Lecture 18: Changing State

Yesterday’s Nobel Prize in Physics

Charles K. Kao
Standard Telecommunication Laboratories, United Kingdom

Willard S. Boyle
Bell Labs, New Jersey

George E. Smith

Charge-Coupled Device (CCD)

Sloan Digital Sky Survey (1998): array of 30, ~4Mpixel CCDs

Moving Collected Charge

Photocell: photoelectric effect (Einstein’s 1921 nobel prize): photons hit silicon plate and knock out electrons (more light, more electrons)

Silicon substrate: not powered, conducts well powered, does not conduct

CCDs Today
cs1120 Story so Far

Computer Science: cs1120 so far

- How to describe information processes by defining procedures (Chapters 3, 4, 5)
  - Programming with procedures, lists, recursion
- How to predict properties about information processes (Chapter 6, 7)
  - Predicting how running time grows with input size
- How to efficiently implement information processes (not much on this)
  - Chapter 3 (rules of evaluation)
  - Chapter 6 (machines)

From Chapter 1/Lecture 1:

The Liberal Arts

- Grammar: study of meaning in written expression
- Rhetoric: comprehension of verbal and written discourse
- Logic: argumentative discourse for discovering truth
- Arithmetic: understanding numbers
- Geometry: quantification of space
- Music: number in time
- Astronomy

Trivium (3 roads)
- Grammar
- Rhetoric
- Logic

Quadrivium (4 roads)
- Arithmetic
- Music
- Geometry
- Astronomy

cs1120 Upcoming

- How to describe information processes by defining procedures
  - Programming with state (Ch 9), objects (Ch 10), languages (Ch 11)
- How to predict properties about information processes
  - Are there problems which can’t be solved by algorithms? (Ch 12)
  - What is the fastest process that can solve a given problem? (Ch 13)
- How to efficiently implement information processes
  - How to implement a Scheme interpreter (Ch 11)

Liberal Arts Checkup

- Grammar: BNF, RTN, rules of evaluation for meaning
  - Not much yet...interfaces between components (PS6-9), program and user (PS8-9)
  - Rules of evaluation, if, recursive definitions
- Trivium
  - Not much yet...wait until November
- Quadrivium
  - Curves as procedures, fractals (PS3)
  - Yes, listen to "Hey Jude!"
  - Soon: read Neil deGrasse Tyson's essay
Introducing Mutation

Evaluation Rule 2: Names

A name expression evaluates to the value associated with that name.

> (define two 2)
> two
2

This has been more-or-less okay so far, since the value associated with a name never changes...

Names and Places

- A name is not just a value, it is a place for storing a value.
- define creates a new place, associates a name with that place, and stores a value in that place

(define x 3)  x: 3

Bang!

set! ("set bang") changes the value associated with a place

> (define x 3)  x: 7
> x
3
> (set! x 7)
> x
7

set! should make you nervous

> (define x 2)
> (nextx) 3
> (nextx) 4
> x 4

Before set! all procedures were functions (except for some with side-effects). The value of (f) was the same every time you evaluate it. Now it might be different!

Defining nextx

(define (nextx)
(set! x (+ x 1))
x)

(define nextx
(lambda ()
(begin
(set! x (+ x 1))
x))))
Evaluation Rules

> (define x 3)
> (+ (nextx) x)
7
or 8
> (+ x (nextx))
9
or 10

DrScheme evaluates application subexpressions left to right, but Scheme evaluation rules allow any order.

Mutable Cons Cell

mcons – creates a mutable cons cell
(mcdr m) – second part of a mutable cons cell

(set-mcar! p v)
Replaces the car of mutable cons p with v.

(set-mcdr! p v)
Replaces the cdr of mutable cons p with v.

These should scare you even more then set!!

(set-mcar! pair 0)
> (mcar pair)
0
> (mcdr pair)
2
> (set-mcdr! pair 1)
> pair
(0 . 1)

Impact of Mutation

• We will need to revise our evaluation rules for names and application expressions: substitution model of evaluation no longer works since values associated with names change
• We need to be much more careful in our programs to think about when things happen: order matters since values change
Charge

- PS5: posted now, due next Wednesday. 
  Monday, 19 October

- Read Chapter 9

- Friday: return Exam 1, Revising our Evaluation Rules to handle mutation