

3EL05

Chapitre 1 Complexité de description

On the Measure of Intelligence

François Chollet *

Google, Inc.

fchollet@google.com

November 5, 2019



ARC PRIZE REMAINS UNDEFEATED.
NEW IDEAS STILL NEEDED.

- > Home
- > ARC-AGI
- > 2024 Results
- > Technical Guide
- > Play
- > Blog

2025 SIGN UP

OpenAI o3 unlocks breakthrough high score on ARC-AGI-Pub. [Learn more.](#)

ARC PRIZE

ARC Prize is a \$1,000,000+ public competition to beat and open source a solution to the ARC-AGI benchmark.

Hosted by [Mike Knoop](#) (Co-founder, Zapier) and [François Chollet](#) (Creator of ARC-AGI, Keras).

ARC Prize 2025 coming soon.

> [See 2024 winners](#)

ARC-AGI

Most AI benchmarks measure skill. But skill is not intelligence. General intelligence is the ability to efficiently acquire new skills.

Chollet's unbeaten 2019 Abstraction and Reasoning Corpus for Artificial General Intelligence ([ARC-AGI](#)) is the only formal benchmark of AGI progress.

It's easy for humans, but hard for AI.

The intelligence of a system is a measure of its skill-acquisition efficiency over a scope of tasks, with respect to priors, experience, and generalization difficulty.

Intuitively, if you consider two systems that start from a similar set of knowledge priors, and that go through a similar amount of experience (e.g. practice time) with respect to a set of tasks not known in advance, the system with higher intelligence is the one that ends up with greater skills (i.e. the one that has turned its priors and experience into skill more efficiently). This definition of intelligence encompasses meta-learning priors, memory, and fluid intelligence. It is distinct from skill itself: skill is merely the output of the process of intelligence.

Dans ce cours:

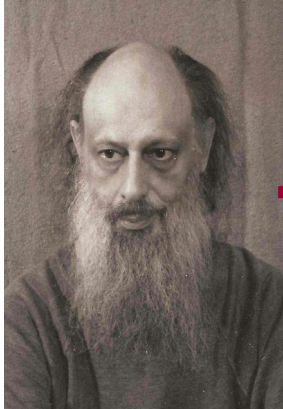
- sciences cognitives
- apprentissage automatique
- langage naturel
- mathématiques et informatique théorique

Sujets directement reliés à l'intelligence artificielle

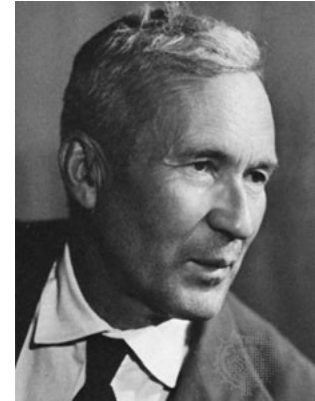
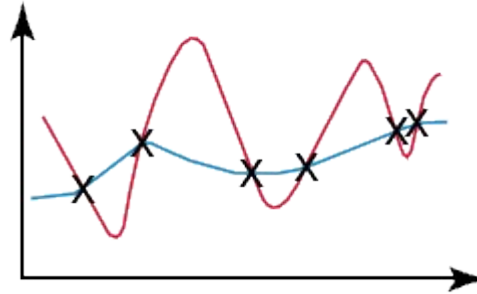
Algorithmic Information Theory

- Information is compression
 - Induction is compression
 - Coding is compression
 - Storing is compression
 - Forgetting is compression
 - Visualizing is compression
 - Sketching is compression
 - Probability is compression
 - Learning is compression
 - Proving is compression
 - Relevance is compression
 - **“Comprehension is compression”**
 - Pattern recognition is compression
- Image synthesis is compression
 - Abduction is compression
 - Analogy is compression
 - Responsibility is compression
 - Intelligence is compression
 - Aesthetics is compression
 - Humour is compression
 - Creativity is compression
 - Emergence is compression
 - ...
- ← Gregory Chaitin

