Title: Can one measure complexity?

Date: May 26, 2005 02:20 PM

URL: http://www.pirsa.org/05050026

Abstract: Imagine doing mechanics without a precise notion of time, or thermodynamics without a definition of temperature. There is a huge recent upspring of "complex systems" research, with research institutes, journals and conferences devoted to it. Yet, there is no commonly agreed notion of what actually is "complexity". Can one give an operational definition of what is complexity, so that one can at least decide objectively and unambiguously whether a human is more complex than a bacterium? Or at least more complex than a stone? In my talk I want to give a review of attempts made during the last 30 years to define "complexity" in such a way that it agrees with the intuitive notion shared by most natural scientists. It will turn out that there are close connections to similar notions in computer science ("complexity of an algorithm") and in information theory ("algorithmic complexity"). There are subtleties which make it very unlikely that the above questions can ever be answered in the affirmative, but existing definitions of complexity can be useful when restricted to more narrowly limited problems.

L1

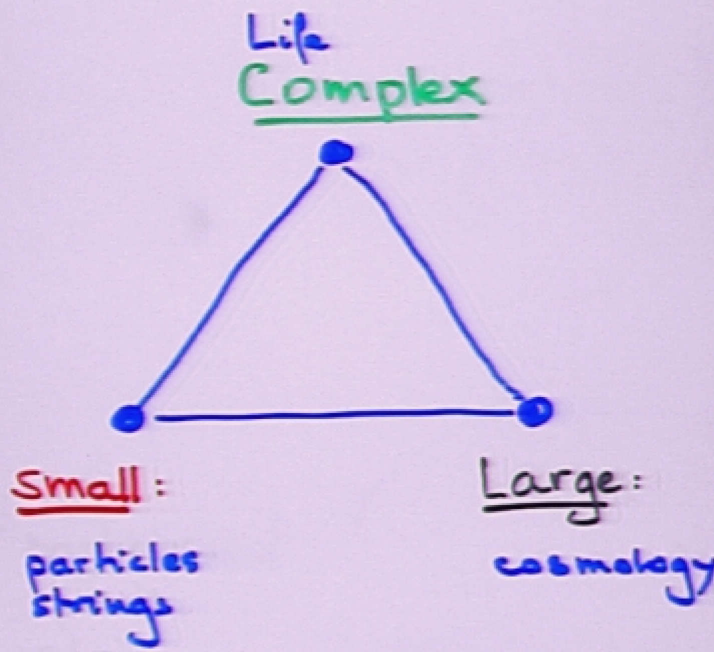
CAN ONE MEASURE Complexity ?

P. Grassberger
Complex Systems Group
John-von-Neumann Inst. for Computing
Jülich, Germany

P.I. may 26th, 2005

L2

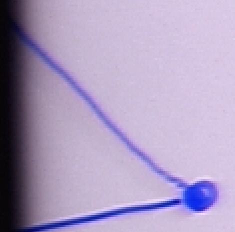
3 Frontiers of physics:



L2

es:

ex



Large:

cosmology

Traditional view:

- "complex" systems ≡ complicated

  ⇒ specialized sciences:

  chemistry
  biology
  economy
  history ...

- There is no common "science of complexity"

  progress only by looking at each phenomenon by itself

Traditional view:

- "complex" systems ≡ complicated

  ⇒ specialized sciences:

  chemistry
  biology
  economy
  history …

- There is no common "science of complexity"

  progress only by looking at each phenomenon by itself

Challenge:

- $\exists$ systems which are complex, without being complicated

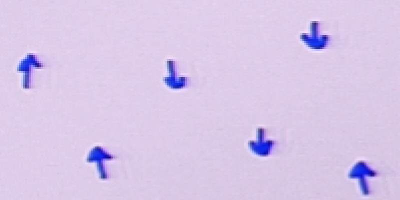
* deterministic chaos

$$x_{n+1} = a - x_n^2$$

$\Rightarrow$ unpredictable
fractals
...

$\rightarrow$ Figs. 1 – 3

* spin glasses



frozen disorder $\Rightarrow$ scaling laws
hierarchies
...

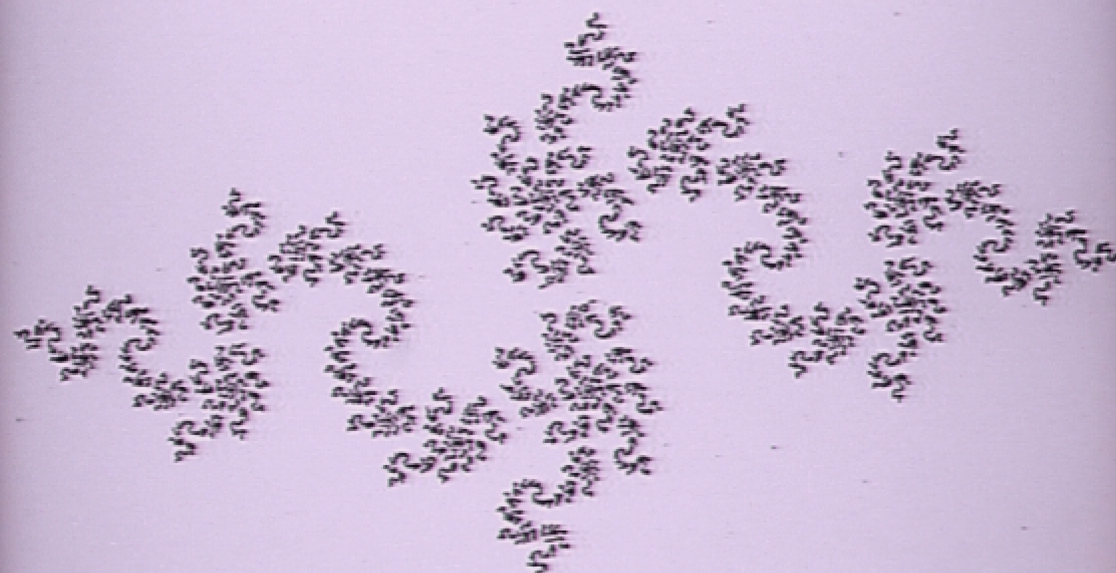


Fig.1: Julia set of the map $z' = z^2 - 0.86 - 0.25i$.

~~Fig 6a~~
Fig. 2

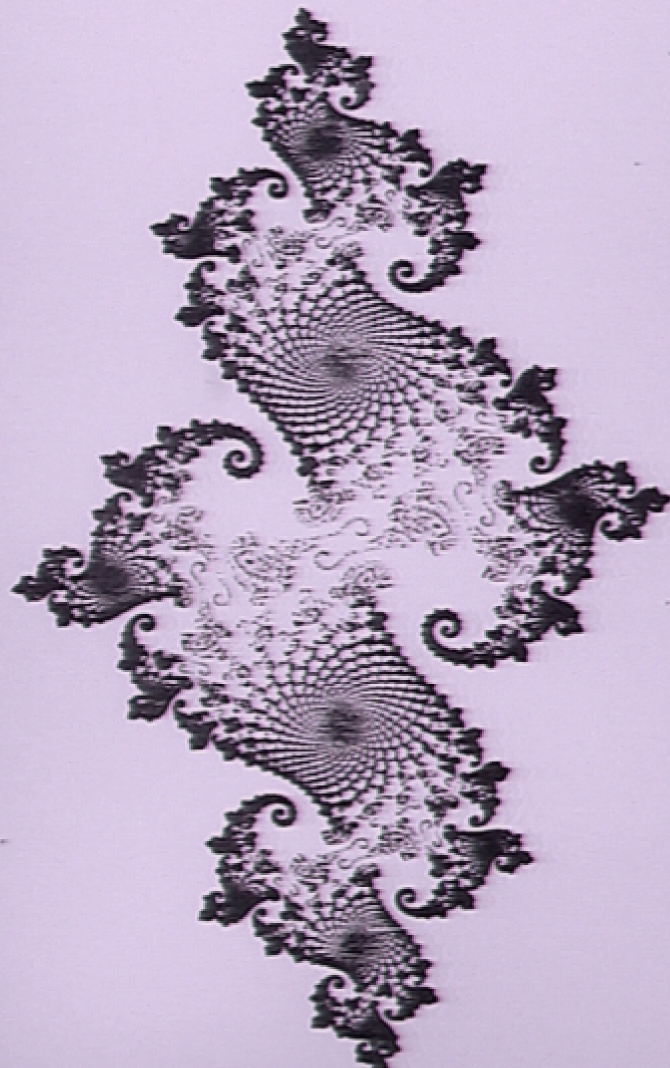


Fig. 3

Challenge:

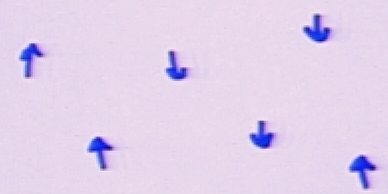
- ∃ systems which are **complex**, <u>without</u> being **complicated**

* deterministic chaos

$$x_{n+1} = a - x_n^2$$

⇒ unpredictable
fractals
...
→ Figs. 1 – 3

* spin glasses



frozen disorder ⇒ scaling laws
hierarchies
...

