Lecture 12 - part 2

Recap last lecture:

1. Finished general graphs (digraphs, topological sort/ordering)
2. Basic combinatorics:

   sequences (k^n), permutations ($n!$), combinations ($\binom{n}{k} = \frac{n!}{(n-k)!k!}$)
3. Recursion/induction

Brain warm-up: a combinatorics problem:

How many different words (strings of letters) can you make from the word

mississippi

Brain warm-up: a combinatorics problem:

How many different words (strings of letters) can you make from the word

mississippi

$$\frac{11!}{4!\,4!\,2!} = \frac{11\cdot\overset{5}{\cancel{10}}\cdot\cancel{9}\cdot\cancel{8}\cdots 5}{4!\cdot\cancel{2}!} = 11\cdot5\cdot3\cdot7\cdot6\cdot5 = 34650$$

$\uparrow$ i's $\quad \uparrow$ s's $\quad \uparrow$ p's

$$\rotatebox{90}{=}$$

$$1\cdot 3\cdot 4$$

$$8$$

Brain warm-up: a combinatorics problem:

Combinatorics will show up on the final exam.
Homework: Read Schaum 5.1 - 5.5!
Solve the solved problems pg 96!

Recap recursions

$$F_0 = 0$$
$$F_1 = 1$$
$$F_n = F_{n-1} + F_{n-2}\ ,\ (n > 1)$$

```
findBinary(decimal){
if (decimal == 0)
      binary = 0
   else
      binary = decimal mod 2 + 10 * (findBinary(decimal div 2)}
```

Recap recursions

$F_0 = 0$
$F_1 = 1$
$F_n = F_{n-1} + F_{n-2}\ ,\ (n > 1)$

Fixed behaviour for initial cases

Inductive/recursive step to smaller

```
findBinary(decimal){
if (decimal == 0)
      binary = 0
   else
      binary = decimal mod 2 + 10 * (findBinary(decimal div 2)}
```

Fixed behaviour for initial cases

Inductive/recursive step to smaller

Closed expressions…

$$f(1) = 1$$
$$f(n) = n^2 + f(n-1)$$

$$n(n+1)(2n+1)/6$$

$$t_0 = 5,\ t_1 = 6$$
$$t_n = t_{n-1} + 6t_{n-2} - 12n + 8, \quad n \geq 2$$

$$t_n = 3^n + (-2)^n + 2n + 3$$

How? Need to prove something is true *<u>for all numbers *n*</u>*

Proving that something holds *for all n*: *induction*

(i) base case
ensure P(0) holds (is true) [can start at some l>0]

(ii) inductive step (step case)
prove that [if P(k) holds, then so does P(k+1)] (for all k)

Alternatively:
(ii) inductive step (step case)’
prove that [prove that if P(k’) holds for all value k’<k then it holds for k] (for all k)

Proving that something holds *for all n*: *induction*

$$\text{Prove: } 1+\sum_{\ell=1}^{n}(\ell\times\ell!)=(n+1)!, \quad \textbf{for } n\geq 1.$$

Base: k=1; 1+1x1! = 2 = 2!

Step: show that if the claim is true for k (assumption),
then it is also true for k+1

Usual trick: express the (k+1) expression in terms of the k-expression,
use assumption

Proving that something holds *for all n*: *induction*

Step: show that if the claim is true for k (assumption),
then it is also true for k+1

Usual trick: express the (k+1) expression in terms of the k-expression,
use assumption

Assumption: $1 + \sum_{l=1}^{k} (l \times l!) = (k+1)!$

Want to show: $1 + \sum_{l=1}^{k+1} (l \times l!) = (k+2)!$

$$1 + \sum_{l=1}^{k+1} (l \times l!) = 1 + \sum_{l=1}^{k} (l \times l!) + [(k+1) \times (k+1)!] = \underbrace{(k+1)!}_{by\ assumption} + [(k+1) \times (k+1)!]$$

$$= (k+1)! + [(k+1) \times (k+1)!] = (k+2) \times (k+1)! = (k+2)! \blacksquare$$

Recursively defined functions and proofs
over integers via induction will appear in exam.

Homework: read Schaum 3.6.
Schaum 1.8; 11.3.

Moving on: uses of *induction* beyond *sequences and functions*

Moving on: uses of *induction* beyond *sequences and functions*

Can be used to provide definitions of:

- **sets,**
- **relations,**
- **sequences,**
- **functions,**
- **trees,**
- **orders,**
- **syntax…**

Structural induction:

- **proving various properties of structures (e.g. *trees, as we will see*)**

Induction is behind the meaning of dots…

$E = \{1,3,5,7,\ldots\}$

The meaning of dots…

E = {1,3,5,7,…}

Definition. The set of odd natural numbers is defined as follows:

basis (*basis clause*)

1) $1 \in E$

inductive step (*inductive clause*)

2) ***if*** $x \in E$, ***then*** $x+2 \in E$

exclusion *(extremal clause)*

3) ***E has no other elements beside those specified by 1 and 2.***

Basis and induction steps may be complicated and contain many lines

Examples

- The odd natural numbers are often specified using dots: {1,3,5,7, ...} with dots (“etc.”). Same for even.
- This is ambiguous. How about “odd numbers divisible only with 1 and themselves” (primes + 1)
- Inductive defition is fully unambiguous.
- **Languages (a term in CS)** are often defined inductively (examples next).
- These languages also define the preorder traversal for binary trees (explained later)
- This symmetric arrangement is a way to enumerate nodes in a binary tree.
  More about (binary) trees in the next lectures.

Induction over structures

(how to prove something holds
***for all* objects in some (infinite)**
inductively defined set)

Language: an important concept in CS…

-Set of "letters" or "symbols" denoted Σ,

e.g. $\Sigma = \{0,1\}; \Sigma = \{a, b, c, \ldots, z\};$

Language: an important concept in CS…

-Set of "letters" or "symbols" denoted Σ, e.g. $\Sigma = \{0,1\}; \Sigma = \{a, b, c, \ldots, z\};$

-Strings: n-tuplet over Σ, or finite sequence with values in Σ:

-00101010, 000, 111; abba, benelux, aaaaa;

Language: an important concept in CS…

-Set of "letters" or "symbols" denoted Σ, e.g. $\Sigma = \{0,1\}; \Sigma = \{a, b, c, \ldots, z\};$

-Strings: n-tuplet over Σ, or finite sequence with values in Σ:

-00101010, 000, 111; abba, benelux, aaaaa;

set of all strings denoted using the "Kleene star":

$$\Sigma^* = \bigcup_{k=0}^{\infty} \Sigma^k; \Sigma^k = \{x_1 x_2 x_3 \ldots x_k \,|\, x_j \in \Sigma\})$$

-empty string: ϵ (sometimes λ)

Language: an important concept in CS…

-Set of "letters" or "symbols" denoted Σ, e.g. $\Sigma = \{0,1\}; \Sigma = \{a, b, c, \ldots, z\};$

-Strings: n-tuplet over Σ, or finite sequence with values in Σ:

-00101010, 000, 111; abba, benelux, aaaaa;

A ***language*** is a subset of strings over some alphabet

(a collection of *words*)

Will do this more seriously shortly

Language: subset of strings

Definition. The language L over $\Sigma = \{a, b\}$ is defined as follows:

basis (*basis clause*)

1) $b \in L$

inductive step (*inductive clause*)

2) ***if*** $x \in L$, ***then*** $abx \in L$

exclusion *(extremal clause)*

3) ***L has no other elements (words) beside those specified by 1 and 2.***

Language: subset of strings

Definition. The language L over $\Sigma = \{a, b\}$ is defined as follows:

basis (*basis clause*)

1) $b \in L$

inductive step (*inductive clause*)

