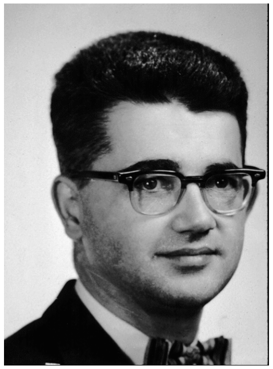
# Lisp

CSCI 334
Stephen Freund

# Lisp

"List Processing"
1960

## John McCarthy



## Good Design

- Motivating Application (AdviceTaker)
- Execution Model
- Theoretical Foundations

## IBM 704

## Factorial

- Two cases:
  - base case: n is 0
  - recursive case: n * fact(n-1)

```
(defun fact (n)
     (cond ((eq n 0) 1)
           (t (* n (fact (- n 1))))))
```

```
(fact 3) -> 6
```

## List Length

- Two cases
  - base case: empty list
  - recursive case: process "car", recurse on "cdr"

```
(defun length (l)
     (cond ((eq l nil) 0)
           (t (+ 1 (length (cdr l))))))
```

## Membership

- Does a list contain a specific atom?

```
(contains 'A '(C A M P)) -> t
(contains 'B '(C A M P)) -> nil
```

```
(defun contains (x l)
     (cond ((eq x nil) nil)
           ((eq x (car l)) t)
           (t (contains x (cdr l)))))
```