Kolmogorov Complexity of Categories

Noson S. Yanofsky

Brooklyn College, CUNY

May 28, 2013

Outline of Talk

Classical Motivation

Consider the following three strings:

1. 00000000000000000000000000000000000000000000000
2. 11011101111101111111011111111111011111111111110
3. 01010010110110101011011101111001100000111111010

Classical Motivation

Consider the following three strings:

1. 00000000000000000000000000000000000000000000000
2. 11011101111101111111011111111111011111111111110
3. 01010010110110101011011101111001100000111111010

1. **Print 45 0's**
2. **Print the first 6 primes**
3. **Print** 010100101101101010110111011110011000001111110

Classical Motivation

Kolmogorov Complexity of Categories

Noson S. Yanofsky

Consider the following three strings:

1. 00000000000000000000000000000000000000000000000
2. 11011101111101111111011111111111011111111111110
3. 0101001011011010101101110111100110000011111010

1. **Print 45 0's**
2. **Print the first 6 primes**
3. **Print** 010100101101101010110111011110011000001111110

Let U be a universal Turing machine, then

$$K(s) = min\{|p| : U(p, \lambda) = s\}.$$

Relative Kolmogorov complexity:

$$K(s|t) = min\{|p| : U(p, t) = s\}.$$

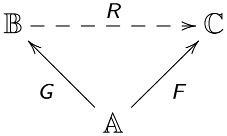
If $K(s) > |s|$ then s is "incompressible" or "random."

The Sammy Programming Language

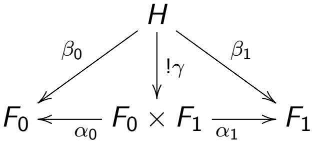
- Constant Categories: $\mathbf{0} = \emptyset$; $\mathbf{1} = \star$; $\mathbf{2} = \star \longrightarrow \star$; **Cat**.
- Constant Functors: $s : \mathbf{1} \longrightarrow \mathbf{2}$; $t : \mathbf{1} \longrightarrow \mathbf{2}$.
- If $\mathbb{C} = \mathbf{Source}(F : \mathbb{A} \longrightarrow \mathbb{B})$, then $\mathbb{C} = \mathbb{A}$.
- If $\mathbb{C} = \mathbf{Target}(F : \mathbb{A} \longrightarrow \mathbb{B})$, then $\mathbb{C} = \mathbb{B}$.
- If $F = \mathbf{Ident}(\mathbb{A})$ then $F = Id_{\mathbb{A}}$.
- If $\mathbb{C} = \mathbf{Op}(\mathbb{A})$ then $\mathbb{C} = \mathbb{A}^{op}$. The **Op** operation also acts on functors.
- $\alpha = \mathbf{Hcomp}(\beta, \gamma)$.
- $\alpha = \mathbf{Vcomp}(\beta, \gamma)$.
- Regular composition of functors is a special case of horizontal composition.
- For categories $\mathbb{A}$ and $\mathbb{B}$, we have $\mathbb{C} = \mathbf{Pow}(\mathbb{A}, \mathbb{B})$ be the category of all functors and natural transformations from $\mathbb{A}$ to $\mathbb{B}$.

The Sammy Programming Language

- For functors $G : \mathbb{A} \longrightarrow \mathbb{B}$ and $F : \mathbb{A} \longrightarrow \mathbb{C}$, a right Kan extension is a pair $(R, \alpha) = \textbf{KanEx}(G, F)$ where $R : \mathbb{B} \longrightarrow \mathbb{C}$ and $\alpha : R \circ G \longrightarrow F$.



- For every $H : \mathbb{B} \longrightarrow \mathbb{C}$ and $\beta : H \circ G \longrightarrow F$ there is a *unique* $\gamma = \textbf{KanInd}(F, G; H, \beta)$ where $\gamma : H \longrightarrow R$ and satisfies $\alpha \cdot \gamma_G = \beta$.



.

