2

On the computational significance of consistency: from ordinary to generalized recursion theory

June 30, 1994[1]

ABSTRACT. The logical character of a transition-based notion of computation is explored in relation to Turing's ordinal logics and certain subsequent developments. The addition of Π^0_1 consistency statements in ordinal logics is portrayed as an attempt to extract fast programs computing a theory's provably recursive functions, and is noted to require (for the attempt to succeed) ordinal notations coding computations extraneous to the ordinal named. This motivates a richer semantic conception of computation based on transitions that include input/output transitions as special cases. In this regard, the equivalence underlying the construction of the next admissible set from trees is linked with the notion of a bisimulation, subsuming both input/output equivalence of programs and equivalence between ordinal notations naming the same ordinals. Complications introduced by non-well-foundedness (reflecting, for instance, non-termination of computational systems) are investigated, including $\Sigma^1_1 - \Pi^1_1$ pre-quotients of r.e. transition predicates. Π^1_1 paths through the Church-Kleene system $\mathcal{O}$ of ordinal notations are employed to analyze this leap from syntactic Π^0_1 consistency statements to a semantic Σ^1_1 notion of a computational process.

The starting point of various generalizations of ordinary recursion theory is the statement

$$\{e\}(x) \simeq y \tag{1}$$

asserting that the program named e returns output y, given input x. In view of Church's thesis, however, it is not clear what relevance a non-r.e. set of input/output triples e, x, y, can have for mechanical computation. Accordingly, the object of generalization in the present paper is not so much the computational power behind (1), but rather the form of (1) itself, which is

[1] *Added, February 11, 1997.* This paper was the basis of a talk at a conference on Proof Theory, Provability Logic and Computation in Berne (March 1994), and was submitted for refereeing for a special issue of the *Annals of Pure and Applied Logic* consisting of selected papers from the conference. The two referees reviewing the paper recommended publication on the condition that it be "more or less completely" revised. The style in which it was written was deemed unacceptable for a mathematical journal. But as the thrust of the work was largely philosophical and programmatic, I feared the paper would become pointless if I were to carry out the referees' advice to "concentrate on the mathematical contents." Consequently, I withdrew the paper, hoping to return to it at some later point. Some two and a half years later, I have been unable to carry out a single revision and am as uncertain as ever of attracting for it the attention of a sufficiently indulgent reader. But as I still believe the point of the paper is worth making, I include it here for the record.

generalized into a possibly non-deterministic and non-terminal "transition," collectively forming a set that is r.e. under some syntactic presentation. An abstract semantic analysis of that syntax then takes us beyond the realm of the r.e., and through both syntactic and semantic notions of consistency.

More concretely, consider transitions of the form $(s, d) \rightarrow_e (s', d')$ where the transition predicate $\rightarrow_e$ relates pairs (s, d) and (s', d') of "control states" s, s' and "data configurations" d, d'. The input/output relation (1) then assumes the special form $(s_0, x) \rightarrow_e (\surd, y)$, where s_0 and $\surd$ are fixed initial and terminal states, respectively. Not all transitions need end at some $(\surd, y)$, however, which is to say that reducing a program to the function it computes is inadequate, if intermediate states (s', d') (with s' not terminal, nor initial) in the computation of that function are also of interest. To say that a program is its name (or text, Gödel numbered) e may ultimately be the safest answer to the question *what is a program?* A more "semantic" characterization would, on the other hand, abstract away syntactic details and provide an answer on the basis simply of the transitions the program can make. That, at least, is what the present paper investigates, employing the notion of a bisimulation (Park [**19**]) to obtain an equivalence on states based solely on transitions between them, and eventually pointing to a set-theoretic notion of computation. Towards that end, the transition $(s, d) \rightarrow_e (s', d')$ is recast as $s \overset{d,d'}{\rightarrow}_e s'$, and then further to

$$e \overset{d,d'}{\rightsquigarrow} e' , \tag{2}$$

where a program $\hat{e}$ is identified with the initial state of the transition predicate $\rightarrow_{\hat{e}}$, which is, in turn, embedded in some monster transition predicate $\rightsquigarrow$. Shifting the basic picture from input/output statements (1) to transitions (2) is, it is argued, far from a triviality, considering the partiality of computation.

Apart from some faint hope in gaining a "semantic" understanding, the move from (1) to (2) can be motivated by the "no-nonsense" problem of isolating programs that run quickly — or more precisely, the problem of

$(*)$ extracting fast programs computing a theory's provably recursive functions.

A "path" to generalized recursion theory is paved by an explosion in logical complexity from what can be viewed as an early attempt (i.e., Turing [**21**]) to solve $(*)$ — an attempt that, upon reflection, leads to a far richer (set-theoretic) notion of computation. Briefly, the point is that the notions of equivalence on (i) ordinal notations from Turing [**21**], (ii) hyperprojective trees in the Barwise, Gandy and Moschovakis (1971) construction of the next admissible set,[2] and (iii) accessible pointed graphs in Aczel [**1**]'s relative consistency argument for the Anti-Foundation Axiom are all bisimilarity (i.e., the largest bisimulation). Thus, the well-known table

[2] The presentation of the next admissible in Barwise [**4**] is different, appealing instead to the theory of constructible sets.

ordinary recursion th	**generalized recursion th**
number ($\in \omega$)	set ($\in L_\alpha$)
Π^0_1	co-α-r.e

extends to

ordinary recursion th	**generalized recursion th**
syntax (code e)	semantics (set of transitions)
speed-up	substitutivity
consistency	bisimilarity

complementing the move from (1) to (2). The move suggests a step beyond not only equational reasoning, as advocated (for example) in Baeten and Weijland [**2**], but also first-order logic and, further than that, ω-logic (styled respectively as Σ^0_1- and Π^1_1-logic in Girard [**11**]). Model-theoretic arguments concerning bisimilarity that involve (respectively) saturation (a tool of first-order logic) and omitting types (a property of ω-logic) illustrate the point. The basic fact is that the Π^1_1 bound on bisimilarity relative to well-founded Δ^1_1 transition predicates (implicit in Barwise, Gandy and Moschovakis [**3**]) is exceeded by bisimilarity relative to a non-well-founded logspace computable predicate.

Of course, the idea of applying generalized recursion theory to programming languages and processes is far from new. Consider, for example, recent papers of Moschovakis (e.g., Moschovakis [**18**]). The "path" described in the present paper, however, goes back further to ancient work of Alan Turing's (Turing [**21**]), analyzed in relation to more modern developments in Feferman [**6, 7**], on which, in turn, a good deal of the present work builds (motivating the concept of relative computability by analyses based on different transition predicates).

1. Background: between absoluteness and open ended-ness

In a conference held in 1946, Gödel described recursion theory as being singularly successful "in giving an absolute definition of an interesting epistemological notion, i.e., one not depending on the formalism chosen" (Gödel [**12**], p. 150). Whereas the recursion-theoretic notion of computation is, according to Church's thesis, stable (capturing, as is claimed, all functions that can be computed mechanically), formal notions of deduction are defined "relative" to a formal theory that is (by contrast) subject to (Gödel) incompleteness. Curiously, the same functional reduction of computation in Church's thesis has been employed as the basis for a proof-theoretic approach to computation, which, as will be argued shortly, can only imperfectly approximate the "absolute" notion of a program.

Given an input/output specification $S(x, y)$, it is possible, under suitable circumstances, to extract from a constructive proof of

$$(\forall \text{ input } x)\ \ (\exists \text{ output } y) \qquad S(x, y)$$

a program e *realizing* (Kleene [**14**]) the theorem, where

$$\begin{aligned} & e \text{ realizes } (\forall x)(\exists y) S(x,y) \quad \text{implies} \\ (3) \qquad & (\forall x)\ \{e\}(x)\!\downarrow \,\&\, S(x,\{e\}(x)) \ . \end{aligned}$$

The correspondence between proofs and programs is developed further in the *Curry-Howard isomorphism* (e.g., Girard, Lafont and Taylor [**10**]) between a typed λ-calculus and a natural deduction system, under which computation becomes normalization (roughly, the unwinding of lemmas in a proof). Underlying the isomorphism is a theorem stating

$$\begin{aligned} \text{(weak) normalization} \quad : \quad & (\forall \text{ term } t) \ \ (\exists \text{ reduction } r) \\ (4) \qquad & r \text{ takes } t \text{ to a normal form} \ . \end{aligned}$$

Now, although deduction may not be "absolute", the proofs of (3) and (4) can be formalized, which is to say that every program e provably extracted in some theory T is provably total in T —

$$T \ \vdash \ \mathsf{Tot}(e) \ ,$$

where $\mathsf{Tot}(e)$ is $(\forall x)(\exists y)\mathrm{T}_1(e,x,y)$ and T_1 is (the appropriate form of) Kleene's computation predicate.

