Chapter 1

Further Directions

1.1 Stateless Languages

In this thesis, the base language we have used has been the stateful language Idealized Algol. One area to explore is whether the techniques we have developed allow us to build models of effectful versions of stateless languages such as PCF.

One immediate problem we face is that the addition of nondeterministic effects to PCF allows us to distinguish terms that cannot be distinguished within PCF itself. Consider, for instance, the PCF term

$$M = \lambda x^{\mathtt{bool}}.\,\mathsf{If}\ x\ \mathsf{then}\ \Omega\ \mathsf{else}\ (\mathsf{If}\ x\ \mathsf{then}\ \mathbb{t}\ \mathsf{else}\ \Omega)$$

is observationally equivalent (in PCF) to Ω – informally, because the term x must have the same 'value' both time it is called, so if it is false the first time, then it must be false the second time. However, the same term inside PCF with finite nondeterminism is not observationally equivalent to Ω: if we substitute nondeterministic choice in for x, then it could be false the first time and true the second, causing the term to converge to $\mathbb{t}$.

This means that if $\mathcal{G}$ is a 'truly fully abstract' model of PCF – i.e., one in which observational equivalence of terms corresponds to equality of morphisms rather than intrinsic equivalence[1] – then there can be no natural inclusion functor from $\mathcal{G}$ into a model of PCF with finite nondeterminism.

[1]As an example, take any model that is full abstract in our sense, and take the quotient by the intrinsic equivalence relation.

Indeed, in such a model $[\![M]\!]$ and $[\![\Omega]\!]$ are the same morphism, so they would have to be sent to the same morphism in the model of nondeterministic PCF.

This is a problem for us, since both our techniques – the Kleisli category and the $\mathcal{G}/\mathcal{X}$ construction – have a natural functor coming out of the original category $\mathcal{G}$.

All is not lost, however, since we can still try and impose some condition on the original model $\mathcal{G}$ of PCF to say that it is still fine-grained enough to distinguish between certain morphisms that are denotations of observationally equivalent PCF terms. One convenient way to do this is to ask that $\mathcal{G}$ should embed into some model $\mathcal{G}^{IA}$ of Idealized Algol. An obvious case when this holds is when $\mathcal{G}$ is the (unquotiented) Hyland-Ong category of arenas and innocent strategies, and $\mathcal{G}^{IA}$ is the category of arenas and single-threaded strategies.

For example, fix $X \in \{\mathbb{B}, \mathbb{N}\}$ and let *Nondeterministic PCF* be the langauge PCF, together with a constant $\mathtt{ask}_X \colon \mathtt{bool}$ that plays the role of a nondeterministic oracle. I.e., it is the language with types given by the following inductive grammar.

$$T ::= \mathtt{bool} \mid \mathtt{nat} \mid T \to T$$

And with a type theory as shown in Figure 1.1. We endow this language with an operational semantics of may testing. Define a *canonical form* of the language to be a term taking one of the following forms.

- At type $\mathtt{bool}$, the constants $\mathbb{t}$ and $\mathbb{f}$;
- at type $\mathtt{nat}$, the numerals n; and
- at type $S \to T$, terms of the form $\lambda x^S.M$.

We then define a relation $M \Downarrow c$ (read '*M may converge to c*'), for M a term of Nondeterministic PCF and c a canonical form, inductively as in Figure 1.2.

It is a quick check to see that every term of Nondeterministic PCF may be regarded as a term of IA_X, giving us a denotational semantics of the language within $\mathrm{Kl}_{R_X}\,\mathcal{G}$, where $\mathcal{G}$ is the category of arenas and single-threaded strategies. But we can be more precise than this: we know from [HO00] that the denotation of any term of ordinary PCF is an innocent strategy, so that the denotational semantics of PCF within $\mathcal{G}$ factors through the inclusion $\mathcal{G}_{inn} \hookrightarrow \mathcal{G}$, where $\mathcal{G}_{inn}$ is the category of arenas and innocent strategies. In particular, the denotation within $\mathrm{Kl}_{R_X}\,\mathcal{G}$ of any term of ordinary PCF lives

