Conclusion

To define the semantics of pairing in Moggi's framework, a product operation $\bar{\times} : MX \times MY \to M(X \times Y)$ for the computational strong monad M is needed. After having studied the algebraic properties of the two standard products '$\overrightarrow{\times}$' and '$\overleftarrow{\times}$', which correspond to left-to-right and right-to-left evaluation, we have looked for possible alternative products. In the special case, where M is given by the free construction for a decent non-deterministic theory, we found two such alternative products: the strict product '$\check{\times}$' and the parallel product '$\hat{\times}$'.

Acknowledgments

This paper was written during my one-year visit at Imperial College, London. The visit was made possible by a grant of the Deutsche Forschungsgemeinschaft. I have to thank all the people of the TFM section of Imperial College for their stimulating environment, and for giving me the opportunity to participate in their Section Workshop.

References

[1] C.A. Gunter. Relating total and partial correctness interpretations of non-deterministic programs. In P. Hudak, editor, *Principles of Programming Languages (POPL '90)*, pages 306–319. ACM, 1990.

[2] R. Heckmann. *Power Domain Constructions*. PhD thesis, Universität des Saarlandes, 1990.

[3] R. Heckmann. Power domain constructions. *Science of Computer Programming*, 17:77–117, 1991.

[4] R. Heckmann. Power domains supporting recursion and failure. In J.-C. Raoult, editor, *CAAP '92*, pages 165–181. *Lecture Notes in Comp. Sc. 581*, Springer-Verlag, 1992.

[5] M.C.B Hennessy and G.D. Plotkin. Full abstraction for a simple parallel programming language. In J. Becvar, editor, *Foundations of Computer Science*, pages 108–120. *Lecture Notes in Comp. Sc. 74*, Springer-Verlag, 1979.

[6] R. Hoofman. Powerdomains. Technical Report RUU-CS-87-23, Rijksuniversiteit Utrecht, November 1987.

[7] C.J. Jones. *Probabilistic Non-Determinism*. PhD thesis, University of Edinburgh, 1990.

[8] C.J. Jones and G.D. Plotkin. A probabilistic powerdomain of evaluations. In *LICS '89*, pages 186–195. IEEE Computer Society Press, 1989.

[9] J. Lambek and P. J. Scott. *Introduction to Higher Order Categorical Logic*, volume 7 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, 1986.

[10] S. Mac Lane. *Categories for the Working Mathematician*. Springer-Verlag, 1971.

[11] M.G. Main. Free constructions of powerdomains. In A. Melton, editor, *Mathematical Foundations of Programming Semantics*, pages 162–183. *Lecture Notes in Comp. Sc. 239*, Springer-Verlag, 1985.

[12] E. Moggi. Computational lambda-calculus and monads. In *4th LICS Conference*, pages 14–23. IEEE, 1989.

[13] E. Moggi. Notions of computation and monads. *Information and Computation*, 93:55–92, 1991.

[14] G.D. Plotkin. A powerdomain construction. *SIAM Journal on Computing*, 5(3):452–487, 1976.

[15] M.B. Smyth. Power domains. *Journal of Computer and System Sciences*, 16:23–36, 1978.

On the Equivalence of State-Transition Systems

Michael Huth

mrah@doc.ic.ac.uk

Department of Computing, Imperial College
180 Queen's Gate, London SW7 2BZ, United Kingdom

Abstract

We study frameworks for the equivalence of abstract state-transition systems represented as posets. A basic notion of equivalence is proposed. A least fix-point operator transforms basic equivalences into strong equivalences (=*Lagois Connections*) which makes Lagois Connections into a category. In the absence of divergence, the two notions of equivalence coincide. We generalize these notions by adding a logical level to express divergence more precisely. Then both generalized notions of equivalence coincide even in the presence of divergence.

1 Introduction

We consider posets as *abstract state-transition systems*. Elements s in a poset $\langle P, \sqsubseteq \rangle$ model *states* and $s \sqsubseteq s'$ expresses the intuition that state s' can be reached by performing a finite sequence of state transitions beginning with state s. The reasons for including the axiom of *antisymmetry* will become apparent in this study. Such an approach excludes an immediate analysis of transition graphs; but it does not constitute a conceptual exclusion, for antisymmetry can be restored by rewriting such systems in form of a *tree of traces*.

