Relative Constructibility via Restricted Abstract State Machines

Lau Chee Loong Desmond

December 28, 2024

Abstract

We restrict the computational power of Gurevich's abstract state machines to obtain two classes of machine models for generalised computation of sets, namely *restricted abstract state machines* (*RASMs*) and *restricted abstract state machines with parameters* (*RASMPs*). We derive from each class, a relative computability relation on sets, which generalises the Turing reducibility relation on reals. it is then proven that the relation computability relation derived from RASMPs is equivalent to relative constructibility.

Table of Contents

1 Introduction

1.1 Generalising Computation

Conventionally, across nearly all fields of study, a *computation* refers to a *terminating run* of some *computer*. In the abstract, modern equivalents of the term *computer* range from *algorithm* to *program*. It should thus be non-contentious to say that, any difficulty in formally defining a computation must come from formally defining a program.

So, what exactly is a program? To a software engineer, it might be a compilable piece of code written in any known programming language, e.g. Java. To a mathematician, it could be any Turing machine. Here, the class of compilable Java codes and the class of Turing machines are two classes defined by different *models of computation*, at arguably different levels of abstraction. In principle, a model of computation ought to

- include the definition of a run, as well as
- specify precise conditions for a run to be terminating.

Following the line of thinking inspired by our two examples, it seems convenient — even natural — to define a program as any object admitted by a fixed model of computation.

Models of computation studied by computer scientists and mathematicians are aplenty. They include common programming languages, along with the semantics (see e.g.[2] and [5]) — in various paradigms — associated with each language. They also include stalwarts from almost a century ago, three of which are

(CT1) Gödel's operation schema,
(CT2) Church's λ-calculus, and of course
(CT3) Turing machines.

One obvious commonality of said models of computation is, they admit only finitary programs and runs. It follows that these models forbid infinitary computations, which is understandable and justified, considering infinitary computations are intuitively unrealisable in the real world.

But what if someone takes mathematical and philosophical interest in defining and studying *infinitary computations*, real-world considerations be damned? Is there inherent technical difficulty in doing so? The answer is, "No, not really," as evidenced by a multitude of well-received attempts (as much as a subject this niched can be well-received) at generalising computation to the infinite. Such generalisations are appropriately termed *models of generalised computation*.

Since many decades ago, mathematical logicians have tried their hand at adapting models among (CT1) to (CT3) to allow computations with, and on, arbitrary sets.

From the perspective of recursion theory, where the structure of Turing degrees under Turing reducibility and the jump operator is the main object study, leading candidates include α-recursion and E-recursion. Both paradigms depend on models of computation that generalises (CT1) with schemata on arbitrary sets (see e.g. [1] and [3]). These schemata give rise to analogues of important results in classical recursion theory, on sets that can be much larger than reals.

From the perspective of "computer as a machine", we have Koepke's ordinal Turing machines (see [10]). Ordinal Turing machines are a generalisation of (CT3): they work like standard Turing machines, except their tapes are now ORD-length and they can each refer to finitely many ordinals as parameters.

In any case, the models of set computation introduced hitherto rely very much on the Church-Turing thesis for justification of properness. Such justification usually goes along the lines of,

> *if X nicely generalises a model among (CT1) to (CT3) and the Church-Turing thesis says that the object of generalisation is a model of computation, then X should also be a model of computation.*

This kind of arguments often lead to unintuitive definitions insofar as computation is concerned, sometimes to the extent of an apology being issued (e.g. [12]). Indeed, modern intuition of computation has evolved to become rather high-level, and the models which motivated the Church-Turing thesis have been relegated to play the roles of convenient mathematical foundations, instead of anything resembling practical mental models of computation.

Perhaps the most celebrated attempt at amalgamating and formalising useful mental models of computations and algorithms, comes in the form of a series of papers (starting with [6]) by Gurevich on abstract state machines. Abstract state machines far more closely resemble "high-level" meta-theoretic representations of Plotkin-style semantics (found e.g. in [4]), than they do anything among (CT1) to (CT3). That these machines have been adapted for use in fields ranging from software development to systems engineering is testament to the intuitiveness of their design. Having said that, how well they are equipped to handle the intuition of computations on infinite sets is yet to be vigorously tested, for literature in this respect is scarce.

1.2 Generalising Computability

A notion of computability is meant to chart the limits of what can and cannot be computed. In the classical case, we hold the Church-Turing thesis philosophically responsible for this purpose, a responsibility rooted in its convincing representation of computer-hood. Indeed, the hallowed status afforded to the Church-Turing thesis stems from the unlikely convergence in power of what seems to be independently-devised models of computation.

Certain models of computation (with simple modifications) naturally induce a notion of relative computability on a set bigger than the natural numbers. Such a relative computability relation typically extends the notion of computability, in that being computable is equivalent to being minimally relative computable. A most notable example is the Turing reducibility relation induced by oracle machines, which are essentially Turing machines equipped with quick access to a real number serving as an oracle.

It can be argued that a robust model of generalised computation should induce a (binary) relative computability relation on arbitrary sets, and that the relative computability relation thus induced should satisfy the following three desiderata.

(D1) *Universality*: the relation should be defined on all sets of ordinals.
(D2) *Transitivity*: the relation should be transitive.
(D3) *Coherence*: if the relative computability of two sets is witnessed under known computational constraints, then the same should be witnessed after lifting these constraints.

The astute reader might ask, "should universality in (D1) not require the inclusion of literally all sets in the domain of the relation?" The truth is, (D1) is phrased this way for convenience of analysis. Note that there is little substantive compromise in this apparently weaker presentation, because every set can be simply encoded into a binary string, according to Proposition 2.2. We shall assume that the designation of (D2) and (D3) as desiderata is intuitive, if not self-evident. Unsurprisingly, each of the well-known models of generalised computation introduced in the previous subsection induces relative computability relation on sets. Let us now briefly check (D1) to (D3) against these relations.

The relative computability relation associated with α-recursion, written $\leq_\alpha$, is only defined on $\mathcal{P}(\alpha)$, the powerset of α. As such, it is not universal. Naively, we can define a universal relation $\leq_A$ by referring to $\leq_\alpha$ as α ranges over all *admissible* ordinals:

$$A \leq_A B \text{ iff there is } \alpha \text{ for which } A \leq_\alpha B.$$

However, interpreting α to be a measure of the amount of computational resources available, as is natural, Theodore Slaman's answer in [15] tells us $\leq_A$ is doomed to be incoherent. It seems, then, that there is no easy fix to the non-universality of the $\leq_\alpha$'s.

On the other hand, the relative computability relation of E-recursion, written $\leq_E$, is both universal and transitive out of the box. Further, it is common in E-recursion to study "runs" restricted to arbitrary *E-recursively closed* initial segment of L. Restrictions of this type are regarded as computational constraints imposed on the evaluation of $\leq_E$. Unfortunately, with respect to these constraints, $\leq_E$ is not known to behave coherently (see e.g. "**The Divergence-Admissibility Split**" on p.11 of [12]).

Taking a leaf from how classical Turing machines were modified to become classical oracle machines, we can straightforwardly augment ordinal Turing machines with set

oracles. This allows one to define a relative computability relation $\preceq_A$ on arbitrary sets in the same way real oracles are used to define the Turing reducibility relation on the reals. In fact, with reference to $\preceq_\alpha$ in Definition 19 of [11],

$$A \preceq_A B \text{ iff there is } \alpha \text{ for which } A \preceq_\alpha B.$$

Such a definition is easily seen to be universal and coherent. It is stated in Theorem 20 of [11] that restricting this relative computability relation to subsets of admissible ordinals α may result in a lack of transitivity. We can however verify that $\preceq_\alpha$ is transitive whenever α is regular, and in turn, use it to conclude the transitivity of $\preceq_A$.

Table 1: The relative computable relations native to three major models of generalised computation.

Property \ Relation	$\leq_\alpha$	$\leq_E$	$\preceq_\alpha$
Universal?	✗	✓	✗
Transitive?	✓	✓	✗
Coherent?	✗	✗	✓
Appears in...	α-recursion	E-recursion	α-computability

Table 2: Two universal relative computability relations derived from separate classes of non-universal relations.

Property \ Relation	$\leq_A$	$\preceq_A$
Universal?	✓	✓
Transitive?	✓	✓
Coherent?	✗	✓
Derived from...	the $\leq_\alpha$'s	the $\preceq_\alpha$'s

Preliminary investigations into the limits of their computational power quickly uncover something philosophically unsatisfying, if not unsettling. This spurs us into modifying the definition of an abstract state machine to better represent set computation.

Our modifications result in a new type of machine (abbreviated to RASMP) that possesses more concrete guardrails than abstract state machines, without losing too much intuitiveness. Coupled with an input/output paradigm, RASMPs naturally induce a relative computability relation on all sets, which can then be used to define (generalised) computable sets. It turns out that the degrees born from this relation coincide with the degrees of constructibility. In addition, the relation satisfies transitivity when restricted to reasonably nice sets. The upshot of all this, is a model of set computation which avoids

the drawbacks of other models highlighted in the preceding paragraphs. It may not be the best model to do recursion theory in, nor does it claim to be the easiest model to analyse, but we hope more work can be done in the future to explore its limits.

2 Restricting Abstract State Machines

In this section, we progressively chip away at the very useful, but very inclusive, definition of a sequential abstract state machine, to arrive at a more restrictive machine model which also better captures our intuition of generalised computation.

2.1 Abstract State Machines

Gurevich introduced abstract state machines in [6] as general models of what we may conceptualise as an algorithm. He followed up with a number of collaborations and papers (e.g. [8]) on various subclasses of such machines. If we follow the convention of using "algorithm" and "computer program" interchangeably, then abstract state machines are natural candidates for capturing the general notion of computability. However, the claim that such machines are embodiments of algorithms seems to have undergone insufficient scrutiny. We shall start this section by refuting this claim through a set-theoretic argument.

Briefly, a *sequential abstract state machine* — or sequential ASM (henceforth just ASM) — $\mathfrak{M}$ comprises a class of states $S(\mathfrak{M})$, a class of initial states $I(\mathfrak{M})$, and a class transition function $\tau_{\mathfrak{M}} : S(\mathfrak{M}) \longrightarrow S(\mathfrak{M})$ such that

(A1) every $s \in S(\mathfrak{M})$ is a first-order structure with the same finite signature, $\sigma(\mathfrak{M})$,
(A2) $I(\mathfrak{M}) \subset S(\mathfrak{M})$, and
(A3) for all $s \in S(\mathfrak{M})$, $\tau_{\mathfrak{M}}(s)$ has the same base set as s, and
(A4) if $s_1, s_2 \in S(\mathfrak{M})$ and s_1 and s_2 are isomorphic, then $\tau_{\mathfrak{M}}(s_1)$ and $\tau_{\mathfrak{M}}(s_2)$ are isomorphic.

A (finite) terminating run of $\mathfrak{M}$ is a sequence of states $(s_i : i \leq n < \omega)$ for which

(B1) $s_0 \in I(\mathfrak{M})$,
(B2) $\tau_{\mathfrak{M}}(s_n) = s_n$, and
(B3) for all $m < n$, $s_{m+1} = \tau_{\mathfrak{M}}(s_m) \neq s_m$.

For brevity's sake, some details are withheld here. First, Gurevich originally require states to have only function and constant symbols in their signatures. However, this restriction is purely cosmetic because every relation in a structure can be represented as a boolean function. Second, τ is supposed to satisfy the *bounded exploration postulate*, which vaguely means that τ must perform the same updates to all states with the same interpretations of ground terms in a particular finite set. We shall keep this requirement

(or at least its spirit) in mind for this subsection, even if we need not formally verify it. Finally, our definition above can be generalised so that $\tau_{\mathfrak{M}}$ need only be a binary relation – resulting in ASMs which might not be sequential – but that is unnecessary for the point we seek to put across.

Intuitively, a state of an ASM is a configuration of some generalised Turing machine. Each initial state corresponds to an initial configuration, from which one can read off the input (or one may take the initial state itself to be the input). The last state of a terminating run, sometimes called a *final state*, corresponds in spirit to an end configuration of a Turing machine run, from which one can read off the output. For example, if we have a specific symbol $\ulcorner X \urcorner$ in the signature to represent an analogue of the Turing machine tape, then a terminating run $\vec{r} = (s_1, \ldots, s_n)$ should give an algorithmic transformation of the input representation X^{s_1} into the output representation X^{s_n}.

We argue against ASMs as a good notion of generalised algorithms. Consider the example below.

Example 2.1. Let κ be an infinite cardinal. Expand the signature of $(\kappa; \in)$ to include a unary relation symbol $\ulcorner A \urcorner$ and a constant symbol $\ulcorner c \urcorner$. Now if A_1 and A_2 are subsets of κ, then we can define a sequential ASM $\mathfrak{M}(\kappa, A_1, A_2)$ as follows:

$$\begin{aligned} s_1 &:= (\kappa; \in, A_1, 0) \\ s_2 &:= (\kappa; \in, A_2, 1) \\ S(\mathfrak{M}(\kappa, A_1, A_2)) &:= \{s_1, s_2\} \\ I(\mathfrak{M}(\kappa, A_1, A_2)) &:= \{s_1\} \\ \tau_{\mathfrak{M}(\kappa, A_1, A_2)} &:= \{(s_1, s_2), (s_2, s_2)\}. \end{aligned}$$

Observe that A_1 and A_2 both interpret $\ulcorner A \urcorner$ on the base set κ, and that s_1 is not isomorphic to s_2 because

- s_1 interprets $\ulcorner c \urcorner$ as 0 and s_2 interprets $\ulcorner c \urcorner$ as 1, and
- there is no automorphism of $(\kappa; \in)$ other than the identity on κ.

This means $S(\mathfrak{M}(\kappa, A_1, A_2))$, $I(\mathfrak{M}(\kappa, A_1, A_2))$ and $\tau_{\mathfrak{M}(\kappa, A_1, A_2)}$ satisfy (A1) to (A4) in place of $S(\mathfrak{M})$, $I(\mathfrak{M})$ and $\tau_{\mathfrak{M}}$ respectively. Furthermore, c is specifically added to the signature so that the finite set $\{c\}$ witnesses $\mathfrak{M}(\kappa, A_1, A_2)$ satisfy the bounded exploration postulate.

Fix subsets A_1 and A_2 of κ. The only terminating run of $\mathfrak{M}(\kappa, A_1, A_2)$ is (s_1, s_2). Morally, this is saying that we can algorithmically get from A_1 to A_2. If we set A_1 to be a simply definable set like the empty set, then every subset of κ is algorithmically derivable from A_1. This seems incredible given the information content of arbitrary sets of ordinals, as exemplified by the next proposition.

Proposition 2.2. *Let M be a transitive model of* ZFC *and $X \in M$. Then there is a set of ordinals $c \in M$ such that if N is any transitive model of* ZF *containing c, then $X \in N$.*

Proof. Let Y' be the transitive closure of X (under the membership relation $\in$) and set $Y := Y' \cup \{X\}$. Then Y is $\in$-transitive. Choose a bijection f from a cardinal κ into Y. Use $\in'$ to denote the unique binary relation R on κ such that

$$R(\alpha, \beta) \iff f(\alpha) \in f(\beta).$$

Now apply Gödel's pairing function to code $\in'$ as a subset c of κ. To recover X from c, first apply the inverse of the pairing function followed by the Mostowski collapse to get Y. Then X is definable from Y as the unique $\in$-maximal element of Y. This decoding process is absolute for transitive models of ZF because all its components are. □

Proposition 2.2 allows us to code any set as a set of ordinals such that the decoding can be done in an absolute manner. If we believe that such a decoding scheme (particularly, Gödel's pairing function) is algorithmic, then the definition of ASMs entails that we can algorithmically derive every set in V from just $\emptyset$. Since most set theorists are of the opinion that sets in V can be extremely complicated and far from constructible in a very strong sense, this conclusion is wholly unsatisfactory, if not patently false. In other words, ASMs are *too* encompassing a paradigm for algorithms, and thus should not be used to draw the line for generalised computability.

Remark 2.3. In Example 2.1, we can set A_1 and A_2 to be subsets of $H(\kappa)$, and alter the definition of $\mathfrak{M}(\kappa, A_1, A_2)$ such that

$$s_1 := (H(\kappa); \in, A_1, 0)$$
$$s_2 := (H(\kappa); \in, A_2, 1).$$

In this way, we obtain a more literal argument that every set is algorithmically derivable from every other set, sidestepping the need for Proposition 2.2 as motivation. Nonetheless, Proposition 2.2 will be invoked in later parts of this section, so it is as good a time now as any to introduce it.

For much of the rest of this section, we will make educated modifications to the definition of an ASM, in an attempt to formulate a philosophically justified notion of algorithms.

2.2 Modifying the Satisfaction Relation

An obvious cause of the triviality identified in the preceding paragraph, is the lack of restriction on the transition function $\tau_{\mathfrak{M}}$ of an ASM $\mathfrak{M}$. Ideally, $\tau_{\mathfrak{M}}(s)$ should be describable is a way which only depends locally on itself and s. One formalisation of a "describable local property" applicable to first-order structures is the notion of a *first-order property*. A first-order formula ϕ over the signature of a structure $\mathfrak{A}$ is a first-order property of $\mathfrak{A}$ iff $\mathfrak{A} \models \phi$, where $\models$ is Tarski's satisfaction relation. However, in our use case, the formula ϕ might need to refer to two different structures, so some slight modifications to the satisfaction relation are in order.

Definition 2.4. Given an signature σ, define $2 \cdot \sigma$ to be the disjoint union of σ and itself. In other words, for each symbol $\ulcorner X \urcorner \in \sigma$ and each $i < 2$, we choose a fresh symbol $\ulcorner X_i \urcorner$ and cast it to be of the same type and arity as $\ulcorner X \urcorner$. We then set

$$2 \cdot \sigma := \{\ulcorner X_i \urcorner : \ulcorner X \urcorner \in \sigma \text{ and } i < 2\}.$$

Definition 2.5. A *binary first-order formula* over a signature σ is a first-order formula over $2 \cdot \sigma$.