$$\frac{}{\Gamma, x\colon T \vdash x\colon T} \qquad \frac{\Gamma \vdash M\colon S \to T \qquad \Gamma \vdash N\colon S}{\Gamma \vdash MN\colon T} \qquad \frac{\Gamma, x\colon S \vdash M\colon T}{\Gamma \vdash \lambda x^S.M\colon S \to T}$$

$$\frac{}{\Gamma \vdash \mathbb{t}\colon \texttt{bool}} \qquad \frac{}{\Gamma \vdash \mathbb{f}\colon \texttt{bool}}$$

$$\frac{\Gamma \vdash M\colon \texttt{bool} \qquad \Gamma \vdash N\colon T \qquad \Gamma \vdash P\colon T}{\Gamma \vdash \mathsf{If}\, M\ \mathsf{then}\ N\ \mathsf{else}\ P\colon T}\ T \in \{\texttt{bool}, \texttt{nat}\}$$

$$\frac{}{\Gamma \vdash n\colon \texttt{nat}} \qquad \frac{\Gamma \vdash M\colon \texttt{nat}}{\Gamma \vdash \mathsf{succ}\, M\colon \texttt{nat}} \qquad \frac{\Gamma \vdash M\colon \texttt{nat}}{\Gamma \vdash \mathsf{pred}\, M\colon \texttt{nat}}$$

$$\frac{\Gamma \vdash M\colon \texttt{nat} \qquad \Gamma \vdash N\colon T \qquad \Gamma \vdash P\colon T}{\Gamma \vdash \mathsf{If0}\, M\ \mathsf{then}\ N\ \mathsf{else}\ P\colon T}\ T \in \{\texttt{bool}, \texttt{nat}\}$$

$$\frac{\Gamma \vdash M\colon T \to T}{\Gamma \vdash \mathbf{Y}_T M\colon T} \qquad \frac{}{\mathsf{ask}_X\colon \texttt{bool}}$$

Figure 1.1: Type theory for a variant of PCF with nondeterminism. Note that if the nondeterminism is infinite (i.e., $X = \mathbb{N}$), then it is common (see [Lai15] for nondeterminism and [EPT18] for the probabilistic case) to modify the conditional construct $\mathsf{If0}\, M$ then N else P so that the value of M is passed into P (so that the typing rule for If0 becomes

$$\frac{\Gamma \vdash M\colon \texttt{nat} \qquad \Gamma \vdash N\colon T \qquad \Gamma \vdash P\colon \texttt{nat} \to T}{\Gamma \vdash \mathsf{If0}\, M\ \mathsf{then}\ N\ \mathsf{else}\ P\colon T}\ T \in \{\texttt{bool}, \texttt{nat}\} \quad .$$

The idea behind this is that if M evaluates to 0 then N will be evaluated, and if it evaluates to $n+1$ then $P\,n$ will be evaluated. In ordinary deterministic PCF, this does not change the expressive power of the language, since we could always write

$$\mathsf{If0}\, M\ \mathsf{then}\ N\ \mathsf{else}\ (P\, M)\,.$$

However, this does not work if M may exhibit nondeterministic behaviour, since it may evaluate to two different values the two times it is called. We will stick with the variant shown above for the sake of simplicity, but nothing changes in our proofs if we choose the other version.