Algorithmic Information Theory



Ray Solomonoff



Andrei Kolmogorov



Gregory Chaitin

11001000011000011101111011101100111110100100001001010111

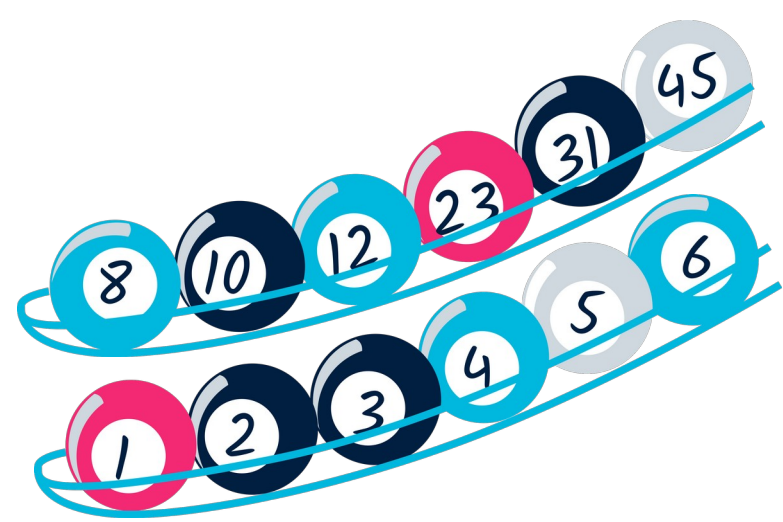
(premier) théorème d'incomplétude
de Gödel, constante Omega

Information algorithmique

Peut servir à :

- Définir la notion d'aléatoire (ch. 3)
- Prouver le théorème de Gödel en quelques lignes (ch. 3)
- Améliorer les algorithmes d'apprentissage automatique (ch. 4)
- Expliquer des phénomènes cognitifs (ch. 5)

What is the most probable continuation of
1223334444



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

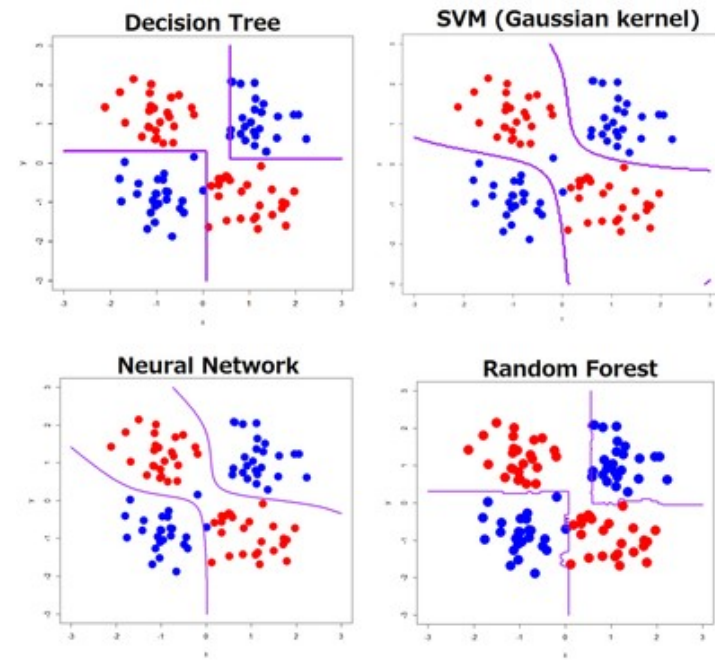
What would be the most "incredible" outcome?

- 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.

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"?

