Lecture 16: Recursive Definitions

Jingjin Yu | Computer Science @ Rutgers

October 30, 2020

RUTGERS

THE STATE UNIVERSITY OF NEW JERSEY

Algorithmic Robotics & Control Lab

Outline

- Lecture 15 review
- Recursive definitions: intuition & pictorial examples
- Specifying recursive definition
  - Basic examples
- Bunnies & Fibonacci sequence
- Proving a deterministic formula for the Fibonacci sequence
- Fibonacci sequence, golden ratio, and nature
- Programming recursive functions
- Recursively defined sets

- A repeating note: **make sure you read the textbook**

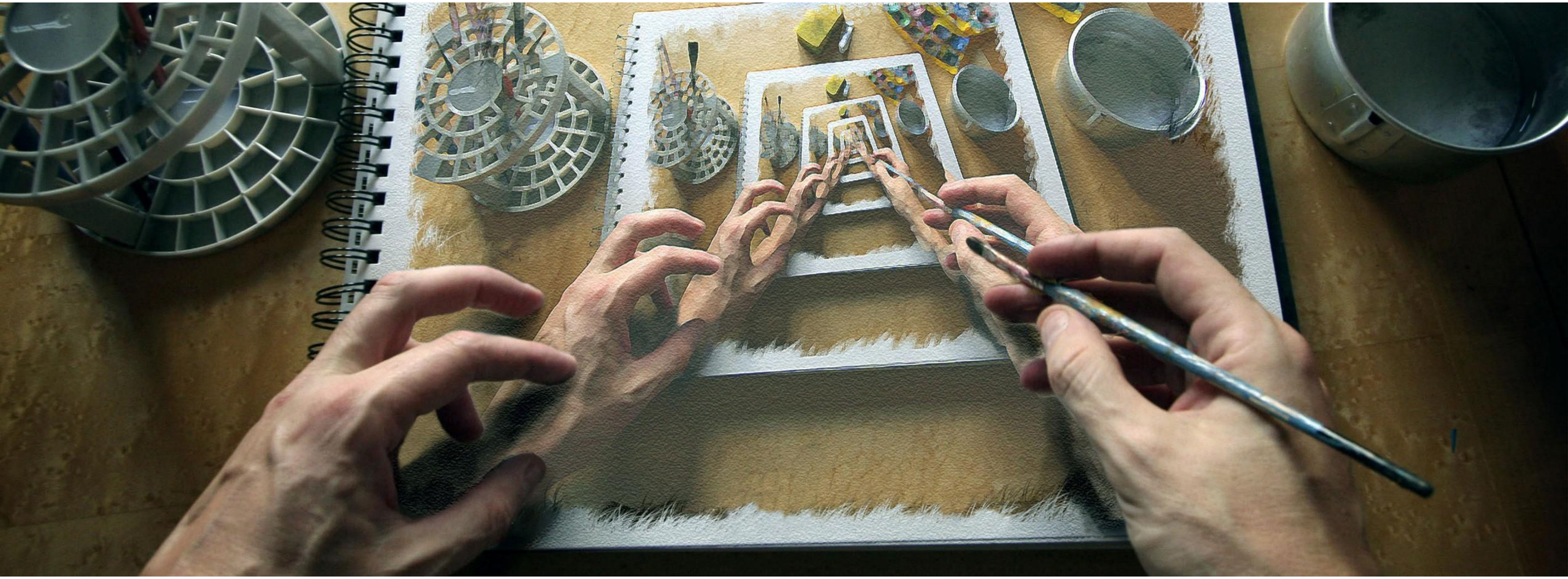
L15: Strong Induction

- Let $P(n)$ be a predicate depending on a positive integer $n \geq 1$. To show $P(n)$ is true for all n, two steps are needed:
  - (1) Basis step (or base case): verify that $P(1)$ is true.
  - (2) Inductive step: assume $P(i)$ holds for all $1 \leq i \leq k$, show

$$P(1) \land \cdots \land P(k) \rightarrow P(k+1)$$

- Note: we may use all of $P(1), \ldots, P(k)$ in the inductive step to prove that $P(k+1)$ hold. But we do not need to