$$\frac{}{c \Downarrow c} \qquad \frac{M \Downarrow \lambda x.M' \qquad M'[N/x] \Downarrow c}{M\,N \Downarrow c}$$

$$\frac{M \Downarrow \mathbb{tt} \qquad N \Downarrow c}{\mathsf{If}\,M\ \mathsf{then}\ N\ \mathsf{else}\ P \Downarrow c} \qquad \frac{M \Downarrow \mathbb{ff} \qquad P \Downarrow c}{\mathsf{If}\,M\ \mathsf{then}\ N\ \mathsf{else}\ P \Downarrow c}$$

$$\frac{M \Downarrow n}{\mathsf{succ}\,M \Downarrow n+1} \qquad \frac{M \Downarrow n+1}{\mathsf{pred}\,M \Downarrow n} \qquad \frac{M \Downarrow 0}{\mathsf{pred}\,M \Downarrow 0}$$

$$\frac{M \Downarrow 0 \qquad N \Downarrow c}{\mathsf{If0}\,M\ \mathsf{then}\ N\ \mathsf{else}\ P \Downarrow c} \qquad \frac{M \Downarrow n+1 \qquad P \Downarrow c}{\mathsf{If0}\,M\ \mathsf{then}\ N\ \mathsf{else}\ P \Downarrow c}$$

$$\frac{M(\mathbf{Y}M) \Downarrow c}{\mathbf{Y}M \Downarrow c} \qquad \frac{}{\mathsf{ask}_X \Downarrow \mathbb{tt}} \qquad \frac{}{\mathsf{ask}_X \Downarrow \mathbb{ff}}$$

Figure 1.2: Big-step operational semantics for Nondeterministic PCF with may testing

within $\mathrm{Kl}_{R_X}\,\mathcal{G}_{inn}$. Moreover, the denotation of the new term ask_X, being given by the identity morphism in $\mathcal{G}$, also lives in $\mathrm{Kl}_{R_X}\,\mathcal{G}_{inn}$, meaning that the whole of Nondeterministic PCF is interpreted within this category.

It is also quick to check that our operational semantics in Figure 1.2 may be regarded as a subset of the rules for IA_X with May Testing, as we studied in §**??**. Specifically, if M is a term of PCF and c a canonical form, then $M \Downarrow c$ if and only if

$$\_\,,() \vdash M \Downarrow c, ()$$

in IA_X.

We want to use our Computational Adequacy result for IA_X with May Testing (Corollary **??**) to get a Computational Adequacy result for Nondeterministic PCF. The only slight problem is that Corollary **??** is stated for terms of type `com`, whereas our language has no such type. We get around this by using `nat` as our main ground type rather than `com`, and by using the following easy lemma.

Lemma 1.1.1. *Let M:* `nat` *be a term of IA_X. Then there exists n such* *that $\Gamma, s \vdash M \Downarrow n, s'$ if and only if*

$$\Gamma, s \vdash \mathsf{If0}\, M \text{ } \mathsf{then\ skip\ else\ skip} \Downarrow \mathsf{skip}, s' \,.$$

We will write $\delta \colon \mathbb{N} \to \mathbb{C}$ for the denotation of the term-in-context

$$x \colon \mathtt{nat} \vdash \mathsf{If0}\, x \text{ } \mathsf{then\ skip\ else\ skip} \colon \mathtt{com}\,.$$

Then we immediately get the following result as a special case of Corollary **??**.

Corollary 1.1.2 (Computational Adequacy for Nondeterministic PCF)**.** *Let $M \colon \mathtt{nat}$ be a closed term of Nondeterministic PCF. Consider the denotation $[\![M]\!] : 1 \to \mathbb{N}$ in $\mathrm{Kl}_{R_X}\, \mathcal{G}_{inn}$ as a morphism $1 \to (X \to \mathbb{N})$ in $\mathcal{G}_{inn}$, and thence as a morphism $1 \to (X \to \mathbb{N})$ in $\mathcal{G}$.*

Then there exists some sequence $u \in X^*$ such that the composite

$$1 \xrightarrow{[\![M]\!]} (X \to \mathbb{N}) \xrightarrow{X \to \delta} (X \to \mathbb{C}) \xrightarrow{\eta_u} (\mathrm{Var} \to \mathbb{N}) \xrightarrow{[\![\mathsf{new}]\!]} \mathbb{N} \xrightarrow{t_{|u|}} \mathbb{C}$$

is not equal to $\bot$, if and only if $M \Downarrow n$ for some n.

