Full Title[*]

Subtitle[†]

W. J. Gowers[‡]

PhD Student

Department of Computer Science

University of Bath

Bath, United Kingdom

W.J.Gowers@bath.ac.uk

Abstract

Text of abstract ....

Keywords keyword1, keyword2, keyword3

1 Introduction

Text of paper ...

2 PCF with Countable Nondeterminism

The language we shall study is similar to that given in [Laird 2015]. It is the language formed by adding an unbounded natural-number choice primitive on to the base language PCF. That is, it is the language formed by taking the simply-typed λ-calculus over the ground type $\mathtt{nat}$, and adding the constants:

- Structural constants $\mathtt{0}\colon \mathtt{nat}, \mathtt{suc}\ , \mathtt{pred}\ : \mathtt{nat} \to \mathtt{nat}$ and $\mathtt{If0}\ : \mathtt{nat} \to \mathtt{nat} \to \mathtt{nat} \to \mathtt{nat}$ for $\mathtt{nat}$
- Recursion combinators $\mathrm{Y}_T\colon (T \to T) \to T$ for each T.
- Countable choice $?\colon \mathtt{nat}$.

A small-step operational semantics for this language, taken from [Laird 2015], is given in Figure 1. Our main task will be to give a Denotational Semantics for this language.

Given a term M of ground type $\mathtt{nat}$, we say that M *must converge*, and write $M \Downarrow^{\mathrm{must}}$, if every evaluation π of M (according to the operational semantics given in Figure 1) terminates at some numeral value. We write $M \Downarrow \mathsf{n}$ if there exists some evaluation π of M that terminates at n.

$$\frac{}{(\lambda x.M)N \longrightarrow M[N/x]} \qquad \frac{}{? \longrightarrow \mathsf{n}}\ n \in \mathbb{N}$$

$$\frac{}{\mathrm{Y}M \longrightarrow M(\mathrm{Y}M)} \qquad \frac{}{\mathtt{If0}\ \mathtt{0}\ x\ y \longrightarrow x}$$

$$\frac{}{\mathtt{If0}\ (\mathsf{suc}\ \mathsf{n})\ x\ y \longrightarrow y} \qquad \frac{M \longrightarrow M'}{MN \longrightarrow M'N}$$

$$\frac{M \longrightarrow M'}{\mathsf{suc}\ M \longrightarrow \mathsf{suc}\ M'} \qquad \frac{M \longrightarrow M'}{\mathsf{pred}\ M \longrightarrow \mathsf{pred}\ M'}$$

$$\frac{M \longrightarrow M'}{\mathtt{If0}\ M \longrightarrow \mathtt{If0}\ M'}$$

Figure 1. Operational Semantics for a PCF-like language with countable nondeterminism.

Given a type T, we define an order on the terms of type T, by saying that $M \sqsubseteq_{m\&m} N$ if for all compatible contexts $C[-]$ of type $\mathtt{nat}$ we have

$$C[M] \Downarrow \mathsf{n} \Rightarrow C[N] \Downarrow \mathsf{n}$$
$$C[M] \Downarrow^{\mathrm{must}} \Rightarrow C[N] \Downarrow^{\mathrm{must}} \ .$$

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

One aspect of the language that makes it particularly difficult to model is that function application is not continuous, either with respect to functions or with respect to arguments.

For example, it is easy to write a sequence of functions $<\!n\colon \mathtt{nat} \to \mathtt{nat}$ that return 0 if their argument is less than n and diverge otherwise:

[*] with title note

[†] with subtitle note

[‡] with author1 note

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

$$
\begin{aligned}
&<0 = \lambda m.\Omega \\
&<1 = \lambda m.\texttt{If0}\ m\ \texttt{0}\ \Omega \\
&<2 = \lambda m.\texttt{If0}\ m\ \texttt{0}\ (\texttt{If0}\ (\texttt{pred}\ m)\ \texttt{0}\ \Omega) \\
&\ldots
\end{aligned}
$$

It is easy enough to see (for example, by the denotational semantics) that the least upper bound of these functions is the function $\lambda x.\texttt{0}$. However, it is clear that $<n\ ?\ \not\Downarrow^{\text{must}}$ for each n, while $\lambda x.\texttt{0}\ ?$ must converge to $\texttt{0}$. Thus, $\lambda x.\texttt{0}\ ?$ is not the least upper bound of the $<n\ ?$.

This is a bit of a problem, since continuity often plays an important role in full abstraction proofs – firstly, for proving Computational Adequacy of the recursion combinator, and, secondly, for deducing full abstraction from compact definability. Semantically, we shall deal with the first problem by adopting a novel approach to Computational Adequacy, and with the second problem by appealing to the stronger *universality* result: namely, that every *recursive* strategy is definable.

3 The base model

We will give a model based on Abramsky, Jagadeesan and Malacaria's Game Semantics for PCF [Abramsky et al. 2000]. We do this partly for the sake of simplicity, and partly to demonstrate the power of our methods – since AJM games are typically seen as less flexible than the Hyland-Ong games of [Hyland and Ong 2000]. However, we stress that our construction can be carried out just as well in Hyland and Ong's model, and we will keep this model in mind throughout, and mention whenever a slightly modified argument is needed for that model.