Definition 2.6. Let $\mathfrak{A}_0$ and $\mathfrak{A}_1$ be first-order structures with the same base set A and signature σ. For each $a \in A$, choose a fresh constant symbol $\ulcorner a \urcorner$ not in $2 \cdot \sigma$. Denote

$$2 \cdot \sigma \cup \{\ulcorner a \urcorner : a \in A\}$$

as $\sigma(A, 2)$.

Definition 2.7. Let $\mathfrak{A}_0$ and $\mathfrak{A}_1$ be first-order structures with the same base set A and signature σ. Let $\ulcorner X \urcorner \in \sigma(A, 2)$. Define

- $X^{(\mathfrak{A}_0, \mathfrak{A}_1)} := a$ if $\ulcorner X \urcorner = \ulcorner a \urcorner$ for some $a \in A$,
- $X^{(\mathfrak{A}_0, \mathfrak{A}_1)} := Y^{\mathfrak{A}_0}$ if $\ulcorner X \urcorner = \ulcorner Y_0 \urcorner$ for some $\ulcorner Y \urcorner \in \sigma$, and
- $X^{(\mathfrak{A}_0, \mathfrak{A}_1)} := Y^{\mathfrak{A}_1}$ if $\ulcorner X \urcorner = \ulcorner Y_1 \urcorner$ for some $\ulcorner Y \urcorner \in \sigma$.

If $\ulcorner f \urcorner \in \sigma(A, 2)$ is a function symbol and t is a term over $\sigma(A, 2)$ of which interpretation under $(\mathfrak{A}_0, \mathfrak{A}_1)$ is known, then

$$f(t)^{(\mathfrak{A}_0, \mathfrak{A}_1)} := f^{(\mathfrak{A}_0, \mathfrak{A}_1)}(t^{(\mathfrak{A}_0, \mathfrak{A}_1)}).$$

Definition 2.8. Let $\mathfrak{A}_0$ and $\mathfrak{A}_1$ be first-order structures with the same base set A and signature σ. We define what $(\mathfrak{A}_0, \mathfrak{A}_1) \models_2 \phi$ means for $\phi \in \sigma(A, 2)$ by recursion on the complexity of ϕ.

- If $\phi = \ulcorner t_1 = t_2 \urcorner$ for terms t_1, t_2 over $\sigma(A, 2)$, then

$$(\mathfrak{A}_0, \mathfrak{A}_1) \models_2 \phi \text{ iff } t_1^{(\mathfrak{A}_0, \mathfrak{A}_1)} = t_2^{(\mathfrak{A}_0, \mathfrak{A}_1)}.$$

- If $\phi = \ulcorner R(t_1, \ldots, t_n) \urcorner$ for a relation symbol $\ulcorner R \urcorner \in \sigma(A, 2)$ and terms $t_1, \ldots, t_n$ over $\sigma(A, 2)$, then

$$(\mathfrak{A}_0, \mathfrak{A}_1) \models_2 \phi \text{ iff } R^{(\mathfrak{A}_0, \mathfrak{A}_1)}(t_1^{(\mathfrak{A}_0, \mathfrak{A}_1)}, \ldots, t_n^{(\mathfrak{A}_0, \mathfrak{A}_1)}).$$

- If $\phi = \ulcorner \neg \psi \urcorner$ for a formula ψ over $\sigma(A, 2)$, then

$$(\mathfrak{A}_0, \mathfrak{A}_1) \models_2 \phi \text{ iff } \neg((\mathfrak{A}_0, \mathfrak{A}_1) \models_2 \psi).$$

- If $\phi = \ulcorner \psi_1 \wedge \psi_2 \urcorner$ for formulas ψ_1 and ψ_2 over $\sigma(A, 2)$, then

$$(\mathfrak{A}_0, \mathfrak{A}_1) \models_2 \phi \text{ iff } ((\mathfrak{A}_0, \mathfrak{A}_1) \models_2 \psi_1 \wedge (\mathfrak{A}_0, \mathfrak{A}_1) \models_2 \psi_2).$$

- If $\phi = \ulcorner \exists x\ \psi \urcorner$ for a free variable $\ulcorner x \urcorner$ and a formula ψ over $\sigma(A, 2)$, then

$$(\mathfrak{A}_0, \mathfrak{A}_1) \models_2 \phi \text{ iff } \exists a \in A((\mathfrak{A}_0, \mathfrak{A}_1) \models_2 \psi[x \mapsto a]),$$

  where $\psi[x \mapsto a]$ is the result of replacing all instances of $\ulcorner x \urcorner$ in ψ by $\ulcorner a \urcorner$.

Before adding more requirements to the transition function of an ASM, let us enforce that every state in an ASM $\mathfrak{M}$ must have the same base set $b(\mathfrak{M})$. This can be done without loss of generality in the original definition of an ASM, because we can extend the base set of each state in $\mathfrak{M}$ to be the union of the base sets of states in $\mathfrak{M}$, while arbitrarily extending the interpretation of function and relation symbols in the signature by each state.

We stipulate that the transition function $\tau_{\mathfrak{M}}$ of an ASM $\mathfrak{M}$ must be finitely describable over $2 \cdot \sigma(\mathfrak{M})$. That is, there must exist a first-order sentence $\phi^{\tau}_{\mathfrak{M}}$ over $2 \cdot \sigma(\mathfrak{M})$ such that for all $s_1, s_2 \in S(\mathfrak{M})$,

$$\tau_{\mathfrak{M}}(s_1) = s_2 \iff (s_1, s_2) \models_2 \phi^{\tau}_{\mathfrak{M}}.$$

This new imposition on transition functions seems sufficiently severe to rule out the cases in Example 2.1. Or does it?

Example 2.9 (2.1 revisited). Let κ, A_1, A_2, s_1, s_2 and $\mathfrak{M}(\kappa, A_1, A_2)$ be as in Example 2.1. Set

$$\phi^{\tau}_{\mathfrak{M}(\kappa,A_1,A_2)} := \ulcorner \forall x\ (x \in_1 c_1 \iff (\forall y\ (\neg(y \in_1 x)))) \urcorner.$$

Then given any two expansions s_1' and s_2' of $(\kappa; \in)$ with signature σ containing $\ulcorner c \urcorner$,

$$(s_1', s_2') \models_2 \phi^{\tau}_{\mathfrak{M}(\kappa,A_1,A_2)} \text{ iff } s_2' \text{ interprets } \ulcorner c \urcorner \text{ to be } 1.$$

In particular, $(s_1, s_2) \models_2 \phi^{\tau}_{\mathfrak{M}(\kappa,A_1,A_2)}$ and $(s_2, s_2) \models_2 \phi^{\tau}_{\mathfrak{M}(\kappa,A_1,A_2)}$, so $\tau_{\mathfrak{M}(\kappa,A_1,A_2)}$ is still a valid transition function.

It turns out that we can intentionally choose the set of states in an ASM to trivialise the definition of its transition function. To overcome this, we make it so that the set of states depends only on the base set and signature of the structures involved. What this means is, if $\mathfrak{M}_1$ and $\mathfrak{M}_2$ are ASMs, $s_1 \in S(\mathfrak{M}_1)$, $s_2 \in S(\mathfrak{M}_2)$ and s_1, s_2 share the same base set and signature, then $S(\mathfrak{M}_1) = S(\mathfrak{M}_2)$. To wit, we include $b(\mathfrak{M})$ as a primary datum of an ASM, and derive $S(\mathfrak{M})$ from $b(\mathfrak{M})$ and $\sigma(\mathfrak{M})$ as follows:

$$S(\mathfrak{M}) := \{s : s \text{ is a first-order structure with base set } b(\mathfrak{M}) \text{ and signature } \sigma(\mathfrak{M})\}.$$

Now, there may be a nagging worry that we over-corrected, that our modified definition of an ASM is too restrictive. Indeed, this shortcoming becomes obvious when we try to define an ASM corresponding to a Turing machine.

Difficulty 2.10. Consider a Turing machine M with an ω-length tape and n states, for some $n < \omega$. An ASM $\mathfrak{M}$ representing M naturally has $b(\mathfrak{M})$ include ω and the set of M's states. A suitable transition function $\tau_{\mathfrak{M}}$ ought to speak of M's states, and to do that it is imperative that each of these states identify uniquely with a term over $\sigma(\mathfrak{M})$. For example, we can have a constant symbol $\ulcorner c_i \urcorner \in \sigma(\mathfrak{M})$ uniquely represent the i-th state of M.

However, as members of $S(\mathfrak{M})$ are allowed to freely interpret the $\ulcorner c_i \urcorner$s, there would be a problem of singling out the "next state" of $\mathfrak{M}$ based on an arbitrary "current state", using the relation $\models_2$ alone.

Difficulty 2.10 tells us $S(\mathfrak{M})$ should depend not just on $b(\mathfrak{M})$ and $\sigma(\mathfrak{M})$, but also on how symbols in $\sigma(\mathfrak{M})$ are to be interpreted in $b(\mathfrak{M})$. This leads us to the objects of focus in the next subsection.

2.3 Theories with Constraints in Interpretation

Theories with constraints in interpretation (henceforth, *TCI*s) were conceived in [14] as a convenient means of looking at generic objects produced by set-theoretic forcing. Their utility in that respect will be clearer in the next section. Nevertheless, we introduce them here so that they can be used to constrain the possible states of an ASM.

Definition 2.11. (Lau, [14]) A *first-order theory with constraints in interpretation* (*first-order TCI*) — henceforth, just *theory with constraints in interpretation* (*TCI*) — is a tuple $(T, \sigma, \dot{\mathcal{U}}, \vartheta)$, where

- T is a first order theory with signature σ,
- $\dot{\mathcal{U}}$ is a unary relation symbol not in σ,
- ϑ is a function (the *interpretation constraint map*) with domain $\sigma \cup \{\dot{\mathcal{U}}\}$,
- if $x \in ran(\vartheta)$, then there is y such that
  - either $x = (y, 0)$ or $x = (y, 1)$, and
  - if $\vartheta(\dot{\mathcal{U}}) = (z, a)$, then $y \subset z^n$ for some $n < \omega$, and
- if $\vartheta(\dot{\mathcal{U}}) = (z, a)$, then
  - $z \cap z^n = \emptyset$ whenever $1 < n < \omega$, and
  - $z^m \cap z^n = \emptyset$ whenever $1 < m < n < \omega$.

We call members of the interpretation constraint map *interpretation constraints*.

Definition 2.12. (Lau, [14]) Let $(T, \sigma, \dot{\mathcal{U}}, \vartheta)$ be a TCI. We say $\mathcal{M} := (U; \mathcal{I}) \models^* (T, \sigma, \dot{\mathcal{U}}, \vartheta)$ — or $\mathcal{M}$ *models* $(T, \sigma, \dot{\mathcal{U}}, \vartheta)$ — iff all of the following holds:

- $\mathcal{M}$ is a structure,
- σ is the signature of $\mathcal{M}$,

- $\mathcal{M} \models T$,
- if $\vartheta(\dot{\mathcal{U}}) = (y, 0)$, then $U \subset y$,
- if $\vartheta(\dot{\mathcal{U}}) = (y, 1)$, then $U = y$, and
- for $\dot{X} \in \sigma$,
  - if $\dot{X}$ is a constant symbol and $\vartheta(\dot{X}) = (y, z)$, then $\mathcal{I}(\dot{X}) \in y \cap U$,
  - if $\dot{X}$ is an n-ary relation symbol and $\vartheta(\dot{X}) = (y, 0)$, then $\mathcal{I}(\dot{X}) \subset y \cap U^n$,
  - if $\dot{X}$ is an n-ary relation symbol and $\vartheta(\dot{X}) = (y, 1)$, then $\mathcal{I}(\dot{X}) = y \cap U^n$,
  - if $\dot{X}$ is an n-ary function symbol and $\vartheta(\dot{X}) = (y, 0)$, then

$$\{z \in U^{n+1} : \mathcal{I}(\dot{X})(z \restriction_n) = z(n)\} \subset y \cap U^{n+1}, \text{ and}$$

  - if $\dot{X}$ is an n-ary function symbol and $\vartheta(\dot{X}) = (y, 1)$, then

$$\{z \in U^{n+1} : \mathcal{I}(\dot{X})(z \restriction_n) = z(n)\} = y \cap U^{n+1}.$$

We say $(T, \sigma, \dot{\mathcal{U}}, \vartheta)$ has a model if there exists $\mathcal{M}$ for which $\mathcal{M} \models^* (T, \sigma, \dot{\mathcal{U}}, \vartheta)$.

To specify an ASM $\mathfrak{M}$, we should also specify a TCI $\mathfrak{T}(\mathfrak{M}) = (T, \sigma, \dot{\mathcal{U}}, \vartheta)$ such that

$$T = \emptyset$$
$$\sigma = \sigma(\mathfrak{M})$$
$$\vartheta(\dot{\mathcal{U}}) = (b(\mathfrak{M}), 1).$$

We can then stipulate that

$$S(\mathfrak{M}) := \{s : s \models^* \mathfrak{T}(\mathfrak{M})\}.$$

Let us now check that any Turing machine can be easily represented as an ASM.

Example 2.13. As in Difficulty 2.10, let M be a Turing machine with an ω-length tape and n states, for some $0 < n < \omega$. Let the set of M's states be $\{c_i : i < n\}$, with c_0 being the only initial state and c_{n-1} the only final state.

Choose a binary relation symbol $\ulcorner \in \urcorner$, unary relation symbols $\dot{\mathcal{U}}$, $\ulcorner R \urcorner$, unary function symbols $\ulcorner \mathrm{Pre} \urcorner$, $\ulcorner \mathrm{Suc} \urcorner$, and constant symbols $\ulcorner h \urcorner$, $\ulcorner t \urcorner$, all of them distinct from one another. Set

$$b(\mathfrak{M}) := \omega$$
$$\sigma(\mathfrak{M}) := \{\ulcorner \in \urcorner, \ulcorner \mathrm{Pre} \urcorner, \ulcorner \mathrm{Suc} \urcorner, \ulcorner h \urcorner, \ulcorner t \urcorner, \ulcorner R \urcorner\}.$$

Define a function ϑ on $\sigma(\mathfrak{M}) \cup \{\dot{\mathcal{U}}\}$ as follows:

$$\begin{aligned}
\vartheta(\dot{\mathcal{U}}) &:= (\omega, 1)\\
\vartheta(\ulcorner\in\urcorner) &:= (\in \cap\, \omega, 1)\\
\vartheta(\ulcorner\mathrm{Pre}\urcorner) &:= (\mathrm{Pre}, 1)\\
\vartheta(\ulcorner\mathrm{Suc}\urcorner) &:= (\mathrm{Suc}, 1)\\
\vartheta(\ulcorner h\urcorner) &:= (\omega, 0)\\
\vartheta(\ulcorner t\urcorner) &:= (\omega, 0)\\
\vartheta(\ulcorner R\urcorner) &:= (\omega, 0),
\end{aligned}$$

where Pre and Suc are the predecessor and successor functions on ω, respectively. Set

$$\mathfrak{T}(\mathfrak{M}) := (\emptyset, \sigma(\mathfrak{M}), \dot{\mathcal{U}}, \vartheta).$$

Intuitively, we want $\ulcorner h\urcorner$ to interpret the current position of the read/write head of M, $\ulcorner t\urcorner$ to interpret the current state M is in, and $\ulcorner R\urcorner$ to interpret the current contents of M's tape. Notice that all members of ω are definable over the membership relation without parameters. This implies references to them can be made in sentences over $\sigma(\mathfrak{M})$. We will use $\langle n\rangle$ to refer to the $n < \omega$ in such sentences.

Define

$$\begin{aligned}
I(\mathfrak{M}) := \{s : s \models^* \mathfrak{T}(\mathfrak{M}) \text{ and}\\
s \models \ulcorner h = \langle 0\rangle \wedge t = \langle 0\rangle\urcorner \text{ and}\\
s \models \ulcorner \exists x\, (R(x)) \wedge \forall x\, \forall y\, (R(x) \wedge R(y) \implies x = y)\urcorner\}.
\end{aligned}$$

Each member of $I(\mathfrak{M})$ represents an initial configuration of M with

- the read/write head at the 0-th cell of the tape,
- the current state being the initial state,
- '1' written on exactly one cell of the tape, and
- '0' written on all other cells of the tape.

More specifically, '1' appears on the k-th cell of M's tape in an initial configuration iff M takes k as input.

We are almost done with specifying our ASM $\mathfrak{M}$; the only thing left to do is to give a transition function $\tau_{\mathfrak{M}}$ accurately representing transitions among the Turing configurations of M. It suffices to concoct a suitable sentence $\phi^{\tau}_{\mathfrak{M}}$ based on the transition function

$$\delta : \{c_i : i < n - 1\} \times 2 \longrightarrow \{c_i : i < n\} \times 2 \times \{\mathrm{left}, \mathrm{right}\}$$

of M. For each pair $(j,k) \in n \times 2$, let

$$\begin{aligned}
\phi_{j,k} &:= \ulcorner (q(R_0(h_0),k) \wedge t_0 = \langle j \rangle) \Longrightarrow (q(R_1(h_0),k') \wedge t_1 = \langle j' \rangle \wedge h_1 = \mathrm{Pre}(h_0) \wedge \varphi) \urcorner \\
&\quad \text{if } j < n-1 \text{ and } \delta((c_j,k)) = (c_{j'},k',\mathrm{left}), \\
\phi_{j,k} &:= \ulcorner (q(R_0(h_0),k) \wedge t_0 = \langle j \rangle) \Longrightarrow (q(R_1(h_0),k') \wedge t_1 = \langle j' \rangle \wedge h_1 = \mathrm{Suc}(h_0) \wedge \varphi) \urcorner \\
&\quad \text{if } j < n-1 \text{ and } \delta((c_j,k)) = (c_{j'},k',\mathrm{right}), \text{ and} \\
\phi_{j,k} &:= \ulcorner h_0 = h_1 \wedge t_0 = t_1 \wedge \forall x\ (R_0(x) \iff R_1(x)) \urcorner \\
&\quad \text{if } j = n-1,
\end{aligned}$$

where $q(\psi,k)$ is a shorthand for

$$\begin{aligned}
\neg\psi \text{ if } k = 0; \\
\psi \text{ if } k = 1
\end{aligned}$$

and φ is the sentence

$$\ulcorner \forall x\ (\neg x = h_0 \implies (R_0(x) \iff R_1(x))) \urcorner.$$

Finally,

$$\phi^{\tau}_{\mathfrak{M}} := \bigwedge \{\phi_{j,k} : (j,k) \in n \times 2\},$$

is a first-order sentence over $2 \cdot \sigma(\mathfrak{M})$ that gives us the representation we want.