Note that Corollary 1.1.2, which does not depend on any specific details of the models $\mathcal{G}_{inn}$ and $\mathcal{G}$ beyond the fact that they are computationally adequate for ordinary PCF and Idealized Algol, uses morphisms δ, η_u, $[\![\mathsf{new}]\!]$ and $\mathsf{t}_{|u|}$ that do not occur in the category $\mathcal{G}_{inn}$.

In the case that $\mathcal{G}$ is the category of arenas and single-threaded strategies, and $\mathcal{G}_{inn}$ the category of arenas and innocent strategies, we can restate Corollary 1.1.2 in a way that does not refer to Idealized Algol terms at all.

Corollary 1.1.3. *Let $M \colon \mathtt{nat}$ be a closed term of Nondeterministic PCF and consider the denotation $[\![M]\!] : 1 \to \mathbb{N}$ in $\mathrm{Kl}_{R_X}\, \mathcal{G}_{inn}$ as a morphism $1 \to (X \to \mathbb{N})$ in $\mathcal{G}_{inn}$.*

Then, for all $n \in \mathbb{N}$, there exists some sequence $s \in [\![M]\!]$ such that $s|_{\mathbb{N}} = qn$ if and only if $M \Downarrow n$.

Proof. Corollary 1.1.2, together with our earlier analysis, immediately tells us that we have

$$\exists s \in [\![M]\!] \ . \ s|_{\mathbb{N}} = qn \text{ for some } n \Leftrightarrow \exists n \,.\, M \Downarrow n\,.$$

The problem is that the two n's might be different. However, we can obtain the desired result for a given n from this one by composing with an appropriate term `nat` $\to$ `nat` that converges if and only if its input is equal to n. $\square$

The passage to full abstraction, via innocent definability for PCF is essentially the same as for Idealized Algol (though if $X = \mathbb{N}$, then we need to cut down to a category of *recursive* strategies).

If we so desire, we can continue with the programme from §**??**, declaring two Kleisli strategies $A \to (X \to B)$ to be equivalent if they contain the same plays after restriction to $A \to B$. We can then do away with the plays in X altogether and give a model that involves only nondeterministic strategies.

At this point, we run into a well-known problem: under what circumstances should a nondeterministic strategy be described as *innocent*? The answer that we get from our model is that it is innocent if and only if it takes the form

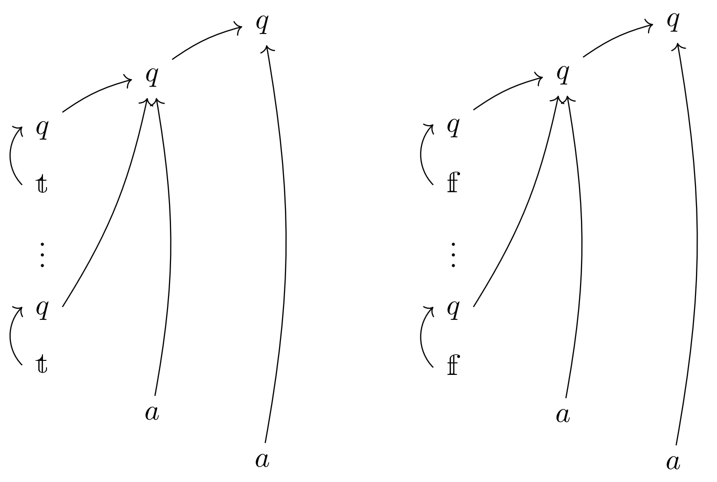
$$A \xrightarrow{\sigma} (X \to B) \xrightarrow{[\![\mathsf{ask}_X]\!] \to B} B\,,$$

where σ is an innocent deterministic strategy. This is the definition considered – and ultimately rejected for being too indirect – by Harmer in his thesis [Har99, §3.7]. This definition is correct, but, perhaps surprisingly, does not coincide with the same definition that we would get by naively applying the usual definition of innocence to nondeterministic strategies.

