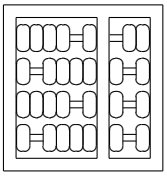

INSTITUTO DE COMPUTAÇÃO
UNIVERSIDADE ESTADUAL DE CAMPINAS

On an Abstract Theory of Computational Models and the Converse of Rice's Theorem

Igor Carboni Oliveira *Arnaldo Vieira Moura*
Walter Carnielli

Technical Report - IC-09-34 - Relatório Técnico

September - 2009 - Setembro

On an Abstract Theory of Computational Models and the Converse of Rice's Theorem

Igor Carboni Oliveira* Arnaldo Vieira Moura† Walter Carnielli‡

Abstract

In this paper we put forward an abstract definition of computational model and provide a negative answer to a conjecture involving the converse of Rice's theorem. In particular, any reasonable formal system in which the notion of computation is involved should satisfy our general definition of computational model. Furthermore, in order to give a natural counter-example to our conjecture we present a new interpretation to a result of Bernardi [1] involving undecidability in sufficiently strong formal theories. Finally, we raise the question if an abstract theory of computational models can lead to new problems and interesting results in theoretical computer science.

1 Introduction

H. G. Rice [5] proved that any property defined over the set of languages accepted by Turing machines is either trivial or undecidable. Although this fascinating result was published more than half a century ago, in this paper we ask for the first time the following question: Does the converse of Rice's Theorem hold? In other words, what is the relation between undecidability and expressiveness in computation?

Before formally introducing this question, we present a simple generalization of the undecidability of the halting problem. This basic result will lead us to a conjecture that is a generalization of the classical Rice's Theorem. Then, we show the conjecture to be false by presenting a counter-example to it. The ideas therein proved essential to formulate the converse of Rice's theorem.

After this initial discussion, we present a natural statement for the converse of Rice's Theorem. In addition, we briefly discuss how the Chomsky Hierarchy [7] provides intuitive arguments that may lead to the belief in the validity of this result. As a somewhat surprising and counter-intuitive result we then prove that the converse of Rice's Theorem does not hold.

Although our initial concern was about the validity of this last conjecture, the solution presented here is perhaps more interesting than the original problem. The problem, as stated, is a technical question from computability theory. However, we found that it was

*Institute of Computing, University of Campinas. Research supported by FAPESP grant 08/07040-0.
†Institute of Computing, University of Campinas. Research supported by FAPESP grant 02/07473-7.
‡Institute of Philosophy and Human Sciences, University of Campinas.

more profitable to search for a counter-example if the problem was investigated from a more conceptual viewpoint.

Accordingly, we introduce an abstract definition of computational model. In particular, any formal system in which the notion of computation is involved shall satisfy our general definition of computational model. We formulate a second conjecture in this new conceptual framework, and its solution is used to construct a counter-example to our original problem. This new conjecture, intuitively, asks for the following question: if every property of a computational model is undecidable, does it have too much power?

To solve this last problem, we provide a new interpretation to a result of Bernardi [1] that involves undecidability in Peano Arithmetic. The same result also holds for Robinson Arithmetic (Q). Due to its similarity with Rice's original result, we state it as Rice's Theorem for Q. This result is the fundamental step leading to a solution of the original problem.

Finally, we discuss how the abstract definition of computational model used in the proof of the conjecture can be used to prove interesting results about a wide range of traditional computational models. In particular, we exhibit a natural class of computational models that cannot solve their own halting problem. This leads us to raise the possibility for the development of an abstract theory of computational models.

2 The Converse of Rice's Theorem

In this section we state the main conjecture involving the converse of Rice's Theorem. We will adopt the following conventions.

Definition 1. ***[Preliminary Definitions].*** *Fix a finite alphabet Σ and let $\mathcal{T} = \{\langle M\rangle \mid M$ is a Turing machine$\}$ be the set of strings from Σ^* representing Turing Machines under a standard encoding. If M is a Turing machine, then $L(M)$ is the set of strings accepted by M. If $\mathcal{M} \subseteq \mathcal{T}$ is a set of strings representing Turing Machines, then $L(\mathcal{M}) = \{L \mid L = L(M)$ for some $\langle M\rangle \in \mathcal{M}\}$, i.e., the set of languages accepted by machines represented in $\mathcal{M}$. A property $P \subseteq S$, over a set S, is trivial if $P = \emptyset$ or $P = S$. We define RE to be the set of recursively enumerable languages over Σ. If S is a set, then $\mathcal{P}(S)$ denotes set of subsets of S. If $g : A \to B$ is a function, then the image of g is the set $Im(g) = \{g(a) \mid a \in A\}$. If T is a set of first-order sentences, then $Th(T)$ denotes the set of strings from Σ^* representing the theorems provable from T in first-order logic. The symbol λ represents the empty string. If w is a string then $|w|$ is the size of w.*

The next lemma is a simple generalization of the undecidability of the halting problem.