2) ***if*** $x \in L$, ***then*** $abx \in L$

exclusion *(extremal clause)*

3) ***L has no other elements (words) beside those specified by 1 and 2.***

b, abb, ababb, abababb…abababab b

In other words, words of the form $(ab)^n b,\ n \geq 0$

Looking ahead: language of binary trees over {a,b}

Definition. The language L over $\Sigma = \{a, b, +\}$ is defined as follows:

basis (*basis clause*)

1) $a \in L, b \in L$

inductive step (*inductive clause*)

2) ***if*** $x, y \in L$, ***then*** $+xy \in L$

exclusion *(extremal clause)*

3) *L has no other elements (words) beside those specified by 1 and 2.*

b, +aa, +ab, ++abb, +b+aa, ++aa+ab, +++aa+ab++abb…

These are encoded binary trees

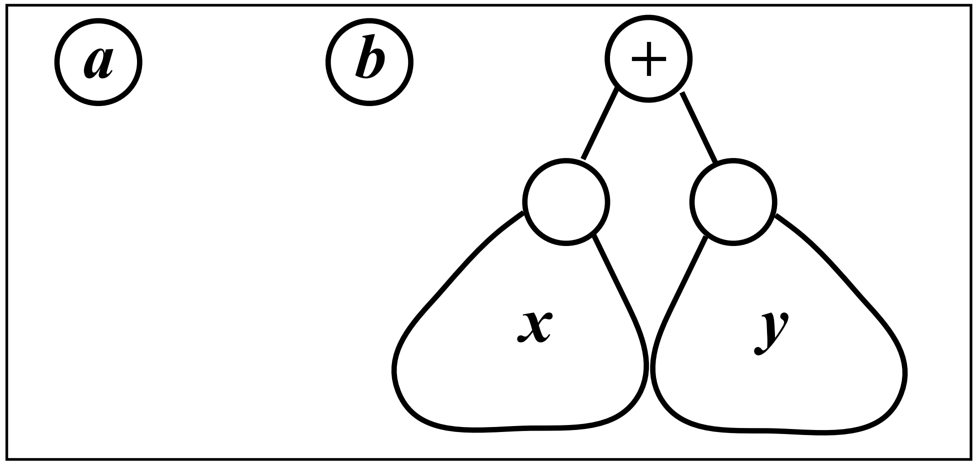
Looking ahead: language of binary trees

b, +aa, +ab, ++abb, +b+aa, ++aa+ab, +++aa+ab++abb…

Why are these *trees*?!!?

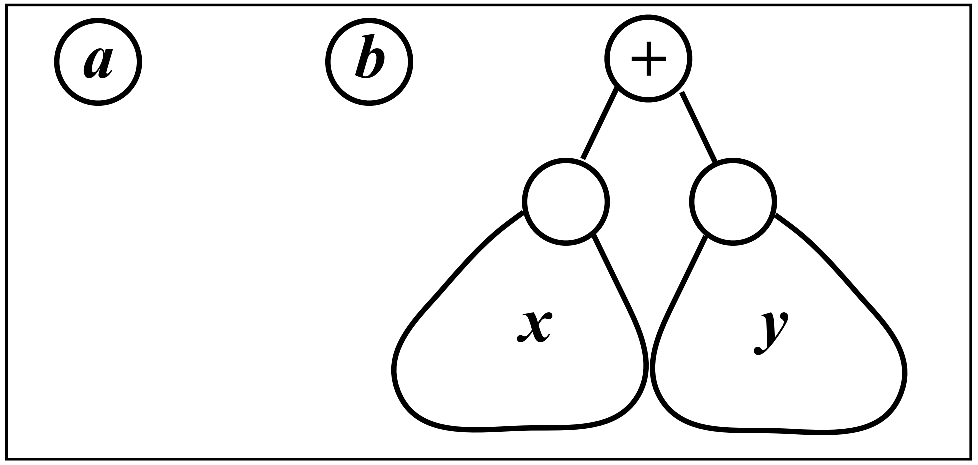
Looking ahead: language of binary trees

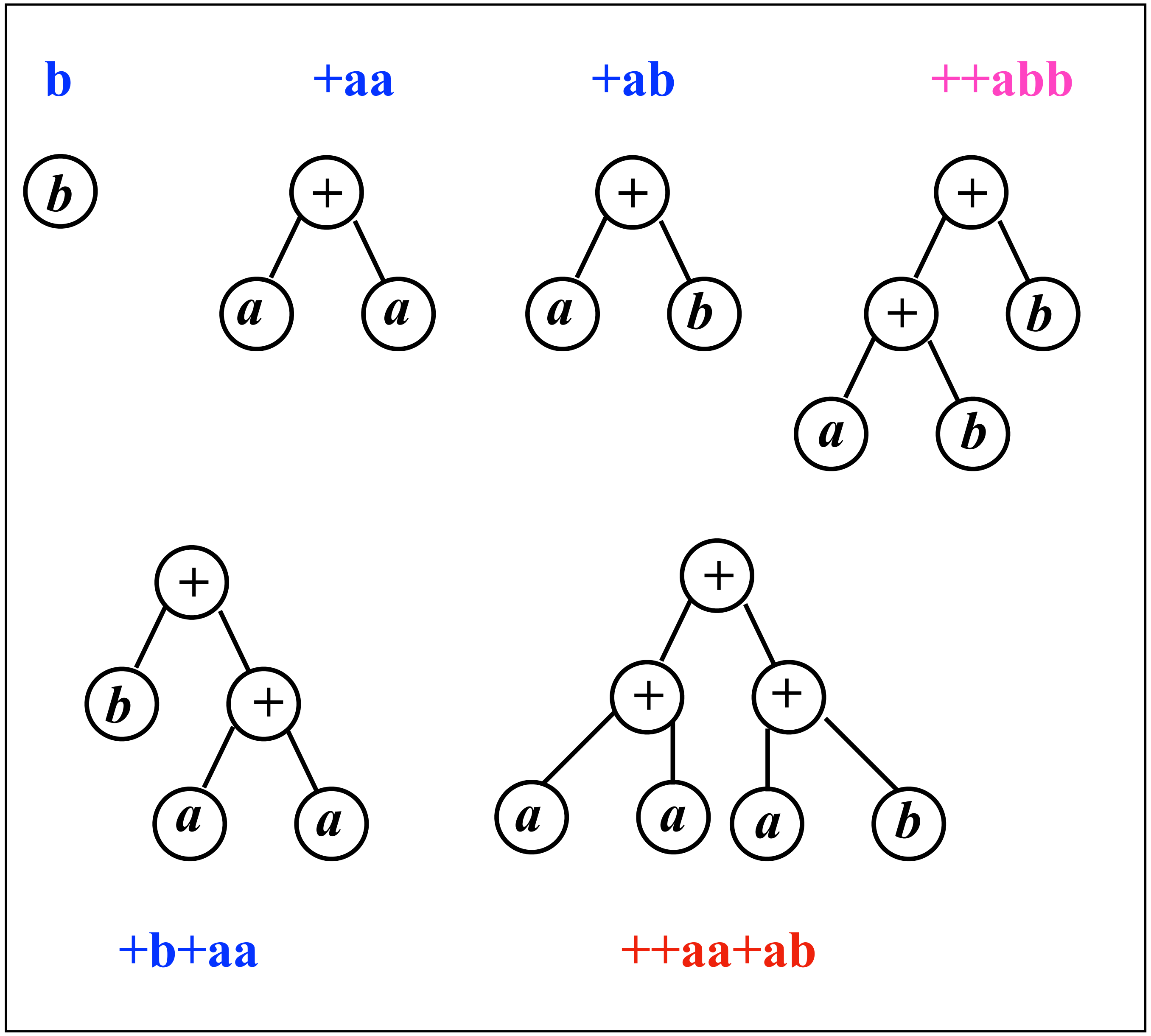
1) $a \in L, b \in L$
2) *if* $x, y \in L$, *then* $+xy \in L$