The Sammy Programming Language

- Left Kan extensions are made with the **Op** operation.
- Using Kan extensions, one can derive products, coproducts, pushouts, pullbacks, equalizers, coequalizers, (and constructible) limits, colimits, ends, coends, etc.
- If $G : \mathbb{A} \longrightarrow \mathbb{B}$ is a right adjoint (left adjoint, equivalence, isomorphism), then its left adjoint (right adjoint, quasi-inverse, inverse) $G^* : \mathbb{B} \longrightarrow \mathbb{A}$ can be found as a simple Kan extension of the identity $Id_{\mathbb{A}}$ along G, that is, $G^* = \mathbf{KanEx}(G, Id_{\mathbb{A}})$.
- There are also Kan liftings operations.
- Other operations...

Remarks About Sammy

- Not the first programing language for Categories
  - Rydeheard and Burstall: *Computational Category Theory*
  - Tatsuya Hagino: *A Categorical Programming Language*

Remarks About Sammy

- Not the first programing language for Categories
  - Rydeheard and Burstall: *Computational Category Theory*
  - Tatsuya Hagino: *A Categorical Programming Language*
- Not the best programing language for Categories
  - e.g. **Target** from **Source** and **Op**

Remarks About Sammy

- Not the first programing language for Categories
  - Rydeheard and Burstall: *Computational Category Theory*
  - Tatsuya Hagino: *A Categorical Programming Language*
- Not the best programing language for Categories
  - e.g. **Target** from **Source** and **Op**
- Notice that numbers, strings, trees, graphs, arrays, and other typical data types are not mentioned in Sammy. They can be derived. Categories and algorithms are more "primitive" than numbers, strings, trees, etc.

Remarks About Sammy

- Not the first programing language for Categories
  - Rydeheard and Burstall: *Computational Category Theory*
  - Tatsuya Hagino: *A Categorical Programming Language*
- Not the best programing language for Categories
  - e.g. **Target** from **Source** and **Op**
- Notice that numbers, strings, trees, graphs, arrays, and other typical data types are not mentioned in Sammy. They can be derived. Categories and algorithms are more "primitive" than numbers, strings, trees, etc.
- In need of a Church-Turing type thesis that says that anything that can be described by category theory can be described by Sammy.
- No discussion of "self-delimiting."
- Easily encode and decode Sammy programs as a number... or as a functor $P : \mathbf{1} \longrightarrow \mathbb{N}$. Self-Reference!

Basic Definitions and Theorems

$$K_{Sammy}(\mathbb{C}) = K(\mathbb{C}) = \text{ The smallest number of operations needed to describe } \mathbb{C}.$$

An *invariance theorem*. The Kolmogorov complexity does not depend on which programing language is used.

Theorem

There exists a constant c such that for all categorical structures $\mathbf{X}$ *we have* $|K_{Sammy}(\mathbf{X}) - K_{Saunders}(\mathbf{X})| \leq c$.

Basic Definitions and Theorems

$$K_{Sammy}(\mathbb{C}) = K(\mathbb{C}) = \text{ The smallest number of operations needed to describe } \mathbb{C}.$$

An *invariance theorem*. The Kolmogorov complexity does not depend on which programing language is used.

Theorem

There exists a constant c such that for all categorical structures $\mathbf{X}$ we have $|K_{Sammy}(\mathbf{X}) - K_{Saunders}(\mathbf{X})| \leq c$.

Theorem

There exists a constant c_{Kan} such that for all $G : \mathbb{A} \longrightarrow \mathbb{B}$ and $F : \mathbb{A} \longrightarrow \mathbb{C}$ if $(Lan_G(F), \alpha)$ is the left Kan extension, then

$$K((Lan_G(F), \alpha)) \leq K(F) + K(G|F) + c_{Kan}$$

Basic Theorems

Theorem

If $\mathbb{A}$ *and* $\mathbb{B}$ *are two equivalent categories, then* $K_{Sammy}(\mathbb{A}) \approx K_{Sammy}(\mathbb{B})$.

Basic Theorems

Theorem

If $\mathbb{A}$ and $\mathbb{B}$ are two equivalent categories, then $K_{Sammy}(\mathbb{A}) \approx K_{Sammy}(\mathbb{B})$.

Conclusion:

Kolmogorov complexity is an invariant of categorical structure.

