Linearly-used state in models of call-by-value

Rasmus Ejlers Møgelberg[1] * and Sam Staton[2] **

[1] IT University of Copenhagen, Denmark
[2] Computer Laboratory, University of Cambridge, UK

Abstract. We investigate the phenomenon that *every monad is a linear state monad*. We do this by studying a fully-complete state-passing translation from an impure call-by-value language to a new linear type theory: the enriched call-by-value calculus. The results are not specific to store, but can be applied to any computational effect expressible using algebraic operations, even to effects that are not usually thought of as stateful. There is a bijective correspondence between generic effects in the source language and state access operations in the enriched call-by-value calculus.

From the perspective of categorical models, the enriched call-by-value calculus suggests a refinement of the traditional Kleisli models of effectful call-by-value languages. The new models can be understood as enriched adjunctions.

1 Introduction

Computational effects such as store effects, input/output and control effects are usually associated with the imperative style of programming, and functional programming languages exhibiting such behaviour are thought of as "impure". However, computational effects can be encapsulated within a purely functional language by the use of monads [12]. The central idea behind this is to distinguish between a type of values such as (nat), and a type of computations $T(\mathrm{nat})$ that may return a value of type nat but can also do other things along the way. Imperative behaviour can then be encoded using *generic effects* in the sense of Plotkin and Power [19]. For example, one can add global store by adding a pair of terms, $\mathtt{assign}_l \colon \mathrm{Val} \to T(1)$ and $\mathtt{deref}_l \colon T(\mathrm{Val})$, for each cell l in the store, or one can add nondeterminism by adding a constant $\mathtt{random} \colon T(1+1)$ computing a random boolean. Computational effects that can be described using generic effects are called *algebraic* and these account for a wide range of effects with the notable exception of control effects such as continuations.

It is striking that there is no notion of state in the theory of monads. After all, imperative behaviour is often about changing or branching on the state of the machine. The notion of state is most naturally associated with certain effects

* Research supported by the Danish Agency for Science, Technology and Innovation.
** Research supported by EPSRC Fellowship EP/E042414/1 and ANR Projet CHOCO.

like store effects, but in this paper we shall see that all algebraic effects can be viewed in this way.[3]

The notion of state plays an important role in operational semantics and Hoare logic. Indeed, Plotkin and Power [15] have suggested that configurations, i.e. pairs $\langle M, s\rangle$ consisting of a term M to be evaluated and the current state s of the machine, might form a basis for defining general operational semantics.

In this paper we show how the theory of algebraic effects can be formulated by taking the notion of state as primitive, rather than the notion of monad. On the syntactic side we introduce the *enriched call-by-value calculus* (ECBV). We show that a special state type in ECBV gives rise to a language that is equivalent to a fine-grained monadic call-by-value calculus (FGCBV) [11]. On the semantic side we introduce a notion of enriched call-by-value model which generalises monad models.

Central to our treatment of state is the idea of linear usage: computations cannot copy the state and save it for later, nor can they discard the state and insert a new one instead. This special status of the state was already noted by Strachey [25] and Scott [24] and has been developed by O'Hearn and Reynolds [13].

Linear usage of state can be expressed syntactically using a *linear state passing style*, in which a stateful computation of type $A \to B$ is considered as a linear map of type $!A \otimes \underline{\mathrm{S}} \multimap !B \otimes \underline{\mathrm{S}}$. The type $\underline{\mathrm{S}}$ of states must be used linearly, but A and B can be used arbitrarily. These type constructions form the basis of ECBV, which can be considered as a kind of non-commutative linear logic that is expressive enough to describe the linear usage of state.

Earlier metalanguages for effects, such as the monadic metalanguage [12], call-by-push-value [10], and the enriched effect calculus [2] have an explicit monadic type constructor. There is no monadic type constructor in ECBV: there is a state type $\underline{\mathrm{S}}$ instead. Still, in this fragment one can express all algebraic notions of effects, even the ones that we are not used to thinking of as "state-like", using what we call *state access operations*. For example, the generic effects $\mathtt{assign}_l$, $\mathtt{deref}_l$, and $\mathtt{random}$ correspond to the following state access operations:

$$\mathtt{write}_l \colon !\mathrm{Val} \otimes \underline{\mathrm{S}} \multimap \underline{\mathrm{S}}, \quad \mathtt{read}_l \colon \underline{\mathrm{S}} \multimap !\mathrm{Val} \otimes \underline{\mathrm{S}}, \quad \mathtt{random} \colon \underline{\mathrm{S}} \multimap !(1+1) \otimes \underline{\mathrm{S}}\ . \quad (1)$$

The equivalence of FGCBV and ECBV is proved for extensions of the two calculi along any algebraic effect theory. The generalisation is formulated using a notion of effect theory [16] which captures notions of algebraic effects.

The categorical models of ECBV provide a new general notion of model for call-by-value languages. In brief, an enriched model consists of two categories $\mathbf{V}$ and $\mathbf{C}$ such that $\mathbf{V}$ has products and distributive coproducts, and $\mathbf{C}$ is enriched in $\mathbf{V}$ with copowers and coproducts. The objects of $\mathbf{V}$ interpret ordinary "value" types, and the objects of $\mathbf{C}$ interpret "computation" types (such as the state type $\underline{\mathrm{S}}$) which must be used linearly. This class of models encompasses all Kleisli categories (which have been axiomatised as Freyd categories) and many

[3] For this reason we use the terminology "store effects" for the specific (memory access operations) and reserve "state" for the general notion.

Eilenberg-Moore categories (which provide a natural notion of model for call-by-push-value and the enriched effect calculus).

Our approach to proving the equivalence of FGCBV and ECBV is semantic. We show that the traditional models of effectful call-by-value languages, using monads and Kleisli constructions, form a coreflective subcategory of the models of ECBV. The state-passing translation is the unit of the coreflection. Our main semantic result provides a bijective correspondence between comodel structures on a state object and model structures on the induced linear state monad. This extends Plotkin and Power's correspondence between algebraic operations and generic effects [19] with a third component: state access operations.

The enriched call-by-value calculus is a fragment of the enriched effect calculus (EEC, [2]). In Section 8 we show that EEC is a conservative extension of ECBV. This shows that the linear state monad translation from FGCBV into EEC is fully complete: any term of translated type in the target language corresponds to a unique term in the source language. This result indicates that EEC is a promising calculus for reasoning about linear usage of effects. The related paper [3] shows how the linear-use continuation passing translation arises from a natural dual model construction on models of EEC. In fact, from the point of view of EEC the two translations are surprisingly similar: the linearly used state translation is essentially dual to the linearly used continuations translation.

Acknowledgements. We thank Alex Simpson for help and encouragement. Also thanks to Lars Birkedal, Jeff Egger, Masahito Hasegawa, Shin-ya Katsumata and Paul Levy for helpful discussions.

2 Source calculus: Fine-Grained Call-by-Value

Our source language is a call-by-value language equipped with an equational theory to be thought of as generated by some operational semantics, as in [14]. We use a variant of fine-grained call-by-value [11], because the explicit separation of judgements into value and producer judgements fits well with a similar division in the target language.

