A Hypercomputational Alien

Udi Boker and Nachum Dershowitz

School of Computer Science, Tel Aviv University
Ramat Aviv, Tel Aviv 69978, Israel

Abstract

Is there a *physical* constant with the value of the halting function? An answer to this question, as holds true for other discussions of hypercomputation, assumes a fixed interpretation of nature by mathematical entities. We discuss the subjectiveness of viewing the mathematical properties of nature, and the possibility of comparing computational models having different views of the world. For that purpose, we propose a conceptual framework for power comparison, by linking computational models to hypothetical physical devices. Accordingly, we deduce a mathematical notion of relative computational power, allowing for the comparison of arbitrary models over arbitrary domains.

In addition, we claim that the method commonly used in the literature for "strictly more powerful" is problematic, as it allows for a model to be more powerful than itself. On the positive side, we prove that Turing machines and the recursive functions are "complete" models, in the sense that they are not susceptible to this anomaly, justifying the standard means of showing that a model is more powerful than Turing machines.

Key words: hypercomputation, Turing machine, computability, computational power, computational models

1 Prelude

E.V., youngest daughter of E.T., arrived on Earth a couple of days ago. Considering the sophisticated equipment of her spaceship, the speculation is that she—presumably like other of her planet's inhabitants—is capable of hypercomputation. E.V., in addition to her strange and fascinating beauty, acts extremely friendly and appears willing to share her advanced knowledge with

Email address: {udiboker,nachumd}@tau.ac.il (Udi Boker and Nachum Dershowitz).

us earthlings. The problem is an apparent lack of common language, and, moreover, the very different ways in which she and we humans seem to view the universe.

Yesterday, E.V. demonstrated for us an interesting device, one which we believe to be a computer of some sort, but with most peculiar input and output entities. E.V. seems to pay great attention while mixing colored clays and shaping them into a ball, which she proceeds to insert into the device, and then meaningfully stares at the colored ball sliding out of the machine a few moments later. It has been suggested that we compare the computational power of E.V.'s computer, henceforth nicknamed *Nuri*, with our own Turing machines (TMs). Now, it is clear that Nuri and TMs operate over different domains. Though unlikely, it might even be that Nuri's domain is of a different cardinality (for instance, it may be sensitive to non-enumerable colors or dimensions of the clay shapes). The question is: How can we know whether Nuri's architecture is as powerful, or perhaps more powerful, than TMs?

Nuri, as a physical device, should not be compared directly with our TM, which is an *abstract* computational model. It is very likely that Nuri's physical input/output entities will remain gibberish to us, exactly as the flashing pixels on our monitors might be undecipherable for E.V. Thus, the object we really wish to compare with the TM is the *computational model* of Nuri, *as viewed by* E.V. Generally speaking, every physical device might be understood as implementing various computational models, depending on the user's interpretation of its physical interface.

Today has seen a blossoming of suggestions for the possible computational models of Nuri. Before speculating whether a suggested computational model really fits the way E.V. sees Nuri, we need to define how to compare the power of the suggested models with TM. *Comparing the power of arbitrary computational models operating over arbitrary domains, is the main subject of this paper.*

What we are basically interested in knowing is if E.V. is capable of computing with a device (e.g. Nuri) that implements her model all that we humans can compute with a device (e.g. P.C.) that implements the TM. If yes, we'll say that E.V.'s model is as powerful as ours. If it turns out that E.V. can make additional computations with her Nuri, over and above those of our TMs, we will be forced to conclude that the Nuri model is actually *more* powerful; that is, it is hypercomputational.

Now, there are two common meanings to *hypercomputability*:

(1) Computing more than a TM. See, for example, [15, p. 1]: "Hypercomputation: computing more than the Turing machine".
(2) Computing a non-TM (*incomputable*) function. See, for example, [11], or

> [7, p. 1]: "Hypercomputation is the computation of functions or numbers that cannot be computed in the sense of Turing ...."

Assuming a fixed view of the world, the two meanings coincide: once you have an incomputable function you may add it as an oracle to TM and get the hypercomputation of the first definition. This is not the case, however, with E.V. She might compute what seems to us to be the halting function, while for her it's just the parity function, as she has a different view (coding) of her machines. So, to venture a claim that E.V. is capable of hypercomputation, we need to adopt the first definition above, which we will name *strong-hypercomputability*, to avoid any confusion.

The need to compare computational models embodying different worldviews is not unique to the current situation provoked by E.V. We find it also relevant, for example, when comparing a model operating over the reals with TM.

A reasonable starting point for comparing models over different domains might be the belief that isomorphic models are of identical power. That is, models computing the same set of functions, up to a different naming of their domain elements, are—for all intents and purposes—deemed equipotent. But can we be sure that TM is not isomorphic to a strongly-hypercomputational model? Surprisingly, perhaps, it turns out that a computational model can be isomorphic to one computing more functions. Fortunately, we can show that TMs are an exception and are not susceptible to this anomaly. So, as long as the Nuri model is countable, we know how to proceed with our investigation.

Isomorphism may be a good starting point, but it is not general enough, if only because Nuri's domain of operations may be of a larger cardinality than our countable, earth-bound devices. The common approach, in this case, would be to require an injection between the domains, up to which the stronger model mimics the functionality of the weaker one. We will claim that such an approach is too permissive, allowing one to hide computational power in the mapping. A somewhat philosophical outlook on these questions will lead us to a formal method for comparing arbitrary computational models over arbitrary domains. With this method in hand, we will be ready to return to an analysis of E.V.'s Nuri.