Lemma 1. ***[The Halting Problem for $\mathcal{M}$ is Undecidable].*** *Let $\mathcal{M} \subseteq \mathcal{T}$ be a decidable set satisfying $L(\mathcal{M}) = RE$. Then $H_{\mathcal{M}} = \{\langle M\rangle \mid \langle M\rangle \in \mathcal{M}$ and $\langle M\rangle \in L(M)\}$ is undecidable.*

Proof. This lemma follows from a similar result presented in section 3. □

Based on the previous lemma, our initial intuition was that the same path that leads to Rice's Theorem could be used to prove the following general result.

Conjecture 1. ***[Generalization of Rice's Theorem].*** *Let $\mathcal{M} \subseteq \mathcal{T}$ be a decidable set satisfying $L(\mathcal{M}) = RE$ and P be a non-trivial property over RE. Then the language $L^P_{\mathcal{M}} = \{\langle M\rangle \mid \langle M\rangle \in \mathcal{M}$ and $L(M) \in P\}$ is undecidable.*

Note that the original Rice's Theorem is the particular case in which $\mathcal{M} = \mathcal{T}$. The additional ingredient presented in these statements is that we are not trying to decide a property for the set of all Turing machines; we consider only a decidable set of machines that is as powerful as the whole set of Turing machines. Although lemma 1 holds, we will present later a simple counter-example to this conjecture.

Now consider the following possibility raised by the Chomsky Hierarchy [7]. Each level of this hierarchy corresponds to a particular class of grammars. Furthermore, there is a set of languages associated to each particular level. Since there is a constructive correspondence between type 0 grammars and Turing machines, it is impossible to decide any non-trivial property of languages if we are at the highest level of the hierarchy. However, it is possible to decide non-trivial properties of languages for the lower levels of the hierarchy. For instance, given a context-sensitive grammar, it is possible to decide if it produces a fixed string w. If we take one step further, we can investigate this phenomenon for any decidable set of grammars in addition to the original levels of the hierarchy. Motivated by the constructive equivalence between grammars and Turing machines (i.e., the function that converts grammars into Turing machines, and vice-versa, is computable), we started investigating the following problem.

Conjecture 2. ***[Converse of Rice's Theorem].*** *Let $\mathcal{M} \subseteq \mathcal{T}$ be a decidable set of Turing Machines satisfying:*

i) $L(\mathcal{M})$ is infinite;
ii) For every non-trivial property P of $L(\mathcal{M})$, the language $L^P_{\mathcal{M}} = \{\langle M\rangle \mid \langle M\rangle \in \mathcal{M}$ and $L(M) \in P\}$ is undecidable.

Then $L(\mathcal{M}) = RE$.

The first condition is a basic property of the levels within Chomsky Hierarchy. It also rules out the possibility for trivial counter-examples.

In the next section we provide a negative answer to both conjectures. In order to solve the second problem we introduce a new approach based on an abstract definition of computational model. This will enable us to present a natural counter-example to conjecture 2.

3 Abstract Computational Models

The notion of computation is an ubiquitous phenomenon in mathematics and science in general. Although it might not be explicit defined in some cases, it is possible to identify several structures in which this notion can be made precise. In this section we present a wider view of computation and give a general definition that may be useful to study the expressiveness of several formal systems and mathematical structures.

Definition 2. ***[Abstract Computational Model].*** *Let $I \subseteq \Sigma^*$ (instances of the computational model) and $l : I \to \mathcal{P}(\Sigma^*)$ (language association function). Then $\mathcal{C} = \langle I, l \rangle$ is an abstract computational model if $\mathcal{C}$ satisfies the following axioms:*

i) "Finite Representation Axiom"
I is a decidable set.

ii) "Algorithmic Behavior Axiom"
There exists a computable function $f : I \to \mathcal{T}$ such that, for every $i \in I$, if $f(i) = \langle M \rangle$ then $l(i) = L(M)$.

For succinctness, we will use the terms abstract computational model and computational model interchangeably.

The first axiom states that the instances of the computational model are finite objects and that there is an algorithm that decides whether a given string represents an instance of the computational model or not. Intuitively, it expresses that each instance has a finite number of instructions and that such instructions are well defined. The second item forces the computational model to have an algorithmic aspect, i.e., its language association function is not arbitrary. In addition, this axiom puts a limitation on the power of the abstract computational model: it is only able to recognize recursively enumerable languages. Note that the language association function may be based on any acceptance criterion; it is only required that axiom ii) holds.