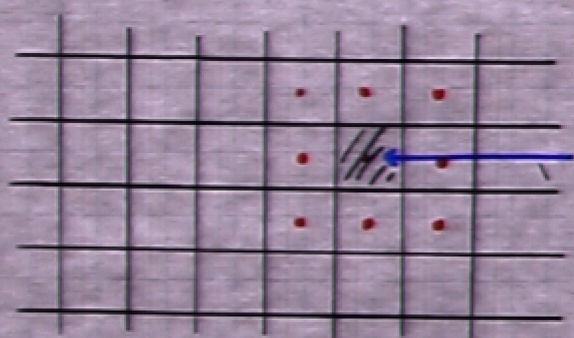
15

* Chess vs. "go"

chess : many rules

go : very few rules,
but at least as complex
strategies
(no good computer go yet)

* Game of "Life"
(H. Conway, M. Gardiner)



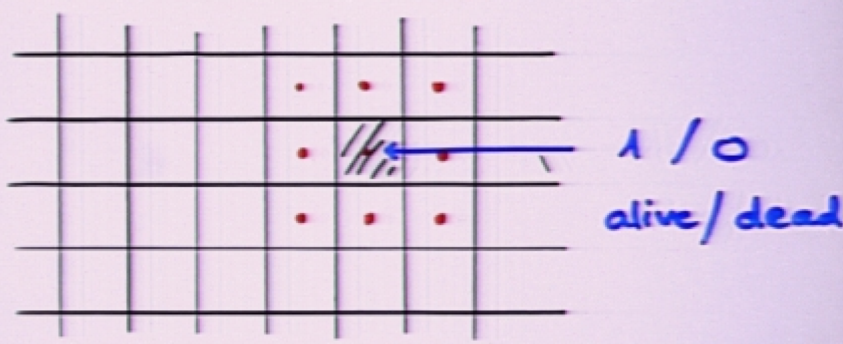
1 / 0
alive/dead

evolution $s_i(t) \rightarrow s_i(t+1)$

depends on 9-site neighbourhood :

go : very few rules,
but at least as complex
strategies
(no good computer go yet)

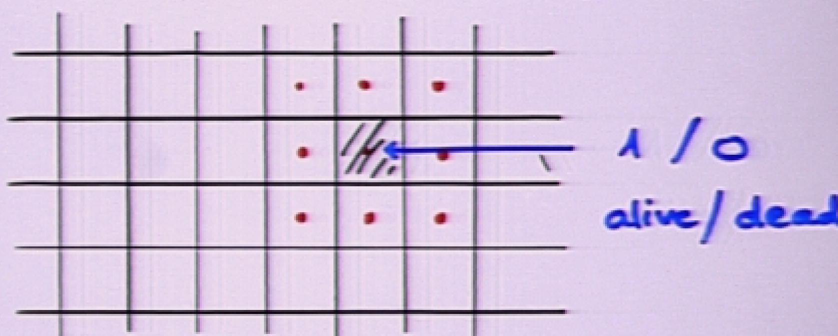
* Game of "Life"
(H. Conway, M. Gardiner)



evolution $s_i(t) \longrightarrow s_i(t+1)$
depends on 9-site neighbourhood :

go : very few rules,
but at least as complex
strategies
(no good computer go yet)

* Game of "Life"
(H. Conway, M. Gardiner)

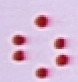


evolution $s_i(t) \longrightarrow s_i(t+1)$
depends on 9-site neighbourhood :

L5a

- >3 living neighbours:
  → death by crowding
- ≤ 1 living neighbours
  → death by loneliness
- 3 living neighbours
  → birth

 block

 beehive

 boat

.... etc.

} stationary

 blinker periodic

- ≤ 1 living neighbours
  → death by loneliness

- 3 living neighbours
  → birth

block
beehive
boat
.... etc.
} stationary

blinker — periodic

"glider" — periodic + shift

space ships
glider guns
glider gun factories
⋮

arbitrarily complex patterns

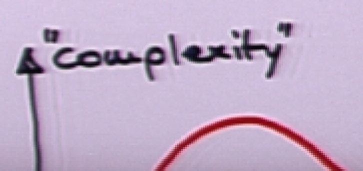
Conway et al:

Life is a universal computer

(→ any output after suitable input:

e.g. suitable input
⇒ proof of Fermat's last theorem)

But:



random →

space ships
glider guns
glider gun factories
⋮

arbitrarily complex patterns

Conway et al:

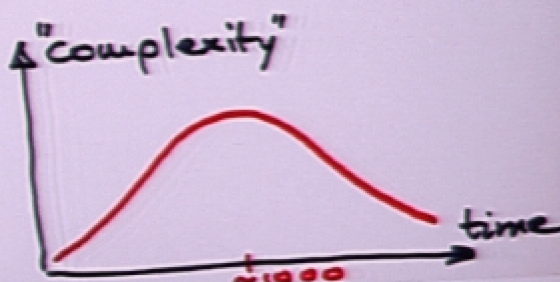
Life is a universal computer

(→ any output after suitable input:

e.g. suitable input
→ proof of Fermat's last theorem)

But:

random input →



arbitrarily complex patterns

Conway et al:

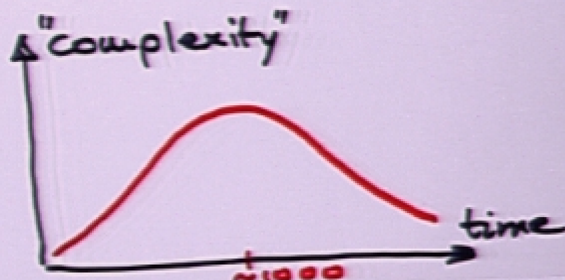
Life is a universal computer

(→ any output after suitable input:

e.g. suitable input
⇒ proof of Fermat's last theorem)

But:

random input →



N.B.:

how surprising is this ?

$$G_{\alpha\beta} = G\,T_{\alpha\beta}$$

$$\begin{cases} \partial_\mu F^\mu{}_\nu = j_\nu \\ \partial_\mu \tilde{F}^\mu{}_\nu = 0 \end{cases}$$

$$i\hbar\dot{\psi} = H\psi$$

all of these <u>simple</u>,

but $\longrightarrow$ very complex patterns !

Can there be a

(hopefully simple) theory of

COMPLEXITY per se ?

What is complexity ?

?

Is there an unambiguous, objective way to rank

objects (phenomena

Can there be a

(hopefully simple) theory of

COMPLEXITY per se ?

What is complexity ?

?

Is there an unambiguous, objective way to rank objects (phenomena, behaviour, ...) by complexity ?

Are YOU more complex than an insect (a bacterium? a virus?)

If no objective answer possible (i.e., if $\nexists$ quantitative measure of complexity) :

can there ever be a really systematic theory of complexity?

Imagine :

- thermodynamics without precise definition of "cold"/"hot"

- mechanics without an objective way to say what means "heavy", "fast", ...

Quotations (all from same paper):

"... why systems may be very complex. First, ... many elements ... :

brain	~	$10^{11} - 10^{12}$	neurons
world	~	10^{10}	people
laser	~	10^{18}	photons
fluid	~	10^{23}	molecules"

→ is fluid more complex than world?

"... the brain is the most complex system ..."

"... an enormous amount of information ..."

Quotations (all from same paper):

"... why systems may be very complex.
First, ... many elements ... :

brain	~	$10^{11} - 10^{12}$	neurons
world	~	10^{10}	people
laser	~	10^{18}	photons
fluid	~	10^{23}	molecules"

→ is fluid more complex than world?

"... the brain is the most complex system ..."

"... an enormous amount of information ..."

Mathematics:

* C. of an algorithm
  - time complexity of algorithm A:

    $\hat{=}$ CPU time needed to perform A
  - space complexity:

    $\hat{=}$ needed storage in fast memory

    objects (phenomena, behavior) are not algorithms

* Kolmogorov - Chaitin ("algorithmic") complexity

  U = universal computer (i.e., PC)

  S = string of "letters" (e.g. binary)

  $C_U(S)$ = length of shortest program

* C. of an algorithm

- time complexity of algorithm A:

  $\hat{=}$ CPU time needed to perform A

- space complexity:

  $\hat{=}$ needed storage in fast memory

objects (phenomena, behavior) are not algorithms

* Kolmogorov - Chaitin ("algorithmic") complexity

U = universal computer (i.e., PC)
S = string of "letters" (e.g. binary)

$C_U(S)$ = length of shortest program

$\hat{=}$ CPU time needed to perform A

- space complexity :

  $\hat{=}$ needed storage in fast memory

objects (phenomena, behaviour) are not algorithms

* Kolmogorov - Chaitin ("algorithmic") complexity

U = universal computer (i.e., PC)

S = string of "letters" (e.g. binary)

$C_U(S)$ = length of shortest program

• space complexity :

$\hat{=}$ needed storage in fast memory

objects (phenomena, behavior) are not algorithms

* Kolmogorov - Chaitin ("algorithmic") complexity

U = universal computer (i.e., PC)

S = string of "letters" (e.g. binary)

$C_U(S)$ = length of shortest program

such that U first prints S and then stops

example: U = your PC
S = text of "Moby Dick"

$C_U(S) \leq$ length of moby_dick.zip + length of "unzip .... lpr ... "

<u>If length of $S \to \infty$:</u>

$C_U(S)$ becomes basically independ. of U (up to length of emulation program) such that U first prints S and then stops

example: U = your PC
S = text of "Moby Dick"

$C_U(S) \leq$ length of moby_dick.zip + length of "unzip ... lpr ..."