What is a reasonable mathematical framework for the equivalence of two such abstract state-transition systems $\langle P, \sqsubseteq \rangle$ and $\langle Q, \sqsubseteq \rangle$? Assuming that P and Q are intuitively equivalent, we should be able to associate to each state s in P a state $f(s)$ in Q and likewise a state $g(t)$ in P for each state $t \in Q$. So we should impose the existence of a pair of set-theoretic functions $f: P \to Q$ and $g: Q \to P$. Further, *reachability* should be preserved under these state transformations: if $s \sqsubseteq s'$ in P, then $f(s) \sqsubseteq f(s')$ should follow in Q. Therefore, we want f and g to be *monotone maps*. Finally, we expect the state $g(f(s))$ to be reachable from the original state s. Thus, our monotone maps should satisfy the inequalities $\mathrm{id}_P \sqsubseteq gf$ and $\mathrm{id}_Q \sqsubseteq fg$.

Definition 1 Let P and Q be posets. A pair of monotone maps $f: P \to Q$ and $g: Q \to P$ is called a *poset system* [10]; we denote this by $\langle f, g \rangle: P \to Q$. We call a poset system a *basic equivalence* iff $\mathrm{id}_P \sqsubseteq gf$ and $\mathrm{id}_Q \sqsubseteq fg$ hold. $\square$

In [10], another notion of equivalence had been proposed which is a basic equivalence $\langle f, g \rangle: P \to Q$ satisfying the equations $fgf = f$ and $gfg = g$; so f and g are additionally *quasi-inverses* of each other. The functions $gf: P \to P$ and $fg: Q \to Q$ are then *closure operators* with isomorphic image [10]. Such poset systems are called *Lagois Connections* [10]; they are a generalization of the case where one of the abstract state-transition systems sits as a closure inside the other, i.e., where $\mathrm{id}_P = gf$ or $\mathrm{id}_Q = fg$. Lagois Connections have a theory similar to the one of *Galois Connections* [4, 9, 10].

Definition 2 A *poset system* $\langle f, g\rangle : P \to Q$ is called a *strong equivalence* iff it is a basic equivalence satisfying the equations $fgf = f$ and $gfg = g$. □

So the terms *strong equivalence* and *Lagois Connection* have the same meaning. We will address three main issues in this piece of work.

- Are basic and strong equivalences really equivalences in the mathematical sense?
- If so, are they really different concepts?
- How do they match with the most common equivalence results in semantics?

2 The Category of Basic Equivalences

There is a natural way of making poset systems into a category [9]; its objects are posets, its morphisms are poset systems $\langle f, g\rangle : P \to Q$ and $\langle i, j\rangle : Q \to R$, composition is $\langle f, g\rangle \circ \langle i, j\rangle := \langle if, gj\rangle$ and identities are $\langle \mathrm{id}_P, \mathrm{id}_P\rangle : P \to P$. Basic equivalences are readily seen to be closed under that composition; so basic equivalences form naturally a category and composition verifies the transitivity of this equivalence notion.

The case of strong equivalences is problematic. Lagois Connections simply are not closed under this composition [10]. But there is an intuitive way of constructing a Lagois Connection out of a basic equivalence $\langle f, g\rangle : P \to Q$: if we *feed back* the maps f and g, we obtain for each $n \geq 1$ basic equivalences $\langle (fg)^n f, (gf)^n g\rangle : P \to Q$ which form an ascending chain. *If* our posets have limits of ω-chains, then we expect the limit of this chain to be a Lagois Connection. Therefore, we will have to abandon the realm of posets and consider *dcpos*, posets in which every directed set has a supremum [5, 7, 12, 15] (of course, ω-dcpos would suffice).

Definition 3 Let D and E be dcpos. A *dcpo system* $\langle f, g\rangle : D \to E$ is a poset system such that f and g are Scott-continuous [5, 7, 12, 15], i.e., they preserve suprema of all directed sets. Define

$$\mathcal{B}(D, E) := \{\langle f, g\rangle : D \to E \text{ dcpo system} \mid \mathrm{id}_D \sqsubseteq gf \;\&\; \mathrm{id}_E \sqsubseteq fg\} \quad (1)$$
$$\mathcal{S}(D, E) := \{\langle f, g\rangle : D \to E \in \mathcal{B}(D, E) \mid fgf = f \;\&\; gfg = g\} \quad (2)$$

and order $\mathcal{B}(D, E)$ and $\mathcal{S}(D, E)$ by

$$\langle f, g\rangle \sqsubseteq \langle i, j\rangle :\Leftrightarrow f \sqsubseteq i \;\&\; g \sqsubseteq j. \quad (3)$$

