

From Complexity to Intelligence

Introduction to Algorithmic Information Theory

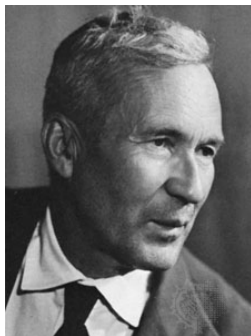
Nils Holzenberger

September 14, 2026

Outline

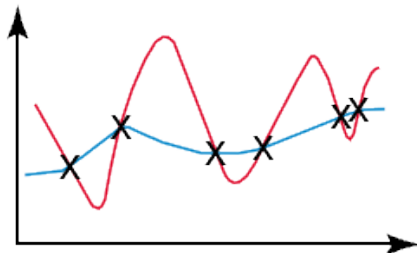
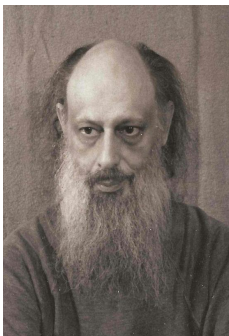
- 1 Algorithmic information for Artificial Intelligence
 - Chaitin - Solomonoff - Kolmogorov
 - What is it good for?
- 2 Kolmogorov complexity
 - Turing machine
- 3 Complexity of integers
- 4 Conditional complexity

Andrei Kolmogorov



11001000011000011101111011101100111110100100001001010111

Ray Solomonoff

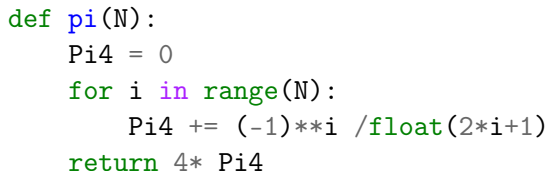


Chaitin



- computability
- Chaitin's Ω
- "Comprehension is compression"

- Algorithmic information is pure information
- Algorithmic information is the result of maximal compression



AIT for mathematics

- (re-)define probabilities
- define randomness
- prove Gödel's theorem (proof fits on 1 slide)

AIT for machine learning

You're trying to explain your favourite ML system's decisions.

You come up with three (locally) equally performing rules.

Which decision rule is preferable as an “explanation”?

- $\text{Modified_Kröner_Levi_test} < 7 \rightarrow \text{at risk.}$
- $3.873 > \text{IG} > 3.871 \rightarrow \text{at risk}$
- $\text{Age} > 40 \text{ and Sugar} > 17 \text{ and BMI} > 35 \rightarrow \text{at risk.}$

AIT for cognitive science

What is the most probable continuation of 1223334444?

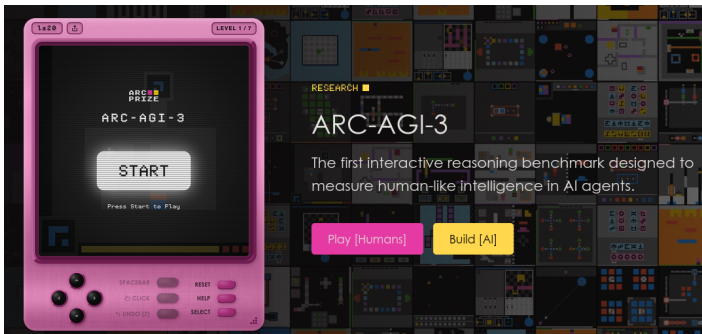
AIT for cognitive science

In a lottery game like the French National lottery, six numbers are drawn among 49 and the resulting combination is presented in increasing order.

What would be the most “incredible” outcome?

- 1, 2, 3, 4, 5, 6 because it is the simplest series of 6 numbers.
- 7, 14, 21, 28, 35, 42, 49 because it is evenly distributed between 1 and 49.
- Last week's outcome (as occurred on 10 September 2009 in Bulgaria)
- None, because all outcomes have the same probability.