Consider, for example, the denotation of the term

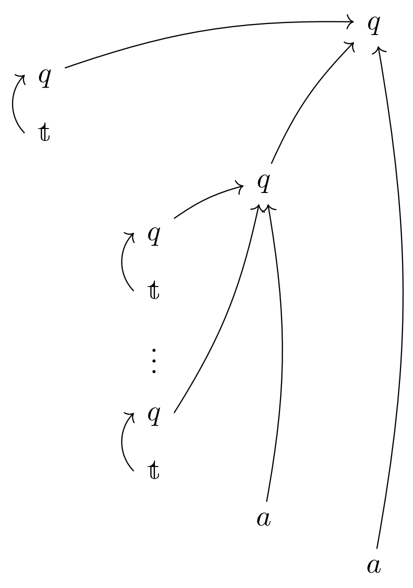
$$\mathsf{If}\ \mathsf{ask}_{\mathbb{B}}\ \mathsf{then}\ (\lambda f.f\,\mathbb{t})\ \mathsf{else}\ (\lambda f.f\,\mathbb{f})\,,$$

which has maximal plays taking one of the following two forms.



Note that this strategy displays typically non-innocent behaviour: if player P has played $\mathbb{t}$ on the left, then she must play $\mathbb{t}$ again whenever player O asks, even though the original move $\mathbb{t}$ occurs outside the current P-view.

In the Kleisli category, the innocence of this strategy becomes clear: plays are either of the form



or of the form

$$\begin{tikzcd}
 & & & q \\
q \arrow[rrru] & & & \\
\mathbb{ff} \arrow[u, bend left] & & & \\
 & & q \arrow[ruuu] & \\
 & q \arrow[ru] & & \\
 & \mathbb{ff} \arrow[u, bend left] & & \\
 & \vdots & & \\
 & q \arrow[ruuuu] & & \\
 & \mathbb{ff} \arrow[u, bend left] & & \\
 & & a \arrow[uuuuuu] & \\
 & & & a\,, \arrow[uuuuuuuuuu]
\end{tikzcd}$$

where now the move $\mathbb{tt}$ or $\mathbb{ff}$ in the nondeterministic oracle game (on the very left) locks the 'branching-time information' into the P-view.

There is a surprising and elegant definition of nondeterministic innocence (see Levy [Lev14], with the correction given by Tsukada and Ong in [TO15, Proposition 46]), which we shall not detail here. Tsukada and Ong have also demonstrated that it is instructive to consider strategies as presheaves over plays, where we assign to each individual play a set of possible branches that lead us to that play. In this formalism, the innocent nondeterministic strategies are preciesly those presheaves that are sheaves with respect to a certain natural Grothendieck topology.

What, then, can our Kleisli category model do for us? One answer is that it provides a general setting in which we can prove results such as adequacy without worrying about the specific details of a nondeterministic construction. For example, we can prove a computational adequacy result for Tsukada and Ong's model $\mathcal{G}_{TO}$ from Corollary 1.1.3 by noting that the denotational semantics in that category factors through the natural functor $\mathrm{Kl}_{R_{\mathbb{B}}}\,\mathcal{G} \to \mathcal{G}_{TO}$. Of course, Tsukada and Ong have provided their own proof of Computational Adequacy in the finite nondeterminism case, but this method could be useful if, for example, we wanted to extend their model to deal with countable nondeterminism.

1.2 Stateful Effects

In Chapters **??** and **??**, we presented a categorical algebra of scoped state via sequoidal categories. A natural question to ask is whether we could use the techniques we have developed in the remainder of this thesis in the stateful case. Indeed, the state monad

$$A \mapsto (W \to (A \times W))$$

can be constructed in categories of games. More ambitiously, we might hope to capture a parametric notion of local state, via the action

$$(R, W).A = R \to (A \times W)\,.$$

of $\mathcal{G}^{\mathrm{op}} \times \mathcal{G}$ on $\mathcal{G}$.

In this section, we will present some of the problems involved in this case.

