A Fully Abstract Game Semantics for Countable Nondeterminism

W. John Gowers[1]
Computer Science Department, University of Bath
Claverton Down Road, Bath. BA2 7QY, United Kingdom
W.J.Gowers@bath.ac.uk
https://orcid.org/0000-0002-4513-9618

James D. Laird
Department of Computer Science, University of Bath
Claverton Down Road, Bath. BA2 7QY, 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 yield proofs of adequacy for models of nondeterministic PCF, such as those given by Tsukada and Ong.

2012 ACM Subject Classification Dummy classification

Keywords and phrases semantics, nondeterminism, games and logic

Digital Object Identifier 10.4230/LIPIcs.CVIT.2016.23

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

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

[1] funded by EPSRC grant EP/K018868/1

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. [2] 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 [7, 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. This is a problem in itself for semanticists, for whom continuity is often a key ingredient in proofs of computational adequacy and full abstraction.

We shall explore some of these problems 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 $<n\ k$ 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, $<m$? may diverge – since ? may evaluate to $m + 1$, say. But $(\lambda k.k; \texttt{0})$? always converges to 0.

Lack of continuity is a problem in denotational semantics 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 an unboundedly 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 [8], 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.

[2] 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.

The idea of adding infinite traces into strategies in order to model unbounded nondeterminism goes back to Roscoe's work on CSP [17], and is very similar to work by Levy [13] on game semantics for a higher order language. In particular, we will need something similar to Levy's notion of a *lively* strategy – one that is a union of deterministic strategies – a property that does not automatically hold 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 [18] and in the more general work on concurrency by Winskel et al. (e.g., see [19] and [5]), 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 some of these models of nondeterministic stateless languages.

Related work by Laird [11, 12] 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 [4]. A crucial observation in both [4] and [11] 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 [13], is the first example of using the technique to establish a compositional full abstraction result for may and must testing.

Levy's work in [13] is part of a tradition of techniques used to handle unbounded nondeterminism operationally, normally using Labelled Transition Systems (see, for example, [16]). The contribution of this work is to apply the basic idea of including infinitary information to a compositional setting, where the semantics is built using the algebraic structure of higher-order programs.

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. Firstly, in the absence of continuity, it no longer suffices to show that we can define every *compact* strategy; instead, we need a *universality* result allowing us to define certain infinite strategies – specifically, the *recursive* ones.

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

$$\frac{}{\langle s, (\lambda x.M)\ N\rangle \longrightarrow \langle s, M[N/x]\rangle} \qquad \frac{}{\langle s, \mathbf{Y}_T M\rangle \longrightarrow \langle s, M(\mathbf{Y}_T M)\rangle}$$

$$\frac{}{\langle s, \mathtt{suc}\ \mathtt{n}\rangle \longrightarrow \langle s, \mathtt{n}+1\rangle} \qquad \frac{}{\langle s, \mathtt{pred}\ \mathtt{n}\rangle \longrightarrow \langle s, 0 \sqcup (\mathtt{n}-1)\rangle} \qquad \frac{}{\langle s, \mathtt{If0}\ 0 M N\rangle \longrightarrow \langle s, M\rangle}$$

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

$$\frac{}{\langle s, (\mathsf{mkvar}\ M N) \coloneqq L\rangle \longrightarrow \langle s, N\ L\rangle} \qquad \frac{}{\langle s, v \coloneqq n\rangle \longrightarrow \langle \langle s \mid v \mapsto n\rangle, \mathsf{skip}\rangle}$$

$$s(v) = n\ \frac{}{\langle s, \mathtt{@}v\rangle \longrightarrow \langle s, n\rangle} \qquad \frac{}{\langle s, \mathsf{skip}; M\rangle \longrightarrow \langle s, M\rangle}$$

$$\frac{}{\langle s, \mathtt{new}_T \lambda v.M\rangle \longrightarrow \langle \langle s \mid v \mapsto 0\rangle, M\rangle} \qquad \frac{\langle s, M\rangle \longrightarrow \langle s, M'\rangle}{\langle s, E[M]\rangle \longrightarrow \langle s, E[M']\rangle}$$

$$\frac{}{\langle s, ?\rangle \longrightarrow \langle s, \mathtt{n}\rangle}\ n \in \mathbb{N}$$

