CS2209A 2017
Applied Logic for Computer Science

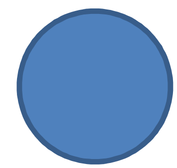
Lecture 20

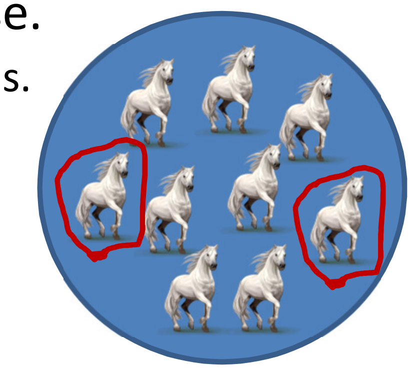
Recurrences and sequences

Instructor: Marc Moreno Maza

Puzzle: all horses are white

- Claim: all horses are white.
- Proof (by induction):
  - P(n): any n horses are white.
  - Base case: P(0) holds vacuously
  - Induction hypothesis: any k horses are white.
  - Induction step: if any k horses are white, then any k+1 horses are white.
    - Take an arbitrary set of k+1 horses. Take a horse out.
      - The remaining k horses are white by induction hypothesis.
    - Now put that horse back in, and take out another horse.
      - Remaining k horses are again white by induction hypothesis.
    - Therefore, all the k+1 horses in that set are white.
  - By induction, all horses are white.





What's wrong here?

Puzzle: all horses are white

(1) Let P(n) be: any n horses are white.

(2) Base case: 0 horses are white.

(3) Induction hypothesis.: if any set of k horses are white, then any set of k + 1 horses are white.

- What is the trick here?
- The problem is that the induction step relies on **some assumption about k that is not quite valid**.

Puzzle: all horses are white

(1) Let P(n) be: any n horses are white.

(2) Base case: 0 horses are white.

(3) Induction hypothesis: if any set of k horses are white, then any set of k + 1 horses are white.

- More precisely, it needs **a different base case**.
- The problem occurs when the proof says "Now, put that horse back in and remove another horse".
- But then we need to guarantee that there is "another horse" in the set.
- It would be true if k >= 1.
- However, for our base case we chose k >= 0.

Puzzle: all horses are white

(1) Let P(n) be: any n horses are white.

(2) Base case: 0 horses are white.

(3) Induction hypothesis: if any set of k horses are white, then any set of k + 1 horses are white.

- Thus, the proof does not go through without **the base case of k = 1**
- This is just an example of caveats to watch out for when doing induction proofs: **make sure there are no assumptions about k that could not be handled by the base case.**

Revisit: Sums

- How to write long sums, e.g., 1+2+... (n-1)+n concisely?
  - Sum notation ("sum from 1 to n"): $\sum_{i=1}^{n} i = 1 + 2 + \ldots + n$
    - If n=3, $\sum_{i=1}^{3} i$ = 1+2+3=6.
    - The name "i" does not matter. Could use another letter not yet in use.
- In general, let $f\colon \mathbb{Z} \to \mathbb{R}$, $m, n \in \mathbb{Z}, m \leq n$.
  - $\sum_{i=m}^{n} f(i) = f(m) + f(m+1) + \cdots + f(n)$
    - If m=n, $\sum_{i=m}^{n} f(i)$ =f(m)=f(n).
    - If n=m+1, $\sum_{i=m}^{n} f(i)$ = f(m)+f(m+1)
    - If n>m, $\boldsymbol{\sum_{i=m}^{n} f(i)}$ **=** $\boldsymbol{(\sum_{i=m}^{n-1} f(i)) + f(n)}$
    - Example: $f(x) = x^2$. $2^2 + 3^2 + 4^2 = \sum_{i=2}^{4} i^2 = 29$