One issue, which we have touched on already, is that the state monad is not a monoidal functor, which was the condition we needed in order to ensure that its Kleisli category inherited a monoidal structure. Indeed, there is not obvious natural morphism

$$(W \to (A \times W)) \times (W \to (B \times W)) \to (W \to ((A \times B) \times W)\,.$$

The parametric version suffers from a different problem. Suppose we have two Melliès morphisms

$$f\colon A \to (R \to (B \times W)) \qquad\qquad g\colon B \to (R' \to (C \times W'))\,.$$

Then their composition will be given by a Melliès morphism

$$f;g\colon A \to ((R \times R') \to (C \times (W \times W')))\ :$$

in other words, we will have combined the two inputs R and R' and the two outputs W and W', without any way of specifying that the output state W from f should be used as the input state R' from g.

The difficulty is that we might want W and R' to represent different variables, in which case this is the right thing to do, but we might want them to represent the same storage cell, in which case, the output from W should be fed into R'. It's not clear how to deal with this in a category-theoretic way.

Linear type theory provides a perspective on this problem. Note that the state monad and the parametric action above can both be defined inside an arbitrary monoidal category. However, if we want our language to exhibit any typically stateful behaviour, then we need to be able to refer to the same variable more than once: there is little point writing a value into a cell if we are not able to read from it later. This is not a problem in a monolithic stateful system – such as we have with the state monad – but if we want to refer to individual storage cells using variables in the language, then we need our category to admit diagonals, to allow us to refer to a storage cell twice. Since the state action above does not interact in any way with the Cartesian structure of the category (i.e., the diagonals and terminal morphisms), it cannot hope to capture local state in this way.

One avenue that could shed some light on this issue is the sequoidal semantics which we developed for game semantics, in which the stateful behaviour is intimately connected to the Cartesian structure of the category, via the exponential.

There is also a *local state monad*, due to Plotkin and Power [PP02] and based on a construction by Levy [Lev01], which is defined on the category $[\mathbf{Inj}, \mathbf{Set}]$ of functors from $\mathbf{Inj}$, the category of natural numbers and injections, into the category of sets. It is given by the following formula, where V denotes a fixes set of values.

$$(TA)(n) = V^n \to \left(\int^{p\colon \mathbf{Inj}} V^p \times A(p) \times \mathbf{Inj}(n, p)\right)$$

Here, if $A\colon \mathbf{Inj} \to \mathbf{Set}$ is a functor, then we think of $A(n)$ as the set of all values that A takes on in the presence of n local variables taking values in V.

We cannot construct this coend in the category of games, but there is a clear link with our constructions with probabilistic monads. Note that the 'state action'

$$(W, A) \mapsto W \to (A \times W)$$

has mixed variance in W, suggesting a modification of our $\mathcal{C}/\mathcal{X}$ construction that uses a coend rather than an ordinary colimit.

1.3 Relational Models

The archetypal Cartesian closed category is the category $\mathbf{Set}$ of sets and functions. Most of the work in this thesis has been to do with reader actions (including reader monads as a special case), and we have already provided a classification of the reader actions on **Set** – up to the equivalence relation that identifies actions of monoidal categories $\mathcal{X}$ and $\mathcal{Y}$ if the resulting categories $\mathbf{Set}/\mathcal{X}^{\mathrm{op}}$ and $\mathbf{Set}/\mathcal{Y}^{\mathrm{op}}$ are isomorphic – in Propositions **??** and **??**: namely, they are classified by lax monoidal functors $F\colon \mathbf{Set} \to \mathbf{Set}$, in such a way that if a reader action given by an oplax monoidal functor $j\colon \mathcal{X} \to \mathbf{Set}$ corresponds to the functor $F\colon \mathbf{Set} \to \mathbf{Set}$, then $\mathbf{Set}/\mathcal{X}^{\mathrm{op}}$ is isomorphic to the category $F_*\mathbf{Set}$ formed from **Set** by base change along F. Thus, the theory in the case of **Set** reduces to the theory of **Set**-enriched base change.