2.4 Absoluteness and the Bounded Exploration Postulate

A good notion of algorithm ought to be absolute to a certain degree. In set-theoretic contexts, absoluteness for transitive models of set theory comes up the most often.

Definition 2.14. A sentence ϕ in the language of set theory with set parameters is *absolute for transitive models of* ZFC iff for all A and B such that

- $A \subset B$,
- $(A;\in)$ and $(B;\in)$ are transitive models of ZFC, and
- all parameters of ϕ are members of A,

$$(A;\in) \models \phi \iff (B;\in) \models \phi.$$

Definition 2.15. Let ϕ be a formula in the language of set theory, with set parameters and free variables $x_1,\ldots,x_n$. Suppose ϕ defines an n-ary relation R (in V). Then ϕ is an *absolute definition of* (or, ϕ *absolutely defines*) R *for transitive models of* ZFC iff for all A such that

- $(A;\in)$ is a transitive model of ZFC, and

- all parameters of ϕ are members of A,

$$\{(a_1, \ldots, a_n) \in A^n : (A; \in) \models \phi[x_1 \mapsto a_1, \ldots, x_n \mapsto a_n]\} = R \cap A.$$

Recall that it is implicit in our modified definition of an ASM $\mathfrak{M}$, that $\phi^{\tau}_{\mathfrak{M}}$ needs to have the following property:

$$\forall s_0 \in S(\mathfrak{M})\ \exists! s_1 \in S(\mathfrak{M})\ ((s_0, s_1) \models_2 \phi^{\tau}_{\mathfrak{M}}), \tag{2.1}$$

where

$$S(\mathfrak{M}) = \{s : s \models^* \mathfrak{T}(\mathfrak{M})\}.$$

However, given any $\mathfrak{T}(\mathfrak{M})$ and $\phi^{\tau}_{\mathfrak{M}}$, it is not immediate that (2.1) is absolute for transitive models of ZFC. As a result, it is difficult to ascertain that the definition of $\tau_{\mathfrak{M}}$ from $\mathfrak{T}(\mathfrak{M})$ and $\phi^{\tau}_{\mathfrak{M}}$ is absolute for transitive models of ZFC. This implores us to make the dependency between the current and the next state of an ASM even more local.

At this point, it is useful to make precise Gurevich's bounded exploration postulate, for it is an example of highly local dependency. Combined with our stipulation that a transition function must be definable, we expect to obtain a level of local definability from which the absoluteness we want can be easily derived. Consequently, abiding by the bounded exploration postulate should net us absoluteness at no additional cost.

Definition 2.16. Let $\mathfrak{M}$ be an ASM and $s \in S(\mathfrak{M})$. Define $\Delta_{\mathfrak{M}}(s)$ to be the unique function from $\sigma(\mathfrak{M})$ into $\mathcal{P}(b(\mathfrak{M}))$ such that

- if $\ulcorner X \urcorner \in \sigma(\mathfrak{M})$ is an n-ary function symbol, then

$$\Delta_{\mathfrak{M}}(s)(\ulcorner X \urcorner) = \{\vec{z} \in b(\mathfrak{M})^n : \ulcorner X \urcorner^s(\vec{z}) \neq \ulcorner X \urcorner^{\tau_{\mathfrak{M}}(s)}(\vec{z})\},$$

- if $\ulcorner X \urcorner \in \sigma(\mathfrak{M})$ is an n-ary relation symbol, then

$$\Delta_{\mathfrak{M}}(s)(\ulcorner X \urcorner) = \{\vec{z} \in b(\mathfrak{M})^n : \neg(\ulcorner X \urcorner^s(\vec{z})) \iff \ulcorner X \urcorner^{\tau_{\mathfrak{M}}(s)}(\vec{z})\},$$

- if $\ulcorner X \urcorner \in \sigma(\mathfrak{M})$ is a constant symbol, then

$$\Delta_{\mathfrak{M}}(s)(\ulcorner X \urcorner) = \begin{cases} \emptyset & \text{if } \ulcorner X \urcorner^s = \ulcorner X \urcorner^{\tau_{\mathfrak{M}}(s)} \\ b(\mathfrak{M}) & \text{otherwise.} \end{cases}$$

Note that if $b(\mathfrak{M}) = \emptyset$, then $S(\mathfrak{M})$ is a singleton and for every $s \in S(\mathfrak{M})$, $\tau_{\mathfrak{M}}(s) = s$. Necessarily, any such $\mathfrak{M}$ must have $\Delta_{\mathfrak{M}}(s_0) = \Delta_{\mathfrak{M}}(s_1)$ for all $s_0, s_1 \in S(\mathfrak{M})$.

Definition 2.17. Let $\mathfrak{M}$ be an ASM. Its transition function $\tau_{\mathfrak{M}}$ satisfies the *bounded exploration postulate* iff there is a finite set of ground terms $\mathcal{D}$ over $\sigma(\mathfrak{M})$ such that for all $\ulcorner X \urcorner \in \sigma(\mathfrak{M})$ and $s_0, s_1 \in S(\mathfrak{M})$,

$$\forall t \in \mathcal{D}\ (t^{s_0} = t^{s_1}) \implies \Delta_{\mathfrak{M}}(s_0) = \Delta_{\mathfrak{M}}(s_1).$$

As every relation is a boolean function, there is a canonical way of simulating relation symbols in a ground term. Thus from a computability point of view, the bounded exploration postulate seems intuitive enough at first glance. Upon closer inspection, however, the emphasis on changes in valuations gives rise to artefacts such as the one below.

Example 2.18. Consider ASMs $\mathfrak{M}_1$ and $\mathfrak{M}_2$ such that

$$\sigma(\mathfrak{M}_1) = \sigma(\mathfrak{M}_2) = \{\ulcorner \mathrm{In} \urcorner, \ulcorner \mathrm{Out} \urcorner\}$$
$$\mathfrak{T}(\mathfrak{M}_1) = \mathfrak{T}(\mathfrak{M}_2),$$

$b(\mathfrak{M}_1)$ is infinite, and

$$\phi^{\tau}_{\mathfrak{M}_1} = \ulcorner \forall x\ ((\mathrm{In}_1(x) \iff \neg \mathrm{In}_0(x)) \wedge (\mathrm{Out}_1(x) \iff \neg \mathrm{Out}_0(x)))\urcorner$$
$$\phi^{\tau}_{\mathfrak{M}_2} = \ulcorner \forall x\ ((\mathrm{In}_1(x) \iff x \neq x) \wedge (\mathrm{Out}_1(x) \iff \mathrm{Out}_0(x)))\urcorner.$$

Here, $\ulcorner \mathrm{In} \urcorner$ and $\ulcorner \mathrm{Out} \urcorner$ are supposed to represent input and output tapes respectively, with cells indexed by $b(\mathfrak{M}_1) = b(\mathfrak{M}_2)$ and entries from a binary alphabet. $\tau_{\mathfrak{M}_1}$ flips each bit of the input and output tapes, whereas $\tau_{\mathfrak{M}_2}$ resets the input tape to its all-zeroes state, an operation one can think of as a complete erasure of input tape data.

Observe that $\tau_{\mathfrak{M}_1}$ satisfies the bounded exploration postulate because for all $s_0, s_1 \in S(\mathfrak{M}_1)$,

$$\Delta_{\mathfrak{M}}(s_0) = \Delta_{\mathfrak{M}}(s_1) = \{(\ulcorner \mathrm{In} \urcorner, b(\mathfrak{M}_1)), (\ulcorner \mathrm{Out} \urcorner, b(\mathfrak{M}_1))\}.$$

On the other hand, $\tau_{\mathfrak{M}_2}$ does not satisfy the bounded exploration postulate because for $s_0, s_1 \in S(\mathfrak{M}_1)$, $\Delta_{\mathfrak{M}}(s_0) = \Delta_{\mathfrak{M}}(s_1)$ only if $\ulcorner \mathrm{In} \urcorner^{s_0} = \ulcorner \mathrm{In} \urcorner^{s_1}$, and whether $\ulcorner \mathrm{In} \urcorner^{s_0} = \ulcorner \mathrm{In} \urcorner^{s_1}$ cannot be determined by just examining a finite fragment of s_0 and a finite fragment of s_1. This means the transition function of any ASM built on $\tau_{\mathfrak{M}_2}$ by enlarging its signature while maintaining its function ("erase all data from the input tape and do nothing else"), cannot possibly satisfy the postulate.

We hence find ourselves in a weird position whereby universal bit-flipping — which requires some information of prior states — is an acceptably bounded, but universal erasure — which requires no prior knowledge — is not.

Besides the erasure of a tape's contents, one would expect an ASM $\mathfrak{M}_{\text{copy}}$ which copies the input tape's contents to the output tape, and does nothing else, to only carry out steps as valid as universal bit-flipping. Yet, by an argument similar to that for $\tau_{\mathfrak{M}_2}$ not satisfying the bounded exploration postulate in Example 2.18, $\mathfrak{M}_{\text{copy}}$ fails to satisfy the same postulate.

With the aforementioned flaws to 2.17 in mind, we seek to reformulate the bounded exploration postulate.

Definition 2.19. Let σ be a signature. We say a sentence ϕ over $2 \cdot \sigma$ is *bounded for* σ iff for each $\ulcorner X \urcorner \in \sigma$ there are $\phi^{\ulcorner X \urcorner}$ and $\psi^{\ulcorner X \urcorner}$ such that

- $\phi^{\ulcorner X \urcorner}$ is a formula over $\{\ulcorner Y_0 \urcorner : \ulcorner Y \urcorner \in \sigma\}$ $(\subset 2 \cdot \sigma)$,
- $\psi^{\ulcorner X \urcorner}$ is a sentence over $2 \cdot \sigma$,
- if $\ulcorner X \urcorner$ is an n-ary function symbol, then
  - $\phi^{\ulcorner X \urcorner}$ contains $n+1$ free variables, and
  - for some variable symbols $y_1, \ldots, y_{n+1}$,

    $$\psi^{\ulcorner X \urcorner} = \ulcorner \forall y_1 \ldots \forall y_{n+1} \, (X_1(y_1, \ldots, y_n) = y_{n+1} \iff \phi^{\ulcorner X \urcorner}(y_1, \ldots, y_{n+1}))\urcorner,$$

- if $\ulcorner X \urcorner$ is an n-ary relation symbol, then
  - $\phi^{\ulcorner X \urcorner}$ contains n free variables, and
  - for some variable symbols $y_1, \ldots, y_n$,

    $$\psi^{\ulcorner X \urcorner} = \ulcorner \forall y_1 \ldots \forall y_n \, (X_1(y_1, \ldots, y_n) \iff \phi^{\ulcorner X \urcorner}(y_1, \ldots, y_n))\urcorner,$$

- if $\ulcorner X \urcorner$ is a constant symbol, then
  - $\phi^{\ulcorner X \urcorner}$ contains 1 free variable, and
  - for some variable symbol y,

    $$\psi^{\ulcorner X \urcorner} = \ulcorner \forall y \, (X_1 = y \iff \phi^{\ulcorner X \urcorner}(y))\urcorner, \text{ and}$$

- $\phi = \bigwedge \{\psi^{\ulcorner X \urcorner} : \ulcorner X \urcorner \in \sigma\}$.

We call $\phi^{\ulcorner X \urcorner}$ the *witness for* $\ulcorner X \urcorner$ *to* ϕ *being bounded for* σ.

A few things are easily verifiable from Definition 2.19. Let σ, ϕ and the $\phi^{\ulcorner X \urcorner}$s be as in the definition. Suppose A is a set and $\mathfrak{A}$ is a structure with signature σ and base set A.

(C1) We can efficiently recover ϕ from the $\phi^{\ulcorner X \urcorner}$s, up to substitution of variable symbols.

(C2) If there exists a structure $\mathfrak{B}$ with signature σ and base set A, for which $(\mathfrak{A}, \mathfrak{B}) \models_2 \phi$, then such $\mathfrak{B}$ must be unique. Furthermore, $(\mathfrak{A}, \mathfrak{B}) \models_2 \phi$ is absolute for transitive models of ZFC containing $\mathfrak{A}$ and $\mathfrak{B}$.

(C3) If there is no structure $\mathfrak{B}$ with signature σ and base set A, for which $(\mathfrak{A}, \mathfrak{B}) \models_2 \phi$, then the cause of said non-existence is necessarily witnessed by a Δ_0-definable (i.e. definable by a first-order formula over $\{\in\}$ with only bounded quantifiers) fragment of $\mathfrak{A}$'s partial valuation map of one of the $\phi^{\ulcorner X \urcorner}$s, with parameters in $A^{<\omega}$. As Δ_0 definitions are absolute for transitive models of ZFC, the existence of a witness to the non-existence such $\mathfrak{B}$ must be too.

(C4) Points (C2) and (C3) guarantee that whenever $\mathfrak{M}$ is an ASM with $\sigma(\mathfrak{M}) = \sigma$, the statement

$$\forall s_0 \in S(\mathfrak{M}) \, \exists! s_1 \in S(\mathfrak{M}) \, ((s_0, s_1) \models_2 \phi)$$

is absolute for transitive models of ZFC containing $\mathfrak{T}(\mathfrak{M})$.

(C5) Suppose $\phi = \phi^{\tau}_{\mathfrak{M}}$ for some ASM $\mathfrak{M}$. Then in the context of $\mathfrak{M}$, for each $\ulcorner X \urcorner \in \sigma$, the value of the next state's interpretation of $\ulcorner X \urcorner$ at a point depends only on finitely many values of the current state's interpretations of symbols in σ. This agrees in spirit with the requirement of bounded exploration.

2.5 Input and Output

In Example 2.13, we are able to easily read the input of a Turing machine off its corresponding initial state in the ASM representation. Similarly, we expect to easily read the output off any of its final states. These are self-evident desiderata if we were to port the input-output paradigm of computability to ASMs.

To simplify things, let us require any input or output of an ASM $\mathfrak{M}$ to be a subset of $b(\mathfrak{M})$.

Definition 2.20. A *loading function* of an ASM $\mathfrak{M}$ is a function l from $\mathcal{P}(b(\mathfrak{M}))$ onto $I(\mathfrak{M})$, such that for some first-order formula ϕ with one free variable x over $\sigma(\mathfrak{M})$,

$$l(A) = s \text{ iff } \forall a \in b(\mathfrak{M})\ (a \in A \iff s \models \phi[x \mapsto a])$$

whenever $A \in \mathcal{P}(b(\mathfrak{M}))$ and $s \in S(\mathfrak{M})$. In this case, we say ϕ is a witness for l (as a loading function of $\mathfrak{M}$).

Definition 2.21. An *unloading function* of an ASM $\mathfrak{M}$ is a function u from $S(\mathfrak{M})$ into $\mathcal{P}(b(\mathfrak{M}))$, such that for some first-order formula ϕ with one free variable x over $\sigma(\mathfrak{M})$,

$$u(s) = A \text{ iff } \forall a \in b(\mathfrak{M})\ (a \in A \iff s \models \phi[x \mapsto a])$$

whenever $A \in \mathcal{P}(b(\mathfrak{M}))$ and $s \in S(\mathfrak{M})$. In this case, we say ϕ is a witness for u (as an unloading function of $\mathfrak{M}$).

The next two facts are easy to see.

Fact 2.22. Let $\mathfrak{M}$ be an ASM. If ϕ is a witness for l_1 and l_2 as loading functions of $\mathfrak{M}$, then $l_1 = l_2$.

Fact 2.23. Let $\mathfrak{M}$ be an ASM. If ϕ is a witness for u_1 and u_2 as unloading functions of $\mathfrak{M}$, then $u_1 = u_2$.

Let $\mathfrak{M}$ be an ASM, ϕ be a witness for a loading function of $\mathfrak{M}$, and $\mathfrak{T}(\mathfrak{M}) = (\emptyset, \sigma(\mathfrak{M}), \dot{\mathcal{U}}, \vartheta)$. Assume $|b(\mathfrak{M})| > 1$.

Replace

- each n-ary function symbol $\ulcorner f \urcorner \in \sigma(\mathfrak{M})$ with an $(n+1)$-ary relation symbol $\ulcorner R^f \urcorner$, and
- each constant symbol $\ulcorner c \urcorner \in \sigma(\mathfrak{M})$ with a unary relation symbol $\ulcorner R^c \urcorner$.

Make the same replacements in the definition of ϑ. Fix distinct members of $b(\mathfrak{M})$, x and y. Add fresh constant symbols $\ulcorner c \urcorner$, $\ulcorner 0 \urcorner$ and $\ulcorner 1 \urcorner$ and a fresh unary relation symbol $\ulcorner \mathrm{In} \urcorner$ to $\sigma(\mathfrak{M})$, and extend ϑ so that

$$\begin{aligned}
\vartheta(\ulcorner c \urcorner) &= (b(\mathfrak{M}), 0) \\
\vartheta(\ulcorner 0 \urcorner) &= (\{x\}, 1) \\
\vartheta(\ulcorner 1 \urcorner) &= (\{y\}, 1) \\
\vartheta(\ulcorner \mathrm{In} \urcorner) &= (b(\mathfrak{M}), 0).
\end{aligned}$$

Now, there is a standard way to speak of functions and constants as if they are relations satisfying either functionality or constancy, wherever relevant. In this way, every first-order sentence over the original $\sigma(\mathfrak{M})$ can be translated into a semantically equivalent sentence over the new $\sigma(\mathfrak{M})$, if we identify each non-relational $\ulcorner X \urcorner$ in the original $\sigma(\mathfrak{M})$ with $\ulcorner R^X \urcorner$. Let θ be such a translation of $\phi^\tau_{\mathfrak{M}}$. Let ψ denote a first-order sentence stating that

- "$\ulcorner R^f \urcorner$ is a function" for each function symbol $\ulcorner f \urcorner$ in the original $\sigma(\mathfrak{M})$, and
- "$\ulcorner R^c \urcorner$ is a constant" for each constant symbol $\ulcorner c \urcorner$ in the original $\sigma(\mathfrak{M})$.

