Calculating Fibonacci with `reduce` and `map`

Carlos Tavares

November 5, 2024

Hypothesis 1: Straight Sum with reduce I

Hypothesis: Calculate the Fibonacci function using a straight sum with `reduce`.

Solution:

```
r = reduce (lambda x,y:  x+y, range (0, n+1))
```

Results:

n	0	1	2	3	4	5	6	7	8	9	10
Fibonacci(n)	0	1	1	2	3	5	8	13	21	34	55
Straight Sum	0	1	3	6	10	15	21	28	36	45	55

Table: Comparison of Fibonacci values and results from a straight sum

Conclusion: wrong!

Hypothesis 2: Straight Sum Starting in 1, 1 with `map`

Hypothesis: Use a straight sum starting with values 1, 1 and apply `map` to generate the sequence.

Solution:

```
r = reduce (lambda x,y:  x+y, map (lambda x:  0 if x==0 or
x == 2 else (1 if x == 1 else x), range (0, n+1)))
```

Results:

n	0	1	2	3	4	5	6	7	8	9	10
Fibonacci(n)	0	1	1	2	3	5	8	13	21	34	55
Straight Sum 1,1	0	1	1	4	8	13	19	26	34	43	53

Table: Comparison of Fibonacci values and results from a straight sum

Conclusion: wrong!

Hypothesis 3: Using pairs in reduce and map I

Hypothesis: Use pairs to store past and current results

```
if n == 0:
    r = 0
else:
    r = reduce (lambda x,_:  (x [1], x[1]+x[0]), map
(lambda x:  (0,1) if x==1 else x, range (1, n+1)))[1]
```

n	0	1	2	3	4	5	6	7	8	9	10
Fibonacci(n)	0	1	1	2	3	5	8	13	21	34	55
Pair combination	0	(0,1)	(1,1)	(1,2)	(2,3)	(3,5)	(5,8)	(8,13)	(13, 21)	(21,34)	(34,55)

Table: Comparison of Fibonacci values and results from the combinations of pairs ([1] of each pair holds the result)

Conclusion: Correct!