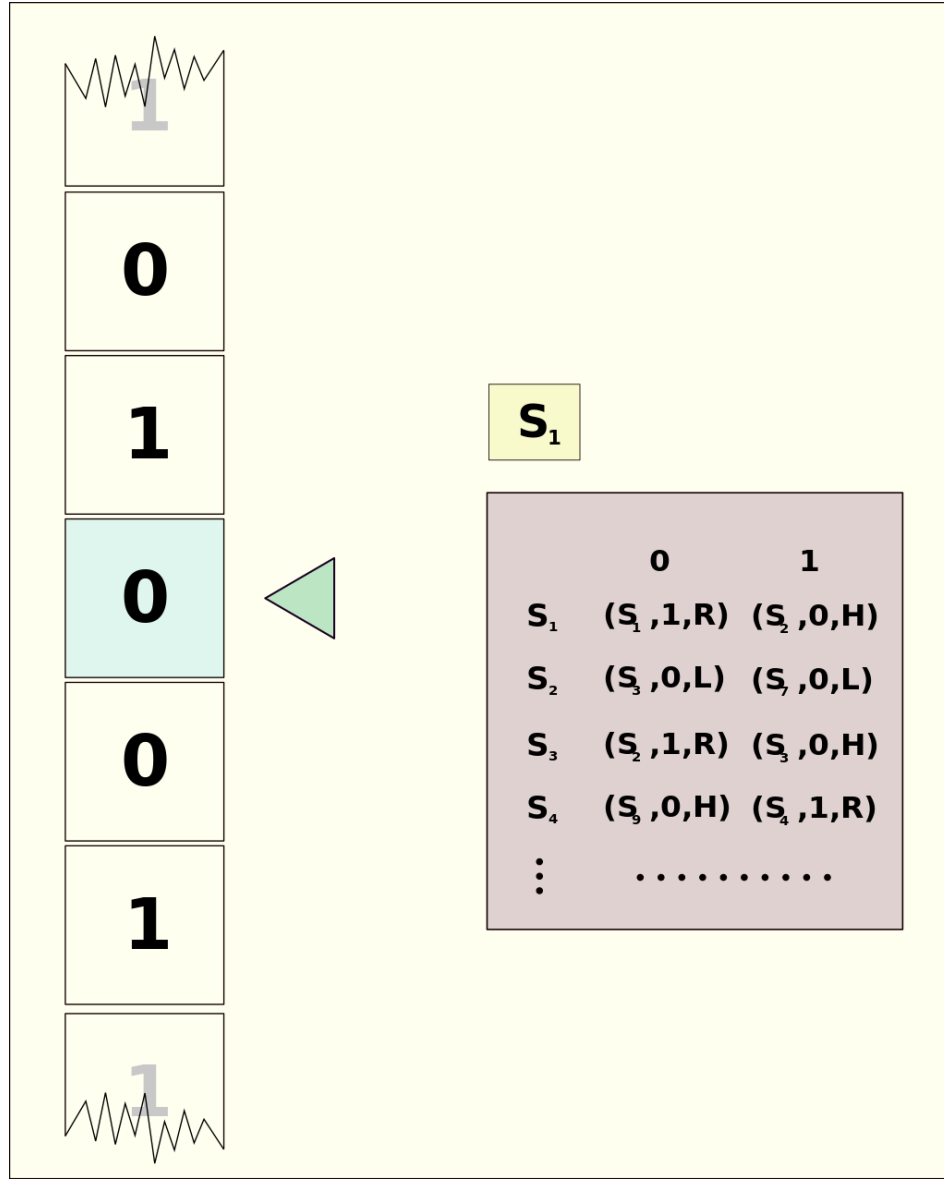
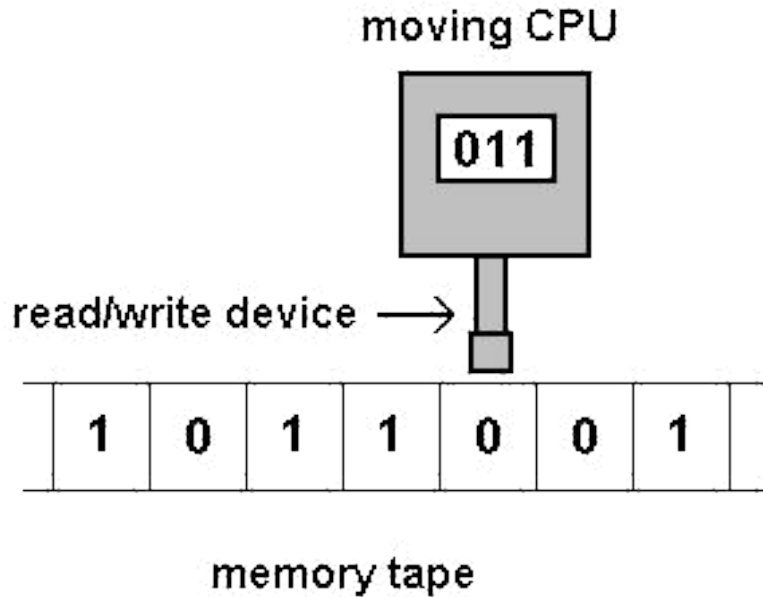