<u>If length of $S \to \infty$:</u>

$C_U(S)$ becomes basically independ. of U (up to length of emulation program)

Attempt #1: $\lim\limits_{|S| \to \infty} \frac{1}{|S|} C_U(S) = c(S)$

= complexity per letter of S

Main drawback:

If S can be random

$\Rightarrow$ $C(S)$ is measure of randomness

(if S drawn from prob. distrib., then $\langle C(S) \rangle$ = Shannon entropy)

sequences which are not random:

- 3.14592653589793238462 64...
- 1.41421356 23731...
- DNA of A. Einstein
- computer code for selling airline tickets

$\Rightarrow$ for these, $C(S)$ is good complexity measure

If S can be random

$\Rightarrow$ $c(S)$ is measure of randomness

(If S drawn from prob. distrib., then $\langle c(S) \rangle$ = Shannon entropy)

sequences which are not random:

- 3.14592653589793238462 64...
- 1.41421356237 31...
- DNA of A. Einstein
- computer code for selling airline tickets

$\Rightarrow$ for these, $C(S)$ is good complexity measure

If S can be random

$\Rightarrow$ $c(S)$ is measure of randomness

(if S drawn from prob. distrib., then $\langle c(S) \rangle$ = Shannon entropy)

sequences which are not random:

- 3.1459265358979323846264...
- 1.414213562373l...
- DNA of A. Einstein
- computer code for selling airline tickets

$\Rightarrow$ for these, $C(S)$ is good complexity measure

natural sciences / daily life :

are random objects complex ?

→ Fig. 4

Ferrari = complex car

Ferrari after accident :

- symmetries lost
- windshield broken

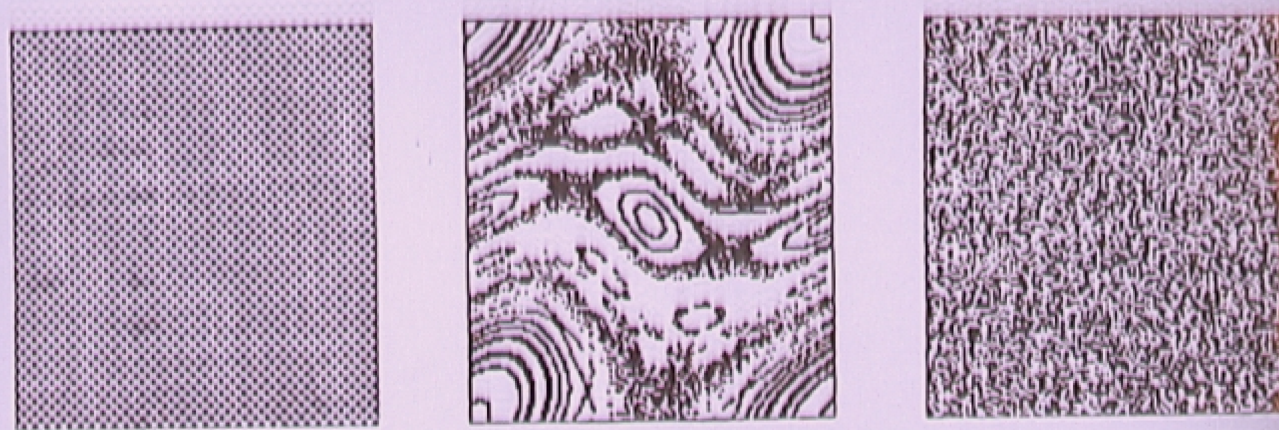
⋮

⇒ even more complex ?

What is more complex :

a cow ?

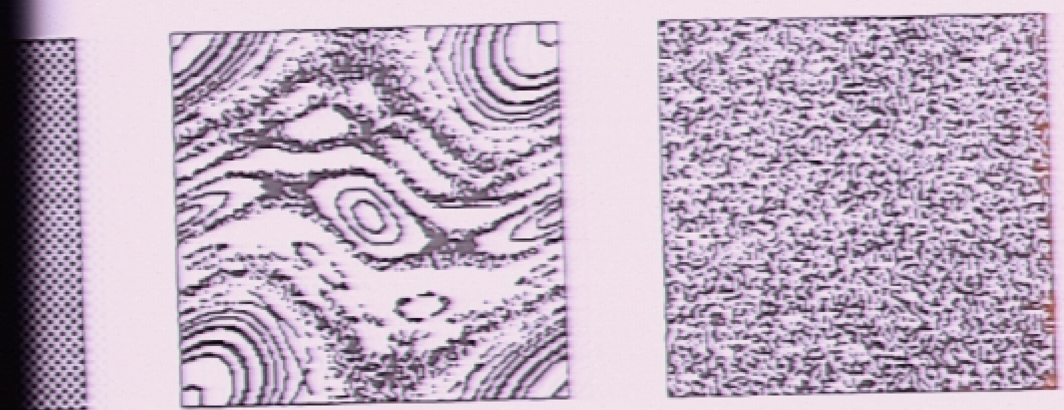
ground beef ?



a) b) c)

complex ?

Fig. 4



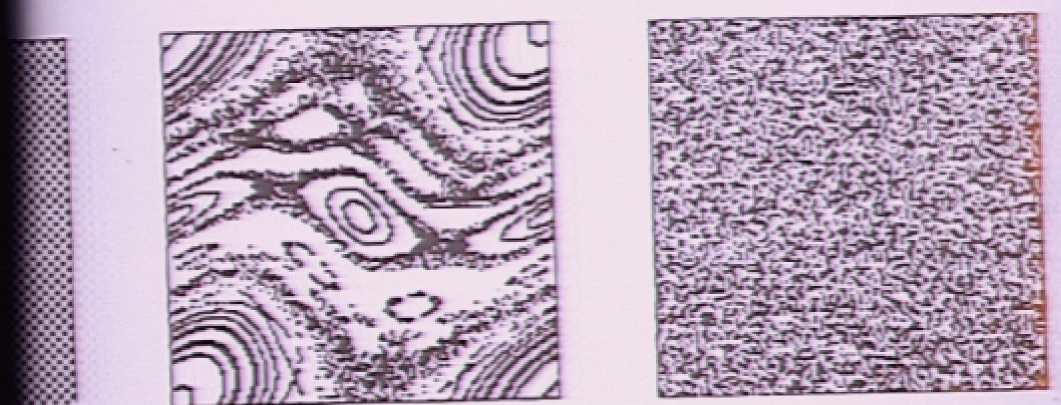
a) b) c)

complex ?

maybe this is Fig. co...

~~a~~ random num...

Fig. 4



a) b) c)

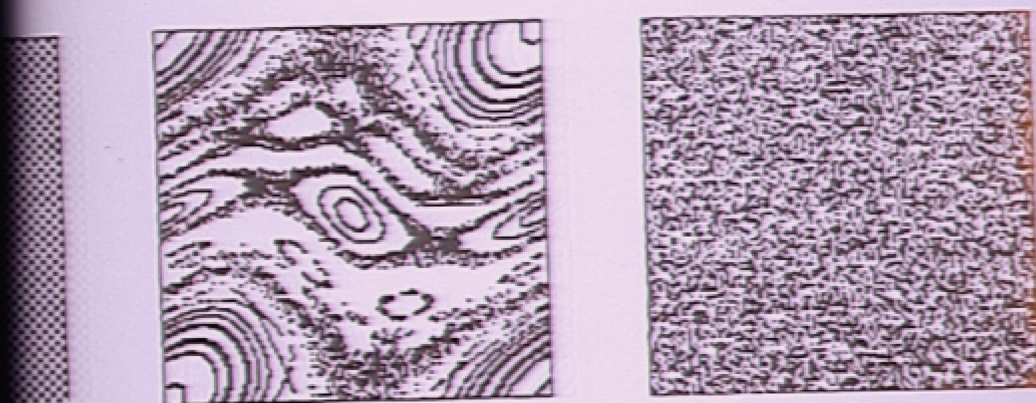
complex ?

Fig.

maybe this is

~~≡~~ random

Fig.



a) b) c)

complex ?

maybe this is Fig. 1 co...

~~∃~~ random num...

natural sciences / daily life :

are random objects complex ?

→ Fig. 4

Ferrari = complex car

Ferrari after accident :

- symmetries lost
- windshield broken

⋮

⇒ even more complex ?

What is more complex :

a cow ?

ground beef ?

Ferrari = complex car

Ferrari after accident:

- symmetries lost
- windshield broken

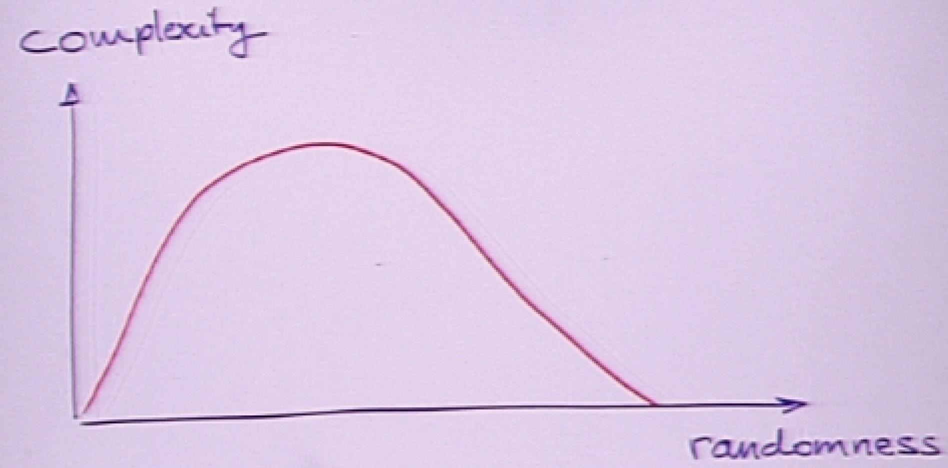
$\vdots$

$\Rightarrow$ even more complex?