□

The letters $\mathcal{B}$ and $\mathcal{S}$ stand for *basic* and *strong* equivalences respectively. If $[D \to E]$ denotes the function space of all Scott-continuous maps $f : D \to E$ in the pointwise order, this is a dcpo where directed suprema of sets $\{f_i \mid i \in I\}$ are computed componentwise: $(\sqcup^{\uparrow}_{i \in I} f_i)(d) = \sqcup^{\uparrow}_{i \in I} f_i(d)$ [5, 7, 12, 15]. This makes $\mathcal{B}(D, E)$ and $\mathcal{S}(D, E)$ into dcpos.

Proposition 1 Let D and E be dcpos. Then $\mathcal{B}(D, E)$ is a dcpo and suprema of directed sets are computed componentwise:

$$\sqcup^{\uparrow}_{i \in I} \langle f_i, g_i\rangle = \langle \sqcup^{\uparrow}_{i \in I} f_i, \sqcup^{\uparrow}_{i \in I} g_i\rangle. \quad (4)$$

Moreover, $\mathcal{S}(D, E)$ is a subdcpo of $\mathcal{B}(D, E)$, i.e., the inclusion $\mathcal{S}(D, E) \hookrightarrow \mathcal{B}(D, E)$ is Scott-continuous. □

We can recover our poset systems $\langle f, g\rangle : P \to Q$ from dcpo systems by identifying the posets P and Q with the algebraic dcpos of their *ideal completions* [4, 7] $\mathrm{Idl}(P)$ and $\mathrm{Idl}(Q)$, and by construing f and g as Scott-continuous maps which *preserve all finite elements*. Recall, that a function $h : D \to E$ preserves finite elements iff $h(\mathrm{K}(D) \subseteq \mathrm{K}(E)$ where $\mathrm{K}(D)$ denotes the poset of *finite*, or *compact elements* [4, 5, 7, 17] of a dcpo D.

Definition 4 For dcpos D and E, define

$$\mathcal{BF}(D, E) := \{\langle f, g\rangle \in \mathcal{B}(D, E) \mid f(\mathrm{K}(D)) \subseteq \mathrm{K}(E),\, g(\mathrm{K}(E)) \subseteq \mathrm{K}(D)\} \quad (5)$$
$$\mathcal{SF}(D, E) := \mathcal{BF}(D, E) \cap \mathcal{S}(D, E) \quad (6)$$

in the order induced by $\mathcal{B}(D, E)$. □

The $\mathcal{F}$ in $\mathcal{BF}$ stands for preserving $f$inite elements. Given algebraic dcpos D and E, dcpo systems in $\mathcal{BF}(D, E)$, respectively $\mathcal{SF}(D, E)$, correspond to basic, respectively strong, equivalences between the abstract state-transition systems $\mathrm{K}(D)$ and $\mathrm{K}(E)$. Before we make abstract state-transition systems into a category, we should discuss minimal specifications on posets as representations of abstract state-transition systems. First, such a poset P should satisfy the *descending chain condition* [3]: there do not exist strictly descending chains $a_0 > a_1 > \cdots$ in P; this means that a state of a computation should be reachable by an initial state (a minimal element of P). Second, no strictly ascending chain $b_0 < b_1 < \cdots$ in P should have an upper bound in P. Otherwise, such an upper bound u would be a state with an *infinite previous history* or an *infinite cause*. We do not, however, exclude the possibility of having infinitely many states which can cause a particular state s: the domain in Figure 1 is a representation of such an abstract state-transition system.

