A Fully Abstract Game Semantics for Countable Nondeterminism

W. J. Gowers
Department of Computer Science
University of Bath
Bath, United Kingdom
W.J.Gowers@bath.ac.uk

James Laird
Department of Computer Science
University of Bath
Bath, United Kingdom
J.D.Laird@bath.ac.uk

Abstract

The concept of fairness for a concurrent program means that the program must be able to exhibit an unbounded amount of nondeterminism without diverging. Game semantics models of nondeterminism show that this is hard to implement; for example, Harmer and McCusker's model only admits infinite nondeterminism if there is also the possibility of divergence. We solve a long standing problem by giving a fully abstract game semantics for a simple stateful language with a countably infinite nondeterminism primitive. We see that doing so requires us to keep track of infinitary information about strategies, as well as their finite behaviours. The unbounded nondeterminism gives rise to further problems, which can be formalized as a lack of continuity in the language. In order to prove adequacy for our model (which usually requires continuity), we develop a new technique in which we simulate the nondeterminism using a deterministic stateful construction, and then use combinatorial techniques to transfer the result to the nondeterministic language. Lastly, we prove full abstraction for the model; because of the lack of continuity, we cannot deduce this from definability of compact elements in the usual way, and we have to use a stronger universality result instead. We discuss how our techniques may also be applied to Tsukada and Ong's model of PCF with finite nondeterminism, yielding an alternative proof of computational adequacy.

Keywords semantics, nondeterminism, games and logic

1 Introduction

Picture two concurrent processes P and Q with shared access to a variable v that holds natural numbers and is initialized to 0. The execution of P consists in an infinite loop that increments the value of v at each iteration. Meanwhile, Q performs some computation A, and then prints out the current value of v and terminates the whole program. Since we cannot predict in advance how may cycles of the loop in P will have elapsed by the time the computation A has completed, the value that ends up printed to the screen may be arbitrarily large. Furthermore, under the basic assumption that the task scheduler is *fair*; i.e., any pending task must eventually be executed, our program must always terminate by printing out some value to the screen.

We have therefore built an *unbounded nondeterminism* machine, that can print out arbitrarily large natural numbers but which never diverges. This is strictly more powerful than finitary choice nondeterminism. [1] What we have just shown is that if we want to solve the problem of building a fair task scheduler, then we must in particular be able to solve the problem of building an unbounded nondeterminism machine.

This is an important observation to make about concurrent programming, because the task of implementing unbounded nondeterminism is difficult – indeed, considerably more so than that of implementing bounded nondeterminism. Dijkstra argues in [Dijkstra 1997, Ch. 9] that it is impossible to implement unbounded nondeterminism, showing that the natural constructs from which we construct imperative programs satisfy a *continuity* property that unbounded nondeterminism lacks. For semanticists, this lack of continuity is a problem in itself, since the standard proofs of computational adequacy and full abstraction typically make use, implicitly or otherwise, of the fact that composition within the model is continuous with respect to some ordering.

We shall explore some of the problems relating to unbounded nondeterminism, and how they may be solved, using game semantics to give a fully abstract model of a simple stateful language – Idealized Algol – enhanced with a countable nondeterminism primitive. We begin with a pair of examples that will illustrate the lack of continuity, from a syntactic point of view. Let `nat` be our natural number type and consider a sequence of functions $<n\colon \texttt{nat} \to \texttt{nat}$, where $<m\ k$ evaluates to 0 if $k < n$ and diverges otherwise. In that case, the least upper bound of the $<n$ is the function that combines all their convergent behaviours; i.e., the function $\lambda k.k;\texttt{0}$ that evaluates its input and then returns 0. If $?\colon \texttt{nat}$ is an unbounded nondeterminism machine, then function application to ? is not continuous; indeed, $<m$? always has the possibility of divergence – since ? may evaluate to $m + 1$, say. But $(\lambda k.k;\texttt{0});?$ always converges to 0.

Lack of continuity is a problem because fixed-point combinators are typically built using least upper bounds, and proving adequacy of the model typically requires that these least upper bounds be preserved. In a non-continuous situation, we will need to come up with new techniques in order to prove adequacy without using continuity.

A closely connected problem with unbounded nondeterminism is that it leads to terms that may be distinguished only by their *infinitary* behaviour. A program that flashes a light and unboundedly

[1] Using recursion, we can build a program out of finite nondeterminism that can produce arbitrarily large natural numbers; however, this program also admits the possibility of divergence.

LICS'18, 9-12 July, 2018, Oxford, UK
2018.

nondeterministic number of times cannot reliably be distinguished in finite time from a program that flashes that light forever: however long we watch the light flash, there is always a chance that it will stop at some point in the future. From a game semantics point of view, this corresponds to the observation that it is not sufficient to consider sets of finite plays in order to define strategies: we must consider infinite sequences of moves as well.

1.1 Related Work

Our game semantics model bears closest resemblance to that of Harmer and McCusker [Harmer and McCusker 1999], which is a fully abstract model of Idealized Algol with *finite* nondeterminism. Indeed, our work can be viewed as an extension of the Harmer-McCusker model with the extra information on infinite plays that we need to model countable nondeterminism.

The idea of adding infinite traces into strategies in order to model unbounded nondeterminism goes back to Roscoe's work on CSP [Roscoe 1993], and is very similar to work by Levy [Levy 2008] on game semantics for a higher order language. In particular, we will need something similar to Levy's *liveliness* condition on strategies, which is a way of saying that a strategy is a union of deterministic strategies – something that is not automatic when we start tracking infinite plays.

An alternative approach to the game semantics of nondeterminism can be found in Tsukada and Ong's sheaf model of nondeterministic PCF [Tsukada and Ong 2015] and in the more general work on concurrency by Winskel et al. (e.g., see [Winskel 2013] and [Castellan et al. 2016]), in which there is a very natural interpretation of nondeterminism. Although we are able to give a model of Idealized Algol with countable nondeterminism in the more traditional Harmer-McCusker style, it seems necessary to introduce this extra machinery in order to model stateless languages such as PCF (and certainly to model concurrency). In the last section of this paper, we will show how our methods can be applied under very general circumstances, and in particular to these models of nondeterministic stateless languages.

Related work by Laird [Laird 2015, 2016] discusses a semantics for PCF with unbounded nondeterminism based on sequential algorithms and explores the role played by continuity; however, this semantics is not fully abstract. Laird's work is interesting because it shows that we can obtain a traditional adequacy proof for a semantics with one-sided continuity: composition is continuous with respect to functions, but not with respect to arguments.

The idea of using some constrained version of continuity to prove adequacy for countable nondeterminism goes back to Plotkin's work on power-domains [Apt and Plotkin 1981]. A crucial observation in both [Apt and Plotkin 1981] and [Laird 2015] is that this sort of proof requires a Hoare logic in which we can reason about all the countable ordinals. We cannot use these techniques here, however, because our composition is not continuous on either side.

1.2 Contributions

The main concepts of game semantics and the steps we take to establish full abstraction are well-established, with a few exceptions. The idea of including infinitary information in strategies is not new, but this particular presentation, though closely related to that of [Levy 2008], is the first example of using the technique to establish a full abstraction result for may and must testing.

There are two points in the traditional Full Abstraction proof that depend on composition being continuous, and we have had to come up with ways of getting round them. The easier of the two is that continuity is necessary for the usual proof that we can derive Full Abstraction from definability of compact strategies; in order to get around this, we have had to appeal instead to the stronger *universality* result of [Hyland and Ong 2000], that says that every *recursive* strategy is definable.

For the proof of adequacy, we have had to come up with a new technique, which can be thought of as a kind of synthesis between the two usual methods of proving adequacy – one involving logical relations and the other using more hands-on operational techniques. We do this by separating out the deterministic, continuous part of the strategy from the nondeterministic, discontinuous part. Using the stateful language, we can simulate individual evaluation paths of a nondeterministic program using a deterministic device that corresponds to the idea of 'mocking' a random number generator for testing purposes. This allows us to appeal to the adequacy result for deterministic Idealized Algol. We then rely on more combinatorial techniques in order to factor the nondeterminism back in.

This new technique is actually very generally applicable. In the last section of the paper, we show that it may be used to prove adequacy for models of nondeterministic PCF under very mild assumptions. The Tsukada-Ong model, for example, satisfies these assumptions, allowing us to obtain an adequacy result for PCF with countable nondeterminism.