What is more complex:

a cow?

ground beef?



Hogg & Huberman

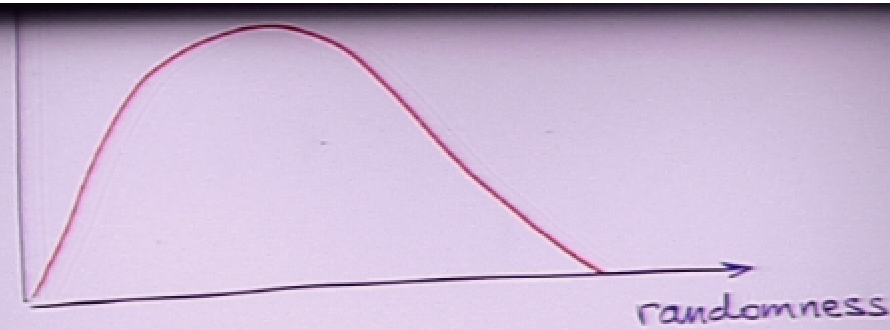
Crutchfield & Young

"complexity = between order & chaos"

" = at the edge of chaos

→ huge ~~nonsense~~ literature on complexity measures:

Lopez-Ruiz, Mancini, Calbet
Landsberg, Shiner, Davison
Andrienko et al



Hogg & Huberman

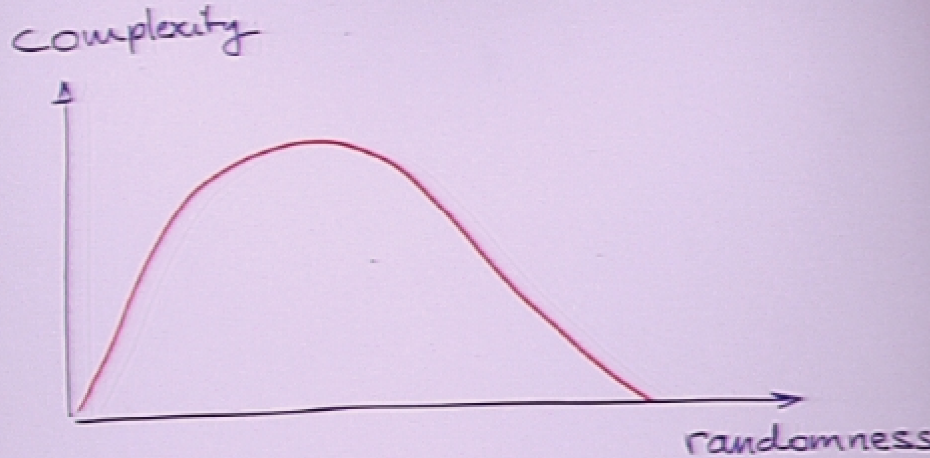
Crutchfield & Young

"complexity = between order & chaos"

" = at the edge of chaos

→ huge <u>nonsense</u> literature on complexity measures:

Lopez-Ruiz, Mancini, Calbet
Landsberg, Shiner, Davison
Andrienko et al
Chen, Xu, Gu, Meng, Liu, ...



Hogg & Huberman

Crutchfield & Young

"complexity = between order & chaos"

" = at the edge of chaos

→ huge nonsense literature on complexity measures:

Lopez-Ruiz, Mancini, Calbet
Landsberg, Shiner, Davison
Andrienko et al
Chen, Xu, Gu, Meng, Liu, ...

Objects are never completely characterized, in practice

(position of every quark ?
in every nucleus ?
in every molecule ?
precisely now ?)

"Macrostate" vs. "microstate"

object $\in$ ensemble

Information needed to specify object
= information for ensemble
+ information for object when ensemble is already specified

Attempt #2:

complexity = "meaningful" information
≡ inform. about ensemble

Objects are never completely characterized, in practice

(position of every quark ?
in every nucleus ?
in every molecule ?
precisely now ?)

"Macrostate" vs. "microstate"

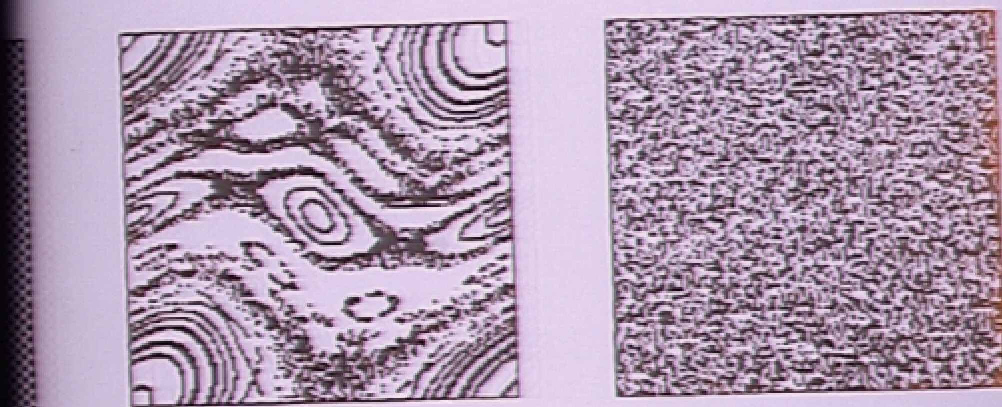
object $\in$ ensemble

Information needed to specify object
= information for ensemble
+ information for object when ensemble is already specified

Attempt #2:

complexity = "meaningful" information
$\equiv$ inform. about ensemble

Fig. 4


a) b) c)

complex ?

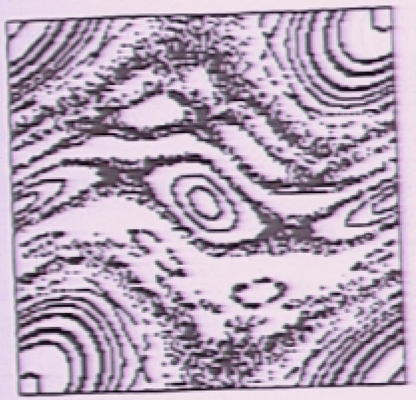
maybe this is complex

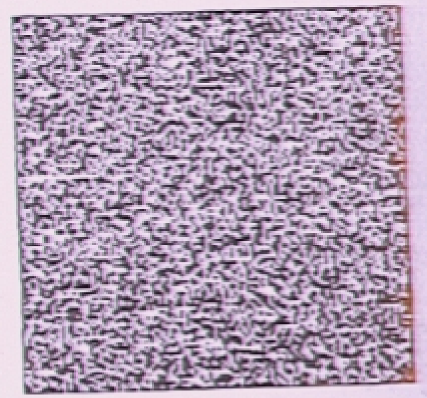
~~$\exists$~~ random number

Fig. 1

Fig. 4


a)


b)


c)

complex ?

maybe this is Fig. 1 complex

~~=~~ random number

(position of every quark ?
in every nucleus ?
in every molecule ?
precisely now ?)

"Macrostate" vs. "microstate"

object $\in$ ensemble

Information needed to specify object
= information for ensemble
+ information for object when ensemble is already specified

Attempt #2:

complexity = "meaningful" information
$\equiv$ inform. about ensemble

(position of every quark ?
in every nucleus ?
in every molecule ?
precisely now ?)

"Macrostate" vs. "microstate"

object $\in$ ensemble

Information needed to specify object
= information for ensemble
+ information for object when ensemble is already specified

Attempt #2:

complexity = "meaningful" information
$\equiv$ inform. about ensemble

(position of every quark ?
in every nucleus ?
in every molecule ?
precisely <u>now</u> ?)

"Macrostate" vs. "microstate"

object $\in$ ensemble

Information needed to specify object
= information for ensemble
+ information for object when ensemble is already specified

<u>Attempt #2:</u>

complexity = "meaningful" information
$\equiv$ inform. about ensemble

Problem:

micro/macro distinction o.k.
for gas
liquid
crystal ...

what about brain ?

single synapse : micro or macro ?

most simple:
2-step hierarchy micro/macro

intermediate:
strict hierarchy (by scales or logical)

most complex:
"tangled" hierarchies

micro / macro distinction o.k.

for gas
liquid
crystal ...

what about brain ?

single synapse : micro or macro ?

most simple :
2-step hierarchy micro / macro

intermediate :
strict hierarchy (by scales or logical)

most complex :
"tangled" hierarchies

for gas
liquid
crystal ...

what about brain ?

single synapse : micro or
macro ?

most simple :
2-step hierarchy micro / macro

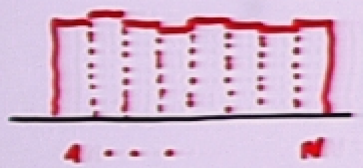
intermediate :
strict hierarchy (by scales or
logical)

most complex :
"tangled" hierarchies

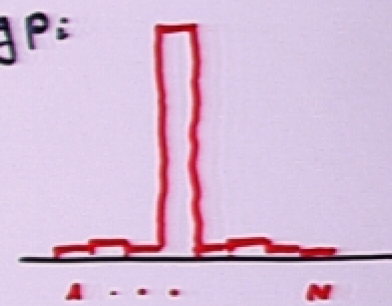
Shannon theory:

"object" = random variable,
i.e. has prob's assigned

$H = \langle \text{information} \rangle$
$= -\sum_i p_i \log p_i$



H large



H small

J. Rissanen:

<u>total</u> information should include specification of $\{p_i\}$

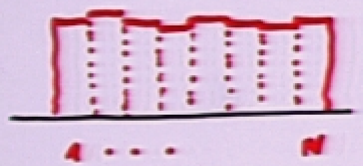
– precise specification → ∞ many digits
⇒ total inform. = ∞

⇒ # of digits for p_i should depend

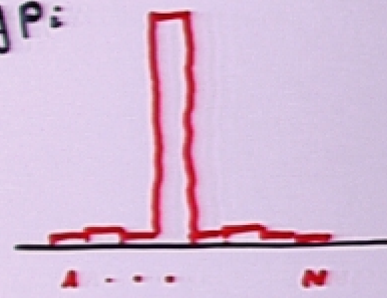
Shannon theory:

"Object" = random variable,
i.e. has prob's assigned

$H = \langle \text{information} \rangle$
$= -\sum_i p_i \log p_i$



H large



H small

J. Rissanen:

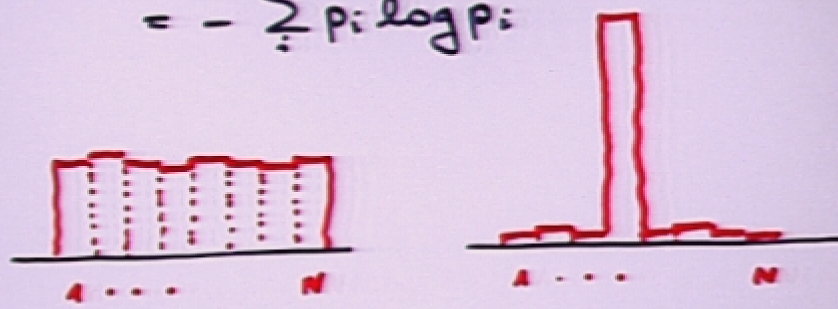
total information should include specification of $\{p_i\}$

– precise specification → ∞ many digits
⇒ total inform. = ∞

⇒ # of digits for p_i should depend

i.e. has prob's assigned

$$H = \langle \text{information} \rangle$$
$$= -\sum_i p_i \log p_i$$



H large   H small

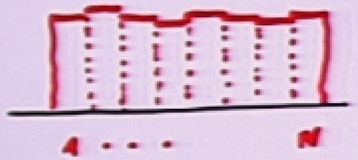
J. Rissanen:

total information should include specification of $\{p_i\}$

– precise specification → ∞ many digits
⇒ total inform. = ∞

⇒ # of digits for p_i should depend

$$H = \langle \text{information} \rangle$$
$$= - \sum_i p_i \log p_i$$



H large

H small

J. Rissanen:

<u>total</u> information should include specification of $\{p_i\}$

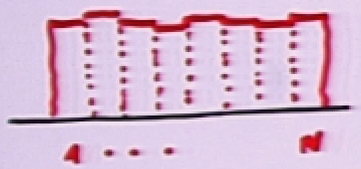
– precise specification → ∞ many digits
⇒ total inform. = ∞

⇒ # of digits for p_i should depend

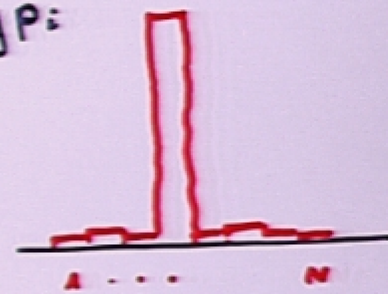
Shannon theory:

"Object" = random variable,
i.e. has prob's assigned

$$H = \langle \text{information} \rangle = -\sum_i p_i \log p_i$$



H large



H small

J. Rissanen:

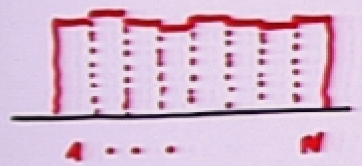
<u>total</u> information should include specification of $\{p_i\}$

~ precise specification → ∞ many digits
⇒ total inform. = ∞

⇒ # of digits for p_i should depend

i.e.: has prob's assigned

$$H = \langle \text{information} \rangle$$
$$= - \sum_i p_i \log p_i$$



H large

H small

J. Rissanen:

<u>total</u> information should include specification of $\{p_i\}$

– precise specification → ∞ many digits
⇒ total inform. = ∞

⇒ # of digits for p_i should depend

J. Rissanen:

total information should include specification of $\{p_i\}$

– precise specification → ∞ many digits
⇒ total inform. $= \infty$

⇒ # of digits for p_i should depend

18

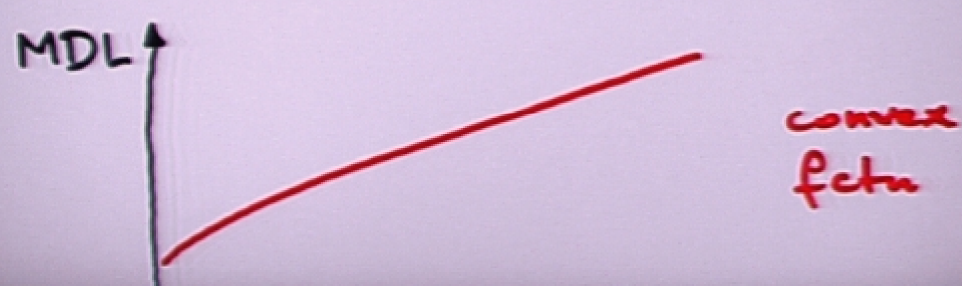
on length of object

optimal (= minimal total inform.):

$$\# \text{ digits} \propto \log N$$

"Minimum Description Length"
MDL

Object = sequence of N letters:



L18

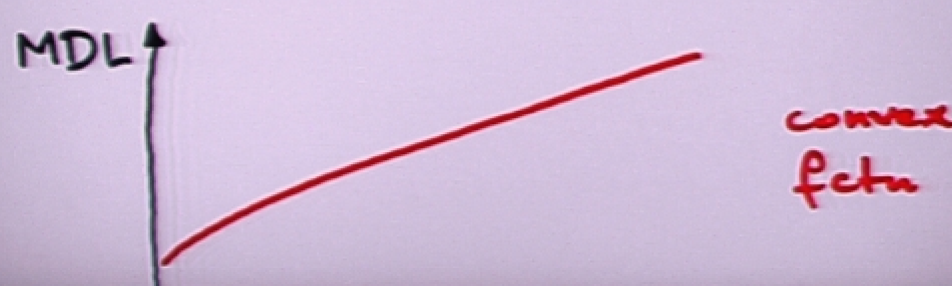
on length of object

optimal (= minimal total inform.):

digits $\propto$ log H

"Minimum Description Length"
MDL

Object = sequence of N letters:



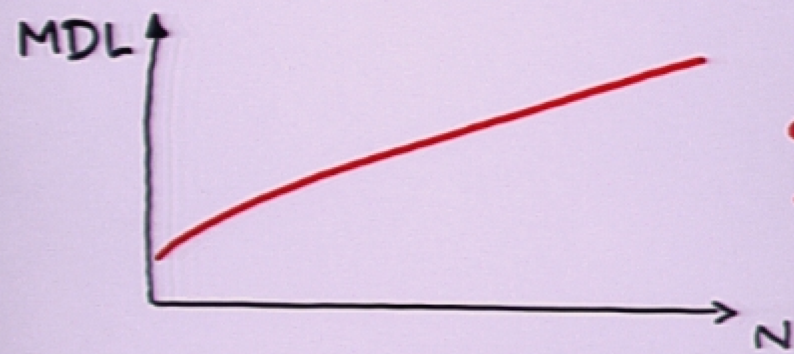
on length of object

optimal (= minimal total inform.):

digits $\propto$ log H

"Minimum Description Length"
MDL

Object = sequence of N letters:



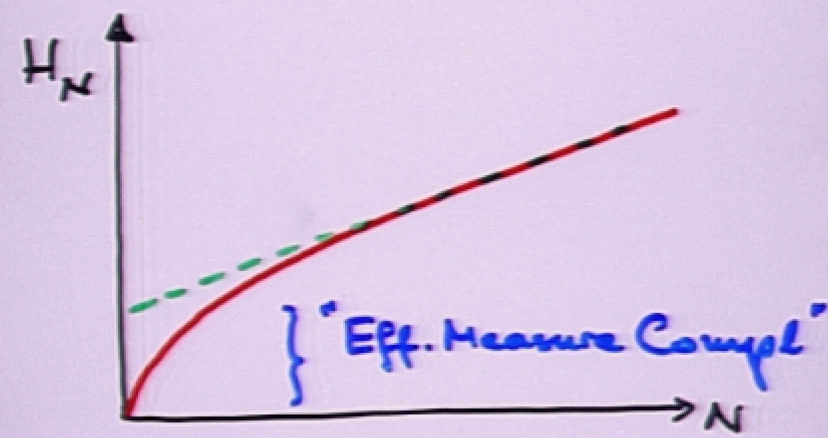
convex fctn

Similar curves for $C_u(S)$ versus N:

$$C_u(S_1 \circ S_2) \leq C_u(S_1) + C_u(S_2)$$

concatenation

Similar for Shannon entropy:



(19

Similar curves also for

$H(S)$ versus N (Shannon)

$H(S_1 \circ S_2) \leq H(S_1) + H(S_2)$

Def.:

$$MI(S_1 : S_2) = H(S_1) + H(S_2) - H(S_1 \circ S_2)$$

& similar for algor. version

"mutual information"

Attempt #3:

$C = MI$, eventually suitably averaged over most relevant binary partitions of object

$$H(S_1 \circ S_2) \leq H(S_1) + H(S_2)$$

Def.:

$$MI(S_1 : S_2) = H(S_1) + H(S_2) - H(S_1 \circ S_2)$$

& similar for algor. version

"mutual information"

Attempt #3:

C = MI , eventually suitably averaged over most relevant binary partitions of object

R. Shaw, G. Chaitin, P.G., G. Tononi et al, M. Gell-Mann, ...

Def.:

$$MI(S_1 : S_2) = H(S_1) + H(S_2) - H(S_1 \circ S_2)$$

& similar for algor. version

"mutual information"

Attempt #3:

C = MI, eventually suitably averaged over most relevant binary partitions of object

R. Shaw, G. Chaitin, P.G., G. Tononi et al., M. Gell-Mann, ...

"Object is complex, if total ≠ sum of parts"

Similar:

- object is complex, if there are correlations between itself & its environment
- ... if it can do meaningful tasks (= it "fits" into environment)

Applications of MI numerous !!