- ❑ $\text{Modified_Kr\"oner_Levi_test} < 7 \rightarrow \text{at risk.}$
- ❑ $3.873 > \text{IG} > 3.871 \rightarrow \text{at risk}$
- ❑ $\text{Age} > 40 \text{ and } \text{Sugar} > 17 \text{ and } \text{BMI} > 35 \rightarrow \text{at risk.}$

Machine de Turing

- 1) La machine utilisée par Alan Turing pour déchiffrer le code cryptographique utilisé par la marine allemande durant la deuxième guerre mondiale
- 2) Un ordinateur abstrait sans limite de mémoire
- 3) Un automate à pile utilisé pour parser les grammaires non contextuelles

Machine de Turing



Machine de Turing

Programme	Sortie
0	A
10	B
110	C
111	D

Machine de Turing

Programme	Sortie
0	0
1	1
10	2
11	3
100	4
...	...

Complexité de Kolmogorov

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

Un programme que l'on peut
donner à une machine de Turing
(→ une chaîne de 0 et 1)

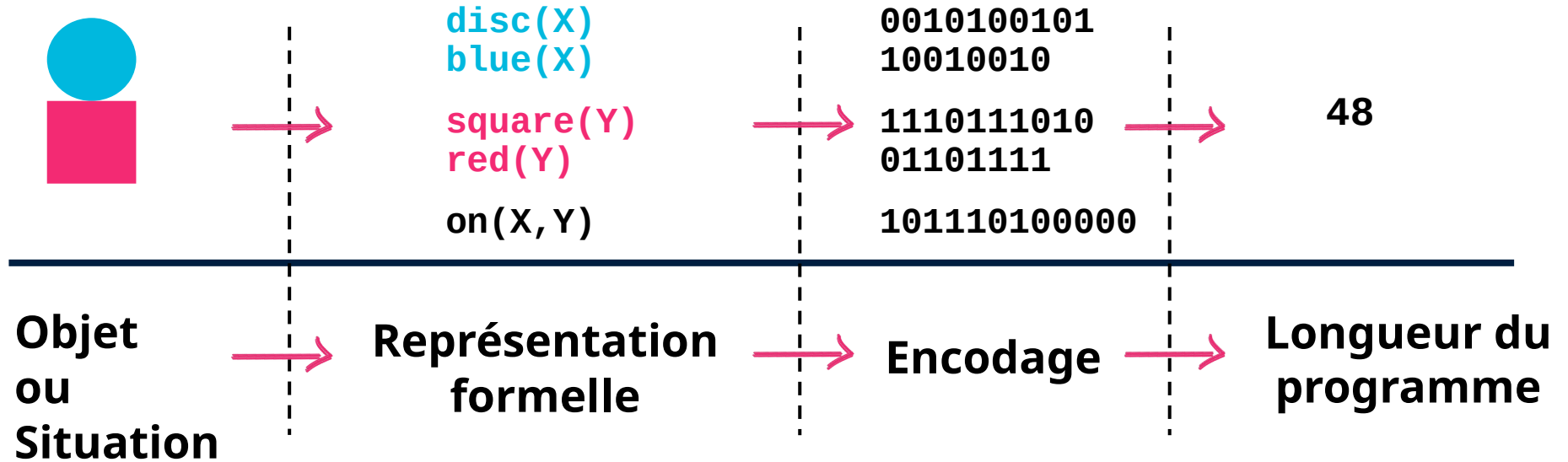
La longueur du programme

Une machine de
Turing, déjà
programmée

Un objet dont on veut mesurer la
complexité : un nombre entier, une
chaîne de caractères, une personne
physique...

*La longueur du plus court
programme permettant de
générer x (avec une machine
de Turing donnée)*

Complexité de Kolmogorov



Complexité de Kolmogorov

- La valeur de la complexité dépend de la machine de Turing choisie
- Je peux programmer une machine de Turing qui écrit les 1000 premières décimales de π quel que soit le programme en entrée $\rightarrow C(\pi) = 0$
- Gardez en tête quelle est la machine