1.1. Provably recursive functions versus provably recursive programs. The family of programs e that are *provably total* may look nice functionally, but is flawed upon closer inspection. More precisely, if the *provably recursive functions of* T are defined (following Kreisel [**15**]) as members of the set

$$\mathrm{Fcn}(T) \ := \ \{\{e\} \mid T \vdash \mathsf{Tot}(e)\} \ ,$$

then

Proposition 1. *If T contains primitive recursive arithmetic then* $\mathrm{Fcn}(T)$ *forms a (time-)complexity class — i.e.,*

$$(5) \qquad T \vdash \mathsf{Tot}(e) \text{ and } t_{e'} < t_e \text{ a.e. } \ \text{imply} \ \ \{e'\} \in \mathrm{Fcn}(T)$$

where t_e counts the number of steps in the computation given by the Kleene T_1*-predicate.*

On the other hand,

Proposition 2. *The conclusion of (5) can be strengthened to $T \vdash \mathsf{Tot}(e')$ (with or without the assumption $\{e\} = \{e'\}$) if and only if T proves all true Π^0_1 sentences.*

Proposition 1 and the sufficiency in Proposition 2 follow from $t_e(x)$ being primitive recursively related to $(\mu y)\mathrm{T}_1(e,x,y)$, making the clause '$t_{e'} < t_e$ a.e.' Π^0_1. As for the necessity in Proposition 2, given a true Π^0_1 sentence

$(\forall x)R(x)$ (where R is a primitive recursive predicate), define

$$\begin{array}{lcl} e & : & \text{on input } n, \text{ check } (\forall m \leq n+1)\ R(m); \\ & & \text{return } 0 \text{ (in any case).} \end{array}$$

$$\begin{array}{lcl} e' & : & \text{on input } n, \text{ check } R(n); \\ & & \text{if so return } 0, \text{ else loop.} \end{array}$$

Then $t_{e'} < t_e$, $\{e'\} = \{e\}$, $T \vdash \mathsf{Tot}(e)$, and if $T \vdash \mathsf{Tot}(e')$, then $T \vdash (\forall x)R(x)$. A similar argument occurs in Turing's completeness theorem below, but first, we pause to ask if such pathological programs can be avoided. Blum's speed-up theorem stating that some functions have no optimal programs would seem to suggest not, as discussed at further length in Fernando [**8**] (where the propositions above are drawn from, and where normalization for a calculus under a Curry-Howard isomorphism is cited as an example of such a function). To round off the relationship between Π^0_1 sentences and programs of bounded complexity, Kreisel [**15**] has shown that

$$\mathrm{Fcn}(T) \quad = \quad \mathrm{Fcn}(T + \{\textit{all true } \Pi^0_1 \textit{ sentences}\}) \ .$$

That is, overcoming Π^0_1-incompleteness is equivalent to extracting efficient programs, rather than new, more complex functions. (Normalization implies consistency but not conversely.) The restriction to programs of bounded complexity reduces the Π^0_2-completeness of totality (and input/output equivalence) to Π^0_1. A roller-coaster ride through logical complexity will, however, soon take us far beyond Π^0_2.

1.2. Turing through programs of bounded complexity. The remainder of this section draws considerably from Feferman [**6**], which describes Turing's attack on the incompleteness of T by repeatedly adding (for instance) consistency statements transfinitely. To carry this out, the theories must be *presented* and (in particular) the ordinals $0, 1, \ldots, \omega, \omega+1, \ldots$ must be *named*. Such a system of ordinal notations can be built, relative to a standard enumeration of the partial recursive functions $\{e\}$, by defining an r.e. binary relation $\prec$ on ω by

$$\frac{a > 0}{a \prec 2^a} \qquad\qquad \frac{\{e\}(n) \text{ is defined and } > 0}{\{e\}(n) \prec 3 \cdot 5^e}$$

$$\frac{a \prec b \qquad b \prec c}{a \prec c} \ ,$$

and assigning ordinal *heights*

$$|a| \quad = \quad \{|b| \mid b \prec a\}$$

whenever $\prec$ is well-founded on $\{b \mid b \prec a\}$.[3] In particular, $|\cdot|$ is defined on the set $\mathcal{O} \subset \omega$ of *Church-Kleene ordinal notations* generated by

$$\frac{}{1 \in \mathcal{O}} \qquad\qquad \frac{a \in \mathcal{O}}{2^a \in \mathcal{O}}$$

$$\frac{\{e\}(n) \in \mathcal{O} \text{ and } \{e\}(n) \prec \{e\}(n+1) \text{ for all } n}{3 \cdot 5^e \in \mathcal{O}} \ .$$

The *recursive ordinals* are the ordinals $|a|$ *named* by $a \in \mathcal{O}$, and collectively form the least non-recursive ordinal ω_1^{CK}. The progression $\{T_a\}_{a \in \mathcal{O}}$ of theories

$$\begin{aligned} T_1 &= T \\ T_{2^a} &= T_a + \mathsf{Con}[T_a] \\ T_{3 \cdot 5^e} &= \bigcup_{n<\omega} T_{\{e\}(n)} \end{aligned}$$

then constitutes an ordinal logic.

An ordinal logic provides, in Turing's view, a very rough and admittedly oversimplified model of mathematical reasoning (see Feferman [**6**], p. 129). A proof of a mathematical assertion A consists of an "intuitive" step (of complexity Π^1_1) in which an ordinal notation $a \in \mathcal{O}$ is chosen, and a "mechanical" step (of complexity Σ^0_1) establishing that $T_a \vdash A$. It turns out (perhaps not too surprisingly) that the burden of proving a true Π^0_1 sentence can be shoved over to the choice of a.

Theorem (Turing). *For every true Π^0_1 sentence A, there is an $a \in \mathcal{O}$ naming ω such that*

$$T_a + \mathsf{Con}[T_a] \quad \vdash \quad A \ .$$

Proof sketch. For A equal to $(\forall x)R(x)$ with R primitive recursive, take $a = 3 \cdot 5^e$ where e satisfies (by the second recursion theorem)

$$\{e\}(n) \;=\; \begin{cases} n_{\mathcal{O}} & \text{if } (\forall x < n)\ R(x) \\ Suc_{\prec}(3 \cdot 5^e) & \text{otherwise} \end{cases}$$

with $0_{\mathcal{O}} = 1$, $(n+1)_{\mathcal{O}} = 2^{n_{\mathcal{O}}}$, and $Suc_{\prec}(a) = 2^a$. $\dashv$

Commenting on the proof of the theorem, Turing [**21**] had this to say.

> The completeness theorem as usual is of no value. Although it shows, for instance, that it is possible to prove Fermat's last theorem ... (if it is true), yet the truth of the theorem would really be assumed by taking a certain formula as an ordinal formula [ordinal notation].

[3] The assumption of well-foundedness can be dropped so long as $|a|$ is allowed to be a non-well-founded set, as in the set theory of Aczel [**1**]; see subsection 2.2 below.

With an eye on a more general notion of ordinal logics T_a not necessarily based on the addition of consistency statements, Turing had further established the incompatibility of completeness with *invariance* under the choice of an ordinal notation: the property that whenever $|a| = |a'|$, $T_a \equiv T_{a'}$. Referring the interested reader to Feferman [**6**] for details on Turing's work, let us turn instead to a fundamental negative result concerning ordinal logics obtained after Turing, two ideas behind which will be of use to us later, namely

> Σ^1_1 *boundedness*: the height of every Σ^1_1 well-order is a recursive ordinal.
>
> *Non-standard ordinal notations*: a Σ^1_1 set $\mathcal{O}^* \supset \mathcal{O}$ indistinguishable from $\mathcal{O}$ up to Δ^1_1 sets can be defined by requiring that for such $a \in \mathcal{O}^*$, there are no Δ^1_1 $\prec$-descending sequences from a. More precisely, $\mathcal{O}^*$ is the intersection of all Δ^1_1 sets X of natural numbers that contain (i) 1, (ii) 2^a whenever $a \in X$, and (iii) $3 \cdot 5^e$ whenever $(\forall n)\ \{e\}(n) \in X$ & $\{e\}(n) \prec \{e\}(n+1)$.