2 Introduction

Our goal is to formalize comparisons of computational models, that is, the determination when one set of partial functions is computationally more powerful than another set. We seek a robust definition of relative power, one that does not depend itself on any notion of computability. It should allow one to

compare arbitrary models over arbitrary domains in some quasi-ordering that captures the intuitive concept of computational strength. Such a comparison notion (or notions) should also allow one to prove statements like "analogue machines are strictly more powerful than digital devices," even though the two models operate over domains of different cardinalities.

With a satisfactory comparison notion in place, we look into mathematical relations between computational models, and properties they confer on models. We call a model that is not as powerful as any of its proper expansions "complete." We investigate completeness, and check whether some classical models enjoy this property.

Extensionality. We are only interested in the computational aspect of computational models (extensionality), that is, which problems can be solved by a model, regardless of the solution's complexity or the model's mechanisms. Hence, a computational model is represented simply by a set of (partial) functions (or multivalued functions) over the domain of its operation.

The Problem. Though model comparison is a common practice in the literature, it is usually done without a formal comparison notion and without justification for the chosen method. To the best of our knowledge, there is currently no satisfactory general means for comparing arbitrary computational models operating over arbitrary domains. A notion is lacking via which one could show, for example, that analogue computers are strictly more powerful than Turing machines, as well as show that finite automata are more powerful than some weak analogue model. In Section 5, we list some of the familiar comparison methods and discuss their ramifications.

The Framework. In Section 3, we propose a general, philosophical, definition of a computational model and of relative computational power. We understand a computational model to be a mathematical modeling and idealization of some hypothetical physical device, from a specific point of view of the world. A model B is at least as powerful as A if it has the potential to do whatever A does, under any possible view of the world. Accordingly, we provide, in Section 4, a method (Definition 3) for comparing arbitrary models over arbitrary domains.

Completeness. In Section 6, we show that the method usually used in the literature for "more powerful" ($\succsim$) is mathematically problematic, as it allows for a model to be more powerful than itself ($A \succsim A$). We define a model that

is not as powerful as any of its proper expansions to be *complete*. The standard method of comparison is suitable only for such complete models. On the positive side, we prove in Section 7.1 that Turing machines and the recursive functions are complete with respect to the desired comparison notions.

Computability. In Section 7, we show that some of the models known to be of equivalent power to Turing machines (the recursive functions, random access machines and counter machines) are indeed so by our suggested general notion.

Strong Hypercomputation. In Section 7.1, we prove that Turing machines and the recursive functions are complete models. Accordingly, we provide a simpler comparison notion for showing that a model is strongly hypercomputational. This notion provides a justification for the (otherwise improper) comparison method used in the literature for showing that a model is strongly hypercomputational.

Note. We use the Z-standard [4] for function arrows. For example, $\pfun$ denotes a partial function, $\surj$ is used for a total surjective function, and $\inj$ is an injection. We use double-arrows for mappings (multi-valued functions). So $\rightrightarrows$ denotes a total surjective mapping.

3 Conceptual Framework

We first propose a general, philosophical, definition of a computational model, and—in Section 3.2—of relative computational power. In Section 4, we will formalize these definitions for comparing arbitrary models over arbitrary domains.

3.1 What is a Computational Model?

We can think of a computational model as a mathematical modeling and idealization of some hypothetical physical device, from a specific point of view of the world (see Fig. 1).

- A physical device gets a physical input and returns a physical output. For example, an electric device may take some electric voltage at two of its pins as input, and return a voltage at two other pins as output.

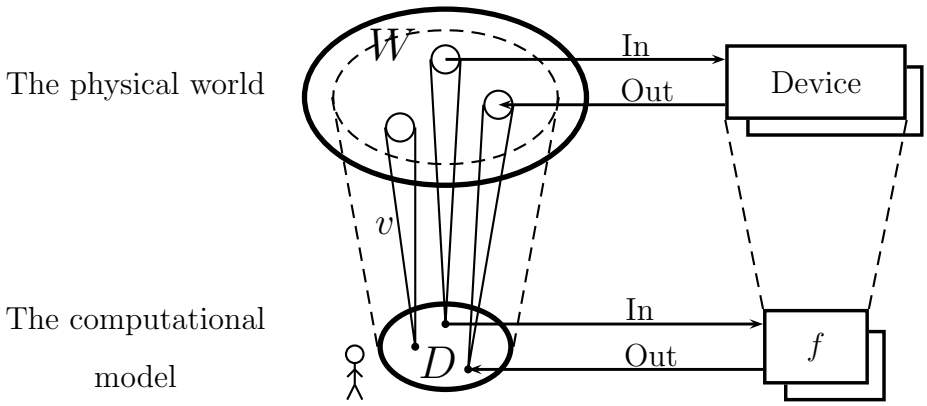

Fig. 1. A computational model is a mathematical modeling of some hypothetical physical devices, from a specific point of view of the world

---

- A corresponding computational model takes a specific point of view of the physical world. For example, a model of a digital computer might view a voltage lower than 0.5v as the binary value $\mathtt{0}$ and of 0.5v or higher as $\mathtt{1}$. That is, the domain of the model, D, is a "view" of the physical world, W. This view is a partial surjective function $v : W \twoheadrightarrow D$.
- The device computes a function on world entities (in our example above, $\xi : \mathbf{R} \to \mathbf{R}$), while from the model's point of view it computes a function on its domain (in our example, $f : \{\mathtt{0}, \mathtt{1}\} \to \{\mathtt{0}, \mathtt{1}\}$).