We use α to range over type constants. The types are given by the grammar

$$\sigma ::= \alpha \mid 1 \mid \sigma \times \sigma \mid 0 \mid \sigma + \sigma \mid \sigma \multimap \sigma \,.$$

The fine-grained call-by-value calculus (FGCBV) has two typing judgements, one for values and one for producers. These are written $\Gamma \vdash^v V : \sigma$ and $\Gamma \vdash^p M : \sigma$. The latter should be thought of as typing computations which produce values in the type judged but may also perform side-effects along the way. In both judgements the variables of the contexts are to be considered as placeholders for values. The function space $\multimap$ is a call-by-value one, which takes a value and produces a computation. In fact this language is equivalent to Moggi's monadic λ_c: the type construction $(1 \multimap (-))$ is a monad. Typing rules along with equality rules are given in Figure 1.

$$\frac{}{\Gamma, x\colon\sigma, \Gamma' \vdash^v x\colon\sigma} \qquad \frac{}{\Gamma \vdash^v \star\colon 1} \qquad \frac{\Gamma \vdash^v V\colon\sigma_1 \times \sigma_2}{\Gamma \vdash^v \pi_i(V)\colon\sigma_i} \qquad \frac{\Gamma \vdash^v V\colon\sigma_i}{\Gamma \vdash^v \mathtt{in}_i(V)\colon\sigma_1+\sigma_2}$$

$$\frac{\Gamma \vdash^v V_1\colon\sigma_1 \quad \Gamma \vdash^v V_2\colon\sigma_2}{\Gamma \vdash^v \langle V_1, V_2\rangle\colon\sigma_1\times\sigma_2} \qquad \frac{\Gamma \vdash^v V\colon\sigma_1+\sigma_2 \quad \Gamma, x_i\colon\sigma_i \vdash^v W_i\colon\tau \quad (i=1,2)}{\Gamma \vdash^v \mathtt{case}\ V\ \mathtt{of}\ (\mathtt{in}_1(x_1).W_1; \mathtt{in}_2(x_2).W_2)\colon\tau}$$

$$\frac{\Gamma \vdash^v V\colon 0}{\Gamma \vdash^v \mathtt{image}(V)\colon\sigma} \qquad \frac{\Gamma \vdash^v V\colon\sigma}{\Gamma \vdash^p \mathtt{return}\, V\colon\sigma} \qquad \frac{\Gamma \vdash^p M\colon\sigma \quad \Gamma, x\colon\sigma \vdash^p N\colon\tau}{\Gamma \vdash^p M\,\mathtt{to}\,x.\,N\colon\tau}$$

$$\frac{\Gamma, x\colon\sigma \vdash^p N\colon\tau}{\Gamma \vdash^v \lambda x\colon\sigma.\, N\colon\sigma\rightharpoonup\tau} \qquad \frac{\Gamma \vdash^v V\colon\sigma\rightharpoonup\tau \quad \Gamma \vdash^v W\colon\sigma}{\Gamma \vdash^p V\,W\colon\tau}$$

$$M = \star \qquad \pi_i(\langle V_1, V_2\rangle) = V_i \qquad \langle \pi_1(V), \pi_2(V)\rangle = V \qquad \mathtt{image}(V) = W[V/x]$$

$$\mathtt{case}\ \mathtt{in}_i(V)\ \mathtt{of}\ (\mathtt{in}_1(x_1).W_1; \mathtt{in}_2(x_2).W_2) = W_i[V/x_i] \qquad \lambda x\colon\sigma.\, M(V) = M[V/x]$$

$$\mathtt{case}\ V\ \mathtt{of}\ (\mathtt{in}_1(x_1).W[\mathtt{in}_1(x_1)/x]; \mathtt{in}_2(x_2).W[\mathtt{in}_2(x_2)/x]) = W[V/x] \qquad \lambda x\colon\sigma.\,(V\,x) = V$$

$$M\,\mathtt{to}\,x.\,\mathtt{return}\,x = M \quad \mathtt{return}\,V\,\mathtt{to}\,x.\,N = N[V/x]$$

$$(M\,\mathtt{to}\,x.\,N)\,\mathtt{to}\,y.\,P = M\,\mathtt{to}\,x.\,(N\,\mathtt{to}\,y.\,P)$$

Fig. 1. Fine-grained call-by-value. (Equality rules subject to usual conventions.)

We can define derived case constructs on producer terms:

$$\begin{aligned}\mathtt{case}^p\ M\ \mathtt{of}\ (\mathtt{in}_1(x_1).N_1; \mathtt{in}_2(x_2).N_2)&\\ \stackrel{\text{def}}{=}\ & M\,\mathtt{to}\,x.\,(\mathtt{case}\ x\ \mathtt{of}\ (\mathtt{in}_1(x_1).\lambda w\colon 1.\, N_1; \mathtt{in}_2(x_2).\lambda w\colon 1.\, N_2))(\star)\\ \mathtt{image}^p(M)\ \stackrel{\text{def}}{=}\ & M\,\mathtt{to}\,x.\,(\mathtt{image}(x))\,\star\end{aligned}$$

where z, w are fresh variables. These constructions have derived typing rules

$$\frac{\Gamma \vdash^p M\colon\sigma_1+\sigma_2 \quad \Gamma, x_i\colon\sigma_i \vdash^p N_i\colon\tau\ (i=1,2)}{\Gamma \vdash^p \mathtt{case}^p\ M\ \mathtt{of}\ (\mathtt{in}_1(x_1).N_1; \mathtt{in}_2(x_2).N_2)\colon\tau} \qquad \frac{\Gamma \vdash^p M\colon 0}{\Gamma \vdash^p \mathtt{image}^p(M)\colon A} \tag{2}$$

FGCBV is a skeleton on which one can add specific effects. We will make this precise in Section 2.1, but we begin with some examples. In the case of global store, given by some set of cells Loc holding values of some type Val, we add the following *generic effects* [19] to FGCBV: for each cell $l \in \mathrm{Loc}$, we add producer term constants $\mathtt{deref}_l$ and $\mathtt{assign}_l$ with typing judgements $\Gamma \vdash^p \mathtt{deref}_l\colon \mathrm{Val}$ and $\Gamma \vdash^p \mathtt{assign}_l(V)\colon 1$ if $\Gamma \vdash^v V\colon \mathrm{Val}$. We add to the theory of equality in Figure 1 the seven equations for global store proposed by Plotkin and Power [18], for example the two equations

$$\mathtt{deref}_l\,\mathtt{to}\,x.\,\mathtt{assign}_l(x) = \mathtt{return}\,(\star) \tag{3}$$

$$\mathtt{assign}_l(V)\,\mathtt{to}\,x.\,\mathtt{assign}_l(W) = \mathtt{assign}_l(W) \tag{4}$$

which state that reading a cell and then writing the same value is the same as doing nothing, and that the effect of two writes equals that of the second.

In the case of non-determinism, the generic effect is "$\mathtt{random}$" with typing judgement $\Gamma \vdash^p \mathtt{random} : 1 + 1$. The equations are perhaps most easily described using the *algebraic operation* corresponding to $\mathtt{random}$, defined as $M \mathtt{or} N \stackrel{\text{def}}{=} \mathtt{case}^p\ \mathtt{random}\ \mathtt{of}\ (\mathtt{in}_1(x).M; \mathtt{in}_2(x).N)$. The derived typing rule says that $\Gamma \vdash^p M\ \mathtt{or}\ N : \sigma$ if $\Gamma \vdash^p M : \sigma$ and $\Gamma \vdash^p N : \sigma$. There are three equations: associativity, commutativity and idempotency of "$\mathtt{or}$".

2.1 Effect theories

We do not want to allow arbitrary extensions of FGCBV. In this section we define *effect theories*, which are particularly well-behaved extensions that include the examples above, of global store and nondeterminism. Effect theories are important from the semantic point of view because they are a kind of presentation for enriched algebraic theories, as will be clarified in Section 6.

Plotkin and Pretnar have also defined a notion of effect theory [17, §3]. Their effect theories can be accommodated in our model. The main difference is in the presentation: we use generic effects rather than algebraic operations.

By a value signature we shall simply mean a signature for a many-sorted algebraic theory in the usual sense. This means a set of type constants ranged over by α, β, and a set of term constants f with a given arity $f : (\alpha_1, \ldots, \alpha_n) \to \beta$, where the α_i, β range over type constants. We can extend FGCBV along a value signature by adding the type constants and the typing rule

$$\frac{\Gamma \vdash^v t_i : \alpha_i\ (i = 1, \ldots, n)}{\Gamma \vdash^v f(t_1, \ldots, t_n) : \beta} \tag{5}$$

for every term constant $f : (\alpha_1, \ldots, \alpha_n) \to \beta$ in the signature. A value theory is a value signature with a set of equations, i.e. pairs of terms typable in the same context $\Gamma \vdash^v V = W : \beta$, where V, W are formed only using variable introduction and the rule (5).