Defining a *path Z through $\mathcal{O}$* as a subset Z of $\mathcal{O}$ linearly ordered by $\prec$ such that for every $a \in \mathcal{O}$ there is a unique $b \in Z$ with $|a| = |b|$, the negative result is

Theorem (Feferman and Spector). *Π^1_1 paths through $\mathcal{O}$ exist, and for all such paths Z, there is a true Π^0_1 sentence not provable in $\bigcup_{a \in Z} T_a$.*

It is not altogether surprising then that Turing should be dissatisfied with ordinal logics as a practical method towards proving theorems (Feferman [**6**], p. 140). As an investigation into computation, rather than deduction, however, it would seem fair to say the following. Although Turing is remembered now mainly for "low level" programming (i.e., Turing machines, and perhaps even more importantly, cryptanalysis hacking), at least two somewhat abstract lines of research can be extracted from the work above that he initiated. First, one might carry on with an ordinal classification of programs of bounded complexity, paying careful attention to the notations used, and perhaps (in view of incompleteness along Π^1_1 paths through $\mathcal{O}$) abandoning any hope of assigning all such programs a recursive ordinal. Second, as to his completeness proof being "of no value", one might argue that, on the contrary, *by loading computations into an ordinal notation, he, in fact, anticipated a richer notion of computation, of not inconsiderable interest to current theoretical investigations.* That notion of computation is essentially the "intuitive" component of his ordinal logics that, moreover, by resisting elimination, led (already in Turing [**21**]) to the concept of *relative computability*. Pursuing this line of investigation further, the present paper is intended as a contribution to the argument made in Feferman [**7**] that relative computability is relevant to ordinary computation, in view of the myriad levels of abstraction relative to which a computation can be analyzed.

2. From ordinal notations to processes: bisimulations

The ordinal notations isolated in Turing's completeness proof constitute early examples of computational processes with the following "open-ended" features: intermediate computational states, non-provable termination, and countable non-determinism. These features are presently fleshed out.

2.1. Programs as states in a transition system. Without spelling out precisely what a transition is, it would appear plausible from Church's thesis to stipulate that the transitions undergone by a machine form, under a suitable syntactic presentation, an r.e. set. That is, to say that a transition predicate is mechanical is to assert that its transitions t can be named as theorems of some formal system. Indeed, an automaton (= program) e is typically characterized by its legal moves $i \vdash i'$ between so-called "instantaneous descriptions" i and i'. An instantaneous description i consists of a control state s and a data (e.g., tape) component d (plus an indication of the tape symbol currently being scanned, which we will assume is made implicit in the presentation of d). Roughly, what we will do is rewrite $(s_0, d) \rightarrow_e (s', d')$ as $(e, d) \rightarrow (e', d')$ where e' is the program described by $\rightarrow_e$ with initial state s', and $\rightarrow$ contains a copy of $\rightarrow_e$. Replacing e by p, the picture

$$(p, d) \quad \rightarrow \quad (p', d')$$

appears explicitly in so-called *mixed computation* (Bjørner, Ershov and Jones [**5**], p. xxvi), which we will recast as

$$p \overset{d,d'}{\rightsquigarrow} p' \tag{6}$$

to set the stage for an analysis of programs p relative to data d. That is, packaging the pairs (d, d') as *labels* l forming a set L, and passing from a program p back to the somewhat more abstract notion of a *state* s populating some set S, (6) leads to the notion of a *(labelled) transition system*, a triple $(L, S, \rightarrow)$ where $\rightarrow \subseteq S \times L \times S$. By presenting programs as states (following so-called *structural operational semantics*, Plotkin [**20**]), we can use a notion of equivalence on states to get a notion of equivalence on programs. But first, returning to the transition predicate $\rightsquigarrow \subseteq \omega \times \omega \times \omega$ occuring in (6), observe

Proposition 3. *There is an r.e. predicate $\rightsquigarrow \subset \omega \times \omega \times \omega$ that contains a copy of every r.e. transition predicate in that for every r.e. $\rightarrow \subseteq \omega \times \omega \times \omega$, there is a function f (that is, in fact, 1-1 and recursive) with domain equal to the set $\{s \mid (\exists l, t)\ s \xrightarrow{l} t \text{ or } t \xrightarrow{l} s\}$ of $\rightarrow$-states such that (i) for all s, l, t,*

$$s \xrightarrow{l} t \quad \textit{iff} \quad f(s) \overset{l}{\rightsquigarrow} f(t) \ ,$$

and (ii) for every u,

$$f(s) \overset{l}{\rightsquigarrow} u \ \text{ or } \ u \overset{l}{\rightsquigarrow} f(s) \quad \textit{implies} \quad (\exists t \in \mathit{dom}(f))\ f(t) = u \ .$$

Proof. Fix a binary primitive recursive function $\langle\cdot,\cdot\rangle : \omega\times\omega\to\omega$ and unary primitive recursive functions $i, j : \omega\to\omega$ such that for all $n, m<\omega$,

$$i(\langle n,m\rangle) = n\ ,\quad j(\langle n,m\rangle) = m \quad\text{and}\quad \langle i(n), j(n)\rangle = n\ .$$

Let U be an r.e. set such that $\{U_e\}_{e<\omega}$ enumerates all r.e. sets, where

$$U_e \quad=\quad \{n \mid \langle e,n\rangle\in U\}\ ,$$

and define $\rightsquigarrow$ to be

$$\{(s,l,s')\in\omega\times\omega\times\omega \quad|\quad i(s)=i(s') \text{ and } \langle j(s),\langle l, j(s')\rangle\rangle\in U_{i(s)}\}\ .$$

Note that $\rightsquigarrow$ is slightly different from a standard parametrization of binary r.e. predicates (although the modifications are minor enough). It is, in a suitable sense, an effective disjoint sum of all r.e. transition systems: given an r.e. $\to$, define f on $\to$-states s by $f(s)=\langle e,s\rangle$ where $U_e=\{\langle s,\langle l,t\rangle\rangle \mid s\overset{l}{\to}t\}$. $\dashv$

There are countably many transition predicates meeting the specification of $\rightsquigarrow$ given in Proposition 3, although, as will become plain shortly, they share the same abstract structure. For now, observe that $\rightsquigarrow$ generalizes the functional analysis of programs, $\{e\}(x)\simeq y$ being the special case $[e]\overset{x,y}{\rightsquigarrow}\surd$, for some fixed encoding $[e]$ of e. More generally, returning to a transition predicate $\to$ with data pair labels as in (6), it is simple enough to define transformations from $\to$ to $\to_*$

$$\frac{p\overset{d,d'}{\to}q}{p\overset{d,d'}{\to}_* q} \qquad\qquad \frac{p\overset{d,d''}{\to}_* q \qquad q\overset{d'',d'}{\to}p'}{p\overset{d,d'}{\to}_* p'}\ ,$$

and then to $\to_{at}$

$$\frac{p\overset{d,d'}{\to}_*\surd}{p\overset{d,d'}{\to}_{at}\surd}\ ,$$

setting the stage for a characterization of input/output equivalence by a more general notion to be defined next. (Note that if $\to$ is r.e., then $\to_*$ and $\to_{at}$ are r.e. as well; i.e., $\cdot_*$ and $\cdot_{at}$ can be interpreted inside $\rightsquigarrow$.)

2.2. Bisimulations and extensionality generalized. For an equivalence R on states to "respect" $\to$, it is natural to expect that if sRs' then s' can "simulate" every transition from s

$$(\forall l\in L)\qquad(\forall x\overset{l}{\leftarrow}s)(\exists y\overset{l}{\leftarrow}s')\ xRy\ ,$$

and conversely, s can simulate s'

$$(\forall l\in L)\qquad(\forall y\overset{l}{\leftarrow}s')(\exists x\overset{l}{\leftarrow}s)\ xRy\ .$$

It will prove useful to consider binary relations $R\subseteq S\times S'$ between the states of possibly two distinct transition systems $\langle L,S,\to\rangle$ and $\langle L,S',\to'\rangle$ and to formulate the requirements above within the "back-and-forth" operator $\cdot^{bf}$

that sends R to the relation R^{bf} consisting of pairs $(s, s') \in S \times S'$ such that for every $l \in L$,

$$(\forall x \stackrel{l}{\leftarrow} s)(\exists y \stackrel{l}{\leftarrow}' s')\ xRy \quad \text{and} \quad (\forall y \stackrel{l}{\leftarrow}' s')(\exists x \stackrel{l}{\leftarrow} s)\ xRy\} \ .$$

R is a *bisimulation* (Park [**19**]) if $R \subseteq R^{bf}$. Because R occurs positively in R^{bf}, the operator $R \mapsto R^{bf}$ is ($\subseteq$-)monotone, and the largest bisimulation $\underline{\leftrightarrow}$ (between $\langle L, S, \rightarrow \rangle$ and $\langle L, S', \rightarrow' \rangle$) can be calculated co-inductively as $\bigcap \{ \underline{\leftrightarrow}_\alpha \mid \alpha < \text{card}(S \times S')^+ \}$, where

$$\underline{\leftrightarrow}_\alpha \quad = \quad \bigcap_{\beta < \alpha} (\underline{\leftrightarrow}_\beta)^{bf}$$

