

Jesse Huynh
CS 450
Homework 5

Part 1.

Question 2 a

Apply takes a procedure and applies it to a list. Cons is there to provide a pair or list for Apply to apply the procedure to. We cannot use the provided replacement due to how stream-map is evaluated it would end up providing the same arguments over and over and get stuck in a loop.

Question 2 b

This will not work because apply applies a procedure to a list. Cons helps make sure the arguments are a pair to be input to which stream-map is applied. Without cons, we would be giving 3 arguments to the apply procedure instead of just two, and the arguments for stream-map would not be in proper list form.

Part 2.

Question 1.

	Output stream	a	a-list	pow	Input stream
	()	0	()		(9 8 7 4 3 6 9 1 7)
A-list is empty; consume					
	()	783	(7 8 3)	100	(8 7 4 3 6 9 1 7)
$87 + 83 \geq 100$; consume					
	()	8526	(8 5 2 6)	1000	(7 4 3 6 9 1 7)
$87 + 526 < 1000$; produce					
	(8)	526	(5 2 6)	100	(7 4 3 6 9 1 7)
$87 + 26 \geq 100$; consume					
	(8)	5869	(5 8 6 9)	1000	(4 3 6 9 1 7)
$87 + 869 < 1000$; produce					
	(85)	869	(8 6 9)	100	(4 3 6 9 1 7)
$87 + 69 \geq 100$; consume					
	(85)	9038	(9 0 3 8)	1000	(3 6 9 1 7)
$87 + 38 < 1000$; produce					
	(859)	038	(0 3 8)	100	(3 6 9 1 7)
$87 + 38 \geq 100$; consume					
	(859)	0641	(0 6 4 1)	1000	(6 9 1 7)
$87 + 641 < 1000$; produce					
	(8590)	641	(6 4 1)	100	(6 9 1 7)
$87 + 41 \geq 100$; consume					

	(8590)	6932	(6 9 3 2)	1000	(9 1 7)
87 + 932 >= 1000; consume					
	(8590)	70103	(7 0 1 0 3)	10000	(1 7)
87 + 103 < 10000; produce					
	(85907)	0103	(0 1 0 3)	1000	(1 7)
87 + 103 < 1000; produce					
	(859070)	103	(1 0 3)	100	(1 7)
87 + 3 < 100; produce					
	(8590701)	03	(0 3)	10	(1 7)
87 + 3 >= 10; consume					
	(8590701)	117	(1 1 7)	100	(7)
87 + 17 >= 100; consume					
	(8590701)	1779	(1 7 7 9)	1000	()
Stream is empty, return a-list					
	(85907011779)				

2.

$$2 + 1/3 (3) = 3$$

$$2 + 1/3 (2 + 2/5(3)) = 3.06666667$$

$$2 + 1/3 (2 + 2/5(2 + 3/7(3))) = 3.1047619$$

$$2 + 1/3 (2 + 2/5(2 + 3/7(2 + 4/9(3)))) = 3.12380952$$

$$2 + 1/3 (2 + 2/5(2 + 3/7(2 + 4/9(2 + 5/11(3))))) = 3.13304473$$

$$2 + 1/3 (2 + 2/5(2 + 3/7(2 + 4/9(2 + 5/11(2 + 6/13(3))))) = 3.137487375$$

$$2 + 1/3 (2 + 2/5(2 + 3/7(2 + 4/9(2 + 5/11(2 + 6/13(2 + 7/15(3))))) = 3.13961594$$

$$3. f(x) = 1 / x+3$$

$$= 0x + 1 / x + 3$$

This yields the matrix (0 1)
(1 3)

4.

I wrote this one out by hand, attaching scanned image below.

$$f(x) = \frac{(Ax+B)}{(Cx+D)}$$

$$g(x) = \frac{(ax+b)}{(cx+d)}$$

$$f \circ g = \frac{A\left(\frac{ax+b}{cx+d}\right) + B}{C\left(\frac{ax+b}{cx+d}\right) + D}$$

$$= \frac{\left(\frac{Aax+Ab}{cx+cd}\right) + B}{\left(\frac{Cax+Cb}{cx+cd}\right) + D} = \frac{\left(\frac{Aax+Ab}{cx+cd} + \frac{Bcx+Bd}{cx+cd}\right)}{\left(\frac{Cax+Cb}{cx+cd} + \frac{Dcx+Dd}{cx+cd}\right)}$$

$$= \frac{\left(\frac{Aax+Ab+Bcx+Bd}{cx+cd}\right)}{\left(\frac{Cax+Cb+Dcx+Dd}{cx+cd}\right)}$$

$$= \frac{Aax+Bcx+Ab+Bd}{Cax+Dcx+Cb+Dd} = \frac{x(Aa+Bc) + Ab+Bd}{x(Ca+Dc) + Cb+Dd}$$

$$A = \begin{pmatrix} A & B \\ C & D \end{pmatrix}$$

$$B = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

$$AB = \begin{pmatrix} Aa+Bc & Ab+Bd \\ Ca+Dc & Cb+Dd \end{pmatrix} = \frac{(Aa+Bc)x + (Ab+Bd)}{(Ca+Dc)x + (Cb+Dd)}$$

Problem Discussion:

Problem 1, mult-stream

For this problem, I figured I would try to follow the pseudo code as closely as possible but branch off if I could not figure something out. I began by creating any helper procedures suggested, such as `number->listofdigits` as well as the suggested one in Piazza, `list->stream`. I also made a function to help me get the power based off of the length of the length of a-list. The last helper I made was a function that adds a leading zero to a-list to adjust the length of the new a-list. After that, I tried to copy the pseudocode for `action`, `consume`, and `produce`.

I initially kept having issues with the system expecting <promises> and getting lists instead, which had to do with how my system was passing the lists/streams. I tried implementing an `if` statement to rerun the procedure with `(list-to-stream strm)` but it would not work for some reason, so I ended up using `set!` to change `strm` completely even though this was not suggested. I tried my `if` statement in a few places, inside `mult-stream` as well as inside `action` but I would keep getting the same error which is why I settled on using `set!`. Doing this actually cleared up most of the errors I had, stemming from `stream-cdr` calls. For testing, I really only used the example from the provided texts however once I fixed my `list->stream` issue I was able to pass the `gradescope` tests easily.

Problem 2, pi

I started this problem by following what was stated in the handout, and created constructors and selectors for making a 2x2 matrix and selecting individual elements from it. I then created the procedure to multiply matrices as well as adding them and using `map` on them to add the matrix to each element of a stream. I also added procedures to shift the matrix, and find the floor of the matrix. I then followed the handout as best I could and also asked for a few tips from fellow classmates who pointed me in the right direction without giving me actual answers (like if I was heading in the right direction regarding my helper functions).

The hardest part of this assignment for me was actually understanding what was going on in regards to the math and how the procedure should be running. Once I was able to write it out and do the math by hand, it became a lot easier for me to program it. The second hardest part was figuring out how to shift the matrix properly.

The only real issues I ran into were some problems with parentheses placements which I fixed easily after trying to run the program.