Update $\phi^\tau_{\mathfrak{M}}$ as follows:

$$\begin{aligned}\phi^\tau_{\mathfrak{M}} := \ulcorner (c_0 = 0_0 \implies (\psi_1 \wedge c_1 = 1_0 \wedge \forall x\, (\mathrm{In}_0(x) \iff \phi_1(x) \iff \mathrm{In}_1(x))))\\ \wedge (\neg c_0 = 0_0 \implies (c_1 = c_0 \wedge \forall x\, (\mathrm{In}_0(x) \iff \mathrm{In}_1(x)) \wedge \theta))\urcorner,\end{aligned}$$

where ψ_1 and ϕ_1 are the results of replacing every symbol $\ulcorner z \urcorner \in \sigma(\mathfrak{M})$ occurring in ψ and ϕ respectively, with $\ulcorner z_1 \urcorner \in 2 \cdot \sigma(\mathfrak{M})$.

Next, define

$$\varphi := \bigwedge \{\ulcorner \forall x\, (\neg R(x))\urcorner : \ulcorner R \urcorner \neq \ulcorner \mathrm{In} \urcorner \text{ and } \ulcorner R \urcorner \text{ is a relation symbol in } \sigma(\mathfrak{M})\}.$$

Then for every $A \in \mathcal{P}(b(\mathfrak{M}))$ there is a unique $s \in S(\mathfrak{M})$ such that

$$\forall a\, (a \in b(\mathfrak{M}) \implies (a \in A \iff s \models \ulcorner \varphi \wedge c = 0 \wedge \mathrm{In}(x)\urcorner[x \mapsto a])).$$

In other words,

$$\phi_l := \ulcorner \varphi \wedge c = 0 \wedge \mathrm{In}(x) \urcorner$$

defines a function l from $\mathcal{P}(b(\mathfrak{M}))$ into $S(\mathfrak{M})$. Resetting $I(\mathfrak{M}) := ran(l)$, we have that ϕ_l is a witness for a loading function of the new $\mathfrak{M}$.

For the new $\mathfrak{M}$, the relation symbol $\ulcorner \mathrm{In} \urcorner$ morally interprets a generalised Turing tape of shape $(b(\mathfrak{M}); \sigma(\mathfrak{M}) \setminus \{\ulcorner \mathrm{In} \urcorner\}$. Each initial state has the input written on the tape, while the other components of the machine are set to default values. As such, variability of initial states occurs only on the tape, as is the case for any Turing machine.

Identifying each non-relational $\ulcorner X \urcorner$ in the original $\sigma(\mathfrak{M})$ with $\ulcorner R^X \urcorner$, we can conclude that for every $A \subset b(\mathfrak{M})$,

- the original $\mathfrak{M}$ terminates on input A iff the new $\mathfrak{M}$ terminates on input A, and
- if the original and the new $\mathfrak{M}$ terminate on input A with runs $(s_i : i < m)$ and $(s'_i : i < n)$ respectively, then $n = m + 1$ and for each $i < m$, s_i is a reduct of s'_{i+1}.

This means that our modifications to $\mathfrak{M}$ do not change its essence as an algorithm. There is thus no loss in computability resulting from a Turing-machine-like standardisation

of loading functions. A similar (in fact, simpler) argument can be used to justify a Turing-machine-like standardisation of unloading functions.

In accordance to the discussion above, we want an ASM $\mathfrak{M}$ to have a signature $\sigma(\mathfrak{M})$ containing two distinguished unary relation symbols $\ulcorner \mathrm{In} \urcorner$ and $\ulcorner \mathrm{Out} \urcorner$, representing the input and output tapes respectively. Moreover, we want the specification of $\mathfrak{M}$ to include a first-order sentence $\phi^d_{\mathfrak{M}}$ over $\sigma(\mathfrak{M}) \setminus \{\ulcorner \mathrm{In} \urcorner, \ulcorner \mathrm{Out} \urcorner\}$ that describes the default "non-tape" configuration of $\mathfrak{M}$. As in the case of transition functions, we want the loading function described by $\phi^d_{\mathfrak{M}}$ to be absolute for transitive models of ZFC. To do so, we want to first impose a greater amount of structure on the states of an ASM.

Within the context of Turing computability, a countable set A is computable iff there is a computable subset of ω coding an isomorphic copy of A. By Proposition 2.2, every set can be similarly coded as a subset of a large enough limit ordinal. It makes sense then to simplify things by requiring the uniform base set of an ASM $\mathfrak{M}$ to be a limit ordinal $\kappa(\mathfrak{M})$. To successfully decode a set of ordinals using the Gödel pairing function, we also need the true membership relation to be a part of each state of $S(\mathfrak{M})$. In other words, $\ulcorner \in \urcorner \in \sigma(\mathfrak{M})$, and whenever $s \in S(\mathfrak{M})$, s interprets $\ulcorner \in \urcorner$ as the true membership relation so that s an expansion of the structure $(\kappa(\mathfrak{M}); \in)$.

Now we can formally state our requirements for $\phi^d_{\mathfrak{M}}$. Taking a leaf from Definition 2.19, for each

$$\ulcorner X \urcorner \in \sigma(\mathfrak{M}) \setminus \{\ulcorner \in \urcorner, \ulcorner \mathrm{In} \urcorner, \ulcorner \mathrm{Out} \urcorner\}$$

there must be $\phi^{\ulcorner X \urcorner}$ and $\psi^{\ulcorner X \urcorner}$ such that

- $\phi^{\ulcorner X \urcorner}$ is a formula over $\{\ulcorner \in \urcorner\}$,
- $\psi^{\ulcorner X \urcorner}$ is a sentence over $\sigma(\mathfrak{M}) \setminus \{\ulcorner \mathrm{In} \urcorner, \ulcorner \mathrm{Out} \urcorner\}$,
- if $\ulcorner X \urcorner$ is an n-ary function symbol, then
  - $\phi^{\ulcorner X \urcorner}$ contains $n+1$ free variables, and
  - for some variable symbols $y_1, \ldots, y_{n+1}$,

$$\psi^{\ulcorner X \urcorner} = \ulcorner \forall y_1 \ldots \forall y_{n+1} \, (X(y_1, \ldots, y_n) = y_{n+1} \iff \phi^{\ulcorner X \urcorner}(y_1, \ldots, y_{n+1})) \urcorner,$$

- if $\ulcorner X \urcorner$ is an n-ary relation symbol, then
  - $\phi^{\ulcorner X \urcorner}$ contains n free variables, and
  - for some variable symbols $y_1, \ldots, y_n$,

$$\psi^{\ulcorner X \urcorner} = \ulcorner \forall y_1 \ldots \forall y_n \, (X(y_1, \ldots, y_n) \iff \phi^{\ulcorner X \urcorner}(y_1, \ldots, y_n)) \urcorner,$$

- if $\ulcorner X \urcorner$ is a constant symbol, then
  - $\phi^{\ulcorner X \urcorner}$ contains 1 free variable, and
  - for some variable symbol y,

$$\psi^{\ulcorner X \urcorner} = \ulcorner \forall y \, (X = y \iff \phi^{\ulcorner X \urcorner}(y)) \urcorner, \text{ and}$$

- $\phi^d_{\mathfrak{M}} = \bigwedge\{\psi^{\ulcorner X \urcorner} : \ulcorner X \urcorner \in \sigma\}$.

We call such $\phi^d_{\mathfrak{M}}$ *simple for* $\sigma(\mathfrak{M})$, and $\phi^{\ulcorner X \urcorner}$ the *witness for* $\ulcorner X \urcorner$ *to* $\phi^d_{\mathfrak{M}}$ *being simple for* $\sigma(\mathfrak{M})$ Additionally,

$$\ulcorner \phi^d_{\mathfrak{M}} \wedge \forall y\ (\neg \mathrm{Out}(y)) \wedge \mathrm{In}(x) \urcorner$$

must be a witness for a (necessarily unique, by Fact 2.22) loading function $l_{\mathfrak{M}}$ of $\mathfrak{M}$. We call $l_{\mathfrak{M}}$ the *default loading function* of $\mathfrak{M}$. The *default unloading function* $u_{\mathfrak{M}}$ of $\mathfrak{M}$ is the unique (by Fact 2.23) unloading function of $\mathfrak{M}$ for which $\ulcorner \mathrm{Out}(x) \urcorner$ is a witness.

Analogous to (C2), given $A \subset \kappa(\mathfrak{M})$, if there is s satisfying

$$\forall a \in \kappa(\mathfrak{M})\ (a \in A \iff s \models \ulcorner \phi^d_{\mathfrak{M}} \wedge \forall y\ (\neg \mathrm{Out}(y)) \wedge \mathrm{In}(a) \urcorner),$$

then such s is unique and absolute for transitive models of ZFC containing A and $\kappa(\mathfrak{M})$. By an argument similar to that through which (C4) is concluded, we can see that

$$\forall A \subset \kappa(\mathfrak{M})\ \exists! s \in S(\mathfrak{M})\ \forall a \in \kappa(\mathfrak{M})\ (a \in A \iff s \models \ulcorner \phi^d_{\mathfrak{M}} \wedge \forall y\ (\neg \mathrm{Out}(y)) \wedge \mathrm{In}(a) \urcorner)$$

is absolute for transitive models of ZFC containing $\kappa(\mathfrak{M})$. It goes without saying that a unique A satisfying

$$\forall a \in \kappa(\mathfrak{M})\ (a \in A \iff s \models \ulcorner \mathrm{Out}(a) \urcorner)$$

exists for any given $s \in S(\mathfrak{M})$, and is absolute for transitive models of ZFC containing s and $\kappa(\mathfrak{M})$. Likewise obvious is the absoluteness of the sentence

$$\forall s \in S(\mathfrak{M})\ \exists! A \subset \kappa(\mathfrak{M})\ \forall a \in \kappa(\mathfrak{M})\ (a \in A \iff s \models \ulcorner \mathrm{Out}(a) \urcorner)$$

for transitive models of ZFC. As a consequence, our designated witnesses for the default loading and unloading functions must also be absolute for transitive models of ZFC.

2.6 Constraining the Interpretation Constraints

Now that we require $\ulcorner \in \urcorner$ to be a member of the signature of every ASM, and for it to always be interpreted as the true membership relation, should there be further requirements on the interpretation constraints of $\mathfrak{T}(\mathfrak{M})$?

Example 2.24. Choose a limit ordinal κ and $A \subset \kappa$. Let $\mathfrak{M}$ be an ASM with $\kappa(\mathfrak{M}) = \kappa$, signature $\sigma(\mathfrak{M}) = \{\ulcorner \in \urcorner, \ulcorner \mathrm{In} \urcorner, \ulcorner \mathrm{Out} \urcorner, \ulcorner R \urcorner\}$ and associated TCI $\mathfrak{T}(\mathfrak{M}) = (\emptyset, \sigma(\mathfrak{M}), \dot{\mathcal{U}}, \vartheta)$, such that $\ulcorner \in \urcorner$, $\ulcorner \mathrm{In} \urcorner$ and $\ulcorner \mathrm{Out} \urcorner$ are as stipulated and $\ulcorner R \urcorner$ is a unary relation symbol. To wit, it must be the case that

$$\begin{aligned}
\vartheta(\dot{\mathcal{U}}) &= (\kappa, 1) \\
\vartheta(\ulcorner \in \urcorner) &= (\in \cap\, \kappa, 1) \\
\vartheta(\ulcorner \mathrm{In} \urcorner) &= (\kappa, 0) \\
\vartheta(\ulcorner \mathrm{Out} \urcorner) &= (\kappa, 0).
\end{aligned}$$

Now suppose

$$\vartheta(\ulcorner R\urcorner) = (A, 1)$$
$$\phi^d_{\mathfrak{M}} = \ulcorner\urcorner$$

and

$$\begin{aligned}\phi^\tau_{\mathfrak{M}} = \ulcorner(&\forall x\ (R_0(x) \iff \mathrm{Out}_0(x)) \implies \\ &\forall x\ ((\mathrm{In}_0(x) \iff \mathrm{In}_1(x)) \wedge (\mathrm{Out}_0(x) \iff \mathrm{Out}_1(x)))) \\ &\wedge (\neg(\forall x\ (R_0(x) \iff \mathrm{Out}_0(x))) \implies \\ &\forall x\ ((\mathrm{In}_0(x) \iff \mathrm{In}_1(x)) \wedge (R_0(x) \iff \mathrm{Out}_1(x))))\urcorner.\end{aligned}$$

Then basically, $\mathfrak{M}$ is a machine that starts with $\ulcorner R\urcorner$ holding A, before copying A onto the output tape. As it does this regardless of input, $\mathfrak{M}$ computes A from any input, even if A is a very complicated set by any other measure.

In the usual input-output paradigm of computability where the output complexity should depend solely on the input complexity, Example 2.24 cautions against hiding too much information in the states of an ASM. One way to fix this is to enforce certain definability constraints on the interpretation constraints.

Fix an ASM $\mathfrak{M}$ with $\mathfrak{T}(\mathfrak{M}) = (\emptyset, \sigma(\mathfrak{M}), \dot{\mathcal{U}}, \vartheta)$. We want A to be definable over the structure $(\kappa(\mathfrak{M}); \in)$ for all $(A, z) \in ran(\vartheta)$. In other words, if $(A, z) \in ran(\vartheta)$, then for some $n < \omega$ there is a formula φ over $\{\ulcorner\in\urcorner\}$ with free variables $x_1, \ldots, x_n$ such that

$$A = \{(\alpha_1, \ldots, \alpha_n) \in \kappa^n : (\kappa; \in) \models \varphi[x_1 \mapsto \alpha_n, \ldots, x_n \mapsto \alpha_n]\}.$$

Due to the fact that every state of $\mathfrak{M}$ is an expansion of $(\kappa(\mathfrak{M}); \in)$, we can actually include the definition of any such A in $\phi^\tau_{\mathfrak{M}}$ — more specifically, in each $\phi^{\ulcorner X\urcorner}$ for which $\vartheta(\ulcorner X\urcorner) = (A, z)$. Doing this would allow all the "programming" to occur in the formulation of $\phi^\tau_{\mathfrak{M}}$; all symbols in $\sigma(\mathfrak{M})$ besides $\ulcorner\in\urcorner$ can simply be viewed as declarations of the "programming variables" we expect to occur in $\phi^\tau_{\mathfrak{M}}$.

Justifiably, we shall stipulate that whenever $(A, z) \in ran(\vartheta)$, $z = 0$ and $A = \kappa^n$ for some $n < \omega$. We are now ready to formally define our modified version of an ASM.

Definition 2.25. A *sequential restricted abstract state machine* — or sequential RASM (henceforth just RASM) — $\mathfrak{M}$ is fully specified by a limit ordinal $\kappa(\mathfrak{M})$, a finite signature $\sigma(\mathfrak{M})$, and sentences $\phi^\tau_{\mathfrak{M}}$ and $\phi^d_{\mathfrak{M}}$ such that

(D1) $\{\ulcorner\in\urcorner, \ulcorner\mathrm{In}\urcorner, \ulcorner\mathrm{Out}\urcorner\} \subset \sigma(\mathfrak{M})$,
(D2) there is a unique $\vartheta_{\mathfrak{M}} : (\sigma(\mathfrak{M}) \cup \{\dot{\mathcal{U}}\}) \longrightarrow V$ with the following properties:

- $\vartheta(\dot{\mathcal{U}}) = (\kappa(\mathfrak{M}), 1)$,
- $\vartheta(\ulcorner\in\urcorner) = (\in, 1)$,

- $\vartheta(\ulcorner \mathrm{In} \urcorner) = (\kappa(\mathfrak{M}), 0)$,
- $\vartheta(\ulcorner \mathrm{Out} \urcorner) = (\kappa(\mathfrak{M}), 0)$, and
- for all $\ulcorner X \urcorner \in \sigma(\mathfrak{M}) \setminus \{\ulcorner \in \urcorner, \ulcorner \mathrm{In} \urcorner, \ulcorner \mathrm{Out} \urcorner\}$ there is $n < \omega$ satisfying

$$\vartheta(\ulcorner X \urcorner) = (\kappa(\mathfrak{M})^n, 0),$$

(D3) there is a unique $\mathfrak{T}(\mathfrak{M})$ for which $\mathfrak{T}(\mathfrak{M}) = (\emptyset, \sigma(\mathfrak{M}), \dot{\mathcal{U}}, \vartheta_{\mathfrak{M}})$,
(D4) there is a unique $S(\mathfrak{M})$ for which $S(\mathfrak{M}) = \{s : s \models^* \mathfrak{T}(\mathfrak{M})\}$,
(D5) $\phi^{\tau}_{\mathfrak{M}}$ is bounded for $\sigma(\mathfrak{M})$,
(D6) for all $s_0 \in S(\mathfrak{M})$ there is a unique $s_1 \in S(\mathfrak{M})$ for which $(s_0, s_1) \models_2 \phi^{\tau}_{\mathfrak{M}}$,
(D7) $\phi^{d}_{\mathfrak{M}}$ is simple for $\sigma(\mathfrak{M})$, and
(D8) for all $A \subset \kappa(\mathfrak{M})$ there is a unique $s \in S(\mathfrak{M})$ for which

$$\forall a \in \kappa(\mathfrak{M})\ (a \in A \iff s \models \ulcorner \phi^{d}_{\mathfrak{M}} \wedge \forall y\ (\neg \mathrm{Out}(y)) \wedge \mathrm{In}(a) \urcorner).$$

In this case, we say $\mathfrak{M} = (\kappa(\mathfrak{M}), \sigma(\mathfrak{M}), \phi^{\tau}_{\mathfrak{M}}, \phi^{d}_{\mathfrak{M}})$.

Remark 2.26. Fix any limit ordinal κ. Notice that there are only countably many RASMs $\mathfrak{M}$ satisfying $\kappa(\mathfrak{M}) = \kappa$. Indeed, this countable set of RASM specifications is absolute for transitive models of ZFC containing κ.