(e.g., Milner [**16**]). The relation $\underline{\leftrightarrow}$ is also called *bisimilarity* and is manifestly Σ^1_1 in $\rightarrow$ and $\rightarrow'$

$$s \ \underline{\leftrightarrow} \ s' \quad \text{iff} \quad (\exists R)\ R \text{ is a bisimulation relating } s \text{ to } s' \ .$$

The extensional character of bisimilarity $\underline{\leftrightarrow}$ is nicely illustrated by two simple examples. First, input/output equivalence can be recovered as bisimilarity relative to the transition predicate $\rightarrow_{at}$ described in the end of subsection 2.1[4]

$$p \text{ and } q \text{ are input/output equivalent} \quad \text{iff} \quad p \ \underline{\leftrightarrow}_{at} \ q \ .$$

This example is somewhat degenerate inasmuch as the notion of a bisimulation is of interest mainly for transition predicates $\rightharpoondown$ with long chains of transitions $s \stackrel{l}{\rightharpoondown} s' \stackrel{l'}{\rightharpoondown} s'' \stackrel{l''}{\rightharpoondown} s''' \cdots$. Our second example is of this kind, and furthermore makes do with a singleton label set (i.e., $|L| = 1$), in contrast to $\rightarrow_{at}$, which uses labels heavily. More specifically, a binary relation B is turned into a transition predicate $\rightarrow_B = \{(a, *, b) \mid aBb\}$, by introducing some fixed label $*$ to the pairs in B. We will henceforth speak of bisimulations and bisimilarity over a binary relation B, when we actually have in mind the transition predicate $\rightarrow_B$ formed from it. Now, defining the Mostowski collapse c_B of a well-founded binary relation B by

$$c_B(a) \quad = \quad \{c_B(b) \mid bBa\} \ ,$$

then, as pointed out in Aczel [**1**],

$$c_B(a) = c_B(a') \quad \text{iff} \quad a \underline{\leftrightarrow} a' \ ,$$

where $\underline{\leftrightarrow}$ is bisimilarity relative to B (i.e., $\rightarrow_B$). Aczel [**1**] goes on to develop a set theory supporting this equivalence, whether or not B is well-founded (so that, in particular, c_B is always well-defined).

---

[4]The map $\cdot_{at}$ can be arranged to respect bisimilarity (i.e., $s \ \underline{\leftrightarrow}_{at} \ t$ if $s \ \underline{\leftrightarrow} \ t$) by adjoining a fresh label $\surd$ and postulating

$$s \stackrel{\surd}{\rightarrow} t \quad \text{iff} \quad s = t = \surd \ .$$

Note that the last condition above is not satisfiable in $\leadsto$, but in some r.e. transition predicate with a copy in $\leadsto$.

Returning to the coinductive characterization of $\underline{\leftrightarrow}$, note that $S \times S' = \underline{\leftrightarrow}_0 \supseteq \underline{\leftrightarrow}_1 \supseteq \cdots \supseteq \underline{\leftrightarrow}_\omega \supseteq \underline{\leftrightarrow}_{\omega+1} \supseteq \cdots \supseteq \underline{\leftrightarrow}$, where $\underline{\leftrightarrow}$ may differ from $\underline{\leftrightarrow}_\omega$, as illustrated by the notorious pair

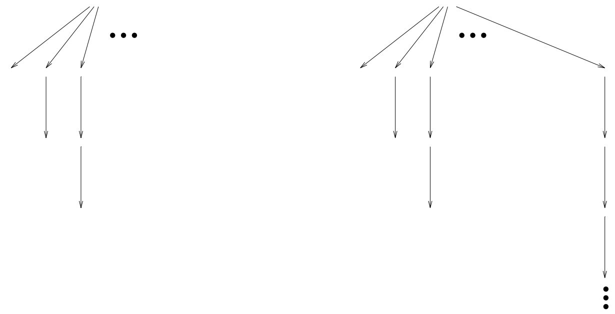

Figure 1

Confusing these trees is problematic inasmuch as the infinite branch can be interpreted as a failure of termination (or, say, an unfair merging of an infinite stream 1^∞ of 1's with 0, where 0 is forever neglected). The *Approximation Induction Principle* (AIP) of (for example) Baeten and Weijland [**2**] is (under an interpretation of equality as bisimilarity) the statement $\underline{\leftrightarrow} = \underline{\leftrightarrow}_\omega$. Just how badly AIP fails is a measure of the complexity (and hence, scope) of $\underline{\leftrightarrow}$. Nevertheless, a routine saturation argument can be employed to justify $\underline{\leftrightarrow} = \underline{\leftrightarrow}_\omega$.

Proposition 4. *For every transition predicate $\rightarrow \subseteq S \times L \times S$, there is a transition predicate $\rightarrow' \subseteq S' \times L \times S'$ with $\underline{\leftrightarrow}' = \underline{\leftrightarrow}'_\omega$ such that the first-order structure $\langle S, \{\xrightarrow{l}\}_{l\in L}\rangle$ is elementarily equivalent to $\langle S', \{\xrightarrow{l}'\}_{l\in L}\rangle$.*

Proposition 4 is an immediate consequence of the existence of recursively saturated elementary extensions (e.g., Barwise [**4**], p. 74), and the fact that a coinductive process of the sort that yields bisimilarity closes off at ω for recursively saturated models (e.g., Barwise [**4**], pp. 139, 210). At any rate, the moral of Proposition 4 is clear: first-order logic is inadequate for analyzing bisimilarity. The opposite of a saturation argument is an omitting types argument, a close companion of ω-completeness, and therefore ω-logic (or, as Girard [**11**] would put it, Π^1_1-logic). Curiously, omitting types is employed in Fernando [**9**] to bring out the first-order dependence on data, which is analyzed by reformulating transitions of the form $p \overset{d,d'}{\rightarrow} p'$ as $d \overset{p,p'}{\rightarrow} d'$. An altogether different use of omitting types is in proving the Gandy-Kreisel-Tait theorem (e.g., Barwise [**4**], p. 116) stating that the only sets a Π^1_1 theory is committed to are Δ^1_1. This fact is worth keeping in mind when establishing (as we will below) the non-Π^1_1 nature of bisimilarity relative to a certain logspace computable binary relation.

3. Pushing the analogy: semantics from syntax

Whereas termination is essential to an input/output (e.g., functional) view of computation, the question of termination takes on a different character within a more general theory of computational transitions that includes input/output transitions as special cases. Indeed, as has often been pointed out, non-termination is perfectly respectable, if not desirable, in so-called reactive systems (e.g., operating systems) that cease to be of service as soon as they terminate. The cost of admitting such immortality (formulated mathematically as non-well-foundedness) particularly in the presence of countable non-determinism is the subject of the present section.

As is implicit in Barwise, Gandy and Moschovakis [**3**], bisimilarity $\underline{\leftrightarrow}$ is Δ^1_1 in a transition predicate $\rightarrow$, provided $\rightarrow$ (or, more precisely, $\{(s,t) \mid (\exists l)\ s \xrightarrow{l} t\}$) is well-founded. It is easy to see why this should be the case. Recall that $R \subseteq S \times S$ is a bisimulation iff $R \subseteq R^{bf}$. Dually, call R a *co-bisimulation* if $R^{bf} \subseteq R$. Note that $\underline{\leftrightarrow}$ is a cobisimulation, and that there is a least cobisimulation $\sim$ which can be calculated inductively in the usual fashion, and which is a fixed point of $\cdot^{bf}$. Thus, since $\underline{\leftrightarrow}$ is the largest bisimulation, it remains to prove $\underline{\leftrightarrow}\ \subseteq \sim$, assuming $\rightarrow$ is well-founded. That is, it is enough to establish

$$(\exists s, s')\ s \underline{\leftrightarrow} s' \ \&\ s \not\sim s' \quad \text{implies}$$
$$(\exists l, t, t')\ t \underline{\leftrightarrow} t' \ \&\ t \not\sim t' \ \&\ s \xrightarrow{l} t\ ,$$

as this yields an infinite $\bigcup\{\xrightarrow{l}\ \mid l \in L\}$-descending sequence. But then, as $\sim$ is a cobisimulation, it follows from $s \not\sim s'$ that for some l, either (i) for some $t \xleftarrow{l} s$, $(\forall y \xleftarrow{l} s')\ t \not\sim y$, in which case $s \underline{\leftrightarrow} s'$ implies there is a $t' \xleftarrow{l} s'$ such that $t \underline{\leftrightarrow} t'$, or (ii) for some $t' \xleftarrow{l} s'$, $(\forall x \xleftarrow{l} s)\ x \not\sim t'$, and symmetrically, a $t \xleftarrow{l} s$ such that $t \underline{\leftrightarrow} t'$ follows from $s \underline{\leftrightarrow} s'$. Thus, if $\rightarrow$ is well-founded then $\underline{\leftrightarrow}$ is the unique fixed point of $\cdot^{bf}$ and $\underline{\leftrightarrow}$ is $\sim$

$$\begin{aligned} s \underline{\leftrightarrow} s' \quad & \text{iff} \quad (\exists R)\ \ Bis(R)\ \&\ Cobis(R)\ \&\ sRs' \\ & \text{iff} \quad (\forall R)\ \ Bis(R)\ \&\ Cobis(R)\ \supset\ sRs' \\ & \text{iff} \quad (\forall R)\ \ Cobis(R)\ \supset\ sRs'\ , \end{aligned}$$

where $Bis(R)$ and $Cobis(R)$ are the obvious predicates (arithmetic in $\rightarrow$) defining bisimulations and cobisimulations, respectively. Note that, in general, $s \underline{\leftrightarrow} s'$ is not equivalent to $(\exists R)\ Cobis(R)\ \&\ sRs'$ nor to $(\forall R)\ Bis(R) \supset sRs'$ (because of relations such as $S \times S$ and $=$).