We would like to make a philosophical observation at this point. Observe that axiom ii) implies that the abstract computational model can be simulated by a Turing machine. In particular, any instruction or operation of the computational model can be performed from a combination of the very simple rules of the Turing machine. Therefore, this new definition gives a very appealing way of reasserting the Church-Turing thesis.

It is straightforward to verify that many traditional computational models are also abstract computational models in the sense of definition 2. Among them, finite-state machines, pushdown automata, boolean circuits and Turing machines. Actually, it is clear that this definition encompasses most of (if not all) the formal system in which the notion of computation can be defined.

In addition, we may look at the expressiveness of certain structures that are not directly related to computation. For instance, we could have I representing the set of finite groups (described by their multiplication table) and let l be the function that associates to each particular group the set of sequences of elements whose multiplication results in the respective unit element.

Definition 3. ***[Expressiveness of Computational Models].*** *A computational model $\mathcal{C}_1 = \langle I_1, l_1 \rangle$ is stronger than a computational model $\mathcal{C}_2 = \langle I_2, l_2 \rangle$ if $Im(l_2) \subseteq Im(l_1)$, that is, $\mathcal{C}_2$ is not able to represent more languages than $\mathcal{C}_1$.*

Lemma 2. ***[Theories as Computational Models].*** *Let $I = \{\langle \psi \rangle \mid \psi$ is a first-order sentence $\}$ and $l(\langle \psi \rangle) = Th(\{\psi\})$. Then $\mathcal{C} = \langle I, l \rangle$ is a computational model.*

Proof. First, I is a decidable set since it is easy to verify if a given string w is a first-order formula: the algorithm just apply the recursive definition of well-formed formulae. Second, given a sentence ψ it is possible to construct a Turing machine M (that may not halt on some inputs) such that $L(M) = Th(\{\psi\})$. M generates all valid proofs and accepts a formula φ if and only if it eventually finds a proof of φ from the usual first-order axioms and ψ. The construction can easily be made algorithmic. Hence, there exists a computable function $f : I \to \mathcal{T}$ satisfying the conditions of definition 2 and therefore $\mathcal{C}$ is a computational model. □

The next lemma proves the following expected result: if a computational model is as powerful as the Turing Machine model, then its Halting Problem cannot be decided by Turing Machines. Note that it is a generalization of lemma 1.

Lemma 3. ***[The Halting Problem for $\mathcal{C}$ is Undecidable].*** *Let $C = \langle I, l\rangle$ be a computational model such that $Im(l) = RE$. Then $H_{\mathcal{C}} = \{i \mid i \in I \text{ and } i \in l(i)\}$ is undecidable.*

Proof. Consider the language $H'_{\mathcal{C}} = \{i \mid i \in I \text{ and } i \notin l(i)\}$. Since I is decidable, it follows that $H'_{\mathcal{C}}$ reduces to $H_{\mathcal{C}}$. It suffices to prove that $H'_{\mathcal{C}}$ is undecidable. Suppose that there exists a Turing machine M such that $L(M) = H'_{\mathcal{C}}$. Since $Im(l) = RE$, there exists $i \in I$ such that $l(i) = H'_{\mathcal{C}}$. Then we have the following contradiction:

$i \in H'_{\mathcal{C}}$ *iff*
$i \in I$ and $i \notin l(i)$ *iff*
$i \notin l(i)$ *iff*
$i \notin H'_{\mathcal{C}}$.

It follows that $H'_{\mathcal{C}}$ is undecidable. □

Next we state similar versions of the conjectures of the previous section, now using the formalism of abstract computational models. Although the first problem does not depend on these new ideas, we use this new conceptual framework to solve the second problem.

Definition 4. *Let $\mathcal{C} = \langle I, l\rangle$ be a computational model. $\mathcal{C}$ is said to be Rice undecidable if $Im(l)$ is infinite and, for every non-trivial property P over $Im(l)$, the language $L^P_{\mathcal{C}} = \{i \mid i \in I \text{ and } l(i) \in P\}$ is undecidable.*

Conjecture 3. ***[Generalization of Rice's Theorem - Conceptual Version].*** *Let $\mathcal{C} = \langle I, l\rangle$ be a computational model such that $Im(l) = RE$. Then $\mathcal{C}$ is Rice undecidable.*

Proposition 1. ***[Conjecture 3 does not hold].*** *There exists a computational model $\mathcal{C} = \langle I, l\rangle$ satisfying $Im(l) = RE$ that is not Rice undecidable.*