An *effect signature* consists of a value theory and a set of effect constants each with an assigned arity $e : \bar{\beta}; \bar{\alpha}_1 + \ldots + \bar{\alpha}_n$ consisting of a list of type constants and a formal sum of lists of type constants. FGCBV can be extended along an effect signature by adding, for every $e : \bar{\beta}; \bar{\alpha}_1 + \ldots + \bar{\alpha}_n$ a typing judgement

$$\frac{\Gamma \vdash^v \bar{V} : \bar{\beta}}{\Gamma \vdash^p e(\bar{V}) : \bar{\alpha}_1 + \ldots + \bar{\alpha}_n} \tag{6}$$

The hypothesis is to be understood as a vector of typing judgements, and in the conclusion, the vectors $\bar{\alpha}_i$ should be interpreted as the product of the types in the vector.

For example, the theory for global store has one value type constant Val, and for each location $l \in \mathrm{Loc}$ a pair of effect constants $(\mathtt{deref}_l : 1; \mathrm{Val})$ and $(\mathtt{assign}_l : \mathrm{Val}; 1)$. In this case term constants in the value theory can be used to

add basic operations manipulating values in Val. In the case of nondeterminism, the effect constant `random` has arity $1; 1+1$.

An effect theory comprises an effect signature and a set of equations. The equations are pairs of producer terms-in-context $\Gamma \vdash^p M = N : \bar{\alpha}_1 + \ldots + \bar{\alpha}_n$ of a restricted kind. We impose the following restrictions: firstly, Γ must consist of variables with type constants, i.e., of the form $x : \alpha$. Secondly, the terms M and N must be built from a first order fragment of *effect terms*. This first order fragment consists of the first nine rules of Figure 1, the derived rules for sum types in producer terms (2) and the rules (5) and (6).

We write FGCBV_E for FGCBV augmented with an effect theory E.

3 Target calculus: Enriched call-by-value

The target language for the linear state translation is a new calculus called the *enriched call-by-value calculus* (ECBV), that we now introduce. It is a fragment of the enriched effect calculus (EEC), which was introduced by Egger et al. [2] as a calculus for reasoning about linear usage in computational effects. The types of ECBV can be understood as a fragment of linear logic that is expressive enough to describe the linear state monad, $\underline{\mathsf{S}} \multimap !(-) \otimes \underline{\mathsf{S}}$. We will not dwell on the connection with linear logic here.

The enriched call-by-value calculus has two collections of types: value types and computation types. We use $\alpha, \beta, \ldots$ to range over a set of *value type constants*, and $\underline{\alpha}, \underline{\beta}, \ldots$ to range over a disjoint set of *computation type constants*. We then use $\mathsf{A}, \mathsf{B}, \ldots$ to range over value types, and $\underline{\mathsf{A}}, \underline{\mathsf{B}}, \ldots$ to range over computation types, which are specified by the grammar below:

$$\begin{aligned} \mathsf{A} &::= \alpha \mid 1 \mid \mathsf{A} \times \mathsf{B} \mid 0 \mid \mathsf{A} + \mathsf{B} \mid \underline{\mathsf{A}} \multimap \underline{\mathsf{B}} \\ \underline{\mathsf{A}} &::= \underline{\alpha} \mid \underline{0} \mid \underline{\mathsf{A}} \oplus \underline{\mathsf{B}} \mid !\mathsf{A} \otimes \underline{\mathsf{B}} \ . \end{aligned}$$

Note that the construction $!\mathsf{A} \otimes \underline{\mathsf{B}}$ is indivisible: the strings $!\mathsf{A}$ and $\underline{\mathsf{A}} \otimes \underline{\mathsf{B}}$ are not well-formed types. Note also that unlike EEC [2] there is no inclusion of computation types into value types. Moreover, there are no type constructors corresponding to F or U as known from CBPV [10].

The enriched call-by-value calculus has two basic typing judgements, written

$$\Gamma \mid - \vdash t : \mathsf{B} \quad \text{and} \quad \Gamma \mid z : \underline{\mathsf{A}} \vdash t : \underline{\mathsf{B}} \tag{7}$$

In the first judgement, B is a value type, and in the second judgement, both $\underline{\mathsf{A}}$ and $\underline{\mathsf{B}}$ need to be computation types. The second judgement should be thought of as a judgement of linearity in the variable $z : \underline{\mathsf{A}}$. The typing rules are given in Figure 2. In the figure, Γ is an assignment of value types to variables, and Δ is an assignment of a computation type to a single variable, as in (7). The equality theory includes α, β and η rules and is exactly as for EEC [2, Sec. 3].

We can talk about type isomorphisms in ECBV in the usual way. For value types, an isomorphism $\mathsf{A} \cong \mathsf{B}$ is given by two judgements, $x : \mathsf{A} \mid - \vdash t : \mathsf{B}$ and $y : \mathsf{B} \mid - \vdash u : \mathsf{A}$, such that $u[t/y] = x$, $t[u/x] = y$. For computation types, $\underline{\mathsf{A}} \cong \underline{\mathsf{B}}$

$$\frac{}{\Gamma, x\colon \mathsf{A} \mid - \vdash x\colon \mathsf{A}} \qquad \frac{}{\Gamma \mid z\colon \underline{\mathsf{A}} \vdash z\colon \underline{\mathsf{A}}} \qquad \frac{}{\Gamma \mid - \vdash \star\colon 1}$$

$$\frac{\Gamma \mid - \vdash t\colon \mathsf{A} \quad \Gamma \mid - \vdash u\colon \mathsf{B}}{\Gamma \mid - \vdash \langle t, u\rangle\colon \mathsf{A}\times\mathsf{B}} \qquad \frac{\Gamma \mid - \vdash t\colon \mathsf{A}_1 \times \mathsf{A}_2}{\Gamma \mid - \vdash \pi_i(t)\colon \mathsf{A}_i}$$

$$\frac{\Gamma \mid - \vdash t\colon \mathsf{A}_i}{\Gamma \mid - \vdash \mathrm{in}_i(t)\colon \mathsf{A}_1 + \mathsf{A}_2} \qquad \frac{\Gamma \mid \Delta \vdash t\colon \underline{\mathsf{A}}_i}{\Gamma \mid \Delta \vdash \underline{\mathrm{in}}_i(t)\colon \underline{\mathsf{A}}_1 \oplus \underline{\mathsf{A}}_2}$$

$$\frac{\Gamma \mid - \vdash t\colon 0}{\Gamma \mid - \vdash \mathrm{image}(t)\colon \mathsf{A}} \qquad \frac{\Gamma \mid - \vdash s\colon \mathsf{A}_1 + \mathsf{A}_2 \quad \Gamma, x_i\colon \mathsf{A}_i \mid - \vdash t_i\colon \mathsf{C}\ (i=1,2)}{\Gamma \mid - \vdash \mathrm{case}\, s\, \mathrm{of}\, (\mathrm{in}_1(x_1).\, t_1; \mathrm{in}_2(x_2).\, t_2)\colon \mathsf{C}}$$

$$\frac{\Gamma \mid \Delta \vdash t\colon \underline{0}}{\Gamma \mid \Delta \vdash \underline{\mathrm{image}}(t)\colon \underline{\mathsf{A}}} \qquad \frac{\Gamma \mid \Delta \vdash s\colon \underline{\mathsf{A}}_1 \oplus \underline{\mathsf{A}}_2 \quad \Gamma \mid x_i\colon \underline{\mathsf{A}}_i \vdash t_i\colon \underline{\mathsf{C}}\ (i=1,2)}{\Gamma \mid \Delta \vdash \underline{\mathrm{case}}\, s\, \mathrm{of}\, (\underline{\mathrm{in}}_1(x_1).\, t_1; \underline{\mathrm{in}}_2(x_2).\, t_2)\colon \underline{\mathsf{C}}}$$