3.1 Games

A *game* is a tuple $A = (M_A, \lambda_A, P_A)$, where

- M_A is a set of *moves*,
- $\lambda_A \colon M_A \longrightarrow \{O,P\} \times \{Q,A\}$ is a function that designates each move as either a P-*move* or an O-*move*, and as either a *question* or an *answer*.
- P_A is a non-empty prefix-closed set of finite sequences of elements of M_A called *positions*, satisfying the following two conditions:

  Alternation any non-empty position in P_A starts with an O-move, and if $sab \in P_A$ then a and b have opposite signs - one is a P-move and the other an O-move.

  Well-bracketing If $s \in P_A$, then the corresponding QA-sequence of questions and answers is well-bracketed.

3.2 Strategies

A (deterministic) *strategy* for a game A is a non-empty, prefix-closed subset *sigma* $\subseteq P_A^+$ satisfying *determinism*: if $sa, sb \in \sigma$ are both P-positions, then $a = b$. Here, P_A^+ is the set of P-positions in A.

We say that a strategy σ is *history-free* if whenever $sab, tac \in \sigma$ are both P-positions, then $b = c$. Such a strategy can be though of as a partial function $M_A^- \rightharpoonup M_A^+$, where M_A^- is the set of O-moves and M_A^+ the set of P-moves in A. If this function can be given as a partial recursive function, we say that the strategy σ is *recursive*.

3.3 Connectives

Given games A and B, we can define games $A \times B$, $A \otimes B$ and $A \multimap B$ as follows:

- $M_{A\times B} = M_{A\otimes B} = M_{A\multimap B} = M_A + M_B$.
- $\lambda_{A\times B} = \lambda_{A\otimes B} = [\lambda_A, \lambda_B]$. $\lambda_{A\multimap B} = [\neg \circ \lambda_A, \lambda_B]$, where $\neg \colon \{O,P\} \times \{Q,A\} \longrightarrow \{O,P\} \times \{Q,A\}$ is the function that exchanges O and P and leaves Q and A alone.
- $P_{A\times B}$ is the union of P_A and P_B. $P_{A\otimes B}$ and $P_{A\multimap B}$ are defined by the following formula:

$$\{s \in (M_A + M_B)^* \ :\ s|_A \in P_A,\ s|_B \in P_B,\ s \text{ is alternating}\}\,.$$

  Note that this is a different strategy for $P_{A\otimes B}$ and $P_{A\multimap B}$, because the definition of an 'alternating sequence' depends on the sign function λ.

3.4 Composition of strategies

Given strategies $\sigma \colon A \multimap B$ and $\tau \colon B \multimap C$, we define a set

$$\sigma\|\tau = \{\mathfrak{s} \in (M_A + M_B + M_C)^* \ :\ \mathfrak{s}|_{A,B} \in \sigma,\ \mathfrak{s}|_{B,C} \in \tau\}\,.$$

We then define

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

It may be proved [Abramsky et al. 2000] that this notion of composition is associative and has identities given by the obvious copycat strategy for the game $A \multimap A$. We therefore end up with a category whose objects are games and where the morphisms from a game A to a game B are the strategies for the game $A \multimap B$.

It can also be shown that the composition of two history-free strategies is history free, and that the composition of two recursive strategies is recursive. Furthermore, the identity strategy is always history-free and recursive.

We write $\mathcal{G}$ for the category of games and strategies, $\mathcal{G}_{hf}$ for the category of games and history-free strategies, and $\mathcal{G}_{rec}$ for the category of games and recursive strategies.

Lastly, it is well-known that the connectives $\otimes$ and $\multimap$ exhibit $\mathcal{G}$ as a monoidal closed category (with tensor unit the empty game I with no moves) and that $\times$ gives products in $\mathcal{G}$. All these properties pass on to $\mathcal{G}_{hf}$ and $\mathcal{G}_{rec}$ in the expected way.

3.5 Exponentials

Given a game A, we define a game $!A$ as follows:

- $M_{!A} = M_A \times \omega$.
- $\lambda_{!A} = \lambda_A \circ \mathrm{pr}_1$.
- $P_{!A} = \{s \in M_{!A}^* \ : \ s \text{ is alternating, } s|_n \in P_A \forall n\}$.

It can be shown [Abramsky et al. 2000] that this construction gives rise to a *linear exponential comonad* on $\mathcal{G}$, and therefore that the co-Kleisli category $\mathcal{G}^!$ is a Cartesian closed category.

3.6 Modelling programming languages

The results from [Abramsky et al. 2000] that we will be interested are the following:

- $\mathcal{G}_{hf}$ is a fully abstract model of PCF.
- $\mathcal{G}_{rec}$ is also a fully abstract model of PCF. Moreover, it is a *universal* model: every morphism between the denotation of types is the denotation of a PCF term.

We will also be interested in a result of Laird [Laird [n. d.]], which tells us about the natural language for $\mathcal{G}$ itself:

- $\mathcal{G}$ is a fully abstract model of λ_{co}, an extension of PCF with coroutines. In particular, $\mathcal{G}$ is a sound and adequate model of a stateful language such as Idealized Algol.

