MASSACHUSETTS INSTITUTE OF TECHNOLOGY
Department of Electrical Engineering and Computer Science
6.034 Artificial Intelligence, Fall 2001
**Recitation 1**, September 7

---

**Remembrance of Schemes Past; What is AI?; What is I?** Prof. Robert C. Berwick

**Agenda**

1. Administrivia: running the course
2. ReMEMbrances of Schemes Past: Stimulus and response; the AWP principle
3. Simple ideas can be powerful: Generate and Test
4. What is intelligence?
    a. How is knowledge structured? (REPRESENTATION)
    b. How do we recognize alternatives? (MATCHING)
    c. How do we SEARCH among alternatives?
5. Food for thought: what is AI? What is I?

**1. Administrivia**

Who I am - Bob Berwick; 35-423, x8918, berwick@ai.mit.edu
Easiest to reach me by email!!!
Weekly 6.034 cycle: Tutorials @ beginning of week; Recitations at end; Problem sets & online tutor – due late Tues/Weds.

**2. ReMEMbrance of Schemes Past (à la Récherche du Temps Perdue)**

Stimulus and response: Suppose we want to compute whether an element is a member *somewhere* in a list.

```
(mem 'a '(b c d a))
;Value: #t
```

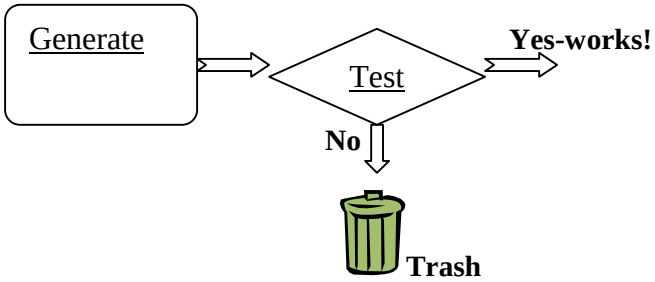
Remember principle AWP! Decompose it into the base case first (# of elements in list *l* 0); then the rest of the list *l*. Now YOU write it. We want this to work for numbers, so you should use `eqv?` as the equivalence test predicate:

```
(define (mem n l)
  (if
                  ...)
```

**3. Simple (trivial) Ideas can be Powerful: Generate and Test**

**Generate:** Enumerate possible solutions
**Test:** Try out the solution and see if it works

Is this a good paradigm for intelligence or not? What makes it good? What makes it not so good? What assumptions about knowledge representation does it make?



**Applications**

Toy: cryptarithmetic.

```
  S E N D
 +M O R E
M O N E Y
```

What should representation be? S=1, E=2, etc. (map letters **S, E, N, D, M, O, R, Y** to digits)

**How should we solve this?** (i) Method of 'constraint analysis'; (2) Method "S"

(S) Try out *all possible* combinations of mappings – write Scheme procedures for this. Duh?

1. Procedure to **generate** all possible assignments of 8 letters to digits 0-9
2. Procedure to **test** whether the assignment "works"

Let's display the **test** procedure first, then see if you can work out the **generate** procedure.

```
(define (test vals)
  (let((s (first vals))
       (e (second vals))
       (n (third vals))
       (d (fourth vals))
       (m (fifth vals))
       (o (sixth vals))
       (r (seventh vals))
       (y (eighth vals)))
    (let ((send  (+ (* 1000 s) (* 100 e) (* 10 n) d))
          (more  (+ (* 1000 m) (* 100 o) (* 10 r) e))
          (money (+ (* 10000 m) (* 1000 o) (* 100 n)(* 10 e) y)))
      (if (= money (+ send more))
          (map cons *vals* vals)))))

(define *vals* '(s e n d m o r y))
(define vals '(9 5 6 7 1 0 8 2))
(test vals)
;Value:  (s . 9) (e . 5) (n . 6) (d . 7) (m . 1) (o . 0) (r . 8) (y . 2)
```

How about the **generate** procedure? We have to figure out a way to produce all combinations of 9-choose-8 lists, e.g., (7 6 5 4 3 2 1 0), (8 6 5 4 3 2 1 0), (9 6 5 4 3 2 1 0)…
`Nvars`= number of letters (8); `l`= low value of digits (0); `h`= high boundary value of digits (10);

To think about this, how could we write a procedure that would generate all the assignments to *one* letter, *s*? Here's what we want **generate** to do:

```
(generate 0 nil)
((s . 6) (e . 8) (n . 5) (d . 3) (m . 0) (o . 7) (r . 2) (y . 1))((s . 9) (e . 5)
 (n . 6) (d . 7) (m . 1) (o . 0) (r . 8) (y . 2))…
```

First try to just generate the numbers, and forget about the 'no duplicate assignments' condition. What goes in the first blank line slot that we can 'ignore' for now?

```
(define (generate k vals)
    (cond ((>= k nvars)
           (____________________))
          (else
           (do ((v l (+ v 1)))
             ((>=  ___  ____ ))
            (generate  _____ (cons_______)))))
```

Now we should add a filtering condition to ensure there are no duplicate assignments of digits to letters.

G&T- Not so toy: VLSI layout; chess; Airport gate routing (Winston's company, `www.ascent.com`). What else can you think of?

**4. What is Intelligence? What is artificial intelligence?**
Is this intelligence?

- Seeing objects: (blindspot), constancy of external world, edges
- Speech perception: twasbrilligandtheslithytoves
- Learning language when you're a child
- Learning language when you're an adult
- Kahneman and Tversky: are Stanford undergrads intelligent?

Which of these can be done now by computer/require intelligence?

- Playing a decent game of table tennis
- Writing an intentionally funny story
- Giving competent legal advice in a specialized area of law
- Translating spoken English into spoken Warlpiri in real time

How about these?

- Are bees intelligent? Conscious?
- Is Deep Blue intelligent? Conscious?
- Are thermostats intelligent? Conscious?

**5. Food for Thought**
Historical sources: the earliest Turing tests for intelligence

Descartes, Condorcet, Vaucausson - 17th century
(Cartesians: *Discours sur la méthode, cinquième partie* )

Even earlier:
"Rabbah said: 'If the righteous desired it, they could [by living a life of absolute purity] be creators.' Rabbah created a person, and sent him to Rabbi Zera. Rabbi Zera spoke to him, but received no answer. Thereupon he said unto him, 'Thou art a creature of the magicians. Return to thy dust.' " (Babylonian *Masechet Sanehedrin*, 65b)