Turning now to non-well-foundedness, what price must be paid for it? And what, in abstract, structural terms, do we get in return?

3.1. The challenge of non-well-foundedness. Recall that an *admissible set* (e.g., Barwise [**4**]) is a transitive set that (under the standard interpretation of $\in$) is a model of a certain weak set theory, KP. KP is characterized (among other axioms) by restricted, so-called Δ, forms of comprehension and collection. The paradigmatic example is where Δ is interpreted as (hereditarily) Δ^1_1, a principal motivation being to capture various facts about the Δ^1_1 sets on which a generalization can be based. In particular, two facts that generalize to some degree or another will interest us.

(a) The Π^1_1 sets are exactly the *inductive* sets — i.e., sets of the form $\{n \mid I_\varphi(0,n)\}$ for some elementary formula $\varphi(x,y,R)$ positive in the binary relation symbol R, where $I_\varphi(x,y)$ is the least fixed point of the operator

$$B \ \mapsto \ \{(m,n) \mid \varphi(m,n,B)\}$$

on binary relations B. Thus, the Δ^1_1 sets are exactly the sets that are both inductive and coinductive; in other words, the *hyperelementary* sets. This fact (due to Kleene and Spector) generalizes to countable sets, but may fail for uncountable sets (e.g., Moschovakis [**17**]).

(b) The set of Δ^1_1 well-orders is not Σ^1_1, implying that there are non-well-founded Δ^1_1 linear orders that have *no* Δ^1_1 descending sequences — so-called pseudo-well-orders. Defining the *well-founded part* $wf(\sqsubset)$ *of* a linear order $\sqsubset$ as the set of pairs (x,y) such that $x \sqsubset y$ and there are no $\sqsubset$-descending sequences from y, it follows that such a pseudo-well-order must have a well-founded part containing a copy of every Δ^1_1 well-order (as in the Feferman-Spector [**6**] result mentioned in section 1.2 above). Of course, there are no pseudo-well-orders if well-foundedness is hyperelementary (i.e., when the β-axiom mentioned in Barwise [**4**] holds).

Given a set A, Barwise, Gandy and Moschovakis [**3**] define

$$A^+ \ = \ \bigcap\{M \mid M \text{ is admissible and } A \in M\}$$

and go on to construct A^+ by a quotient construction similar to that of Aczel [**1**], where transitions are unlabelled and taken to be ordered pairs. It is crucial that $\underline{\leftrightarrow}$ be hyperelementary to push the argument in Aczel [**1**] through KP's restricted schemes of separation and collection. Assuming A is a non-empty transitive set closed under pairing, Barwise, Gandy and Moschovakis [**3**] encode A^+ by hyperelementary subsets of A that are well-founded trees. An inessential difference from Aczel [**1**] is that the transition predicates are unwound into trees, and subjected to the Mostowski collapse. An essential difference is that the predicates are assumed to be well-founded — an inductive condition, just as hyperelementarity is. But whereas well-foundedness makes $\underline{\leftrightarrow}$ inductive (in $\rightarrow$), hyperelementarity alone does not. To make this point, it will be sufficient to consider a linear order, which simplifies matters considerably. As usual, if $\sqsubset$ is a well-order, we will write $\mid \sqsubset \mid$ for its ordinal copy.

Proposition 5. *Let $\sqsubset$ be a linear order on ω, and for $a \in \omega$, write $wf(a)$ for the well-founded part of the order $\{(x,y) \mid x \sqsubset y \sqsubset a\}$. Let $\underline{\leftrightarrow}$ be bisimilarity relative to $\sqsubset$, with αth coinductive approximation $\underline{\leftrightarrow}_\alpha$.*

(i) *For all $a,b \in \omega$, and for every ordinal α,*

$$a \,\underline{\leftrightarrow}_\alpha\, b \quad \text{iff} \quad |wf(a)| \cap \alpha \ = \ |wf(b)| \cap \alpha \ ,$$

whence $\underline{\leftrightarrow} \; = \; \underline{\leftrightarrow}_\beta$ *for* $\beta = |wf(\sqsubset)| + 1$, *and*

$$a \underline{\leftrightarrow} b \quad \textit{iff} \quad a = b \;\; \textit{or} \;\; \textit{neither } a \textit{ nor } b \textit{ is in } wf(\sqsubset) \; .$$

(ii) *Let A be an infinite set, and $\sqsubset$ be A-hyperelementary, non-well-founded but have no A-hyperelementary descending sequences. If A-hyperelementary sets admit hyperelementary well-orderings, then $\underline{\leftrightarrow}$ cannot be A-inductive.*

Proof. Prove part (i) by induction on ordinals α; the case where α is 0 or a limit is immediate. If $\alpha = \beta + 1$, then

$$\begin{aligned} a \underline{\leftrightarrow}_\alpha b \quad & \text{iff} \quad (\forall x \sqsubset a)(\exists y \sqsubset b)\; x \underline{\leftrightarrow}_\beta y \;\; \& \\ & \qquad\;\; (\forall y \sqsubset b)(\exists x \sqsubset a)\; x \underline{\leftrightarrow}_\beta y \qquad \text{(by definition)} \\ & \text{iff} \quad (\forall x \sqsubset a)(\exists y \sqsubset b)\; |wf(x)| \cap \beta \; = \; |wf(y)| \cap \beta \\ & \qquad\;\; \& \;\; (\forall y \sqsubset b)(\exists x \sqsubset a) \\ & \qquad\quad\; |wf(x)| \cap \beta \; = \; |wf(y)| \cap \beta \quad \text{(by ind. hyp.)} \\ & \text{iff} \quad |wf(a)| \cap \alpha \; = \; |wf(b)| \cap \alpha \; , \end{aligned}$$

where the last equivalence follows from the linearity of $\sqsubset$.

As for part (ii), assume towards a contradiction that $\underline{\leftrightarrow}$ were A-inductive. Then (adapting the proof of the Closure Theorem of Moschovakis [**17**], p. 33) the formula

$$\begin{aligned} \psi(t, n, m, T) \quad \text{iff} \quad & [t = 0 \;\&\; (n, m) \in \{(x, y) \mid T(0, x, y)\}^{bf}] \text{ or} \\ & [t = 1 \;\&\; (\exists x, y)(T(0, x, y) \;\&\; x \;\underline{\not\leftrightarrow}\; y)] \end{aligned}$$

(with $\underline{\leftrightarrow}$ equal to the 0-th section $J_\psi(0, \cdot, \cdot)$ of the largest fixed point J_ψ) defines a coinduction that takes one more step than coinduction based on $\cdot^{bf}$ (but not more than the least A-hyperelementary ordinal). Hence, $\underline{\leftrightarrow} = \underline{\leftrightarrow}_\alpha$ for some A-hyperelementary ordinal α, and there is an a ($\notin wf(\sqsubset)$) such that whenever $b \underline{\leftrightarrow}_\alpha a$, there is a descending sequence from b. Now, using an A-hyperelementary well-ordering of $\{x \mid x \underline{\leftrightarrow}_\alpha a\}$, construct an A-hyperelementary descending sequence from a, contradiction. $\dashv$

The significance of A-inductiveness is shown in part by its identification, over countable admissible sets A, with A-recursive enumerability. Setting aside the matter of inductiveness (which follows from well-foundedness) and concentrating simply on the ordinals, note that $\underline{\leftrightarrow}$ cannot equal $\underline{\leftrightarrow}_\beta$ for some ordinal β in A^+, or else β and $\beta + 1$ would be indistinguishable in A^+, which is absurd. This suggests the following elaboration of a picture found in Barwise [**4**].

