The Dinner Party Puzzle

Mr. and Mrs. Smith hosted a dinner party for 9 other couples.

During cocktails, some people shook hands.

> No one shook his or her own hand.
>
> No two people shook hands twice.
>
> No one shook their spouse’s hand.

During dinner, Mr. Smith asked everyone how many hands they shook.

He got back 19 different answers.

How many hands did Mrs. Smith shake?

The Dinner Party Puzzle

Theorem. If there were $n \geq 1$ couples in all then Mrs. Smith shook $n-1$ hands.

Proof.

Base Case: $\boldsymbol{k=1}$. Since spouses do not shake hands, Mrs. Smith shook $k-1=0$ hands.

Inductive Hypothesis: $P(k)$: If k couples, then Mrs. Smith shook $k-1$ hands.

Inductive Step. Suppose $P(k)$ is true for some $k \geq 1$.

Consider the party with $k+1$ couples.

There are $2k+2$ people, and $2k+1$ distinct responses.

The maximum number of handshakes for any one person is $2k$.

The responses are $0,1, \ldots, 2k$

The person with $2k$ handshakes shook hands with everyone, except 0, $2k$, and his/her spouse.

The people with 0 and $2k$ handshakes must be a couple!

Disregarding them, we are left with k couples, and $2k-1$ distinct handshakes.

From $P(k)$ we conclude that Mrs. Smith shook $k-1$ hands from among these k couples.

Total number of handshakes for Mrs. Smith = $k-1+1=k$.

Therefore, we have established $P(k) \rightarrow P(k+1)$.

By PI, the theorem is valid.

A Scheme Function

(define $(mystery\ A\ B)$

$(if\ \ (null?\ \ A)\ \ B$

$\left(cons\ \ (car\ A)\ (mystery\ (cdr\ A)\ B)\right)$

)

)

What is $(mystery$ ‘$(a\ b)$ ‘$(1\ 2\ 3))$?

$\left(cons\ \ a\ (mystery\ (b)\ (1\ 2\ 3))\right)$

$\left(cons\ \ a\ (cons\ b\ (mystery\ \ (\ )\ (1\ 2\ 3))\right)$

$(cons\ \ a\ (cons\ b\ (1\ 2\ 3)))$

$(cons\ \ a\ (b\ 1\ 2\ 3))$

$(a\ b\ 1\ 2\ 3)$

How do we *prove* that $mystery$ always returns the concatenation of lists A and B?

The Concatenation Operator

Let $\circ$ denote the concatenation operation.

Properties of the concatenation operator:

If X, Y are lists then:

1. $(\,) \circ X = X$
2. $(cons\ \ a\ \ X \circ Y) = (cons\ a\ X) \circ Y$

We want to prove that $\forall A, B$: $\quad (mystery\ A\ B) = A \circ B$

Proof that $mystery$ implements concatenation

(define (mystery A B)

$(if\ (null?\ \ A)\ \ B$

$\big(cons\ \ (car\ A)\ (mystery\ (cdr\ A)\ B)\big)$

))

$(\,)\ \circ\ \ X\ = X$

$(cons\ \ a\ \ X \circ Y) = (cons\ a\ X) \circ Y$

Claim: $\forall A, B$: $\quad (mystery\ A\ B)\ =\ A \circ B$

Proof: (by induction on the length of A)

Base Case: length(A) = 0 or $A = (\,)$

$$(mystery\ A\ B) = B = (\,) \circ B = A \circ B$$

Inductive Hypothesis: length(A) = k $\implies$ $(mystery\ A\ B) = A \circ B$

Inductive Step: length(A) = $k + 1$

$$\begin{aligned}(mystery\ A\ B)\ &= \big(cons\ (car\ A)\ (mystery\ (cdr\ A)\ B)\big)\\ &= (cons\ (car\ A)\ (cdr\ A) \circ B)\\ &= \big(cons\ (car\ A)(cdr\ A)\big) \circ B\\ &= A \circ B\end{aligned}$$