2 Idealized Algol with Countable Nondeterminism

We describe a simple type theory and operational semantics for Idealized Algol with countable nondeterminism. This is similar to the approach adopted in [Harmer and McCusker 1999], which extends Idealized Algol with *finite* nondeterminism. The types of our language are defined inductively as follows:

$$T ::= \mathtt{nat} \mid \mathtt{com} \mid \mathtt{Var} \mid T \to T\,.$$

Meanwhile, the terms are those given in [Abramsky and McCusker 1999], together with the nondeterministic choice:

$$\begin{aligned} M ::= {} & x \mid \lambda x.M \mid M\ M \mid \mathbf{Y}_T \mid \\ & \mathtt{n} \mid \mathtt{skip} \mid \mathtt{suc} \mid \mathtt{pred} \mid \\ & \mathtt{If0} \mid \_\,;\,\_ \mid \_ \mathrel{:=} \_ \mid \\ & \mathtt{@} \mid \mathtt{new}_T \mid \mathtt{mkvar} \mid ?\,. \end{aligned}$$

The typing rule for ? is $\Gamma \vdash ?: \mathtt{nat}$. We shall use the letter v to range over variables of type $\mathtt{Var}$.

We define a small-step operational semantics for the language; this presentation is equivalent to the big-step semantics given in [Harmer and McCusker 1999], except with a different rule for the countable rather than finite nondeterminism.

$$\frac{}{\langle s, (\lambda x.M)\ N\rangle \longrightarrow \langle s, M[N/x]\rangle} \qquad \frac{}{\langle s, \mathbf{Y}_T M\rangle \longrightarrow \langle s, M(\mathbf{Y}_T M)\rangle} \qquad \frac{}{\langle s, \mathsf{suc}\ \mathsf{n}\rangle \longrightarrow \langle s, \mathsf{n}+1\rangle} \qquad \frac{}{\langle s, \mathsf{pred}\ \mathsf{n}\rangle \longrightarrow \langle s, 0 \sqcup (\mathsf{n}-1)\rangle}$$

$$\frac{}{\langle s, \mathtt{If0}\ \mathtt{0}MN\rangle \longrightarrow \langle s, M\rangle} \qquad \frac{}{\langle s, \mathtt{If0}\ (\mathsf{n}+1)MN\rangle \longrightarrow \langle s, N\rangle} \qquad \frac{}{\langle s, @(\mathsf{mkvar}\ MN)\rangle \longrightarrow \langle s, M\rangle} \qquad \frac{}{\langle s, (\mathsf{mkvar}\ MN) \coloneqq L\rangle \longrightarrow \langle s, N\ L\rangle}$$

$$\frac{}{\langle s, v \coloneqq n\rangle \longrightarrow \langle\langle s \mid v \mapsto n\rangle, \mathsf{skip}\rangle} \qquad s(v) = n\ \frac{}{\langle s, @v\rangle \longrightarrow \langle s, n\rangle} \qquad \frac{}{\langle s, \mathsf{skip}; M\rangle \longrightarrow \langle s, M\rangle} \qquad \frac{}{\langle s, \mathsf{new}_T \lambda v.M\rangle \longrightarrow \langle\langle s \mid v \mapsto 0\rangle, M\rangle}$$

$$\frac{\langle s, M\rangle \longrightarrow \langle s, M'\rangle}{\langle s, E[M]\rangle \longrightarrow \langle s, E[M']\rangle} \qquad \frac{}{\langle s, ?\rangle \longrightarrow \langle s, \mathsf{n}\rangle}\ n \in \mathbb{N}$$

Figure 1. Small-step operational semantics for Idealized Algol with countable nondeterminism

First, we define a Felleisen-style *evaluation context* E inductively as follows.

$$\begin{aligned} E ::= -\mid EM \mid \mathsf{suc}\ E \mid \mathsf{pred}\ E \mid \mathtt{If0}\ E \mid \\ E;\ \_\ \mid E \coloneqq\ \_\ \mid @E\ \_\ \mid \mathsf{mkvar}\ E \mid \mathsf{new}_T E \end{aligned}$$

We then give the appropriate small-step rules in Figure 1. In each rule, $\langle s, M\rangle$ is a *configuration* of the language, where M is a term, and s is a *store*; i.e., a function from the set of variables free in M to the set of natural numbers. If s is a store and v a variable, we write $\langle s \mid v \mapsto n\rangle$ for the state formed by updating the value of the variable v to n.

If $\langle \varnothing, M\rangle$ is a configuration with empty store, we call M a *closed term.* Given a closed term M of ground type `com` or `nat`, we write that $M \Downarrow x$ (where x = skip in the `com` case and is a natural number in the `nat` case) if there is a finite sequence $M \longrightarrow M_1 \longrightarrow \cdots \longrightarrow M_n = x$. If there is no infinite sequence $M \longrightarrow M_1 \longrightarrow M_2 \longrightarrow \cdots$, then we say that M *must converge*, and write $M \Downarrow^{\text{must}}$. In general, we refer to a (finite or infinite) sequence $M \longrightarrow M_1 \longrightarrow \cdots$ that either terminates at an observable value or continues forever as an *evaluation* π of M. Since the only case where we have any choice in which rule to use is the application of the rule for ?, π may be completely specified by a finite or infinite sequence of natural numbers.

Let T be an Idealized Algol type, and let $M, N\colon T$ be closed terms. Then we write $M \sqsubseteq_{m\&m} N$ if for all compatible contexts $C[-]$ of ground type we have

$$\begin{gathered} C[M] \Downarrow V \Rightarrow C[N] \Downarrow V \\ C[M] \Downarrow^{\text{must}} \Rightarrow C[N] \Downarrow^{\text{must}} \end{gathered}$$

We write $M \equiv_{m\&m} N$ if $M \sqsubseteq_{m\&m} N$ and $N \sqsubseteq_{m\&m} M$.

3 Game Semantics

We will use the Hyland-Ong version of Game Semantics, as in [Abramsky and McCusker 1999].

3.1 Arenas

An *arena* is given by a triple $A = (M_A, \lambda_A, \vdash_A)$, where

- M_A is a countable set of moves,
- $\lambda_A\colon M_A \to \{O,P\} \times \{Q,A\}$ designates each move as either an O-*move* or a P-*move*, and as either a *question* or an *answer.* We define $\lambda_A^{OP} = \mathrm{pr}_1 \circ \lambda_A$ and $\lambda_A^{QA} = \mathrm{pr}_2 \circ \lambda_A$. We also define $\neg\colon \{O,P\} \times \{Q,A\} \to \{O,P\} \times \{Q,A\}$ to be the function that reverses the values of O and P while leaving $\{Q,A\}$ unchanged.
- $\vdash_A$ is an *enabling relation* between $M_A + \{*\}$ and M_A satisfying the following rules:
  - If $a \vdash_A b$ and $a \neq b$, then $\lambda_A^{OP}(a) \neq \lambda_A^{OP}(b)$.
  - If $* \vdash_A a$, then $\lambda_A(a) = OQ$ and $b \nvdash_A a$ for all $b \in M_A$.
  - If $a \vdash_A b$ and b is an answer, then a is a question.

  We say that a move $a \in M_A$ is *initial* in A if $* \vdash_A a$.

Our base arenas will be the *flat arenas* for the types `nat` and `com`. Given a set X, the flat arena on X is the arena with a single O-question q and a P-answer x for each $x \in X$, where $* \vdash q$ and $q \vdash x$ for each x. The denotation of the type `nat` will be the flat arena $\mathbb{N}$ on the set of natural numbers, while the denotation of the type `com` will be the flat arena $\mathbb{C}$ on the singleton set $\{a\}$.

Given an arena A, a *justified string* in A is a sequence s of moves in A, together with *justification pointers* that go from move to move in the sequence. The justification pointers must be set up in such a way that every non-initial move m in s has exactly one justification pointer going back to an earlier move n in s such that $n \vdash_A m$. We say that n *justifies* m. It is easy to see that every justified string must begin with an initial move, an hence with an O-question.

A *legal play* s is a justified string in A that strictly alternates between O-moves and P-moves and is such that the corresponding QA-sequence formed by applying λ_A^{QA} to moves is well-bracketed. We write L_A for the set of legal plays in A.

