Class 16: Anonymous Procedures

Held: Friday, February 19, 2010

Summary: We revisit procedures in Scheme. We consider a variety of techniques for writing anonymous procedures, procedures without names.

Related Pages:
- EBoard.
- Lab: Anonymous Procedures.
- Reading: Anonymous Procedures.

Notes:
- No reading for Monday! If all goes well, we’ll go over the exam. If not, we’ll do something else.
- Are there questions on Assignment 4
- Crud is going around campus. Please take care of yourselves. Get enough sleep. Eat well. Wash.
- EC (academic) for today’s CS table.
- EC (academic) for next Wednesday’s talk on Open Information Culture (4:30 in Burling).
- EC (academic) for next week’s convocation (Catherine Keller).

Overview:
- What is a procedure?
- Describing procedures.
- Anonymous procedures.
- Anonymous procedures through lambda expressions.
- Other kinds of anonymous procedures.

What is a procedure?
- One of the basic building blocks of programming.
- Essentially, an encapsulated algorithm
  - Takes some inputs
  - Generates some outputs

Describing Procedures
- Traditionally (define name-of-procedure (lambda (parameters) body))
- When applying one of these, we effectively substitute the arguments we call the procedure on for the parameters in the body, and then evaluate the body.
Anonymous Procedures

- Procedures without names.
- Useful when we just need a procedure once
  - Saves the need to define it (code is more concise)
  - Saves the need to document it
  - Makes code work on its own
- In Scheme, wherever you write a procedure name, you can also write the corresponding lambda expression.
- Very useful with map
- Examples
  - \((\text{lambda } (x\ y)\ (*\ (+\ x\ y)\ (-\ x\ y)))\ 2\ 3)\)
  - \((\text{map } (\text{lambda } (x)\ (*\ 5\ (+\ 1\ x)))\ (\text{iota}\ 10))\)

Other ways to build anonymous procedures

- Sectioning!
- Composition!