$$C_M(x) = \min_p \{l(p) \mid M(p) = x\}$$

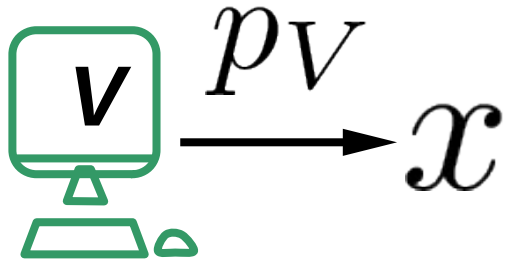
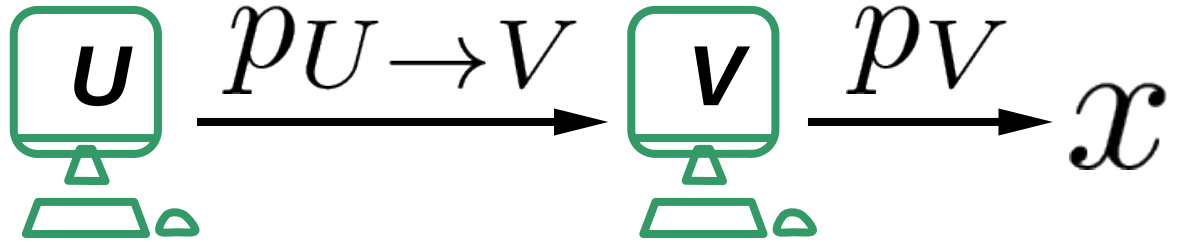
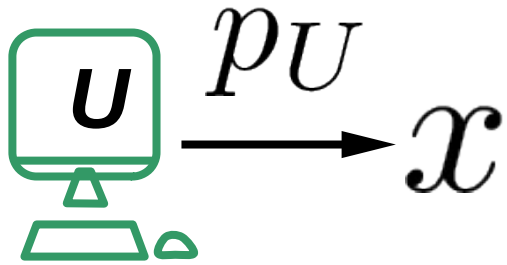
Complexité de Kolmogorov

- La définition dépend de la machine considérée, alors quel intérêt ?
- U et V sont des machines de Turing universelles

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

Théorème d'invariance

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



$$\begin{aligned} C_U(x) &= l(p_U) \leq l(p_{U \rightarrow V}) + l(p_V) \\ &= l(p_{U \rightarrow V}) + C_V(x) \end{aligned}$$

$$C_U(x) - C_V(x) \leq l(p_{U \rightarrow V})$$

Complexité des entiers naturels

- But : encoder les entiers naturels de façon compacte / avec un programme court
- Liste finie d'entiers \rightarrow liste ordonnée
- Encoder n'importe quel entier

Complexité des entiers naturels

Nombre	Représentation binaire	Complexité
1	1	1
2		
3		
4		
5		
6		
...	... $C(n) \approx \log_2(n)$...
10		
11		

Complexité des entiers naturels

N	<i>Code</i>	$\lceil \log_2(n+3) \rceil - 1$
0	0	1
1	1	1
2	00	2
3	01	2
4	10	2
5	11	2

n	<i>Code</i>	$\lceil \log_2(n+3) \rceil - 1$
6	000	3
7	001	3
8	010	3
9	011	3
10	100	3
11	101	3

- Amélioration simple : encoder $n+2$, et laisser tomber le premier bit (toujours égal à 1)
- Pour comparer avec l'encodage précédent, il faudrait prendre en compte la complexité de l'opération de transformation
- On peut faire disparaître ce court programme en modifiant la machine de Turing₄

Complexité des entiers naturels

- Encore mieux : prendre en compte les suites de chiffres
- “1000” est plus facile à encoder que “1343”