3.2 Games and strategies

We follow the approach taken by Abramsky and McCusker [Abramsky and McCusker 1999] – a middle road between the *arenas* of Hyland and Ong and the *games* of [Abramsky et al. 2000] that makes the linear structure more apparent.

Let s be a legal play in some arena A. If m and n are moves in s such that there is a chain of justification pointers leading from m back to n, we say that n *hereditarily justifies* m. Given some set S of initial moves in s, we write $s|_S$ for the subsequence of s made up of all those moves that are hereditarily justified by some move in S.

A *game* is a tuple $A = (M_A, \lambda_A, \vdash_A, P_A)$, where $(M_A, \lambda_A, \vdash_A)$ is an arena and P_A is a non-empty prefix-closed set of legal plays in that arena such that if $s \in P_A$ and I is a non-empty set of initial moves in s, then $s|_I \in P_A$.

Our base games will be the games $\mathbb{N}$ and $\mathbb{C}$ on the arenas of the same names, where $P_{\mathbb{N}} = \{\epsilon, q\} \cup \{qn : n \in \mathbb{N}\}$ and $P_{\mathbb{C}} = \{\epsilon, q, qa\}$.

3.2.1 Connectives

Let A, B be games. Then we may define games $A \times B$, $A \otimes B$, $A \multimap B$ and $!A$ as in [Abramsky and McCusker 1999]. As an example, we give the definition of $A \multimap B$:

$$\begin{array}{rcl} M_{A\multimap B} & = & M_A + M_B \,. \\ \lambda_{A\multimap B} & = & [\neg \circ \lambda_A, \lambda_B] \,. \\ * \vdash_{A\multimap B} n & \Leftrightarrow & * \vdash_B m \,. \\ m \vdash_{A\multimap B} n & \Leftrightarrow & \begin{array}{l} m \vdash_A n \text{ or } m \vdash_B n \\ \text{or (for } m \neq *) * \vdash_B m \text{ and } * \vdash_A n \,. \end{array} \\ P_{A\multimap B} & = & \{s \in L_{A\multimap B} \,:\, s|_A \in P_A \text{ and } s|_B \in P_B\} \,. \end{array}$$

3.2.2 Modelling countable nondeterminism

Our definition of a strategy will be modelled upon that given in [Harmer and McCusker 1999]. We model nondeterministic computations by relaxing the determinism constraint on strategies – so player P may have multiple replies to any given O-move.

In addition, we have to keep track of any possible divergence in the computation; this is so we can distinguish terms such as

$$\texttt{If0}\ ?\ \Omega\ \texttt{0} \qquad\qquad \texttt{0},$$

where the term on the right must converge (to `0`), while the term on the left has a possible divergence. The traditional way of representing divergences in game semantics is by a partiality in the strategy; i.e., an O-move to which P has no reply, but this doesn't work here: `If0` ? Ω `0` and `0` both denote the strategy with maximal play q 0; the partiality in the former is 'hidden' by the move 0.

To fix this problem, we follow [Harmer and McCusker 1999] by modelling a strategy as a pair (T_σ, D_σ), where T_σ is a nondeterministic strategy in the usual sense and D_σ is a set of O-plays after which there is a possibility of divergence.

Tracking divergences explicitly in this way requires some care when we compose strategies using 'parallel composition plus hiding'. Specifically, we need to be able to add new divergences into strategies when they arise through 'infinite chattering' or *livelock*. For example, the denotation of the term

$$M = \mathbf{Y}_{\texttt{nat}\to\texttt{nat}}(\lambda f.\lambda n.n; (f n))$$

is given by a total strategy, without divergences: namely the strategy μ with plays of the form shown in Figure 2(a). However, when we compose this strategy with any total strategy for $\mathbb{N}$ on the left, we expect the resulting strategy to contain divergences, since the term $M\mathsf{n}$ diverges for any n. Semantically, this corresponds to the fact that we have a legal interaction $q\ q\ n\ q\ n\ \cdots$ with an infinite tail in $\mathbb{N}_1$; when we perform 'hiding' by restricting the interaction to $\mathbb{N}$, we have no reply to the initial move q.

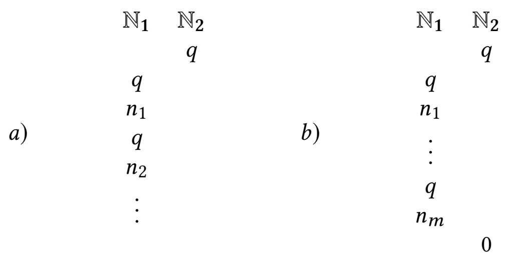

Figure 2. Finite plays alone are not sufficient to distinguish between terms of a language with countable nondeterminism.

The approach adopted in [Harmer and McCusker 1999] is to check specifically for infinite chattering between strategies $\sigma\colon A \multimap B$ and $\tau\colon B \multimap C$ by checking whether there is an infinite increasing sequence of interactions between σ and τ with an infinite tail in B. If there is such a sequence, then it restricts to some O-position in $\sigma;\tau$ and we add in a divergence at that position.

This works very satisfactorily for finite nondeterminism, but not at all for countable nondeterminism. To see why, consider the term

$$N = \mathbf{Y}_{\texttt{nat}\to\texttt{nat}\to\texttt{nat}}(\lambda g.\lambda mn.\texttt{If0}\ m\ \texttt{0}\ n; (g\ (\texttt{pred}\ m)\ n))?$$

This term first chooses a natural number m, and then reads from its input n for a total of m times before eventually returning `0`. Thus, its denotation is the strategy ν with maximal plays of arbitrary length of the form shown in Figure 2(b). Note that this strategy strictly contains the strategy μ that we considered before, and therefore that the denotation of

$$\texttt{If0}\ ?MN$$

has the same denotation as N, even though for any n, $M\mathsf{n} \not\Downarrow^{\text{must}}$, while $N\mathsf{n} \Downarrow^{\text{must}}$. Moreover, if we try to compose $[\![N]\!]$ with the strategy on $\mathbb{N}$ that always returns 1, then we end up with an infinite increasing sequence of positions, which triggers the introduction of a divergent play into the composite strategy – even though no divergence occurs in the evaluation of N.

Aside from showing that this model cannot possibly be sound, this example actually leads to composition not being associative if we naively extend the Harmer-McCusker model from finite to infinite nondeterminism (e.g., see [Harmer 1999, 4.4.1]).

Somehow, the crucial point is that we need to distinguish between terms like M, which contain infinite sequences of moves, and terms like N, which contain arbitrarily large finite sequences of moves. The way that we do this is by making the infinite sequences of moves explicit in our strategies, in the style of [Roscoe 1993] and [Levy 2008]. When we use this technique, the denotation of M will contain an infinite sequence, while the denotation of N will contain arbitrarily long finite sequences, but no infinite sequences.

The games in our model will be the same as those that we considered in the last section, but our definition of a strategy will change.

3.2.3 Strategies

Given an arena A, we define an *infinite justified string* in the obvious way. We define $\overline{P_A}$ to be P_A together with the set of all those infinite justified sequences that have all finite prefixes in P_A.

Let A be a game. A *strategy* σ for A is a pair (T_σ, D_σ), where:

- T_σ is a non-empty prefix-closed subset of $\overline{P_A}$ such that if $s \in T_\sigma$ is a P-position and $sa \in P_A$ then $sa \in T_\sigma$.
- $D_\sigma \subseteq \overline{P_A}$ is a postfix-closed set of plays in $\overline{P_A}$ that either end with an O-move or are infinite. We require D_σ to obey the following rules:

  Divergences come from plays If $d \in D_\sigma$ then there exists some $s \sqsubseteq d$ such that $s \in T_\sigma \cap D_\sigma$.

  Diverge-or-reply If $s \in T_\sigma$ is an O-position, then either $s \in D_\sigma$ or $sa \in T_\sigma$ for some legal play sa.

  Infinite positions are divergent If $s \in T_\sigma$ is infinite, then $s \in D_\sigma$.

3.2.4 Composition of strategies

Given games A, B, C, we define a justified string over A, B, C to be a sequence $\mathfrak{s}$ of moves with justification pointers from all moves except the initial moves in C. Given such a string, we may form the restrictions $\mathfrak{s}|_{A,B}$ and $\mathfrak{s}|_{B,C}$ by removing all moves in either C or A, together with all justification pointers pointing into these games. We define $\mathfrak{s}|_{A,C}$ to be the sequence formed by removing all moves from B from $\mathfrak{s}$ and all pointers to moves in B, *unless* we have a sequence of pointers $a \longrightarrow b \longrightarrow c$, in which case we replace them with a pointer $a \longrightarrow c$.