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

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. We shall 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 type theory and operational semantics for Idealized Algol with countable 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 [3], together with the nondeterministic choice:

$$\begin{aligned} M ::= {} & x \mid \lambda x.M \mid M\ M \mid \mathbf{Y}_T \mid \\ & \mathtt{n} \mid \mathsf{skip} \mid \mathtt{suc}\ \mid \mathtt{pred}\ \mid \\ & \mathtt{If0}\ \mid \_\,;\,\_ \mid \_ \coloneqq \_ \mid \\ & \mathtt{@}^{3} \mid \mathsf{new}_T \mid \mathsf{mkvar}\ \mid ?\,. \end{aligned}$$

The typing rule for $?$ is $\Gamma \vdash ?: \mathtt{nat}$. We shall use 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 [8], except with a different rule for the countable rather than finite nondeterminism.

[3] That is, variable @ccess.

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

$$E ::= - \mid EM \mid \texttt{suc}\ E \mid \texttt{pred}\ E \mid \texttt{If0}\ E \mid$$
$$E;\ \_ \mid E := \ \_ \mid \texttt{@}E \mid \mathsf{mkvar}\ E \mid \texttt{new}_T E$$

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 $x = \mathsf{skip}$ 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

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

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

3 Game Semantics

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 $b \not\vdash_A a$ 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 denotations of the types `nat` and `com` are the flat arenas $\mathbb{N}$ and $\mathbb{C}$ on, respectively, the set of natural numbers and the singleton $\{a\}$.

We assume that our arenas are *enumerated*; i.e., that the set M_A is equipped with a partial surjection $\mathbb{N} \to M_A$. The denotation of any IA type has a natural enumeration.

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*. In particular, every justified string begins with an initial move, and 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 use the approach taken by Abramsky and McCusker [3] – a middle road between the *arenas* of Hyland and Ong and the *games* of [2] 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 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 [3]. As an example, we give the definition of $A \multimap B$:

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

3.2.2 Modelling countable nondeterminism

As in [8], 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 so we can distinguish terms such as `If0` ? Ω `0`, which may diverge, and `0`, which must converge.

To fix this problem, we follow [8] 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 the set of those O-positions where there is a possibility of divergence.

Tracking divergences explicitly in this way requires some care when we compose strategies. 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

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

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

$$a)\quad \begin{array}{cc} \mathbb{N}_1 & \mathbb{N}_2 \\ & q \\ q & \\ n_1 & \\ q & \\ n_2 & \\ \vdots & \end{array} \qquad\qquad b)\quad \begin{array}{cc} \mathbb{N}_1 & \mathbb{N}_2 \\ & q \\ q & \\ n_1 & \\ \vdots & \\ q & \\ n_m & \\ & 0 \end{array}$$

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

for $\mathbb{N}$ on the left, we expect the resulting strategy to contain divergences, since the term $M\mathtt{n}$ diverges for any $\mathtt{n}$. 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.

The approach adopted in [8] 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.

Harmer and McCusker's approach works very satisfactorily for finite nondeterminism, but not at all for countable nondeterminism. To see why, consider the term

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

This term first chooses a natural number m, and then reads from its input n for a total of m times before eventually returning $\mathtt{0}$. 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 $\mathtt{If0}\ ?MN$ has the same denotation as N, even though for any n, $M\mathtt{n} \not\Downarrow^{\mathrm{must}}$, while $N\mathtt{n} \Downarrow^{\mathrm{must}}$. 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 N must converge.

Aside from showing that the naive extension of the Harmer-McCusker model cannot be sound, this example actually leads to composition not being associative (e.g., see [9, 4.4.1]).

What this illustrates is that we can no longer deduce the infinitary behaviour of a strategy by looking at the limits of its finite plays; instead, we need to keep track of infinite sequences of moves explicitly, in the style of [17] and [13]. 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.

