A Few Comments on AI

- Knowledge Representation and Search
- If a computer only has the right knowledge, representation of that knowledge, and way of indexing that knowledge then it can be intelligent
  
  Does this idea work?

Physical Symbol Hypothesis

- Intelligent activity in either human or machine requires [Newell & Simon]:
  - Symbol patterns to represent significant aspects of a problem domain
  - Operations on these patterns to generate potential solutions to problems
  - Search to select a solution from among these possibilities

Representation Schemes

- Schemes should be [page 36 in Luger]:
  - Expressive: The scheme must be adequate to express all necessary information
  - Efficient: Support efficient execution of the resulting code
  - Natural: Provide a natural scheme for expressing the required knowledge

Representation Schemes

- Where a good representation is available, the solution to the problem may be easy.
- Sometimes it’s hard to have a good representation – just imagine a representation for commonsense knowledge in a natural language system!
- Representation may make a big difference in a programming language.

An example

- Task Description
  - To write a program that finds, for a given phone number, all possible encodings by words, and prints them. A phone number is an arbitrary(!) string of dashes -, slashes / and digits. The dashes and slashes will not be encoded. The words are taken from a dictionary which is given as an alphabetically sorted ASCII file (one word per line).
- Participants: 14 programmers (ave. experience: ~ 7 yr)
- Biggest experimental flaw: subjects self selected

Mapping

- The following mapping from letters to digits is given:

  E | J N Q | R W X | D S Y | F T | A M |
  0 |     1   |   2   |   3   |  4  |  5  |
  C I V | B K U | L O P | G H Z
  6   |   7   |   8   |   9   

RIT Computer Science Dept.
The Results

- **Using Lisp**
  - Time (hr): 2 to 8.5, ave: 5
  - Lines of Code: 51 to 182
  - Run Time (median): 30 seconds
- **Using C/C++**
  - Time (hr): 3 to 25, ave: 11
  - Lines of Code: 107 to 614, ave: 277
  - Run Time (median): 54 seconds

- **Using Java**
  - Time (hr): 4 to 63, ave: 9
  - Lines of Code: 107 to 614, ave: 277

- Quickest C/C++ program ran faster than the quickest Lisp program

Lisp

- Stands for List Processing
- Developed by John McCarthy in late 1950's
- Strengths include
  - List processing
  - Symbol manipulation
  - Data and programs have the same syntactic form, so functions can be created and run while the program is executing

A show of hands

- How many have used Lisp?

Learning Lisp

- See my web site for on-line tutorials and information
- [handout given in class]

The Basics

- "No doubt about it, Common Lisp is a big language." - Guy Steele
  - 622 built-in functions (in one pre-ANSI CL)
  - 86 macros
  - 27 special forms
  - 54 variables
  - 62 constants
- C++ has around 48 reserved words
- Lisp includes the kitchen sink!

But…

- We won’t always be using Lisp…
  - We’ll be using emacs Lisp (eLisp)
  - If you have emacs on your computer, then you have eLisp
- We will also use CMUCL
  - CMU Common Lisp
If you’re writing an interpreter…

- Why use Lisp?

Abuse Uses Lisp as its Scripting Language

Richard Gabriel

- The story behind “worse is better”

Example Code

```lisp
(defun rest (a-list)
  "rest gives you all of the list, but the first element."
  (cdr a-list))
```

More Examples

- Check out Lisp in the Great Language Shootout:

  - What’s wrong with these examples?

Why Lisp?

- See the article on:
  - [http://www.paulgraham.com/icad.html](http://www.paulgraham.com/icad.html)