We call such a sequence $\mathfrak{s}$ a *legal interaction* if $\mathfrak{s}|_{A,B} \in P_{A \multimap B}$, $\mathfrak{s}|_{B,C} \in P_{B \multimap C}$ and $\mathfrak{s}|_{A,C} \in P_{A \multimap C}$. We write $\mathrm{int}_\infty(A, B, C)$ for the set of (possibly infinite) legal interactions between A, B and C.

Now, given strategies $\sigma\colon A \multimap B$ and $\tau\colon B \multimap C$, we define

$$T_\sigma \| T_\tau = \{\mathfrak{s} \in \mathrm{int}_\infty(A, B, C) : \mathfrak{s}|_{A,B} \in T_\sigma,\ \mathfrak{s}|_{B,C} \in T_\tau\}\,,$$

and then set

$$T_{\sigma;\tau} = \{\mathfrak{s}|_{A,C} : \mathfrak{s} \in T_\sigma \| T_\tau\}\,.$$

As for divergences in $\sigma; \tau$, our approach is actually simpler than that in [Harmer and McCusker 1999]; we set

$$D_\sigma \lightning D_\tau = \left\{ \mathfrak{s} \in \mathrm{int}_\infty(A, B, C) \;\middle|\; \begin{array}{l} \textbf{either } \mathfrak{s}|_{A,B} \in D_\sigma \\ \quad \text{and } s|_{B,C} \in T_\tau \\ \textbf{or } \mathfrak{s}|_{A,B} \in T_\sigma \\ \quad \text{and } s|_{B,C} \in D_\tau \end{array} \right\}.$$

We then set

$$D_{\sigma;\tau} = \mathrm{pocl}_{A \multimap C}\{\mathfrak{s}|_{A,C} : \mathfrak{s} \in D_\sigma \lightning D_\tau\}\,,$$

where $\mathrm{pocl}\, X$ denotes the *postfix closure* of X; i.e., the set of all O-plays in $P_{A \multimap C}$ that have some prefix in X.

Note that there is no need to consider separately, as Harmer and McCusker do, divergences that arise through 'infinite chattering': in our model, a case of infinite chattering between strategies σ and τ is itself an (infinite) legal interaction between the two strategies, which is necessarily divergent (because it is infinite) and therefore gives rise to some divergence in $\sigma; \tau$.

We need to impose one more condition on strategies:

Definition 3.1. Let σ be a strategy for a game A. We say that σ is *complete* if $T_\sigma = \overline{T_\sigma}$; i.e., T_σ contains an infinite position s if it contains every finite prefix of s.

Any finite-nondeterminism strategy in the sense of [Harmer and McCusker 1999] may be interpreted as a complete strategy by enlarging it with all its infinite limiting plays. However, when we introduce countable nondeterminism, we also introduce strategies that are not complete. For example, the strategy ν that we mentioned above has an infinite increasing sequence of plays $q0 \sqsubseteq q0q0 \sqsubseteq \cdots$, but has no infinite play corresponding to its limit. Nonetheless, we do not want to allow arbitrary strategies: for example, the strategy μ above should include the infinite play $qq0q0\ldots$; the strategy μ° formed by removing this infinite play has no meaning in our language. Indeed, if we compose μ° with the strategy $\mathtt{0}$ for $\mathbb{N}$ on the left, then the resulting strategy does not satisfy diverge-or-reply. The difference with ν is that every play $qq0 \cdots q0 \in T_\nu$ may be completed in ν by playing the move 0 on the right. In other words, ν is the union of complete strategies, while μ° is not.

Definition 3.2. Let σ be a strategy for a game A. We say that σ is *locally complete* if it may be written as the union of countably many complete strategies; i.e., there exist σ_n such that $T_\sigma = \bigcup T_{\sigma_n}$ and $D_\sigma = \bigcup D_{\sigma_n}$.

From now on, we will use 'strategy' to mean *locally complete strategy*.

We need to show that the composition of locally complete strategies is locally complete. Note that the composition of *complete* strategies is not necessarily complete: for example, our term N above can be written as N' ?, where N' is a deterministic term with complete denotation ν'. Then we have $\nu = \top_{\mathbb{N}}; \nu'$, but ν is not complete. However, we can show that the composition of *deterministic* complete strategies is complete; since a locally complete strategy may always be written as the union of complete deterministic strategies, this is sufficient to show that the composition of locally complete strategies is locally complete.

Definition 3.3. We say that a strategy σ for a game A is *deterministic* if

- it is complete;
- whenever sab, sac are P-plays in T_σ we have $b = c$ and the justifier of b is the justifier of c;
- If $s \in D_\sigma$ then either s is infinite or there is no a such that $sa \in T_\sigma$.

Lemma 3.4. *Let A, B, C be games and let $\sigma\colon A \multimap B$, $\tau\colon B \multimap C$ be deterministic complete strategies. Then $\sigma; \tau$ is complete.*

Proof. The proof relies on a lemma from [Hyland and Ong 2000] that states (in our language) that if σ and τ are deterministic strategies and $s \in T_{\sigma;\tau}$ then there is a unique minimal $\mathfrak{s} \in T_\sigma \| T_\tau$ such that $\mathfrak{s}|_{A,C} = s$. That means that if $s_1 \sqsubseteq s_2 \sqsubseteq \cdots$ is an infinite increasing sequence of plays in $T_{\sigma;\tau}$, with infinite limit s, then there is a corresponding infinite increasing sequence of legal interactions $\mathfrak{s}_1 \sqsubseteq \mathfrak{s}_2 \sqsubseteq \cdots$. Then the limit of this sequence is an infinite legal interaction $\mathfrak{s}$ and we must have $\mathfrak{s}|_{A,B} \in \sigma$, $\mathfrak{s}|_{B,C} \in \tau$ by completeness of σ and τ. Therefore, $s = \mathfrak{s}|_{A,C} \in T_{\sigma;\tau}$. □

Corollary 3.5. *The composition of strategies $\sigma : A \multimap B$ and $\tau : B \multimap C$ is a well-formed strategy for $A \multimap C$.*

Proof. The only tricky point is establishing that diverge-or-reply holds for $\sigma;\tau$. Again, it is sufficient to prove this in the case that σ and τ are deterministic and complete. Then it essentially follows from the argument used in [Abramsky and Jagadeesan 1994] that shows that a partiality at an O-position $s \in T_{\sigma;\tau}$ must arise either from a partiality in T_σ or T_τ or from 'infinite chattering' between σ and τ. In the first case, the diverge-or-reply rule for σ and τ gives us a divergence at s in $\sigma;\tau$. In the second case, an infinite chattering between σ and τ corresponds to an infinite interaction $\mathfrak{s} \in \mathrm{int}(A, B, C)$ ending with infinitely many moves in B such that $\mathfrak{s}|_{A,C} = s$. Completeness for σ and τ tells us that $\mathfrak{s}|_{A,B} \in D_\sigma$ and $\mathfrak{s}|_{B,C} \in D_\tau$ and therefore that $\mathfrak{s}|_{A,C} \in D_{\sigma;\tau}$. □

Our proof for Corollary 3.5 really makes use of the fact that a locally complete strategy is *lively* in the sense of [Levy 2008]; i.e., locally deterministic. Our definition is slightly stronger than liveliness, because it insists that the union of complete strategies be *countable*. This will be essential to our definability result.

3.2.5 Associativity of composition

In fact, the proof of associativity of composition is pretty much the same in our model as it is in any other model of game semantics. However, it is worth saying a few words about it, since the model obtained by naively extending the Harmer-McCusker model to unbounded nondeterminism does not have an associative composition. The point is that there is not really a problem with associativity itself, but rather that this naive model gives the wrong result for the composition of strategies with infinite nondeterminism. For example, if ν is the strategy we defined above, and $\mathtt{0}$ is the 'constant 0' strategy on $\mathbb{N}$, then $\mathtt{0};\nu$ has a divergence in the naive model, because the strategies $\mathtt{0}$ and ν appear to be engaged in infinite chattering. In our model, we have fixed that problem, because the strategy ν contains no infinite plays, and so no divergences arise in the composition.