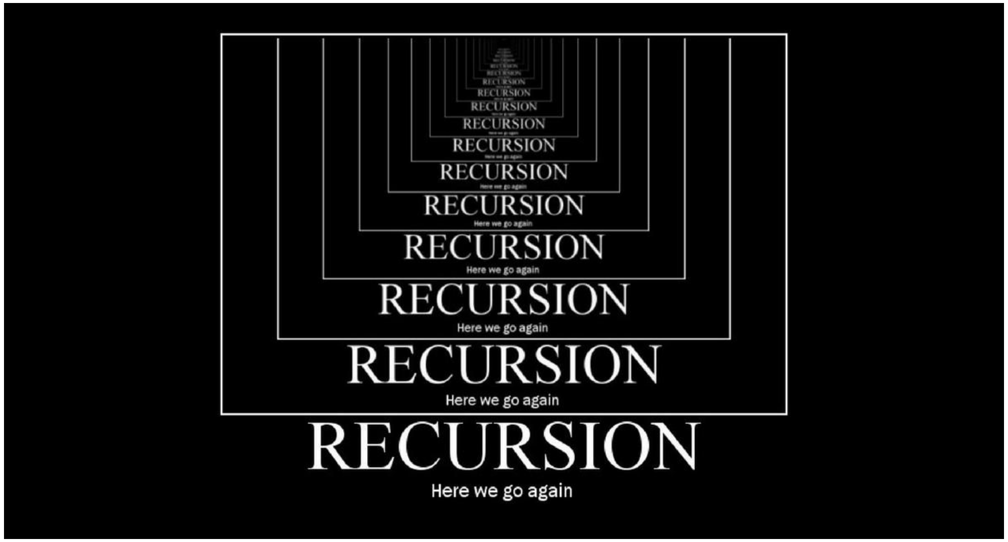
Recursive Definitions: The Intuition



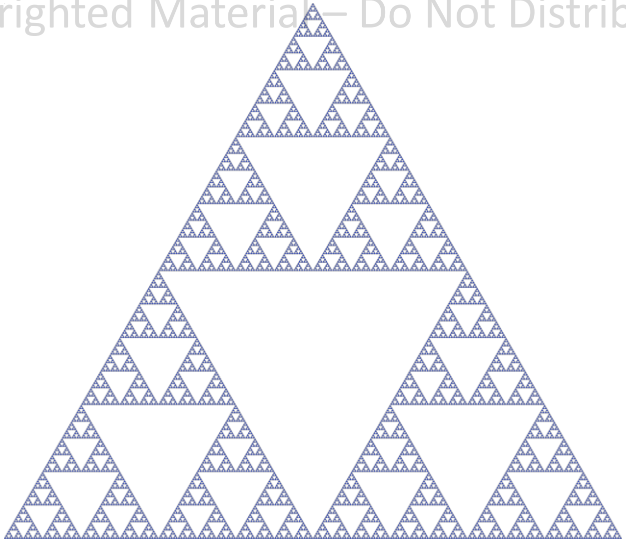
Picture source: https://engineering.tripping.com/building-an-recursive-nested-dropdown-component-in-react-b1c883e06ac4

- Sometimes, it is difficult to define certain things explicitly
- A better way may be recursive or inductive:
  - Define a "base case" (iteration 1)
  - Define iteration $k+1$ based on iteration k

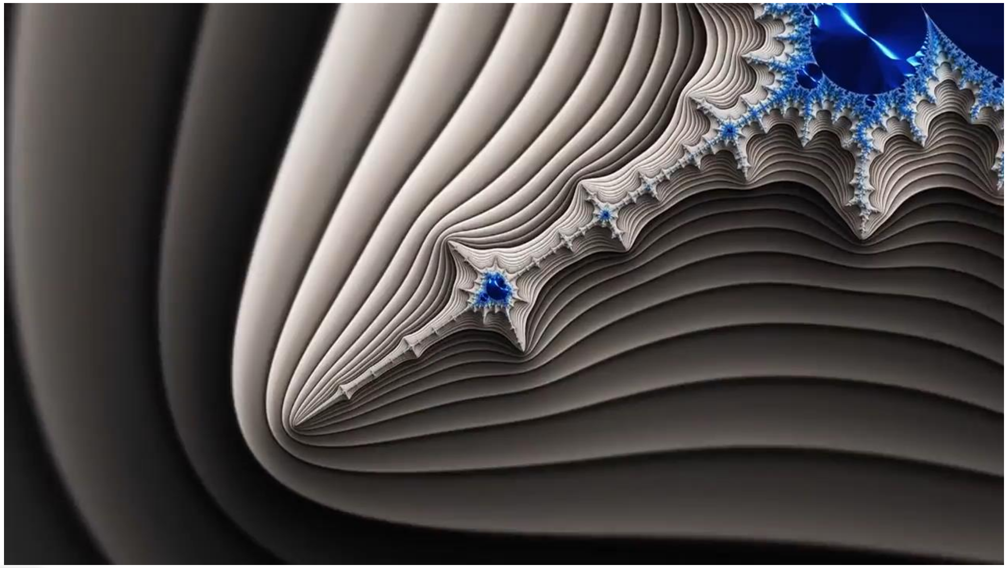
Recursive Definitions: Pictorial Examples