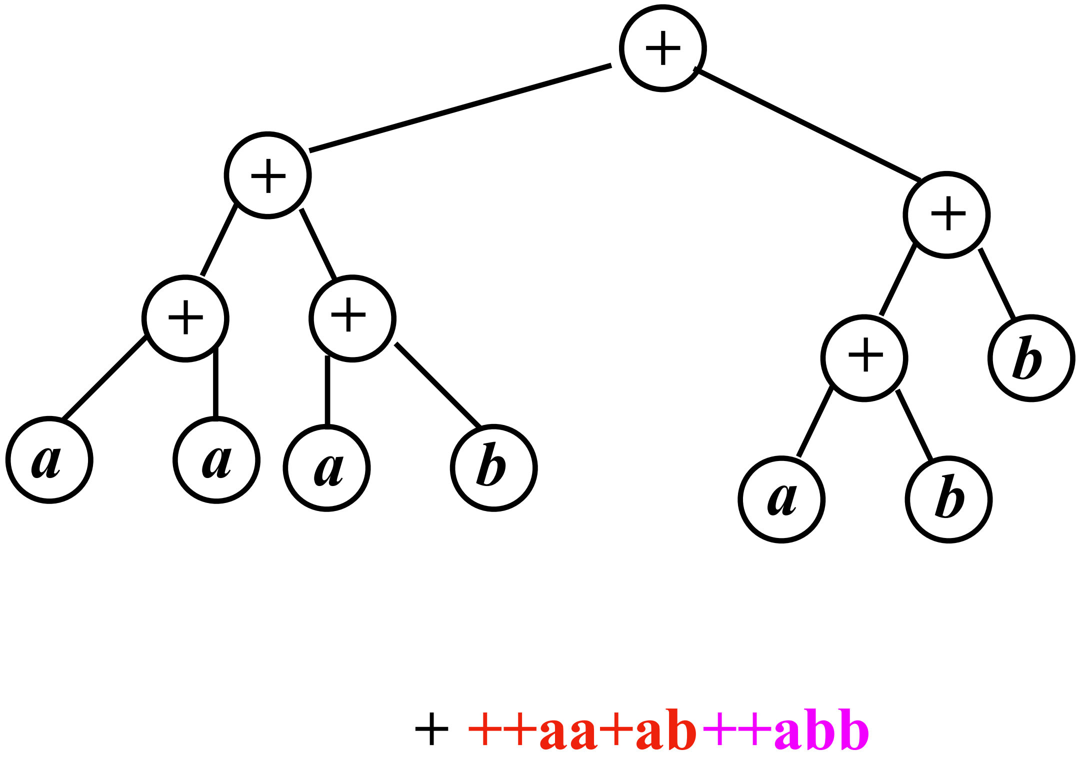


Looking ahead: language of binary trees

1) $a \in L, b \in L$
2) *if* $x, y \in L$, *then* $+xy \in L$







Basic idea (extended binary trees):

1) the empty set is an extended binary tree
2) a vertex r (the root of T) and the left (a) and right (b) subtree (also trees) whose roots are the children of r .

More on this in subsequent lectures

Inductive defs. continued: relations

1) $0 < 1$

2) *if* $x < y$ *then* $x < y + 1$ *and* $x + 1 < y + 1$

3) *the relation* $<$ *has no other elements aside from what is specified by 1)&2)*

$0<1$

$0<2$ $1<2$

$0<3$ $1<3$ $2<3$

$0<4$ $1<4$ $2<4$ $3<4$

$0<5$ $1<5$ $2<5$ $3<5$ $4<5$

equivalently:

1) $(0,1) \in <$

2) *If* $(x, y) \in <$ *then* $(x, y + 1), (x + 1, y + 1) \in <$

3) *no other elements*

Seen so far: inductively defined sets

- numbers
- strings
- relations (pairs)

We can prove properties of inductively defined objects…

Checking if some property P holds all elements of some ***inductively defined set*** *V*

(i) *base case*
ensure P holds for the basis of the induction of V

(ii) *inductive step (step case)*
Prove that P(y) holds for all y in V assuming that P (x) holds for all x from which y can be constructed

we again move to the "smaller" cases

We can prove properties of inductively defined objects…

Checking if some property P holds all elements of some ***inductively defined set*** *V*

(i) *base case*
ensure P holds for the basis of the induction of V

(ii) *inductive step (step case)*
Prove that P(y) holds for all y in V assuming that P (x) holds for all x from which y can be constructed

Inductively defined sets look like:

basis: 1) explicit elements are $\in V$

inductive step (*inductive clause*)
2) ***if*** $x, \ldots, z \in V$, ***then some consturctions of x...z*** $\in V$

exclusion *(extremal clause)*
3) ***V has no other elements beside those specified by 1 and 2.***

Intuitive example

Unstructured tree graph:

1) **A single vertex is a tree**
2) **If T is a tree, then the graph obtained by adding a vertex and connecting it to one of the vertices of T is a tree**
3) **Only graphs obtainable by 1) and 2) are trees**

Lemma: a tree over *n* vertices has *n-1* edges.

induction over vertex numbers of trees!

-basis: n=1 true;

-step: consider any n-vertex graph G; it was constructed from an n-1 vertex graph G' by inductive definition, by adding one edge to a new vertex.

By assumption G has n-2 edges. But since G' was obtained by adding one new edge, G' has n+1 edges.

Digression

Give an inductive definition of the set L of strings that consist of a number of a's followed by the same number of b's.
L over $\{a, b\}$ and $L = \{a^n b^n \,|\, n \geq 0\}$

Solving ...

Digression

Give an inductive definition of the set L of strings that consist of a number of a's followed by the same number of b's.
L over $\{a, b\}$ and $L = \{a^n b^n \mid n \geq 0\}$

Solving . . .

1) $\varepsilon \in L$

2) $x \in L \Rightarrow axb \in L$

3) Nothing else is in L.

Recursive/inductive definitions in (programming) languages: *synthax*

“integers” (whole numbers)

<integer>::=<sign><natural> | <natural>
<natural>::= <digit>|<digit><natural>
<digit>::= 0|1|2|3|4|5|6|7|8|9
<sign>::= + | -

<integer>⇒<sign><natural>⇒ - <natural>⇒ - <digit><natural>
⇒ - 4<natural> ⇒ - 4<digit> ⇒ - 42

So - 42 is an interger…

synthax: statements

<assignment>::=<variable> = <expression>

<statement>::= <assignment>|<compound-statement>|
<if-statement> | <while-statement>

<if-statement>::=**if** <test> **then** <statement> |
if <test> **then** <statement> **else** <statement>

<while-statement>::=**while** <test> **do** <statement>

BNF: Backus-Naur form

Arithmetic expressions R

D = {0,1,2,3,4,5,6,7,8,9} also denoted ε = empty word

1) every element of $D^* - \{\lambda\}$ is in R

2) *if* $x \in R$, *then* $(-x) \in R$

if $x, y \in R$, *then* $(x + y) \in R, (x - y) \in R, (x * y) \in R, (x / y)\ in R$

[3) R has no other elements]

This defines a language. "+", "-", "*", "/" are symbols, with no intrisic meaning.

```
27
0014
-(0014)
((1+13)*8)
(27/(15+12-27))
(3-(-(-(5/7))))
```

Which are valid "arithmetic expressions"?