Revisit: Products

- Similarly for product notation (product from m to n):
  - $\Pi_{i=m}^{n} f(i)$
    $= f(m) \cdot f(m+1) \cdot \ldots \cdot f(n)$
    $= (\boldsymbol{\Pi_{i=m}^{n-1}\ f(i)) \cdot f(n)}$
  - For $f(x) = x, \quad 2 \cdot 3 \cdot 4 \ = \ \Pi_{i=2}^{4} i = 24$
  - $1 \cdot 2 \cdot \ldots \cdot n = \Pi_{i=1}^{n} i = n!$ (n factorial)

Recurrences and sequences

- **Sequence**: enumeration of objects $s_1, s_2, s_3, \ldots, s_n, \ldots,$
  - Sometimes use notation $\{s_n\}$ for the sequence (i.e., set of elements forming a sequence)

- To define a sequence (of things), describe the process which generates that sequence.
  - **Basis (initial conditions):** what are the first (few) element(s) in the sequence.
    - $A_0 = \emptyset$
  - **Recurrence (recursion step, inductive definition)**: a rule to make a next element from already constructed ones.
    - $A_{n+1} = \mathcal{P}(A_n)$

- Resulting sequence:
  - $\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}, \left\{\emptyset, \{\emptyset\}, \{\{\emptyset\}\}, \{\emptyset, \{\emptyset\}\}\right\}, \ldots$

Recurrences and sequences: example

- To define a sequence (of things), describe the process which generates that sequence.
  - **Basis (initial conditions):** what are the first (few) element(s) in the sequence.
    - 0! = 1.   1! = 1.
  - **Recurrence (recursion step, inductive definition)**: a rule to make a next element from already constructed ones.
    - $(n+1)! = n! \cdot (n+1)$
- Resulting sequences:
  - 1, 2, 6, 24, 120, …

Special sequences

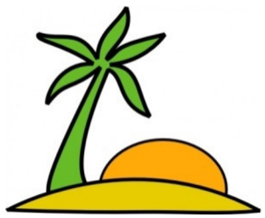
- Geometric progression:
  - **Sequence**: $c, cr, cr^2, cr^3, \ldots, cr^n, \ldots$
  - **Recursive definition**:
    - **Basis**: $s_0 = \mathrm{c}$, for some c ∈ $\mathbb{R}$
    - **Recurrence**: $s_{n+1} = s_\mathrm{n} \cdot r$, where r ∈ $\mathbb{R}$ is a fixed number.
  - **Closed form**: $s_n = c \cdot r^n$

- A **closed form** of a recurrence relation is an expression that defines an n^{th} element in a sequence in terms of n directly.

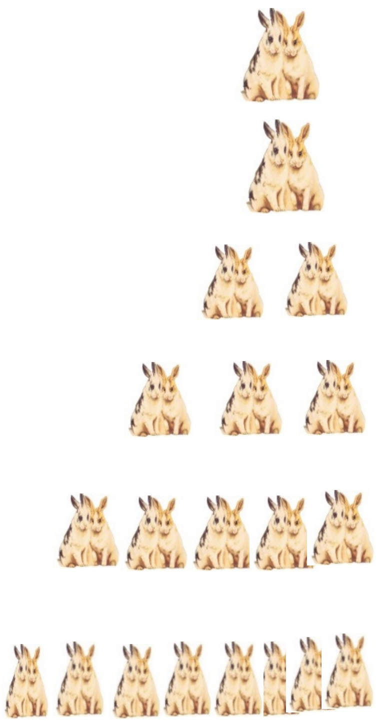
Special sequences

- Arithmetic progression:
  - **Sequence**: $c, c+d, c+2d, c+3d, \ldots, c+nd, \ldots$
  - **Recursive definition**:
    - **Basis**: $s_0 = \mathrm{c}$, for some c $\in \mathbb{R}$
    - **Recurrence**: $s_{n+1} = \mathrm{s_n} + \mathrm{d}$, where $d \in \mathbb{R}$ is a fixed number.
  - **Closed form**: $s_n = c + nd$
    - Closed forms are very useful for analysis of recursive programs, etc.