3.3 A symmetric monoidal closed category

Given a game A, we define a strategy id_A on $A \multimap A$, where T_{id_A} is given by

$$\{s \in P_{A_1 \multimap A_2} \ : \ \text{for all even-length } t \sqsubseteq s,\ t|_{A_1} = t|_{A_2}\}\,,$$

where we distinguish between the two copies of A by calling them A_1 and A_2, and where D_σ is the set of all infinite plays in T_σ. This is an identity for the composition we have defined, and so we get a category $\mathcal{G}_{ND}$ of games and nondeterministic strategies. Moreover, the connectives $\otimes$ and $\multimap$ exhibit $\mathcal{G}_{ND}$ as a symmetric monoidal closed category.

$\mathcal{G}_{ND}$ has an important subcategory $\mathcal{G}_D$ of deterministic complete strategies; this category is isomorphic to the category considered in [Abramsky and McCusker 1999].

3.4 A Cartesian closed category

We follow the construction given in [Abramsky and McCusker 1999], using the connectives ! and × to build a Cartesian closed category $\mathcal{G}^!_{ND}$ from $\mathcal{G}_{ND}$ whose objects are the well-opened games in $\mathcal{G}_{ND}$ and where a morphism from A to B in $\mathcal{G}^!_{ND}$ is a morphism from $!A$ to B in $\mathcal{G}_{ND}$.

This is very similar to the construction of a co-Kleisli category for a linear exponential comonad, but certain technical issues relating to well-openedness prevent us from presenting it in this way.

3.5 Constraining strategies

Given a non-empty justified string s in an arena A, we define the *P-view* $\ulcorner s \urcorner$ of s inductively as follows.

$$\begin{aligned}
\ulcorner sm \urcorner &= m\,, && \text{if } m \text{ is initial;}\\
\ulcorner sntm \urcorner &= \ulcorner s \urcorner nm\,, && \text{if } m \text{ is an } O\text{-move and}\\
& && n \text{ justifies } m;\\
\ulcorner sm \urcorner &= \ulcorner s \urcorner m\,, && \text{if } m \text{ is a } P\text{-move.}
\end{aligned}$$

We say that a play sm ending in a P-move is *P-visible* if the justifier of m is contained in $\ulcorner m \urcorner$. We say that a strategy σ for a game A is *visible* if every P-position $s \in T_\sigma$ is P-visible. It can be shown that the composition of visible strategies is visible, and that we can build a Cartesian closed category using our exponential.

The resulting category $\mathcal{G}^!_{D,vis}$ of games and deterministic visible strategies is a fully abstract model of Idealized Algol [Abramsky and McCusker 1999].

3.6 Enumerated games and recursive strategies

Most full abstraction results go via a definability result that says that all *compact* strategies are definable [Curien 2007]. However, deducing full abstraction from compact definability makes essential use of continuity properties that are absent when we deal with countable nondeterminism. We will therefore need to appeal to a stronger result – that of *universality*, which states that *every* strategy is definable. Clearly, universality does not hold for any of our categories of games – for example, there are many non-computable functions $\mathbb{N} \to \mathbb{N}$. However, Hyland and Ong proved in [Hyland and Ong 2000] that every *recursively presentable* innocent strategy is PCF-definable.

In order to define recursively presentable strategies, we need to work with *enumerated games*; i.e., games where the set of moves comes with an enumeration to the natural numbers. Clearly our base games $\mathbb{N}$ and $\mathbb{C}$ can be enumerated, as can the tensor product, linear implication, exponential and product of games. We can then define a *recursive* strategy to be one that is recursively presentable with respect to that enumeration of moves.

We shall tacitly assume that the games in our category $\mathcal{G}_{ND}$ come with an enumeration so that we can reason about recursive strategies later. For any game that is the denotation of an Idealized Algol type, this enumeration will be as we have just described.

The relevant result pertaining to recursive strategies is as follows.

Proposition 3.6 (Recursive Universality for Idealized Algol)**.** *Let S be an Idealized Algol type and let $\sigma\colon [\![S]\!]$ be a recursive strategy. Then there exists a term $M\colon S$ of Idealized Algol such that $\sigma = [\![M]\!]$.*

Proof. This follows from the corresponding results for PCF, together with the *innocent factorization* result of [Abramsky and McCusker 1999]. See also [Murawski and Tzevelekos 2013]. □

3.7 Deterministic Factorization

Our definability results will hinge on a *factorization theorem*, showing that every nondeterministic strategy may be written as the composition of a deterministic strategy with the nondeterministic 'oracle' $\top_{\mathbb{N}}$. We can then deduce universality from universality in the model of deterministic Idealized Algol.

Note that our result is a bit simpler than the corresponding result in [Harmer and McCusker 1999]; this is because it is easier to model a countable source of nondeterminism than a 'finite but arbitrarily large' source.

Proposition 3.7. *Let $\sigma\colon I \to A$ be a strategy for a game A in $\mathcal{G}_{ND}$. Then we may write σ as $\top_{\mathbb{N}}; \mathrm{Det}(\sigma)$, where $\mathrm{Det}(\sigma)\colon !\mathbb{N} \to A$ is a deterministic strategy and $\top_{\mathbb{N}}$ is the strategy for $!\mathbb{N}$ that is given by*

- $T_{\top_{\mathbb{N}}} = P_{!\mathbb{N}}$.
- *$D_{\top_{\mathbb{N}}}$ is the set of infinite positions in $T_{\top_{\mathbb{N}}}$.*

Proof. We begin by fixing an injection code_A from the set of P-moves in A into the natural numbers. In the enumerated case, this is given to us already.

We first assume that the strategy σ is complete. Then the strategy $\mathrm{Det}(\sigma)$ is very easy to describe. For each O-position $sa \in T_\sigma$, we have some set B of possible replies to sa, which we order as $b_1, b_2, \cdots$, where $\mathrm{code}_A(b_1) < \mathrm{code}_A(b_2) < \cdots$. We insert a request to the oracle for a natural number; then, depending on her answer j, we play the next move as follows:

- If $0 < j \le \mathrm{code}_A(b_1)$, then play b_1.
- If $\mathrm{code}_A(b_n) < j \le \mathrm{code}_A(b_{n+1})$ then play b_{n+1}.
- If $j = 0$ and $sa \in D_\sigma$, then play nothing, and put the resulting play inside $D_{\mathrm{Det}(\sigma)}$. Otherwise, play b_1.

Lastly, we close under limits to make the strategy $\mathrm{Det}(\sigma)$ complete. $\mathrm{Det}(\sigma)$ is clearly deterministic. Checking that $\top_{\mathbb{N}}; \mathrm{Det}(\sigma) = \sigma$ is easy for finite plays; for infinite plays, it follows by completeness of σ.

Lastly, if σ is the union of complete strategies $\sigma_1, \sigma_2, \cdots$, we insert an additional request to the oracle immediately after the very first move by player O; after receiving a reply k, we play according to σ_k. □

Note that in the recursive case, $\mathrm{Det}(\sigma)$ is clearly recursively presentable if σ is. Furthermore, if σ is visible, then so is $\mathrm{Det}(\sigma)$.

4 Full abstraction

4.1 Denotational Semantics

The category in which we shall model our language is the category $\mathcal{G}^!_{ND,vis}$ – the Cartesian closed category of (enumerated) games with nondeterministic visible strategies. We have a natural embedding $\mathcal{G}^!_{D,vis} \hookrightarrow \mathcal{G}^!_{ND,vis}$, and we know that $\mathcal{G}^!_{D,vis}$ is a universal and fully abstract model of Idealized Algol.

Any term $M\colon T$ of Idealized Algol with countable nondeterminism may be written as $M = C[?]$, where C is a multi-holed context not involving the constant ?. Then the term $\lambda n.C[n]$ is a term of Idealized Algol, and therefore has a denotation $!\mathbb{N} \to [\![T]\!]$ as in [Abramsky and McCusker 1999]. We define the denotation of M to be given by the composite

$$I \xrightarrow{\top_{\mathbb{N}}} !\mathbb{N} \xrightarrow{[\![\lambda n.C[n]]\!]} [\![T]\!]$$

In other words, we interpret the constant ? using the strategy $\top_{\mathbb{N}}$ for $\mathbb{N}$.

4.2 Computational Adequacy

The *computational adequacy* result for our model can be stated as follows.