Try to not interpret…

In the definition of arithmetic expressions we make a distinction between

- the synthax (the *form*; the *valid strings; the arithmetic expression*)

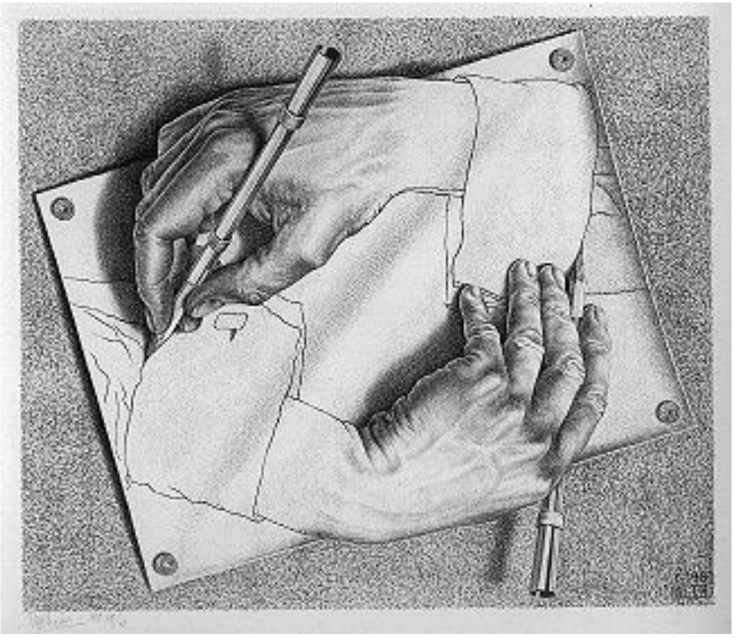
and

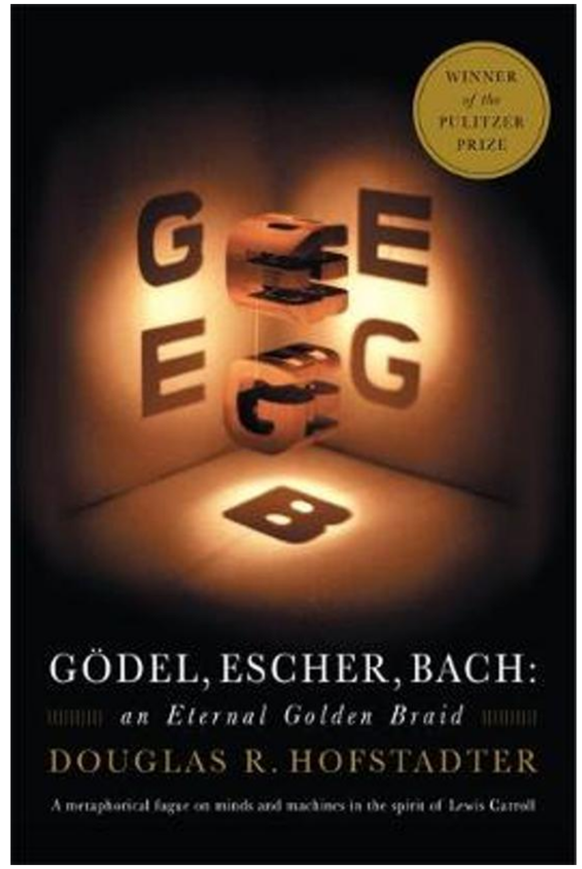
- semantics (the *meaning*; the interpretation; the value or an integer).

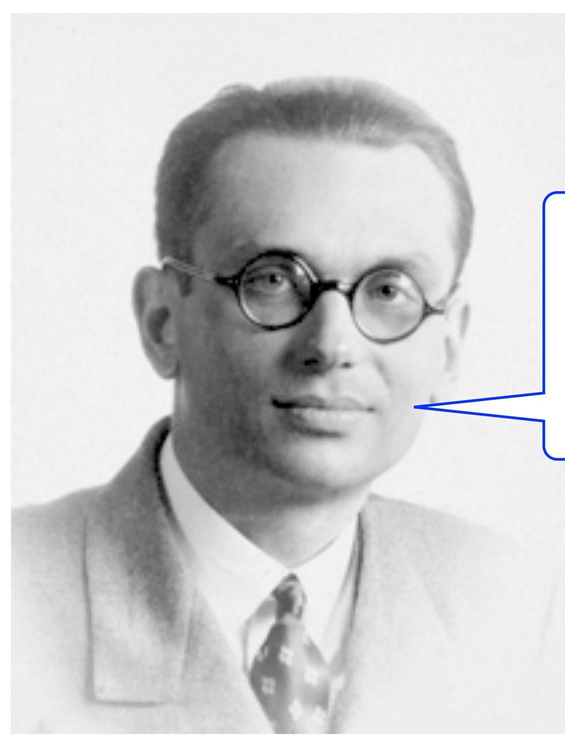
Based on the inductive syntax definition, we can define the semantics precisely, so that each syntactically correct string acquires a unique meaning.

See also §2.2

SKIPPED







This statement is unprovable

SKIPPED

WHAT'S THIS?
DOUGLAS HOFSTADTER'S SIX-WORD AUTOBIOGRAPHY.
AFTER ALL THOSE 700-PAGE TOMES, I GUESS HE WANTED TO TRY FOR BREVITY.
HUH. LET'S SEE...
I'M SO META, EVEN THIS ACRONYM
...WHOA.
I THINK HE NAILED IT.

Mu Puzzle

SKIPPED

1) $MI \in L$
2) *if* $xI \in L$, *then* $xIU \in L$
   if $Mx \in L$, *then* $Mxx \in L$
   if $xIIIy \in L$, *then* $xUy \in L$
   if $xUUy \in L$, *then* $xy \in L$
3) *L has no other elements*

MI

MIU **MII**

MIUIU **MIIU** **MIIII**

$M(IU)^4$ **$M(IIU)^4$** **MUI** **MI^8**

Mu Puzzle

SKIPPED

1) $MI \in L$
2) *if* $xI \in L$, *then* $xIU \in L$

   if $Mx \in L$, *then* $Mxx \in L$

   if $xIIIy \in L$, *then* $xUy \in L$

   if $xUUy \in L$, *then* $xy \in L$
3) ***L has no other elements***

Example:

MI→MII→MIIII→MIIIIIIII→MIIIIIIIIIIIIIIII→
MIIIIIIIUIIIIII→ MIIIIIIIUUIII→MIIIIIIIUUU→
MIIIIIIIUUUU→MIIIIIIIUU→MIIIIIII→MIIUII

Mu Puzzle

SKIPPED

1) $MI \in L$
2) $\textit{if } xI \in L, \textit{ then } xIU \in L$
   $\textit{if } Mx \in L, \textit{ then } Mxx \in L$
   $\textit{if } xIIIy \in L, \textit{ then } xUy \in L$
   $\textit{if } xUUy \in L, \textit{ then } xy \in L$
3) ***L has no other elements***

Example:

MI→MII→MIIII→MIIIIIIII→MIIIIIIIIIIIIIIII→
MIIIIIIIUIIIIII→ MIIIIIIIUUIII→MIIIIIIIUUU→
MIIIIIIIUUUU→MIIIIIIIUU→MIIIIIII→MIIUII

$$MU \overset{?}{\in} L$$

Recursion — digression

“self-similar” objects…

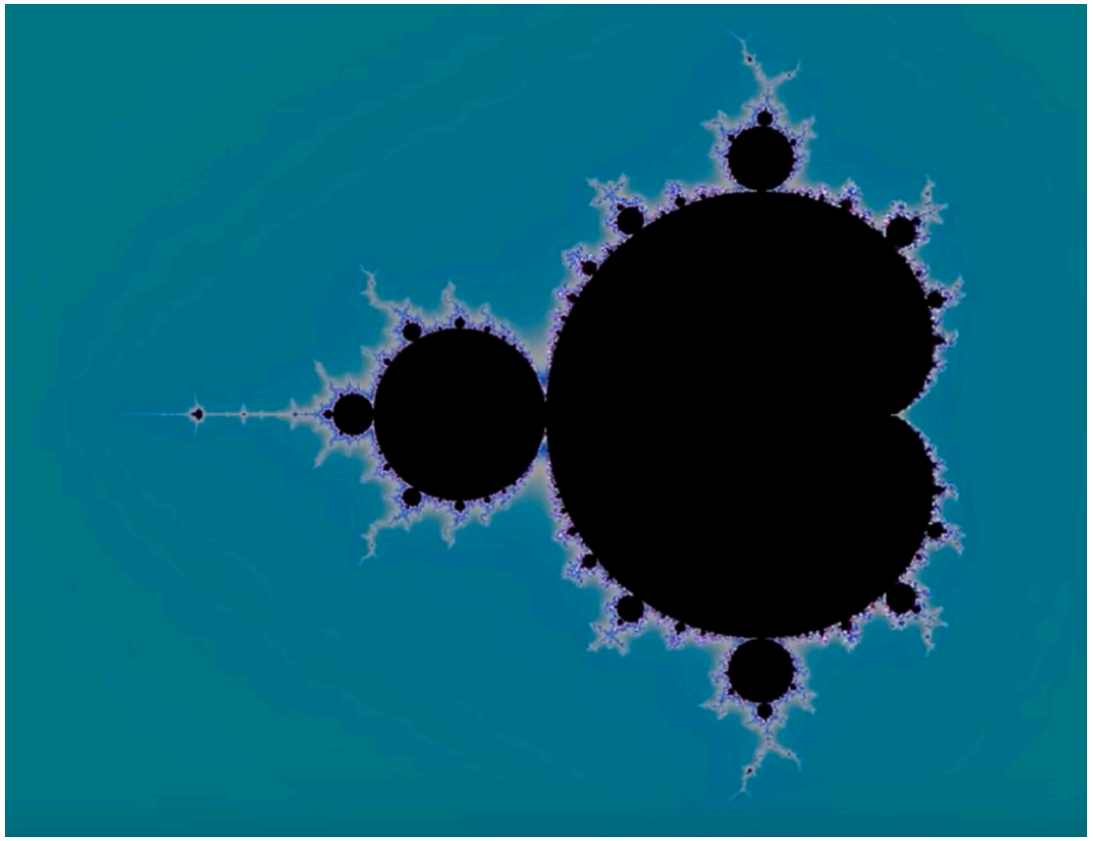
SKIPPED

Fractals are self-similar geometric objects.
One of the best known fractals is the Mandelbrot fractal.

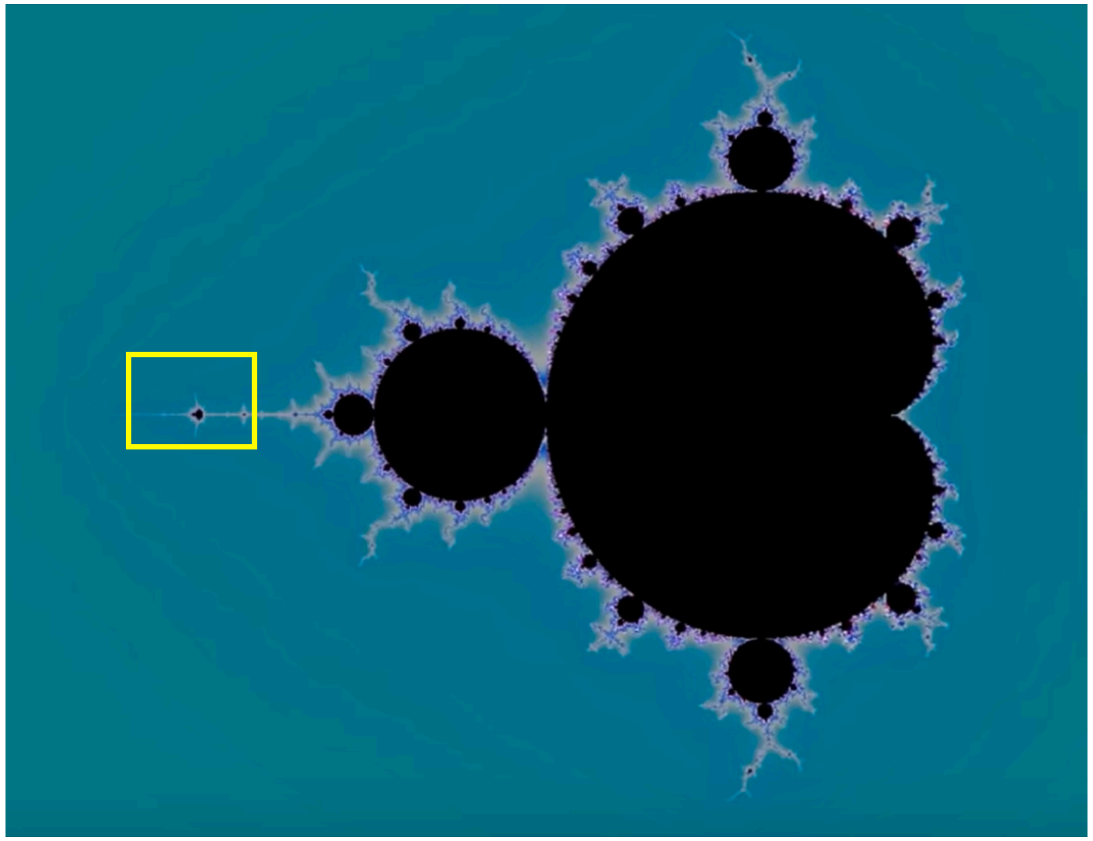
$$z_{n+1} = z_n^2 + c; z_0 = 0$$

The point c will be colored, if this sequence is not bounded from above. Color depends on when it grows above some value.

Unlike many other fractals, this figure does not repeat itself when zoomed. But it is self similar