Proof. Let I be the set of strings representing Turing machines that halt on the empty word λ within at most k steps (k is an integer constant) and $l : I \to \mathcal{P}(\Sigma^*)$ be the usual language association function of Turing machines. Then $\mathcal{C} = \langle I, l\rangle$ is clearly a computational model (f is the identity function). In addition, $Im(l) = RE$, since for every language $L \in RE$ there is a Turing machine M recognizing L that halts on input λ in at most k steps (M

ignores its algorithm and promptly takes the right decision on input λ). Now consider a property P over $Im(l)$ where $l(i) \in P$ iff $\lambda \in l(i)$. Clearly, P is nontrivial over $Im(l)$. By the construction of I, we can easily decide if $\lambda \in l(i)$. Hence, the language $L_{\mathcal{C}}^{P} = \{i \mid i \in I$ and $\lambda \in l(i)\}$ is decidable. □

From the proof of the previous proposition it is immediate that conjecture 1 is not valid as well. This counter-example is somewhat artificial; it exists because the definition of computational model is too general. We discuss this issue in a subsequent section.

There is perhaps an intuitive explanation for this negative result. For sufficiently powerful computational models such as type 0 grammars, Turing Machines and recursive functions, there is an algorithm that transforms an instance of one model into an equivalent instance of the other model. Therefore, these computational models are all undecidable in the sense of Rice's theorem. However, it does not follow from the axioms of abstract computational model that such constructive correspondence exists for arbitrary abstract computational models.

Now we consider the conceptual version of the converse of Rice's Theorem. Intuitively, it expresses that if the non-trivial properties on an abstract computational model are undecidable, then it must be as powerful as the Turing Machine model.

Conjecture 4. ***[Converse of Rice's Theorem - Conceptual Version].*** *Let $\mathcal{C} = \langle I, l\rangle$ be a Rice undecidable computational model. Then $Im(l) = RE$.*

Proposition 2. ***[Conjecture 4 does not hold].*** *There exists a Rice undecidable computational model $\mathcal{C} = \langle I, l\rangle$ such that $Im(l) \neq RE$.*

Proof. The proof of this result is presented in the next section. □

Now we can solve the original problem involving Turing Machines.

Theorem 1. ***[The Converse of Rice's Theorem does not hold].*** *There is a decidable set $\mathcal{M} \subseteq \mathcal{T}$ of Turing Machines satisfying conditions i) and ii) of conjecture 2, and such that $L(\mathcal{M}) \neq RE$.*

Proof. Let $\mathcal{C} = \langle I, l\rangle$ be the counter-example to conjecture 4 provided by proposition 2. Let $f_{\mathcal{C}}$ be the computable function associated with $\mathcal{C}$ given by item (ii) of Definition 2. We construct a computable function $f'_{\mathcal{C}} : I \to \mathcal{T}$. Function $f'_{\mathcal{C}}$ computes exactly as $f_{\mathcal{C}}$, but it also adds enough irrelevant tuples to the description of the Turing machines thus produced. Formally, if $f_{\mathcal{C}}(i) = \langle M\rangle$, then $f'_{\mathcal{C}}(i) = \langle M'\rangle$ with $L(M) = L(M')$ and $|i| \leq |\langle M'\rangle|$.

First, it is easy to see that $Im(f'_{\mathcal{C}})$ is decidable, since $w \in Im(f'_{\mathcal{C}})$ if and only if there exists $i \in I$ satisfying $|i| \leq |w|$ such that $f'_{\mathcal{C}}(i) = w$. This last condition can be verified by an exhaustive search. Let $\mathcal{M} = Im(f'_{\mathcal{C}})$.

Second, by the definition of computational model and the construction of $f'_{\mathcal{C}}$, it follows that $L(\mathcal{M}) = L(Im(f'_{\mathcal{C}})) = L(Im(f_{\mathcal{C}})) = Im(l)$. Since $\mathcal{C}$ is Rice undecidable, then $L(\mathcal{M})$ is infinite.

Third, if the language $L_{\mathcal{M}}^{P} = \{\langle M\rangle \mid \langle M\rangle \in \mathcal{M}$ and $L(M) \in P\}$ is decidable for some non-trivial property P of $L(M)$, then the language $L_{\mathcal{C}}^{P} = \{i \mid i \in I$ and $l(i) \in P\}$ is also decidable, contradicting again the fact that $\mathcal{C}$ is Rice undecidable.

Finally, $L(\mathcal{M}) \neq RE$, since $L(\mathcal{M}) = Im(l)$ and $\mathcal{C}$ is a counter-example to conjecture 4. To sum up, we have proved that $\mathcal{M}$ is a counter-example to conjecture 2.

□

4 Rice's Theorem and the Theory Q

The proof of proposition 2 is based on an undecidability result regarding Robinson Arithmetic (Q). For completeness, we present a simplified proof for a particular case that is sufficient for our purposes. A general development can be found in Bernardi [1].