Proposition 4.1 (Computational Adequacy)**.** *Let $M\colon$ $\mathtt{com}$ be a closed term of Idealized Algol with countable nondeterminism. $M \Downarrow \mathsf{skip}$ if and only if $qa \in T_{[\![M]\!]}$. $M \Downarrow^{must}$ if and only if $D_{[\![M]\!]} = \varnothing$.*

Traditional proofs of computational adequacy using logical relations make essential use of the continuity of composition with respect to a natural ordering on strategies (see, for example, [Harmer and McCusker 1999] and [Harmer 1999] for the finite nondeterminism case). In our case, since composition is not continuous in the language itself, we cannot use this technique. In order to prove adequacy, we use a new technique that involves using a deterministic stateful construction to model the nondeterminism inside a deterministic world in which continuity holds. To do this, we shall return to the concept of an *evaluation* π of a term as a sequence of natural numbers that we use to replace the constant ?.

Lemma 4.2. *Let $M = C[?]$ be a term of type $\mathtt{com}$, where $C[-]$ is a multi-holed context of Idealized Algol. Write σ_M for the denotation of the term $\lambda n.C[n]$.*

- *If $M \Downarrow \mathsf{skip}$ then there exists some total deterministic strategy $\sigma\colon !\mathbb{N}$ such that $qa \in T_{\sigma;\sigma_M}$.*
- *If $M \not\Downarrow^{must}$ then there exists some total deterministic strategy $\sigma\colon !\mathbb{N}$ such that $D_{\sigma;\sigma_M} \neq \varnothing$.*

Proof. Let $n_1, \ldots, n_k, d$ be a finite sequence of natural numbers. We define an Idealized Algol term $N_{n_1,\ldots,n_k,d}\colon (\mathtt{nat} \to \mathtt{com}) \to \mathtt{com}$ to be the following.

$$\lambda f.\mathtt{new}_{\mathtt{nat}}(\lambda v.f(v := (suc\ @v); \mathtt{case}_{k+1}\ @v\ \Omega\ n_1 \cdots n_k d))\,.$$

Here, $\mathtt{case}_{k+1}\ a\ n_0\ \cdots\ n_k\ d$ is a new shorthand that evaluates to n_i if a evaluates to i, and evaluates to d if a evaluates to $j > k$. This term adds an extra variable v to the program; each time f is called, it increments the value of v and maps its value to one of the

n_i. The result is that when f calls its argument, it will receive n_1 the first time, n_2 the second and so on.

Now let π be a finite evaluation $\langle s, C[?]\rangle$ that converges to skip. Encode π as a sequence $n_1, \ldots, n_k$. Let d be some arbitrary number. Then we can show that the following term also converges to skip in the same way:

$$N_{n_1,\ldots,n_k,d}(\lambda n.C[n])\,.$$

The idea here is similar to one used in testing; we want to test the behaviour of a nondeterministic program, and to do so we *mock* the random number generator in order to simulate a particular evaluation path using purely deterministic programs.

If instead π is a finite evaluation of $\langle s, C[?]\rangle$ that diverges (but nevertheless only involves finitely many calls to the nondeterministic oracle), then the term $N_{n_1,\ldots,n_k,d}(\lambda n.C[n])$ will diverge according to the same execution path.

Digging into the construction of new within Idealized Algol, as given in [Abramsky and McCusker 1999], we see that for any term F of type $\mathtt{nat} \to \mathtt{com}$ the denotation of $N_{n_1,\ldots,n_k,d}F$ is given by the composite

$$I \xrightarrow{\mathsf{cell}_0} !\mathsf{Var} \xrightarrow{![\![\lambda v.v{:=}(\mathtt{suc}\ @v);\mathtt{case}_{k+1}\ @v\ \Omega\ n_1\cdots n_k d]\!]} !\mathbb{N} \xrightarrow{[\![F]\!]} \mathbb{C}\,.$$

We set σ_π to be the composite of the left two arrows. Observe that σ_π is the strategy with unique maximal infinite play as follows.

$$q\ n_1\ \cdots\ q\ n_k\ q\ d\ q\ d\ \cdots$$

Setting $F = \lambda n.C[n]$, we see that $[\![F]\!] = \sigma_M$. So, by adequacy for the Idealized Algol model, $qa \in T_{\sigma_\pi;\sigma_M}$ if and only if we have $N_{n_1,\ldots,n_k,d}(\lambda n.C[n]) \Downarrow$ skip, which is the case if and only if $M \Downarrow$ skip along the evaluation π. Similarly, $D_{\sigma_\pi;\sigma_M} \neq \varnothing$ if and only if $N_{n_1,\ldots,n_k,d}(\lambda n.C[n])$ diverges, which is equivalent to saying that M diverges along the evaluation π.

Lastly, we need to deal with the case that there is an infinite evaluation $\pi = n_1, n_2, \ldots$ of M that consults the nondeterministic oracle infinitely often. In this case, M must certainly diverge along the evaluation π. For each j, we define $\pi_n^{(j)}$ to be the strategy for $!\mathbb{N}$ corresponding to the term $N_{n_1,\ldots,n_j,\Omega}$. So $\pi_n^{(j)}$ has a unique finite maximal play

$$q\ n_1\ q\ n_2\ \cdots\ q\ n_j\ q\,,$$

at which point the strategy has a partiality.

Evaluation of the term $N_{n_1,\ldots,n_j,\Omega}(\lambda n.C[n])$ must diverge, since it will proceed according to the evaluation π and eventually reach the divergence (since π consults the oracle infinitely often). This implies that $D_{\sigma_\pi^{(j)};\sigma_M} \neq \varnothing$ for all j.

We define σ_π to be the least upper bound of the $\sigma_\pi^{(j)}$ (e.g., in the sense of [Harmer and McCusker 1999]). Since composition is continuous for deterministic (!) strategies, we deduce that $D_{\sigma_\pi;\sigma_M} \neq \varnothing$.

σ_π has plays of the form

$$q\ n_1\ q\ n_2\ \cdots\,,$$

and so it is total. □

From this result, we can prove the converse, which we will also need.

Lemma 4.3. *Let $M = C[?]$ be as before. Let $\sigma\colon !\mathbb{N}$ be a total deterministic strategy.*

- *If $qa \in T_{\sigma;\sigma_M}$ then $M \Downarrow$* skip.
- *If $D_{\sigma;\sigma_M} \neq \varnothing$ then $M \not\Downarrow^{must}$.*

Proof. Since σ is total and deterministic, it must have a maximal infinite play s_σ of the form

$$q\ m_1\ q\ m_2\ \cdots\,,$$

where $m_1, m_2, \ldots$ is some infinite sequence of natural numbers. If the strategy σ_M contains some play $\mathfrak{s}$ such that $\mathfrak{s}|_{!\mathbb{N}} = s$, then $\sigma = \sigma_\pi$ for some infinite evaluation π of M. Otherwise, let t be the maximal sub-play of s such that $\mathfrak{s}|_{!\mathbb{N}} = t$ for some $\mathfrak{s} \in \sigma_M$. Then, if we replace σ with the strategy σ' that plays according to t and subsequently plays $q\ d\ q\ d\ \cdots$ for our fixed value d, we will have $\sigma';\sigma_M = \sigma;\sigma_M$. In either case, $\sigma' = \sigma_\pi$ for some evaluation π of the term M.

Now suppose that there exists $\sigma\colon !\mathbb{N}$ such that $qa \in T_{\sigma;\sigma_M}$. We may assume that $\sigma = \sigma_\pi$ for some evaluation π of M. Therefore, $qa \in T_{\sigma_\pi;\sigma_M}$, which means that $M \Downarrow$ skip along the evaluation π. The corresponding statement for must convergence follows in the same way. □

We can now prove our computational adequacy result.

Proof of Proposition 4.1. We prove this for must convergence; the corresponding proof for may convergence is very similar.

Let $M = C[?]$ be a closed term, where C is a deterministic multi-holed context. Then, by Lemmas 4.2 and 4.3, we see that $M \not\Downarrow^{\mathrm{must}}$ if and only if there exists some total deterministic $\sigma\colon !\mathbb{N}$ such that $D_{\sigma;\sigma_M} \neq \varnothing$. By examining the definition of composition, we see that this is the case if and only if $D_{[\![M]\!]} = D_{\top_N;\sigma_M} \neq \varnothing$. □

We define *intrinsic equivalence of strategies* as follows. If σ, τ are two strategies for a game A, we say that $\sigma \sim \tau$ if for all test morphisms $\alpha\colon A \to \mathbb{C}$ we have $\sigma;\alpha = \tau;\alpha$. Having defined this equivalence, we may prove *soundness* in the usual way.