Definition 5 Let ASS be the class of all algebraic dcpos D such that $\mathrm{K}(D)$ satisfies the descending chain condition and no strictly ascending chain in $\mathrm{K}(D)$ has an upper bound in $\mathrm{K}(D)$. Define $\mathcal{BF}$ to be the four-sorted structure $(\mathrm{ob}(\mathcal{BF}), \mathrm{morph}(\mathcal{BF}), \circ, \mathrm{id})$ such that $\mathrm{ob}(\mathcal{BF})$ is the class ASS, $\mathrm{morph}(\mathcal{BF})$ is the class of hom-sets $\mathcal{BF}(D, E)$ for D and E in ASS, composition is

$$\langle f, g\rangle \circ \langle i, j\rangle := \langle if, gj\rangle \quad (7)$$

and identities are

$$\langle \mathrm{id}_D, \mathrm{id}_D\rangle : D \to D. \quad (8)$$

Define a relation $\sim_{\mathrm{f}}$ on ASS by

$$D \sim_{\mathrm{f}} E \text{ iff } \mathcal{BF}(D, E) \neq \emptyset. \quad (9)$$

□

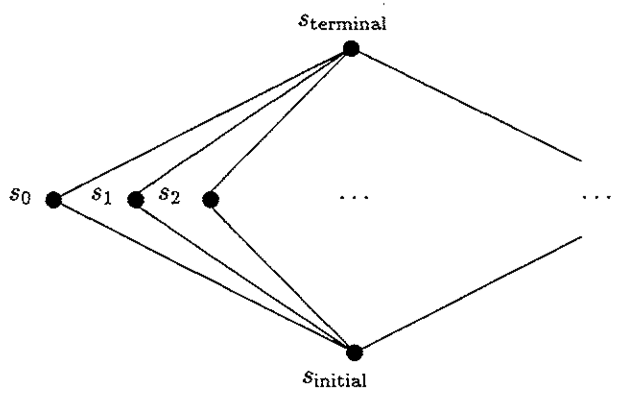

Figure 1: An abstract state-transition system with a state s_{terminal} which can be reached from infinitely many states

We readily note that $\mathcal{BF}$ is a self-dual category, i.e., $\mathcal{BF}$ is equivalent to its dual category [8], and that $\sim_{f}$ is an equivalence relation on ASS.

Theorem 1 $\mathcal{BF}$ is a self-dual category and $\sim_{f}$ is an equivalence relation on $\text{ASS} = \text{ob}(\mathcal{BF})$.

PROOF. The self-duality follows from the order-isomorphism

$$\langle f, g\rangle \mapsto \langle g, f\rangle : \mathcal{BF}(D, E) \to \mathcal{BF}(E, D). \tag{10}$$

Composition is clearly well-defined and associative; the pairs $\langle \text{id}_D, \text{id}_D\rangle$ are two-sided identities with respect to this composition. The existence of identities ensures the reflexivity of $\sim_{f}$, the self-duality of $\mathcal{BF}$ makes $\sim_{f}$ symmetric and the well-definedness of composition in $\mathcal{BF}$ makes $\sim_{f}$ transitive. □

Things don't work that smoothly for strong equivalences. At least, we can introduce a least fix-point operator which transforms basic into strong equivalences. This will be the key for obtaining a composition for strong equivalences.

Definition 6 For dcpos D and E and $\langle f, g\rangle \in \mathcal{B}(D, E)$ define

$$\begin{aligned}
\langle f_0, g_0\rangle &:= \langle f, g\rangle && (11)\\
\langle f_{n+1}, g_{n+1}\rangle &:= \langle (fg)f_n, (gf)g_n\rangle, \qquad n \geq 0 && (12)\\
\langle f_\omega, g_\omega\rangle &:= \textstyle\bigsqcup_{n\geq 0}\langle f_n, g_n\rangle. && (13)
\end{aligned}$$

□

The pair $\langle f_\omega, g_\omega\rangle$ is well-defined; it is the least Lagois Connection above $\langle f, g\rangle$ in $\mathcal{S}(D, E)$.

Theorem 2 For dcpos D and E, the map

$$\Omega_{\langle D,E\rangle}: \mathcal{B}(D, E) \to \mathcal{B}(D, E),\ \Omega_{\langle D,E\rangle} := \lambda\langle f, g\rangle.\langle f_\omega, g_\omega\rangle \tag{14}$$

is a Scott-continuous closure operator on $\mathcal{B}(D, E)$ with image $\mathcal{S}(D, E)$. □

This suggests a composition for Lagois Connections. Given dcpo systems $\langle f, g\rangle \in \mathcal{SF}(D, E)$ and $\langle i, j\rangle \in \mathcal{SF}(E, F)$, the dcpo system $\langle f, g\rangle \circ \langle i, j\rangle$ is a morphism in $\mathcal{BF}(D, F)$, so we could define the strong composition of $\langle f, g\rangle \in \mathcal{SF}(D, E)$ and $\langle i, j\rangle \in \mathcal{SF}(E, F)$ to be the least fix point of $\Omega_{\langle D,F\rangle}$ applied to $\langle f, g\rangle \circ \langle i, j\rangle$:

$$\langle f, g\rangle * \langle i, j\rangle := \Omega_{\langle D,F\rangle}\big(\langle f, g\rangle \circ \langle i, j\rangle\big). \tag{15}$$