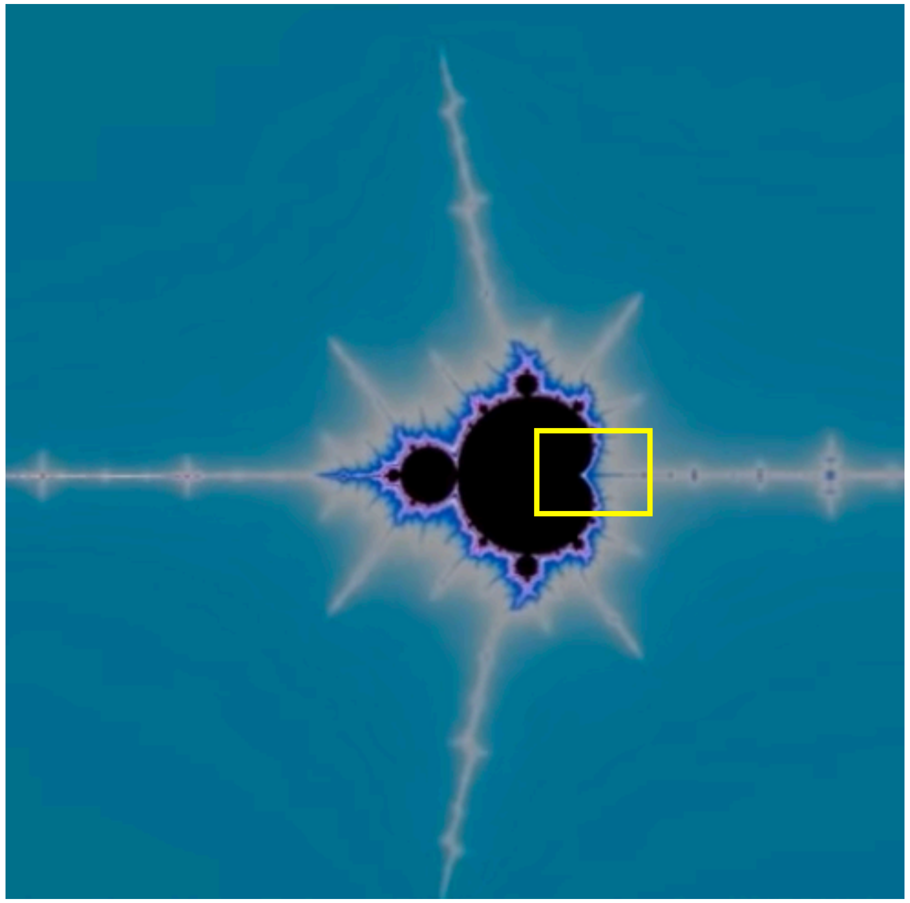


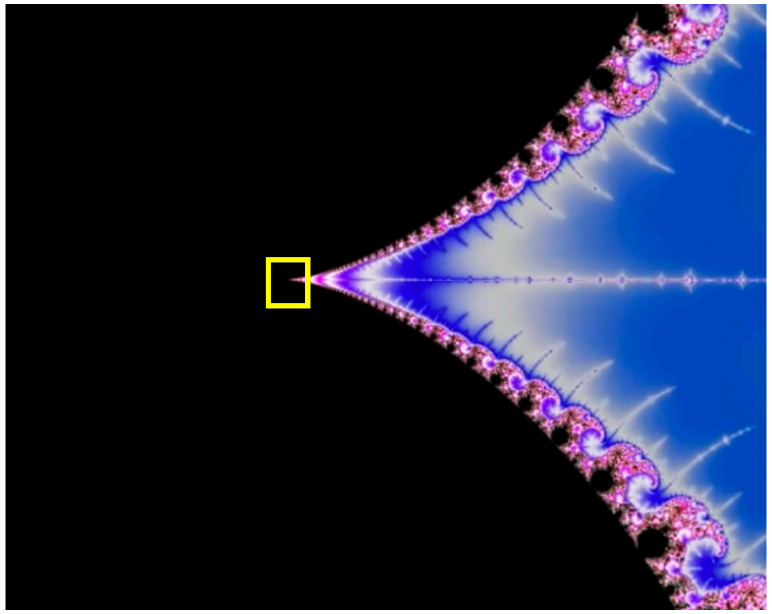
1



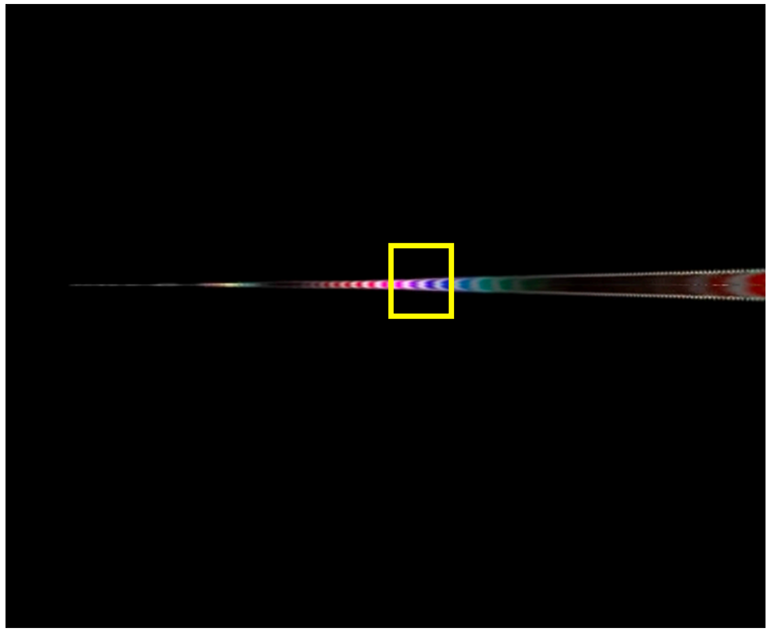
10

SKIPPED



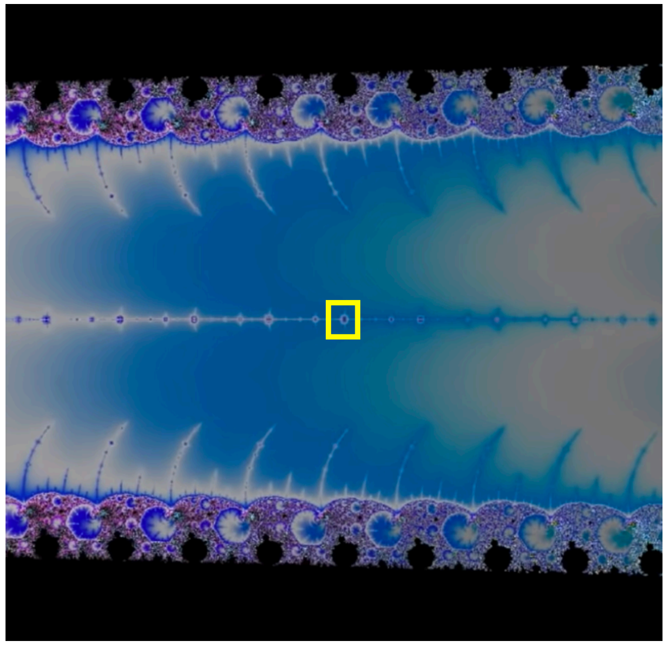


1.000

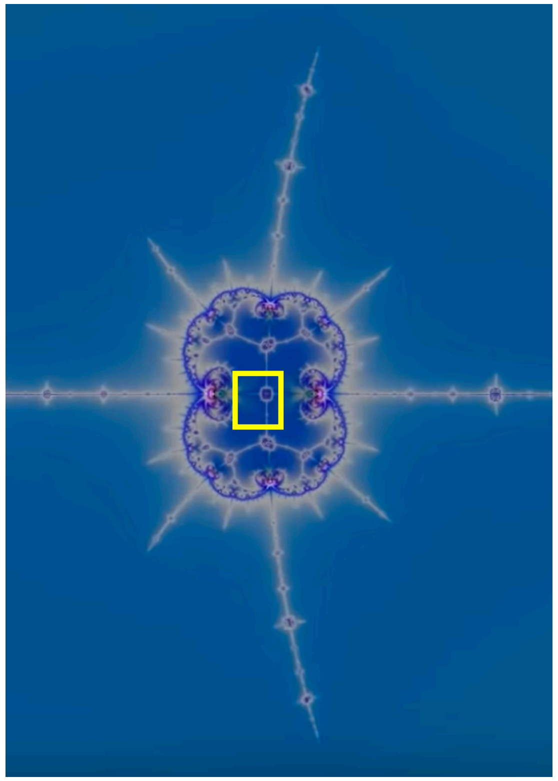


100.000

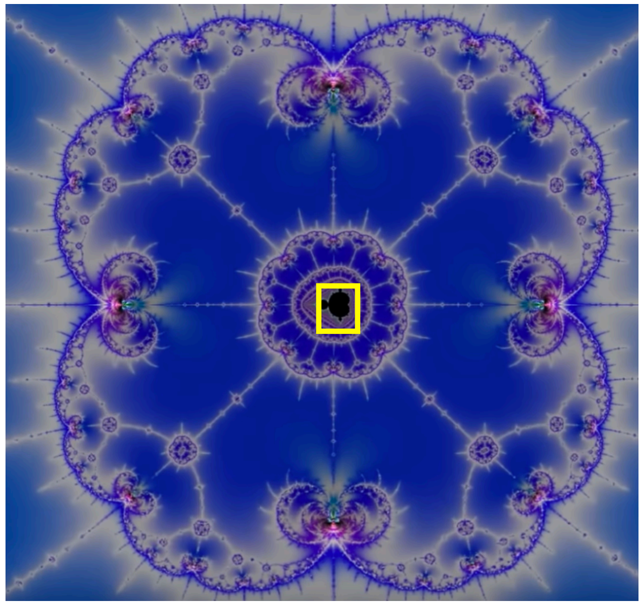
SKIPPED



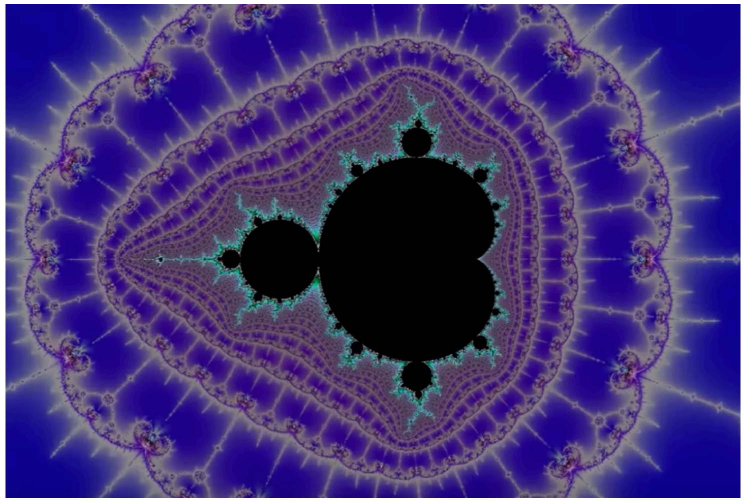
100.000.000



1.000.000.000



10.000.000.000

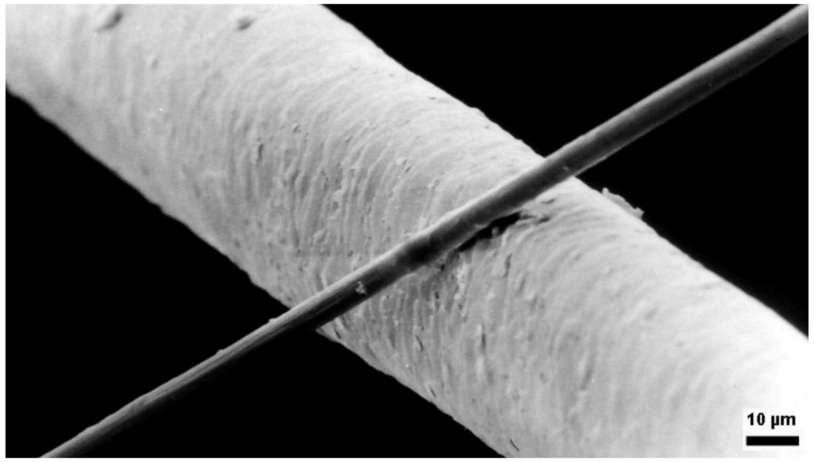


100.000.000.000

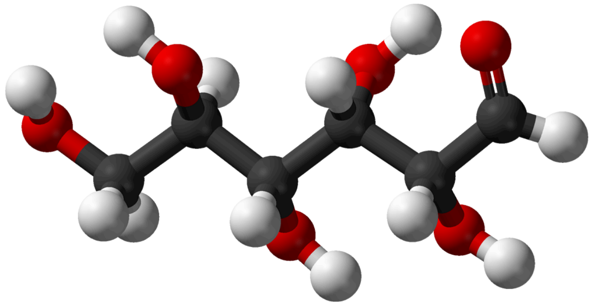
SKIPPED

1

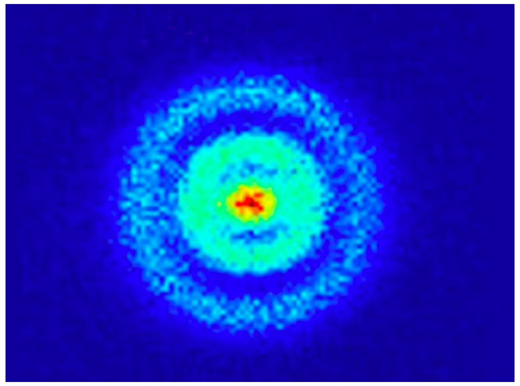
1000



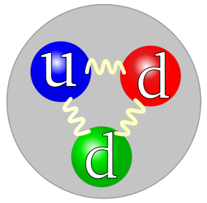
100.000



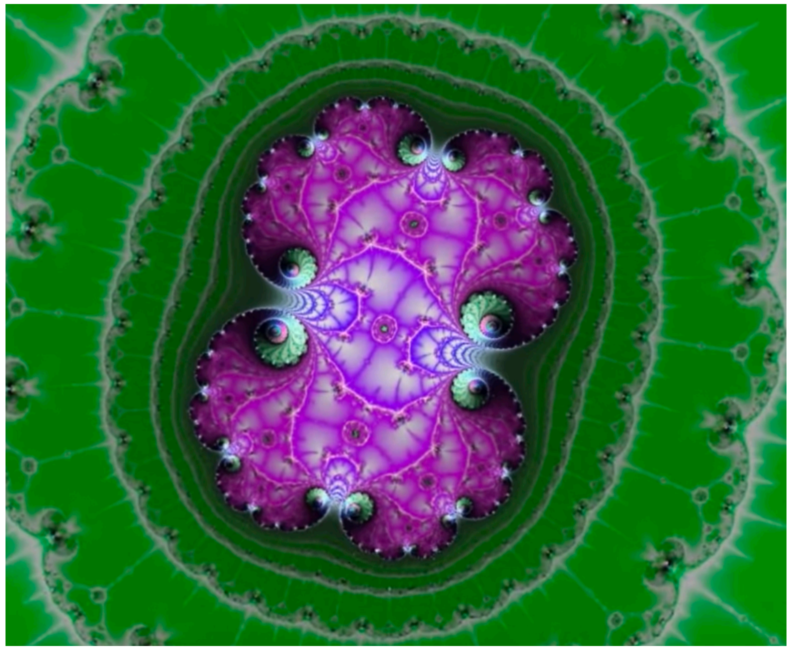
1.000.000.000



100.000.000.000

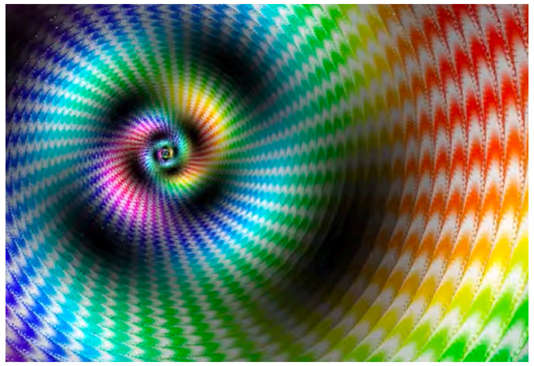


SKIPPED



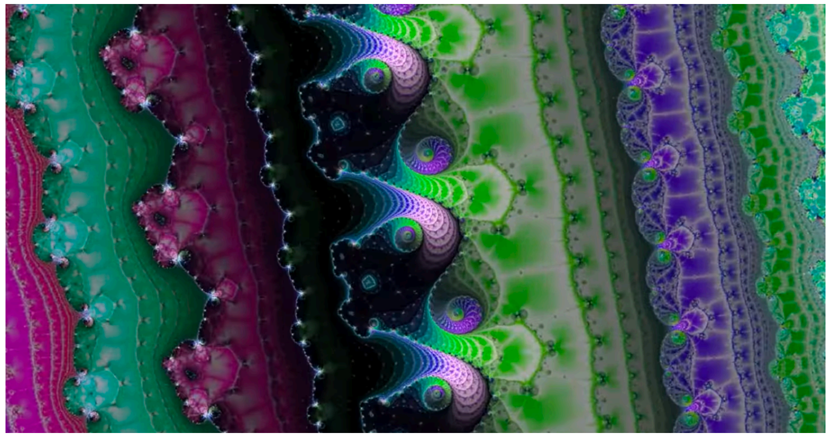
10.000.000.000.000.000 = 10^{16}

10^{22}



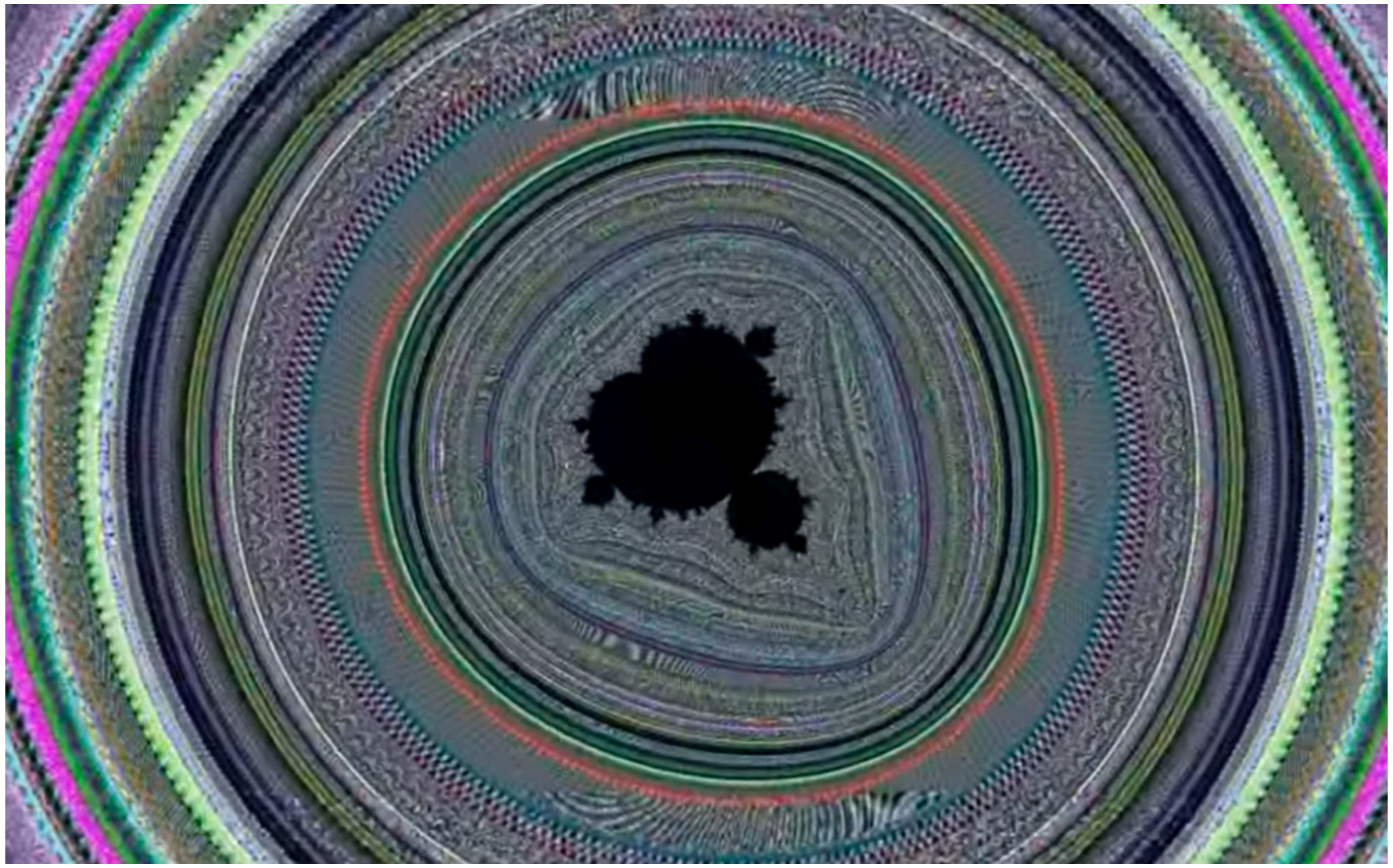
10^{40}

10^{44}



10^{76}

SKIPPED



10^{198}

Mu Puzzle

SKIPPED

1) $MI \in L$

2) **(a)** ***if*** $xI \in L$, *then* $xIU \in L$

   (b) ***if*** $Mx \in L$, *then* $Mxx \in L$

   (c) ***if*** $xIIIy \in L$, *then* $xUy \in L$

   (d) ***if*** $xUUy \in L$, *then* $xy \in L$