$$\frac{\Gamma \mid z\colon \underline{\mathsf{A}} \vdash t\colon \underline{\mathsf{B}}}{\Gamma \mid - \vdash \underline{\lambda} z\colon \underline{\mathsf{A}}.\, t\colon \underline{\mathsf{A}} \multimap \underline{\mathsf{B}}} \qquad \frac{\Gamma \mid - \vdash s\colon \underline{\mathsf{A}} \multimap \underline{\mathsf{B}} \quad \Gamma \mid \Delta \vdash t\colon \underline{\mathsf{A}}}{\Gamma \mid \Delta \vdash s[t]\colon \underline{\mathsf{B}}}$$

$$\frac{\Gamma \mid - \vdash t\colon \mathsf{A} \quad \Gamma \mid \Delta \vdash u\colon \underline{\mathsf{B}}}{\Gamma \mid \Delta \vdash\ !t \otimes u\colon\ !\mathsf{A} \otimes \underline{\mathsf{B}}} \qquad \frac{\Gamma \mid \Delta \vdash s\colon\ !\mathsf{A} \otimes \underline{\mathsf{B}} \quad \Gamma, x\colon \mathsf{A} \mid z\colon \underline{\mathsf{B}} \vdash t\colon \underline{\mathsf{C}}}{\Gamma \mid \Delta \vdash \mathrm{let}\ !x \otimes z\ \mathrm{be}\ s\ \mathrm{in}\ t\colon \underline{\mathsf{C}}}$$

Fig. 2. Typing rules for the enriched call-by-value calculus

is witnessed by closed terms of type $\underline{\mathsf{A}} \multimap \underline{\mathsf{B}}$, $\underline{\mathsf{B}} \multimap \underline{\mathsf{A}}$ composing in both directions to identities. We note the following type isomorphisms, inherited from EEC:

$$\underline{\mathsf{A}} \cong\ !1 \otimes \underline{\mathsf{A}} \qquad\qquad !\mathsf{A} \otimes (!\mathsf{B} \otimes \underline{\mathsf{C}}) \cong\ !(\mathsf{A} \times \mathsf{B}) \otimes \underline{\mathsf{C}} \tag{8}$$

$$\underline{0} \cong\ !0 \otimes \underline{\mathsf{B}} \qquad\qquad (!\mathsf{A} \otimes \underline{\mathsf{C}}) \oplus (!\mathsf{B} \otimes \underline{\mathsf{C}}) \cong\ !(\mathsf{A} + \mathsf{B}) \otimes \underline{\mathsf{C}} \tag{9}$$

Given an effect signature E (Sec. 2.1), we add effects to ECBV as follows. We assume that there is a distinguished computation type constant $\underline{\mathsf{S}}$, called the state type. For each effect constant $e : \bar{\beta}; \bar{\alpha}_1 + \ldots + \bar{\alpha}_n$, we add a closed term, called a *state access operation*:

$$e\colon\ !\bar{\beta} \otimes \underline{\mathsf{S}} \multimap\ !(\bar{\alpha}_1 + \cdots + \bar{\alpha}_n) \otimes \underline{\mathsf{S}} \tag{10}$$

In order to add the equations from an effect theory to ECBV, we need to give interpretations to effect terms. In Section 4 we are going to translate all of FGCBV into ECBV, so we postpone this to there.

We write $\mathrm{ECBV}^{\underline{\mathsf{S}}}_{E}$ for the enriched effect calculus extended over the effect theory E as described above.

Examples 1. The effect theories of global store and of non-determinism will give rise to the state access operations $\mathtt{read}_l$, $\mathtt{write}_l$, and $\mathtt{random}$ in (1).

Note that $\mathtt{read}_l \colon \underline{\mathsf{S}} \multimap !\mathrm{Val} \otimes \underline{\mathsf{S}}$ returns a state that must be used linearly and a result value of the read operation that can be used arbitrarily. One of the equations (3) for global store requires $\mathtt{write}_l(\mathtt{read}_l(s)) = s$; another one (4) says $\mathtt{write}_l(!v \otimes (\mathtt{write}_l(!w \otimes s))) = \mathtt{write}_l(!v \otimes s)$.

4 The state-passing translation

We now describe the state-passing translation from FGCBV to ECBV. We translate FGCBV_E types σ to $\mathrm{ECBV}_E^{\underline{\mathsf{S}}}$ value types σ°:

$$\begin{array}{lll} \alpha^\circ = \alpha & (\sigma \times \tau)^\circ = \sigma^\circ \times \tau^\circ & 1^\circ = 1 \\ (\sigma \rightharpoonup \tau)^\circ = !(\sigma^\circ) \otimes \underline{\mathsf{S}} \multimap !(\tau^\circ) \otimes \underline{\mathsf{S}} & (\sigma + \tau)^\circ = \sigma^\circ + \tau^\circ & 0^\circ = 0 \end{array}$$

The translation takes value type judgements $\Gamma \vdash^v V : \sigma$ to ECBV judgements $\Gamma^\circ \mid - \vdash V^\circ \colon \sigma^\circ$ and it takes producer judgements $\Gamma \vdash^p M : \sigma$ to ECBV judgements $\Gamma^\circ \mid s \colon \underline{\mathsf{S}} \vdash M^\circ \colon !(\sigma^\circ) \otimes \underline{\mathsf{S}}$, as follows.

$$\begin{array}{c} x^\circ = x \qquad \star^\circ = \star \qquad \langle V, W\rangle^\circ = \langle V^\circ, W^\circ\rangle \\ (\pi_i(V))^\circ = \pi_i(V^\circ) \quad (\mathtt{image}(V))^\circ = \mathrm{image}(V^\circ) \quad (\mathtt{in}_i(V))^\circ = \mathrm{in}_i(V^\circ) \\ (\lambda x \colon \sigma.\, N)^\circ = \underline{\lambda} z \colon !\sigma^\circ \otimes \underline{\mathsf{S}}.\ \mathrm{let}\ !x \otimes s\ \mathrm{be}\ z\ \mathrm{in}\ N^\circ \qquad (V\,W)^\circ = V^\circ[!(W^\circ) \otimes s] \\ (M\ \mathtt{to}\ x.\, N)^\circ = \mathrm{let}\ !x \otimes s\ \mathrm{be}\ M^\circ\ \mathrm{in}\ N^\circ \qquad (\mathtt{return}\, V)^\circ = !(V^\circ) \otimes s \\ (\mathtt{case}\ V\ \mathtt{of}\ (\mathtt{in}_1(x_1).W_1; \mathtt{in}_2(x_2).W_2))^\circ = \mathrm{case}\ V^\circ\ \mathrm{of}\ (\mathrm{in}_1(x_1).W_1^\circ; \mathrm{in}_2(x_2).W_2^\circ) \end{array}$$

We translate generic effects to state operations: $(e(\bar{V}))^\circ = e(!\langle V_1^\circ, \ldots, V_m^\circ\rangle \otimes s)$. We are now in a position to add the equations of an effect theory to ECBV. For each equation $\Gamma \vdash^p M = N : \bar{\alpha}_1 + \ldots + \bar{\alpha}_n$ in the effect theory, we add the equation $\Gamma^\circ \mid s \colon \underline{\mathsf{S}} \vdash M^\circ = N^\circ \colon !(\bar{\alpha}_1^\circ + \ldots + \bar{\alpha}_n^\circ) \otimes \underline{\mathsf{S}}$ to $\mathrm{ECBV}_E^{\underline{\mathsf{S}}}$.

Theorem 2 (Soundness). *If $V = W$ then $V^\circ = W^\circ$; if $M = N$ then $M^\circ = N^\circ$.*

Theorem 3 (Fullness on types). *Let A be a value type of ECBV formed using no other computation type constants than $\underline{\mathsf{S}}$. Then there exists a FGCBV type σ such that $\sigma^\circ \cong \mathsf{A}$.*

Proof. By induction on the structure of types. The interesting case $\underline{\mathsf{A}} \multimap \underline{\mathsf{B}}$ uses the fact that any computation type not using any $\underline{\alpha}$ other than $\underline{\mathsf{S}}$ is isomorphic to one of the form $!\mathsf{C} \otimes \underline{\mathsf{S}}$, which follows from the isomorphisms (8) – (9). □

We now state our main syntactic result.