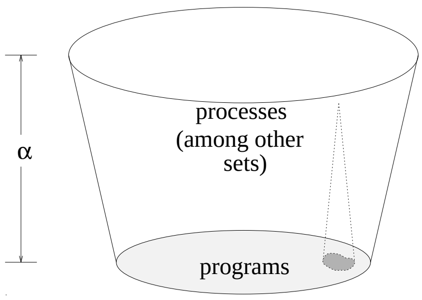

Figure 2

The idea behind Figure 2 is that a process is an equivalence class of (syntactic) programs, a minimal requirement on such an equivalence being that it be a bisimulation. The ordinal α is (among other things) a tight bound on the closure ordinal of an operator $\cdot^{bf}$ determining whether or not there is a bisimulation relating two programs. Indeed, the explicit Σ^1_1 formulation of $s \underline{\leftrightarrow} s'$, $(\exists R)$ R is a bisimulation relating s to s', can be read as

$$\text{there is a model where } s \text{ and } s' \text{ are equal },$$

since a bisimulation R yields a model after division: define $\rightarrow/R$ by

$$s \xrightarrow{l}/R\; t \quad \text{iff} \quad (\exists s', t')\; s'Rs \;\&\; t'Rt \;\&\; s' \xrightarrow{l} t' \ ,$$

which, by the definition of a bisimulation, amounts to

$$s \xrightarrow{l}/R\; t \quad \text{iff} \quad (\exists t'Rt)\; s \xrightarrow{l} t' \ .$$

Note that $\{(s,s) \mid (\exists l,t)\ s \xrightarrow{l} t\}$ is a bisimulation between $\rightarrow/R$ and $\rightarrow$, and that $\rightarrow/R$ and $\rightarrow$ induce the same relation of bisimilarity, with, further, $(\rightarrow/R)/R' = \rightarrow/R' = (\rightarrow/R')/R$ when $R' \supseteq R$. We will be mainly concerned with $\rightarrow/\underline{\leftrightarrow}$, which will be abbreviated as $\rightarrow_\circ$. For a binary relation $>$, we will also write $>_\circ$ for $\rightarrow_\circ$ where $\rightarrow$ is $\{(n,0,m) \mid n > m\}$. Call a transition predicate $\rightarrow$ *reduced* if $\rightarrow$ is $\rightarrow_\circ$. (Note that $\rightarrow_\circ$ is reduced.) If, further, $\underline{\leftrightarrow}$ is $=$ (i.e., $a \underline{\leftrightarrow} b$ implies $a = b$), then the transition predicate $\rightarrow$ is *strongly extensional* (Aczel [**1**]). (We will confine ourselves to $\rightarrow \subseteq \omega \times \omega \times \omega$, so that the states in strongly extensional quotients are not $\underline{\leftrightarrow}$-equivalence classes, but representatives of such.)

A non-inductive notion of bisimilarity can be obtained over a transition predicate that is not only hyperarithmetic, but by Grigorieff [**13**] computable in logspace. The key is Σ^1_1 boundedness.

Theorem 6. *(i) (Fernando* [**9**]*) Bisimilarity* $\underline{\leftrightarrow}$ *over* $\succ$ *is* $\Sigma^1_1 - \Pi^1_1$.

(ii) $\succ_\circ$ *is* $\Sigma^1_1 - \Pi^1_1$.

Proof. Membership in Σ^1_1 is obvious. To see that the predicates lie outside Π^1_1, fix an $a_0 \in \mathcal{O}^* - \mathcal{O}$, and conclude from Proposition 5 (not to mention the extraction from $\mathcal{O}^*$ of Π^1_1 paths) that

$$\begin{aligned} \{(b,0,c) \mid a_0 \succ b \;\&\; b \succ c \;\&\; b \in \mathcal{O}\} &= \\ \{(b,0,c) \mid a_0 \succ b \;\&\; b \succ c \;\&\; b \not\underline{\leftrightarrow}\, a_0\} &= \\ \{(b,0,c) \mid a_0 \succ b \;\&\; b \succ c \;\&\; b \not\succ_\circ a_0\} \end{aligned}$$

is well-founded with height ω_1^{CK}. $\dashv$

Before closing this subsection, let us remark that an infinite descending sequence in a transition system may also reflect computations over infinite streams of data — e.g. infinite precision reals.

3.2. Operational semantics reduced. The measure of complexity provided by Theorem 6 concerns a particular presentation of $\rightsquigarrow_\circ$; its robustness might be tested by trading off the complexity of transition predicate with the complexity of bisimilarity over that transition predicate. Towards that end, the following definitions will be useful. Given two transition predicates $\rightarrow$ and $\rightarrow' \subseteq \omega \times \omega \times \omega$, a binary relation R (on ω) *copies* $\rightarrow$ *into* $\rightarrow'$ if R is a bisimulation between $\rightarrow$ and $\rightarrow'$ such that

$$dom(R) \;=\; \{s \mid (\exists l, t)\; s \xrightarrow{l} t \text{ or } t \xrightarrow{l} s\}\;.$$

The predicate $\rightarrow$ *has a copy in* $\rightarrow'$ if there is a binary relation R copying $\rightarrow$ into $\rightarrow'$.

Theorem 7. *Fix an $a_0 \in \mathcal{O}^* - \mathcal{O}$, and suppose R copies $\{(b,0,c) \mid a_0 \succ b \;\&\; b \succ c\}$ into a predicate $\rightarrow$.*

(i) *If $\rightarrow$ is Σ^1_1, then bisimilarity $\underline{\leftrightarrow}$ over $\rightarrow$ is not Π^1_1.*
(ii) *If R and $\underline{\leftrightarrow}$ are Π^1_1, then $\rightarrow$ is* not *Π^1_1.*

Hence, if $\rightarrow$ is strongly extensional, then $\rightarrow$ is not Σ^1_1 nor, if R is Π^1_1, Π^1_1.

Proof. Fix a $b_0 \prec a_0$ such that $b_0 \notin \mathcal{O}$, and choose an s_0 such that $b_0 R s_0$. As $\{(b,0,c) \mid a_0 \succ b \;\&\; b \succ c\}$ is a linear order, R restricted to $\{b \in \mathcal{O} \mid a_0 \succ b\}$ must be a function (by Proposition 5), call it f. Appealing again to Proposition 5, note that

$$\begin{aligned} \{(s,0,t) \mid s_0 \xrightarrow{0} s \;\&\; s \xrightarrow{0} t \;\&\; s_0 \not\underline{\leftrightarrow}\, s\} &= \\ \{(f(b),0,f(c)) \mid b_0 \succ b \;\&\; b \succ c \;\&\; b \in \mathcal{O}\} &= \\ \{(f(b),0,f(c)) \mid b_0 \succ b \;\&\; b \succ c \;\&\; f(b) \not\rightarrow_\circ f(b)\} \end{aligned}$$

is well-founded with height ω_1^{CK}, and that $\rightarrow_\circ$ is Π^1_1 if both $\underline{\leftrightarrow}$ and $\rightarrow$ are. (For the last equivalence, recall that Π^1_1 functions are Δ^1_1 — $f(x) = y$ iff $(\forall z \neq y)\; f(x) \neq z$.) $\dashv$

Moving in the opposite direction, it turns out that, as complex as $\rightsquigarrow_\circ$ is, that complexity need not provide a natural reduction of arithmetic transition predicates. That is to say, reducibility to $\rightsquigarrow_\circ$ becomes severely restricted if the reduction f of $\langle L, S, \rightarrow\rangle$ to $\langle \omega, \omega, \rightsquigarrow_\circ\rangle$ is required to have the form

$$f(x,y,z) \;=\; (g(x), h(y), g(z))$$

where for all $s, t \in S$ and $l \in L$,

$$s \xrightarrow{l} t \quad \text{iff} \quad g(s) \overset{h(l)}{\leadsto_\circ} g(t)$$

and, further, for every $u \in \omega$,

$$g(s) \overset{h(l)}{\leadsto_\circ} u \quad \text{implies} \quad (\exists t \in S)\ g(t) = u \quad .$$

The latter is a form of "end extension" condition, that is essential for g to be a bisimulation. The requirement in a bisimulation that h be the identity is not much of a restriction when there is only one label. Let us be careful about taking the matter of labelling for granted, however, as a great deal of complexity can be shoved over there. For example, consider the embedding

$$(\dagger) \qquad A \ \mapsto \ \{(s_0, n, \surd) \mid n \in A\}$$

of a set A into the transition predicate above with two states s_0 and $\surd$. An obvious consequence of this is

Proposition 8. *There is a Π^0_1 strongly extensional transition predicate without a copy in $\leadsto$ (or $\leadsto_\circ$).*