Definition 2.27. If $\mathfrak{M} = (\kappa(\mathfrak{M}), \sigma(\mathfrak{M}), \phi^{\tau}_{\mathfrak{M}}, \phi^{d}_{\mathfrak{M}})$ is an RASM, define

- $S(\mathfrak{M})$ as in Definition 2.25,
- $\tau_{\mathfrak{M}} := \{(s_0, s_1) \in S(\mathfrak{M})^2 : (s_0, s_1) \models_2 \phi^{\tau}_{\mathfrak{M}}\}$,
- $l_{\mathfrak{M}} := \{(A, s) \in \mathcal{P}(\kappa(\mathfrak{M})) \times S(\mathfrak{M}) :$
$$\forall a \in \kappa(\mathfrak{M})\ (a \in A \iff s \models \ulcorner \phi^{d}_{\mathfrak{M}} \wedge \forall y\ (\neg \mathrm{Out}(y)) \wedge \mathrm{In}(a) \urcorner)\}$$
- $I(\mathfrak{M}) := ran(l_{\mathfrak{M}})$
- $u_{\mathfrak{M}} := \{(s, A) \in S(\mathfrak{M}) \times \mathcal{P}(\kappa(\mathfrak{M})) :$
$$\forall a \in \kappa(\mathfrak{M})\ (a \in A \iff s \models \ulcorner \mathrm{Out}(a) \urcorner)\}.$$

The next proposition highlights the important properties of RASM, based on what we have discussed so far.

Proposition 2.28. *Let $\mathfrak{M}$ be an RASM. Then the following are absolute for transitive models of* ZFC *containing $\mathfrak{T}(\mathfrak{M})$:*

- *the definition of $S(\mathfrak{M})$,*
- *the definition of $\tau_{\mathfrak{M}}$ and $\tau_{\mathfrak{M}}$ being a function from $S(\mathfrak{M})$ into $S(\mathfrak{M})$,*
- *the definition of $l_{\mathfrak{M}}$ and $l_{\mathfrak{M}}$ being a function from $\mathcal{P}(\kappa(\mathfrak{M}))$ into $S(\mathfrak{M})$,*
- *the definition of $I(\mathfrak{M})$,*
- *the definition of $u_{\mathfrak{M}}$ and $u_{\mathfrak{M}}$ being a function from $S(\mathfrak{M})$ into $\mathcal{P}(\kappa(\mathfrak{M}))$.*

Now that RASMs have been officially defined, it is imperative to revisit Example 2.13, for any notion of generalised algorithm that cannot represent an arbitrary Turing cannot be too useful.

Example 2.29 (2.13 revisited). Let us describe changes to the $\mathfrak{M}$ defined in Example 2.13 that would qualify it for an RASM. First, an observation: there are formulas ϕ_P and ϕ_S over $\ulcorner\in\urcorner$, each with two free variables x and y, such that for all $s \in S(\mathfrak{M})$,

$$s \models \forall x\, \forall y\, ((\mathrm{Pre}(x) = y \iff \phi_P(x,y)) \wedge (\mathrm{Suc}(x) = y \iff \phi_S(x,y))).$$

Since the definition of the functions Pre and Suc are absolute for transitive models of ZFC, we can omit $\ulcorner\mathrm{Pre}\urcorner$ and $\ulcorner\mathrm{Suc}\urcorner$ from $\sigma(\mathrm{M})$.

Next, we shall replace $\ulcorner R\urcorner$ with $\ulcorner\mathrm{Out}\urcorner$ while adding $\ulcorner\mathrm{In}\urcorner$ to $\sigma(\mathrm{M})$. We also want to have another constant symbol $\ulcorner e\urcorner$ in $\sigma(\mathrm{M})$. Set

$$\vartheta_{\mathfrak{M}}(\ulcorner e\urcorner) := (\omega, 0)$$
$$\phi^d_{\mathfrak{M}} := \ulcorner h = \langle 0\rangle \wedge t = \langle 0\rangle \wedge e = \langle 0\rangle\urcorner.$$

We are left to define the $\phi^{\ulcorner X\urcorner}$s for $\ulcorner X\urcorner \in \sigma(\mathfrak{M})$. The only non-trivial ones are for $\ulcorner X\urcorner \in \{\ulcorner h\urcorner, \ulcorner t\urcorner, \ulcorner e\urcorner, \ulcorner\mathrm{In}\urcorner, \ulcorner\mathrm{Out}\urcorner\}$. So let

$$\begin{aligned}
\phi^{\ulcorner h\urcorner}(x) := \ulcorner &(e_0 = \langle 0\rangle \wedge h_0 = x) \vee \\
&\bigvee\{(\neg e_0 = \langle 0\rangle) \wedge q(\mathrm{Out}_0(h_0), k) \wedge t_0 = \langle j\rangle \wedge x = \mathrm{Pre}(h_0) : \\
&\exists x\, \exists y\, (\delta((c_j, k)) = (x, y, \mathrm{left}))\} \vee \\
&\bigvee\{(\neg e_0 = \langle 0\rangle) \wedge q(\mathrm{Out}_0(h_0), k) \wedge t_0 = \langle j\rangle \wedge x = \mathrm{Suc}(h_0) : \\
&\exists x\, \exists y\, (\delta((c_j, k)) = (x, y, \mathrm{right}))\}\urcorner
\end{aligned}$$

$$\begin{aligned}
\phi^{\ulcorner t\urcorner}(x) := \ulcorner &(e_0 = \langle 0\rangle \wedge t_0 = x) \vee \\
&\bigvee\{(\neg e_0 = \langle 0\rangle) \wedge q(\mathrm{Out}_0(h_0), k) \wedge t_0 = \langle j\rangle \wedge x = \langle j'\rangle : \\
&\exists x\, \exists y\, (\delta((c_j, k)) = (c_{j'}, x, y))\}\urcorner \\
\phi^{\ulcorner e\urcorner}(x) := \ulcorner &(e_0 = \langle 0\rangle \wedge x = \langle 1\rangle) \vee ((\neg e_0 = \langle 0\rangle) \wedge x = e_0)\urcorner \\
\phi^{\ulcorner \mathrm{In}\urcorner}(x) := \ulcorner &\mathrm{In}_0(x)\urcorner \\
\phi^{\ulcorner \mathrm{Out}\urcorner}(x) := \ulcorner &(e_0 = \langle 0\rangle \wedge \mathrm{In}_0(x)) \vee ((\neg e_0 = \langle 0\rangle) \wedge (\neg x = h_0) \wedge \mathrm{Out}_0(x)) \vee \\
&\bigvee\{(\neg e_0 = \langle 0\rangle) \wedge q(\mathrm{Out}_0(h_0), k) \wedge t_0 = \langle j\rangle \wedge x = h_0 \wedge (\neg x = x) : \\
&\exists x\, \exists y\, (\delta((c_j, k)) = (x, 0, y))\} \vee \\
&\bigvee\{(\neg e_0 = \langle 0\rangle) \wedge q(\mathrm{Out}_0(h_0), k) \wedge t_0 = \langle j\rangle \wedge x = h_0 \wedge x = x : \\
&\exists x\, \exists y\, (\delta((c_j, k)) = (x, 1, y))\}\urcorner,
\end{aligned}$$

where $x = \mathrm{Pre}(h_0)$ and $x = \mathrm{Suc}(h_0)$ are shorthand notations for semantically equivalent formulas over $\{\ulcorner \in_0 \urcorner, \ulcorner h_0 \urcorner\}$.

A rotuine check would reveal that the aforementioned details are sufficient to specify $\mathfrak{M}$ as an RASM, and that they allow $\mathfrak{M}$ to accurately simulate the Turing machine M.

From the next subsection onwards, the construction of RASMs used in proofs and arguments can be too cumbersome to formally spell out. In such cases, only high level details will be given. The reader should always try to convince themselves that those details can be properly realised by RASMs.

2.7 Transfinite Runs

Consider a finite terminating run $\vec{s} = (s_1, \ldots, s_n)$ of an RASM $\mathfrak{M}$. Then by combining witnesses to $\phi^{\tau}_{\mathfrak{M}}$ being bounded for $\sigma(\mathfrak{M})$, we can conjure witnesses to some sentence ϕ being bounded for $\sigma(\mathfrak{M})$, such that using ϕ in place of $\phi^{\tau}_{\mathfrak{M}}$ in the definition of $\mathfrak{M}$ gives us an RASM $\mathfrak{M}'$ with $\tau_{\mathfrak{M}'}(s_1) = s_n$. Therefore, as far as computability is concerned, allowing finite steps of arbitrary sizes in computations is no more powerful than allowing only one-step computations. This goes to show that RASMs are in fact built for runs of infinite lengths.

Given an infinite ordinal δ, what does it mean to have an RASM computation of length δ? Naturally, every state of the computation must depend on the states prior. For successor ordinals $\alpha < \delta$, the α-th state of the computation depends solely on its unique previous state, according to the transition function of the RASM. But what about limit ordinals less than δ? How do we describe the states associated with these ordinals in terms of states occurring in the computation before them? This is equivalent to asking how we are going to process the limit of a sequence of states.

Fix an RASM $\mathfrak{M}$, a limit ordinal δ, and let Sq be a sequence of states of $\mathfrak{M}$. We want to have a uniform definition of $lim(Sq) \in S(\mathfrak{M})$. One natural way to do so is to take ordinal limits point-wise (according to the order topology on $\kappa(\mathfrak{M})$ induced by $\in$), while treating relations as boolean functions. The obvious hindrance to this method is, the limit of a sequence of ordinals does not always exist. However, for various purposes, there are suitable ersatzes.

Definition 2.30. Let T be a function from some ordinal into ORD. Then $lim\ inf(T)$, $lim\ sup(T)$ and $lim(T)$ refer respectively to the limit inferior, the limit superior and the limit of T as a sequence, with respect to the order topology induced on $\kappa(\mathfrak{M})$ by $\in$.

Fact 2.31. If T is a function from some ordinal into ORD, then $lim\ inf(T)$ and $lim\ sup(T)$ always exist.

Fact 2.32. If T is a function from $\delta < \kappa(\mathfrak{M})$ into ORD and

$$sup\{T(\alpha) : \alpha \in \delta\} < \kappa(\mathfrak{M}),$$

then $lim\ inf(T)$ and $lim\ sup(T)$ always exist in $\kappa(\mathfrak{M})$.

Point-wise $lim\ inf$ and $lim\ sup$ operations are commonly used to define the limit stages of transfinite computations. Hamkins and Lewis, in [7], defined each cell in the tape of an infinite time Turing machine to take the $lim\ sup$ of its previous values. Koepke, in his definition of ordinal Turing machines ([10] and [11]), instead require each cell of the machine tape to take the $lim\ inf$ of its previous values. Point-wise $lim\ inf$ and $lim\ sup$ always exist in these generalised Turing machines, for computations shorter than the lengths of their tapes. This is because single-step movements of these machines are always small, and consequently the hypothesis of Fact 2.32 applies.

Hereon, we shall identify relations with boolean functions and constants with 0-ary functions wherever convenient, in part so that only function symbols need to be considered in definitions and case analyses.

Definition 2.33. Let $\mathfrak{M}$ be an RASM and Sq be a function from some ordinal into $S(\mathfrak{M})$. Given $\ulcorner X \urcorner \in \sigma(\mathfrak{M})$ an n-ary function symbol, denote by $\ulcorner R^{\ulcorner X \urcorner} \urcorner$ a distinguished $(n+2)$-ary relation symbol and define the relation

$$R_{Sq}^{\ulcorner X \urcorner} := \{(\alpha, \beta_1, \ldots, \beta_n, \beta) \in dom(Sq) \times \kappa(\mathfrak{M})^{n+1} : X^{Sq(\alpha)}(\beta_1, \ldots, \beta_n) = \beta\}.$$

Definition 2.34. Let $\mathfrak{M}$ be an RASM. Given $\ulcorner X \urcorner \in \sigma(\mathfrak{M})$ an n-ary function symbol, we say ϕ is a *simple limit formula for* $(\ulcorner X \urcorner, \mathfrak{M})$ iff ϕ is a formula over $\{\ulcorner \in \urcorner, \ulcorner R^{\ulcorner X \urcorner} \urcorner\}$ with free variables $x_1, \ldots, x_{n+1}$.

Definition 2.35. Let $\mathfrak{M}$ be an RASM. Given $\ulcorner X \urcorner \in \sigma(\mathfrak{M})$ an n-ary function symbol and ϕ a *simple limit formula for* $(\ulcorner X \urcorner, \mathfrak{M})$, we say ϕ *preserves ordinal limits* iff interpreting $\ulcorner \in \urcorner$ as $\in$ and $\ulcorner R^{\ulcorner X \urcorner} \urcorner$ as $R_{Sq}^{\ulcorner X \urcorner}$,

$$(\kappa(\mathfrak{M}); \in, R_{Sq}^{\ulcorner X \urcorner}) \models \phi[x_1 \mapsto \alpha_1, \ldots, x_{n+1} \mapsto \alpha_{n+1}]$$

for all functions Sq from a limit ordinal $< \kappa(\mathfrak{M})$ into $S(\mathfrak{M})$ and all $(\alpha_1, \ldots, \alpha_{n+1}) \in \kappa(\mathfrak{M})^{n+1}$ such that

$$\alpha_{n+1} = lim\{(\beta, \ulcorner X \urcorner^{Sq(\beta)}(\alpha_1, \ldots, \alpha_n)) : \beta < dom(Sq)\}.$$

Definition 2.36. Let $\mathfrak{M}$ be an RASM and Sq be a function from a limit ordinal $< \kappa(\mathfrak{M})$ into $S(\mathfrak{M})$. A state $s \in S(\mathfrak{M})$ is a *simple limit of* Sq iff for each n-ary function symbol $\ulcorner X \urcorner \in \sigma(\mathfrak{M})$ there is a simple limit formula $\phi_{lim}^{\ulcorner X \urcorner}$ for $(\ulcorner X \urcorner, \mathfrak{M})$ such that, interpreting $\ulcorner \in \urcorner$ as $\in$ and $\ulcorner R^{\ulcorner X \urcorner} \urcorner$ as $R_{Sq}^{\ulcorner X \urcorner}$,

$$\ulcorner X \urcorner^s(\alpha_1, \ldots, \alpha_n) = \alpha_{n+1} \iff (\kappa(\mathfrak{M}); \in, R_{Sq}^{\ulcorner X \urcorner}) \models \phi[x_1 \mapsto \alpha_1, \ldots, x_{n+1} \mapsto \alpha_{n+1}]$$

whenever $(\alpha_1, \ldots, \alpha_{n+1}) \in \kappa(\mathfrak{M})^{n+1}$. In this case, we say $\{\phi_{lim}^{\ulcorner X \urcorner} : \ulcorner X \urcorner \in \sigma(\mathfrak{M})\}$ *witnesses* s *is a simple limit of* Sq.

It is immediate from Definition 2.36 and our assumption that $\sigma(\mathfrak{M})$ contains only function symbols, that a simple limit of a $< \kappa(\mathfrak{M})$-length ordinal sequence of states of $\mathfrak{M}$ must be unique if it exists. Furthermore, for any such sequence Sq and any set Y, whether Y witnesses the existence of a simple limit of Sq, as well as the identity of said limit if it exists, is absolute for transitive models of ZFC containing Sq.

Proposition 2.37. *Let $\mathfrak{M}$ be an RASM. For each n-ary function symbol $\ulcorner X \urcorner \in \sigma(\mathfrak{M})$ there is simple limit formula $\phi_{lim}^{\ulcorner X \urcorner}$ for $(\ulcorner X \urcorner, \mathfrak{M})$ that preserves ordinal limits, such that whenever Sq is a function from $\delta < \kappa(\mathfrak{M})$ into $S(\mathfrak{M})$, $\{\phi_{lim}^{\ulcorner X \urcorner} : \ulcorner X \urcorner \in \sigma(\mathfrak{M})\}$ witnesses the existence of a simple limit s of Sq satisfying*

$$\ulcorner X \urcorner^s(\alpha_1, ..., \alpha_n) = \begin{cases} lim\ inf\{(\beta, \ulcorner X \urcorner^{Sq(\beta)}(\alpha_1, ..., \alpha_n)) : \beta < \delta\} & \text{if it exists } < \kappa(\mathfrak{M}) \\ 0 & \text{otherwise} \end{cases}$$

for every n-ary function symbol $\ulcorner X \urcorner \in \sigma(\mathfrak{M})$.

Proof. By the definition of an RASM and the observation that

$$lim\ inf(T) = lim(T)$$

whenever T is a function from some ordinal in ORD and $lim(T)$ exists. The reader should check that the $lim\ inf$ values specified by the proposition relative to $\ulcorner X \urcorner$ and Sq, in case they exist, are definable over $(\kappa(\mathfrak{M}); \in, R_{Sq}^{\ulcorner X \urcorner})$. In fact, there is one such definition that depends only on $\ulcorner X \urcorner$ and not on $\mathfrak{M}$. □

Colloquially, Proposition 2.37 tells us that every RASM is *strongly closed under limit inferiors*. The following definitions generalise this concept.

Definition 2.38. Let $\mathfrak{M}$ be an RASM. Then $\mathfrak{M}$ is *strongly closed under* a set Y iff for each n-ary function symbol $\ulcorner X \urcorner \in \sigma(\mathfrak{M})$ there is a simple limit formula $\phi_{lim}^{\ulcorner X \urcorner}$ for $(\ulcorner X \urcorner, \mathfrak{M})$, such that

$$Y = \{\phi_{lim}^{\ulcorner X \urcorner} : \ulcorner X \urcorner \in \sigma(\mathfrak{M})\}$$

and whenever Sq is a function from some limit ordinal $< \kappa(\mathfrak{M})$ into $S(\mathfrak{M})$, there is $s \in S(\mathfrak{M})$ witnessed by Y to be a simple limit of Sq.

Definition 2.39. Let $\mathfrak{M}$ be an RASM. Then $\mathfrak{M}$ is *practically closed under* a set Y iff for each n-ary function symbol $\ulcorner X \urcorner \in \sigma(\mathfrak{M})$ there is a simple limit formula $\phi_{lim}^{\ulcorner X \urcorner}$ for $(\ulcorner X \urcorner, \mathfrak{M})$, such that

$$Y = \{\phi_{lim}^{\ulcorner X \urcorner} : \ulcorner X \urcorner \in \sigma(\mathfrak{M})\}$$

and whenever Sq is a function from some limit ordinal $\delta < \kappa(\mathfrak{M})$ into $S(\mathfrak{M})$ satisfying

- $Sq(0) \in I(\mathfrak{M})$,
- $Sq(\alpha + 1) = \tau_{\mathfrak{M}}(Sq(\alpha))$ for all $\alpha < \delta$, and

- Y witnesses $Sq(\gamma)$ is a simple limit of $Sq \restriction \gamma$ for all limit $\gamma < \delta$,

there is $s \in S(\mathfrak{M})$ witnessed by Y to be a simple limit of Sq.