There are, however, many more examples in the literature of models of programming languages that ultimately derive from Kleisli categories for monads not of the reader kind. The problem with this, from our point of view, is that the resulting Kleisli categories will not be automatically Cartesian. This means that we need different methods to ensure that we end up with a category suitable for modelling a compositional semantics.

For example, the Kleisli category for the powerset monad on **Set** is the category **Rel** of sets and relations. This is a monoidal closed category, so can be used to model multiplicative linear type theory. By using a linear exponential comonad based on finite multisets, we can transform it into a Cartesian closed category.

An important category derived from the category of sets and relations is the category **Coh** of coherence spaces. A coherence space is a set with some additional structure on it – namely, the structure of an undirected graph – and a morphism between coherence spaces is a relation between sets that respects this structure in a particular way (see [Mel14] for details).

Danos and Ehrhard in [DE11] develop a probabilistic version of coherence spaces, which is still monoidal closed. By using the finite multiset comonad, we can transform it into a Cartesian closed category. A striking result of Ehrhard, Pagani and Tasson [EPT18] is that this category is already fully abstract for a probabilistic variant of PCF.

This suggests an alternative way to get round the Cartesianness hurdle when adding effects: start with a model of the base language that is constructed by applying a linear exponential comonad to a monoidal closed category, and then delay application of the linear exponential comonad until after we have added the effect.

Unfortunately, there is no real guarantee that a linear exponential comonad on the base category will give us a linear exponential comonad on a Kleisli category or one of the other effectful categories we have considered. Some work is required to investigate the conditions in which we can lift an exponential construction in this way, given the functorial results from Chapter **??**. This will pave the way for us to understand the relational and coherence-space models using the ideas developed in this thesis.

1.4 Adequacy Proofs

In Chapters **??** and **??** we proved Computational Adequacy via a factorization and operational techniques. This is good, because it allows us to assume very little about the underlying model, other than that it is itself computationally adequate. Nevertheless, our techniques rely very heavily on the fact that our base language is Idealized Algol. Although we outlined ways to get around this in Section 1.1, it would also be useful to have a conventional logical relations-based proof of Computational Adequacy.

Assume that $\mathcal{G}$ is a Cartesian closed category enriched in directed-complete partial orders (dcpos). If $M\colon \mathcal{G} \to \mathcal{G}$ is a monoidal monad, then the Kleisli category associated to M inherits an order, whereby $\sigma \leq \tau\colon A \to B$ in $\mathrm{Kl}_M\,\mathcal{G}$ if $\sigma \leq \tau\colon A \to MB$ in $\mathcal{G}$. Moreover, since composition in the Kleisli category is given by a composition in the base category, this partial order will be respected by composition, as will be limits of directed sets.

Suppose again that $\mathcal{G}$ is a dcpo-enriched category, and that a monoidal category $\mathcal{X}$ acts on $\mathcal{G}$ via a lax action. Then the construction of Melliès yields an $[\mathcal{X}, \mathbf{Dcpo}]$-enriched category $\mathrm{Mell}_{\mathcal{X}}\,\mathcal{G}$. Since **Dcpo** is cocomplete [Mes77], we can construct the coends we need to define the Day convolution product in $[\mathcal{X}, \mathbf{Dcpo}]$.

If $\mathcal{X}$ is small, or if the required colimits exist for other reasons, then we may take the change of base with respect to the colimit functor $[\mathcal{X}, \mathbf{Dcpo}] \to \mathbf{Dcpo}$ to $\mathrm{Mell}_{\mathcal{X}}\,\mathcal{G}$ to give us a **Dcpo**-enriched category $\mathcal{G}/\mathcal{X}$.