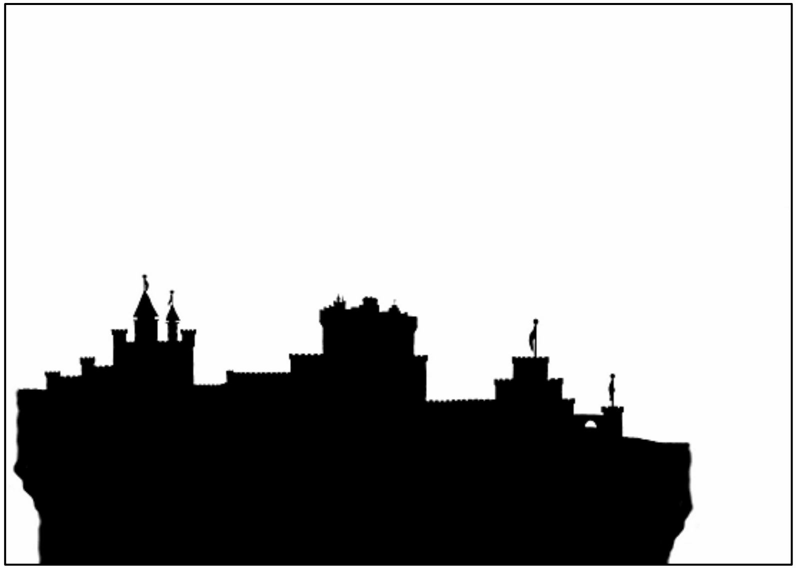
https://medium.com/@shmuel.lotman/the-2-00am-javascript-blog-about-recursion-yes-a-m-509d181fd7d4



Sierpiński triangle - wikipedia



Fractals animation: https://www.youtube.com/watch?v=8cgp2WNNKmQ



https://xkcd-time.fandom.com/wiki/Recursion

Recursive Definitions

- Specifying a recursive definition $f(n)$, $n \geq 0$
  - (1) Basis step: specify $f(0)$
  - (2) Specify $f(n+1)$ in terms of $f(n)$

- **Ex**: $f(0) = 1, f(n+1) = 2f(n)$

- Variation:
  - (1) Basis step: specify $f(0), f(1), \dots, f(k)$ for some $k \geq 1$
  - (2) Specify $f(n+1)$ in terms of $f(n-k), \dots, f(n)$

- **Ex**: $f(0) = 0, f(1) = 1, f(n+1) = f(n) + f(n-1)$.
  - This is the famous Fibonacci sequence

Basic Examples

- **Ex**: $f(0) = 3, f(n+1) = 2f(n) + 3$. Find $f(1)$ to $f(4)$.

$f(1) = f(0+1) = 2f(0)+3 = 9$

- **Ex**: Give a recursive definition of a^n

$f(0) = a^0 = 1, \quad f(n) = a^n, \quad f(n+1) = a^{n+1},$

$f(n+1) = f(n+1) = a \cdot f(n)$

$f(n+1) = a \cdot f(n)$

- **Ex**: Give a recursive definition of $\sum_{k=0}^{n} a_k$

$\rightarrow \{a_0, a_1, \ldots\ldots, a_k, \ldots\}$

$f(0) = \sum_{k=0}^{0} f(0) = \sum_{k=0}^{0} a_k = a_0$

$f(n+1) = f(n) + a_{n+1} \quad f(n) = \quad f(n+1) = \sum_{k=0}^{n+1} a_k = \sum_{k=0}^{n} a_k + a_{n+1}$

Bunnies and Fibonacci Series

- Rule
  - A pair of bunnies give birth to a pair of bunnies every month
  - Newly born bunnies will be ready to give birth after one month
- Q: You start with one pair of bunnies at month 0, # pairs at month n?