Clearly, for any RASM $\mathfrak{M}$ and any set Y,

$$\mathfrak{M} \text{ is strongly closed under } Y \implies \mathfrak{M} \text{ is practically closed under } Y.$$

We are ready to define transfinite terminating runs of RASMs.

Definition 2.40. Let $\mathfrak{M}$ be an RASM that is practically closed under Y. A *short terminating run of* $(\mathfrak{M}, Y)$ *with input* A *and output* B is a function Sq from some successor ordinal $\delta < \kappa(\mathfrak{M})$ into $S(\mathfrak{M})$ such that

- $A, B \subset \kappa(\mathfrak{M})$,
- $Sq(0) = l_{\mathfrak{M}}(A)$,
- $Sq(\alpha + 1) = \tau_{\mathfrak{M}}(Sq(\alpha)) \neq Sq(\alpha)$ for all $\alpha < \delta - 1$,
- Y witnesses $Sq(\gamma)$ is a simple limit of $Sq \restriction \gamma$ for all limit $\gamma < \delta$,
- $\tau_{\mathfrak{M}}(Sq(\delta - 1)) = Sq(\delta - 1)$, and
- $u_{\mathfrak{M}}(Sq(\delta - 1)) = B$.

Proposition 2.41. *Suppose $\mathfrak{M}$ is an RASM practically closed under Y. Then for some signature σ', as long as $\mathfrak{M}'$ is an RASM satisfying*

- $\sigma(\mathfrak{M}') = \sigma'$ *and*
- $\kappa(\mathfrak{M}') = \kappa(\mathfrak{M})$,

and Z is a set of simple limit formulas preserving ordinal limits with $\mathfrak{M}'$ strongly closed under Z, there is an RASM $\mathfrak{M}^*$ such that

(1) $\sigma(\mathfrak{M}^*) = \sigma'$,
(2) $\kappa(\mathfrak{M}^*) = \kappa(\mathfrak{M})$, *and*
(3) for all $A, B \subset \kappa(\mathfrak{M})$, a short terminating run of $(\mathfrak{M}, Y)$ with input A and output B exists iff a short terminating run of $(\mathfrak{M}^*, Z)$ with input A and output B exists.

Proof. Define

$$\sigma' := \sigma(\mathfrak{M}) \cup \{\ulcorner R^{\ulcorner X \urcorner} \urcorner : \ulcorner X \urcorner \in \sigma(\mathfrak{M})\} \cup \{\ulcorner c_0 \urcorner, \ulcorner c_1 \urcorner\},$$

where $\ulcorner c_0 \urcorner, \ulcorner c_1 \urcorner$ are constant symbols distinct from all members of

$$\sigma(\mathfrak{M}) \cup \{\ulcorner R^{\ulcorner X \urcorner} \urcorner : \ulcorner X \urcorner \in \sigma(\mathfrak{M})\}.$$

Let $\mathfrak{M}'$ and Z be as given by the proposition. We need to define $\mathfrak{M}^*$ so that (1) to (3) are satisfied. We are thus forced to set

$$\sigma(\mathfrak{M}^*) := \sigma'$$
$$\kappa(\mathfrak{M}^*) := \kappa.$$

But this means $S(\mathfrak{M}^*) = S(\mathfrak{M}')$, so $\mathfrak{M}^*$ is strongly closed under Z. We briefly describe the default "non-tape" configuration and the transition function of $\mathfrak{M}^*$.

The symbols in $\sigma(\mathfrak{M})$ have default interpretations following $\mathfrak{M}$. For all $\ulcorner X \urcorner \in \sigma(\mathfrak{M})$, the default interpretation of $\ulcorner R^{\ulcorner X \urcorner} \urcorner$ is the empty set. The two extra constant symbols $\ulcorner c_0 \urcorner$ and $\ulcorner c_1 \urcorner$ default to 0 and 1 respectively.

At every step of a computation of $\mathfrak{M}^*$, it checks if "$c_0 = c_1$". If the answer is no, it copies the previous $\mathfrak{M}$ portion data to the $R^{\ulcorner X \urcorner}$s within the fibers denoted by c_0 (which is supposed to count the steps in every short terminating run), simulates the next step of its $\mathfrak{M}$ portion, and increments both c_0 and c_1. Otherwise, the moral implication is that a limit stage has been reached, whence it first verifies if Y is functional on the current $R^{\ulcorner X \urcorner}$s (note that the $R^{\ulcorner X \urcorner}$s might contain gibberish in states not reachable from any initial state). If so, it computes point-wise limits of the $R^{\ulcorner X \urcorner}$s according to Y, copies the resulting limit state information to its $\mathfrak{M}$ portion, and increments just c_1. Otherwise, it terminates. We can argue by induction that $\mathfrak{M}^*$ always correctly computes the states of a short terminating run of $\mathfrak{M}$, primarily because the preservation of ordinal limits by the simple limit formulas in Z allows both

- the preservation of data previously stored in the $R^{\ulcorner X \urcorner}$s, and
- the pair of constants c_0 and c_1 to flag limit stages of a terminating run.

In conclusion, the $\mathfrak{M}$ portion of $\mathfrak{M}^*$ simulates (with some latency) the entirety of each short terminating run of $\mathfrak{M}$. As said portion contains both the input and output tapes of $\mathfrak{M}^*$, (3) holds. □

The following definition is now well-motivated.

Definition 2.42. Let $\mathfrak{M}$ and $Y_{\text{lf}} := \{\phi_{lim}^{\ulcorner X \urcorner} : \ulcorner X \urcorner \in \sigma(\mathfrak{M})\}$ be as in Proposition 2.37. A *short terminating run of* $\mathfrak{M}$ *with input* A *and output* B is a short terminating run of $(\mathfrak{M}, Y_{\text{lf}})$ with input A and output B.

Basically, we are mandating the taking of limit inferiors point-wise wherever possible to be the standard method by which we compute limit states of transfinite runs. This is not a real restriction, for we have established from Propositions 2.37 and 2.41 that our new standard is among the most powerful ways to take limits of an RASM's states, while still maintaining absoluteness and the spirit of local definability.

Until now, we have only considered "short" transfinite runs, that is, runs that take less steps than the length of the input/output tape. What about longer runs? After all, the ability for Turing machines to run forever is the reason they can "output" non-computable computably enumerable sets. Allowing a Turing machine to run transfinitely (e.g. in [7]) allows it to explicitly decide these sets. Through such means, oracle computation becomes more straightforward: the oracle can be incorporated into the input, and the result of the machine being fed that oracle can be directly written on the output tape. We want to have an analogous concept of "long" runs on RASMs. It turns out that removing the length constraint on terminating runs works just fine;

Definition 2.43. Let $\mathfrak{M}$ be an RASM. A *terminating run of* $\mathfrak{M}$ *with input* A *and output* B is a function Sq from some successor ordinal δ into $S(\mathfrak{M})$ such that

- $A, B \subset \kappa(\mathfrak{M})$,
- $Sq(0) = l_{\mathfrak{M}}(A)$,
- $Sq(\alpha + 1) = \tau_{\mathfrak{M}}(Sq(\alpha)) \neq Sq(\alpha)$ for all $\alpha < \delta - 1$,
- for every limit $\gamma < \delta$, n-ary function symbol $\ulcorner X \urcorner \in \sigma(\mathfrak{M})$, and $\alpha_1, \ldots, \alpha_n \in \kappa(\mathfrak{M})$,

$$\ulcorner X \urcorner^{Sq(\gamma)}(\alpha_1, ..., \alpha_n) = \begin{cases} lim\ inf\{(\beta, \ulcorner X \urcorner^{Sq(\beta)}(\alpha_1, ..., \alpha_n)) : \beta < \gamma\} & \text{if it exists} \\ 0 & \text{otherwise,} \end{cases}$$

- $\tau_{\mathfrak{M}}(Sq(\delta - 1)) = Sq(\delta - 1)$, and
- $u_{\mathfrak{M}}(Sq(\delta - 1)) = B$.

A key realisation informing Definition 2.43 is that the existence, the uniqueness and the computation of limit inferiors, with regards to sequences of ordinals indexed by ordinals, are absolute. This, coupled with Proposition 2.28, gives us the absoluteness of terminating runs. More precisely, let κ be a limit ordinal and A, B be two sets of ordinals. Due to the absoluteness of taking limit inferior in general, the sets

$$\mathrm{SRuns}_\kappa(A, B) := \{(\mathfrak{M}, Sq) : \mathfrak{M} \text{ is an RASM with } \kappa(\mathfrak{M}) = \kappa \text{ and } Sq \text{ is a short terminating run of } \mathfrak{M} \text{ with input } A \text{ and output } B\}$$

and

$$\mathrm{Runs}_\kappa(A, B) := \{(\mathfrak{M}, Sq) : \mathfrak{M} \text{ is an RASM with } \kappa(\mathfrak{M}) = \kappa \text{ and } Sq \text{ is a terminating run of } \mathfrak{M} \text{ with input } A \text{ and output } B\}$$

as defined are absolute for transitive models of ZFC containing A, B and κ.

Remark 2.44. For a limit ordinal κ and two sets of ordinals A and B, that $\mathrm{Runs}_\kappa(A, B)$ is a set and not a proper class stems from the fact that by our definition of terminating runs, Sq must be an injection into $S(\mathfrak{M})$ whenever $(\mathfrak{M}, Sq) \in \mathrm{Runs}_\kappa(A, B)$. The aforementioned fact also means that all terminating runs of an RASM $\mathfrak{M}$ must be of length $< (2^{\kappa(\mathfrak{M})})^+$.

3 Computability and Comparisons

In this section, we derive from our restricted abstract state machines, various notions of generalised (relative) computability. We then compare these notions with existing ones.

3.1 Relative Computability

As prefaced in the previous subsection, we aim to use our definition of terminating runs to capture the notion of relative computability right off the bat.

Definition 3.1. For any limit ordinal κ and any two sets of ordinals A and B, B is *κ-computable from A* — denoted $B \leq_\kappa^C A$ — iff

- $A, B \subset \kappa$, and
- for some RASM $\mathfrak{M}$ with $\kappa(\mathfrak{M}) = \kappa$, there is a terminating run of $\mathfrak{M}$ with input A and output B.

In this case, $\mathfrak{M}$ *witnesses* $B \leq_\kappa^C A$.

Definition 3.2. We say B is *computable from* A, or $B \leq^C A$, iff $B \leq_\kappa^C A$ for some limit ordinal κ.

By Proposition 2.28, Remark 2.26 and the absoluteness of how the $\mathrm{Runs}_\kappa(A, B)$s are defined, $\leq_\kappa^C$ is absolute for transitive models of ZFC containing κ. Further, $\leq^C$ is absolute for transitive models of ZFC with the same ordinals.

Given any candidate for relative computability, we want to approach a definition for outright computability, i.e. computability without qualification. Perhaps the most natural way to do so is to define computable sets as sets that can be computed with the simplest of inputs.

Definition 3.3. Let A be a set of ordinals.

(1) A is *κ-computable* iff $A \leq_\kappa^C \emptyset$.
(2) A is *computable* iff A is κ-computable for some limit ordinal κ.

Let us look into some basic properties of the relations $\leq_\kappa^C$ and $\leq^C$, in an attempt to reflect on the validity of our notion of (relative) computability. Intuitively, any good notion of relative computability should be transitive.

Proposition 3.4. *Let κ be a limit ordinal. Suppose $A \leq_\kappa^C B$ and $B \leq_\kappa^C C$. Then $A \leq_\kappa^C C$.*

Proof. Let $\mathfrak{M}_1$ witness $B \leq_\kappa^C C$ and $\mathfrak{M}_2$ witness $A \leq_\kappa^C B$. We shall construct an RASM $\mathfrak{M}^*$ with $\kappa(\mathfrak{M}^*) = \kappa$ that essentially simulates $\mathfrak{M}_1$ followed by $\mathfrak{M}_2$, so that $\mathfrak{M}^*$ witnesses $A \leq_\kappa^C C$. Set $\sigma(\mathfrak{M}^*)$ to be a disjoint union of $\sigma(\mathfrak{M}_1)$ and $\sigma(\mathfrak{M}_2)$, and

$$\phi^d_{\mathfrak{M}^*} := \phi^d_{\mathfrak{M}_1} \wedge \phi^d_{\mathfrak{M}_2}.$$

At the beginning, the $\mathfrak{M}^*$ simulates $\mathfrak{M}_1$ in its $\mathfrak{M}_1$ portion, updating this portion of each configuration while keeping the "non-tape" $\mathfrak{M}_2$ portion undisturbed. In the background, checks are consistently being done. At the start of each step, before an update to its $\mathfrak{M}_1$

portion is carried out, $\mathfrak{M}^*$ checks if said portion will change going from the current state to the next. If so, it continues to simulate $\mathfrak{M}_1$. Otherwise, it copies the contents of the output tape to the input tape, erases all data from the output tape, and starts having its $\mathfrak{M}_2$ portion simulate $\mathfrak{M}_2$. From then on, the "non-tape" $\mathfrak{M}_1$ portion of each configuration is left untouched, so that $\mathfrak{M}^*$ terminates exactly when its simulation of $\mathfrak{M}_2$ ends.

Checking that $\mathfrak{M}^*$ fulfils our requirement is straightforward. □

It makes sense that a machine is more powerful when endowed with more space. This might not quite be the case for RASMs, but we can get close to this sort of monotonicity with a mild assumption.

Proposition 3.5. *Suppose κ, κ' are limit ordinals, $\kappa < \kappa'$, $A \leq^C_\kappa B$, and κ is definable over the structure $(\kappa'; \in)$. Then $A \leq^C_{\kappa'} B$.*

Proof. Let $\mathfrak{M}$ be an RASM witnessing $A \leq^C_\kappa B$. Define an RASM $\mathfrak{M}'$ with $\kappa(\mathfrak{M}') = \kappa'$ and $\sigma(\mathfrak{M}') = \sigma(\mathfrak{M})$. Since κ is definable over $(\kappa'; \in)$, we can relativise both $\phi^d_{\mathfrak{M}}$ and $\phi^\tau_{\mathfrak{M}}$ to κ' so that if $\ulcorner X \urcorner \in \sigma(\mathfrak{M})$ is an n-ary function symbol and $\alpha_1, \ldots, \alpha_n \in \kappa'$, then

- the default value of $X(\alpha_1, \ldots, \alpha_n)$
  - follows what is given by $\phi^d_{\mathfrak{M}}$ if $\alpha_1, \ldots, \alpha_n \in \kappa$, and
  - is set to 0 otherwise, and
- the next-stage value of $X(\alpha_1, \ldots, \alpha_n)$
  - follows the result of computation by $\tau_{\mathfrak{M}}$ if $\alpha_1, \ldots, \alpha_n \in \kappa$, and
  - is set to 0 otherwise.

Set $\phi^d_{\mathfrak{M}'}$ and $\phi^\tau_{\mathfrak{M}'}$ to be the relativised versions of $\phi^d_{\mathfrak{M}}$ and $\phi^\tau_{\mathfrak{M}}$ respectively. Then $\mathfrak{M}'$ witnesses $A \leq^C_{\kappa'} B$. □

3.2 Further Machine Modifications

The definability requirement in Proposition 3.5 hinders us from concluding the transitivity of $\leq^C$. One can read this as a sign that $\leq^C$, and by extension the $\leq^C_\kappa$s, is not quite the right notion of relative computability.

Moreover, it can be unsatisfactory to see that

$$\{A \subset \kappa : A \leq^C_\kappa B\} \text{ is countable}$$

for all limit ordinals κ and $B \subset \kappa$. An ideal relative computability relation uniformly definable in κ should have its predecessor sets grow along with κ. Towards such a relation, we shall introduce an augmentation to RASMs.

Definition 3.6. An *RASM with parameters* (henceforth *RASMP*)

$$\mathfrak{M} = (\kappa(\mathfrak{M}), \sigma(\mathfrak{M}), \phi^{\tau}_{\mathfrak{M}}, \phi^{d}_{\mathfrak{M}}, \vartheta_{\mathfrak{M}})$$

is defined like an RASM, except constants in $\sigma(\mathfrak{M})$ are allowed to evaluate to $(\{\alpha\}, 1)$ under $\vartheta_{\mathfrak{M}}$ for any $\alpha \in \kappa(\mathfrak{M})$. In this case, we call α a *parameter* of $\mathfrak{M}$.

Remark 3.7. Fix any limit ordinal κ. There are $|\kappa|$ many RASMPs $\mathfrak{M}$ satisfying $\kappa(\mathfrak{M}) = \kappa$. Also, this set of RASMP specifications is absolute for transitive models of ZFC containing κ (cf. Remark 2.26).

The definitions and properties so far associated with RASMs adapt perfectly to RASMPs. We will use the adapted definitions in the context of RASMPs without further clarifications.

Definition 3.8. For any limit ordinal κ and any two sets of ordinals A and B, B is *κ-computable with parameters from A* — denoted $B \leq^{P}_{\kappa} A$ — iff

- $A, B \subset \kappa$, and
- for some RASMP $\mathfrak{M}$ with $\kappa(\mathfrak{M}) = \kappa$, there is a terminating run of $\mathfrak{M}$ with input A and output B.

In this case, $\mathfrak{M}$ *witnesses* $B \leq^{P}_{\kappa} A$.

Definition 3.9. We say B is *computable with parameters from A*, or $B \leq^{P} A$, iff $B \leq^{P}_{\kappa} A$ for some limit ordinal κ.

Definition 3.10. Let A be a set of ordinals.

(1) A is *κ-computable with parameters* iff $A \leq^{P}_{\kappa} \emptyset$.
(2) A is *computable with parameters* iff A is κ-computable with parameters for some limit ordinal κ.