Definition 5. ***[Properties on Theories].*** *Let T be a first-order theory and Γ be a set of sentences. Γ is a property on T if the following holds for any sentences φ and ψ:*

$$T \vdash \varphi \leftrightarrow \psi \implies \big[\varphi \in \Gamma \iff \psi \in \Gamma\big].$$

A property Γ is trivial if it is the empty set or the set of all sentences.

Therefore, a property on a theory is just the union of equivalence classes of sentences under the relation of provable equivalence. We will need some standard definitions from logic.

Definition 6. ***[Additional Definitions].*** *Q denotes the usual axioms of Robinson Arithmetic. For each formula φ, $\ulcorner\varphi\urcorner$ denotes the Gödel number of φ. A property Γ on a theory T is decidable if the set $\{\ulcorner\varphi\urcorner \mid \varphi \in \Gamma\}$ is decidable. $\overline{0}$ denotes the term 0. $\overline{n}$ denotes the term $s(\overline{n-1})$ for each positive integer n. If T is a first-order theory, then $Mod(T)$ denotes the set of models of T. Two theories T_1 and T_2 are said to be incompatible if they have no common consistent extension. A set Δ of theories is incompatible if for any two distinct theories $T_1, T_2 \in \Delta$, T_1 and T_2 are incompatible theories.*

Definition 7. ***[Definition of Representability].*** *Let S be a set of natural numbers and T be a theory. S is representable in T if there exists a formula φ with one free variable such that for any natural number n,*

$$\begin{array}{ll} T \vdash \varphi(\overline{n}) & \text{if } n \in S, \\ T \vdash \neg\varphi(\overline{n}) & \text{otherwise.} \end{array}$$

Lemma 4. ***[Representability of Decidable Sets].*** *Every decidable set of natural numbers is representable in Q.*

Proof. See e.g. Carnielli [3]. □

Lemma 5. ***[Diagonal Lemma].*** *For any formula φ with one free variable, there exists a sentence ψ such that*

$$Q \vdash \psi \leftrightarrow \varphi(\overline{\ulcorner\psi\urcorner}).$$

Proof. See e.g. Boolos [4]. □

The following result is a special case of a theorem proved in Bernardi [1].

Theorem 2. ***[Undecidability involving Provable Equivalence].*** *Every non-trivial property on Q is undecidable.*

Proof. Suppose that Γ is a non-trivial decidable property on Q. By the non-triviality of Γ, there exists sentences $\psi_1 \in \Gamma$ and $\psi_0 \notin \Gamma$. Since Γ is decidable, by lemma 4 there exists a formula φ such that for any sentence ψ,

$$\begin{array}{ll} Q \vdash \varphi(\overline{\ulcorner \psi \urcorner}) & \text{if } \psi \in \Gamma, \quad (1) \\ Q \vdash \neg\varphi(\overline{\ulcorner \psi \urcorner}) & \text{otherwise.} \quad (2) \end{array}$$

Now consider the formula with a free variable z, given by

$$(\varphi(z) \rightarrow \psi_0) \wedge (\neg\varphi(z) \rightarrow \psi_1)$$

Then, by lemma 5, there exists a sentence ψ^* such that

$$Q \vdash \psi^* \leftrightarrow \quad (\varphi(\overline{\ulcorner \psi^* \urcorner}) \rightarrow \psi_0) \wedge (\neg\varphi(\overline{\ulcorner \psi^* \urcorner}) \rightarrow \psi_1). \quad (3)$$

When $\psi^* \in \Gamma$, we get $Q \vdash \psi^* \leftrightarrow \psi_0$ by (1) and (3). Therefore $\psi^* \notin \Gamma$ since $\psi_0 \notin \Gamma$ and Γ is a property on Q. This implies a contradiction. The other case is similar. □

Now we are able to construct a counter-example to proposition 2.

Definition 8. ***[Computational Model $\mathcal{Q}$].*** *Let $I_{\mathcal{Q}} = \{\langle\psi\rangle \mid \psi$ is a first-order sentence in the language of Arithmetic$\}$ and $l_{\mathcal{Q}}(\langle\psi\rangle) = Th(\, Q \cup \{\psi\})$. We define $\mathcal{Q} = \langle I_{\mathcal{Q}}, l_{\mathcal{Q}}\rangle$.*

Since Q is a finitely axiomatizable theory, it is clear from the proof of lemma 2 that $\mathcal{Q}$ is a computational model.

Lemma 6. ***[Infinite Extensions of Theory Q].*** *$Im(l_{\mathcal{Q}})$ is infinite.*

Proof. We prove by induction that for every integer k there exists at least k consistent incompatible recursively axiomatizable extensions of Q. Since in this proof Q will always be extended only with a finite number of additional axioms, there is a single sentence (the conjunction of these axioms) that gives rise to the same extension. Hence, in particular, $Im(l_{\mathcal{Q}})$ cannot be finite.