AIT for Artificial Intelligence



AIT for Artificial Intelligence

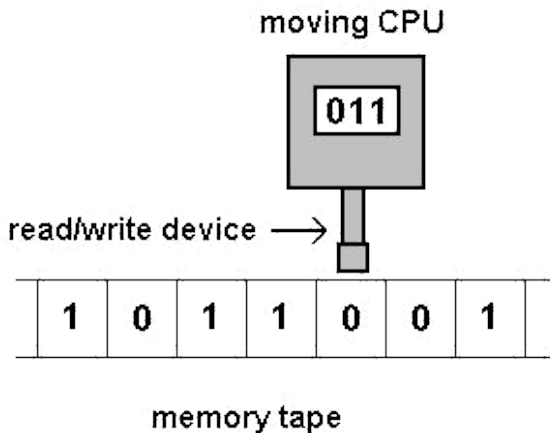
specific tasks, such as board games and video games. We argue that solely measuring skill at any given task falls short of measuring intelligence, because skill is heavily modulated by prior knowledge and experience: unlimited priors or unlimited training data allow experimenters to “buy” arbitrary levels of skills for a system, in a way that masks the system’s own generalization power. We then articulate a new formal definition of intelligence based on Algorithmic Information Theory, describing intelligence as *skill-acquisition efficiency* and highlighting the concepts of *scope*, *generalization difficulty*, *priors*, and *experience*, as critical pieces to be accounted for in characterizing intelligent systems. Using this definition, we propose a set of guidelines for what a general AI benchmark should look like.

Turing Machine

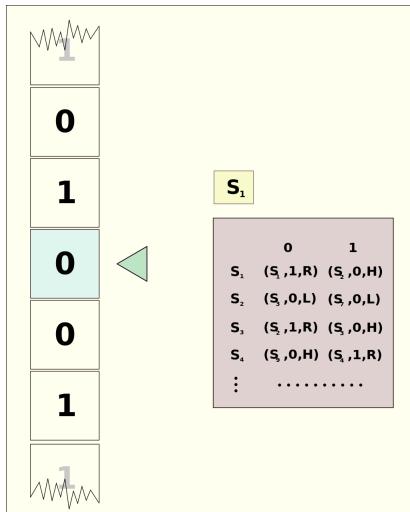
What is a Turing Machine?

- the machine used by Alan Turing to decipher the cryptographic code used by the German navy during WWII
- an abstract computer without memory limitation used to define computability
- a stack automaton used to parse context-free grammars

Turing Machine



Turing Machine



Simple machine

Program	Output
0	A
10	B
110	C
111	D

Another simple machine

Program	Output
0	0
1	1
10	2
11	3
100	4
...	...

Kolmogorov complexity

$$C(x) = \min_p \{\ell(p) \mid M(p) = x\}$$

- x : some object. An integer, a string of characters, a person...
- $C(x)$: the complexity of object x
- M : a Turing machine
- p : a program that can be fed to M (so p is a string of 0's and 1's)
- $\ell(p)$: the length of program p

The complexity of x is the length of the shortest program that can generate x

Machine-dependency

The value of $C(x)$ depends on the machine M . So for given x , if I pick the right machine, $C(x) = 0$. How is this definition useful?

- Invariance theorem:

$$\forall x \quad |C_U(x) - C_V(x)| \leq A_{U,V}$$

where U and V are universal Turing machines.

- In theoretical math and CS results, we care about asymptotics.
- In practical application, we generally care about relative values or complexity differences.

Invariance theorem

- Let p_U (resp. p_V) be the minimal program that can generate object x using machine U . $C_U(x) = \ell(p_U)$, $C_V(x) = \ell(p_V)$.
- U is a universal Turing machine so let $p_{U \rightarrow V}$ be the program to simulate V using U . $V = U(p_{U \rightarrow V})$.
- $V(p_V) = x$ and $U(p_{U \rightarrow V} + p_V) = x$
- By definition of complexity, $C_U(x) \leq \ell(p_{U \rightarrow V} \circ p_V)$
- So $C_U(x) \leq \ell(p_{U \rightarrow V}) + \ell(p_V) = \ell(p_{U \rightarrow V}) + C_V(x)$
- *Mutandem mutandis*, $C_V(x) \leq \ell(p_{V \rightarrow U}) + C_U(x)$
- Combine the two: $|C_U(x) - C_V(x)| \leq \max\{\ell(p_{U \rightarrow V}), \ell(p_{V \rightarrow U})\}$

Binary coding of digits

Number	Binary encoding	Complexity
0	0	1
1	1	1
2	10	2
3	11	2
4	100	3
...		

$$C(n) = 1 + \lceil \log_2(n+1) \rceil \quad (\text{for } n > 0)$$

Can we do better?