This result – that our model of PCF may be embedded inside a game semantics model of a stateful language – will be useful when we come to prove Computational Adequacy for our model. It is a result specific to AJM games. For Hyland-Ong games, the category of games and *innocent* strategies, which is our model of PCF, may be embedded into the category of games and *visible* strategies, which is actually a fully abstract model of Idealized Algol. However, we will only need soundness and adequacy, so we can carry out our proof with AJM games as well.

4 Strategies for Nondeterminism

The results that we have quoted in the previous section all apply to *deterministic* PCF. For nondeterministic PCF, we follow the approach of Harmer and McCusker [Harmer and McCusker 1999], who modelled nondeterministic programs using *nondeterministic strategies*; i.e., strategies where we have relaxed the nondeterminism condition. So, for example, if we model the type `nat` by the game $\mathbb{N}$ with maximal positions qn for $n \in \mathbb{N}$, where q is a fixed opening move, then we can model the term ? by the strategy $\top_{\mathtt{nat}}$ that is the whole of $P_{\mathbb{N}}$.

However, this model is not yet rich enough to model may-and-must equivalence. Consider, for example, the following two of terms of type `nat`:

$$\begin{gathered} \mathtt{If0}\ ?\ \Omega\ \mathtt{0} \\ \mathtt{0}\,. \end{gathered}$$

In the first case, there are two possibilities: either the program returns `0` or it diverges. In the second case, the program always returns `0`.

Since divergence is represented by *partiality* in the strategy, these terms both have the same denotation: the strategy with unique maximal play $q0$. However, they are not must-equivalent, since $\mathtt{0} \Downarrow^{\text{must}}$, while the other term may diverge. The point is that in the first case the addition of the play $q0$ into the strategy obscures the partiality present in the denotation of Ω.

Harmer and McCusker's solution to this problem is to augment each strategy by the addition of a set of *divergences*: O-positions at which we signify that play may diverge.

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

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

where $n;P$ is a shorthand for $\mathtt{If0}\ n\ P\ P$, is given by a total strategy, without divergences: namely the strategy with plays of the form

$$\begin{array}{cc} \mathbb{N} & \mathbb{N} \\ & q \\ q & \\ n_1 & \\ q & \\ n_2 & \\ \vdots & \end{array}\quad.$$

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 $M\mathtt{n}$ diverges for any $\mathtt{n}$.

The approach adopted in [Harmer and McCusker 1999] is to check specifically for infinite chattering between strategies $\sigma : A \multimap B$ and $\tau : B \multimap C$ by checking whether the set $\sigma \| \tau$ contains any infinite increasing sequence of plays ending with moves 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

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

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 with maximal plays of the form:

$$\begin{array}{cc} \mathbb{N} & \mathbb{N} \\ & q \\ q & \\ n_1 & \\ \vdots & \\ q & \\ n_m & \\ & 0 \end{array} .$$

Note that this strategy strictly contains the one we considered before, and therefore that the denotation of

$$\mathtt{If0}\ ?MN$$

has the same denotation as N, even though it has all the divergent evaluations of M, while $N\mathsf{n} \Downarrow^{\mathrm{must}}$ for all n. 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 divergence into the composite – even though no divergence occurs in the evaluation of N.

Aside from making violating soundness for the model, 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 oves. 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]. Then the denotation of M will contain an infinite sequence, while the denotation of N will contain arbitarily 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.

4.1 Strategies

Here, we follow [Harmer and McCusker 1999], but with the addition of infinite positions. Given a set $X \subseteq M^*$, we write $\bar{X}$ for the ideal closure of X; i.e., X together with the set of all infinite sequences $s \in M^\omega$ all of whose finite prefixes are contained in X.

A *strategy* for a game $A = (A, \zeta_A)$ is given by a pair (T_σ, D_σ), where:

- T_σ is a prefix-closed subset of $\bar{P_A}$ such that if $s \in T_\sigma$ is a P-position and $sa \in P_A$, then $sa \in T_\sigma$. In other words, T_σ a nondeterministic strategy in the usual sense (possibly incorporating infinite plays). We consider all infinite positions in A to be O-positions.
- $D_\sigma \subseteq T_\sigma$ is a *postfix*-closed set of O-positions satisfying the following conditions:

  All divergences arise from positions If d is minimal in D_σ then either $d \in T_\sigma$ or d is an infinite position that is the limit of positions in T_σ.

  Diverge-or-reply If $s \in T_\sigma$ is an O-position, then either $s \in D_\sigma$ or there exists $sa \in T_\sigma$.

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

We say that a strategy σ is *complete* if $T_\sigma = \bar{T_\sigma}$; i.e., if $s \in T_\sigma$ whenever s is an infinite position, all of whose finite prefixes live in T_σ. For example, the denotation of M above is complete, but the denotation of N is not complete.

We say that σ is *locally complete* if T_σ can be written as the union of complete strategies.