Computing with Sammy

- The coequalizer $\mathbf{1} \underset{t}{\overset{s}{\rightrightarrows}} \mathbf{2} \xrightarrow{\rho} \omega$ gives the (infinite) natural numbers as a monoid.
- $\mathbb{N} = \omega^{\mathbf{2}}$ gives the totally ordered category of natural numbers: $0 \longrightarrow 1 \longrightarrow 2 \longrightarrow \cdots$
- $P : \mathbf{1} \longrightarrow \mathbb{N}$ is a natural number.

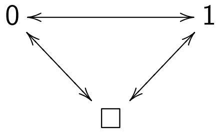
Computing with Sammy

- The coequalizer $\mathbf{1} \overset{s}{\underset{t}{\rightrightarrows}} \mathbf{2} \xrightarrow{\rho} \omega$ gives the (infinite) natural numbers as a monoid.
- $\mathbb{N} = \omega^{\mathbf{2}}$ gives the totally ordered category of natural numbers: $0 \longrightarrow 1 \longrightarrow 2 \longrightarrow \cdots$
- $P : \mathbf{1} \longrightarrow \mathbb{N}$ is a natural number.

Theorem

Any partially computable function of natural numbers can be computed with Sammy.

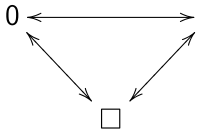
Mimicking Turing machines

- $\mathbb{N} = \ 0 \longrightarrow 1 \longrightarrow 2 \longrightarrow \cdots$ An infinite Turing machine tape
- $P : \mathbf{1} \longrightarrow \mathbb{N}$ is the position on the tape.
- $\widehat{\mathbf{3}} =$ 

. The alphabet.
- $F : \mathbb{N} \longrightarrow \widehat{\mathbf{3}}$ assigns to every position of the tape a 0, 1, or $\square$

Mimicking Turing machines

- $\mathbb{N} = 0 \longrightarrow 1 \longrightarrow 2 \longrightarrow \cdots$ An infinite Turing machine tape
- $P : \mathbf{1} \longrightarrow \mathbb{N}$ is the position on the tape.
- $\widehat{\mathbf{3}} =$



. The alphabet.

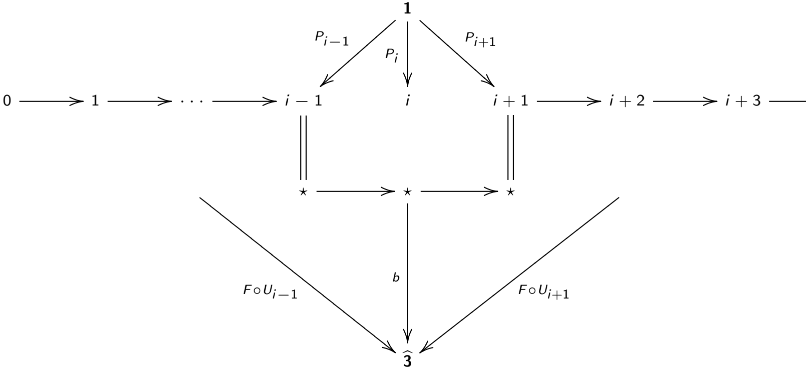
- $F : \mathbb{N} \longrightarrow \widehat{\mathbf{3}}$ assigns to every position of the tape a 0, 1, or $\square$

Theorem

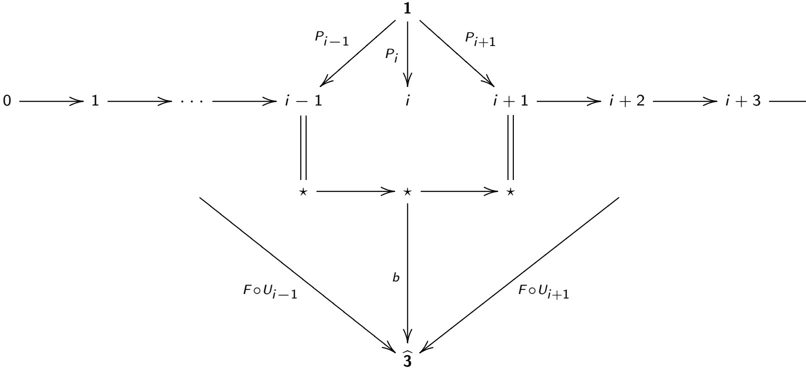
For s a string, there is a $F_s : \mathbb{N} \longrightarrow \widehat{\mathbf{3}}$ that describes s.

$$K_{Classical}(s) = O(K_{Sammy}(F_s))$$

Mimicking Turing machines



Mimicking Turing machines



Conclusion:

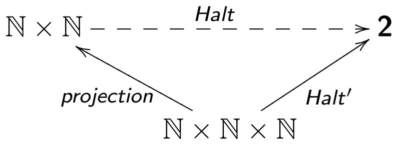
Our Kolmogorov complexity is a generalization of classical Kolmogorov complexity.