A computational model, by itself, can be viewed as a "black box," computing a set of partial functions. The domain and range of functions are identical, except that the range is extended with $\perp$, representing "undefined."

The modeling of a hypothetical device from a specific point of view of the world will be at the heart of our method of comparing different models. The world can be chosen to be any set of cardinality at least as large as the cardinality of the model's domain.

The idea that a model encapsulates a point of view of the world is shared by Minsky [13]:

> We use the term "model" in the following sense: To an observer B, an object A* is a model of an object A to the extent that B can use A* to answer questions that interest him about A. The model relation is inherently ternary. ... It is understood that B's use of a model entails the use of encodings for input and output, both for A and for A*. If A is the world, questions for A are experiments.

Different Domain and Range. There are models with different domain and range, e.g. numeral input and boolean output. A generalized view is to consider the "actual" model's domain to be the union of the original domain

and range.

Uniform Computation. It is common to have models with functions of any fixed arity, like the recursive functions, for example. We consider the "actual" domain (and range) to be the set of all finite tuples of elements of the original domain. This is the view taken for Turing machines, in the BSS model [1, pp. 69–70], and implicitly with recursive functions when comparing them to Turing machines.

Computing over Structures. There are models defined over structures, that is, over sets together with "built-in" functions and relations. See, for example, [3,18,1]. We consider the structure's set as the domain, and include the structure's functions and relations in the model.

3.2 Comparing Computational Power

We generally say that a model B is at least as powerful as A, written $B \succsim A$, if it can do whatever A does. When both models have the same domain representation, it means "containment": B is at least as powerful as A if it computes all the functions that A does. The question is how one should compare models operating over different domains, as they compute formally-different functions.

We extend the above characterization as follows: B is at least as powerful as A if it has the potential to do whatever A does for every possible user (an abstract user, not necessarily human). In other words, for every view that A has of the world ($v : W \twoheadrightarrow \mathrm{dom}\ A$), there is a view by B of the world ($u : W \twoheadrightarrow \mathrm{dom}\ B$), such that B has the abstraction capabilities of A, and all the functionality of A from A's point of view (see Fig. 2, Definition 2, and Definition 3).

Assumption. We want to allow the world-domain W to be as big as required, as well as the resolution of its elements to be enlarged as much as required. That is, all elements $x \in W$ may be considered as sets of the same cardinality.

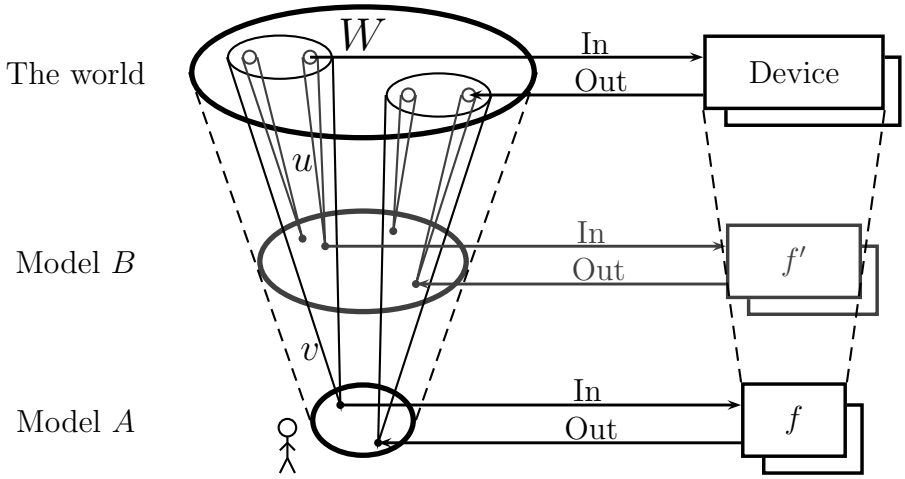

Fig. 2. The "stronger" model, B, should have the potential to provide all the functionality of the "weaker" model, A, from any user point of view

4 The Formal Comparison Notion

We need to formalize the conceptual framework of the previous section.

Definition 1 (Computational Model)

- A domain *is a nonempty set of elements.*
- A computational model A *over domain* D *is an abstraction of an object that computes a set of partial functions* $f : D \rightarrowtail D$*, which may be interpreted as total functions* $f : D \rightarrow D \cup \{\perp\}$*.*
- *We write* dom A *for the domain over which model* A *operates.*
- *The* extensionality *of a model* A*, denoted* ext A*, is the set of partial functions that* A *computes.*
- *For models* A *and* B*, and a function* f *we shall write* $f \in A$ *as shorthand for* $f \in$ ext A*, and* $A \subseteq B$ *as short for* ext $A \subseteq$ ext B*.*
- *We say that a model* B properly expands *model* A *if* $B \supsetneq A$*.*

Some clarifications regarding function notations:

- Two partial functions, f and g, over the same domain are (extensionally) *equal*, denoted $f = g$, if they are defined for exactly the same elements of the domain ($f(x) = \perp$ iff $g(x) = \perp$) and have the same value whenever they are both defined ($f(x) = g(x)$ if $f(x) \neq \perp$).
- A function $f : D \rightarrowtail D'$ is defined over the subsets of D, $f : \mathscr{P}(D) \rightarrowtail \mathscr{P}(D')$, by $f(X) := \{f(x) : x \in X\}$.
- A *mapping* $\rho : D \rightrightarrows D'$ is a binary relation between D and D', that is, a subset of $D \times D'$. Its inverse ρ^{-1} is defined as usual as $\{\langle y, x\rangle : \langle x, y\rangle \in \rho\}$. Any mapping may also be viewed as a total function $\rho : D \rightarrow \mathscr{P}(D')$, from D to subsets of D', in the sense that $\rho : x \mapsto \{y : \langle x, y\rangle \in \rho\}$. Thus, its

inverse, $\rho^{-1} : D' \rightrightarrows D$, is the function $\rho^{-1} : D' \to \mathscr{P}(D)$, from D' to the subsets of D, such that $\rho^{-1} : y \mapsto \{x : y \in \rho(x)\}$.
- A *total surjective mapping* $\rho : D \twoheadrightarrow D'$ is a total function, $\rho : D \to \mathscr{P}(D')$, from D to the subsets of D', such that $\bigcup_{x \in D} \rho(x) = D'$, and $\rho(x) \neq \emptyset$ for all $x \in D$.

We formalize the conceptual characterization of "as powerful" (see Fig. 2), by demanding the "stronger" model to have all the functionality of the "weaker" model up to different views of the world. The "stronger" model should also have an "abstraction" function, which ensures that when it has a more detailed view of the world, it may also gather various points into a single one, obtaining the abstraction capabilities of the "weaker" model.

Definition 2 (Conceptual Power Comparison) *Model B is* at least as powerful as *model A if for every domain W (the world) and view $v : W \twoheadrightarrow \mathrm{dom}\ A$, there are a view $u : W \twoheadrightarrow \mathrm{dom}\ B$ and abstraction function $g \in B$, s.t.*

(a) for every function $f \in A$ there is a function $f' \in B$, s.t. $v \circ u^{-1} \circ f' \circ u \circ v^{-1}(x) = \{f(x)\}$ for all $x \in \mathrm{dom}\ A$,
(b) $g(z) = g(y)$ iff $v \circ u^{-1}(z) = v \circ u^{-1}(y)$ for all $y, z \in \mathrm{dom}\ B$, and
(c) $v \circ u^{-1} \circ g(y) = v \circ u^{-1}(y)$ for all $y \in \mathrm{dom}\ B$.

Here "view" is a partial surjective function, and (by the conceptual assumption) all elements $x \in W$ may be considered as sets of a fixed cardinality. The first condition, (a), says that B computes every function of A, up to the mapping between the domains ($v \circ u^{-1}$). Condition (b) says that the abstraction function $g \in B$ distinguishes between the equivalence classes generated by the mapping, while (c) says that the distinction is made by choosing a representative element within each class.

Definition 2 may be simplified, omitting the world-domain. The two different views of the world are replaced by a "correlation" mapping between the model domains.

Definition 3 (Power Comparison Notion)

(1) Model B is (computationally) at least as powerful *as model A, denoted $B \succsim A$, if there are a total surjective mapping $\rho : \mathrm{dom}\ B \twoheadrightarrow \mathrm{dom}\ A$ (*correlation mapping*) and function $g \in B$ (*abstraction function*), such that:*
  (a) for every function $f \in A$ there is a function $f' \in B$ such that $\rho \circ f' \circ \rho^{-1}(x) = \{f(x)\}$ for all $x \in \mathrm{dom}\ A$,
  (b) $g(z) = g(y)$ iff $\rho(z) = \rho(y)$ for all $y, z \in \mathrm{dom}\ B$, and
  (c) $\rho \circ g(y) = \rho(y)$ for all $y \in \mathrm{dom}\ B$.

(2) Model B is (computationally) more powerful *than A, denoted $B \succnsim A$, if*

$B \succsim A$ *but* $A \not\succsim B$.

(3) *Models* A *and* B *are* (computationally) equivalent *if* $A \succsim B \succsim A$, *in which case we write* $A \approx B$.

Proposition 1 *The computational power relation* $\succsim$ *between models is a quasi-order. Computational equivalence* $\approx$ *is an equivalence relation.*

Theorem 1 *Definitions 2 and 3.1 are equivalent. That is* $B \succsim A$ *by Definition 3.1 iff* B *is at least as powerful as* A *by Definition 2.*

Proof. Let $\succsim_{\mathrm{C}}$ stand for the conceptual definition (Definition 2).

(1) $B \succsim_{\mathrm{C}} A$ implies $B \succsim A$: Since the world W is assumed to be as big as required, it follows that there exist a view v of A by a total surjective function $v : W \twoheadrightarrow \mathrm{dom}\ A$. Hence, the corresponding view of B is also a total surjective function $u : W \twoheadrightarrow \mathrm{dom}\ B$. Define a total surjective mapping $\rho : \mathrm{dom}\ B \rightrightarrows \mathrm{dom}\ A$, by $\rho := v \circ u^{-1}$. We have that $B \succsim A$ via ρ.