It is easy to see that the denotation of any term from nondeterministic PCF must be locally complete. An example of a strategy that is not locally complete is given by taking the denotation of M above and considering it as a strategy with no infinite positions. The difference between this strategy and the denotation of N, is that every finite position in N is eventually completed in the copy of $\mathbb{N}$ on the right, so it is still locally complete.

We will require our strategies to be locally complete from here on. The problem with allowing non-locally complete strategies is that their composition does not necessarily satisfy the diverge-or-reply rule.

4.2 Composition of strategies

Let $\sigma\colon A \multimap B$ and $\tau\colon B \multimap C$ be strategies. We define $T_{\sigma;\tau}$ in the usual way. In order to define $D_{\sigma;\tau}$, we first define

$$\sigma \lightning \tau = \operatorname{pocl}\left\{\mathfrak{s} \in \overline{(M_A \sqcup M_B \sqcup M_C)^*} \;\middle|\; \begin{array}{l} \textbf{either } \mathfrak{s}|_{A,B} \in D_\sigma \\ \quad \text{and } s|_{B,C} \in T_\tau \\ \textbf{or } \mathfrak{s}|_{A,B} \in T_\sigma \\ \quad \text{and } s|_{B,C} \in D_\tau \end{array}\right\},$$

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

Lastly, we define

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

We need to show firstly that composition preserves local completeness. This is not completely trivial, since the composition of *complete* strategies need not be complete: for example, we may write $N = N'\ ?$, where N' is a deterministic strategy. Both N' and ? have complete denotations, but their composite, N, has a denotation that is only locally complete.

However, we can show that the composition of *deterministic* complete strategies σ and τ is complete. The proof makes use of a lemma from [Abramsky and Jagadeesan 1994] that says that if $s \in \sigma;\tau$ is some position, where σ and τ are both deterministic, then there is a unique minimal position $\mathfrak{s} \in \sigma\|\tau$ such that $\mathfrak{s}|_{A,C} = s$. This means that if $s_1 \sqsubseteq s_2 \sqsubseteq \cdots$ is an infinite increasing sequence in $\sigma;\tau$, then there is a corresponding infinite increasing sequence $\mathfrak{s}_1 \sqsubseteq \mathfrak{s}_2 \sqsubseteq \cdots$ in $\sigma\|\tau$, whose limit must be contained in $\sigma\|\tau$ by completeness of σ and τ. It follows that the limit of $s_1, s_2, \ldots$ is contained in $\sigma;\tau$.

Now the fact that composition preserves local completeness is an easy consequence of monotonicity of composition with respect to the prefix ordering, together with the fact that if a strategy σ is locally complete – i.e., if it is the union of some collection (σ_α) of complete strategies, then the σ_α may always be chosen to be deterministic.

We also need to prove that the composition of two strategies satisfies the diverge-or-reply rule. Once again, it is sufficient to prove this in the case that the strategies are both deterministic and complete. In that case, we can, for example, use the argument from [Abramsky and Jagadeesan 1994] that shows that a partiality in the composite $\sigma;\tau$ of two strategies $\sigma\colon A \multimap B$ and $\tau\colon B \multimap C$ (i.e., some O-position $s \in \sigma;\tau$ such that $sa \notin \sigma;\tau$ for any a) must arise either from a partiality in one of σ or τ or from an ‘infinite chattering’ between the two strategies. A partiality in either of σ or τ gives rise to a divergence in that strategy, by diverge-or-reply, and hence to a divergence at the appropriate place in $\sigma;\tau$. In the case that there is some infinite chattering between σ and τ; i.e., an infinite sequence $\mathfrak{s} \in \sigma\bar{\|}\tau$ such that $\mathfrak{s}|_{A,C}$ is finite, we must have $\mathfrak{s} \in \sigma\|\tau$ by completeness for σ and τ. Therefore, $\mathfrak{s}|_{A,B}$ and $\mathfrak{s}|_{B,C}$ are both divergences in their respective strategies, and therefore $\mathfrak{s}|_{A,C} \in D_{\sigma;\tau}$.

The last thing we need to show – namely, that composition is associative – follows in the usual way. The important difference between our model and that in [Harmer and McCusker 1999] is that in our model ‘divergence through infinite chattering’ is an organic consequence of the usual definition of composition, not an extra condition that we have imposed upon the model.

To illustrate this, we return to our examples M and N above, and show how the composite $M\ 1$ has a non-empty set of divergences, while $N\ 1$ is divergence-free.

In order for the strategy denotation of M to satisfy local completeness, it must contain all its infinite positions. Infinite positions in $!\mathbb{N}$ are P-winning, so the corresponding positions in $[\![M]\!] : !\mathbb{N} \multimap \mathbb{N}$ are infinite O-positions and are therefore contained in $D_{[\![M]\!]}$. Since the play $q \in\ !\,[\![1]\!]\,;[\![M]\!]$ is the restriction to $\mathbb{N}$ of the infinite position

$$\begin{array}{cc}
\mathbb{N} & \mathbb{N} \\
 & q \\
q & \\
1 & \\
q & \\
1 & \\
\vdots &
\end{array}$$

whose right-hand component is contained in $D_{[\![M]\!]}$, it must therefore be contained in $D_{![\![1]\!];[\![M]\!]}$.