The Power of Categories

The following predicate is totally computable and hence constructible in Sammy: $Halt'(x, y, t) = 1$ if Turing machine y on input x stops within t steps.

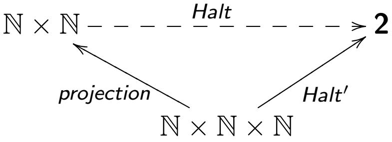
The Power of Categories

The following predicate is totally computable and hence constructible in Sammy: $Halt'(x, y, t) = 1$ if Turing machine y on input x stops within t steps.



The Power of Categories

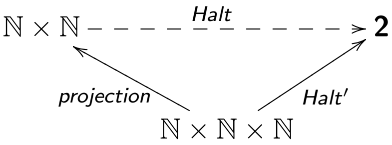
The following predicate is totally computable and hence constructible in Sammy: $Halt'(x, y, t) = 1$ if Turing machine y on input x stops within t steps.



Essentially: $Halt(x, y) = Colimit_t Halt'(x, y, t)$.

The Power of Categories

The following predicate is totally computable and hence constructible in Sammy: $Halt'(x, y, t) = 1$ if Turing machine y on input x stops within t steps.

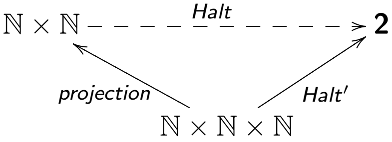


Essentially: $Halt(x, y) = Colimit_t Halt'(x, y, t)$.

Sammy can solve the Halting problem.

The Power of Categories

The following predicate is totally computable and hence constructible in Sammy: $Halt'(x, y, t) = 1$ if Turing machine y on input x stops within t steps.



Essentially: $Halt(x, y) = Colimit_t Halt'(x, y, t)$.

Sammy can solve the Halting problem.

Conclusion:

Our Kolmogorov complexity is a PROPER generalization of classical Kolmogorov complexity.

The Power of Categories

- Can Sammy solve everything?

The Power of Categories

- Can Sammy solve everything?
- No.

The Power of Categories

- Can Sammy solve everything?
- No.

Theorem

The functor $K_{Sammy} : \mathbb{C}at \longrightarrow \mathbb{N}$ is not constructible with any Sammy program.

The Power of Categories

- Can Sammy solve everything?
- No.

Theorem

The functor $K_{Sammy} : \mathbb{C}at \longrightarrow \mathbb{N}$ is not constructible with any Sammy program.

- So what exactly is the power of categorical constructions?

The Power of Categories

- Can Sammy solve everything?
- No.

Theorem

The functor $K_{Sammy} : \mathbb{C}at \longrightarrow \mathbb{N}$ is not constructible with any Sammy program.

- So what exactly is the power of categorical constructions?
- Conjecture: I think it goes through the arithmetic hierarchy and stops at some level of the projective hierarchy.

Algebraic Structure and More

Theorem

$\mathbb{T}$ an algebraic theory. $K(\mathbb{T}) \approx K(Alg(\mathbb{T}, Set))$.

Theorem

If $\mathbb{T}$ is Morita equivalent to $\mathbb{T}'$, then $K(\mathbb{T}) \approx K(\mathbb{T}')$.

Theorem

A monad has the same Kolmogorov complexity as its category of Eilenberg-Moore algebras.

Theorem

Morita equivalent monads have equal Kolmogorov complexity.

Algebraic Structure and More

Theorem

$\mathbb{T}$ an algebraic theory. $K(\mathbb{T}) \approx K(Alg(\mathbb{T}, Set))$.

Theorem

If $\mathbb{T}$ is Morita equivalent to $\mathbb{T}'$, then $K(\mathbb{T}) \approx K(\mathbb{T}')$.