Let κ be a limit ordinal and A, B be two sets of ordinals. Like $\mathrm{SRuns}_{\kappa}(A, B)$ and $\mathrm{Runs}_{\kappa}(A, B)$, the sets

$$\mathrm{SRunsP}_{\kappa}(A, B) := \{(\mathfrak{M}, Sq) : \mathfrak{M} \text{ is an RASMP with } \kappa(\mathfrak{M}) = \kappa \text{ and } Sq \text{ is a short terminating run of } \mathfrak{M} \text{ with input } A \text{ and output } B\}$$

and

$$\mathrm{RunsP}_{\kappa}(A, B) := \{(\mathfrak{M}, Sq) : \mathfrak{M} \text{ is an RASMP with } \kappa(\mathfrak{M}) = \kappa \text{ and } Sq \text{ is a terminating run of } \mathfrak{M} \text{ with input } A \text{ and output } B\}$$

have absolute definitions for transitive models of ZFC containing A, B and κ.

Adapting Proposition 2.28 to RASMPs, in conjunction with Remark 3.7, the absoluteness of $\kappa^{<\omega}$, and the absoluteness of the definition of the $\mathrm{RunsP}_\kappa(A, B)$s, gives us that $\leq^P_\kappa$ is absolute for transitive models of ZFC containing κ. As a result, like $\leq^C$, $\leq^P$ is absolute for transitive models of ZFC with the same ordinals.

By Remark 3.7, the predecessor sets

$$\{A \subset \kappa : A \leq^P_\kappa B\}$$

are of size $|\kappa|$ for all limit ordinals κ and $B \subset \kappa$. Therefore, in the $\leq^P_\kappa$s we get a class of relative computability relations with predecessor sets that grow uniformly in κ.

A routine check confirms that the parameterised version of Proposition 3.4 holds with the same proof.

Proposition 3.11. *Let κ be a limit ordinal. Suppose $A \leq^P_\kappa B$ and $B \leq^P_\kappa C$. Then $A \leq^P_\kappa C$.*

If we substitute the parametrised counterparts of relative computability for the non-parametrised ones in Proposition 3.5, we can do away with the definability hypothesis.

Proposition 3.12. *Suppose κ, κ' are limit ordinals, $\kappa < \kappa'$ and $A \leq^P_\kappa B$. Then $A \leq^P_{\kappa'} B$.*

Proof. We modify the construction of $\mathfrak{M}'$ in the proof of Proposition 3.5. To ensure κ can be referred to in the relativisation process, we add two fresh constant symbols $\ulcorner c \urcorner$ and $\ulcorner d \urcorner$ to $\sigma(\mathfrak{M}')$ and set

$$\vartheta_{\mathfrak{M}'}(\ulcorner c \urcorner) := (\{\kappa\}, 1)$$
$$\vartheta_{\mathfrak{M}'}(\ulcorner d \urcorner) := (\kappa', 0).$$

As there is no way to formulate $\phi^d_{\mathfrak{M}'}$ such that $\ulcorner c \urcorner$ occurs in it, we choose $\phi^d_{\mathfrak{M}'}$ to be any universally valid sentence over $\sigma(\mathfrak{M}')$. For example, setting all return values to 0 works. Define $\tau_{\mathfrak{M}'}$ as follows. If "$d = 0$" in the current state, make use of c to set up the $\sigma(\mathfrak{M})$ portion of the next state according to the relativised version of $\phi^d_{\mathfrak{M}}$, then increment d. Otherwise, do nothing to d and modify the $\sigma(\mathfrak{M})$ portion of the next state according to $\phi^\tau_{\mathfrak{M}}$ relativised to c. The $\mathfrak{M}'$ defined thus witnesses $A \leq^P_{\kappa'} B$. □

Remark 3.13. The $\mathfrak{M}'$ defined based on $\mathfrak{M}$ in Proposition 3.12 simulates $\mathfrak{M}$ with finite overhead. That is, given any $A, B \subset \kappa$, if there is an α-length terminating run of $\mathfrak{M}$ with input A and output B, then for some $n < \omega$ there is an $(\alpha + n)$-length terminating run of $\mathfrak{M}'$ with input A and output B.

Notice that the construction in Proposition 3.12 of $\mathfrak{M}'$ from any $\mathfrak{M}$ witnessing $A \leq^P_\kappa B$ and any κ' can be made uniform. So let $\mathcal{G}$ be a definable class function mapping each such pair $(\mathfrak{M}, \kappa')$ to such a $\mathfrak{M}'$.

The next proposition is immediate from Propositions 3.11 and 3.12.

Proposition 3.14. *Suppose $A \leq^P B$ and $B \leq^P C$. Then $A \leq^P C$.*

Proposition 3.12, the transitivity of $\leq^P$, and the nice relationship between limit ordinals κ and the sizes of $\leq^P_\kappa$ predecessor sets, all point to $\leq^P$, $\leq^P_\kappa$ being the superior and more valid notion of relative computability. In fact, they also point to an RASMP being a better model of generalised algorithm than an RASM. For this reason, the rest of this subsection will focus on $\leq^P$, $\leq^P_\kappa$ and RASMPs.

We are now able to solidify our case for using point-wise limit inferiors, whenever they exist, to compute limit states of terminating runs which are not short. Let us first generalise the concepts of ordinal limits preservation and simple limit states (Definitions 2.35 and 2.36) to RASMPs and sequences of arbitrary lengths.

Definition 3.15. Let $\mathfrak{M}$ be an RASMP. Given $\ulcorner X \urcorner \in \sigma(\mathfrak{M})$ an n-ary function symbol and ϕ a *simple limit formula for* $(\ulcorner X \urcorner, \mathfrak{M})$, we say ϕ *preserves ordinal limits* iff interpreting $\ulcorner \in \urcorner$ as $\in$ and $\ulcorner R^{\ulcorner X \urcorner} \urcorner$ as $R^{\ulcorner X \urcorner}_{Sq}$,

$$(\lambda; \in, R^{\ulcorner X \urcorner}_{Sq}) \models \phi[x_1 \mapsto \alpha_1, \ldots, x_{n+1} \mapsto \alpha_{n+1}]$$

for all functions Sq from a limit ordinal into $S(\mathfrak{M})$, all limit ordinals

$$\lambda \geq max\{\kappa(\mathfrak{M}), dom(Sq)\}$$

and all $(\alpha_1, \ldots, \alpha_{n+1}) \in \kappa(\mathfrak{M})^{n+1}$ such that

$$\alpha_{n+1} = lim\{(\beta, \ulcorner X \urcorner^{Sq(\beta)}(\alpha_1, \ldots, \alpha_n)) : \beta < dom(Sq)\}.$$

Definition 3.16. Let $\mathfrak{M}$ be an RASMP and Sq be a function from a limit ordinal into $S(\mathfrak{M})$. A state $s \in S(\mathfrak{M})$ is a *simple limit of* Sq iff for each n-ary function symbol $\ulcorner X \urcorner \in \sigma(\mathfrak{M})$ there is a simple limit formula $\phi^{\ulcorner X \urcorner}_{lim}$ for $(\ulcorner X \urcorner, \mathfrak{M})$ such that, interpreting $\ulcorner \in \urcorner$ as $\in$ and $\ulcorner R^{\ulcorner X \urcorner} \urcorner$ as $R^{\ulcorner X \urcorner}_{Sq}$,

$$\ulcorner X \urcorner^s(\alpha_1, \ldots, \alpha_n) = \alpha_{n+1} \iff (\lambda; \in, R^{\ulcorner X \urcorner}_{Sq}) \models \phi[x_1 \mapsto \alpha_1, \ldots, x_{n+1} \mapsto \alpha_{n+1}]$$

whenever λ is a limit ordinal $\geq max\{\kappa(\mathfrak{M}), dom(Sq)\}$ and $(\alpha_1, \ldots, \alpha_{n+1}) \in \kappa(\mathfrak{M})^{n+1}$. In this case, we say $\{\phi^{\ulcorner X \urcorner}_{lim} : \ulcorner X \urcorner \in \sigma(\mathfrak{M})\}$ *witnesses s is a simple limit of* Sq.

Similarly, we can generalise Definitions 2.38 and 2.39.

Definition 3.17. Let $\mathfrak{M}$ be an RASMP. Then $\mathfrak{M}$ is *strongly closed under* a set Y iff for each n-ary function symbol $\ulcorner X \urcorner \in \sigma(\mathfrak{M})$ there is a simple limit formula $\phi^{\ulcorner X \urcorner}_{lim}$ for $(\ulcorner X \urcorner, \mathfrak{M})$, such that

$$Y = \{\phi^{\ulcorner X \urcorner}_{lim} : \ulcorner X \urcorner \in \sigma(\mathfrak{M})\}$$

and whenever Sq is a function from some limit ordinal into $S(\mathfrak{M})$, there is $s \in S(\mathfrak{M})$ witnessed by Y to be a simple limit of Sq.

Definition 3.18. Let $\mathfrak{M}$ be an RASMP. Then $\mathfrak{M}$ is *practically closed under* a set Y iff for each n-ary function symbol $\ulcorner X \urcorner \in \sigma(\mathfrak{M})$ there is a simple limit formula $\phi_{lim}^{\ulcorner X \urcorner}$ for $(\ulcorner X \urcorner, \mathfrak{M})$, such that

$$Y = \{\phi_{lim}^{\ulcorner X \urcorner} : \ulcorner X \urcorner \in \sigma(\mathfrak{M})\}$$

and whenever Sq is a function from some limit ordinal into $S(\mathfrak{M})$ satisfying

- $Sq(0) \in I(\mathfrak{M})$,
- $Sq(\alpha + 1) = \tau_{\mathfrak{M}}(Sq(\alpha))$ for all $\alpha < \delta$, and
- Y witnesses $Sq(\gamma)$ is a simple limit of $Sq \restriction \gamma$ for all limit $\gamma < \delta$,

there is $s \in S(\mathfrak{M})$ witnessed by Y to be a simple limit of Sq.

Note that Definitions 3.17 and 3.18 are almost word-for-word the same as Definitions 2.38 and 2.39. This only changes made are

- substituting "RASMP" for "RASM" and
- removing the requirement for δ to be $< \kappa(\mathfrak{M})$.

Proposition 3.19. *Proposition 2.37 holds with "RASMP" substituted for "RASM" and the requirement for δ to be $< \kappa(\mathfrak{M})$ removed.*

Proof. Any formula that works for all RASMs $\mathfrak{M}$ with $\ulcorner X \urcorner \in \sigma(\mathfrak{M})$ in the context of proving Proposition 2.37 (such a formula exists by the last sentence of the proof of Proposition 2.37) also works here. □

Proposition 3.20. *Suppose $\mathfrak{M}$ is an RASMP practically closed under Y. Then for some signature σ', as long as $\mathfrak{M}'$ is an RASMP satisfying*

- $\sigma(\mathfrak{M}') = \sigma'$ *and*
- $\kappa(\mathfrak{M}') = \kappa(\mathfrak{M})$,

and Z is a set of simple limit formulas preserving ordinal limits with $\mathfrak{M}'$ strongly closed under Z, there is an RASM $\mathfrak{M}^*$ such that

(1) $\sigma(\mathfrak{M}^*) = \sigma'$,
(2) $\kappa(\mathfrak{M}^*) = (2^{\kappa(\mathfrak{M})})^+$, *and*
(3) for all $A, B \subset \kappa(\mathfrak{M}^*)$, a terminating run of $(\mathfrak{M}, Y)$ with input $A \restriction \kappa(\mathfrak{M})$ and output $B \restriction \kappa(\mathfrak{M})$ exists iff a terminating run of $(\mathfrak{M}^*, Z)$ with input A and output $B \restriction \kappa(\mathfrak{M})$ exists.

Proof. Let us try to follow the proof of Proposition 2.41 with $\mathfrak{M}^\dagger := \mathcal{G}(\mathfrak{M}, (2^{\kappa(\mathfrak{M})})^+)$ specified in place of the arbitrary "$\mathfrak{M}$" given by said proposition. Note that

- Remark 2.44 remains true with "$\mathrm{RunsP}_\kappa(A, B)$" substituted for "$\mathrm{Runs}_\kappa(A, B)$" and "RASMP" substituted for "RASM", and

- (3) holds with $\mathfrak{M}^\dagger$ in place of $\mathfrak{M}^*$.

Also observe that by having $\mathfrak{M}^\dagger$ as our starting point and choosing σ' to be $\sigma(\mathfrak{M}^\dagger)$, the argument from the second paragraph onwards in the proof of Proposition 2.41 applies here if we just assume the hypotheses for $\mathfrak{M}'$ and Z to be as stated in this proposition. □

Propositions 3.19 and 3.20 allude to our method of computing limit states being utmost encompassing computability-wise amidst "locally definable" competition, regardless of sequence lengths.

Clearly, $\leq^P$ is reflexive, witnessed by the one-step copying of input tape contents onto the output tape. Hence $\leq^P$ is a preorder. We can quotient $\leq^P$ by a canonical equivalence relation to produce a proper-class-sized partial order analogous to the order of Turing degrees.

Definition 3.21. Let A, B be two sets of ordinals. We define two equivalence relations below.

(1) For any limit ordinal κ, we say $A \equiv^P_\kappa B$ iff $A \leq^P_\kappa B$ and $B \leq^P_\kappa A$.
(2) We say $A \equiv^P B$ iff $A \leq^P B$ and $B \leq^P A$.

Taking quotient of $\leq^P$ by $\equiv^P$ would result in a partial order. Denote this partial order by $(\mathcal{D}_P, \leq_{\mathcal{D}_P})$. Members of $\mathcal{D}_P$ are called *P-degrees*. We would like to study the structure of P-degrees under $\leq_{\mathcal{D}_P}$ the way we study the structure of Turing degrees under Turing reducibility (modulo Turing equivalence). But before that, it would serve us well to consult and compare with existing literature on models of generalised computability.

If classical recursion or computability theory is about the study of subsets of ω through Turing reducibility and degrees, then one can envision a natural extension of this study aiming at analogous notions of reducibility and degrees for subsets of ordinals larger than ω. Indeed, there have been a number of successful extensions, lumped under the subject of *higher recursion theory*. The theories of alpha recursion (or α-recursion) and E-recursion are the most developed among them, but both recursion theories in their typical presentations are built upon schemata akin to Gödel's model of real computation (see [1] and [3] respectively).

Emerging later into the scene is the theory of ordinal computability, started by Koepke in [10]. A comparison with α-recursion theory is given in [11]. We will drawn on parts of this comparative study to give an equivalent definition of $\leq^P$.

Before that, we need a brief introduction to select set-theoretic concepts, for set theory is unavoidable in the study of sets beyond ω.

3.3 Associations with Constructibility

Definition 3.22. We use $L[A]$ to denote the *constructible universe relative to a set* A, and L to denote $L[\emptyset]$. For each ordinal α, let $L_\alpha[A]$ respectively denote the α-th level $L[A]$, so $L_\alpha[\emptyset]$ is just L_α. The reader may refer to Chapter 13 of [9] for more details.

Definition 3.23. Let α be an ordinal and A be a set. Then $\mathbf{\Sigma_1}(L_\alpha[A])$ denotes the set of $X \subset \alpha$ with the following property:

> for some $n < \omega$ there are parameters $y_1, \ldots, y_n \in L_\alpha[A]$ and a Σ_1 formula ϕ with free variables $x_1, \ldots, x_n, z$ over the signature $\{\ulcorner\in\urcorner, \ulcorner A\urcorner\}$, such that
>
> $$X = \{\beta < \alpha : (L_\alpha[A]; \in, A) \models \phi[x_1 \mapsto y_1, \ldots, x_n \mapsto y_n, z \mapsto \beta]\}$$
>
> having $\ulcorner\in\urcorner$ interpreted as $\in$ and $\ulcorner A\urcorner$ as A.

In addition, $\mathbf{\Delta_1}(L_\alpha[A])$ denotes the set of $X \subset \alpha$ such that $X \in \mathbf{\Sigma_1}(L_\alpha[A])$ and $\alpha \setminus X \in \mathbf{\Sigma_1}(L_\alpha[A])$.

Fact 3.24. For every ordinal α and every set A, $\mathbf{\Delta_1}(L_\alpha[A]) \subset \mathbf{\Sigma_1}(L_\alpha[A]) \subset L[A]$.

Kripke-Platek set theory, or KP, is a weak fragment of ZF with close connections to α-recursion theory.

Definition 3.25. An ordinal α is *admissible* iff $L_\alpha \models \mathsf{KP}$.

An admissible ordinal has reasonably strong closure properties. For example, it is closed under ordinal addition, multiplication and exponentiation.

We require the concept of κ-computability with short runs for an upcoming lemma.

Definition 3.26. Let κ be a limit ordinal. We say B is *κ-computable with parameters from A with short runs*, or $B \leq_\kappa^{P,s} A$, if there is an RASMP $\mathfrak{M}$ witnessing $B \leq_\kappa^P A$ and a short terminating run of $\mathfrak{M}$ with input A and output B. Here, $\mathfrak{M}$ is said to witness $B \leq_\kappa^{P,s} A$.

Unsurprisingly, we have $\leq_\kappa^{P,s}$-analogues of Propositions 3.11 and 3.12.

Proposition 3.27. *Let κ be an admissible ordinal. Suppose $A \leq_\kappa^{P,s} B$ and $B \leq_\kappa^{P,s} C$. Then $A \leq_\kappa^{P,s} C$.*

Proof. As in the case of Proposition 3.11. the proof here is exactly the same as the proof of Proposition 3.4. We should additionally note that the termination run of $\mathfrak{M}^*$ — as in the proof of Proposition 3.4 — with input A and output C is still short because it combines two short runs and κ is closed under ordinal addition. □

Proposition 3.28. *Suppose κ, κ' are admissible ordinals, $\kappa < \kappa'$ and $A \leq_\kappa^{P,s} B$. Then $A \leq_{\kappa'}^{P,s} B$.*

Proof. Noting Remark 3.13, the proof here is exactly the same as the proof of Proposition 3.12. □

There is a natural follow-up to Definition 3.26.

Definition 3.29. Let A, B be sets of ordinals. We say B is *computable with parameters from A with short runs*, or $B \leq^{P,s} A$, iff $B \leq^{P,s}_{\kappa} A$ for some limit ordinal κ.

It so happens that $\leq^P$ and $\leq^{P,s}$ are fundamentally the same.

