An extended Rice's Theorem and some of its consequences

N. C. A. da Costa and F. A. Doria

February 8, 2020

Abstract

We state an extended version of Rice's theorem for classical formal languages which include elementary analysis and present and briefly comment some of its consequences. Other examples of noncomputability in strong formal languages are also discussed.

1 Introduction

Rice's Theorem is a devastating result when it comes to recursion theory and to several of its applications in computer science. Roughly, no nontrivial input–output property of Turing machines can be algorithmically verified. One would then expect that for extensions of arithmetic — as arithmetic is the cradle of computer science — we could hope that the richer structures would allow for the existence of some nontrivial testing procedures.

Alas, that's not the case. As we show in Section 2, for many ordinary extended theories including Peano Arithmetic (PA) and Zermelo–Fraenkel set theory (ZF), there is a very general extended Rice's Theorem in each of those theories. There are no general decision procedures for those theories; just like that. Undecidability appears to be the rule in mathematics, and not the contrary.

Section 3 presents some consequences of that extended Rice Theorem. We examine the basis of its surprising generality: we are dealing here with a kind of *linguistic* phenomenon; that is to say, undecidability stems from the formal language we use as the basis for our axiomatic background.

It doesn't seem possible to evade that difficulty.

2 The extended Rice's theorem

Rigorous characterizations are given in the references [2, 4]; we use a more informal approach. There is a central tool here, the *halting function* θ. (Recall that the halting function is the function that settles the halting problem in recursion theory.) We prove:

Proposition 2.1 *There is an explicit expression noted θ for the halting function which uses elementary functions and calculus operations.* $\square$

We now add some extra detail. Our formal background consists of an axiomatic theory which includes Peano Arithmetic, has a recursively enumerable set of theorems, and has a model with a standard arithmetic portion — that is, we require our theories to be sound.

Our theories are supposed to be consistent; axiomatics use the Suppes predicate techniques.

Let $\mathsf{M}_m(a)$ denote Turing machine of Gödel number m that has a as its input. Then we define:

- $\theta(m, a) = 1$ if and only if $\mathsf{M}_m(a)$ halts.
- $\theta(m, a) = 0$ if and only if $\mathsf{M}_m(a)$ diverges.

θ is the *halting function* for M (we use θ both for the halting function and for expressions that describe it through elementary functions). Now let σ be the sign function, $\sigma(\pm a) = \pm 1$, $\sigma(0) = 0$. Then:

Proposition 2.2 *The halting function is explicitly given by:*

$$\theta(n, q) = \sigma(G_{n,q}),$$

$$G_{n,q} = \int_{-\infty}^{+\infty} C_{n,q}(x) e^{-x^2} dx,$$
$$C_{n,q}(x) = |F_{n,q}(x) - 1| - (F_{n,q} - 1).$$
$$F_{n,q}(x) = \kappa_P(x) p_{n,q},$$

where κ_P is an adequate Richardson transform [2] and $p_{n,q}$ is a two–parameter universal Diophantine polynomial. $\square$

Notice that there are infinitely many expressions for the halting function in our language, and there is no decision procedure to check whether an arbitrary expression in that language is an expression for the halting function.

The extended Rice's Theorem

We now reach our main goal. Consider the same language as above and suppose that our theory has a recursively enumerable set of theorems. Moreover suppose that we can introduce predicate symbols $P, Q, \ldots$ in our language. We require:

- Predicate P is nontrivial if and only if there are $q_1 \neq q_2$ so that one proves both $P(q_1)$ and $\neg P(q_2)$.

Then:

Proposition 2.3 *There is a q_0 so that for any nontrivial P, neither $P(q_0)$ nor $\neg P(q_0)$ can be proved in our formal theory.*

Proof: There is a x_0 so that for a model $\mathbf{M}$ with standard arithmetic, $\mathbf{M} \models \theta(x_0) = 0$, while the sentences $\theta(x_0) =$ and $\theta(x_0) \neq 0$ can neither be proved nor disproved in our theory. Then put:

$$q_0 = (\theta(x_0)P(q_1) + (1 - \theta(x_0))P(q_2).$$

Notice that the result is valid for *any* nontrivial P. $\Box$

3 A few consequences

Several quite common questions are answered by applying the extended Rice. For example:

- Conjecture: *"Gödel incompletemess in physics is a consequence of the quantum substractum."*

  False [1]. Counterexample: let H_1 be the Hamiltonian for a classical harmonic oscillator, and let H_2 be the Hamiltonian for a free particle. Then, given $H_0 = \theta_0 H_1 + (1 - \theta_0)H_2$, with $\theta_0 = \theta(x_0)$, is such that the sentence "H_0 is a classical system," is proved in our axiomatic theory for classical mechanics, while we cannot formally decide which kind of classical mechanical system we are dealing with.