3) L has no other elements

Theorem. Every word in L begins with "*M*".

(i) <u>base step</u>

MI begins with M

(ii) <u>induction over the construction</u>

(a) if xI begins with M, then so does xIU
(b) Mxx begins with M
(c) if xIIIy begins with M then so does xUy
(d) if xUUy begins with M then so does xy

Principle of induction (structural)

Checking if some property P holds all elements of some ***inductively defined set*** *V*

(i) *base case*
ensure P holds for the basis of the induction of V

(ii) *inductive step (step case)*
Prove that P(y) holds for all y in V assuming that P (x) holds for all x from which y can be constructed

we again move to the "smaller" cases

Induction over size again.. so integers
"under the hood"..

Mu Puzzle

SKIPPED

1) $MI \in L$
2) **(a)** ***if*** $xI \in L$, *then* $xIU \in L$
   (b) ***if*** $Mx \in L$, *then* $Mxx \in L$
   (c) ***if*** $xIIIy \in L$, *then* $xUy \in L$
   (d) ***if*** $xUUy \in L$, *then* $xy \in L$
3) *L* ***has no other elements***

Theorem. The number of letters I in the word *w* of L is never divisible by 3.

Call this property P of the word w

(i) base step

MI has one "I"

(ii) induction over the construction

(a) if xI satisfies *P*, then so does xIU
(b) Mxx has a double number of Is as M, so *P holds*
(c) if xIIIy satisfies *P* then so does xUy
(d) if xUUy satisfies *P* then so does xy

Mu Puzzle

SKIPPED

1) $MI \in L$
2) (a) *if* $xI \in L$, *then* $xIU \in L$
   (b) *if* $Mx \in L$, *then* $Mxx \in L$
   (c) *if* $xIIIy \in L$, *then* $xUy \in L$
   (d) *if* $xUUy \in L$, *then* $xy \in L$
3) *L* ***has no other elements***

Theorem. The number of letters I in the word *w* of L is never divisible by 3.

Call this property P of the word w

0 <u>IS</u> divisible by 3.
P(MU) is <u>*false*</u>

Comment… properties like P which are maintained by the construcitons are called invariants…

Practice:

The Blurpsen set is the smallest set with the following properties:

(1) Δ is a Blurps.
(2) If x is a Blurps, then xΔΔ and ◊xx◊ are Blurps.
(3) If x and y are Blurps, then xΔy is also a Blurps.

Show that all (words in the laguage) Blurps have an odd number of triangles Δ or contain at least one diamond ◊.

(exercise 66 in exercise sheet)