3.2.3 Strategies

We define an *infinite justified string* in an arena A in the obvious way. We say such a string is *recursive* if it corresponds, via the enumeration on M_A, to a recursive function $\mathbb{N} \to \mathbb{N}$.

We define $\overline{P_A}$ to be P_A together with the set of all those recursive infinite justified sequences that have all finite prefixes in P_A. Note that we deliberately ignore any non-recursive infinitary behaviours, since these cannot be detected by computable contexts.

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 $sa \in T_\sigma$.
- $D_\sigma \subset \overline{P_A}$ 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 $s \sqsubseteq d$ 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 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 $a \to b \to c$, in which case we replace them with a pointer $a \to c$.

We call $\mathfrak{s}$ 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 [8]; we set

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

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 a 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 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 recursive infinite play s if it contains every finite prefix of s.

Any finite-nondeterminism strategy in the sense of [8] may be interpreted as a complete strategy by enlarging it with all its infinite recursive 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 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 $D_\sigma = \bigcup D_{\sigma_n}$.

Henceforth, we 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.** We say that a strategy σ for a game A is *deterministic* if

- it is complete;
- if sab, sac are P-plays in T_σ then $b = c$ 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 4.** *Let A, B, C be games and let $\sigma\colon A \multimap B$, $\tau\colon B \multimap C$ be deterministic strategies. Then $\sigma;\tau$ is complete.*

Proof. The proof relies on a lemma from [10] 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 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 the $\mathfrak{s}_i$ 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 5.** *The composition of strategies $\sigma\colon A \multimap B$ and $\tau\colon B \multimap C$ 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 [1] 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 $\mathfrak{s} \in \mathrm{int}_\infty(A,B,C)$ (with a tail 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 5 really makes use of the fact that a locally complete strategy is *lively* in the sense of [13]; 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

The proof of associativity of composition is the same in our model as it is in any other model of game semantics if we treat infinite plays the same as finite ones. However, it is worth saying a few words about associativity, 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, on the other hand, 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 [3].

3.4 A Cartesian closed category

We follow the construction given in [3], using the connectives ! and $\times$ 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 similar to the construction of a co-Kleisli category for a linear exponential comonad, but 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, 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 [3].

3.6 Recursive strategies

Most full abstraction results go via a definability result that says that all *compact* strategies are definable [6]. 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 [10] that every *recursively presentable* innocent strategy is PCF-definable.

If σ is a complete strategy for a game A, we say σ is *recursive* if $T_\sigma \cap P_A$ and $D_\sigma \cap P_A$ are recursively enumerable subsets of ω^ω (under the enumeration of M_A). Here, we throw away the infinite plays in T_σ and D_σ, but we do not lose any information because σ is complete.

If σ is locally complete, we say that σ is *recursive* if σ is the union of complete recursive strategies $\sigma_1, \sigma_2, \cdots$, where the map $i \mapsto \sigma_i$ is a recursive function $\mathbb{N} \to (\mathbb{N} \to \mathbb{N}) \to 2$.

In the case that σ is recursive and *deterministic*, we can prove the following result.

▶ **Proposition 6** (Recursive Universality for Idealized Algol)**.** *Let S be an Idealized Algol type and let $\sigma\colon [\![S]\!]$ be a recursive deterministic strategy. Then there exists a term $M\colon S$ of Idealized Algol such that $\sigma = [\![M]\!]$.*

Proof. We use the 'innocent factorization' result of [3] to reduce to the innocent case and the proceed in a manner similar to the argument used in [15]. ◀

Note that Proposition 6 is sharper than the result in [10], which only proves that every recursive strategy may be defined *up to observational equivalence*. Idealized Algol allows us to store variables and then use them multiple times without having to read them again, which allows us to to define all recursive visible strategies exactly. Compare with [15], which proves a similar result for *call-by-value* PCF.

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 in [8] because of the unbounded nondeterminism.

