MASSACHVSETTS INSTITVTE OF TECHNOLOGY
Department of Electrical Engineering and Computer Science
6.001—Structure and Interpretation of Computer Programs
Spring 2006

Recitation 2
More Scheme: λ and if

Scheme

1. **Special Forms**

   (a) *if* - `(if` *test consequent alternative*`)`
   If the value of the test is not false (#f), evaluate and return the value of the consequent, otherwise evaluate the alternative.

   (b) *lambda* - `(lambda` *parameters body*`)`
   Creates a procedure with the given parameters and body. Parameters is a list of names of variables. Body is one or more scheme expressions. When the procedure is applied, the body expressions are evaluated in order and the value of the last one is returned.

   (c) *cond* - `(cond (`*test consequent*`) (`*test consequent*`) ... (else` *alternative*`))`
   Alternative to if when there are more than two cases. The value returned is the consequent where the first test evaluates to true (anything but #f). If no tests are true, evaluate and return the alternative, if any. The alternative `else` is optional.

Problems

1. Evaluate the following expressions (assuming `x` is bound to 3).

   (a) `(if #t (+ 1 1) 17)`
   (b) `(if #f #f 42)`
   (c) `(if (> x 0) x (- x))`
   (d) `(if 0 1 2)`
   (e) `(if x 7 (7))`

2. Evaluate the following expressions (assume x is still bound to 3):

   (a) `(lambda (x) x)`

   (b) `((lambda (x) x) 17)`

   (c) `((lambda (x y) x) 42 17)`

   (d) `((lambda (x y) y) (/ 1 0) 3)`

   (e) `((lambda (x y) (x y 3)) (lambda (a b) (+ a b)) 14)`

   (f) `(define y (lambda (x) (+ x 1)))`

   (g) `(y x)`

3. Alyssa P. Hacker is working on a proposal to amend the MIT policy on pets in dorms—she wants to permit dogs in addition to cats. To support her proposal, Alyssa is working with the athletics department to reuse tennis balls as dog toys, though Alyssa still needs to determine how many tennis balls to request in her proposal. Different sized dogs need different numbers, as the larger ones can carry multiple toys at the same time.

   Initially, only four breeds will be allowed on campus, though both puppies and adult dogs of each breed. Chihuahua puppies only need one tennis ball, border collie puppies can handle two, golden retrievers can carry three [1], and a young newfoundland can carry 4. Adult dogs need twice as many toys as their younger counterparts, as they're generally larger and have bigger teeth. Alyssa chooses to represent the puppy breeds by the numbers 1, 2, 3, 4 respectively, and the adults by 5, 6, 7, 8.

   (a) Define a procedure named `adult-breed` which when given a puppy breed returns the corresponding adult.

   (b) Write a procedure `puppy-breed` which when given an adult breed returns the correct puppy breed.

   (c) Write a procedure named `puppy?` which when given a breed returns true if it is a puppy and false otherwise.

[1] Yes, I've really seen this done, though not by a puppy

(d) Write a procedure named `breed-toys` that takes a breed and returns the number of tennis balls required for that dog.

(e) Instead of provisioning for individual dogs, Alyssa wants to order toys by hall, so we'll call the resulting breeds the live on in any given hall a pack. We'll represent a pack as a set of digits corresponding to the breeds present. 817, for example, represents a newfoundland, a chihuaha puppy, and an adult golden retriever. Write a procedure called `empty-pack` that takes no arguments and returns an empty pack.

(f) Write a procedure called `add-to-pack` that takes a pack and returns a new pack with the both the old and new members. For example, `(add-to-pack 15 2) -> 152`.

(g) Write a procedure named `pack-size` which takes a pack and returns the number of dogs in that pack. For example, `(pack-size 237) -> 3`. You might find `quotient` (integer division) useful.

(h) Write a procedure named `pack-toys` which takes a pack and retuns the total number of tennis balls required for the whole pack. The `remainder` procedure may be helpful.

(i) Alyssa needs to plan for the future. Write a procedure called `grown-pack-toys` which takes a pack and returns the number of toys that will be needed once all its members have grown up.