(2) $B \succsim A$ implies $B \succsim_{\mathrm{C}} A$: Let $B \succsim A$ via a total surjective mapping $\rho : \mathrm{dom}\ B \rightrightarrows \mathrm{dom}\ A$. We construct the proof in three steps:

(a) *Specific world and view.* Define a domain W (a world) as a subset of $\mathrm{dom}\ A \times \mathrm{dom}\ B$, by $W := \{\langle a, b\rangle : a \in \rho(b)\}$. Define total surjective functions $v : W \twoheadrightarrow \mathrm{dom}\ A$ and $u : W \twoheadrightarrow \mathrm{dom}\ B$ (the views) by $v(\langle a, b\rangle) := a$ and $u(\langle a, b\rangle) := b$, for all $\langle a, b\rangle \in W$. We have a specific (total) view of A for which the condition of the conceptual definition is satisfied, that is for every function $f \in A$ there is $f' \in B$, s.t. $v \circ u^{-1} \circ f' \circ u \circ v^{-1}(x) = \{f(x)\}$ for all $x \in \mathrm{dom}\ A$.

(b) *Specific world and all views.* Let $m : W \twoheadrightarrow \mathrm{dom}\ A$ be an arbitrary view of A of the world W. By the conceptual assumption, we may consider the domain W as a domain W', by replacing each element $y \in W$ with a set Y of cardinality $|W|$. Accordingly, the view m becomes a view $m' : W' \twoheadrightarrow \mathrm{dom}\ A$, where $|m'^{-1}(x)| = |W|$ for all $x \in \mathrm{dom}\ A$. Define a partial surjective function $\xi : W' \twoheadrightarrow W$, such that $m'(z) = v \circ \xi(z)$ for all $z \in W'$. Define the view of B of the world, as the partial surjective function $u' : W' \twoheadrightarrow \mathrm{dom}\ B$, by $u'(z) := u \circ \xi(z)$ for all $z \in W'$. We have that for every function $f \in A$ there is $f' \in B$, s.t. $v' \circ u'^{-1} \circ f' \circ u' \circ v'^{-1}(x) = \{f(x)\}$ for all $x \in \mathrm{dom}\ A$.

(c) *All worlds and all views.* Let $\widetilde{W}$ be an arbitrary world and $\widetilde{v} : \widetilde{W} \twoheadrightarrow \mathrm{dom}\ A$ an arbitrary view of A of the world $\widetilde{W}$. By the conceptual assumption we can enlarge the world $|\widetilde{W}|$ as required, thus we may assume that $|\widetilde{W}| \geq |W|$. Define a total surjective function $\tau : \widetilde{W} \twoheadrightarrow W$ s.t. $\tau(x) = \tau(y)$ implies that $\widetilde{v}(x) = \widetilde{v}(y)$ for every $x, y \in \widetilde{W}$. Define a view $v : W \twoheadrightarrow A$ by $v := \widetilde{v} \circ \tau^{-1}$. By the previous step there is a corresponding view $u : W \twoheadrightarrow B$, hence we have the required view

$\widetilde{u} : \widetilde{W} \twoheadrightarrow B$ by $\widetilde{u} := u \circ \tau$.

□

Example 1 *Consider a modeling of a simple electric-cable by a model EC, providing only the identity function over the reals. Then* TM $\not\succsim$ *EC and EC* $\not\succsim$ TM*.*

Inclusion of the Identity Function. When the "weak" model includes the identity function ($\lambda x.x$), the general comparison notion may be simplified, requiring the correlation mapping (ρ) to be a surjective function instead of a surjective mapping. If the "stronger" model is closed under functional composition, it may be further simplified, replacing the surjective function with an opposite injection (ψ : dom $A \rightarrowtail$ dom B). This is similar to the embedding notion (Definition 4 below) with the additional requirement for an abstraction function (g). Comparison via a surjective function resembles the "representation" of [20, p. 33], just that here we insist on a total function.

Lemma 1 *Let A be a computational model with the identity function ($\lambda x.x \in A$), then a model $B \succsim A$ iff there exist a total surjective function $\varphi :$ dom $B \twoheadrightarrow$ dom A and a function $g \in B$, such that:*

(1) for every function $f \in A$ there is a function $f' \in B$ such that $\varphi \circ f'(y) = f \circ \varphi(y)$ for all $y \in$ dom B,
(2) $\varphi \circ g(y) = \varphi(y)$ for all $y \in$ dom B, and
(3) $g(z) = g(y)$ iff $\varphi(z) = \varphi(y)$ for all $y, z \in$ dom B.

Proof. The first direction is obvious, as a function is also a mapping. For the other direction, let A be a computational model with the identity function ($\iota := \lambda x.x \in A$), and let model $B \succsim A$ via a total surjective mapping $\rho :$ dom $B \twoheadrightarrow$ dom A. Assume, by contradiction, that ρ is not a function, hence elements $e \neq t \in$ dom A and $z \in$ dom B, such that $\rho^{-1}(e) = \rho^{-1}(t) = z$. Since $B \succsim A$, it follows that there is a function $\iota' \in B$, such that $\rho \circ \iota' \circ \rho^{-1}(x) = \{\iota(x)\}$ for all $x \in$ dom A. Therefore, $\rho \circ \iota'(z) = \{e\} = \{d\}$. Contradiction. □

Theorem 2 *Let A be a computational model with the identity function ($\lambda x.x \in A$). Then a model B, closed under functional composition, is at least as powerful as A ($B \succsim A$) iff there exist an injection $\psi :$ dom $A \rightarrowtail$ dom B and a total function $g \in B$ onto* rng ψ *($g :$ dom $B \twoheadrightarrow$ rng ψ), such that for every function $f \in A$ there is a function $f' \in B$ such that $\psi \circ f(x) = f' \circ \psi(x)$ for all $x \in$ dom A.*

Proof. Let models A and B, injection ψ and function g as defined in the theorem. Define a surjective function $\varphi : \text{dom } B \twoheadrightarrow \text{dom } A$, by $\varphi := \psi^{-1} \circ g$. Since B is closed under functional composition, it obviously follows that the constraints of Lemma 1 are satisfied. Hence, $B \gtrsim A$.