We contrast this with the situation with N. In this case, the strategy-denotation of N contains *no* infinite position, and we have $D_{[\![N]\!]} = \varnothing$. Since a divergence in the composition must arise from a divergence in one of the two component strategies, this means that $D_{![\![1]\!];[\![N]\!]}$ is empty too. This is in contrast to the Harmer-McCusker model, in which $\sigma;\tau$ may contain a divergence even if neither σ nor τ does.

4.3 Proving Full Abstraction

We write $\mathcal{G}_{ND}$ for the category we have defined. It may be made into a symmetric monoidal closed category in a straightforward way, following [Harmer and McCusker 1999]. Given $\sigma\colon A \multimap C$ and $\tau\colon B \multimap D$, $D_{\sigma\otimes\tau}$ is the set of all $s \in P_{(A\otimes B)\multimap(C\otimes D)}$ such that either $s|_{A,C} \in D_\sigma$ or $s|_{B,D} \in D_\tau$.

Similarly, we can define exponentials in our model, where the divergences in $!\sigma$ are those positions that are divergent in one of their components.

Although we have solved the problems with associativity of composition, it turns out to be difficult to prove computational adequacy or full abstraction for this model. The main problem is a lack of continuity: we can follow [Harmer and McCusker 1999] and define an ordering on strategies, but composition is no longer continuous with respect to this ordering. This should not be seen as a defect of our model; indeed, we showed earlier on that the language we are trying to model has a similar lack of continuity, so if our model were continuous it would indicate that it did not model the language faithfully enough.

Our approach to proving full abstraction will be to define a new model C of our language that is fully abstract for nondeterministic PCF, and to show that $\mathcal{C}$ admits a faithful monoidal functor into $\mathcal{G}_{ND}$, which is enough to show that $\mathcal{G}_{ND}$ is also fully abstract. Before we do this, however, we outline an alternative argument that does not go via this supplementary category. The argument is suggested in [Harmer 1999, 3.7.2] as a means of defining an appropriate notion of 'nondeterministic innocent strategy'.

We first define the strategy $\top_{\mathbb{N}}$ on $\mathbb{N}$ by $T_{\top_{\mathbb{N}}} = P_{\mathbb{N}}$ and $D_{\top_{\mathbb{N}}} = \varnothing$. Then we say that a strategy $\sigma \colon A \multimap B$ is *deterministic history free* if it can be written as

$$A \xrightarrow{\text{oracle}_A} !\mathbb{N} \otimes A \xrightarrow{\sigma'} B$$

where $\text{oracle}_A = \text{lunit}; !\top_{\mathbb{N}} \otimes A$ and $\sigma' \colon !\mathbb{N} \otimes A \multimap B$ is a deterministic, history-free strategy. It is possible to show that the composition and tensor product of such strategies can again be written in this form, and it is then possible to prove full abstraction using properties of deterministic strategies. When we have defined our model $\mathcal{C}$, we will return to this point to show that the two arguments are essentially the same.

5 Cones on Monoidal Functors

5.1 Surjective and winning strategies

A useful fact about the category $\mathcal{G}$ of games and deterministic strategies, proved in [Hyland and Schalk 2002], is that it admits a faithful, monoidal and linearly distributive functor into the category **Rel** of sets and relations, with monoidal structure given by the Cartesian product and exponentials given by the multiset comonad W. The functor sends a game A to the set P_A and sends a strategy $\sigma \colon A \multimap B$ to the relation

$$\underline{\sigma} = \{(s|_A, s|_B) : s \in \sigma\} \subseteq P_A \times P_B$$

Given a strategy $\sigma \colon A \multimap B$, we say that σ is *surjective* if $\underline{\sigma}$ is a surjective relation; i.e., if for any $b \in P_B$ there exists some $s \in \sigma$ such that $s|_B = b$. Monoidality and linear distributivity of the Hyland-Schalk functor $\mathcal{G} \longrightarrow$ **Rel** show that the composition of surjective strategies is surjective, as is the tensor product of surjective strategies and the bang of a surjective strategy.

Given a strategy $\sigma \colon A \multimap B$, we say that σ is *winning* if σ is total and if for all positions $s \in \sigma$, if $sb \in \sigma$, where b is an O-move in B, then there is some $t \in \sigma$ such that $sb \sqsubseteq t$ and t ends with a move in B.

Once again, the composition, tensor product and bang of winning strategies is still winning. One way of seeing this is by relating our definition to the definition of a winning strategy given in [Abramsky and Jagadeesan 1994], in which a game comes attached with a set of winning infinite positions. Then we say that a strategy σ is winning if it is total and if every infinite position arising as the limit of plays in σ is a winning position. If A and B are such that every infinite position is winning, then our notion of a winning strategy coincides with that notion.