The base case ($k = 1$) is satisfied by Q. Now assume that the result holds for some $k \geq 1$, i.e., there exist theories $T_1, ..., T_k$ that are consistent incompatible recursively axiomatizable extensions of Q. Since Q is an essentially undecidable theory (see e.g. Carnielli [3]) and these theories are recursively axiomatizable, they must be incomplete (otherwise they would be decidable). In other words, there are sentences $\varphi_1, ..., \varphi_k$ such that, for $1 \leq i \leq k$, $T_i \nvdash \varphi_i$ and $T_i \nvdash \neg\varphi_i$. If T is a first-order theory and ψ is a sentence, then $T \cup \{\psi\}$ is consistent if and only if $T \nvdash \neg\psi$ (see e.g. Mendelson). Furthermore, if two theories are incompatible then two consistent extension of them are incompatible as well. Hence it follows that $T_1 \cup \{\varphi_1\}$, $T_1 \cup \{\neg\varphi_1\}$, ... , $T_k \cup \{\varphi_k\}$, $T_k \cup \{\neg\varphi_k\}$ is a set of $2k$ consistent incompatible recursively axiomatizable extensions of Q. The induction step is complete and the result follows. □

Lemma 7. *If T_1 and T_2 are first-order theories, then $Mod(T_1) = Mod(T_2)$ if and only if $Th(T_1) = Th(T_2)$.*

Proof. See e.g. Shoenfield [2]. □

Lemma 8. *If T is a first-order theory and φ and ψ are sentences, then $T \vdash \varphi \leftrightarrow \psi$ if and only if $Th(T \cup \{\varphi\}) = Th(T \cup \{\psi\})$.*

Proof. By lemma 7 it is sufficient to prove the following equivalence: $T \vdash \varphi \leftrightarrow \psi$ if and only if $Mod(T \cup \{\varphi\}) = Mod(T \cup \{\psi\})$.

First, suppose that $T \vdash \varphi \leftrightarrow \psi$. By the Gödel Completeness Theorem, $T \vDash \varphi \leftrightarrow \psi$, i.e., if $\mathcal{A} \vDash T$ ($\mathcal{A}$ is a model of T) then $\mathcal{A} \vDash \varphi \leftrightarrow \psi$. Let $\mathcal{A} \in Mod(T \cup \{\varphi\})$. Then $\mathcal{A} \vDash T$ and $\mathcal{A} \vDash \varphi$. Since $T \vDash \varphi \leftrightarrow \psi$, we have that $\mathcal{A} \vDash \psi$ and hence $\mathcal{A} \in Mod(T \cup \{\psi\}$. Therefore $Mod(T \cup \{\varphi\}) \subseteq Mod(T \cup \{\psi\})$. The proof of the other inclusion is similar.

Conversely, suppose that $Mod(T \cup \{\varphi\}) = Mod(T \cup \{\psi\})$ and that $T \nvdash \varphi \leftrightarrow \psi$. By the Gödel Completeness Theorem $T \nvDash \varphi \leftrightarrow \psi$, i.e., there exists a structure $\mathcal{A}$ such that $\mathcal{A} \vDash T$ but $\mathcal{A} \nvDash \varphi \leftrightarrow \psi$. Assume, without loss of generality, that $\mathcal{A} \vDash \varphi$ but $\mathcal{A} \nvDash \psi$. Hence $\mathcal{A} \in Mod(T \cup \{\varphi\})$ and $\mathcal{A} \notin Mod(T \cup \{\psi\})$, contradicting the original assumption that $Mod(T \cup \{\varphi\}) = Mod(T \cup \{\psi\})$. □

Theorem 3. ***["Rice's Theorem holds for Theory Q"].*** *$\mathcal{Q}$ is a Rice undecidable computational model.*

Proof. By definition 4 and lemma 6, it remains to prove that if P is a non-trivial property over $Im(l_{\mathcal{Q}})$, then the language $L_Q^P = \{i \mid i \in I_{\mathcal{Q}} \text{ and } l_{\mathcal{Q}}(i) \in P\}$ is undecidable. By the definition of $\mathcal{Q}$, the language L_Q^P is equivalent to the set $\Gamma_P = \{\phi \mid \phi \text{ is a sentence and } Th(Q \cup \{\phi\}) \in P\}$.

