CS355: Programming Paradigms Lab

Lab 5: Streams

September 23^{rd}, 2024

Q1. Copy and paste the definitions of various stream-related procedures from Moodle. Extend the listing to include `stream-ref`, `stream-enumerate-interval`, `stream-filter` and `stream-map`. Also have a version with the list higher order procedures. Write a few programs (such as the second prime number in a large range) and convince yourself that the stream version runs (much) faster than the list version.

Q2. In many applications using streams, we may end up forcing the same promise many times. The solution is to build promises so that the first time they are forced, they store the value that is computed by evaluating them (i.e., the result is *memoized* within the thunk object representing the promise). Re-implement your definition of `delay` such that it returns a memoizable promise. Test by checking that the Scheme programs you wrote in Q1 produce the same output as before.

Q3. The positive rational numbers, which are ratios of two positive integers a/b, can be enumerated by listing them in order of increasing sums $a + b$, with those numbers having the same sum listed in order of increasing numerator a. Those fractions that are not in lowest terms are omitted from the enumeration. Thus the enumeration begins with $1/1, 1/2, 2/1, 1/3, 3/1, \ldots$. Define a stream `positive-rationals` that contains all the positive rational numbers with numerator and denominator having no common divisors greater than 1. Represent the rational number a/b as `(list a b)`. Test your program by listing the first 20 elements of the stream.