Concretely, the morphisms in $\mathcal{G}/\mathcal{X}$ are the same as usual, while the order structure is given by a transfinite construction as in [Fie96, 2.17]. In particular, if $\sigma, \tau\colon A \to B$ in $\mathcal{G}/\mathcal{X}$ are given by morphisms

$$\tilde{\sigma}\colon A \to X.B \qquad\qquad \tilde{\tau}\colon A \to Y.B$$

in $\mathcal{G}$, and if there are morphisms $h\colon X \to Z$, $k\colon Y \to Z$ in $\mathcal{X}$ such that

$$\sigma; (h.B) \leq \tau; (k.B)\,,$$

then $\sigma \leq \tau$ in $\mathcal{G}/\mathcal{X}$.

In either case, we now define the interpretation of the fixpoint combinator $\mathbf{Y}$ using the properties of this new order, which will be enough to prove adequacy along the lines of Theorem **??**.

It is worth mentioning that although we can prove computational adequacy for our various Kleisli categories (and categories of the form $\mathcal{G}/\mathcal{X}$), it does not automatically follow that we can prove adequacy in the same way for the various quotiented categories. An important example is our model of countable nondeterminism with must testing. It is well known (see, e.g., [AP86, AP81]) that countable nondeterminism is inextricably linked to dicontinuity of composition, which robs us of an important ingredient in the standard order-theoretic proof of adequacy.

The solution is to prove adequacy in the normal way for the semantics of $\mathrm{IA}_{\mathbb{N}}$ in $\mathcal{G}_{\mathbb{N}}$, and then to apply the argument from Corollary **??** to reduce this to a Computational Adequacy result for the semantics of Idealized Algol with countable nondeterminism and must testing. Here, we are using an idea due to Levy [Lev08]: that if we want to prove adequacy for countable nondeterminism, then we need a form of *explicit forcing*, in which we uncover some information about the points at which we have made an infinite nondeterministic branch. In our case, the explicit forcing is done through the language $\mathrm{IA}_{\mathbb{N}}$, which creates a complete record of the nondeterministic values we have chosen along the way. We have then applied the second part of Levy's technique – *hiding* – to remove the explicit information and yield a pure, computationally adequate model of countable nondeterminism.

Bibliography

[AP81] 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.

[AP86] K. R. Apt and G. D. Plotkin. Countable nondeterminism and random assignment. *J. ACM*, 33(4):724–767, August 1986.

[DE11] Vincent Danos and Thomas Ehrhard. Probabilistic coherence spaces as a model of higher-order probabilistic computation. *Information and Computation*, 209(6):966 – 991, 2011.

[EPT18] Thomas Ehrhard, Michele Pagani, and Christine Tasson. Full abstraction for probabilistic pcf. *J. ACM*, 65(4):23:1–23:44, April 2018.

[Fie96] Adrian Fiech. Colimits in the category dcpo. *Mathematical Structures in Computer Science*, 6:455–468, 1996.

[Har99] Russell S. Harmer. Games and full abstraction for nondeterministic languages. Technical report, 1999.

[HO00] 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.

[Lai15] J. Laird. Sequential algorithms for unbounded nondeterminism. *Electronic Notes in Theoretical Computer Science*, 319:271 – 287, 2015.

[Lev01] Paul Blain Levy. Call-by-push-value. Technical report, Queen Mary College, 2001.

[Lev08] Paul Blain Levy. Infinite trace equivalence. *Annals of Pure and Applied Logic*, 151(2):170 – 198, 2008.

[Lev14] Paul Blain Levy. Morphisms between plays. Lecture slides, GaLoP, 2014.

[Mel14] Paul-André Melliès. A survival guide on coherence spaces. https://www.irif.fr/ mellies/mpri/mpri-m2/mpri-mellies-lecture-notes-2.pdf, 2014. Preprint on webpage.

[Mes77] J. Meseguer. On order-complete universal algebra and enriched functorial semantics. In Marek Karpiński, editor, *Fundamentals of Computation Theory*, pages 294–301, Berlin, Heidelberg, 1977. Springer Berlin Heidelberg.

[PP02] Gordon Plotkin and John Power. Notions of computation determine monads. In Mogens Nielsen and Uffe Engberg, editors, *Foundations of Software Science and Computation Structures*, pages 342–356, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.

[TO15] 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.