First, we prove that Γ_P is a property on theory Q in the sense of definition 5. Suppose that $Q \vdash \varphi \leftrightarrow \psi$ for sentences φ and ψ in the language of arithmetic. Then

$\varphi \in \Gamma_P$ *iff* (by the definition of Γ_P)
$Th(Q \cup \{\varphi\}) \in P$ *iff* (by lemma 8)
$Th(Q \cup \{\psi\}) \in P$ *iff* (by the definition of Γ_P)
$\psi \in \Gamma_P$.

Therefore Γ_P is a property on Q.

Finally, since $P \subseteq Im(l_{\mathcal{Q}})$ and P is non-trivial, there exist sentences φ and ψ such that $l_{\mathcal{Q}}(\langle\varphi\rangle) \in P$ and $l_{\mathcal{Q}}(\langle\psi\rangle) \notin P$, i.e., $Th(Q \cup \{\varphi\}) \in P$ and $Th(Q \cup \{\psi\}) \notin P$. Hence $\varphi \in \Gamma_P$ and $\psi \notin \Gamma_P$, that is, Γ_P is non-trivial. By theorem 2, Γ_P is undecidable and the proof that $\mathcal{Q}$ is a Rice undecidable computational model follows.

□

Corollary 1. ***[Counter-Example].*** *The computational model $\mathcal{Q}$ contradicts conjecture 4.*

Proof. By theorem 3, $\mathcal{Q}$ is a Rice undecidable computational model. By the definition of $\mathcal{Q}$, every language in $Im(l_{\mathcal{Q}})$ is infinite, since for any sentence φ the theory $Q \cup \{\varphi\}$ proves

infinite theorems. Therefore, there are no finite languages in $Im(l_{\mathcal{Q}})$ and so $Im(l_{\mathcal{Q}}) \neq RE$. □

5 A Glimpse Beyond

In this section we discuss further ideas to be explored based on our previous definitions and results. First, suppose that we add the following axiom to our general definition of computational model.

Definition 9. ***[Closure under Language Complementation Axiom].*** *Let $\mathcal{C} = \langle I, l \rangle$ be a computational model. $\mathcal{C}$ is closed under language complementation if for every language $L \in Im(l)$, we get $\Sigma^* \backslash L \in Im(l)$.*

From this basic additional axiom, it is possible to prove a simple yet interesting result about a wide range of computational models.

Proposition 3. ***[Halting Problem for C is Undecidable in C].*** *Let $\mathcal{C} = \langle I, l \rangle$ be a computational model that is closed under language complementation. Define $H_{\mathcal{C}} = \{i \mid i \in I \text{ and } i \in l(i)\}$. Then $H_{\mathcal{C}} \notin Im(l)$.*

Proof. Suppose that $H_{\mathcal{C}}$ is decidable in $\mathcal{C}$, that is, there exists $i \in I$ such that $l(i) = H_{\mathcal{C}}$. Since $\mathcal{C}$ is closed under language complementation, there exists $\overline{k} \in I$ such that $l(\overline{k}) = \Sigma^* \backslash H_{\mathcal{C}}$. The following contradiction follows:

$\overline{k} \in l(\overline{k})$ *iff*
$\overline{k} \in \Sigma^* \backslash H_{\mathcal{C}}$ *iff*
$\overline{k} \notin H_{\mathcal{C}}$ *iff*
$\neg[\, \overline{k} \in I$ and $\overline{k} \in l(\overline{k})]$ *iff*
$\overline{k} \notin I$ or $\overline{k} \notin l(\overline{k})$ *iff*
$\overline{k} \notin l(\overline{k})$.

Therefore $H_{\mathcal{C}} \notin Im(l)$. □

Moreover, the problems discussed in this paper can be studied from a more general perspective, as it is suggested by the next definition.

Definition 10. ***[Decidability between Computational Models].*** *Let $\mathcal{C}_1 = \langle I_1, l_1 \rangle$ and $\mathcal{C}_2 = \langle I_2, l_2 \rangle$ be computational models. A language L is semidecidable in $\mathcal{C}_1$ if $L \in Im(l_1)$. A language L' is decidable in $\mathcal{C}_1$ if both L' and $\Sigma^* \backslash L'$ are semidecidable in $\mathcal{C}_1$. Otherwise L' is undecidable in $\mathcal{C}_1$. $\mathcal{C}_1$ is Rice undecidable in $\mathcal{C}_2$ if $L^P_{\mathcal{C}_1}$ is undecidable in $\mathcal{C}_2$ for all nontrivial property P over $\mathcal{C}_1$.*

We could also study general properties of other restricted classes of computational models. For instance, if the next axiom is added to the general definition of computational model, the structure $\mathcal{Q} = \langle I_{\mathcal{Q}}, l_{\mathcal{Q}} \rangle$ does not belong to the new class of computational models.