Of course, it will be vital to know whether this operation will make $\mathcal{SF}$ into a category. Unfortunately, this is not so in the presence of divergence. For example, consider the dcpo $\omega+1$ where $\text{K}(\omega + 1) = \{0 < 1 < \cdots\}$ is an abstract state-transition system modeling a diverging stream of computation. Define the dcpo systems $\langle f, f\rangle, \langle g, g\rangle : (\omega + 1) \to (\omega + 1)$ such that $f(n)$, respectively $g(n)$, is the least odd, respectively even, number above n. Then $\langle f, f\rangle, \langle g, g\rangle \in \mathcal{SF}(\omega + 1, \omega + 1)$ is readily seen, but $\langle f, f\rangle * \langle g, g\rangle = \langle \lambda n.\omega, \lambda n.\omega\rangle$ is not even in $\mathcal{BF}(\omega + 1, \omega + 1)$ as ω is not finite in $\omega + 1$.

However, one might ask why $\langle \lambda n.\omega, \lambda n.\omega\rangle$ does not demonstrate an equivalence on $\omega + 1$? After all, each finite element being mapped to ω leads only to diverging computations.

3 Handling Divergence with a Logical Level

We therefore propose to add a logical level to dcpo systems $\langle f, g\rangle \in \mathcal{B}(D, E)$. If $\mathcal{BD}(D, E)$ denotes those $\langle f, g\rangle \in \mathcal{B}(D, E)$ satisfying the logical level still to be defined, we expect for all D and E in ASS, that

- $\mathcal{BF}(D, E)$ is a subset of $\mathcal{BD}(D, E)$,
- $\mathcal{BD}(D, E)$ is closed under $\circ$ in $\mathcal{B}(D, E)$,
- $\mathcal{BD}(D, E)$ is closed under $\Omega_{\langle D,E\rangle}$ in $\mathcal{B}(D, E)$ and
- the relation $\sim_{d}$ induced by the sets $\mathcal{BD}(D, E)$ is a computationally meaningful equivalence relation of the class ASS.

Definition 7 For a dcpo D, define

$$D^\infty := D \setminus \text{K}(D) \tag{16}$$

to be the poset of *states of divergence* of D. Given $\langle f, g\rangle \in \mathcal{B}(D, E)$, we call $\langle f, g\rangle$ a *demonstration of equivalence* iff

$$\begin{aligned}
\uparrow(f^{-1}(E^\infty)) \cap \max(D) &\subseteq D^\infty \text{ and} && (17)\\
\uparrow(g^{-1}(D^\infty)) \cap \max(E) &\subseteq E^\infty, && (18)
\end{aligned}$$

where $\max(D) := \big\{d \in D \mid \uparrow(d) = \{d\}\big\}$. Define

$$\begin{aligned}
\mathcal{BD}(D, E) &:= \{\langle f, g\rangle \in \mathcal{B}(D, E) \mid \langle f, g\rangle \text{ is demonstr. of equiv.}\} && (19)\\
\mathcal{SD}(D, E) &:= \mathcal{BD}(D, E) \cap \mathcal{S}(D, E) && (20)
\end{aligned}$$

in the order induced by $\mathcal{B}(D, E)$. Define the relation $\sim_{d}$ on ASS by

$$D \sim_{d} E \text{ iff } \mathcal{BD}(D, E) \neq \emptyset. \tag{21}$$

□

The $\mathcal{D}$ in $\mathcal{BD}$ stands for *d*ivergence. The conditions on f say the following (the explanation for g is symmetric and we omit it): if f maps $d \in D$ to a state of divergence, this is fine as long as all maximal states in D above d are also states of divergence. For example, $\langle \lambda n.\omega, \lambda n.\omega \rangle : (\omega + 1) \to (\omega + 1)$ is such a demonstration of equivalence. However, there is no demonstration of equivalence between $\omega + 1$ and $\{s\}$ as any equivalence has to map s to $\omega \in (\omega + 1)^\infty$, but s is maximal and not a state of divergence. This should suffice to justify the 'soundness' of this equivalence notion. We verify that it satisfies our proposed constraints.