Relevant { partitionings
averaging over ...
...

depend on applications !

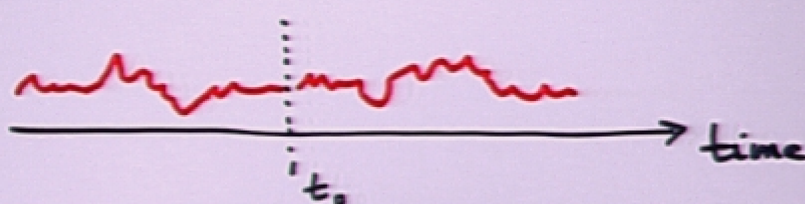
Tononi et al :

complexity of brain states

brain = interconnected on all scales

⇒ average over <u>all</u> binary partitionings of the brain

Time sequences (Shaw, P.G.) :



partitionings past / future

Relation with Forecasting

(C. of Forecasts)

... 1111111 ...

... 101010 ... } easy to forecast

completely random — optimal forecast is also easy (pure guess)

non-trivial correlations — hard

$x_{n+1} = a - x_n^2$:

$a \leq 1.4..$: periodic, easy

$a = 2$: good forecast impossible, but best is easy

Relation with Forecasting

(C. of Forecasts)

... 1111111 ...
... 101010 ... } easy to forecast

completely random — optimal forecast is also easy (pure guess)

non-trivial correlations — hard

$x_{n+1} = a - x_n^2$:

$a \leq 1.4..$: periodic, easy

$a = 2$: good forecast impossible, but best is easy

... 101010 ...

completely random — optimal forecast is also easy (pure guess)

non-trivial correlations — hard

$x_{n+1} = a - x_n^2$:

$a \leq 1.4..$: periodic, easy

$a = 2$: good forecast impossible, but best is easy

wheather forecasts:

- how good ?
- how difficult ?

- good forecast possible :
  ↔ not random

- best forecast is difficult
  ↔ complex

why can forecast be difficult :

strong correlations between future & past

More precise (P.G.) :

F.C. = ⟨ information about past, needed for forecast ⟩

≥ ⟨ MI × distance ⟩ = E.M.C

- good forecast possible :
  $\leftrightarrow$ not random

- best forecast is difficult
  $\leftrightarrow$ complex

why can forecast be difficult :

strong correlations between future & past

More precise (P.G.) :

F.C. $\equiv$ < information about past, needed for forecast >

$\geq$ < MI $\times$ distance > $\equiv$ E.M.C

- good forecast possible :
  $\leftrightarrow$ not random

- best forecast is difficult
  $\leftrightarrow$ complex

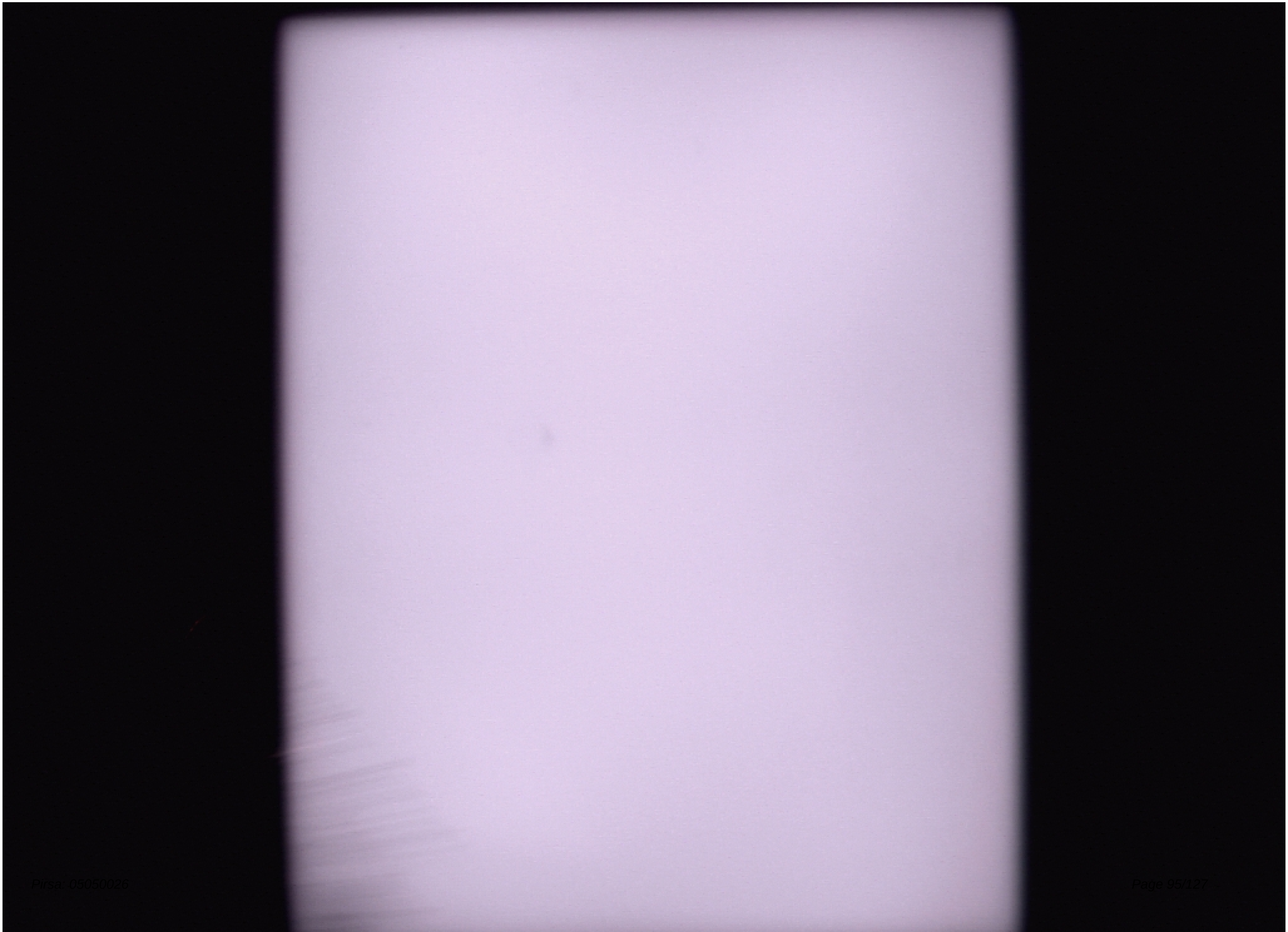
Why can forecast be difficult :

strong correlations between future & past

More precise (P.G.) :

F.C. $\equiv$ $\langle$ information about past, needed for forecast $\rangle$

$$\geq \langle \text{MI} \times \text{distance} \rangle \equiv \text{E.M.C.}$$



F.C. and "Logical depth"

C.H. Bennett :

logical depth of object S
relative to computer U

= CPU time needed to run
the shortest program for S

= difficulty to decode
shortest description of S

(algor. complexity = difficulty to
store it)

F.C. and "Logical depth"

C.H. Bennett :

logical depth of object S relative to computer U

= CPU time needed to run the shortest program for S

= difficulty to decode shortest description of S

(algor. complexity = difficulty to store it)

24a

Examples:

- life (real!) :

  no explicit program

  ~ 10^9 yrs run time on huge parallel computer "earth"

- cellular automaton #30

  (S. Wolfram)

L24a

Examples:

- life (real!) :

  no explicit program

  $\sim 10^9$ yrs run time on huge parallel computer "earth"

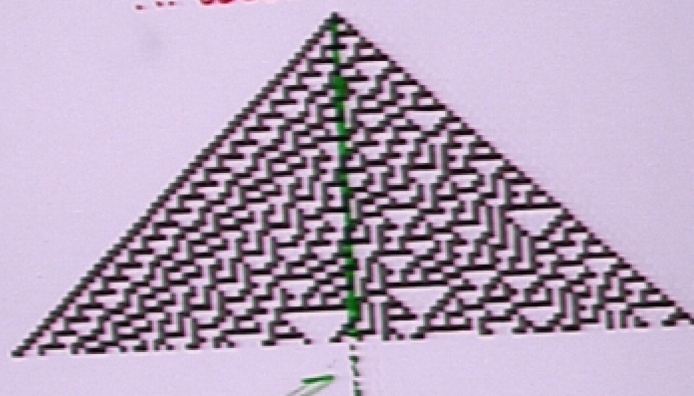
- cellular automaton #30 (S. Wolfram)

246

$\longrightarrow$ space

010 001 000
1 1 0

... 00000 1 00000 ...