▶ **Proposition 7.** *Let $\sigma\colon I \to A$ be a strategy for a game A in $\mathcal{G}_{ND}$. Then we may write σ as $\top_\mathbb{N}; \mathsf{Det}(\sigma)$, where $\mathsf{Det}(\sigma)\colon !\mathbb{N} \to A$ is a deterministic strategy and $\top_\mathbb{N}\colon \mathbb{N}$ is the strategy that contains every play in $!\mathbb{N}$ and has no finite divergences.*

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 $\mathsf{Det}(\sigma)$ 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 $\mathsf{code}_A(b_1) < \mathsf{code}_A(b_2) < \cdots$. 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 $0 < j \leq \mathsf{code}_A(b_1)$, then play b_1.
- If $\mathsf{code}_A(b_n) < j \leq \mathsf{code}_A(b_{n+1})$ then play b_{n+1}.
- If $j = 0$ and $sa \in D_\sigma$, then play nothing, and put the resulting play inside $D_{\mathsf{Det}(\sigma)}$. Otherwise, play b_1.

We close under limits to make the strategy $\mathsf{Det}(\sigma)$ complete. $\mathsf{Det}(\sigma)$ is clearly deterministic. Checking that $\top_\mathbb{N}; \mathsf{Det}(\sigma) = \sigma$ 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 $\mathsf{Det}(\sigma)$ is recursive if σ is and is visible if σ is.

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 $M : T$ 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 [3]. 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 8** (Computational Adequacy). *Let M:* `com` *be a closed term of nondeterministic Idealized Algol. $M \Downarrow$* 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, [8] and [9] 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 encoding the nondeterministic choices that we have made.

▶ **Lemma 9.** *Let $M = C[?]$ be a term of type* `com`*, where $C[-]$ is a multi-holed context of (deterministic) Idealized Algol. Write σ_M for the denotation of the term $\lambda n.C[n]$.*

- *If $M \Downarrow$* skip *then there exists some total deterministic strategy $\sigma : !\mathbb{N}$ such that $qa \in T_{\sigma;\sigma_M}$.*
- *If $M \not\Downarrow^{must}$ then there exists some total deterministic strategy $\sigma : !\mathbb{N}$ 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 $N_{n_1,\ldots,n_k,d}$: (`nat` $\to$ `com`) $\to$ `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 calls the function f, passing in n_1 the first time, n_2 the second and so on, passing in d at every call beyond $k+1$.

Now let π be a finite evaluation of $\langle s, C[?]\rangle$ 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 non-deterministic 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 [3], 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

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

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 $N_{n_1,\ldots,n_k,d}(\lambda n.C[n]) \Downarrow \mathsf{skip}$, which is the case if and only if $M \Downarrow \mathsf{skip}$ 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 [8]). 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 the proof of this result, we can establish the converse, which we will also need.

▶ **Lemma 10.** *Let $M = C[?]$ be as before. Let $\sigma\colon !\mathbb{N}$ be a total deterministic strategy.*

- *If $qa \in T_{\sigma;\sigma_M}$ then $M \Downarrow \mathsf{skip}$.*
- *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 $M \Downarrow \mathsf{skip}$ along π. The corresponding statement for must convergence follows in the same way. ◀

Note that these last two lemmas may be cast entirely in the model of *deterministic* Idealized Algol given in [3], since they only refer to the denotations of deterministic terms. We can therefore prove a more general version of Proposition 8.

▶ **Definition 11.** Let $\sigma \colon A \to B$ be a (deterministic) strategy. We say that σ is *winning* if every play in σ may be extended to a play that ends with a P-move in B; i.e., σ is total and contains no sequences having an infinite tail in A.

This definition is motivated by Lemmas 9 and 10 in the following sense: if $\sigma_M : \mathbb{N} \to \mathbb{C}$ is a strategy, then there exists some σ_M such that $D_{\sigma;\sigma_M} \neq \varnothing$ if and only if σ is not winning.
The following is now an easy corollary of Lemmas 9 and 10.