Theorem 4 (Full completeness). *Suppose $\Gamma \vdash^v V, W : \sigma$ and $\Gamma \vdash^p M, N : \sigma$.*

1. *If $V^\circ = W^\circ$ then $V = W$. If $M^\circ = N^\circ$ then $M = N$.*
2. *For any $\Gamma^\circ \mid - \vdash t \colon \sigma^\circ$ there exists a term $\Gamma \vdash^v V : \sigma$ such that $t = V^\circ$.*
3. *For any $\Gamma^\circ \mid s \colon \underline{\mathsf{S}} \vdash t \colon !(\sigma^\circ) \otimes \underline{\mathsf{S}}$ there exists $\Gamma \vdash^p M : \sigma$ such that $t = M^\circ$.*

Theorem 4 can be proved syntactically as follows. Consider first the fragment of ECBV with no other computation type constants than $\underline{\mathsf{S}}$, and only the value type constants of FGCBV_E. This fragment is equivalent to a variant of ECBV where the only computation types are the ones of the form $!\mathsf{A} \otimes \underline{\mathsf{S}}$ with corresponding variants of the typing rules for $!\mathsf{A} \otimes \underline{\mathsf{B}}$. The translation $(-)^\circ$ gives a bijection from FGCBV_E types to value types of $\mathrm{ECBV}_E^{\underline{\mathsf{S}}}$, and one can define an inverse to this translation. Further type constants can be added to $\mathrm{ECBV}_E^{\underline{\mathsf{S}}}$ without changing the result; this can be proved via a normalization theorem for $\mathrm{ECBV}_E^{\underline{\mathsf{S}}}$ which follows the one for EEC (to appear in [4]).

In Section 7.1 we sketch a semantic proof of Theorems 3 and 4.

5 Categorical models

By studying categorical models, we are able to give a canonical, universal status to the two calculi that we have considered so far, and also to the state-passing translation. In Section 7, the full completeness of the state-passing translation will be explained as an equivalence of free categories.

5.1 Monad models of the fine-grained call-by-value calculus

Terminology. Recall that a distributive category is a category with finite products and coproducts, such that the canonical morphisms $((A \times B) + (A \times C)) \to (A \times (B + C))$ and $0 \to A \times 0$ are isomorphisms.

Definition 5. *A* monad model of FGCBV *(or simply a* monad model*) is a distributive category* $\mathbf{V}$ *with a strong monad* T *and Kleisli-exponentials (that is, exponentials of the form* $(A \to T(B))$*).*

A semantics for FGCBV is given in a monad-model in a standard way. For instance, $[\![\sigma \rightharpoonup \tau]\!] = ([\![\sigma]\!] \to T([\![\tau]\!]))$. A value type judgement $\Gamma \vdash^v V : \sigma$ is taken to a morphism $[\![\Gamma]\!] \to [\![\sigma]\!]$, and a producer type judgement $\Gamma \vdash^p M : \sigma$ is taken to a morphism $[\![\Gamma]\!] \to T([\![\sigma]\!])$. This defines a sound and complete notion of model for FGCBV (e.g. [11]). In particular, the types and terms of FGCBV form a syntactic model, which is initial (with respect to an appropriate notion of morphism).

5.2 Enriched call-by-value models

The categorical notion of model for ECBV involves basic concepts from enriched category theory [9]. Let us recall some rudiments. Following [7, 6], we begin with actions of categories. Let $\mathbf{V}$ be a category with finite products. Recall that an action of $\mathbf{V}$ on a category $\mathbf{C}$ is a functor $\cdot : \mathbf{V} \times \mathbf{C} \to \mathbf{C}$ together with coherent natural unit and associativity isomorphisms, $(1 \cdot \underline{A}) \cong \underline{A}$ and $((A \times B) \cdot \underline{C}) \cong (A \cdot (B \cdot \underline{C}))$. (We underline objects of $\mathbf{C}$ to distinguish them from objects of $\mathbf{V}$.) An *enrichment of a category* $\mathbf{C}$ *in* $\mathbf{V}$ *with copowers* is

determined by an action of $\mathbf{V}$ on $\mathbf{C}$ such that each functor $(- \cdot \underline{A})\colon \mathbf{V} \to \mathbf{C}$ has a right adjoint, $\mathbf{C}(\underline{A}, -)\colon \mathbf{C} \to \mathbf{V}$. Then $A \cdot \underline{B}$ is called a copower, and $\mathbf{C}(\underline{A}, \underline{B})$ is called enrichment. Recall also that a *power* is a right adjoint to $(A \cdot -)\colon \mathbf{C} \to \mathbf{C}$ (we will need this in Section 6).

If $\mathbf{C}$ is enriched in $\mathbf{V}$ with copowers, and $\mathbf{C}$ has finite coproducts, then the coproducts in $\mathbf{C}$ are *enriched* if each functor $(A \cdot -)\colon \mathbf{C} \to \mathbf{C}$ preserves them.

Definition 6. *An* enriched call-by-value model *(or simply* enriched model*) is given by a distributive category* $\mathbf{V}$ *and a category* $\mathbf{C}$ *enriched in* $\mathbf{V}$ *with copowers and enriched finite coproducts. A model of* $\mathrm{ECBV}^{\underline{\mathsf{S}}}$ *is given by an enriched call-by-value model together with a chosen object* $\underline{S}$ *of* $\mathbf{C}$*.*

A semantics for ECBV in an enriched model is given similarly to the semantics of EEC [2]. For each value type A, an object $[\![\mathsf{A}]\!]$ of $\mathbf{V}$ is given, and for each computation type $\underline{\mathsf{A}}$, an object $[\![\underline{\mathsf{A}}]\!]$ of $\mathbf{C}$ is given. The product and sum types are interpreted as products and coproducts in $\mathbf{V}$ and $\mathbf{C}$. We let $[\![!\mathsf{A} \otimes \underline{\mathsf{B}}]\!] = ([\![\mathsf{A}]\!] \cdot [\![\underline{\mathsf{B}}]\!])$, and $[\![\underline{\mathsf{A}} \multimap \underline{\mathsf{B}}]\!] = \mathbf{C}([\![\underline{\mathsf{A}}]\!], [\![\underline{\mathsf{B}}]\!])$. The specified object $\underline{S}$ in an $\mathrm{ECBV}^{\underline{\mathsf{S}}}$ model interprets $\underline{\mathsf{S}}$. A judgement $\Gamma \mid - \vdash t\colon \mathsf{A}$ is interpreted as a morphism $[\![\Gamma]\!] \to [\![\mathsf{A}]\!]$ in $\mathbf{V}$, and a judgement $\Gamma \mid \Delta \vdash t\colon \underline{\mathsf{A}}$ is interpreted as a morphism $[\![\Gamma]\!] \cdot [\![\Delta]\!] \to [\![\underline{\mathsf{A}}]\!]$ in $\mathbf{C}$. The types and terms of ECBV form a syntactic model which is initial with respect to an appropriate notion of morphism.

5.3 From enriched models to monad models and back

Given a monad model $(\mathbf{V}, T)$ there is a monoidal action $\mathbf{V} \times \mathbf{Kl}(T) \to \mathbf{Kl}(T)$ on the Kleisli category defined on objects as the product functor and defined on morphisms using the strength of T. The Kleisli category $\mathbf{Kl}(T)$ is $\mathbf{V}$-enriched because $\mathbf{V}$ has Kleisli exponentials.

Proposition 7. *If* $(\mathbf{V}, T)$ *is a monad model (in the sense of Definition 5) then* $(\mathbf{V}, \mathbf{Kl}(T), 1)$ *is an* $\mathrm{ECBV}^{\underline{\mathsf{S}}}$ *model (in the sense of Definition 6).*

On the other hand, if $(\mathbf{V}, \mathbf{C})$ is an enriched model, we will say that an adjunction $F \dashv U\colon \mathbf{C} \to \mathbf{V}$ is *enriched* if there is a natural coherent isomorphism $F(A \times B) \cong A \cdot F(B)$. When $\mathbf{V}$ is cartesian closed, this is equivalent to the usual definition, i.e. a natural isomorphism $\mathbf{C}(F(-), =) \cong \mathbf{V}(-, U(=))$ (see e.g. [8]).

