the late 50's by John McCarthy at MIT on an IBM 704 computer.
- LISP is about symbolic processing, i.e., symbol manipulation is treating the binary quantities inside the computer like the words and sentences of a language. The words in LISP re called *atoms*. The sentences are called *lists*. Collectively atoms and lists are called *symbolic expressions* (*s-expressions*) or *SEX* for short.
- Examples:

```
(arroyo     (professor ece)
            (degree    phd)
            (area      (ce robotics) ) )
(trip       (gainesville  tallahassee  150)
            (tallahassee  perry        50)
            (perry gainesville         100) )
```

EEL5840: Elements of Machine Intelligence

Subjugator 2006

The AI Language LISP

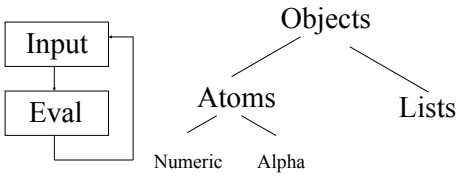
- Uses of LISP
  - > Expert Problem Solvers
  - > Commonsense Reasoning
  - > Learning
  - > Natural Language Interfaces
  - > Education and Intelligent Support Systems
  - > Speech and Vision
  - > The premier symbolic processing language is Common LISP
- Myths
  - > LISP is slow
  - > LISP programs are big
  - > LISP is hard to learn
  - > LISP is hard to debug & read because all those parentheses

EEL5840: Elements of Machine Intelligence

Subjugator 2006

The AI Language LISP

- Tutorial Introduction to LISP
  - > XLISP is available from stat.umn.edu by David Betz & Luke Tierney
  - > We have the latest 16-bit & 32-bit version on www.mil.ufl.edu/eel5840
  - > Install using winzip, pkunzip or in a fresh directory by running the self-extracting file.
  - > Go to the XLISP-STAT resources link to obtain additional information including a manual, the UNIX version and the MAC version.



EEL5840: Elements of Machine Intelligence

Subjugator 2006

LISP Lab 1

Predicate Functions:

(*atom* sex)	t if sex is an atom
(*null* sex)	t if sex is nil or ()
(*eq* sex1 sex2)	t if sex1==sex2 (identical)
(*equal* sex1 sex2)	t if sex1=sex2
(*zerop* sex1)	t if sex=0
(*numberp* sex)	t if sex is a number
(*symbolp* sex)	t if sex is a symbolic atom
(*listp* sex)	t if sex is a list
(*member* sex lis)	nil if sex is not a member of lis

EEL5840: Elements of Machine Intelligence

Subjugator 2006

LISP Lab 1

LIST Functions:

(*car* lis)	returns the 1st sex of list "lis", same as *first*
(*cdr* lis)	returns the list "lis" with the 1st sex removed, same as *rest*
(*car* (*cdr* lis))	same as *second*, i.e., 2nd sex of the list "lis"
(*list* sex1 sex2)	returns the list (sex1 sex2)
(*cons* sex1 lis)	makes sex1 the 1st element of the list "lis"
(*append* lis1 lis2)	a new list with all the elements of lis1 followed by all the elements of lis2

(defun fname (argument-list) <(forms)>)

(*pprint (function-lambda-expression* #'fname))

EEL5840: Elements of Machine Intelligence

Subjugator 2006

LISP Lab 2

Predicate Functions:

(*integerp* sex)	t if sex is an integer
(*floatp* sex)	t if sex is a floating point number
(*and* sex1 sex2)	t if sex1 and sex2 are both true
(*or* sex1 sex2)	t if sex1 or sex2 or both are true
(*not* pred)	nil if pred=t or non-nil, t if pred=nil

User-Defined Functions:

(*cond* <($clause_1$) >…<($clause_n$)>) returns the evaluated action
from the 1st non-nil predicate or nil
where ($clause_i$) ⇒ (predicate-form action-form)

(*defun fname* (argument-list) <(forms)>)

(*pprint* (*function-lambda-expression* #'fname))

Recursive Function Definitions

EEL5840: Elements of Machine Intelligence

Subjugator 2006

See LISP Tutorial 1

The End!