A synthetic undecidability proof of Kolmogorov complexity

Nils Lauermann

Advisor: Fabian Kunze

Programming Systems Lab
Saarland University

June 24, 2021

Background

2021 - Catt/Norrish:
On the Formalisation of
Kolmogorov Complexity
(HOL4)

2021 - Forster et al.:
A Constructive and Synthetic
Theory of Reducibility
(Coq)

The Framework

Model of Computation[1]

$$T : \underbrace{\mathbb{N}}_{\text{code}} \rightarrow \underbrace{\mathbb{N}}_{\text{input}} \rightarrow \underbrace{\mathbb{N}}_{\text{steps}} \rightarrow \underbrace{\texttt{option } \mathbb{N}}_{\text{output}}$$

$$\forall nisr, T\ \ n\ \ i\ \ s = \texttt{Some}\ \ r \rightarrow \forall s', s' \geqslant s \rightarrow T\ \ n\ \ i\ \ s' = \texttt{Some}\ \ r$$

T represents a **partial function**!

[1] [Forster et al., 2021]

Church Thesis (CT)[1]

Assumption: **Every total function $\mathbb{N} \to \mathbb{N}$ is computable by T**

$$\texttt{Axiom}\ \ CT : \forall (f : \mathbb{N} \to \mathbb{N}), \exists (c : \mathbb{N}), \forall (x : \mathbb{N}), \exists s, T\ c\ x\ s = \texttt{Some}\ \ (f(x))$$

$\Rightarrow$ Every Coq function $\mathbb{N} \to \mathbb{N}$ is computed by a code c given by CT.

[1] [Forster et al., 2021]

Kolmogorov Complexity (KC)

Kolmogorov Complexity (KC)

$$kol \quad : \quad \underbrace{\mathbb{N}}_{\textbf{code}} \to \underbrace{\mathbb{N}}_{\text{number k}} \to \underbrace{\mathbb{N}}_{\text{KC of k}} \to \mathbb{P}$$

$$kol\ n\ k\ c \quad :\Leftrightarrow \quad \exists x : \mathbb{N}, \texttt{least}\ (\lambda x \Rightarrow \exists s, T\ n\ x\ s = \texttt{Some}\ \ k)\ x \wedge \log_2 x = c$$

Why $\log_2 x = c$?

Most proofs rely on length as metric (including Kummer's)

Notation: KC_c

$$KC_c(x) = y \sim kol\ c\ x\ y$$

Not all codes are equal!

Codes of interest

$CT\ (\lambda x \Rightarrow 1)$ **Not interesting!**

We want more general codes

Universal Codes

We will need a lot more generality:

Universal codes must simulate any other code with linear overhead!

Why do we need that?

- **Invariance Theorem**:

$$\textit{universal } c \rightarrow \forall c', \exists k, \forall x, KC_c(x) \leqslant KC_{c'}(x) + k$$

KC of function values:

$$\textit{universal } c \rightarrow \forall f : \mathbb{N} \rightarrow \mathbb{N}, \exists k, \forall m, KC_c(f(m)) \leqslant \log_2(m) + k$$

Idea: Simulate code received by (CT f)

From now on, c will be a universal code.

Incomputability of Kolmogorov Complexity

Berry Paradox

History

- first published in 1908[2]
- predates KC by more than 50 years

"The least integer not nameable in fewer than nineteen syllables"

[2] [Russell, 1908]

Berry Paradox for Kolmogorov Complexity

computable $KC_c \rightarrow$

computable $(\lambda x \Rightarrow$ **"The smallest natural number n with** $KC_c(n) > x$"$)$

Why is that function computable?

For all x there exists such an n:

- *universal* $c \rightarrow c$ can simulate identity function
- There are only 2^k numbers y with $log_2(y) = k \Rightarrow KC_c$ is unbounded

$\Rightarrow$ We can compute the least such number n when KC_c is computable

Contradiction!

Apply the function to the size s of itself:

$\Rightarrow KC_C(n) > s \wedge KC_C(n) \leqslant s$

KC_c is not computable!

Berry Paradox for Kolmogorov Complexity in Coq

```
Lemma incomputability (n : nat) :
    LEM → univ n → ¬(exists f, forall x, kol n x (f x)).
```

Excluded Middle is necessary for the unboundedness proof of KC_c

Conclusion

Conclusion

Contributions

- Formalisation of Kolmogorov Complexity in the synthetic setting in Coq
- Proving the **incomputability**, **invariance theorem** and various auxiliary lemmata

Difficulties

- Finding the most suitable definitions
- First concepts of the unboundedness proof of KC were much more involved

Conclusion

The road ahead

- Is Excluded Middle really necessary for the incomputability?
- Possible alternative approach to incomputability proof
- Investigating the relationship between different KC definitions
- Formalisation of Kummer's undecidability proof in Coq (assuming the construction)

Thank you!