Proposition 3.30. *Let A, B be sets of ordinals. Then $B \leq^P A$ iff $B \leq^{P,s}_{\kappa} A$ for some regular ordinal κ.*

Proof. Clearly, $B \leq^P A$ holds if $B \leq^{P,s}_{\kappa} A$ for some regular ordinal κ. Now assume $B \leq^P A$. Then $B \leq^P_{\kappa'} A$ for some limit ordinal κ'. Let $\lambda := (2^{|\kappa'|})^+$, so that λ is a regular ordinal. By Proposition 3.12 and Remarks 2.44 and 3.13, $B \leq^{P,s}_{\lambda} A$. □

Proposition 3.31. *Let A, B be sets of ordinals. Then $B \leq^P A$ iff $B \leq^{P,s} A$.*

Proof. Again, the "if" portion is obvious, so assume $B \leq^P A$. By Proposition 3.30, $B \leq^{P,s}_{\kappa} A$ for some regular ordinal κ, which implies $B \leq^{P,s} A$. □

For a limit ordinal α, an α-machine according to Koepke (see [11]) is essentially a Turing machine with tape length α, oracle length up to α, and finitely many fixed ordinal parameters $< \alpha$. We can think of an input to an α-machine as a code $\langle X, O\rangle$ of a pair of sets, where X represents the non-oracle component of the input and O represents the oracle. At limit steps the limit inferiors of previous head and state positions, along with the previous contents of each cell, are taken. The machine *halts* iff it terminates in $< \alpha$ many steps.

Proposition 3.32. *Let α be a limit ordinal. An α-machine M can be simulated by an RASMP $\mathfrak{M}$ in the sense that for any $A, B \subset \alpha$, M halts on input A with output B iff $\mathfrak{M}$ has a short terminating run with input A and output B.*

Proof. It is not difficult to check that modifying Example 2.29 by substituting α for ω, and adding oracle reads based on how the oracle is coded into the input (through e.g. adding a third tape), works out just fine. The finitely many ordinal parameters of an α-machine can be ported wholesale to an RASMP as the latter's ordinal parameters; the way an RASMP determines the limit states of a computation also perfectly mimicks how limit configurations of an α-machine are defined. Moreover, an α-machine halting on input A with output B is equivalent to the RASMP simulating it having a short terminating run with input A and output B. □

Definition 3.33 (Koepke, [11])**.** Let α be a limit ordinal. A set $B \subset \alpha$ is *α-computable in $A \subset \alpha$*, denoted $B \preceq_\alpha A$, iff there is an α-machine M for which

(1) M halts on input $\langle \{\beta\}, A\rangle$ for all $\beta < \alpha$, and
(2) $B = \{\beta < \alpha : M(\langle \{\beta\}, A\rangle) = 1\}$.

In this case, *M witnesses B is α-computable in A.*

Fact 3.34 (Koepke, [11]). Let α be an admissible ordinal. Then B is α-computable in A iff $B \in \mathbf{\Delta_1}(L_\alpha[A])$.

Recall the following definition referenced in Subsection 1.2.

Definition 3.35. Given sets of ordinals A and B, we say $B \preceq_A A$ iff there exists an admissible ordinal α such that $B \preceq_\alpha A$.

Remark 3.36. It follows rather directly from Fact 3.34 that whenever A and B are sets of ordinals, $B \preceq_A A$ iff $B \in L[A]$. In other words, $\preceq_A$ coincides with the relative constructibility relation.

Lemma 3.37. *Let $B \leq_\kappa^{P,s} A$ for some regular ordinal κ. Then $B \in \mathbf{\Delta_1}(L_\kappa[A])$, so B is α-computable in A by Fact 3.34.*

Proof. Let $\mathfrak{M}$ be an RASMP with parameter set K witnessing $B \leq_\kappa^{P,s} A$, and Sq be a short terminating run of $\mathfrak{M}$ with input A and output B. Suppose we want to find out whether some $\beta < \kappa$ is a member of B. To do so, we draw up a dependency tree of terms of which interpretations among members of $ran(Sq)$ are needed to decide if $\beta \in B$. This tree is finitely branching by the bounded exploration postulate, and it has number of levels no more than $dom(Sq)$. By the regularity of κ, it has size $< \kappa$.

Now, the statement $x \in B$ is semantically equivalent over V to a Σ_1 formula with free variable x and parameters among $K \cup \{dom(Sq)\}$ saying that such a (unique) tree of terms and interpretations exist. The construction of said tree, due to its simplicity, can be carried out correctly in $L[A]$ through transfinite recursion. Moreover, $L[A]$ recognises the tree has size $< \kappa$, so indeed $B \in \mathbf{\Sigma_1}(L_\kappa[A])$.

We can define an RASMP $\mathfrak{M}'$ that copies what $\mathfrak{M}$ does on each input until $\mathfrak{M}$ terminates, and flips every bit of the output tape thereafter. Now $\mathfrak{M}'$ witnesses $\kappa \setminus B \leq_\kappa^{P,s} A$, so by the argument in the preceding paragraph we have $\kappa \setminus B \in \mathbf{\Sigma_1}(L_\kappa[A])$. □

Definition 3.38 (Koepke, [11])**.** Let α be a limit ordinal. A set $B \subset \alpha$ is *α-computably enumerable in $A \subset \alpha$* iff there is an α-machine M for which

$$B = \{\beta < \alpha : M \text{ halts on input } \langle\{\beta\}, A\rangle\}.$$

In this case, *M witnesses B is α-computably enumerable in A*.

Fact 3.39 (Koepke, [11]). Let α be an admissible ordinal. Then B is α-computably enumerable in A iff $B \in \mathbf{\Sigma_1}(L_\alpha[A])$.

Lemma 3.40. *Let $B \in \mathbf{\Sigma_1}(L_\kappa[A])$ for some admissible ordinal κ. Then $B \leq_\kappa^P A$.*

Proof. By Fact 3.39, B is κ-computably enumerable in A, as witnessed by some κ-machine M. Let $\mathfrak{M}$ be an RASMP simulating M as per Proposition 3.32. We define an RASMP $\mathfrak{M}'$ that, when given input A, dovetails and runs M through $\mathfrak{M}$ on each input in

$$\{\langle\{\beta\}, A\rangle : \beta < \kappa\}$$

for arbitrarily many steps $< \kappa$, in order to determine if M halts on said input. In the end, $\mathfrak{M}'$ should consolidate the findings and output them as a subset of κ.

More formally, add to $\sigma(\mathfrak{M})$ fresh constant symbols $\ulcorner c_0 \urcorner$, $\ulcorner c_1 \urcorner$, $\ulcorner c_2 \urcorner$, $\ulcorner d \urcorner$ and a fresh unary function symbol $\ulcorner R \urcorner$, and have the result be $\sigma(\mathfrak{M}')$. Set all constants to 0 and R to be empty by default. We are left to describe the transition function of $\mathfrak{M}'$, which we shall do by cases. In our description, I denotes the input to $\mathfrak{M}'$.

Case 1: "$c_0 = d = 0$". We increment both c_0 and d and set the input tape to display $\langle \emptyset, I \rangle$.

Case 2: "$c_0 \neq 0 \wedge d \neq 0$". We do the dovetailing. In other words, for each β satisfying "$\neg R(\beta) \wedge \beta \leq c_0$", we use $\mathfrak{M}$ to simulate M on input $\langle \{\beta\}, I \rangle$ for up to c_0 steps, using c_1 and c_2 to keep track of the counts. At the end of every step in simulation, we check if $\mathfrak{M}$ terminates. If so, we set $R(\beta)$ to `true` and move on to simulate M on input $\langle \{\beta + 1\}, I \rangle$, if $\beta \neq c_0$. At the end of all the simulations and possible updates to R, set c_1 and c_2 to 0 and increment c_0. The constant d is left unchanged.

Case 3: "$c_0 = 0 \wedge d \neq 0$". We copy the contents of R to the output tape, before terminating.

Case 4: "$c_0 \neq 0 \wedge d = 0$". Just terminate.

Every terminating run of $\mathfrak{M}'$ is of length $\kappa + 2$, and if its input is I, then its output will be the set

$$\{\beta < \alpha : M \text{ halts on input } \langle \{\beta\}, I \rangle\}.$$

This is because

- no state where Case 4 applies is reachable from an initial state, so a run can only terminate from an initial state as a consequence of Case 3, and
- c_0 first returns to 0 in a state other than an initial state after exactly $\kappa + 1$ many steps, at which point M would have been simulated on every input in

$$\{\langle \{\beta\}, I \rangle : \beta < \kappa\}$$

for up to arbitrarily many steps $< \kappa$, with all eventual haltings recorded in R.

In particular, there is a terminating run of $\mathfrak{M}'$ with input A and output B, whence $B \leq^P A$. □

Lemmas 3.37 and 3.40 reflect the difference in computability between considering all terminating runs of RASMPs and considering only short terminating runs, for regular ordinal tape lengths. In tandem, the lemmas also give us a nice characterisation of the reducibility relation $\leq^P$.

Lemma 3.41. *Let A and B be two sets of ordinals. Then $B \leq^P A$ iff $B \in L[A]$.*

Proof. Assume $B \in L[A]$. Then $B \in L_\kappa[A]$ for some admissible ordinal κ. But $L_\kappa[A] \subset \mathbf{\Sigma_1}(L_\kappa[A])$, so by Lemma 3.40, $B \leq^P A$.

Next, assume $B \leq^P A$. Then by Proposition 3.30, $B \leq^{P,s}_\kappa A$ for some regular ordinal κ. Lemma 3.37 tells us $B \in \mathbf{\Delta_1}(L_\kappa[A])$, but $\mathbf{\Delta_1}(L_\kappa[A]) \subset L[A]$, so we are done. □

Corollary 3.42. *A set of ordinals B is computable with parameters iff $B \in L$.*

Proof. Apply Lemma 3.41 with $\emptyset$ in place of A. □

Corollary 3.43. *Let A and B be two sets of ordinals. Then $B \leq^P A$ iff $B \preceq_A A$.*

Proof. Immediate by Remark 3.36. □

4 Discussion and Questions

The following are two basic facts about relative constructibility.

Fact 4.1. Let A, B be sets of ordinals. Then $B \in L[A]$ iff $L[B] \subset L[A]$.

Fact 4.2. Let X be any set. Then there exists a set of ordinals A such that $L[X] = L[A]$.

From a second-order viewpoint of V, we can consider the multiverse of constructible worlds relative to sets in V, i.e.

$$\mathbf{M}_C := \{L[X] : X \in V\}.$$

By Lemma 3.41 and Facts 4.1 and 4.2,

$$(\mathcal{D}_P, \leq_{\mathcal{D}_P}) \cong (\mathbf{M}_C, \subset).$$

Philosophically, this seems to indicate that the amount of computing power a set holds is equivalent to how it can extend a particular model of set theory through transfinite recursion. This is far from an original take, for if we take the notion of computability to an extreme, we can argue that any construction via transfinite recursion is a generalised algorithm. What might be slightly more surprising is, by only allowing transfinite recursion that involves very simple formulas —

- either formulas conservatively extending transition functions of classical Turing machines to incorporate limit stage computations,
- or formulas satisfying a version of Gurevich's bounded exploration postulate

— we can still achieve the same reach. Indeed, the fact that the relative computability relations arising from

- a "low-level" machine model of generalised computation such as ordinal Turing machines,

- a "high-level" machine model of generalised computation such as RASMPs, and
- an algebraic model of generalised computation such as relativised constructibility

happen to coincide, is suggestive of a plausible generalised-computability counterpart to the Church-Turing thesis.

Just as practical implementations of finitary ASMs do not aim to model algorithmic intuition beyond the limits of Church-Turing thesis, RASMPs should not allow for algorithms that are non-constructible. We can view RASMPs as a paradigm for easy high-level descriptions of algorithms on arbitrary sets, one that gives the user enough structure to organise their ideas, and serves as guardrails to rule out clearly non-algorithm descriptions.

It is consistent that $(\mathcal{D}_P, \leq_{\mathcal{D}_P})$ is highly non-trivial; in fact, this level of non-triviality is implied by suitable large cardinals.

Fact 4.3.

(1) ZFC + "$\exists$ a transitive model of ZFC" $\vdash$ "$\exists$ a transitive model of ZFC + '$V \neq L[A]$ for all $A \in V$'".
(2) ZFC + "$\exists$ a strongly compact cardinal" $\vdash$ "$V \neq L[A]$ for all $A \in V$".

In particular, (2) of Fact 4.3 is implied by Exercise 20.2 of [9]. Note that $V \neq L[A]$ for all $A \in V$ is equivalent to $(\mathcal{D}_P, \leq_{\mathcal{D}_P})$ having no maximal element, so Fact 4.3 actually gives us

(E1) ZFC + "$\exists$ a transitive model of ZFC" $\vdash$ "$\exists$ a transitive model of ZFC + '$(\mathcal{D}_P, \leq_{\mathcal{D}_P})$ has no maximal element'",
(E2) ZFC + "$\exists$ a strongly compact cardinal" $\vdash$ "$(\mathcal{D}_P, \leq_{\mathcal{D}_P})$ has no maximal element".

If $(\mathcal{D}_P, \leq_{\mathcal{D}_P})$ has no maximal element, then by applications of iterated forcing over L, one can uncover a rich lattice structure of $(\mathcal{D}_P, \leq_{\mathcal{D}_P})$.

Let us now turn our attention to the bounded relations $\leq_\alpha^{P,s}$ and $\leq_\alpha^P$. Let $\leq_\alpha$, $\leq_{w\alpha}$ and $\preceq_\alpha$ respectively denote the relations "α-recursive in", "weakly α-recursive in" and "α-computable in". The first two relations are well-studied in the once-bustling field of α-recursion theory (see e.g. Part C of [13]), whereas the last relation comes from the emerging field of α-computability theory. It is known (Section 3 of [11]) that

$$\leq_\alpha \subset \leq_{w\alpha} \subset \preceq_\alpha$$

for any admissible ordinal α, and the inclusions cannot be reversed in general. By Fact 3.39 and Lemma 3.40, we have

$$\leq_\alpha \subset \leq_{w\alpha} \subset \preceq_\alpha \subset \leq_\alpha^P \tag{4.1}$$

for admissible ordinals α. The last inclusion cannot be reversed in general because $\leq_\alpha^P$ is transitive for all admissible α but $\preceq_\alpha$ is not. On the other hand, it is clear that $\leq_\alpha^{P,s} \subset \leq_\alpha^P$ for all limit α, and the reverse is not true due to Lemmas 3.37 and 3.40. Further, for admissible α, $\leq_{w\alpha} \neq \leq_\alpha^{P,s} \neq \preceq_\alpha$ in general because $\leq_\alpha^{P,s}$ is transitive unlike the other two relations. We know $\leq_\alpha^{P,s} \neq \leq_\alpha$ in general because $\leq_\alpha^{P,s}$ is upward absolute in the sense of Proposition 3.12, but $\leq_\alpha$ is not by Theodore Slaman's answer in [15].

Table 3: Comparing two of our relative computability relations with more standard ones in the study generalised computability, at arbitrary admissible ordinals α.

Property \ Relation	$\leq_\alpha$	$\preceq_\alpha$	$\leq_\alpha^P$	$\leq_\alpha^{P,s}$
Oracle-analogue?	✓	✓	✗	✗
Transitive?	✓	✗	✓	✓
Upward consistent?	✗	✓	✓	✓
Appears in. . .	α-recursion	α-computability	Def. 3.8	Def. 3.26

Question 4.4. *Is the set*

$$\{\leq_\alpha, \leq_{w\alpha}, \preceq_\alpha, \leq_\alpha^{P,s}\}$$

linearly ordered by inclusion, for all admissible α?

5 References

[1] Gaisi Takeuti. "On the recursive functions of ordinal numbers". In: *Journal of the Mathematical Society of Japan* 12.2 (1960), pp. 119–128.

[2] Dana S Scott and Christopher Strachey. *Toward a mathematical semantics for computer languages*. Vol. 1. Oxford University Computing Laboratory, Programming Research Group Oxford, 1971.

[3] Dag Normann. "Set recursion". In: *Studies in Logic and the Foundations of Mathematics*. Vol. 94. Elsevier, 1978, pp. 303–320.

[4] Gordon D Plotkin. "A structural approach to operational semantics". In: *Tech. Rep. DAIMI FN-19, Computer Science Department, Aarhus University* (1981).

[5] Bertrand Meyer, Prentice Hall, and Bertrand Meyer. "Axiomatic semantics". In: *Introduction to the Theory of Programming Languages, Prentice-Hall, Inc, Englewood Cliffs, NJ* (1990).

[6] Yuri Gurevich. "Sequential abstract-state machines capture sequential algorithms". In: *ACM Transactions on Computational Logic (TOCL)* 1.1 (2000), pp. 77–111.

[7] Joel David Hamkins and Andy Lewis. “Infinite time Turing machines”. In: *The Journal of Symbolic Logic* 65.2 (2000), pp. 567–604.

[8] Andreas Blass and Yuri Gurevich. “Abstract state machines capture parallel algorithms”. In: *ACM Transactions on Computational Logic (TOCL)* 4.4 (2003), pp. 578–651.

[9] Thomas Jech. *Set theory: The third millennium edition, revised and expanded*. Springer, 2003.

[10] Peter Koepke. “Turing computations on ordinals”. In: *Bulletin of Symbolic Logic* 11.3 (2005), pp. 377–397.

[11] Peter Koepke and Martin Koerwien. “Ordinal computations”. In: *Mathematical Structures in Computer Science* 16.5 (2006), pp. 867–884.

[12] GE Sacks. “E-recursive intuitions”. In: *Effective Mathematics of the Uncountable* (2013), pp. 135–149.

[13] Gerald E Sacks. *Higher recursion theory*. Vol. 2. Cambridge University Press, 2017.

[14] Desmond Lau. *Forcing with Language Fragments, Extending Namba Forcing, and Models of Theories with Constraints in Interpretation*. 2024. arXiv: 2402.01213 [math.LO].

[15] Zoorado (https://mathoverflow.net/users/29231/zoorado). *Does $A \leq_\alpha B$ imply $A \leq_\beta B$ for admissible ordinals α [and] β?* MathOverflow. (version: 2024-02-17). eprint: https://mathoverflow.net/q/464328. URL: https://mathoverflow.net/q/464328.