The Fibonacci Numbers

The fibonacci numbers $f(n), n \geq 0$ are defined as follows:

$$f(0) = 0$$
$$f(1) = 1$$
$$f(n) = f(n-1) + f(n-2), n \geq 2$$

The sequence of fibonacci numbers:

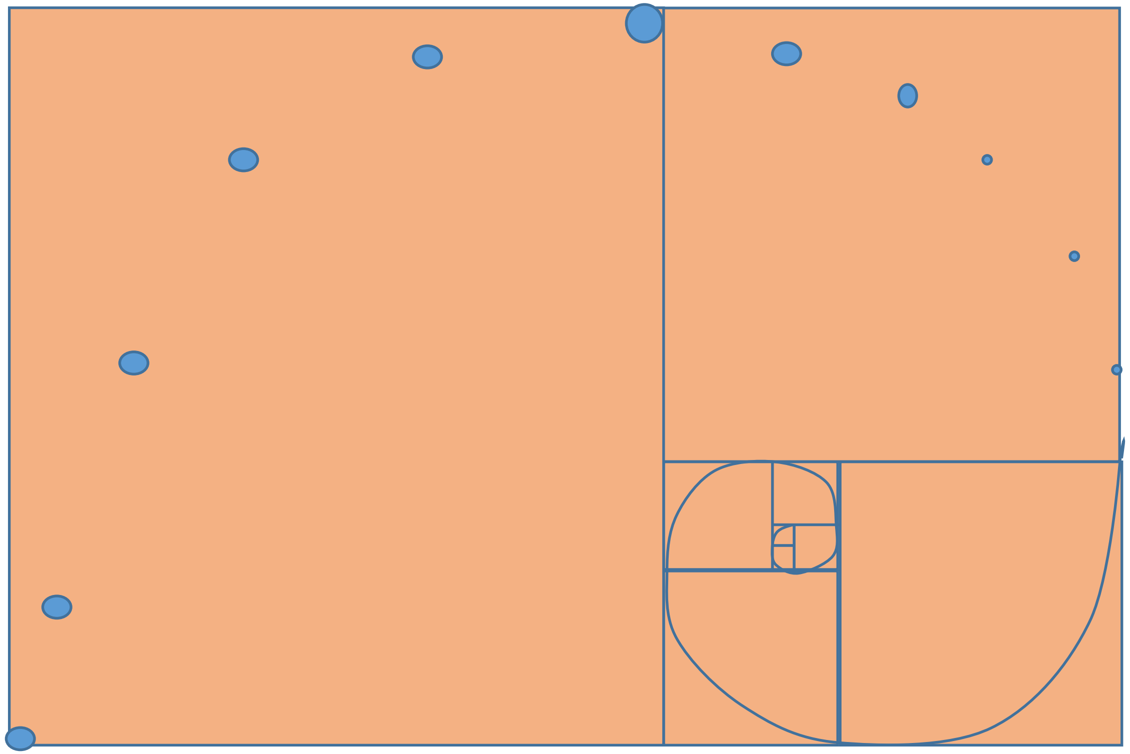
0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, …

Sequence of ratios of consecutive fibonacci numbers:

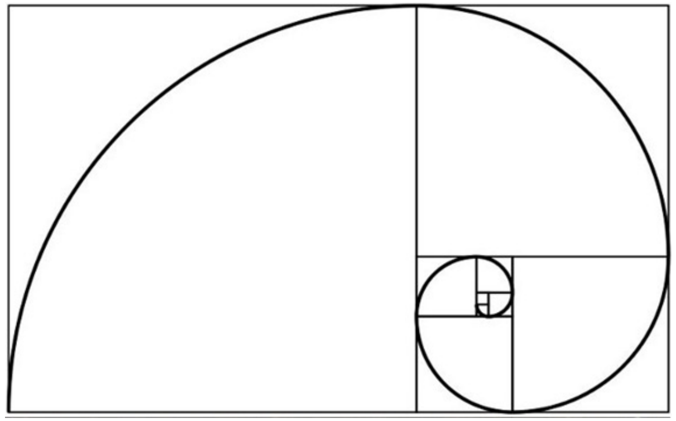
1, 2, 1.5, 1.67, 1.6, 1.625, 1.615, 1.619, 1.618, 1.618, . . . , 1.6180339887498948482…