Efficient encoding

Simple improvement: encode $n+2$, drop initial bit (always equal to 1)

Number	Binary encoding	Efficient coding	Are the two columns really comparable? No, they correspond to different Turing machines. But, if we're allowed to reprogram the machine, the second encoding is indeed better. Can we do better?
0	0	0	
1	1	1	
2	10	00	
3	11	01	
4	100	10	
5	101	11	
6	110	000	
7	111	001	
8	1000	010	
9	1001	011	
10	1010	100	

Positional encoding

“1000” is simpler to encode than “1343”

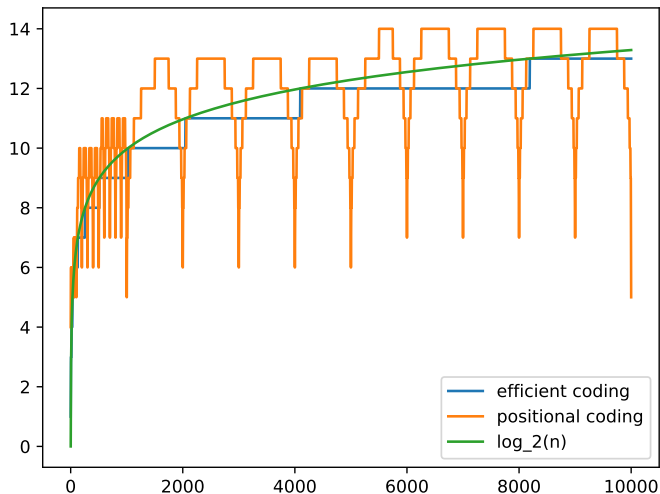
Consider 89995 ($= 9 \cdot 10^4 - 5$)

Efficient coding: 0101111110001101

Positional encoding: 1 011 10 11

- Initial bit: 0 for +, 1 for –
- Second block: digit before exponent (9 which is 011 in efficient coding)
- Third block: exponent (4)
- Fourth block: number to add or subtract (5)

Comparison



Mixed coding

Consider 89995 ($= 9 \cdot 10^4 - 5$)

Efficient coding: 0101111110001101 (16 bits)

Positional encoding: 1 011 10 11 (8 bits)

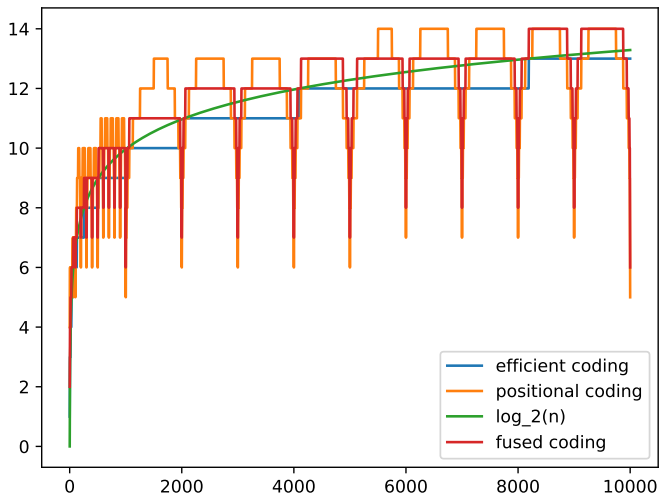
Consider 1343 ($= 1 \cdot 10^3 + 343$)

Efficient coding: 0101000001 (10 bits)

Positional encoding: 0 1 01 01011001 (12 bits)

Mixed coding: first bit says whether it's the efficient encoding (0) or the positional encoding (1), the remaining bits are the encoding

Mixed coding

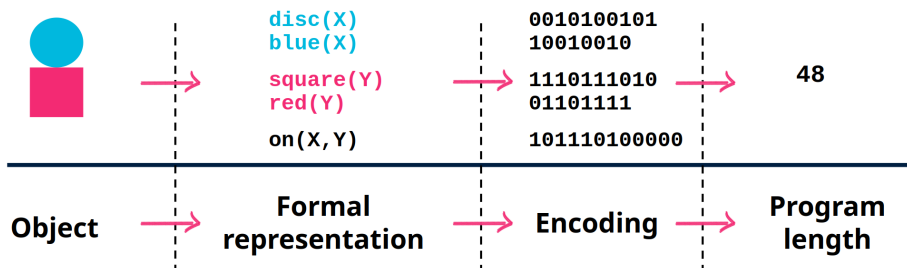


Encoding integers

Can we do even better?

What about encoding a finite list of integers?

Complexity of any object



Intuition

What is the complexity of 101000001101100010101100011010?

- ... knowing that it is the binary code for 06 74 63 86 18?
- ... knowing that it is the phone number of the prime minister?

side information can make compression easier.

Definition

Conditional complexity

$$C_U(x|y) = \{\ell(p) \mid U(p + y) = x\}$$

The conditional complexity of x given y is the shortest program that generates x , knowing y

$$C(x|x) = ?$$

Chain rule

$$C(x, y) \leq C(y) + C(x|y) + O(1)$$

Chain rule (2)

$$C(x, y) = C(y) + C(x|y) + O(\log C(x, y))$$