Complexité des entiers naturels

89995

Codage binaire amélioré

0101111110001101

```
def cba(n): return str(bin(n+2))[3:]
```

Codage binaire positionnel

1

011

10

11



$89995 = 9E4 - 5$

-

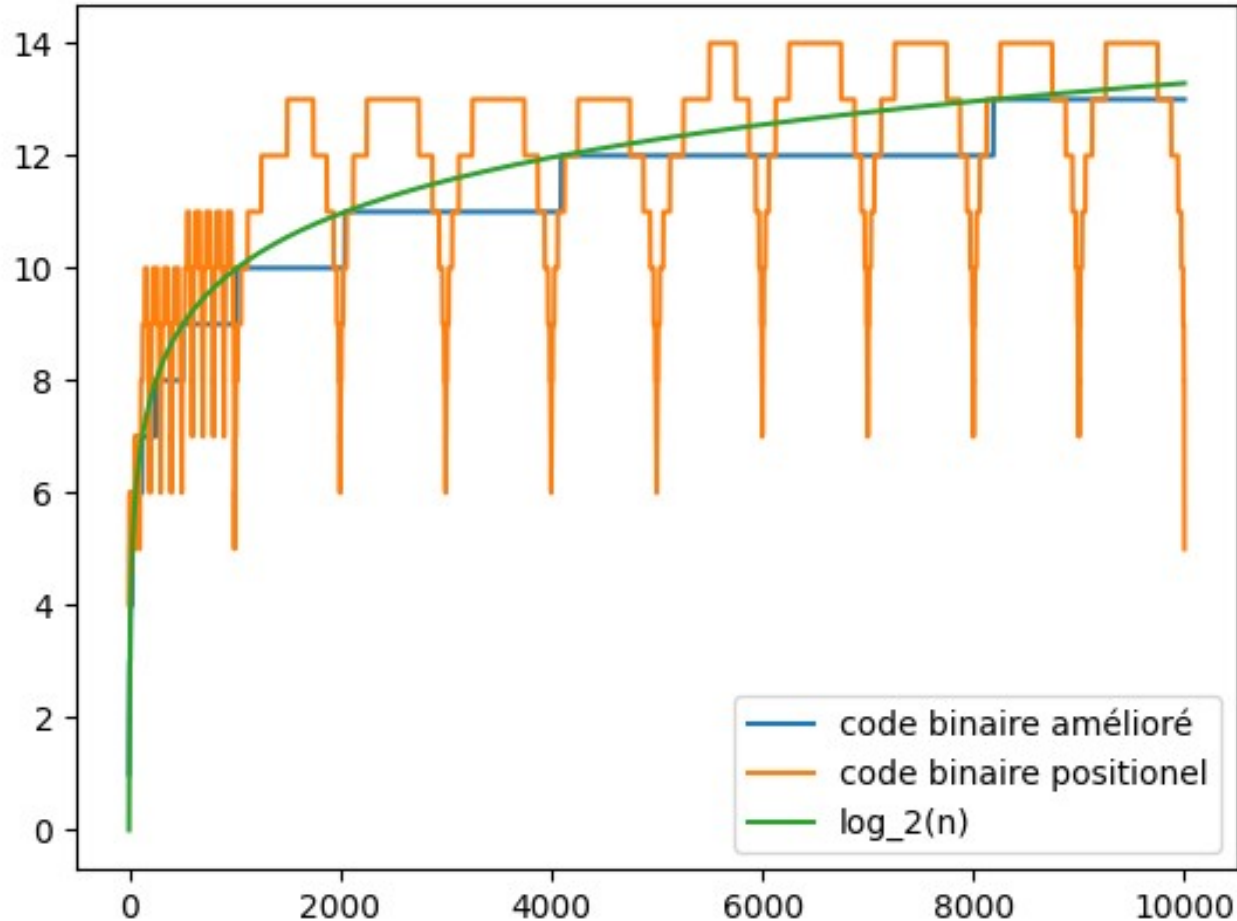
9

4

5

(soustraction)