For the other direction, if $B \gtrsim A$ there are function $\varphi : \text{dom } B \twoheadrightarrow \text{dom } A$ and $g \in B$ as defined in Lemma 1. By the constraints of Lemma 1 on g, we can define an injection $\psi : \text{dom } A \rightarrowtail \text{dom } B$ by $\psi(x) := \varphi^{-1}(x) \in \text{rng } g$. Hence, satisfying the required constraints on ψ and g. □

Example 2 *Real recursive functions* (Rrec) *[14], are more powerful than Turing machines* (TM)*. That is* Rrec $\gtrsim$ TM*. The comparison is done via the injection* $\psi : \mathbf{N} \rightarrowtail \mathbf{R}$*, where* $\psi(n) = n$ *[14, p. 18], and the floor function* $(\lambda x.\, \lfloor x \rfloor)$ *to provide the abstraction capabilities of* Rec *(the above function g) [14, p. 10].*

See also Theorem 5 for the power equivalence of Turing machines and other models.

5 Ramifications of Familiar Notions

Various methods have been used to compare the computational power of competing models.

Extended Domains. It is common to claim that a function is incorporated in any of its extensions. That is, a function $f : D \to D$ is incorporated in $f' : D' \to D'$ if $D \subseteq D'$ and $f = f' \restriction_D$. See, for example, [5, p. 654]: "Here we adopt the convention that a function on $\mathbf{N}$ is in an analog class $\mathcal{C}$ if some extension of it to $\mathbf{R}$ is, i.e. if there is some function $\widetilde{f} \in \mathcal{C}$ that matches f on inputs in $\mathbf{N}$."

By the conceptual framework, "B extends A" can be interpreted as "B having the potential to be at least as powerful as A for a user who has both domain views." For example, one can consider a user who views the world as real numbers, but can identify the natural numbers among them.

This approach is not appropriate as a general power comparison notion, since the extended model B doesn't necessarily have the abstraction capabilities of A. For example, a mathematician working with paper and pencil may consider various physical entities to "be" the symbol 'a' (e.g. *a*, a, **a**, a, `a`). A model that lacks the abstraction of the various 'a's, treating each of them totally differently, is not as powerful. Another example is a model that accurately

doubles a real number ($\lambda x.2x$). It cannot replace the doubling function on the natural numbers for a user who doesn't have the ability to see so accurately (as might be the case with human users). Consider, also, a model with real input/input, but it's a voltage or frequency or something that is really real. Imagine it has a chaotic function h that is exactly the TM-halting function for integer inputs, but is discontinuous. So that unless we give it the exact value as input, the result is meaningless. Would one say that it is hypercomputational in any meaningful sense?

Embedding. Extending the domain is a special case of embedding. A model B embeds A, if there is an injection from the domain of A to the domain of B, via which B has all the functionality of A *over the range of the injection.* Actually, embedding is exactly as extending the domain, up to isomorphism.

Definition 4 (Embedding) *A computational model B* embeds *a model A, denoted $B \succsim_{\mathrm{E}} A$, if there is an injection $\psi : \mathrm{dom}\, A \rightarrowtail \mathrm{dom}\, B$, s.t. for every function $f \in A$ there is $f' \in B$ such that $f' \circ \psi(x) = \psi \circ f(x)$ for all $x \in \mathrm{dom}\, A$.*

For example, Turing machines and the (untyped) λ-calculus were shown by Church [6], Kleene [12], and Turing [19] to embed the partial recursive functions via a unary representation of the natural numbers, and Church numerals, respectively.

The reasons for the inadequacy of embedding as a generic power comparison notion are analogous to that of domain-extending.

Example 3 *Let* RE *be the recursively enumerable predicates over* $\mathbf{N}$. RE *may embed an expansion with infinitely many non-r.e. partial predicates* $\{h_i\}$. *Let*

$$h(n) = \begin{cases} 0 & \textit{program } n \textit{ halts uniformly} \\ 1 & \textit{otherwise} \end{cases} \qquad h_i(n) = \begin{cases} 0 & n < i \vee h(n) = 0 \\ \bot & \textit{otherwise}\,. \end{cases}$$

We have that $\mathrm{RE} \succsim_{\mathrm{E}} \mathrm{RE} \cup \{h_i\}$, *by an injection* $\psi(n) = 2n + h(n)$, *as*

$$h_i'(n) = \begin{cases} 0 & \lfloor n/2 \rfloor < i \textit{ or } n \bmod 2 = 0 \\ \bot & \textit{otherwise} \end{cases} \qquad f' = \begin{cases} f(\lfloor n/2 \rfloor) & f \in \mathrm{RE} \\ h_i'(n) & f = h_i\,. \end{cases}$$

where $f' \in \mathrm{RE}$ *and* $f' = \psi \circ f \circ \psi^{-1}$ *for every* $f \in \mathrm{RE} \cup \{h_i\}$. *(Without loss of generality, we are supposing that* $\psi(0) = h(0) = 0$.*)*

Effective Encoding. A common approach for comparing models over different domains is to require some manner of effectiveness of the encoding; see [8, p. 21] and [9, p. 290], for example. There are basically two methods:

(1) One can demand informal effectiveness: "The coding is chosen so that it is itself given by an informal algorithm in the unrestricted sense" [16, p. 27].
(2) Or one can require encoding effectiveness via a specific model, say, Turing machines: "The Turing-machine characterization is especially convenient for this purpose. It requires only that the expressions of the wider classes be expressible as finite strings in a fixed finite alphabet of basic symbols" [16, p. 28].