The embedding $(\dagger)$ contrasts with the other more interesting extreme studied in Aczel [**1**] where the transition predicate is unlabelled (or, if one prefers, $|L| = 1$) and taken as $\in^{-1}$ (i.e., $\{(a, l, b) \mid b \in a\}$ for some fixed l). We will presently consider a more moderate position, to get a sense of how to pass between these two extremes. In particular, the notion of a copy will be relativized to the (computer science) notion of a trace. A trace is, in its simplest form, a sequence $(l_1, l_2, \ldots, l_n)$ of labels on successive transitions

$$s \xrightarrow{l_1} s_1 \ \& \ s_1 \xrightarrow{l_2} s_2 \ \& \ \cdots \ \& \ s_{n-1} \xrightarrow{l_n} s_n$$

where s_n is a "terminal" state. A terminal state is not meant here to be (necessarily) *any* element of the set

$$\{s \in S \ \mid \ \neg(\exists l, t)\ s \xrightarrow{l} t\} \ ,$$

which is co-r.e., assuming $\rightarrow$ is r.e. Rather, we have in mind a more effective set of states, that can be marked by introducing a fresh label $\surd$ and postulating that for all states s and t,

$$s \xrightarrow{\surd} t \quad \text{iff} \quad s = t \text{ and } s \text{ is terminal.}$$

This modification to the transition predicate can*not* spoil its r.e. character, assuming the terminal states form an r.e. set. Nor can the map $\cdot_{tr}$ that sends $\rightarrow$ to the relation $\rightarrow_{tr}$ consisting of triples $(s, (l_1, l_2, \ldots, l_n), t)$ such that

$$\begin{aligned} &\surd \notin \{l_1, l_2, \ldots, l_n\} \ \& \ t \xrightarrow{\surd} t \ \& \\ &(\exists s_1, \ldots, s_{n-1})\ s \xrightarrow{l_1} s_1 \ \& \ s_1 \xrightarrow{l_2} s_2 \ \& \ \cdots \ \& \ s_{n-1} \xrightarrow{l_n} t \ . \end{aligned}$$

(Since $\rightarrow_{tr}$ is r.e. if $\rightarrow$ is r.e., $\cdot_{tr}$ can be interpreted within $\leadsto$.) A transition predicate $\rightarrow$ is now said to *have a trace-equivalent copy in* $\rightarrow'$ if $\rightarrow_{tr}$ has a copy in $\rightarrow'_{tr}$. Observe that trace-equivalence is coarser than bisimilarity.

Moving some of the complexity away from the labelling, the argument (†) behind Proposition 8 can be used for

Proposition 8′. *(i) There is a well-founded Π^0_1 transition predicate with $|L| = 2$ that has no trace-equivalent copy in $\rightsquigarrow$.*

(ii) *There is a Π^0_1 transition predicate with $|L| = 2$ and branching bound 2 (i.e., for all n $|\{m \mid (\exists l)\ n \xrightarrow{l} m\}| \leq 2$) that has no trace-equivalent copy in $\rightsquigarrow$.*

(iii) *There is a strongly extensional Δ^0_2 transition predicate with $|L| = 2$ that has no trace-equivalent copy in $\rightsquigarrow$.*

Proof. The idea is to code a subset A of ω into a transition predicate $\rightarrow_A$ which has a copy in $\rightsquigarrow$ only if for some fixed label $l \neq \surd$, there is an s such that for all $n < \omega$,

$$\begin{aligned} n \in A \quad \text{iff} \quad & (\exists t)\ s \overset{l}{\rightsquigarrow} [t]_0 \ \&\ [t]_0 \overset{l}{\rightsquigarrow} [t]_1 \ \&\ \cdots \\ & \&\ [t]_{n-1} \overset{l}{\rightsquigarrow} [t]_n \ \&\ [t]_n \overset{\surd}{\rightsquigarrow} [t]_n \ , \end{aligned}$$

where $[t]_i$ gives the ith component of the finite sequence encoded (under some effective scheme) as t. Note that the above equivalence implies A is r.e. Now, given a binary 1-1 primitive recursive function $\langle\cdot,\cdot\rangle$ from $\omega \times \omega$ into $\omega - \{0\}$, define for (i),

$$\begin{aligned} \rightarrow_A \ = \ & \{(0, l, \langle n, n\rangle) \mid n \in A\} \ \cup \\ & \{(\langle n, m\rangle, l, \langle n, m-1\rangle) \mid n \in A \text{ and } 0 < m \leq n\} \\ & \cup \{(\langle n, 0\rangle, \surd, \langle n, 0\rangle) \mid n \in A\} \ , \end{aligned}$$

while for (ii),

$$\begin{aligned} \rightarrow_A \ = \ & \{(\langle 0, n\rangle, l, \langle 0, n+1\rangle) \mid n < \omega\} \ \cup \\ & \{(\langle 0, n\rangle, l, \langle 1, n\rangle) \mid n \in A\} \\ & \cup \{(\langle 1, n\rangle, \surd, \langle 1, n\rangle) \mid n \in A\} \ . \end{aligned}$$

Note that we cannot require $\rightarrow_A$ (for arbitrary $A \subset \omega$) to be both well-founded and finitely branching (by König) as every finite transition predicate has a copy in $\rightsquigarrow$.

As for (iii), let us reduce the transition predicate given by case (i) by choosing representatives of $\underline{\leftrightarrow}$-equivalence classes as follows. Given an m less than some element of A, let m_A be the least $x \in A$ that is $\geq m$. Note that for $m \in A$, $m_A = m$. Now, set $\rightarrow_A$ to

$$\begin{aligned} & \{(0, l, \langle n, n\rangle) \mid n \in A\} \ \cup \ \{(\langle a_0, 0\rangle, \surd, \langle a_0, 0\rangle)\} \ \cup \\ & \{(\langle m_A, m\rangle, l, \langle (m-1)_A, m-1\rangle) \mid m > 0 \text{ and } m \text{ is less} \\ & \qquad\qquad\qquad\qquad\qquad \text{than some element of } A\} \end{aligned}$$

where a_0 is the least element of A. Observe that $x = m_A$ iff $x \in A \ \&\ x \geq m \ \&\ (\forall y < x)(y \in A \supset y < m)$, whence $\rightarrow_A$ is Δ^0_2. $\dashv$

Next, taking up the case $|L| = 1$, let us mention two features of a strongly extensional copy $\Rightarrow$ of $\rightsquigarrow$ that would seem to render it unsuitable for a set-theoretic characterization (as in Aczel [**1**] or Barwise, Gandy and Moschovakis [**3**]). We have in mind here, of course, a restriction of the transition predicate to one label, which we henceforth ignore. Let $V = \{a \mid (\exists b)\ a \Rightarrow b \text{ or } b \Rightarrow a\}$.

(i) A "universal" set exists — i.e., there is an a in V such that $a \Rightarrow b$ for every b in V. (This a is the copy of state 0 in the r.e. transition predicate $\{(0, n+1) \mid n \in \omega\} \cup \{(n+1, m+1) \mid n \rightsquigarrow m\}$ obtained by "rooting" $\rightsquigarrow$.)

(ii) V is not closed under very simple comprehension schemes: V contains an $x = \{x\} \cup \omega_1^{CK}$ but not $\omega_1^{CK} \subset x$. (This is an immediate consequence of Π^1_1 paths through $\mathcal{O}$ and Σ^1_1 boundedness.)