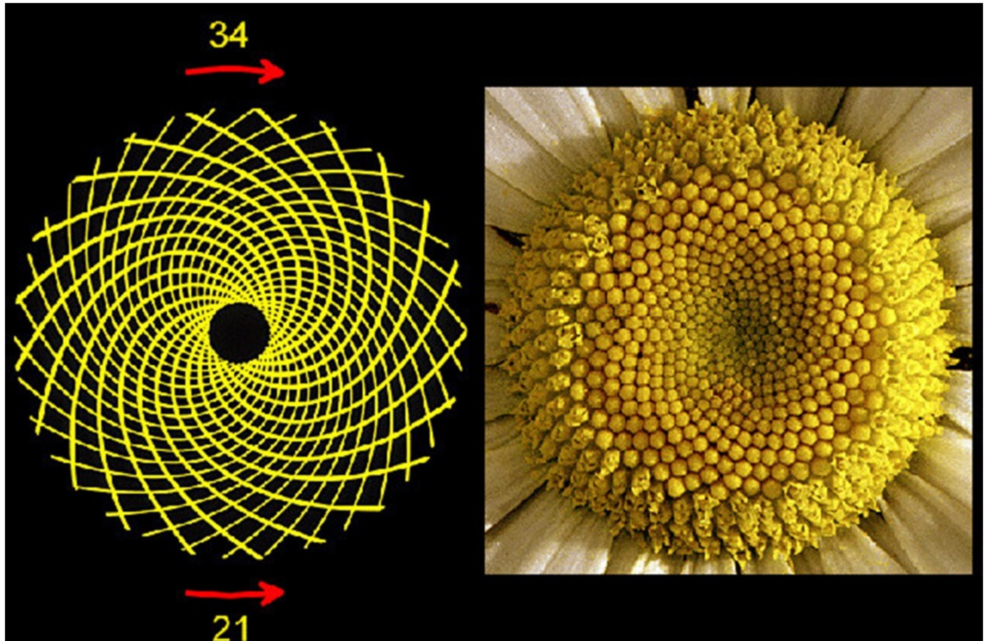
$\frac{f(n)}{f(n-1)} \rightarrow \varphi$ (the golden ratio)