By the conceptual framework, an "effective comparison" means that B is at least as powerful as A for a human user, assuming humans are capable of "effective" representations.

Effectivity is a useful notion; however, it is unsuitable as a general power comparison notion. The first, informal approach is too vague, while the second can add computational power when dealing with subrecursive models and is inappropriate when dealing with non-recursive models.

6 When is a Model More Powerful?

We demonstrate that the method commonly used in the literature for "strictly more powerful" ($\succnsim$) is mathematically improper, as it allows for a model to be more powerful than itself ($A \succnsim A$). We define "complete" models, for which the common method is appropriate.

In general, the *strict part* $\succnsim^*$ of a quasi-order $\succsim^*$ is $\succsim^* \cap \not\precsim^*$. That is, $B \succnsim^* A$ if $B \succsim^* A$ but not $A \succsim^* B$.

The Common Method. Intuitively, one would expect that a proper expansion of a model (additional functions) is also more powerful, that is, for $B \supsetneq A$ to imply $B \succnsim A$. For example, general recursion is considered more powerful than primitive recursion as it expands it (e.g. with the Ackermann function), and a model that computes more than Turing machines is considered more powerful (see, e.g., [17]). Hence, the common method of showing that a model B is more powerful than model A, for some comparison notion $\succsim^*$, is to show that $B \succsim^* C \supsetneq A$.

The Problem. Unfortunately, it turns out that a proper expansion of a model is not necessarily more powerful by the standard comparison methods. That is, $B \supsetneq A$ does not imply $B \succnsim^* A$, where $\succnsim^*$ may be embedding, our suggested notion, or "containment up to isomorphism" (Theorem 3).

Example 4 *Define the set R_2 of "even" recursive functions (*Rec*):*

$$R_2 = \left\{ \lambda n. \begin{cases} 2f(n/2) & n \text{ is even} \\ n & \text{otherwise} \end{cases} : f \in \text{Rec} \right\}$$

R_2 embeds all the recursive functions via the injection $\lambda n.2n$, though $R_2 \subsetneq$ Rec.

See also Example 3, for the embedding of non-r.e. predicates in RE.

Note that the common comparison method (see above) permits a model to be more powerful than itself! For example, one might say that "$R_2 \precsim_{\text{E}} R_2$," since $R_2 \succsim_{\text{E}} \text{Rec} \supsetneq R_2$.

Theorem 3 ([2]) *There are models isomorphic to proper expansions of themselves. That is, there is a set of functions M over a domain D, and a bijection $\pi : D \rightarrowtail\!\!\!\!\!\rightarrow D$, s.t. $\{\pi \circ f \circ \pi^{-1} : f \in M\} \supsetneq M$.*

The Solution. The general solution is to use the strict part of the quasi-order. For example, with embedding one should show that "B may embed A, while there is no injection via which A may embed B."

In addition, one can check whether a specific model is "complete" in the sense that it is not equivalent (with respect to the relevant notion) to any of its proper expansions. For complete models, the common (generally improper) method is suitable, saving the necessity of precluding all possible mappings. We show, in Section 7.1, completeness for Turing machines and the recursive functions.

Definition 5 (Complete Models) *Let $\succsim^*$ be a quasi-order (comparison notion). A computational model A is* complete*, with respect to $\succsim^*$, if $A \precsim^* B \supseteq A$ implies $A = B$ for all B.*

Proposition 2 *Let $\succsim^*$ be a quasi-order (comparison notion), and A a complete model w.r.t. $\succsim^*$. Then $B \precsim^* A$ iff there is a model C, such that $B \succsim^* C \supsetneq A$.*

Theorem 4 *Let A be a model with the identity function, closed under function composition, and complete w.r.t. embedding ($\succsim_{\text{E}}$), then A is complete w.r.t. power-comparison ($\succsim$).*

Proof. Follows directly from Theorem 2. □

Corollary 1 *Let A be a model with the identity function, closed under function composition, and complete w.r.t. embedding. Then a model B is more powerful than A iff B is at least as powerful as A and embeds some proper expansion C of A. That is, $B \succsim A$ iff there is a model C s.t. $B \succsim A \subsetneq C \precsim_E B$.*

7 Computability

Some computational models considered to be of equivalent power to Turing machines are still so according to our suggested comparison notion (Definition 3).

Theorem 5 *Turing machines* (TM)*, the recursive functions* (Rec)*, counter machines* (CM)*, and random access machines* (RAM) *are all of the same computational power. That is,* TM $\approx$ Rec $\approx$ CM $\approx$ RAM*.*

Proof. We base the proof on known results, cited from [10]. The comparison method of [10] is based on embedding, however, specifically with the above models it was done via a bijective function, therefore satisfying the power comparison notion of Definition 3. See [10, pp. 116–118; pp. 131–133; pp. 207–208]. □

7.1 Strong Hypercomputation

Since the term "hypercomputation" as two common meanings—both computing more than TM, and computing a (possibly single) incomputable function—we use the term "strong hypercomputation" to denote the first meaning (see Section 1).

In Section 6, we saw that a proper expansion of a computational model is not necessarily more powerful (by any of the common comparison methods). What does this mean for hypercomputation? Can it be that Turing machines are as powerful as a model that computes additional functions?