month 0 1 2 3 4 6 7 8 9

m.b.p 1 2 3 5 8 21 13 34 34 21 55 34 89 55 144

i.m.b.p 1 1 2 3 5 13 34 55 88

total 2 3 5 8 13

10
144
89
233

Fibonacci Series and More Bunnies

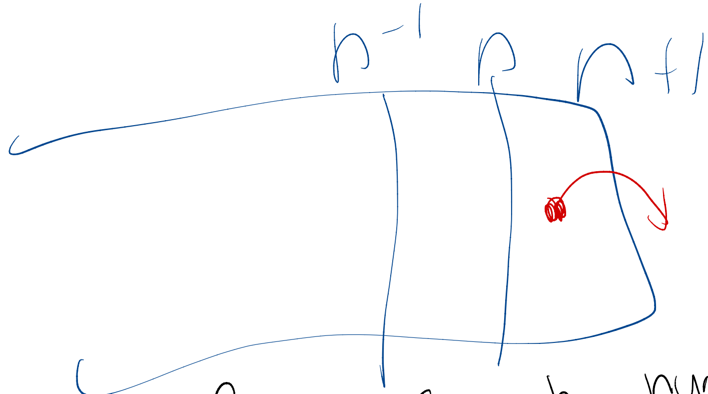
- Canonical form: $f(0) = 0, f(1) = 1, f(n+1) = f(n) + f(n-1)$
  - The base case matters a lot here

$n = 0 \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7 \quad 8 \quad 9 \quad 10$

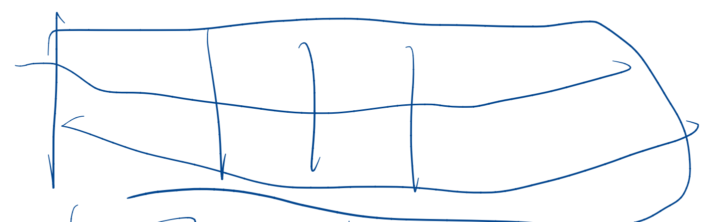
$f(n) = 0 \quad 1 \quad 1 \quad 2 \quad 3 \quad 5 \quad 8 \quad 13 \quad 21 \quad 34 \quad 55$

$1 \quad 3 \quad 4 \quad 7 \quad 11 \quad 18 \quad 29 \quad 47 \quad 76 \quad 123 \quad 199$

- A bunny is jumping along a ladder. It can jump over one rung or two rungs each time. If the ladder has n rungs, how may possible ways be there for the bunny to go through the ladder? E.g., for $n = 10$, the bunny may jump 22222 or 1212211.



$f(n+1) = f(n) + f(n-1)$ rungs



1 2 1 2

1 1 1 2

Formula for the Fibonacci Series

- **Ex**: Prove the Fibonacci sequence has the formula

$$f(n) = \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^n - \frac{1}{\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^n$$

Proof (1) Base case

$$f(0) = \frac{1}{\sqrt{5}}(\;)^0 - \frac{1}{\sqrt{5}}(\;)^0 = 0$$

$$f(1) = \frac{1}{\sqrt{5}} \cdot \left(\frac{1+\sqrt{5}}{2}\right) - \frac{1}{\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right) = \frac{2\sqrt{5}}{2\sqrt{5}}$$

$$\hookrightarrow = 1$$

(2) check $f(n) + f(n-1)$

$$\frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^n - \frac{1}{\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^n + \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^{n-1} - \frac{1}{\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^{n-1}$$

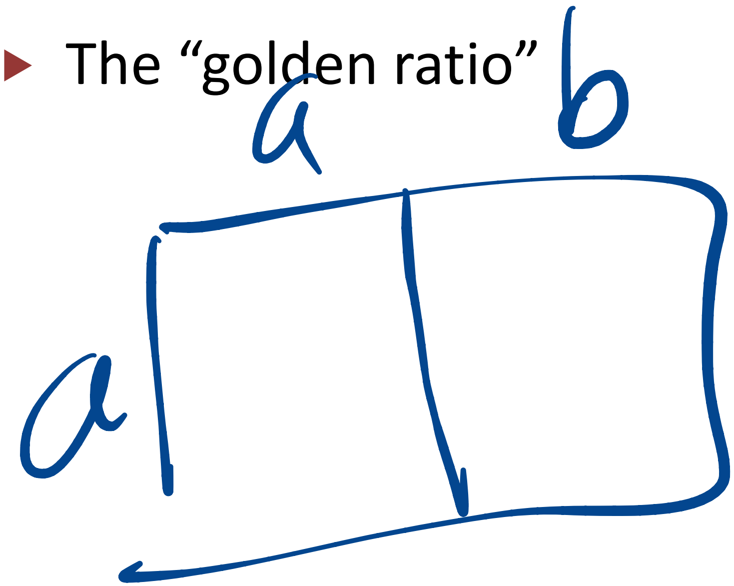
$$= \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^{n-1}\left(\frac{1+\sqrt{5}}{2}+1\right) - \frac{1}{\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^{n-1}\left(\frac{1-\sqrt{5}}{2}+1\right)$$