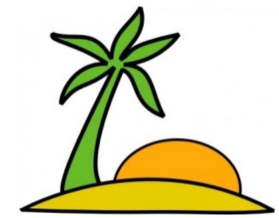
Fibonacci sequence



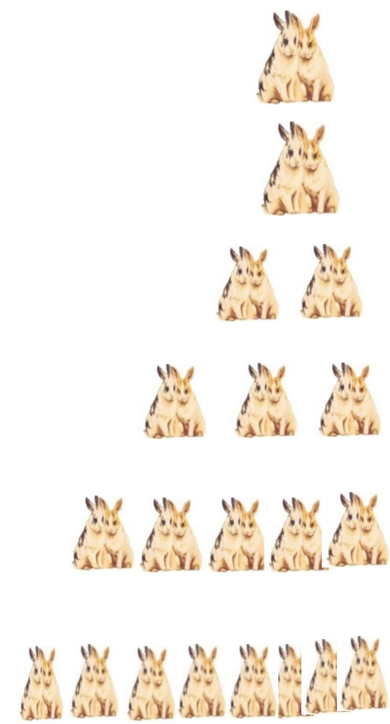
- Imagine that a ship leaves a pair of rabbits on an island (with a lot of food).
- After a pair of rabbits reaches 2 months of age, they produce another pair of rabbits, which in turn starts reproducing when reaching 2 months of age …
- How many pairs rabbits will be on the island in n months, assuming no rabbits die?



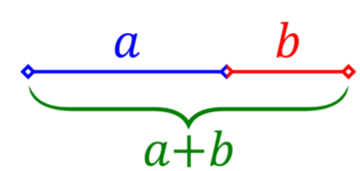
Fibonacci sequence



- **Basis**: $F_1 = 1,\ \mathrm{F}_2 = 1$
- **Recurrence**: $F_n = F_{n-1} + F_{n-2}$
- **Sequence**: 1, 1, 2, 3, 5, 8, 13 …



- **Closed form**: $F_n = \frac{\varphi^n - (1-\varphi)^n}{\sqrt{5}}$
  - Golden ratio: φ
  - $\varphi = \frac{a+b}{a} = \frac{a}{b} = \frac{1+\sqrt{5}}{2}$



Partial sums

- Properties of a **sum**:
  - $\sum_{i=m}^{n} \left(f(i) + g(i) \right) = \sum_{i=m}^{n} f(i) + \sum_{i=m}^{n} g(i)$
  - $\sum_{i=m}^{n} c \cdot f(i) = c \sum_{i=m}^{n} f(i)$

Partial sums

- Sum of arithmetic progression:
  - $s_n = c + nd$ for some c, $d \in \mathbb{R}$
  - **Sequence**: $c, c+d, c+2d, c+3d, \ldots, c+nd, \ldots$
  - **Partial sum**:
    - $\sum_{i=0}^{n} s_n$