We prove that Turing machines and the recursive functions are complete models, thus are not susceptible to such an anomaly. Accordingly, we provide some means to show that a model is strongly hypercomputational.

Definition 6 (Strong Hypercomputation) *A model A is* strongly hypercomputational *if it is more powerful than Turing machines, that is, if $A \succsim$ TM.*

Theorem 6 *The recursive functions* (Rec) *and the partial recursive functions* (PR) *are complete w.r.t. embedding.*

Proof. Assume, on the contrary, that there is a set of functions $M \supsetneq \text{Rec}$, and an injection $\rho : \mathbf{N} \rightarrowtail \mathbf{N}$, such that $\text{Rec} \precsim_{\text{E}} M$ via ρ. Let $S \in M$ be the successor function. There is, by assumption, a function $S' \in \text{Rec}$ such that $S' \circ \rho = \rho \circ S$. Since $\rho(0)$ is some constant and $\rho(S(n)) = S'(\rho(n))$, we have that $\rho \in \text{Rec}$. Since ρ is a recursive injection, it follows that ρ^{-1} is partial recursive (PR). For every $f \in M$, there is an $f' \in \text{Rec}$, such that $f = \rho^{-1} \circ f' \circ \rho$; thus $f \in \text{PR}$. Actually, f is total, since $\text{rng}\,(f' \circ \rho) = \text{rng}\,(\rho \circ f) \subseteq \text{rng}\,\rho$. Therefore, $M = \text{Rec}$.

By the same token, the partial recursive functions (PR) are complete w.r.t. embedding. □

Theorem 7 *Turing machines* (TM) *are complete w.r.t. embedding.*

Proof. The completeness of Turing machines (TM) w.r.t. embedding follows directly from its equivalence to the recursive functions (Rec) via a bijective mapping (Theorem 5), and the completeness of Rec (Theorem 6). □

Corollary 2 *Model A is strongly hypercomputational if any one of the following conditions is satisfied:*

(1) $A \precsim \text{TM}$.
(2) $A \supsetneq \text{TM}$.
(3) There is a model C, such that $A \precsim C \supsetneq \text{TM}$.
(4) There is a model C, such that $A \precsim_{\text{E}} C \supsetneq \text{TM}$ *and also* $A \precsim \text{TM}$.

References

[1] L. Blum, F. Cucker, M. Shub, and S. Smale. *Complexity and Real Computation.* Springer-Verlag, New York, 1998.

[2] U. Boker and N. Dershowitz. Comparing computational power. *Logic Journal of the IGPL*, to appear. Available at: `http://www.cs.tau.ac.il/~udiboker/files/ComparingPower.pdf`.

[3] O. Bournez, F. Cucker, P. J. de Naurois, and J.-Y. Marion. Computability over an arbitrary structure. sequential and parallel polynomial time. *FoSSaCS*, pages 185–199, 2003.

[4] J. P. Bowen. Glossary of Z notation. *Information and Software Technology*, 37(5–6):333–334, May–June 1995. Available at: `http://staff.washington.edu/~jon/z/glossary.html`.

[5] M. L. Campagnolo, C. Moore, and J. F. Costa. Iteration, inequalities, and differentiability in analog computers. *Journal of Complexity*, 16:642–660, 2000.

[6] A. Church. An unsolvable problem of elementary number theory. *American Journal of Mathematics*, 58:345–363, 1936.

[7] B. J. Copeland. Hypercomputation. *Minds and Machines*, 12:461–502, 2002.

[8] E. Engeler. *Formal Languages: Automata and Structures*. Lectures in Advanced Mathematics. Markham Publishing Company, Chicago, IL, 1968.

[9] F. Hennie. *Introduction to Computability*. Addison-Wesley, Reading, MA, 1977.

[10] N. D. Jones. *Computability and Complexity from a Programming Perspective*. The MIT Press, Cambridge, Massachusetts, 1997.

[11] T. D. Kieu. Quantum algorithm for Hilbert's Tenth Problem. *International Journal of Theoretical Physics*, 42:1461–1478, 2003.

[12] S. C. Kleene. Lambda-definability and recursiveness. *Duke Mathematical Journal*, 2:340–353, 1936.

[13] M. L. Minsky. Matter, mind and models. *Proc. International Federation of Information Processing Congress*, 1:45–49, 1965. Available at `http://web.media.mit.edu/~minsky/papers/MatterMindModels.html`.

[14] J. Mycka and J. F. Costa. Real recursive functions and their hierarchy. *Journal of Complexity*, 2004. In print.

[15] T. Ord. Hypercomputation: Computing more than the Turing machine. *Technical report, University of Melbourne, Melbourne, Australia*, 2002.

[16] H. Rogers, Jr. *Theory of Recursive Functions and Effective Computability*. McGraw-Hill, New York, 1966.

[17] H. T. Siegelmann. *Neural Networks and Analog Computation: Beyond the Turing Limit*. Birkhäuser, Boston, 1998.

[18] J. V. Tucker and J. I. Zucker. Abstract versus concrete computation on metric partial algebras. *ACM Transactions on Computational Logic*, 5(4):611–668, 2004.

[19] A. M. Turing. On computable numbers, with an application to the 'Entscheidungsproblem'. *Proceedings of the London Mathematical Society*, 42:230–265, 1936–37.

[20] K. Weihrauch. *Computable Analysis — An introduction*. Springer-Verlag, Berlin, 2000.