Definition 11. ***[Halting Axiom].*** *Let $\mathcal{C} = \langle I, l \rangle$ be a computational model. $\mathcal{C}$ satisfies the halting axiom if there is a computable function $f_{\mathcal{C}}$ in the sense of definition 2 that satisfies the following additional requirement: if $\langle M \rangle \in Im(f_{\mathcal{C}})$ then M halts on all inputs.*

It is possible to give an informal comparison between these ideas and mathematical logic. In logic, a theory is defined by a general set of axioms (common to all theories) and some proper additional axioms. Here, we presented some general axioms that are to be satisfied by any computational model. Also, some other computational models may satisfy new additional axioms. Furthermore, in logic it is possible to study the models of a particular theory. Using the notion presented here we can study specific models that satisfy the general axioms of abstract computational models plus some other proper axioms. For example, finite-state machines are closed under language complementation, and satisfy the other general axioms.

Finally, in a recent paper, Kudlek [8] discusses the existence of universal machines for traditional computational models that are weaker than Turing machines. For example: Is there a pushdown automaton that is able to simulate any other pushdown automaton? In particular, the author puts into discussion the existence of a general definition for universality that could encompass all the cases considered in his paper. We proceed to show that it is possible to define universality in a very natural way using our abstract definition of computational model.

Definition 12. ***[Definition of Universal Machine].*** *Let $\mathcal{C} = \langle I, l \rangle$ be a computational model. Let $l_U : I \times I \to \Sigma^*$ be a function defined by $l_U(i, i') = \{w \mid w \in \Sigma^* \text{ and } i'w \in l(i)\}$. Then $\mathcal{C}$ admits an universal machine if there exists $u \in I$ such that for all $i' \in I$, $l_U(u, i') = l(i')$.*

It is possible to define several other computational concepts that are important in theoretical computer science in this general setting. The following questions arise: Are there other relevant problems and applications? Is it possible to develop an abstract theory of computational models? In our opinion, this is an interesting possibility. For instance, from some basic abstract axioms it is possible to develop a beautiful theory of computational complexity (see Blum [6]).

6 Conclusion

First, this paper is primarily motivated by the following question: What is the relation between undecidability and computational power? Rice's original result states that the computational power of Turing machines leads to general undecidability. We presented here evidence that it is possible to be general undecidable without being able to compute everything.

Second, our development shows that some technical questions become easier if approached from a more conceptual perspective. The following quote by I. N. Herstein [9] is appropriate.

> *"Very often in mathematics the crucial problem is to recognize and discover what are the relevant concepts; once this is accomplished the job may be more than half done."*

In our case, the introduction of a new concept shed light on new ideas and was essential for the construction of a natural counter-example to our original problem.

Finally, in this paper we put into discussion the following issues.

i) What is a computational model?

ii) Is it possible to develop an abstract theory of computational models?

We introduced an abstract definition of computational model. It is possible to argue that this definition is too general. However, new axioms can be added to restrict the class of formal systems satisfying the abstract definition. This gives a partial answer to the first question.

Lastly, we do not yet have an answer to the second question. Nevertheless, we believe that the study of such abstract computational models may bring new concepts and provide new insights into the traditional study of computational models in the field of theoretical computer science.

Acknowledegment

We would like to thank Anderson de Araujo (University of Campinas) and Hirofumi Yoshikawa (Tokyo Institute of Technology) for useful suggestions. In particular, the latter pointed out to us the existence of a simplified counter-example to conjecture 2 that can be obtained by diagonalization. We adopted the present counter-example to motivate the definition of computational model and due to its conceptual relevance.

References

[1] Bernardi C. On the Relation Provable Equivalence and on Partitions in Effectively inseparable sets. Studia Logica, 1981, 40:29–37.

[2] Shoenfield J.R. Mathematical Logic. Addison-Wesley Publishing, 1967.

[3] Epstein R., Carnielli W.A. Computability: Computable Functions, Logic and the Foundations of Mathematics. Wadsworth/Thomson Learning, 2000.

[4] Boolos G., Jeffrey R. Computability and Logic. Cambridge University Press, 1974.

[5] Rice, H.G. Classes of Recursively Enumerable Sets and Their Decision Problems. Trans. Amer. Math. Soc. 74, 358–366, 1953.

[6] Blum, M. A Machine-Independent Theory of the Complexity of Recursive Functions. Journal of the ACM, 14(2):322–336, 1967.

[7] Chomsky, N. Three Models for the Description of Language. IRE Transactions on Information Theory, 2:113–124, 1956.

[8] Kudlek, M. Some Considerations on Universality. arXiv:0906.3199v1 [cs.CC], 2009.

[9] Herstein, I. N. Topics in Algebra, second edition. John Wiley & Sons, 1975.