Fibonacci Spirals

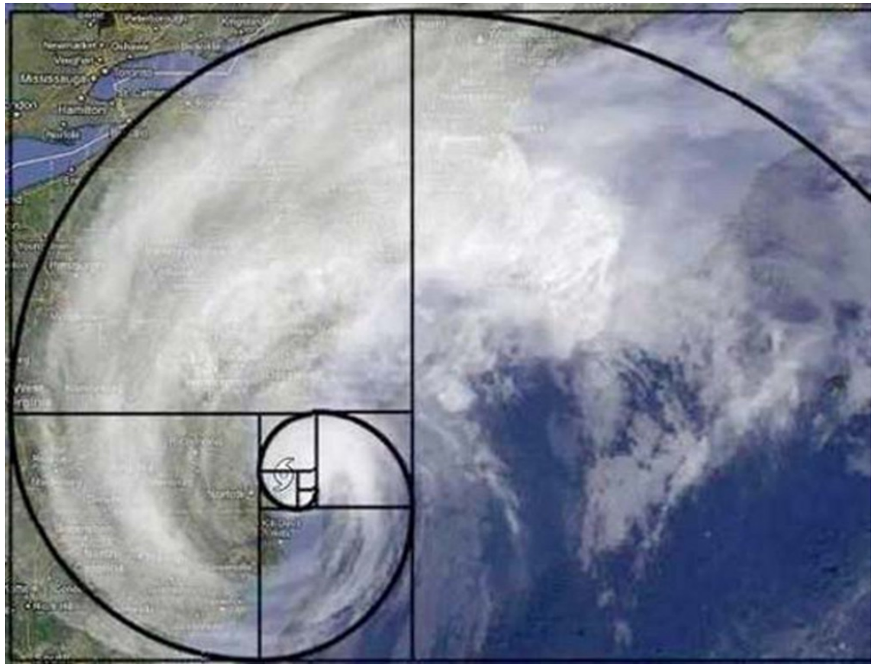


Fibonacci in Nature





Fibonacci in the Skies

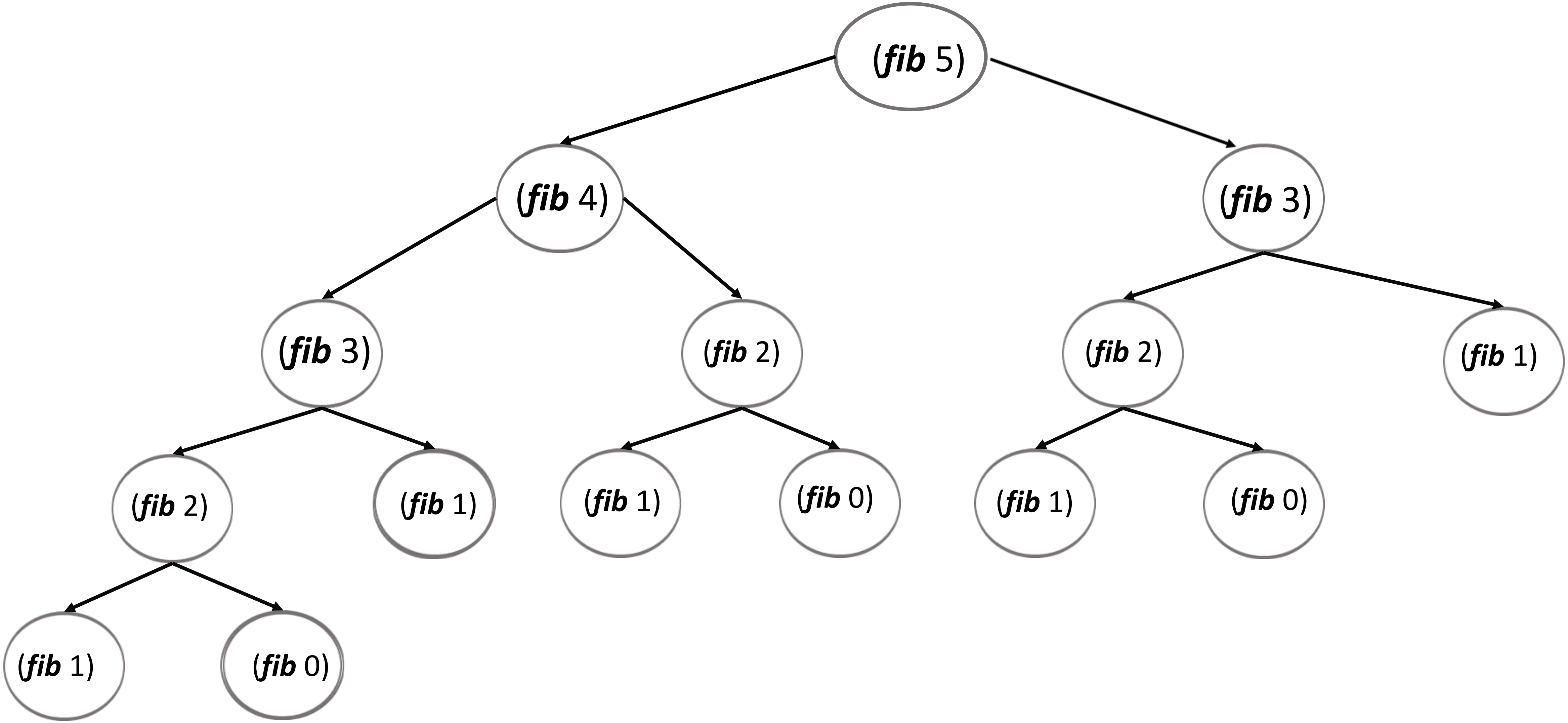


Fibonacci in Scheme

$$(define\ (\boldsymbol{fib}\ n)$$
$$(if\ (=\ n\ 0)\ 0$$
$$(if\ (=\ n\ 1)\ 1$$
$$(+\ (\boldsymbol{fib}\ (-\ n\ 1))\ (\boldsymbol{fib}\ (-\ n\ 2))))))$$