Theorem 4.4 (Soundness). *Let M, N be two closed terms of type T. If $[\![M]\!] \sim [\![N]\!]$ then $M \equiv_{m\&m} N$.*

For proving full abstraction , it is necessary to take the intrinsic quotient in order to identify, for example, the denotations of $\lambda n.\Omega$ and $\lambda n.\mathtt{If0}\ n\ \Omega\ \Omega$ of type $\mathtt{nat} \to \mathtt{nat}$. These terms are clearly observationally equivalent, but their denotations are not equal; for example, $q \in D_{[\![\lambda n.\Omega]\!]}$, but $q \notin D_{[\![\lambda n.\mathtt{If0}\ n\ \Omega\ \Omega]\!]}$.

The point here is that even though q is not explicitly a divergence in the second case, it is nonetheless impossible to prevent the strategy from eventually reaching a divergence.

Given a nondeterministic strategy σ for a game A, we may treat σ as a game in its own right (a sub-game of A). Moreover, for any $s \in T_\sigma$, we have a particular branch of that game in which play starts at s. We say that s is *unreliable* if player P has a strategy for the game starting at s that ensures that play eventually ends up in D_σ.

We then say that a strategy σ is *divergence-complete* if every unreliable point of σ is contained in D_σ. Every strategy σ can clearly be extended to a minimal divergence-complete strategy $\mathrm{dc}(\sigma)$; Murawski's explicit characterization of the intrinsic collapse [Murawski 2008], which may be applied to our model, essentially says that $\sigma \sim \tau$ if and only if $\mathrm{dc}(\sigma) = \mathrm{dc}(\tau)$.

4.3 Universality

Let S, T be Idealized Algol types and let $\sigma \colon S \to T$ be a recursive morphism in $\mathcal{G}^!_{ND,vis}$. We want to prove that σ is the denotation of some term.

By our nondeterministic factorization result, we know that $\sigma = \top_{\mathbb{N}}; \mathrm{Det}(\sigma)$, where $\mathrm{Det}(\sigma)$ is a deterministic strategy. By universality for $\mathcal{G}^!_{D,vis}$, we know that $\mathrm{Det}(\sigma) = [\![M]\!]$ for some closed term $M \colon S \to T$. Then $\sigma = \top_{\mathbb{N}}; \mathrm{Det}(\sigma) = [\![?]\!] ; [\![M]\!] = [\![M\ ?]\!]$.

4.4 Full abstraction

Theorem 4.5 (Full abstraction). *Let M, N be two closed terms of type T. If $M \equiv_{m\&m} N$ then $[\![M]\!] \sim [\![N]\!]$.*

Proof. Let $A = [\![T]\!]$. Suppose that $[\![M]\!] \not\sim [\![N]\!]$; so there is some strategy $\alpha \colon A \to \mathbb{C}$ such that $[\![M]\!] ; \alpha \neq [\![N]\!] ; \alpha$. We can clearly choose α to be recursively presentable; by universality, we have $\alpha = [\![P]\!]$ for some closed term P of type $T \to \mathtt{com}$. Then we have $[\![M]\!] ; [\![P]\!] \neq [\![N]\!] ; [\![P]\!]$; by computational adequacy, it follows that $M \not\equiv_{m\&m} N$. □

5 Nondeterministic PCF

We conclude by making a few remarks about the situation when our base deterministic language is PCF rather than Idealized Algol.

Hyland and Ong [Hyland and Ong 2000] gave a model of PCF using games and *innocent strategies*. If $sab \in P_A$, where b is a P-move such that sab is visible at b, and if $ta \in P_A$ is such that $\ulcorner sa \urcorner = \ulcorner ta \urcorner$, then it is possible to show that there is a unique way to extend ta by b such that $\ulcorner sab \urcorner = \ulcorner tab \urcorner$. We will abuse notation and refer to this play as tab, omitting the justification pointer from b.

If σ is a strategy for A that is deterministic and P-visible, we say that σ is *innocent* if and only if whenever $sab, ta \in \sigma$ are as above, we have $tab \in \sigma$.

Tsukada and Ong [Tsukada and Ong 2015], following Harmer [Harmer 1999], showed that traditional game semantics models are unsuitable for modelling *nondeterministic* PCF: while the definition of *visibility* of strategies can be extended to nondeterministic strategies without difficulty, innocence cannot.

Instead, they use a novel type of game semantics in which strategies are given not by sets of plays but by justified *trees* whose vertices are moves. The difference is subtle but important: from the point of view of countable nondeterminism it allows us to distinguish between terms that take an arbitrarily large finite number of steps to converge and terms that diverge, without needing to track infinitary information. For example, the denotation of our term M above has an infinite branch in it, while the denotation of N has infinitely many finite branches of arbitrary length.

The main advantage of the tree-based model, however, is that it gives rise to a satisfactory synthetic definition of a *nondeterministic innocent* strategy, based on sheaves. The main contribution of their paper is not to prove full abstraction for PCF with countable nondeterminism (even though that language may be interpreted within their category), but for PCF with finite nondeterminism.

Having ourselves given a fully abstract model of Idealized Algol with countable nondeterminism, we shall contribute to the process of 'completing the square' by giving a sketch of how we can use our techniques to prove adequacy for the Tsukada-Ong model of PCF with countable nondeterminism. A side-effect of this will be to give an alternative proof of adequacy for their model of PCF with finite nondeterminism.

The first observation to make is that our Lemmas 4.2 and 4.3 are not really statements about the category $\mathcal{G}_{ND,vis}$, but about the deterministic model $\mathcal{G}_{D,vis}$.

For example, the first parts of the lemmas says that if $C[-]$ is a deterministic Idealized Algol context, then $C[?] \Downarrow \mathrm{skip}$ if and only if there exists some total deterministic strategy $\sigma \colon !\mathbb{N}$ such that $qa \in T_{\sigma;[\![\lambda n.C[n]]\!]}$. Although we have mentioned a term $(C[?])$ living in nondeterministic Idealized Algol, this can be thought of as a statement about $C[-]$ itself, and therefore as a statement about the deterministic language. Moreover, the strategy σ and the denotation of $\lambda n.C[n]$ are both deterministic strategies, so we can interpret Lemmas 4.2 and 4.3 as telling us something about Idealized Algol and its game semantics.

We can therefore forget about our nondeterminism model, with its divergence sets and infinite plays (neither of which are part of the Tsukada-Ong model) and concentrate on the Abramsky-McCusker model of Idealized Algol. Note that although Lemmas 4.2 and 4.3 mention $T_{\sigma;\sigma_M}$ and $D_{\sigma;\sigma_M}$, we can rephrase both of these inside the deterministic model, since the strategies involved are both deterministic.

We want a version of Lemmas 4.2 and 4.3 that can be applied to PCF contexts and the game semantics model $\mathcal{G}_{inn}$ of PCF. Clearly, the lemma as currently stated does not hold in $\mathcal{G}_{inn}$, since the strategies σ that we define are not innocent in general.

Instead, we need to look at the combinatorial structure of the strategies. We observe that if $\sigma_M \colon !\mathbb{N} \to \mathbb{C}$ is a strategy, then there exists a deterministic strategy $\sigma \colon !\mathbb{N}$ such that $qa \in T_{\sigma;\sigma_M}$ if and only if there exists some $s \in T_{\sigma_M}$ such that $s|_{\mathbb{C}} = qa$.

We define a deterministic strategy $\sigma \colon A \multimap B$ to be *winning* if it is divergence-free and if for every O-play $s \in T_\sigma$ there exists some $t \in T_\sigma$ such that $s \sqsubseteq t$ and t ends with a move in B.

Then for any strategy σ_M, there exists a deterministic strategy $\sigma \colon !\mathbb{N}$ such that $D_{\sigma;\sigma_M} \neq \varnothing$ if and only if σ_M is not winning.