▶ **Corollary 12.** *Let $\mathcal{C}$ be a Cartesian closed category that admits a faithful Cartesian functor $J \colon \mathcal{G}^!_{vis} \hookrightarrow \mathcal{C}$. Let $\top_\mathbb{N} \colon 1 \to J\mathbb{N}$ be a morphism in $\mathcal{C}$ and use it to extend the semantics of Idealized Algol of $\mathcal{G}^!_{vis}$ to a semantics of nondeterministic Idealized Algol, as in Section 4.1.*
Suppose we have two predicates $\Downarrow$ skip and $\Downarrow^{must}$ defined on strategies $1 \to J\mathbb{C}$ in $\mathcal{C}$ satisfying the following rules for all strategies $\sigma \colon \mathbb{N} \to \mathbb{C}$ in $\mathcal{G}^!_{vis}$.
- $(\top_\mathbb{N}; J\sigma) \Downarrow$ skip *if and only if there is some $s \in \sigma$ such that $s|_\mathbb{C} = qa$.*
- $(\top_\mathbb{N}; J\sigma) \Downarrow^{must}$ *if and only if σ is winning.*

Then the semantics of nondeterministic Idealized Algol inside $\mathcal{C}$ is adequate in the following sense. For all terms M of nondeterministic Idealized Algol of type `com`*:*
- $M \Downarrow$ skip *if and only if* $[\![M]\!] \Downarrow$ skip.
- $M \Downarrow^{must}$ *if and only if* $[\![M]\!] \Downarrow^{must}$.

We can then deduce Proposition 8 by verifying that the following predicates on strategies $\sigma \colon 1 \to \mathbb{C}$ in the category $\mathcal{G}^!_{ND,vis}$ satisfy the conditions of Corollary 12.
- $\sigma \Downarrow$ skip $\Leftrightarrow qa \in T_\sigma$.
- $\sigma \Downarrow^{\text{must}} \Leftrightarrow D_\sigma = \varnothing$.

4.3 Intrinsic Equivalence and Soundness

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 13** (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 $\lambda n.\texttt{If0}\ n\ \Omega\ \Omega$ of type `nat` $\to$ `nat`. These terms are clearly observationally equivalent, but their denotations are not equal; for example, $q \in D_{[\![\lambda n.\Omega]\!]}$, but $q \notin D_{[\![\lambda n.\texttt{If0}\ n\ \Omega\ \Omega]\!]}$.
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 $\text{dc}(\sigma)$; Murawski's explicit characterization of the intrinsic collapse [14], which may

be applied to our model, essentially says that $\sigma \sim \tau$ if and only if σ and τ have the same *complete* plays and $\mathrm{dc}(\sigma) = \mathrm{dc}(\tau)$.

An important fact about intrinsic equivalence is the following Lemma, whose proof makes use of the fact that the infinite plays in our strategies are given by recursive functions.

▶ **Lemma 14.** *Let σ, τ be strategies for a game A. Suppose that $\sigma ; \alpha = \tau ; \alpha$ for all* recursive *strategies $\alpha \colon A \to \mathbb{C}$. Then $\sigma \sim \tau$.*

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

4.5 Full abstraction

▶ **Theorem 15** (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 $[\![M]\!] ; \alpha \neq [\![N]\!] ; \alpha$. By Lemma 14, we can 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 Conclusion

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

The principal difficulties in modelling nondeterministic stateless languages were overcome by Tsukada and Ong in [18], where they outlined how to define an innocent nondeterministic strategy by retaining 'branching time information' in strategies. An additional benefit of the retention of branching time information is that we no longer need to keep track of infinite plays in order to model unbounded nondeterminism. The model given in [18] is not sound for must-equivalence, but the authors make the claim that it their model may be easily modified to yield a model that *is* sound for this type of equivalence, using the same techniques from [8] that we have used.

We could use our methods to help establish this claim in the case of unbounded nondeterminism; specifically, our proof of adequacy will extend to such a model. Indeed, Corollary 12 can easily be modified to apply to PCF, even though we have used Idealized Algol terms in the proof. Corollary 12 then reduces the proof of adequacy to a combinatorial check on morphisms from $\mathbb{N} \to \mathbb{C}$ on strategies in the well-known category $\mathcal{G}^!_{vis}$, together with an examination of what happens to those strategies when we compose them with $\top_{\mathbb{N}}$.

