

Class 14: Recursion with Natural Numbers (1)

Held: Tuesday, February 13, 2007

Summary: Today we consider recursion over a different domain, that of the natural numbers.

Related Pages:

- EBoard.
- Lab: Numeric Recursion.
- Reading: Numeric Recursion.

Notes:

- HW7 is now due on Friday. I have not completely updated the syllabus to reflect this change.
- We will be continuing the numeric recursion lab tomorrow, so you might choose to reread Numeric Recursion.
- You should also do a new reading on Writing Recursive Procedures.
- Because of a school delay, I may be late for class today.

Overview:

- Writing Recursive Procedures.
- Short Introduction to Numeric Recursion.
- Lab.

Patterns of Recursion

- While we've seen and written a variety of examples of direct recursion, they typically have the following form:

```
(define recursive-proc
  (lambda (params)
    (if (base-case-test)
        (base-case params)
        (combine (partof params)
                  (recursive-proc (simplify params)))))))
```

- In many cases, the combination ends up being a choice between two activities. In those cases, we might write:

```
(define recursive-proc
  (lambda (params)
    (cond
      ((base-case-test)
       (base-case params))
      ((special-case-test)
       (combine (partof params)
                (recursive-proc (simplify params))))
      (else
       (recursive-proc (simplify params))))))
```

- For lists, the simplification was almost always “take the cdr” and the “part-of” was almost always “take the car”.

Recursion with Numbers

- While most of the recursion we’ve been doing has used lists as the structure to recurse over, you can recurse with many different kinds of values.
- It is fairly common to recurse using numbers.
- The natural base cases for integers are when you hit 0 or when you hit 1.
- The natural simplification step for recursive procedure using numbers calls typically involves subtracting 1 from the argument.
 - Other simplifications, such as dividing in half, are also possible.

Lab

- Do the lab.

Copyright © 2007 Samuel A. Rebelsky. This work is licensed under a Creative Commons Attribution-NonCommercial 2.5 License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc/2.5/> or send a letter to Creative Commons, 543 Howard Street, 5th Floor, San Francisco, California, 94105, USA.