We recover the following version of Lemmas 4.2 and 4.3 which now applies to the interpretation of PCF in $\mathcal{G}_{inn}$:

Lemma 5.1. *Let $C[-]$ be a multi-holed context of deterministic PCF. Write $\sigma_C \colon !\mathbb{N} \to \mathbb{N}$ for the denotation of the term $\lambda n.C[n]$.*

- *$C[?] \Downarrow n$ if and only if there exists $s \in \sigma_C$ such that $s|_{\mathbb{N}} = qa$.*

- *$C \Downarrow^{must}$ if and only if σ_C is winning.*

Now we have an inclusion functor $J : \mathcal{G}^!_{inn} \hookrightarrow \mathcal{G}_{TO}$, where $\mathcal{G}_{TO}$ is the Tsukada-Ong category. Since the nondeterministic oracle ? may also be interpreted as a tree-strategy $\top_\mathbb{N}$ in $\mathcal{G}_{TO}$, this gives us a model of PCF with countable nondeterminism inside that category.

In order to prove adequacy for this model, we are only going to use two facts about the category $\mathcal{G}_{TO}$. The first fact is that the inclusion functor J preserves the structure of the model $\mathcal{G}_{inn}$; i.e., it is faithful and strong monoidal. The second fact will capture the properties of the morphism $\top_\mathbb{N} : I \to \mathbb{N}$ in $\mathcal{G}_{TO}$.

Definition 5.2. Let $\sigma, \tau : \mathbb{N}_1 \to \mathbb{N}_2$ be morphisms in $\mathcal{G}^!_{inn}$. We say that $\sigma \approx \tau$ if the following two conditions hold.

- For all $n \in \omega$, there exists $s \in \sigma$ such that $s|_{\mathbb{N}_2} = qn$ if and only if there exists $t \in \tau$ such that $t|_{\mathbb{N}_2} = qn$.
- σ is winning if and only if τ is winning.

Definition 5.3. We say that a Cartesian closed category C is a *sensible game semantics of PCF with countable nondeterminism* if there is a strong monoidal functor $J : \mathcal{G}^!_{inn} \to C$ and a morphism $\top_\mathbb{N} : 1 \to J(\mathbb{N})$ in C such that for all $\sigma, \tau : \mathbb{N} \to \mathbb{N}$ in $\mathcal{G}^!_{inn}$, we have

$$\top_\mathbb{N}; J(\sigma) = \top_\mathbb{N}; J(\tau)$$

if and only if $\sigma \approx \tau$.

This definition gives a rather minimal condition on the morphism $\top_\mathbb{N}$ that ensures that it behaves as it should when substituted into functions $\mathbb{N} \to \mathbb{N}$. Our category $\mathcal{G}_{ND}$, for example, is a sensible game semantics of PCF with countable nondeterminism, as is the Tsukada-Ong category $\mathcal{G}_{TO}$.

Using this condition and Lemma 5.1, it is easy to prove the following version of adequacy, which is sufficient for the purpose of proving full abstraction, for all such models.

Theorem 5.4 (Adequacy for nondeterministic PCF)**.** *Let C be a sensible model of nondeterministic PCF, and define an interpretation $[\![-]\!]$ of nondeterministic PCF in C to be given by the functor J and the morphism $\top_\mathbb{N}$ in the obvious way.*

Let $M, N :$ `nat` be two terms of nondeterministic PCF. Suppose that for all $n \in \omega$, $M \Downarrow n$ if and only if $N \Downarrow n$ and furthermore that $M \Downarrow^{must}$ if and only if $N \Downarrow^{must}$. Then $[\![M]\!] = [\![N]\!]$ in C.

6 Conclusion

Our last result is fairly general, and can be applied to a number of different game semantics models, including the models of concurrent games of [Castellan et al. 2016]. The natural next step is to extend it to give a proof of full abstraction for nondeterministic PCF in the Tsukada-Ong model. The missing ingredient is a universality result for that model. Tsukada and Ong prove compact definability from first principles, rather than using a factorization result, and it is not immediately clear how one defines a recursive element of their model. Nevertheless, we can have some confidence that such a result is possible, allowing our results to be carried over to stateless languages.

It would be interesting to see whether our adequacy proof can be applied to effects other than nondeterminism: although most effects do not engender a failure of continuity in the same way, a similar approach could help us use an adequacy result from one language to prove adequacy for another language, rather than having to prove it from first principles each time.

Lastly, our proof has thrown more light on the important relation between nondeterminism and state. As we have discussed, stateful languages (and visible strategies) pose fewer problems for modelling nondeterminism than do stateless languages (and innocent strategies). However, the fact that we needed to use the stateful structure of the language in our proof of Lemma 4.2 suggests that there is a deeper relationship between the two, which we should explore in future work.

Acknowledgments

This material is based upon work supported by the EPSRC under Grant No. EP/K018868/1. I am grateful to Martin Hyland for an interesting conversation that helped me develop some of this material.

References

Samson Abramsky and Radha Jagadeesan. 1994. Games and Full Completeness for Multiplicative Linear Logic. *The Journal of Symbolic Logic* 59, 2 (1994), 543–574. http://arxiv.org/abs/1311.6057

Samson Abramsky, Radha Jagadeesan, and Pasquale Malacaria. 2000. Full Abstraction for PCF. *Information and Computation* 163, 2 (2000), 409 – 470. https://doi.org/10.1006/inco.2000.2930

Samson Abramsky and Guy McCusker. 1999. Full Abstraction for Idealized Algol with Passive Expressions. *Theor. Comput. Sci.* 227, 1-2 (Sept. 1999), 3–42. https://doi.org/10.1016/S0304-3975(99)00047-X

K. R. Apt and G. D. Plotkin. 1981. A cook's tour of countable nondeterminism. In *Automata, Languages and Programming*, Shimon Even and Oded Kariv (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 479–494.

Simon Castellan, Pierre Clairambault, and Glynn Winskel. 2016. Concurrent Hyland-Ong games. (Sept. 2016). https://hal.archives-ouvertes.fr/hal-01068769 working paper or preprint.

Pierre-Louis Curien. 2007. Definability and Full Abstraction. *Electronic Notes in Theoretical Computer Science* 172 (2007), 301 – 310. https://doi.org/10.1016/j.entcs.2007.02.011 Computation, Meaning, and Logic: Articles dedicated to Gordon Plotkin.

Edsger Wybe Dijkstra. 1997. *A Discipline of Programming* (1st ed.). Prentice Hall PTR, Upper Saddle River, NJ, USA.

R. Harmer and G. McCusker. 1999. A fully abstract game semantics for finite nondeterminism. In *Proceedings. 14th Symposium on Logic in Computer Science (Cat. No. PR00158)*. 422–430. https://doi.org/10.1109/LICS.1999.782637

Russell S. Harmer. 1999. *Games and full abstraction for nondeterministic languages.* Technical Report.

J.M.E. Hyland and C.-H.L. Ong. 2000. On Full Abstraction for PCF: I, II, and III. *Information and Computation* 163, 2 (2000), 285 – 408. https://doi.org/10.1006/inco.2000.2917

J. Laird. 2015. Sequential Algorithms for Unbounded Nondeterminism. *Electronic Notes in Theoretical Computer Science* 319 (2015), 271 – 287. https://doi.org/10.1016/j.entcs.2015.12.017

James Laird. 2016. Higher-order Programs as Coroutines. (2016). to appear.

Paul Blain Levy. 2008. Infinite trace equivalence. *Annals of Pure and Applied Logic* 151, 2 (2008), 170 – 198. https://doi.org/10.1016/j.apal.2007.10.007

A. S. Murawski. 2008. Reachability Games and Game Semantics: Comparing Nondeterministic Programs. In *23rd Annual IEEE Symposium on Logic in Computer Science (LICS 2008)(LICS)*, Vol. 00. 353–363. https://doi.org/10.1109/LICS.2008.24

Andrzej S. Murawski and Nikos Tzevelekos. 2013. Deconstructing General References via Game Semantics. In *Foundations of Software Science and Computation Structures*, Frank Pfenning (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 241–256.

A. W. Roscoe. 1993. Unbounded Non-determinism in CSP. *Journal of Logic and Computation* 3, 2 (1993), 131. https://doi.org/10.1093/logcom/3.2.131

Takeshi Tsukada and C. H. Luke Ong. 2015. Nondeterminism in Game Semantics via Sheaves. In *Proceedings of the 2015 30th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS) (LICS '15)*. IEEE Computer Society, Washington, DC, USA, 220–231. https://doi.org/10.1109/LICS.2015.30

Glynn Winskel. 2013. Strategies as Profunctors. In *Foundations of Software Science and Computation Structures*, Frank Pfenning (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 418–433.