References

1 Samson Abramsky and Radha Jagadeesan. Games and full completeness for multiplicative linear logic. *The Journal of Symbolic Logic*, 59(2):543–574, 1994. URL: http://arxiv.org/abs/1311.6057.

2 Samson Abramsky, Radha Jagadeesan, and Pasquale Malacaria. Full abstraction for PCF. *Information and Computation*, 163(2):409 – 470, 2000. URL: `http://www.sciencedirect.com/science/article/pii/S0890540100929304`, `doi:http://dx.doi.org/10.1006/inco.2000.2930`.

3 Samson Abramsky and Guy McCusker. Full abstraction for Idealized Algol with passive expressions. *Theor. Comput. Sci.*, 227(1-2):3–42, September 1999. URL: `http://dx.doi.org/10.1016/S0304-3975(99)00047-X`, `doi:10.1016/S0304-3975(99)00047-X`.

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

5 Simon Castellan, Pierre Clairambault, and Glynn Winskel. Concurrent Hyland-Ong games. working paper or preprint, September 2016. URL: `https://hal.archives-ouvertes.fr/hal-01068769`.

6 Pierre-Louis Curien. Definability and full abstraction. *Electronic Notes in Theoretical Computer Science*, 172:301 – 310, 2007. Computation, Meaning, and Logic: Articles dedicated to Gordon Plotkin. URL: `http://www.sciencedirect.com/science/article/pii/S1571066107000837`, `doi:https://doi.org/10.1016/j.entcs.2007.02.011`.

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

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

9 Russell S. Harmer. Games and full abstraction for nondeterministic languages. Technical report, 1999.

10 J.M.E. Hyland and C.-H.L. Ong. On full abstraction for PCF: I, II, and III. *Information and Computation*, 163(2):285 – 408, 2000. URL: `http://www.sciencedirect.com/science/article/pii/S0890540100929171`, `doi:http://dx.doi.org/10.1006/inco.2000.2917`.

11 J. Laird. Sequential algorithms for unbounded nondeterminism. *Electronic Notes in Theoretical Computer Science*, 319:271 – 287, 2015. URL: `http://www.sciencedirect.com/science/article/pii/S1571066115000845`, `doi:http://dx.doi.org/10.1016/j.entcs.2015.12.017`.

12 James Laird. Higher-order programs as coroutines. to appear, 2016.

13 Paul Blain Levy. Infinite trace equivalence. *Annals of Pure and Applied Logic*, 151(2):170 – 198, 2008. URL: `http://www.sciencedirect.com/science/article/pii/S0168007207000887`, `doi:http://dx.doi.org/10.1016/j.apal.2007.10.007`.

14 A. S. Murawski. Reachability games and game semantics: Comparing nondeterministic programs. In *23rd Annual IEEE Symposium on Logic in Computer Science (LICS 2008)(LICS)*, volume 00, pages 353–363, 06 2008. URL: `doi.ieeecomputersociety.org/10.1109/LICS.2008.24`, `doi:10.1109/LICS.2008.24`.

15 Andrzej S. Murawski and Nikos Tzevelekos. Block structure vs scope extrusion: between innocence and omniscience. *Logical Methods in Computer Science*, 12(3), 2016. URL: `https://doi.org/10.2168/LMCS-12(3:3)2016`, `doi:10.2168/LMCS-12(3:3)2016`.

16 Corin Pitcher. Functional programming and erratic non-determinism. 2001.

17 A. W. Roscoe. Unbounded non-determinism in CSP. *Journal of Logic and Computation*, 3(2):131, 1993. URL: `http://dx.doi.org/10.1093/logcom/3.2.131`, `doi:10.1093/logcom/3.2.131`.

18 Takeshi Tsukada and C. H. Luke Ong. Nondeterminism in game semantics via sheaves. In *Proceedings of the 2015 30th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, LICS '15, pages 220–231, Washington, DC, USA, 2015. IEEE Computer Society. URL: `http://dx.doi.org/10.1109/LICS.2015.30`, `doi:10.1109/LICS.2015.30`.

19 Glynn Winskel. Strategies as profunctors. In Frank Pfenning, editor, *Foundations of Software Science and Computation Structures*, pages 418–433, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.