

Project Euler: Problem 2

Dominic Walden

September 12, 2017

Contents

1 License	1
-----------	---

1 License

Copyright (C) 2016 Dominic Walden.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license can be found here: <https://gnu.org/licenses/fdl.html>.

Each new term in the Fibonacci sequence is generated by adding the previous two terms. By starting with 1 and 2, the first 10 terms will be:

1, 2, 3, 5, 8, 13, 21, 34, 55, 89, ...

By considering the terms in the Fibonacci sequence whose values do not exceed four million, find the sum of the even-valued terms.

It would be fairly easy to generate the sequence of Fibonacci numbers which do not exceed four million, remove from these all the odd members and sum the remaining.

One could also take the sum of every third element (starting from 2).

```
(define fibonacci-stream
  (lambda (n1 n2)
    (cons-stream (+ n1
                   n2)
```

```

                (fibonacci-stream n2
                    (+ n1
                       n2))))))

(define even-stream
  (lambda (stream)
    (if (even? (stream-car stream))
        (cons-stream (stream-car stream)
                      (even-stream (stream-cdr stream)))
        (even-stream (stream-cdr stream)))))

(define even-fibonacci-stream
  (even-stream (fibonacci-stream 1 1)))

(define max-integer-stream
  (lambda (stream max)
    (if (> (stream-car stream)
            max)
        '()
        (cons-stream (stream-car stream)
                      (max-integer-stream (stream-cdr stream)
                                           max)))))

(define stream-sum
  (lambda (stream)
    (if (null? stream)
        0
        (+ (stream-car stream)
           (stream-sum (stream-cdr stream))))))

(stream-sum (max-integer-stream even-fibonacci-stream
                                4000000))

;Value: 4613732

(define every-nth-element-of-stream
  (lambda (stream n)
    (define dispatch
      (lambda (stream n i)
        (if (= i

```

```

        1)
      (cons-stream (stream-car stream)
                   (dispatch (stream-cdr stream)
                             n
                             n))
      (dispatch (stream-cdr stream)
                n
                (- i 1))))))
(dispatch stream
  n
  n)))

(define every-3rd-element-fibonacci-stream
  (every-nth-element-of-stream (fibonacci-stream 1 2)
                               3))

(stream-sum (max-integer-stream every-3rd-element-fibonacci-stream
                                4000000))

```

;Value: 4613730

Plus the 2 which was left out of "every-3rd-element-fibonacci-stream".