$$= \sum_{i=0}^{n} (c + id\,) = \sum_{i=0}^{n} c \; + \sum_{i=0}^{n} id$$

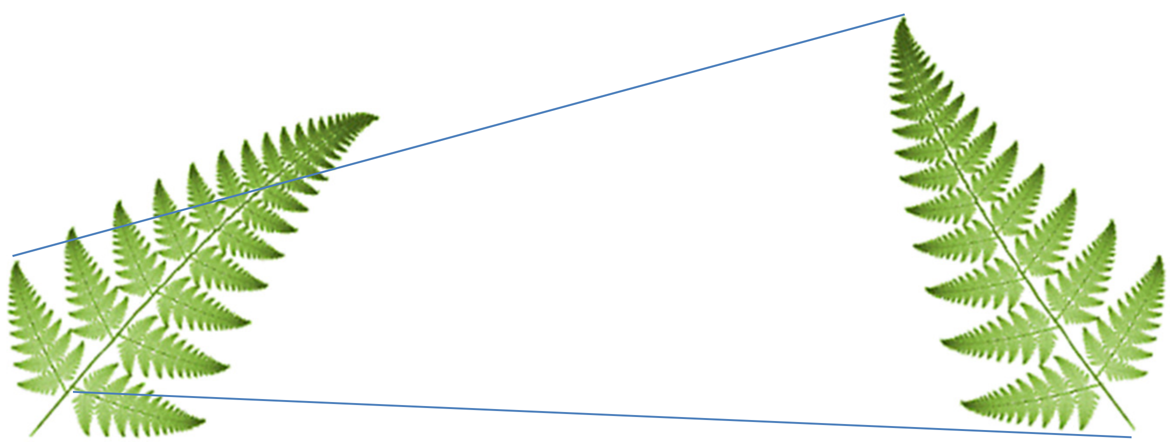
$$= \mathrm{c(n+1) + d} \sum_{i=0}^{n} i = c(n+1) + d\,\frac{n(n+1)}{2}$$

Partial sums

- Sum of geometric progression:
  - $s_n = c \cdot r^n$ for some $c, r \in \mathbb{R}$
  - Sequence: $c, cr, cr^2, cr^3, \ldots, cr^n, \ldots$
  - Partial sum:
    - If r=1, then $\sum_{i=0}^{n} s_n = c(n+1)$
    - If $r \neq 1$, then $\sum_{i=0}^{n} s_n = \frac{cr^{n+1}-c}{r-1}$

Fractals

- Can use recursive definitions to define fractals
  - And draw them
  - And prove their properties.
- **Self-similar**: a part looks like the whole.