Complexité des entiers naturels



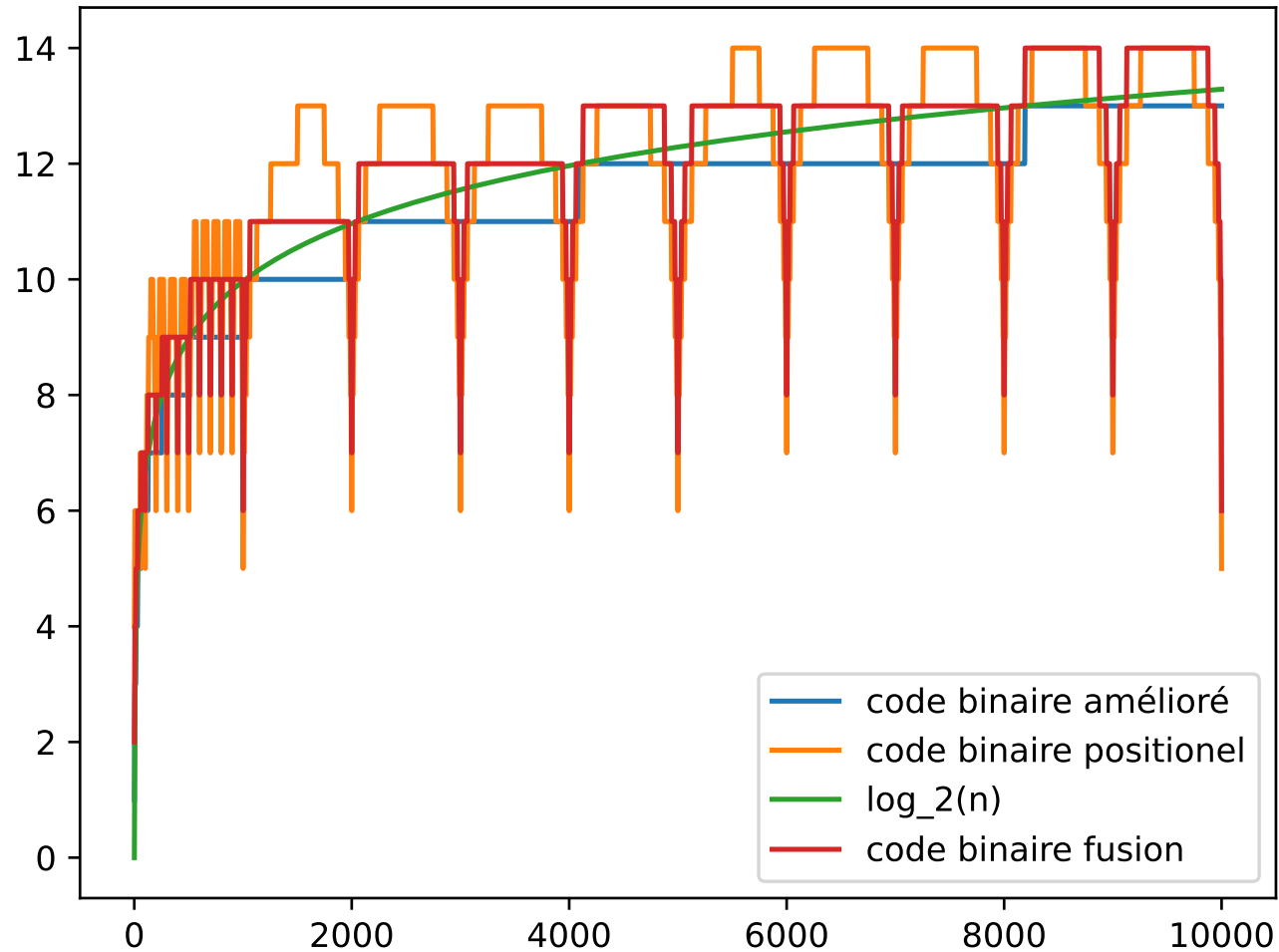
- Chute de complexité pour les nombres ronds
- Encodage de qualité variable
- Combinaison des 2:
 - Premier bit indique encodage choisi
 - On prend le plus court

Complexité des entiers naturels

89995

Codage binaire amélioré	0101111110001101				
Codage binaire positionel	1	011	10	11	
Codage binaire fusion	1	1	011	10	11

Complexité des entiers naturels



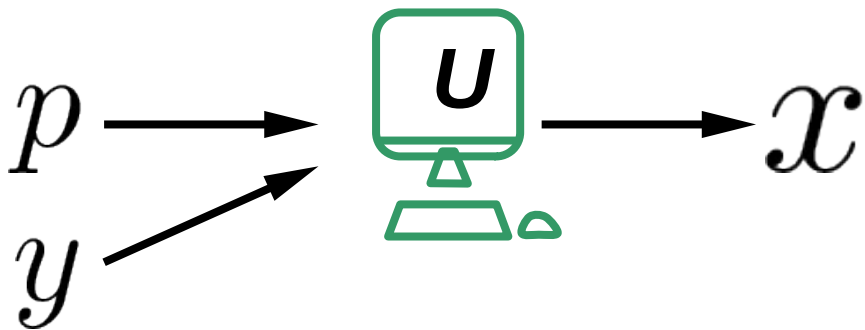
Complexité conditionnelle

Complexité de 101000001101100010101100011010?