The choice of $\underline{S}$ in $\mathrm{ECBV}^{\underline{\mathsf{S}}}$ models gives an enriched adjunction, since $(- \cdot \underline{S})$ is left adjoint to $\mathbf{C}(\underline{S}, -)\colon \mathbf{C} \to \mathbf{V}$. The following proposition (first noted noted for EEC [3], though it does not appear explicitly there) shows that every enriched adjunction arises in this way:

Proposition 8 ([3]). *Let* $(\mathbf{V}, \mathbf{C})$ *be an enriched model. If* $F \dashv U\colon \mathbf{C} \to \mathbf{V}$ *is an enriched adjunction then it is naturally isomorphic to the enriched adjunction induced by* $F(1)$*.*

So we can equivalently consider $\mathrm{ECBV}^{\underline{\mathsf{S}}}$ models as enriched adjunctions.

Given an enriched adjunction, the corresponding monad gives a monad model. In particular:

Proposition 9. *If* $(\mathbf{V}, \mathbf{C}, \underline{S})$ *is an* ECBV$^{\underline{\mathrm{S}}}$ *model then* $(\mathbf{V}, \mathbf{C}(\underline{S}, - \cdot \underline{S}))$ *is a monad model.*

If we start with a monad model, take the corresponding ECBV$^{\underline{\mathrm{S}}}$ model (via Prop. 7) and then go back (via Prop. 9), we get a monad model that is equivalent to the one that we started with. This is simply because $\mathbf{Kl}(T)(1, A \times 1) \cong T(A)$. We have the slogan: *Every monad is a linear state monad.*

We shall prove later that this connection between monad models and enriched models can be understood as a coreflection.

5.4 Remark: Closed Freyd categories

Closed Freyd categories [20] are an alternative way of presenting monad models. Freyd categories are usually defined using premonoidal categories [23], but we will use the following equivalent definition using actions (following [10, App. B]). A distributive closed Freyd category [22] can be described as an enriched model $(\mathbf{V}, \mathbf{C})$ together with an identity on objects functor $J \colon \mathbf{V} \to \mathbf{C}$ that preserves the action (i.e. $J(\mathsf{A} \times \mathsf{B}) = \mathsf{A} \cdot J(\mathsf{B})$).

If $(\mathbf{V}, T)$ is a monad model the inclusion $\mathbf{V} \to \mathbf{Kl}(T)$ is a distributive closed Freyd category, and every distributive closed Freyd category arises in this way. By removing the requirement that $\mathbf{V}$ and $\mathbf{C}$ have the same objects, we discover the more general class of enriched models.

6 Models and comodels of effect theories

We define what it means for monad models and enriched models to model an effect theory (in the sense of Sec. 2.1).

Models of value theories. Let $\mathbf{V}$ be a distributive category. An interpretation of a value signature in $\mathbf{V}$ is given by interpretations of the type constants α as objects $[\![\alpha]\!]$ of $\mathbf{V}$, and interpretations of term constants $f \colon \bar{\alpha} \to \beta$ as morphisms $[\![f]\!] \colon [\![\bar{\alpha}]\!] \to [\![\beta]\!]$. This is extended to interpret a term in context $\Gamma \vdash^v V : \beta$ as a morphism $[\![V]\!] \colon [\![\Gamma]\!] \to [\![\beta]\!]$. An interpretation of a value theory is an interpretation of the signature such that $[\![V]\!] = [\![W]\!]$ for each equation $\Gamma \vdash^v V = W : \beta$.

Interpreting effect theories in monad models. An interpretation of an effect theory E in a monad model $(\mathbf{V}, T)$ is an interpretation of the value theory in $\mathbf{V}$ and an interpretation of each effect constant $e \colon \bar{\beta}; \bar{\alpha}_1 + \cdots + \bar{\alpha}_n$ in E as a Kleisli map $[\![e]\!] \colon [\![\bar{\beta}]\!] \to T([\![\bar{\alpha}_1]\!] + \cdots + [\![\bar{\alpha}_n]\!])$, satisfying the equations of the theory.

Effects and enriched models. In enriched models, according to (10), every effect constant should be interpreted as a morphism $[\![e]\!] \colon [\![\bar{\beta}]\!] \cdot \underline{S} \to ([\![\bar{\alpha}_1]\!] + \cdots + [\![\bar{\alpha}_n]\!]) \cdot \underline{S}$ in $\mathbf{C}$. We can relate these to the Kleisli maps of the monad model, extending the bijective correspondence between algebraic operations and generic effects [19]:

Proposition 10. *Let* $(\mathbf{V}, \mathbf{C})$ *be an enriched model and consider* $\underline{S}$ *in* $\mathbf{C}$*. The following sets are in natural bijection:*

1. State access operations*: morphisms* $A \cdot \underline{S} \to B \cdot \underline{S}$ *in* $\mathbf{C}$.
2. Generic effects *for the induced monad: morphisms* $A \to \mathbf{C}(\underline{S}, B \cdot \underline{S})$ *in* $\mathbf{V}$.
3. Algebraic operations *in* $\mathbf{C}$*: families of morphisms* $\mathbf{C}(\underline{S}, \underline{X})^B \to \mathbf{C}(\underline{S}, \underline{X})^A$ *natural in* $\underline{X}$ *in* $\mathbf{C}$.

(Although we do not assume that $\mathbf{V}$ is cartesian closed, the exponentials mentioned in Item 3 always exist.)

To explain the status of the special object $\underline{S}$ we provide a general notion of model for effect theories.

Interpretations of effect theories in general. For a moment, let $\mathbf{V}$ be a distributive category, and let $\mathbf{A}$ be a category enriched in $\mathbf{V}$ with *powers*. Consider an effect signature E and an interpretation of the value theory in $\mathbf{V}$. A *model* of E in $\mathbf{A}$ consists of an object A of $\mathbf{A}$ together with, for each effect constant $e\colon \bar{\beta}; \bar{\alpha}_1 + \cdots + \bar{\alpha}_n$ in E, a morphism $[\![e]\!]\colon A^{[\![\bar{\alpha}_1]\!]+\cdots+[\![\bar{\alpha}_n]\!]} \to A^{[\![\bar{\beta}]\!]}$ in $\mathbf{A}$.

To describe when a model satisfies equations in an effect theory, we need to give a semantics to effect terms. (Recall that an effect term is a first order term of FGCBV.) In any model A of an effect signature, we interpret an effect term typing judgement $\Gamma \vdash^p M : \tau$ as a morphism $[\![M]\!]\colon A^{[\![\tau]\!]} \to A^{[\![\Gamma]\!]}$ in $\mathbf{A}$, by induction on the structure of typing derivations. For instance, consider the $\mathtt{case}^p$ rule in (2). Given interpretations $[\![M]\!]\colon A^{[\![\sigma_1+\sigma_2]\!]} \to A^{[\![\Gamma]\!]}$ and $[\![N_i]\!]\colon A^{[\![\tau]\!]} \to A^{[\![\Gamma,\sigma_i]\!]}$ $(i = 1, 2)$, we define $[\![\mathtt{case}^p\ M\ \mathtt{of}\ (\mathtt{in}_1(x_1).N_1; \mathtt{in}_2(x_2).N_2)]\!]$ to be the composite

$$A^{[\![\tau]\!]} \xrightarrow{([\![N_1]\!],[\![N_2]\!])} A^{[\![\Gamma,\sigma_1]\!]} \times A^{[\![\Gamma,\sigma_2]\!]} \cong A^{([\![\sigma_1+\sigma_2]\!])\times[\![\Gamma]\!]} \xrightarrow{[\![M]\!]^{[\![\Gamma]\!]}} A^{[\![\Gamma]\!]\times[\![\Gamma]\!]} \xrightarrow{A^{\Delta}} A^{[\![\Gamma]\!]}.$$

As another example, $[\![\mathtt{return}\, V]\!] = A^{[\![V]\!]}$. A model of an effect theory in $\mathbf{A}$ is a model of the effect signature such that every effect equation $\Gamma \vdash M = N : \tau$ in the theory is satisfied, i.e. $[\![M]\!] = [\![N]\!]$.