Unfortunately, this is a terrible Scheme program. Try it out on Dr Racket!

fib is wasteful



How Inefficient?

Evaluating $(fib\ 5)$ requires 8 evaluations of $(fib\ 1)\ and\ (fib\ 0)$.

Evaluating $(fib\ 6)$ requires 13 evaluations of $(fib\ 1)\ and\ (fib\ 0)$.

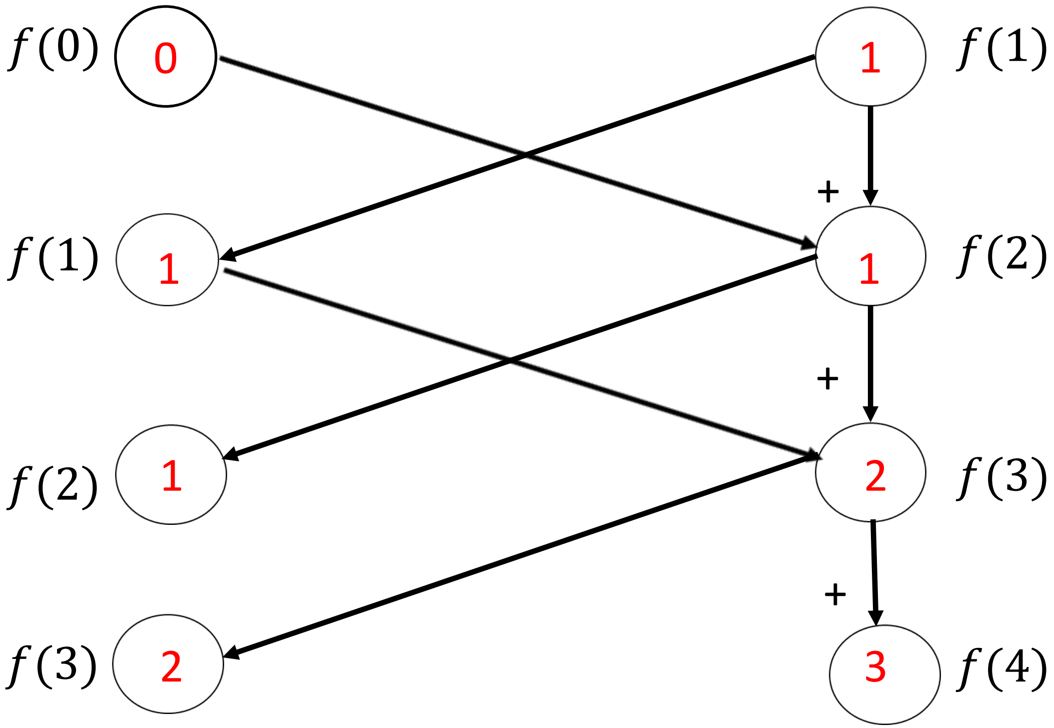
Evaluating $(fib\ 7)$ requires 21 evaluations of $(fib\ 1)\ and\ (fib\ 0)$.

Evaluating $(fib\ n)$ requires $f(n+1)$ evaluations of $(fib\ 1)\ and\ (fib\ 0)$

and $f(n)$ additions!

So $(fib\ 12)$ will require 144 additions! But we can do it naively with 11 additions.