- C'est 06 74 63 86 18 en binaire
- C'est le numéro de téléphone de Bertrand David

Complexité conditionnelle

$$C_U(x|y) = \min_p \{l(p) \mid U(p, y) = x\}$$



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

Complexité conditionnelle

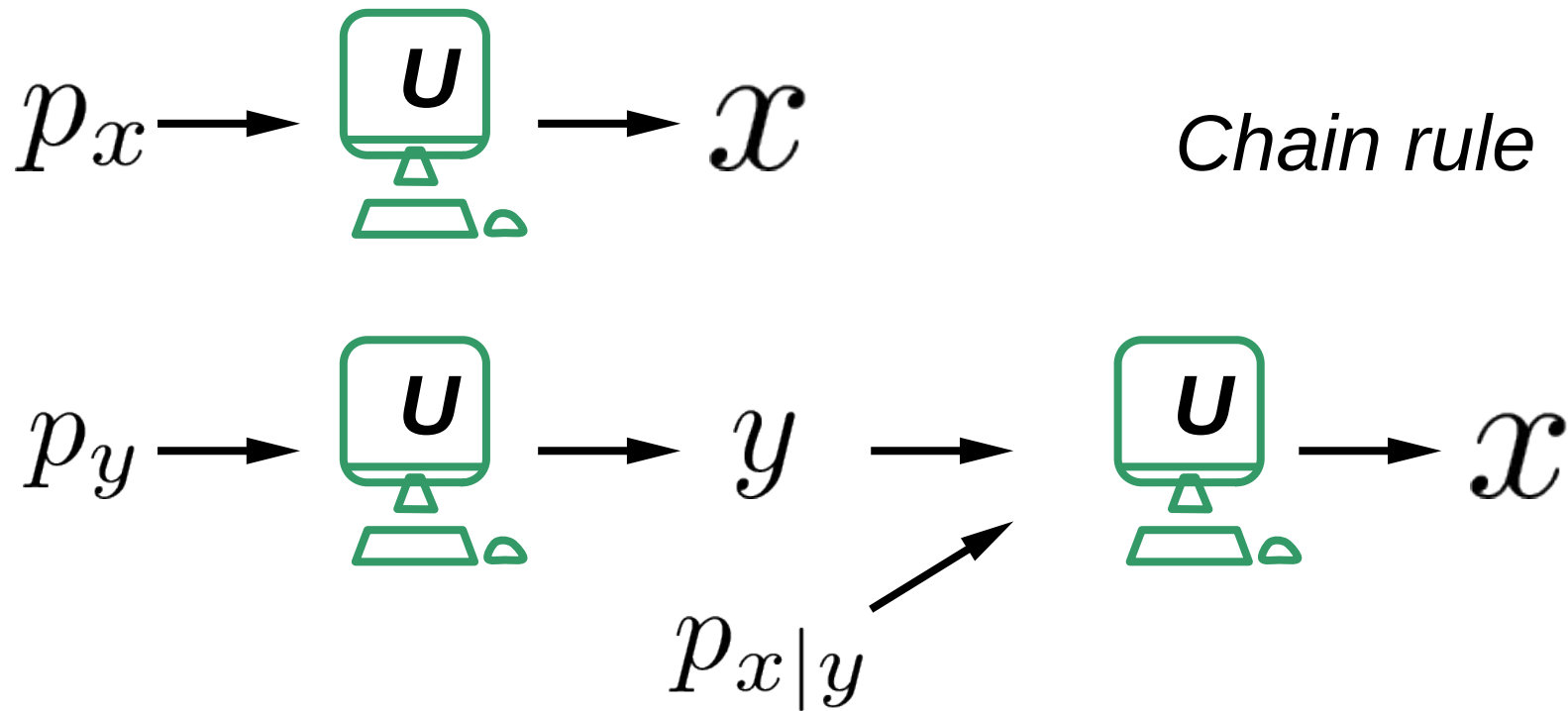
$$C(1415926535897932384626433) = 80$$

$$C(1415926535897932384626433|\pi) = 5$$

Certains nombres sont peu complexes pour vous :

- Anniversaires
- Numéros de téléphone
- ...

Complexité conditionnelle



$$C(x) \leq C(y) + C(x|y) + B$$

← Séparateur entre
2 programmes

Complexité conditionnelle

$$C(x) \leq C(y) + C(x|y) + O(1) \quad \textit{Chain rule}$$

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