The reason we are interested in surjective winning strategies is that they preserve top strategies:

Lemma 5.1. *We say that a game A is P-*winning *if for all O-positions $s \in P_A$ there exists some a such that $sa \in P_A$. Let A, B be P-winning games, and let $\sigma \colon A \multimap B$ be a surjective winning strategy. Write $\top_A$ for the (nondeterministic) strategy given by $T_{\top_A} = P_A$ and $D_{\top_A} = P_A^{\omega}$ (i.e., the only divergent plays are the infinite ones), and do the same for B. Then $\top_A; \sigma = \top_B$.*

Proof. We can show that $T_{\top_A;\sigma} = T_{\top_B}$ from surjectivity of σ, using the Hyland-Schalk functor, since the equivalent result clearly holds in **Rel**. Now suppose that s is some divergent position in $\top_A; \sigma$. Then $s = \mathfrak{s}|_B$, where $\mathfrak{s}$ is a position in σ and $\mathfrak{s}|_A$ is a divergent position in A. Then $\mathfrak{s}|_A$ must be infinite; since σ is winning, this means that $\mathfrak{s}|_B$ must be infinite as well, and therefore that $s \in D_{\top_B}$. □

We now define a category $\mathcal{G}_{sw}$ whose objects are P-winning games and where the morphisms from a game A to a game B are deterministic surjective winning strategies from A to B. We write $\mathcal{G}_{sw,rec}$ for the category where the morphisms are *recursive* surjective, winning strategies.

Write j for the inclusion $\mathcal{G}_{sw,rec} \hookrightarrow \mathcal{G}_{rec}$, and write J for the inclusion functor $\mathcal{G} \hookrightarrow \mathcal{G}_{ND}$ that sends a deterministic strategy σ to the strategy given by $T = \sigma$ and $D = \text{pocl}\{s \in \sigma^- : sa \notin \sigma \forall a\}$ Then the lemma we have just proves means that

$$\top \colon I \Rightarrow J \circ j$$

is a natural transformation, where I is the constant I functor.

Moreover, we would expect Lemma 5.1 to hold in *any* sensible game-semantics model of nondeterministic PCF, so the following makes sense as a definition

Definition 5.2. A *game semantics category for nondeterministic PCF* is a category $\mathcal{D}$ together with a functor $J \colon \mathcal{G}_{rec} \to \mathcal{D}$ and a natural transformation $\top \colon I \Rightarrow J \circ j \colon \mathcal{G}_{sw,rec} \to \mathcal{D}$.

5.2 Generalized monoidal natural transformations

We will need one more technical definition.

Let $\mathcal{C}',\mathcal{C}',\mathcal{D}',\mathcal{D}$ be monoidal categories, let $f \colon \mathcal{C}' \to \mathcal{C}$, $g \colon \mathcal{D}' \to \mathcal{D}$ be oplax monoidal functors and let $F \colon \mathcal{D}' \to \mathcal{C}'$, $G \colon \mathcal{D} \to \mathcal{C}$ be lax monoidal functors, as in Figure 2.

$$\begin{tikzcd}
\mathcal{D}' \arrow[rr, "F"] \arrow[d, "g"'] & {} \arrow[d, Rightarrow, "\phi"] & \mathcal{C}' \arrow[d, "f"] \\
\mathcal{D} \arrow[rr, "G"'] & {} & \mathcal{C}
\end{tikzcd}$$

Figure 2. A generalized monoidal natural transformation ϕ is a particular sort of natural transformation between composites of lax and oplax monoidal functors.

We say that a natural transformation $\phi\colon fF \Rightarrow Gg$ is a *generalized monoidal natural transformation* if the following two diagrams commute for all objects X, Y of $\mathcal{D}'$:

$$\begin{tikzcd}
fFX \otimes fFY \arrow[r, "\phi_X \otimes \phi_Y"] & GgX \otimes GgY \arrow[d, "m_G"] \\
f(FX \otimes FY) \arrow[u, "m_f"] \arrow[d, "fm_F"'] & G(gX \otimes gY) \\
fF(X \otimes Y) \arrow[r, "\phi_{X \otimes Y}"] & Gg(X \otimes Y) \arrow[u, "Gm_g"']
\end{tikzcd}$$

$$\begin{tikzcd}
& fFI_{\mathcal{D}'} \arrow[r, "\phi_I"] & GgI_{\mathcal{D}'} \arrow[dr, "G\epsilon_g"] & \\
fI_{\mathcal{C}'} \arrow[ur, "f\epsilon_F"] \arrow[drr, "\epsilon_f"] & & & GI_{\mathcal{D}} . \\
& & I_{\mathcal{C}} \arrow[ur, "\epsilon_G"] &
\end{tikzcd}$$

The particular structure that generalized natural transformations fit into is that of a *double category* (see [Grandis and Paré 1999] for the definition); in particular, the double category **SymmMonCat** introduced in [Grandis 2004], in which the objects are symmetric monoidal categories, the horizontal morphisms are lax symmetric monoidal functors, the vertical morphisms are oplax symmetric monoidal functors and the 2-cells are generalized monoidal natural transformations as defined above.