$$= \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^{n-1}\left(\frac{1+\sqrt{5}}{2}\right)^2 - \frac{1}{\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^{n-1}\left(\frac{1-\sqrt{5}}{2}\right)^2$$

$$= f(n+1)$$

Fibonacci Series, Golden Ratio, & Nature

- The “golden ratio”



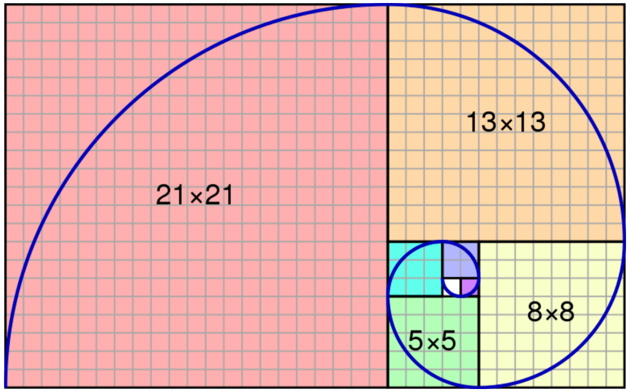
$\frac{1+\sqrt{5}}{2} \simeq 1.618,$

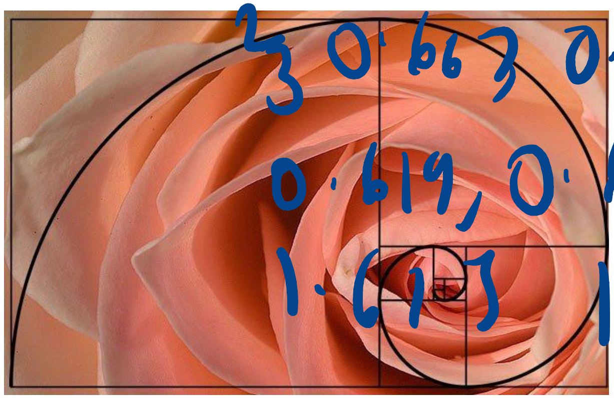
$\frac{a+b}{a} = \frac{a}{b}$

$\frac{2}{1+\sqrt{5}} \simeq 0.618$

$1+\frac{b}{a} = \frac{a}{b}$

2 3 5 8 13 21 32





2/3 0.667 0.6, 0.625, 0.615

0.619, 0.618, 1.5, 1.6, 1.625

1.617 1.619

Fibonacci Series in Sunflowers

https://www.youtube.com/watch?v=z9d1mxgZ0ag

Recursively Functions

- How to compute Fibonacci number recursively using a program?

```
int fibonacci(n) {
  if (n == 0)
    return 0;
  else if (n == 1)
    return 1;
  else {
    return fibonacci(n - 1) + fibonacci(n - 2);
  }
}
```

f(n)

f(n-1) f(n-2)

f(n-3) f(n-2) f(n-3) f(n-2)

- Problems?
  - Recursion tree
- How to resolve?

```
int fibonacci(n) {
  List<int> fib = new List<int>();
  fib[0] = 0;
  fib[1] = 1;
  for (int i = 2; i <= n; i++) {
    fib[i] = fib[i - 1] + fib[i - 2];
  }
}
```

  - Dynamic programming

Recursively Defined Sets

- We can also recursively define sets
- **Ex**: A recursively defined set
  - (1) Basis step: $3 \in S$
  - (2) Recursive step: for $x, y \in S$, $x + y \in S$

- **Ex**: Syntax of propositional logic
  - (1) Basis step: T, F, and s as a propositional variable, are well-formed formulae (in propositional logic).
  - (2) Recursive step: If P and Q are well-formed formulae, then $(\neg P)$, $(P \wedge Q)$, $(P \vee Q)$, $(P \rightarrow Q)$, and $(P \leftrightarrow Q)$ are well-formed formulae.