In an enriched model $(\mathbf{V}, \mathbf{C})$, we have a category $\mathbf{C}$ enriched in $\mathbf{V}$ with *co*powers. This means that $\mathbf{C}^{\mathrm{op}}$ is enriched in $\mathbf{V}$ with powers. A *comodel* in $\mathbf{C}$ is a model in $\mathbf{C}^{\mathrm{op}}$.

Definition 11. *An* $\mathrm{ECBV}^{\mathrm{S}}_{E}$ model *is an* $\mathrm{ECBV}^{\mathrm{S}}$ *model with a given* E*-comodel structure on* $\underline{S}$.

Proposition 12. *Let* $(\mathbf{V}, \mathbf{C})$ *be an enriched model and consider* $\underline{S}$ *in* $\mathbf{C}$. *The following data are equivalent.*

1. *An* E*-comodel structure for the object* $\underline{S}$.
2. *An* E*-model structure for the induced monad model.*
3. *For each effect constant* $e\colon \bar{\beta}; \bar{\alpha}_1 + \cdots + \bar{\alpha}_n$ *a family of morphisms*

$$\textstyle\prod_i \mathbf{C}(\underline{S}, \underline{X})^{[\![\bar{\alpha}_i]\!]} \to \mathbf{C}(\underline{S}, \underline{X})^{[\![\bar{\beta}]\!]} \qquad \text{natural in } \underline{X}$$

equipping each $\mathbf{C}(\underline{S}, \underline{X})$ *with the structure of a model of* E.

Example 13. Let $\mathrm{Val}, \mathrm{Loc}$ be sets of values and locations respectively, and let $\underline{S}$ be the set of functions $(\mathrm{Loc} \to \mathrm{Val})$. The category $\mathbf{Set}$ is enriched in itself with copowers given by products, and indeed $\underline{S}$ is a comodel for the theory for global store in the enriched model $(\mathbf{Set}, \mathbf{Set})$. The induced monad on $\mathbf{Set}$ is $((-) \times \underline{S})^{\underline{S}}$. Power and Shkaravska [21] showed that $\underline{S}$ is the final comodel of global store.

7 Categories of models and full completeness

We sketch how the constructions of Propositions 7 and 9 extend to define adjoint 2-functors between a 2-category of monad models and a 2-category of enriched models. We sketch how to use these results to prove full completeness of the linear state monad translation.

Let **ENR** be the 2-category whose objects are $\text{ECBV}^{\underline{\text{S}}}$ models $(\mathbf{V}, \mathbf{C}, \underline{S})$. A 1-cell $(\mathbf{V}, \mathbf{C}, \underline{S}) \to (\mathbf{V}', \mathbf{C}', \underline{S}')$ is a pair of functors $F\colon \mathbf{V} \to \mathbf{V}'$, $G\colon \mathbf{C} \to \mathbf{C}'$ together with an isomorphism $G\underline{S} \cong \underline{S}'$ and a natural isomorphism $G(A \cdot \underline{B}) \cong (F(A)) \cdot (G(\underline{B}))$ whose mate is an isomorphism $F(\mathbf{C}(\underline{B}, \underline{C})) \cong \mathbf{C}'(G\underline{B}, G\underline{C})$, and such that F preserves products and coproducts and G preserves coproducts (up to isomorphism). The 2-cells are natural coherent isomorphisms.

Let **MND** be the 2-category whose objects are monad models $(\mathbf{V}, T)$. A 1-cell $(\mathbf{V}, T) \to (\mathbf{V}', T')$ is a functor $F\colon \mathbf{V} \to \mathbf{V}'$ together with a natural isomorphism $\phi\colon T'F \cong FT$ making (F, ϕ) a monad morphism [26], and such that F preserves products and coproducts, strengths and Kleisli exponentials. The 2-cells are natural coherent isomorphisms.

These definitions can be extended to 2-categories $\mathbf{ENR}_E$, $\mathbf{MND}_E$ whose objects are models of effect theories in the sense of Section 6 and whose 1-cells are required to preserve the interpretations of the theories.

Theorem 14. *The constructions of Propositions 7 and 9 extend to a 2-adjunction whose unit is an isomorphism:* $\mathbf{Kleisli} \dashv \mathbf{StateMnd}\colon \mathbf{ENR}_E \to \mathbf{MND}_E$.

The 2-adjunction is a restriction of a well known 2-adjunction between the category of monads and the category of adjunctions.

7.1 Full completeness

We now provide a semantic argument to explain Theorems 3 and 4. Since the 2-functor $\mathbf{Kleisli}\colon \mathbf{MND}_E \to \mathbf{ENR}_E$ is a left adjoint, it preserves free constructions up to equivalence. In particular it takes the syntactic monad model $(\mathcal{V}_{\text{FGCBV}}, 1 \multimap (-))$, built from the syntax of FGCBV, to the syntactic enriched model $(\mathcal{V}_{\text{ECBV}}, \mathcal{C}_{\text{ECBV}})$, built from the syntax of $\text{ECBV}^{\underline{\text{S}}}$ with exactly one computation type constant, $\underline{\text{S}}$. In consequence, the morphism of monad models that describes the state-passing translation of Section 4,

$$(\mathcal{V}_{\text{FGCBV}}, 1 \multimap (-)) \longrightarrow (\mathcal{V}_{\text{ECBV}}, \underline{\text{S}} \multimap (!(-) \otimes \underline{\text{S}}))$$

is equivalent to the unit of the 2-adjunction $\mathbf{Kleisli} \dashv \mathbf{StateMnd}$, and thus it is an equivalence of categories. In other words, it is essentially surjective and full and faithful, providing a categorical proof of Theorems 3 and 4 respectively (under the assumption that ECBV has exactly one computation type constant).

8 The Enriched Effect Calculus

The enriched effect calculus (EEC) of Egger et al. [2] extends the enriched call-by-value calculus that we introduced in Section 3 with some type constructions:

$$\begin{aligned} \mathsf{A} &::= \ldots \mid \mathsf{A} \to \mathsf{B} \mid \underline{\alpha} \mid \underline{0} \mid \underline{\mathsf{A}} \oplus \underline{\mathsf{B}} \mid !\mathsf{A} \otimes \underline{\mathsf{B}} \mid !\mathsf{A} \\ \underline{\mathsf{A}} &::= \cdots \mid 1 \mid \underline{\mathsf{A}} \times \underline{\mathsf{B}} \mid \mathsf{A} \to \underline{\mathsf{B}} \mid !\mathsf{A} \ . \end{aligned}$$

The additional types have been used to describe other aspects of effectful computation, such as the traditional monadic call-by-name and call-by-value interpretations, and continuation-passing. The additional types of EEC do not affect the full completeness of the linear state-passing translation (Thm. 4), for the following reason. In Proposition 16 we show that every model of ECBV embeds in a model of EEC; conservativity of EEC over ECBV then follows from a strong normalisation result for EEC [4]. Thus the linear state-passing translation of Section 4 can be understood as a fully complete translation into EEC.

Definition 15 ([2]). *A model of EEC* $(\mathbf{V}, \mathbf{C}, F, U)$ *is given by a cartesian closed category* $\mathbf{V}$ *with coproducts, a* $\mathbf{V}$*-enriched category* $\mathbf{C}$ *with products and coproducts and powers and copowers, and an enriched adjunction* $F \dashv U \colon \mathbf{C} \to \mathbf{V}$*.*

We refer to [2] for the term calculus and interpretation of EEC into EEC models. Here, for brevity, we work directly with models.