Generalized monoidal natural transformations may thus be composed either horizontally or vertically in the usual way; i.e., given a pair of squares as in Figure 2 that may be pasted together along an edge, then we may define a generalized monoidal natural transformation across the pasted square. We write $(\phi|\psi)$ for the horizontal composition of natural transformations ϕ, ψ pasted together along an oplax monoidal functor and $\left(\frac{\phi}{\psi}\right)$ for the vertical composition of natural transformations ϕ, ψ pasted together along a lax monoidal natural functor.

5.3 The cone of a monoidal functor

Let $j\colon \mathcal{C}' \to \mathcal{C}$ be an oplax monoidal functor. We define a *j-category* to be a category $\mathcal{D}$ together with a lax monoidal functor $J\colon \mathcal{C} \to \mathcal{D}$ and a generalized natural transformation $\phi\colon I \to Jj$:

$$\begin{tikzcd}
\mathcal{C}' \arrow[rr, "()"] \arrow[d, "j"'] & {} \arrow[d, Rightarrow, "\phi"] & I \arrow[d, "I"] \\
\mathcal{C} \arrow[rr, "J"'] & {} & \mathcal{D}
\end{tikzcd} .$$

Example 5.3. For example, our previous definition of a game semantics category for nondeterministic PCF can be thought of as a j-category, where j is the inclusion functor $\mathcal{G}_{sw,rec} \hookrightarrow \mathcal{G}_{rec}$.

We define a *morphism of j-categories* going from a j-category $(\mathcal{D}', J', \phi')$ to a j-category $(\mathcal{D}, J, \phi)$ to be a lax monoidal functor $\mathcal{F}\colon \mathcal{D}' \to \mathcal{D}$, sending I to I, such that $J = \mathcal{F} \circ J'$ and ϕ is the vertical composition of ϕ' with the 'identity transformation' given by this identity; i.e.:

$$\begin{tikzcd}
\mathcal{C}' \arrow[rr, "()"] \arrow[d, "j"'] & {} \arrow[dd, Rightarrow, "\phi"'] \arrow[d, Rightarrow, shift left=4, "\phi'"] & I \arrow[d, "I"] \\
\mathcal{C} \arrow[rr, "J'"'] \arrow[d, "\text{id}"'] & {} \arrow[d, Rightarrow, shift left=4, "\text{id}"] & \mathcal{D}' \arrow[d, "\mathcal{F}"] \\
\mathcal{C} \arrow[rr, "J"] & {} & \mathcal{D}
\end{tikzcd} .$$

Our main technical result in this section will be the following:

Proposition 5.4. *The category j-***Cat** *of j-categories and j-category morphisms has an initial object $\mathcal{C}_j$*

By analogy with homotopy theory, we call this initial object the *mapping cone* or *cone* on j.

To see why this is useful, consider another important topic from the theory of semantics. We say that a j-category $(\mathcal{D}, J, \phi)$ satisfies *ϕ-factorization* if for any pair of objects A and B of $\mathcal{C}$ and for every morphism $f\colon JA \to JB$ in $\mathcal{D}$, we may write f as a composite

$$JA \xrightarrow{\text{lunit};\phi_X \otimes JA} JjX \otimes JA \xrightarrow{m_J} J(jX \otimes A) \xrightarrow{Jf'} JB ,$$

where X is some object of $\mathcal{C}'$ and $f'\colon jX \otimes A \to B$ is some morphism in $\mathcal{C}$.

Example 5.5. In the case that j is the inclusion $\mathcal{G}_{sw,rec} \hookrightarrow \mathcal{G}_{rec}$, and $\phi = \top$, ϕ-factorization is the same as *deterministic factorization*: the idea that every morphism can be written as the composition of a deterministic recursive strategy with $\top_X$ for some X.

Then the following result is easy to prove:

Proposition 5.6. *Let $(\mathcal{D}, J, \phi)$ be a j-category satisfying ϕ-factorization. Suppose $\mathcal{F}\colon \mathcal{D}' \to \mathcal{D}$ is a morphism of j-categories. Then $\mathcal{F}$ is a full functor.*

In particular, the unique j-category morphism $C_j \longrightarrow \mathcal{D}$ is a full functor. This is useful, because we are viewing game semantic models of nondeterministic PCF as j-categories (for a particular j), and therefore we can see that C_j admits a full functor into every such model that satisfies deterministic factorization. Since we typically require that our models satisfy such a factorization result, we can form any such model by taking C_j and imposing some equivalence relation on the morphisms. This suggests that we might consider C_j itself as our model of nondeterministic PCF.

5.4 Constructing the cone

We will now define what the cone C_j is and give an outline of why it is initial in the category of j-categories.