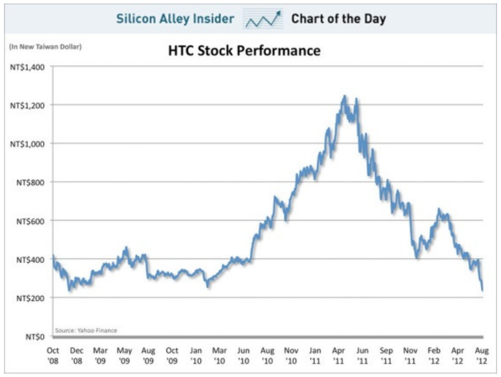
Fractals in nature

- A fern leaf

- Broccoli

- Mountains

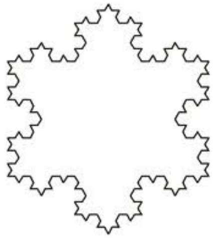
- Stock market



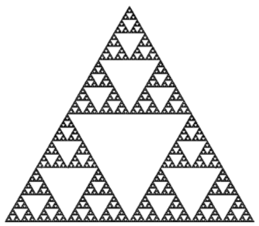
- Heat beat

Mathematical fractals

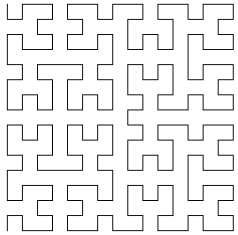
- Koch curve and snowflake



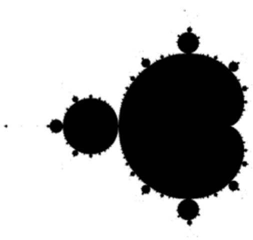
- Sierpinski triangle, pyramid, carpet



- Hilbert space-filling curve

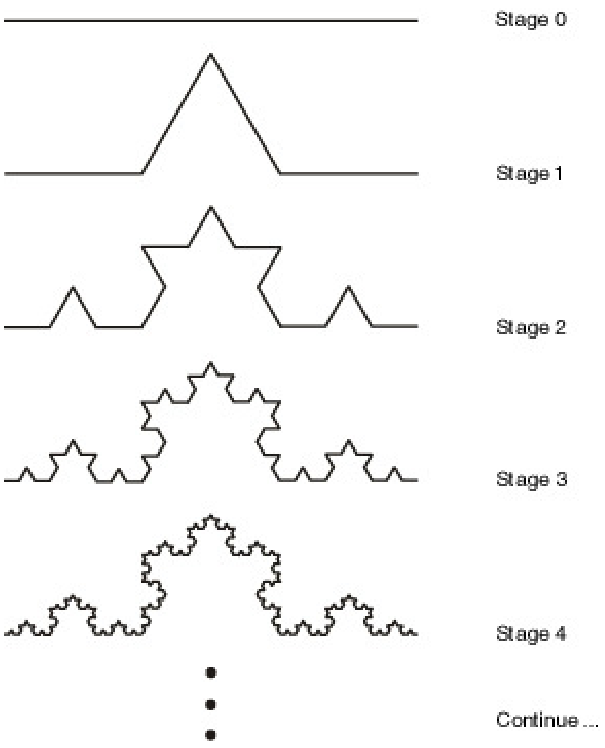


- Mandelbrot set



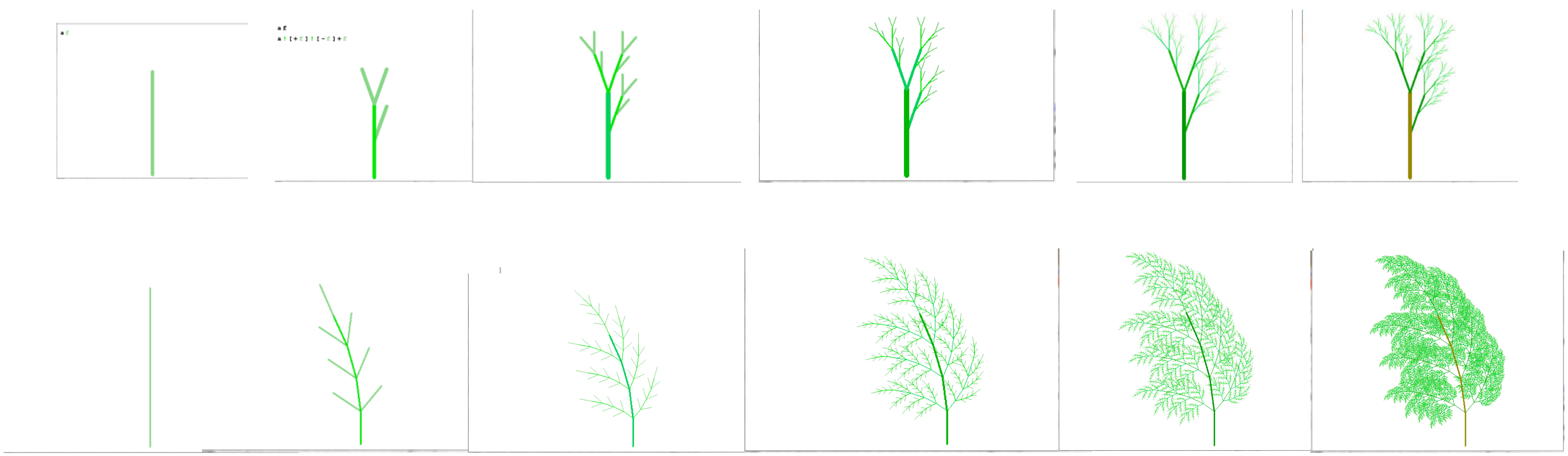
Koch curve

- **Basis***:* an interval
- **Recursive step***:* Replace the inner third of the interval with two of the same length
- …



Playing with fractals

- Fractal Grower by Joel Castellanos:
- http://www.cs.unm.edu/~joel/PaperFoldingFractal/paper.html



Tower of Hanoi game

- Rules of the game:
  - Start with all disks on the first peg.
  - At any step, can move a disk to another peg, as long as it is not placed on top of a smaller disk.
  - Goal: move the whole tower onto the second peg.
- Question: how many steps are needed to move the tower of 8 disks? How about n disks?