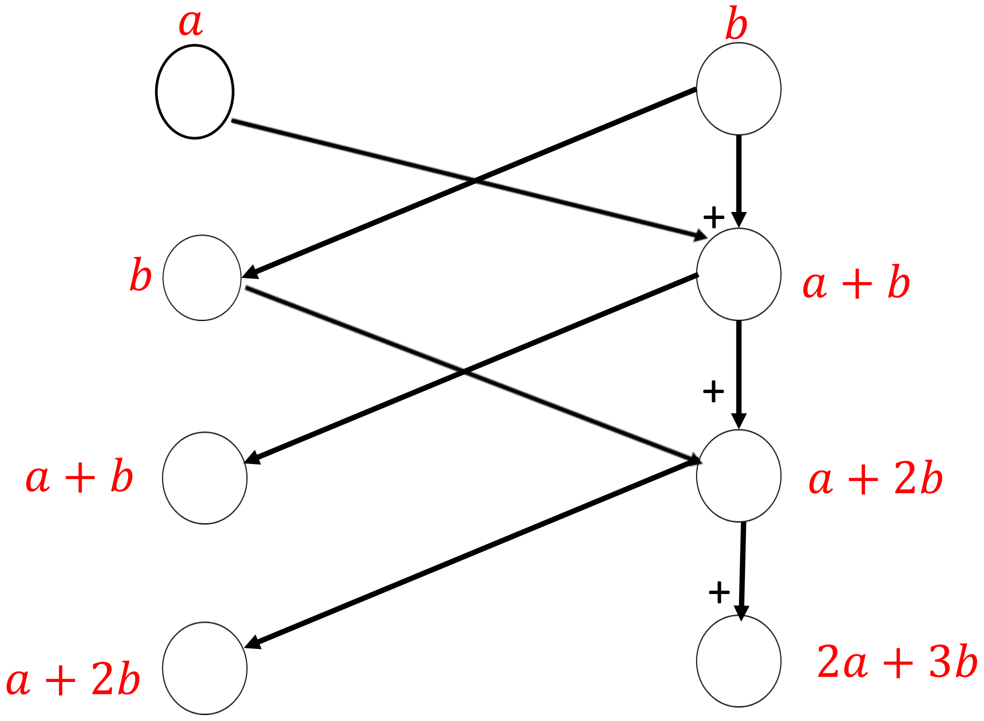
The right way to compute fibonacci numbers



A “Tail-Recursive” fibonacci Program

```
(define (fibiter n a b)
    (if (= n 0) a
            (if (= n 1) b
              (fibiter (- n 1) b (+ a b)))))
```

```
(define (fastfib n)
    (fibiter n 0 1))
```



But how do we prove that ***fastfib*** always returns the correct result?

Proof of correctness

$n = 0$: (fastfib 0) = (fibiter 0 0 1) = 0.

$n = 1$: (fastfib 1) = (fibiter 1 0 1) = 1.

$n = 2$: (fastfib 2) = (fibiter 2 0 1) = (fibiter 1 1 1) = 1

$n = 3$: (fastfib 3) = (fibiter 3 0 1) = (fibiter 2 1 1) = (fibiter 1 1 2) = 2

Looks like we need to prove something about fibiter first!

Proof of correctness

Claim: $\forall n \geq 1$ (fibiter $n\ a\ b$) = $af(n-1) + bf(n)$

Proof:

Basis: $n = 1.$ (fibiter $1\ a\ b) = b = a \cdot f(0) + bf(1)$

Inductive Hypothesis: $P(k)$: (fibiter $k\ a\ b$) = $af(k-1) + bf(k), k > 1.$

Inductive Step: (fibiter $k + 1\ a\ b) =$ (fibiter $k\ b\ a + b$)

$$= bf(k-1) + (a+b)f(k)$$
$$= af(k) + b\big(f(k-1) + f(k)\big)$$
$$= af(k) + bf(k+1)$$

Wrapping it up

Finally, we have that:

(fastfib 0) = 0

and for $n \geq 1$:

(fastfib n) = (fibiter n 0 1) = $0 \cdot f(n-1) + 1 \cdot f(n) = f(n)$

Therefore, fastfib correctly computes the fibonacci numbers.