3.3. The question of strongly extensional Π^1_1 approximations. An example of a strongly extensional copy $\Rightarrow$ of $\rightsquigarrow$ is $\rightsquigarrow_\circ$ restricted to the set

$$A_0 \quad = \quad \{s \in \omega \mid (\forall s' < s)\ s' \underline{\not\leftrightarrow} s\}$$

of least $\underline{\leftrightarrow}$-representatives. That is, for all $s, l, t \in \omega$,

$$\begin{array}{lll} s \stackrel{l}{\Rightarrow} t & \text{iff} & (\exists t' \underline{\leftrightarrow} t)\ s \stackrel{l}{\rightsquigarrow} t' \ \&\ (\forall s' < s)\ s' \underline{\not\leftrightarrow} s \ \ \& \\ & & (\forall t' < t)\ t' \underline{\not\leftrightarrow} t \ . \end{array}$$

Whether $\Rightarrow$ (or any other strongly extensional copy of $\rightsquigarrow$) is Π^1_1 is not known by the present author. The problem is that the first conjunct $\rightsquigarrow_\circ$ is $\Sigma^1_1 - \Pi^1_1$. But now, just as a Π^1_1 path through $\mathcal{O}$ can be built from a relativization $\mathcal{O}^*$ of $\mathcal{O}$ to hyperarithmetic sets, we might consider a notion of pseudo-bisimilarity defined up to hyperarithmetic sets. Given a transition system $\langle L, S, \rightarrow\rangle$, with $L, S \subseteq \omega$, define the binary relation $\sim$ on S by

$$\begin{array}{lll} a \sim b & \text{iff} & \text{there is a hyperarithmetic bisimulation} \\ & & \text{relating } a \text{ to } b \ . \end{array}$$

Proposition 9. *$\sim$ is a bisimulation, and, if $\rightarrow$ is Δ^1_1, is Π^1_1.*

An obvious Π^1_1 approximation of $\Rightarrow$ is then $\{(s, l, t) \mid s \stackrel{l}{\rightarrow}/\!\sim t \text{ and } s, t \in A\}$, for some Δ^1_1 copy $\rightarrow$ of $\rightsquigarrow$, and some "suitable" Π^1_1 set A. More precisely, a subset A of ω is a *suitable representation* if

(S1) $(\forall s \in \omega)(\exists! a \in A)\ a \underline{\leftrightarrow} s$

(S2) $(\forall a \in A)(\forall b, l \text{ s.t. } a \stackrel{l}{\rightarrow} b)(\exists c \in A)\ c \underline{\leftrightarrow} b \ \&\ a \stackrel{l}{\rightarrow}/\!\sim\ c$.

Note that (S1) determines a function f_A, in terms of which (S2) becomes

$$(\forall a \in A)(\forall b, l \text{ s.t. } a \stackrel{l}{\rightarrow} b) \qquad a \stackrel{l}{\rightarrow}/\!\sim\ f_A(b) \ .$$

For a Π^1_1 suitable representation A, Theorem 7 implies that f_A cannot be Π^1_1. Returning to the set A_0 of least $\underline{\leftrightarrow}$-representatives, note that A_0 is Π^1_1 and (with $\rightarrow\ =\ \rightsquigarrow$) meets (S1) by a function that is not manifestly Π^1_1. Unfortunately, it is unlikely that A_0 meets (S2) in view of

Proposition 10. *For all $a, b \in \mathcal{O}^* - \mathcal{O}$, if $a \succ b$ then $a \not\sim b$ (although $a \underline{\leftrightarrow} b$).*

Proof. Otherwise, construct a hyperarithmetic descending sequence from a, appealing to

$$a \succ b \text{ and } a \sim b \quad \text{implies} \quad (\exists c \prec b)\ b \sim c\ ,$$

and repeatedly choosing the first c according to some enumeration of $\succ$. $\dashv$

That is, a suitable representation (relative to $\rightsquigarrow$) cannot contain an element a of $\mathcal{O}^* - \mathcal{O}$. This is easily repaired, however; replace a by an a' bisimilar to a with $a' \succ a'$. Thus, one might try to build successive improvements A_{n+1} of A_n, with $A = \lim_{n\to\infty} A_n$ (i.e., $s \in A$ iff $(\exists N)(\forall n > N)\ s \in A_n$). Alternatively, we might focus on beefing up $\rightarrow^0 = \rightsquigarrow/\sim$ into a Π^1_1 set $\rightarrow_\omega = \bigcup_{n<\omega} \rightarrow^n$ such that for all $a, a' \in A_0$,

$$(\forall b, l) \qquad a \overset{l}{\rightsquigarrow} b \ \&\ b \underline{\leftrightarrow} a' \quad \supset \quad a \overset{l}{\rightarrow}_\omega a'\ ,$$

a Π^1_1 condition suggesting that a strongly extensional copy of $\rightsquigarrow$ be formed from restricting $\rightarrow_\omega$ to A_0. A combination of improvements on A_0 and $\rightsquigarrow/\sim$ might be required, but alas, the author has been unable to carry some such construction to completion.

4. Discussion

The preceding is in no small measure inspired by Feferman [**7**], which concludes with the remark

> ... I think that it can be argued that — whatever a "fundamental" theory may tell us is the basic or underlying mechanism for given technological processes or systems — it is necessary for our design and use of such to think of them at various conceptual levels and with various modular forms of organization, that is *we must think of them in structural terms*. The story of Turing's "oracle" and its significance for actual computability is but one example among many of this characteristic *modus operandi* of human intelligence.

While transition predicates constitute "but one example among many" possible formulations of "conceptual levels", the generality of that approach is promising. Furthermore, that generality can be grounded in processes that are *mechanical* inasmuch as their transitions can be programmed. Construed in ordinary recursion-theoretic terms, the transitions that can be programmed are based on process names (i.e., syntax) and may fail to be substitutive with respect to non-trivial notions of process equality (requiring, as that does, computational notions for sets — i.e., equivalence classes — rather than for natural numbers). Instead, a weaker form of substitutivity is provided by the (key) concept of a bisimulation (applied to a transition system where programs are states), from which a natural notion of process equality is then derived. The highly uncomputable character of the resulting logical abstractions should not disturb us so long as we do not confuse reasoning *about* mechanical computation with carrying out a mechanical computation. Recall how the "absolute" character of Church's thesis contrasts with the "relative" character of Gödel incompleteness (i.e., consistency). "The necessity for intuition" in Turing [**21**] cannot be "eliminated"; rather, the ineluctability of partiality leads to relative computability and various notions of consistency.

References

[1] Peter Aczel. *Non-Well-Founded Sets.* CSLI Lecture Notes Number 14, Stanford, 1988.
[2] J.C.M. Baeten and W.P. Weijland. *Process Algebra.* Cambridge Tracts in Theoretical Computer Science 18. Cambridge University Press, 1990.
[3] J. Barwise, R.O. Gandy, and Y.N. Moschovakis. The next admissible set. *Journal of Symbolic Logic*, 36(1), 1971.
[4] Jon Barwise. *Admissible Sets and Structures.* Perspectives in Mathematical Logic. Springer-Verlag, Berlin, 1975.
[5] D. Bjørner, A.P. Ershov, and N.D. Jones. *Partial Evaluation and Mixed Computation.* Elsevier, IFIP, 1988.
[6] Solomon Feferman. Turing in the land of 0(z). In R. Herken, editor, *The Universal Turing Machine.* Kammerer and Unverzagt, 1988.
[7] Solomon Feferman. Turing's "oracle": from absolute to relative computability and back. In the proceedings of a symposium on Structures in Mathematical Theories, San Sebastian, Spain, 1990.
[8] Tim Fernando. Provably recursive programs and program extraction. In J. Leach Albert et al, editor, *Proc. ICALP '91*, LNCS 510. Springer-Verlag, Berlin, 1991. A revision of the paper (entitled simply 'Provably recursive programs') appears as CWI Technical Report CS-R9212, Amsterdam, March 1992.
[9] Tim Fernando. Bisimulations and predicate logic. Technical Report CS-R9333, Centre for Mathematics and Computer Science (CWI), Amsterdam, May 1993. A revision has been accepted for publication in the *Journal of Symbolic Logic*.
[10] J.-Y. Girard, Y. Lafont, and P. Taylor. *Proofs and Types.* Cambridge Tracts in Theoretical Computer Science 7. Cambridge University Press, 1989.
[11] Jean-Yves Girard. *Proof Theory and Logical Complexity*, volume 1. Bibliopolis, Napoli, 1987.
[12] Kurt Gödel. Remarks before the Princeton bicentennial conference on problems in mathematics. In S. Feferman et al, editor, *Collected Works, volume* II. Oxford University Press, 1990.
[13] Serge Grigorieff. Every recursive linear ordering has a copy in DTIME-SPACE$(n, log(n))$. *Journal of Symbolic Logic*, 55(1), 1990.
[14] Stephen Cole Kleene. On the interpretation of intuitionistic number theory. *Journal of Symbolic Logic*, 10, 1945.
[15] Georg Kreisel. On the interpretation of non-finitist proofs, I. *Journal of Symbolic Logic*, 16(4), 1951.
[16] Robin Milner. *Communication and Concurrency.* Prentice Hall, 1989.
[17] Yiannis N. Moschovakis. *Elementary Induction on Abstract Structures.* North-Holland, Amsterdam, 1974.
[18] Yiannis N. Moschovakis. A model of concurrency with fair merge and full recursion. *Information and Computation*, 93(1), 1991.
[19] David Park. Concurrency and automata on infinite sequences. In P. Deussen, editor, *Proc. 5th GI Conference*, LNCS 104. Springer-Verlag, Berlin, 1981.
[20] Gordon D. Plotkin. A structural approach to operational semantics. Technical Report DAIMI FN-19, Computer Science Department, Aarhus University, 1981.
[21] Alan Turing. Systems of logic based on ordinals. *Proc. London Math. Soc.*, 45, 1939.