Theorem

A monad has the same Kolmogorov complexity as its category of Eilenberg-Moore algebras.

Theorem

Morita equivalent monads have equal Kolmogorov complexity.

Conclusion:

Kolmogorov complexity is an invariant of algebraic structure.

Generalizations

- Categories with all (finite) (co)products
- Categories with all (finite) (co)limits
- Monoidal categories, symmetric monoidal categories, braided monoidal categories, closed categories, etc.
- Enriched categories
- The myriad definitions of weak higher categories, strict higher categories, etc.
- Pare's double-categories
- Joyal's quasi-categories
- Luria's (infinity, n)-categories, etc.
- Categories with Quillen model structures
- Categories with factorization systems
- etc. etc.

Entropy

- There is a relationship between classical Kolmogorov complexity and Shannon's entropy theory.
- $K(x)$ measures the complexity of an individual string
- $H(X)$ measures the complexity of a source of strings, or a whole class of strings.
- $H(X)$ is the average of all the $K(x)$ where x is a string that can be produced by X. $H(X) \approx \sum_{x_i \in X} p(x_i)K(x_i)$.

Entropy for categorical structures:

- Entropy of a category $\mathbb{C}$: $H(\mathbb{C}) = Log_2|Aut(\mathbb{C})|$ (or $H(\mathbb{C}) = pLog_2 \frac{1}{|Aut(\mathbb{C})|}$)
- Entropy of a functor $F : \mathbb{C} \longrightarrow \mathbb{D}$: $H(F) = Log_2|Aut(F)|$
- Entropy of a particular object c in a category $\mathbb{C}$: entropy of the functor $P_c : \mathbf{1} \longrightarrow \mathbb{C}$ that "picks" an object $c \in \mathbb{C}$. $H(c) = H(P_c : \mathbf{1} \longrightarrow \mathbb{C}) = Log_2|Aut(P_c)|$.

Incompleteness via Complexity

Theorem

Consider a consistent, sound, finitely-specified theory, $\mathbf{T}$, *strong enough to formalize arithmetic. There exists a constant* $c_{\mathbf{T}}$, *which depends upon a universal Turing machine* U *and* $\mathbf{T}$ *such that for all but a finite number of* x, *the statements "*$K(x) > n$*," where* $n > c_{\mathbf{T}}$ *will be true but unprovable.*

Incompleteness via Complexity

Theorem

Consider a consistent, sound, finitely-specified theory, $\mathbf{T}$, strong enough to formalize arithmetic. There exists a constant $c_{\mathbf{T}}$, which depends upon a universal Turing machine U and $\mathbf{T}$ such that for all but a finite number of x, the statements "$K(x) > n$," where $n > c_{\mathbf{T}}$ will be true but unprovable.

- By Gregory Chaitin (and Christian Calude).
- The theorem essentially says that a logical theory cannot prove a theorem that is more powerful than the theory itself. "A fifty pound logical system cannot prove a 75 pound theorem."
- We want to understand categorical structures and how much of a phenomenon they can hold.

Entanglement and Special Relativity

- One of the central aspects of quantum information theory is the notion of entanglement.
- If you observe a particle and it is in the spin-up direction, then you instantly know that the entangled twin which is light years away is spinning down.
- Special relativity theory says that one cannot transmit information faster than the speed of light.
- Physicists tell us that entanglement is, in fact, not a violation of the special theory of relativity because this type of information is not what is restricted.
- What type of information does entanglement give?
- What type of information does special relativity restrict?
- We believe that the Kolmogorov complexity measure will be helpful in disentangling these ideas.

Occam's Razor

- Occam's razor is usually seen as a criterion by which to judge different physical theories.
- A theory:
  F : "Physical Phenomena" $\longrightarrow$ "Mathematical Structure"

Occam's Razor

- Occam's razor is usually seen as a criterion by which to judge different physical theories.
- A theory:
  $F :$ "Physical Phenomena" $\longrightarrow$ "Mathematical Structure"
- Universality of the theory demands that the category of "Physical Phenomena" be as large as possible.
- Occam's razor demands that "Mathematical Structure" has low informational content.
- We are interested in using Kolmogorov complexity on both of these categories and the functor to better understand "Why does Occam's razor work so well?"

The End

Thank You