CSE 1729 - Introduction to Principles of Programming — September 12, 2017

Laboratory Assignment 2

Objectives

1. Defining functions in terms of other functions
2. Solve some recursion problems
3. Use the special forms `if` and `cond`

Activities

1. Warm up. Let's define some functions having to do with circles and spheres.
   (a) First thing, let us define a function `pi` (with no parameters) for π, so whenever we need π we simply evaluate `(pi)`.
   We could define it as equal to a constant, as in the following:

   ```
   (define (pi)
      3.14)
   ```

   This is a pretty poor approximation: yours should be better. Hint: Scheme has inverse trig functions `asin`, `acos`, and `atan`; these will give you the angle (in radians) for a given sine, cosine, or tangent value. You should be able to calculate π using one of these.
   (b) The area of a circle with radius r is πr^2. Define a function `(area-of-circle r)` that uses `PI` that you defined as above.
   (c) The surface area of a sphere with radius r is $4\pi r^2$. Define a function for `(surface-area-of-sphere r)` that gives the surface area of a sphere with radius r. This function should use your `area-of-circle` function.
   (d) The volume of a sphere is $\frac{4}{3}\pi r^3$. Define function for `(volume-of-sphere r)` that gives the volume of a sphere with radius r. This function should use your `surface-area-of-sphere` function.

2. The first three values of a particular sequence are $1, 2, 3$. The remaining values in the sequence can be calculated as a function of the three preceding values in the sequence sum of the three preceding values in the sequence as follows:

$$S_n = \begin{cases} 1 & \text{if } n = 1; \\ 2 & \text{if } n = 2; \\ 3 & \text{if } n = 3; \\ S_{n-3} - S_{n-2} + S_{n-1} & \text{otherwise.} \end{cases}$$

   So, the fourth value in the sequence would be $1 - 2 + 3 = 2$. Write a recursive Scheme function `(s n)` that computes the n^{th} value in the sequence.

3. Write a recursive function, named `zeno`, that computes the sum of the first n terms of the following series from Zeno's Dichotomy Paradox, $Z_n = \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \cdots + \frac{1}{2^n}$. You may use the built-in exponentiion function `(expt x n)` which evaluates to x^n.

4. We can determine whether a non-negative number is even using the following recursive definition:

$$\texttt{(even-nn-int? n)} = \begin{cases} \#t & \text{if } n = 0; \\ \#f & \text{if } n = 1; \\ \texttt{(even-nn-int? (- n 2))} & \text{otherwise.} \end{cases}$$

   (a) Write a recursive function, named `even-nn-int?` that determines whether a non-negative number is even or not using the above definition. (Note: do not call it `even?`, as it would clash with an existing function. Don't use the existing `even?`, `odd?`, or `modulo` functions.)

   (b) Using `even-nn-int?`, write a function `even-int?` that determines whether any integer is even.

   (c) Finally, we define `odd-int?` that determines whether any integer is odd.