Clearly every EEC model is an enriched model of ECBV in the sense of Definition 6. Conversely:

Proposition 16. *Every enriched model of ECBV embeds in an EEC model.*

Proof (sketch). Consider an enriched model $(\mathbf{V}, \mathbf{C})$ (in the sense of Def. 6).

For any category $\mathbf{A}$ with finite coproducts, let $\mathrm{FP}(\mathbf{A}^{\mathrm{op}}, \mathbf{Set})$ be the category of finite product preserving functors $\mathbf{A}^{\mathrm{op}} \to \mathbf{Set}$ and natural transformations between them. This category is the cocompletion of $\mathbf{A}$ as a category with finite coproducts (e.g. [5], [23], [9, Thms 5.86, 6.11]).

We will show that $(\mathrm{FP}(\mathbf{V}^{\mathrm{op}}, \mathbf{Set}), \mathrm{FP}(\mathbf{C}^{\mathrm{op}}, \mathbf{Set}))$ is an EEC model, and that $(\mathbf{V}, \mathbf{C})$ embeds in it as an enriched model. For general reasons, $\mathrm{FP}(\mathbf{V}^{\mathrm{op}}, \mathbf{Set})$ and $\mathrm{FP}(\mathbf{C}^{\mathrm{op}}, \mathbf{Set})$ have products and coproducts, and the Yoneda embeddings $(\mathbf{V} \hookrightarrow \mathrm{FP}(\mathbf{V}^{\mathrm{op}}, \mathbf{Set}), \mathbf{C} \hookrightarrow \mathrm{FP}(\mathbf{C}^{\mathrm{op}}, \mathbf{Set}))$ preserve them. Since $\mathbf{V}$ is distributive, $\mathrm{FP}(\mathbf{V}^{\mathrm{op}}, \mathbf{Set})$ is cartesian closed (see [5]).

We now show that $\mathrm{FP}(\mathbf{C}^{\mathrm{op}}, \mathbf{Set})$ is enriched in $\mathrm{FP}(\mathbf{V}^{\mathrm{op}}, \mathbf{Set})$ with powers and copowers. Recall the construction of Day [1], which induces a monoidal biclosed structure on $\hat{\mathbf{A}}$ $(= [\mathbf{A}^{\mathrm{op}}, \mathbf{Set}])$ for every monoidal structure on any category $\mathbf{A}$. We develop this in two ways. First, the monoidal action of $\mathbf{V}$ on $\mathbf{C}$ induces a monoidal action $\hat{\mathbf{V}} \times \hat{\mathbf{C}} \to \hat{\mathbf{C}}$ which has right adjoints in both arguments. Secondly, the monoidal action of $\hat{\mathbf{V}}$ on $\hat{\mathbf{C}}$ restricts to a monoidal action $\mathrm{FP}(\mathbf{V}^{\mathrm{op}}, \mathbf{Set}) \times \mathrm{FP}(\mathbf{C}^{\mathrm{op}}, \mathbf{Set}) \to \mathrm{FP}(\mathbf{C}^{\mathrm{op}}, \mathbf{Set})$ and the right adjoints restrict too. (This second observation relies on the fact that $\mathbf{V}$ is considered with the cartesian monoidal structure.) Thus $\mathrm{FP}(\mathbf{C}^{\mathrm{op}}, \mathbf{Set})$ is enriched in $\mathrm{FP}(\mathbf{V}^{\mathrm{op}}, \mathbf{Set})$ with copowers and powers.

Finally, the enriched adjunction $F \dashv U \colon \mathrm{FP}(\mathbf{C}^{\mathrm{op}}, \mathbf{Set}) \to \mathrm{FP}(\mathbf{V}^{\mathrm{op}}, \mathbf{Set})$ can be induced by any choice of object of $\mathrm{FP}(\mathbf{C}^{\mathrm{op}}, \mathbf{Set})$, by Proposition 8. □

The construction described in this proof is inspired by the following situation. Let $\mathbf{Set}_{\mathrm{f}}$ be the category of finite sets, and let $\mathbb{T}$ be a Lawvere theory. Then $(\mathbf{Set}_{\mathrm{f}}, \mathbb{T}^{\mathrm{op}})$ is almost an enriched model, except that the category $\mathbb{T}^{\mathrm{op}}$ is not $\mathbf{Set}_{\mathrm{f}}$-enriched in general. Our construction, applied to $(\mathbf{Set}_{\mathrm{f}}, \mathbb{T}^{\mathrm{op}})$, yields the basic motivating example of an EEC model: $\mathrm{FP}(\mathbf{Set}_{\mathrm{f}}^{\mathrm{op}}, \mathbf{Set})$ is the category of sets (since $\mathbf{Set}_{\mathrm{f}}$ is the free category with finite coproducts on one generator) and $\mathrm{FP}(\mathbb{T}, \mathbf{Set})$ is the category of algebras of the theory.

References

1. Day, B.: On closed categories of functors. In: LNM 137. Springer (1970)
2. Egger, J., Møgelberg, R., Simpson, A.: Enriching an effect calculus with linear types. In: CSL'09. Springer (2009)
3. Egger, J., Møgelberg, R., Simpson, A.: Linearly-used continuations in the enriched effect calculus. In: FOSSACS'10. Springer (2010)
4. Egger, J., Møgelberg, R., Simpson, A.: The enriched effect calculus (2011), in preparation
5. Fiore, M.P.: Enrichment and representation theorems for categories of domains and continuous functions (March 1996), unpublished manuscript
6. Gordon, R., Power, A.: Enrichment through variation. J. Pure Appl. Algebra 120, 167–185 (1997)
7. Janelidze, G., Kelly, G.: A note on actions of a monoidal category. Theory Appl. of Categ. 9(4), 61–91 (2001)
8. Kelly, G.M.: Adjunction for enriched categories. In: LNM 106. Springer (1969)
9. Kelly, G.M.: Basic Concepts of Enriched Category Theory. CUP (1982)
10. Levy, P.B.: Call By Push Value. Kluwer (Dec 2003)
11. Levy, P., Power, J., Thielecke, H.: Modelling environments in call-by-value programming languages. Inform. and Comput. 185 (2003)
12. Moggi, E.: Notions of computation and monads. Inform. and Comput. 93 (1991)
13. O'Hearn, P.W., Reynolds, J.C.: From Algol to polymorphic linear lambda-calculus. J. ACM 47 (2000)
14. Plotkin, G.: Call-by-name, call-by-value, and the λ-calculus. Theoret. Comp. Sci. 1, 125–159 (1975)
15. Plotkin, G., Power, J.: Tensors of comodels and models for operational semantics. Electr. Notes Theor. Comput. Sci 218, 295–311 (2008)
16. Plotkin, G., Pretnar, M.: A logic for algebraic effects. In: LICS'08. IEEE Press (2008)
17. Plotkin, G., Pretnar, M.: Handlers of algebraic effects. In: ESOP'09. Springer (2009)
18. Plotkin, G.D., Power, J.: Notions of computation determine monads. In: FOSSACS'02 (2002)
19. Plotkin, G.D., Power, J.: Algebraic operations and generic effects. Appl. Categ. Structures 11(1) (2003)
20. Power, Thielecke: Closed Freyd- and κ-categories. In: ICALP'99. Springer (1999)
21. Power, A.J., Shkaravska, O.: From comodels to coalgebras: State and arrays. Electr. Notes Theor. Comput. Sci 106, 297–314 (2004)
22. Power, J.: Generic models for computational effects. Theoret. Comput. Sci. 364(2), 254–269 (2006)
23. Power, J., Robinson, E.: Premonoidal categories and notions of computation. Math. Structures Comput. Sci. 7(5), 453–468 (1997)
24. Scott, D.: Mathematical concepts in programming language semantics. In: Proceedings of the Spring Joint Computer Conference (1972)
25. Strachey, C.: The varieties of programming language. In: Proc. International Computing Symposium (1972)
26. Street, R.: The formal theory of monads. J. Pure Appl. Algebra 2(2), 149–168 (1972)