Recall that we start with an oplax symmetric monoidal functor $j\colon C' \longrightarrow C$. The objects of C_j are the objects of C. Given objects A, B of C, the class of morphisms $A \longrightarrow B$ is given by the colimit of the functor

$$C'^{\mathrm{op}} \xrightarrow{j^{\mathrm{op}}} C^{\mathrm{op}} \xrightarrow{C[\_\_ \otimes A, B]} \mathbf{Set}\,.$$

That is, a morphism from A to B is a pair (X, f), where X is an object of C' and $f\colon jX \otimes A \longrightarrow B$ is a morphism in C. Furthermore, (X, f) and (X', f') are considered to be equivalent if there is some morphism $h\colon X' \longrightarrow X$ in C' making the following diagram commute:

$$\begin{tikzcd}
jX' \otimes A \arrow[r, "f'"] \arrow[d, "jh\otimes A"'] & B \\
jX \otimes A \arrow[ur, "f"'] &
\end{tikzcd}.$$

Given morphisms $(X, f)\colon A \longrightarrow B$, $(Y, g)\colon B \longrightarrow C$, the composite is given by $(Y \otimes X, f; g)$, where $f; g$ is the following composite in C:

$$\begin{aligned}
j(Y \otimes X) \otimes A &\xrightarrow{m_j \otimes A} (jY \otimes jX) \otimes A \\
&\xrightarrow{\text{assoc}} jY \otimes (jX \otimes A) \\
&\xrightarrow{jY \otimes f} jY \otimes B \xrightarrow{g} C\,.
\end{aligned}$$

We can prove that this composition is associative using the usual coherence theorems for monoidal categories. Similarly, the symmetric monoidal structure from C induces a symmetric monoidal structure on C_j.

There is a functor $J\colon C \longrightarrow C_j$ that is the identity on objects and which sends a strategy $f\colon A \longrightarrow B$ to the strategy

$$jI \otimes A \xrightarrow{\epsilon_j \otimes A} I \otimes A \xrightarrow{\text{lunit}^{-1}} A \xrightarrow{f} B\,.$$

This functor is (strong) symmetric monoidal with respect to the monoidal structure on C_j. Lastly, the natural transformation $\phi\colon I \Rightarrow J \circ j$ is given by the composite

$$\phi_X = jX \otimes I \xrightarrow{\text{runit}^{-1}} jX\,,$$

considered as a morphism from I to jX in C_j. Proving that this is indeed a natural transformation makes essential use of the equivalence relation on morphisms in C_j. We can also prove that ϕ is a generalized monoidal natural transformation. Therefore, C_j is a j-category.

Now suppose that $(\mathcal{D}, F, \psi)$ is another j-category. We define a functor $H\colon C_j \longrightarrow \mathcal{D}$ that sends an object $A = JA$ in C_j to the object FA in $\mathcal{D}$. Moreover, we send a morphism $f\colon jX \otimes A \longrightarrow B$ from JA to JB in C_j to the following composite in $\mathcal{D}$:

$$FA \xrightarrow{\text{lunit};\psi_X \otimes FA} FjX \otimes FA \xrightarrow{m_F} F(jX \otimes A) \xrightarrow{Ff} FB\,.$$

Proving that this is indeed a functor is a little tricky, and relies on ψ being a generalized monoidal natural transformation. We can then go through and check that H is indeed a morphism of j-categories.

In order to prove uniqueness, note that, as a morphism of j-categories, H is determined on morphisms of the form Jf by the equation $F = H \circ J$ and on morphisms of the form ϕ_X by the equation $\psi = \left(\frac{\phi}{H}\right)$. But now, if $f\colon jX \otimes A \longrightarrow B$ is any morphism from A to B in C_j, we may write f as the composite

$$JA \xrightarrow{\text{lunit};\phi_X \otimes A} JjX \otimes JA \xrightarrow{m_J} J(jX \otimes A) \xrightarrow{Jf} JB\,;$$

i.e., C_j satisfies ϕ-factorization. Since the functor H is required to be monoidal, this implies that it is determined everywhere on C_j by the requirement that it be a morphism of j-categories. Therefore, C_j is initial in the category of j-categories.

A consequence of the initiality of the cone construction is that it is *functorial*; that is, given a square of the following form:

$$\begin{tikzcd}
C' \arrow[r, "F"] \arrow[d, "j"'] \arrow[dr, phantom, "\Downarrow \psi"] & \mathcal{D}' \arrow[d, "j'"] \\
C \arrow[r, "G"'] & \mathcal{D}
\end{tikzcd},$$

where horizontal arrows correspond to lax monoidal functors and vertical arrows to monoidal functors, we get an induced lax monoidal functor $C_j \longrightarrow C_j$, in a way that respects horizontal pasting of squares.

6 Denotational Semantics

In order to model *countable* nondeterminism, rather than arbitrary nondeterminism, we want to constrain our model so that

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. nnnnnnn and Grant No. mmmmmmm. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation.

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

Marco Grandis and Robert Paré. 1999. Limits in double categories. *Cah. Topologie Géom. Différ. Catégoriques* 40, 3 (1999), 162–220.

Pare Robert Grandis, Marco. 2004. Adjoint for double categories. *Cahiers de Topologie et Géométrie Différentielle Catégoriques* 45, 3 (2004), 193–240. http://eudml.org/doc/91684

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

Martin Hyland and Andrea Schalk. 2002. Games on graphs and sequentially realizable functionals. Extended abstract. In *Logic in Computer Science, 2002. Proceedings. 17th Annual IEEE Symposium on.* IEEE, 257–264.

James Laird. [n. d.]. Functional programs as coroutines: A semantic analysis. *Logical methods in computer science* ([n. d.]). To appear.

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

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. 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

A Appendix

Text of appendix …