"normal"
"~~random~~"
sequence:

time

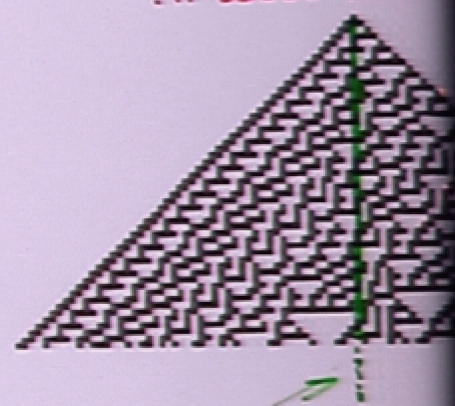
easiest way to produce:
copy bit by bit : N bits, ~N elem. steps
direct simulation :

rule # 30:

111	110	101	100	011	010	001	000
0	0	0	1	1	1	1	0

→ s

... 00000 100000



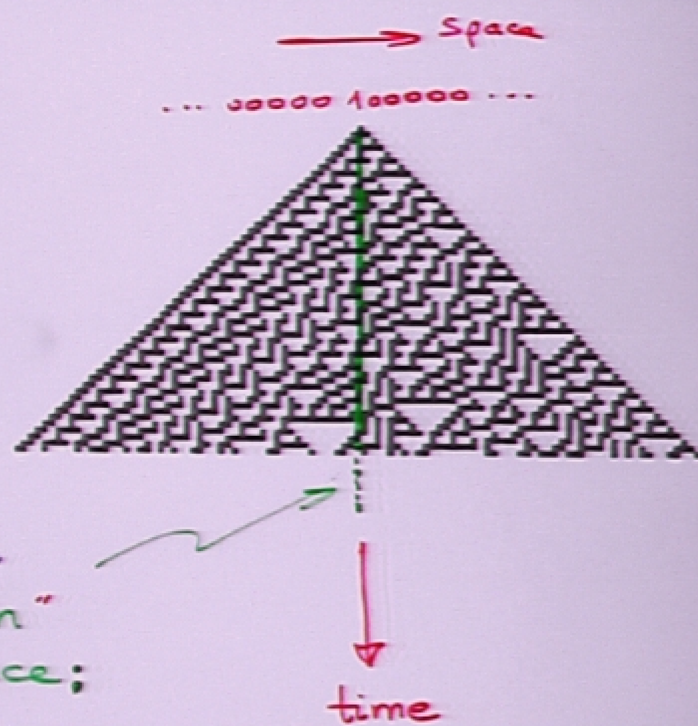
time

"normal"
"random"
sequence:

easiest way to produce:
copy bit by bit : N bits, ~N

shortest program: direct simulation :
$\sim N^2$ elementary steps

100 011 010 001 000
1 1 1 1 0



"normal"
"random"
sequence;

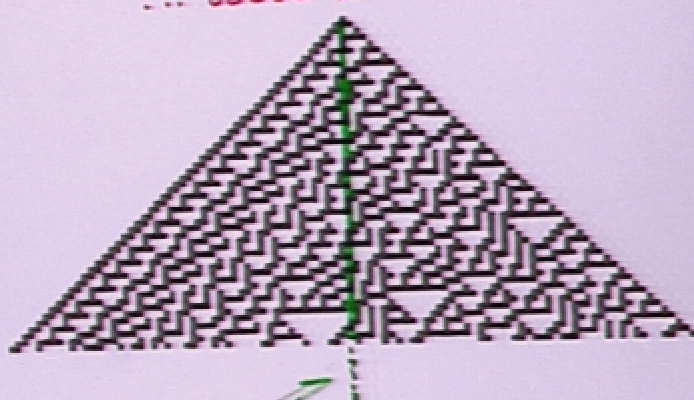
<u>easiest</u> way to produce:
copy bit by bit : N bits, ~N elem. steps

gram: direct simulation :
$\sim N^2$ elementary steps

→ space

011 010 001 000
1 1 1 0

... 00000 100000 ...



"normal"
"random"
sequence;

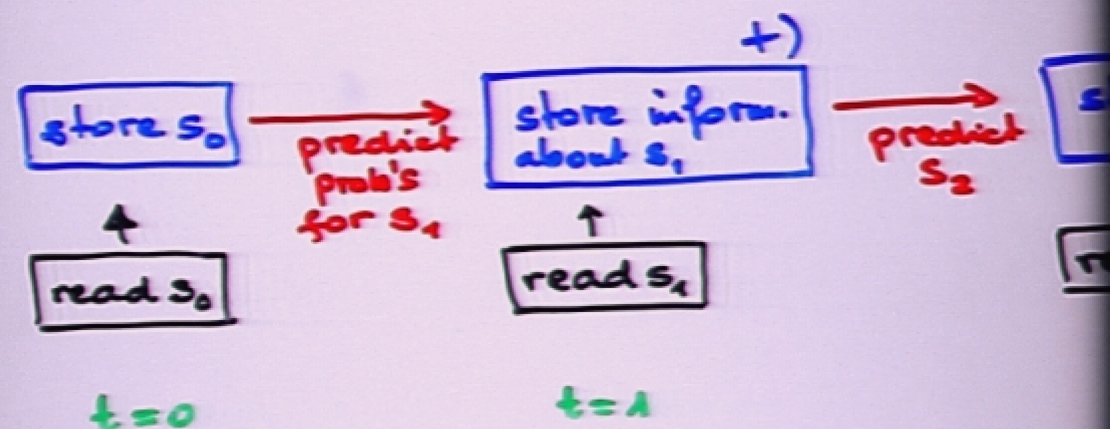
↓ time

easiest way to produce:
copy bit by bit : N bits, ~N elem. steps

: direct simulation :
~ N^2 elementary steps

General strategy to encode optimally a string :

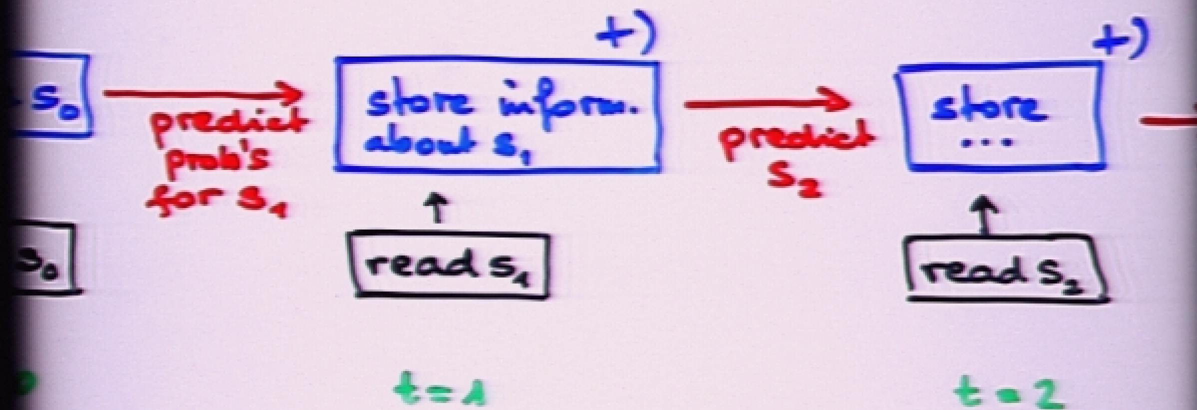
$S = s_0 s_1 s_2 \ldots$



+) store only that part of the informati[on]
that was still missing after the fo[...]
but before the observation

strategy to encode optimally a string

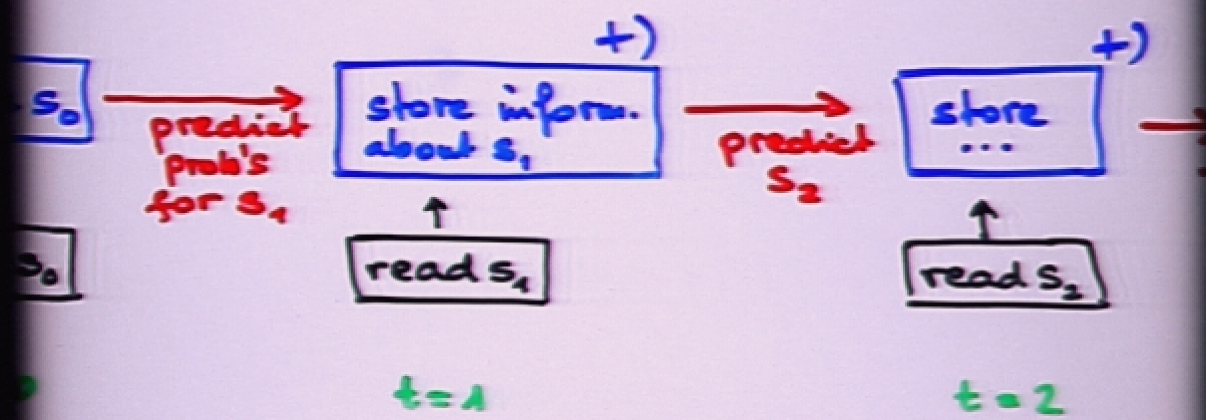
$s_1 s_2 \ldots\ldots$:



ore only that part of the information
hat was still missing after the forecast,
t before the observation

strategy to encode optimally a string

$s_1 s_2 \ldots\ldots$:



ore only that part of the information
at was still missing after the forecast,
t before the observation

optimal decoding:

- forecast
- supply stored info about s_n
- decode s_n

$\Rightarrow$ **forecasting complexity sets lower limit on logical depth**

NB: $\nexists$ algorithm for finding shortest program

→ only upper bds computable for $C_u(S)$

→ only lower bds computable on depth

- forecast
- supply stored info about s_n
- decode s_n

⇒ $\boxed{\text{forecasting complexity sets lower limit on logical depth}}$

NB: $\nexists$ algorithm for finding shortest program

→ only <u>upper</u> bds computable for $C_u(S)$

→ only <u>lower</u> bds computable on depth

task	complexity measure
perform algorithm (on single/multiple CPU)	time c. (time limited) space c. (space ") for single CPU / parallel CPU's
store & retrieve & ~~transmit~~ shortest code	Kolmogorov-Chaitin c. randomness
—"— with restriction on method	e.g. Lempel-Ziv
—"— with given prob. measure μ	Shannon entropy
decode shortest code	"logical depth" Bennett
	range & strength of correl. (mutual entropies)
[describe set of strings;] verify ("scan") that string conforms	complexity of grammar RLC
verify grammar, using know…	μ-complexi…

task	complexity measure
perform algorithm (on single/multiple CPU)	time c. (time limited) space c. (space ") for single CPU / parallel CPU's
store & retrieve & transmit shortest code	Kolmogorov-Chaitin c. randomness
—"— with restriction on method	e.g. Lempel-Ziv
—"— with given prob. measure μ	Shannon entropy
decode shortest code	"logical depth" Bennett range & strength of correl. (mutual entropies)
[describe set of strings;] verify ("scan") that string conforms	complexity of grammar RLC
verify grammar, using known measure μ	μ-complexity —"— μLC

(17

task	complexity measure
perform algorithm (on single/multiple CPU)	time c. (time limited) space c. (space ") for single CPU / parallel CPU's
store & retrieve & transmit shortest code	Kolmogorov-Chaitin c. randomness
—"— with restriction on method	e.g. Lempel-Ziv
—"— with given prob. measure μ	Shannon entropy
decode shortest code	"logical depth" Bennett range & strength of correl. (mutual entropies)
[describe set of strings;] verify ("scan") that string conforms	complexity of grammar RLC
verify grammar, using known measure μ	μ-complexity —"— μLC

task	complexity measure
perform algorithm (on single/multiple CPU)	time c. (time limited) space c. (space ") for single CPU / parallel CPU's
store & retrieve & transmit shortest code	Kolmogorov-Chaitin c. randomness
—"— with restriction on method	e.g. Lempel-Ziv
—"— with given prob. measure μ	Shannon entropy
decode shortest code	"logical depth" Bennett range & strength of correl. (mutual entropies)
[describe set of strings;] verify ("scan") that string conforms	complexity of grammar RLC
verify grammar, using known measure μ	μ-complexity —"— μLC

perform algorithm (on single/multiple CPU)	time c. (time limited) space c. (space ") for single CPU / parallel CPU's
store & retrieve & transmit shortest code	Kolmogorov-Chaitin c. randomness
—"— with restriction on method	e.g. Lempel-Ziv
—"— with given prob. measure μ	Shannon entropy
decode shortest code	"logical depth" Bennett range & strength of correl. (mutual entropies)
[describe set of strings;] verify ("scan") that string conforms	complexity of grammar RLC
verify grammar, using known measure μ	μ-complexity —"— μLC

	effective measure c.
find shortest code of pattern, "understand" it	"meaning"
define meaning & complexity uniquely	? ?

	effective measure c.
find shortest code of pattern, "understand" it	"meaning"
define meaning & complexity uniquely	? ?

• decode s_n

⇒ $\boxed{\text{forecasting complexity sets lower limit on logical depth}}$

NB : $\nexists$ algorithm for finding shortest program

→ only upper bds computable for $C_u(S)$

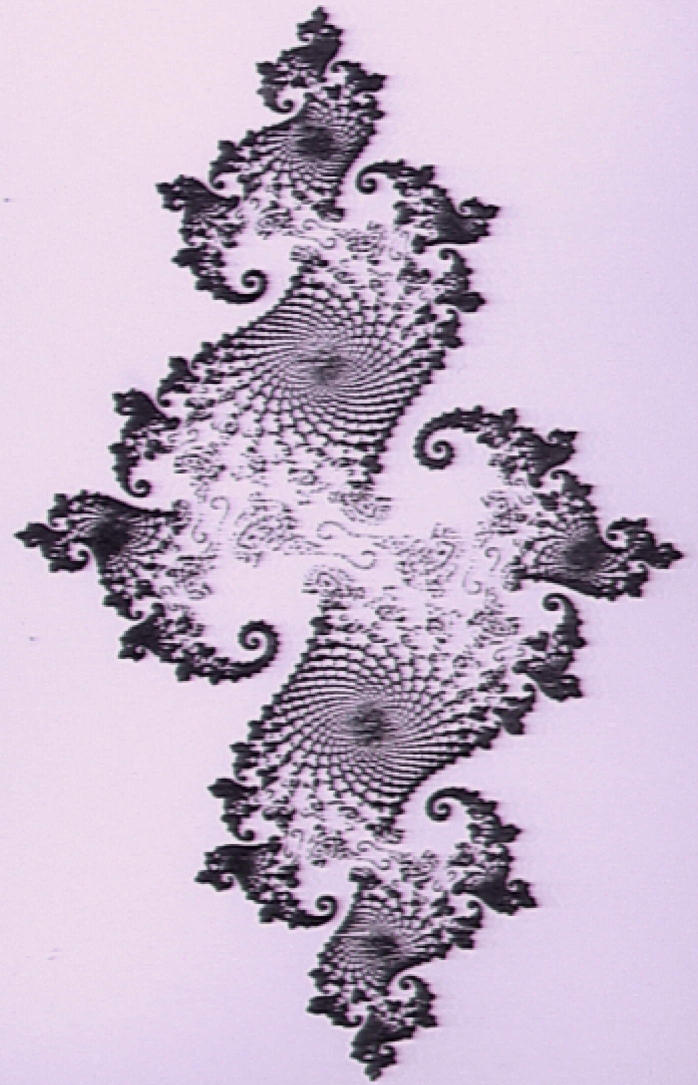
→ only lower bds computable on depth

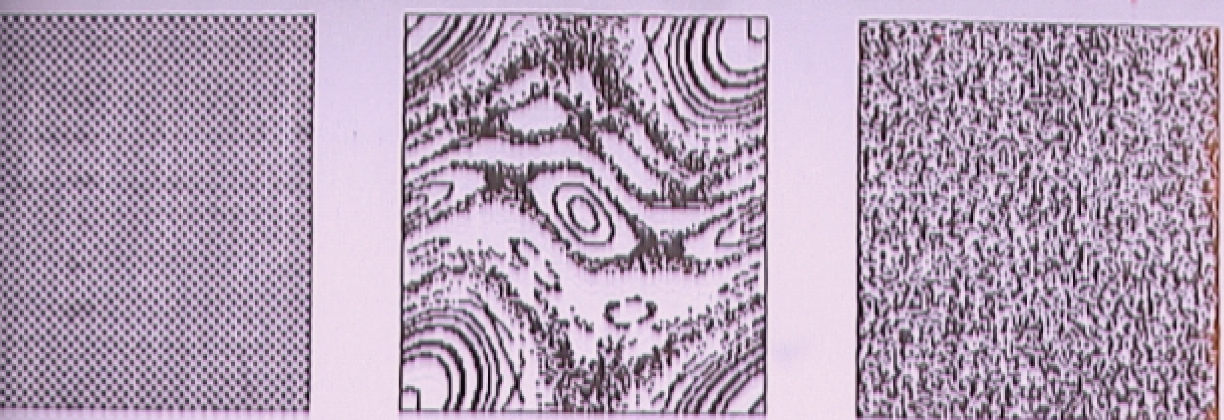
Fig. 4

a) b) c)

complex ?

maybe this is





a) b) c)

complex ?

maybe this is complex

≠ random numbers

$$x_{n+1} = 2 - x_n^2$$

$\Big\downarrow$ symbol dyn

$x_n < 0 \Rightarrow s_n = -1$

$x_n > 0 \rightarrow s_n = +1$

$$x_{n+1} = 2 - x_n^2$$

$$\Big\Downarrow \text{symbol dyn}$$

$x_n < 0 \longrightarrow s_n = -1$

$x_n > 0 \longrightarrow s_n = +1$

$x_{n+1} = 1.8732 - x_n^2$

$$x_{n+1} = 2 - x_n^2$$

$\Downarrow$ symbol dyn

$x_n < 0 \quad \to \quad s_n = -1$

$x_n > 0 \quad \to \quad s_n = +1$

$x_{n+1} = 1.8732 - x_n^2$

$$x_{n+1} = 2 - x_n^2$$

$\Downarrow$ symbol dyn

$x_n < 0 \;:\; \to s_n = -1$

$x_n > 0 \;\;\;\; \to s_n = +1$