- Conjecture: *"Linear systems do not exhibit the Gödel phenomenon."*

  False [1]. Counterexample: if L_1 and L_2 are two different linear systems, then $L_0 = \theta_0 L_1 + (1 - \theta_0)L_2$. L_0 can be proved to be linear, but we cannot decide which linear system is described by L_0.

- Conjecture: *We can algorithmically separate chaotic from nonchaotic systems.*

  False [2]. The counterexample is constructed as in the previous examples.

Hard problems exist everywhere

Problems such as Fermat's Last Theorem (FLT) [3], or Riemann's Hypothesis (RH), can be formulated as Diophantine problems. Therefore we can apply the construction sketched above and formulate, given an arbitrary nontrivial predicate P, infinitely many problems as difficult as FLT or RH. One proceeds as follows: given the Diophantine polynomial p_{FLT} that codes some problem, say, FLT, we obtain the corresponding Richardson's transform and apply the other transformations to arrive at θ_{FLT}.

Details are in the reference.

Other examples

Every axiomatic theory with a recursively enumerable set of theorems, with enough arithmetic to talk about computable stuff and which is sound possesses akind of "bounding function," which can be seen as a computable Busy–Beaver–like function for that theory. Say, given PA, we have a family of fast–growing functions indexed by ordinals which is dominated in the standard model by function F_{ϵ_0} where index ϵ_0 denotes the ordinal that allows for the proof of the consistency of PA. Actually we can prove in PA [4]:

$$\text{“}\mathsf{F}_{\epsilon_0} \text{ is total”}. \rightarrow \text{ “PA is consistent.”}$$

(For any ordinal $\alpha < \epsilon_0$ we cannot prove it.) Notice that we can neither prove nor disprove that F_{ϵ_0} is in fact total. Actually it is intuitively total — but quite frequently intuitive axioms lead to counterintuitive results.[1]

We use that function to prove the following result:

Proposition 3.1 *We can explicitly and algorithmically generate a countably infinite set of algorithms of which it is true that they are polynomial algorithms, while one can neither prove nor disprove it within our theory.*

Sketch of proof: Consider the BGS machine [4] $\mathsf{P}_{m,p} = \langle \mathsf{M}_m, |x|^p + p \rangle$, where M_m is an arbitrary Turing machine, and $|x|^p + p$ is a polynomial bound of a clock that stops M_m after a polynomial clock whose bound is given by that expression.

Then consider the set $\mathsf{P}^{\mathsf{F}}_{m,p}(x) = \langle \mathsf{M}_m(x), |x|^{\mathsf{F}(p)} + \mathsf{F}(p) \rangle$, all m, p. $\square$

4 Comments

Undecidability and incompleteness as presented here are linguistic phenomena, that is to say, these phenomena depend on the language we use to describe the mathematical objects. On the other hand, as we can only access mathematical objects through a formal language, rich enough languages lead to undecidability and incompleteness.

Nevertheless many conjectures have been made on the possible existence of a relation between Heisenberg's Uncertainty Principle (UP) and Gödel incompleteness (from which we can derive undecidability). Penrose explicitly made such conjecture in his bestselling book *The Emperor's New Mind.* We've summarized Penrose's conjectures as follows:

- There are "usefully harnessable" noncomputable processes in Nature.
- However noncomputability cannot be found at the classical level, but for a few artificial examples, or for chaotic processes.

[1] Just consider the Axiom of Choice, which at first looks so intuitive.

- Moreover, if we ever manage to find noncomputability at the classical level, that kind of noncomputability cannot be controlled to allow us to step beyond the limits imposed by our current acceptable Turing–computable objects.

These three conjectures are false, as we have already seen.

As for the last conjecture, we have proposed that a combined analog + digital computer can lead to very interesting results and can perhaps extend in a nontrivial way our current concept of computation.

5 Acknowledgments

We thank Professor J. Acacio de Barros for discussions and criticism. FAD wishes to thank Professor C. A. Cosenza for comments. The present ideas are part of an ongoing discussion with Professor G. J. Chaitin. Support is acknowledged from CNPq–Brazil and the Production Engineering Program, Coppe/UFRJ, Rio, Brazil.

References

[1] N. C. A. da Costa and F. A. Doria, "Classical Physics and Penrose's Thesis," *Found. Phys. Letters*, **4**, 363–373 (1991).

[2] N. C. A. da Costa and F. A. Doria, "Undecidability and incompleteness in Classical Mechanics," *Int. J. Theoretical Phys.* **30**, 1041 – 1043 (1991).

[3] N. C. A. da Costa, F. A. Doria and A. F. Furtado do Amaral, "A dynamical system where proving chaos is equivalent to proving Fermat's conjecture," *Int. J. Theoretical Phys.* **32**, 2187 – 2206 (1993).

[4] F. A. Doria, "Axiomatics, the social sciences, and the Gödel phenomenon: a toolkit," in F. A. Doria, ed., *The Limits of Mathematical Modeling in the Social Sciences,* 1 – 90, World Scientific (2017).