References

References

Catt, E. and Norrish, M. (2021).
On the formalisation of kolmogorov complexity.
In *Proceedings of the 10th ACM SIGPLAN International Conference on Certified Programs and Proofs*, pages 291–299.

Forster, Y., Jahn, F., and Smolka, G. (2021).
A constructive and synthetic theory of reducibility.
Unpublished draft.

Russell, B. (1908).
Mathematical logic as based on the theory of types.
American journal of mathematics, 30(3):222–262.

LOC

Component	LOC
Preliminaries	244
Definitions	13
Invariance Theorem	26
Incomputability	252
Univ code constr.	125
Other lemmata	42
Def./proofs for Kummer (wip)	481
total	1165

Incomputability of Kolmogorov Complexity

Incomputability: Proof Outline[3]

Lemma 1:

$\forall n (f : \mathbb{N} \to \mathbb{N}), univ\ \ n \to \exists c : \mathbb{N}, \forall m\ \ k : \mathbb{N}, kol\ \ n\ \ (f(m))\ \ k \to k \leqslant \log_2 m + c$

n is a universal code:

Due to CT any function can be simulated with some constant overhead c

Theorem 2:

$\forall n : \mathbb{N}, LEM \to univ\ \ n \to \neg(\exists f : \mathbb{N} \to \mathbb{N}, \forall x : \mathbb{N}, kol\ \ n\ \ x\ \ (f(x)))$

Assume $f : \mathbb{N} \to \mathbb{N}$ with $\forall x : \mathbb{N}, kol\ \ n\ \ x\ \ (f(x))$

Define $g : \mathbb{N} \to \mathbb{N} := \lambda m \Rightarrow \min\{x : \mathbb{N} \mid m \leqslant f(x)\}$

$$\left.\begin{array}{l} \overset{\text{Def. g}}{\Rightarrow} \forall m, m \leqslant f(g(m))) \\ \overset{\text{Lem. 1}}{\Rightarrow} \exists c, \forall m, f(g(m)) \leqslant \log_2(m) + c \end{array}\right\} \quad (\exists c, \forall m, m \leqslant \log_2(m) + c) \to \bot$$

[3] [Catt and Norrish, 2021]

The proof in Coq

Define $g : \mathbb{N} \to \mathbb{N} := \lambda m \Rightarrow \min\{x : \mathbb{N} \mid m \leqslant f(x)\}$

- Use least witness operator
- We need to show: $\forall m : \mathbb{N}, \exists x : \mathbb{N}, m \leqslant f(x)$

$\forall m : \mathbb{N}, \neg\neg\exists x : \mathbb{N}, m \leqslant f(x)$

- To show: Kolmogorov Complexity is unbounded (for *univ n*)
- Create list L containing all outputs of n with all inputs of length $\leqslant m$
  - We need to know if n terminates
    $\Rightarrow$ Use Excluded Middle (through double negation)
- $\mathbb{N}$ is infinite: $\exists x, x \notin L$

$\Rightarrow m \leqslant f(x)$

Construction of a Universal Code

Construction of a Universal Code

Reminder: Universal Code

$$univ\ (n : \mathbb{N}) : \mathbb{P} := \forall m : \mathbb{N}, \exists g : \texttt{list}\ \mathbb{B}, \forall x : \mathbb{N}, (T\ \ m\ x) \approx (T\ \ n\ (\texttt{decode}(g +\!\!+\ \texttt{encode}\ \ x)))$$

- We require Church Thesis for partial functions (PCT):
- Define $(f : \mathbb{N} \to \mathbb{N} \to \texttt{option}\ \ \mathbb{N})$:
  - Receives an input $(\texttt{decode}(g +\!\!+\ \texttt{encode}\ \ x))$ and step count s
  - g contains the code m to be simulated:

$$g = \underbrace{\texttt{false} :: \cdots :: \texttt{false}}_{|\texttt{encode}\ \ m|} :: \texttt{true} :: \texttt{encode}\ \ \ m$$

  - return $(T\ \ m\ x\ s)$
- The code returned by (PCT f) is universal