Proposition 2 Let D, E and F be elements in ASS. Then we have the following.

1. Each $\langle f, g \rangle \in \mathcal{B}(D, E)$ induces a bijection between $\max(D)$ and $\max(E)$; if $\langle f, g \rangle \sqsubseteq \langle i, j \rangle$ in $\mathcal{B}(D, E)$, then $\langle f, g \rangle$ and $\langle i, j \rangle$ induce the same bijection,
2. $\mathcal{BD}(D, E)$ is an upper set in $\mathcal{B}(D, E)$,
3. $\mathcal{BD}(D, E)$ contains $\mathcal{BF}(D, E)$ and
4. if $\langle f, g \rangle \in \mathcal{BD}(D, E)$ and $\langle i, j \rangle \in \mathcal{BD}(E, F)$, then $\langle f, g \rangle \circ \langle i, j \rangle := \langle if, gj \rangle$ is in $\mathcal{BD}(D, F)$. □

Since $\mathcal{BD}(D, E)$ is an upper set in $\mathcal{B}(D, E)$ and since $\circ$ is well-defined on $\mathcal{BD}$, we can draw several conclusions: $\mathcal{BD}$ is a category, $\sim_d$ is an equivalence relation and $*$ is a well-defined operation on the hom-sets in $\mathcal{BD}$.

Theorem 3 $\mathcal{BD}$ is a self-dual category and $\sim_d$ is an equivalence relation on ASS $= \mathrm{ob}(\mathcal{BF})$. Moreover, $\mathcal{BF}$ is a full subcategory of $\mathcal{BD}$ and $\sim_f$ is a subclass of $\sim_d$.

PROOF. The proof that $\mathcal{BD}$ is a self-dual category is similar to the corresponding proof for $\mathcal{BF}$. Since $\mathcal{BF}(D, E) \subseteq \mathcal{BD}(D, E)$, the rest is clear. □

4 The Category of Lagois Connections

We have seen that $*$ is a well-defined operation on the hom-sets in $\mathcal{BD}$. Note that the identities on $\mathcal{BD}$ serve also as two-sided identities for $*$ applied to strong equivalences as $\langle \mathrm{id}_D, \mathrm{id}_D \rangle$ is already a strong equivalence for D in ASS; but how can we prove that $*$ is associative?

The solution follows a general category-theoretical pattern which we have so far not been able to match with standard concepts in category theory [2, 8].

Definition 8 Let $\mathcal{C}$ be a category and Ω a class of set-theoretic functions

$$\Omega_{\langle A,B \rangle} : \mathcal{C}(A, B) \to \mathcal{C}(A, B) \tag{22}$$

for all objects A and B in $\mathcal{C}$. We call Ω *an associative structure over* $\mathcal{C}$ iff for all morphisms $f \in \mathcal{C}(B, C)$ and $g \in \mathcal{C}(A, B)$, we have

$$\Omega_{\langle A,C \rangle}(f \circ (\Omega_{\langle A,B \rangle} g)) = \Omega_{\langle A,C \rangle}(f \circ g) \tag{23}$$
$$= \Omega_{\langle A,C \rangle}((\Omega_{\langle B,C \rangle} f) \circ g). \tag{24}$$

Define

$$\mathcal{C}[\Omega](A, B) := \mathrm{im}(\Omega_{\langle A,B \rangle}) = \{\Omega_{\langle A,B \rangle} f \mid f \in \mathcal{C}(A, B)\} \tag{25}$$
$$\mathrm{id}^\Omega_A := \Omega_{\langle A,A \rangle} \mathrm{id}_A \text{ and} \tag{26}$$
$$f * g := \Omega_{\langle A,C \rangle}(f \circ g). \quad \square \tag{27}$$

The four-sorted structure $(\mathrm{ob}(\mathcal{C}), \mathcal{C}[\Omega], *, \mathrm{id}^\Omega)$ is a category and Ω induces a functor $\Omega : \mathcal{C} \to \mathcal{C}[\Omega]$.

Theorem 4 Let $\mathcal{C}$ be a category and Ω an associative structure over $\mathcal{C}$. Then we have the following:

1. For all objects A and B in $\mathcal{C}$, the function $\Omega_{\langle A,B \rangle}$ is a retraction on $\mathcal{C}(A, B)$,
2. the four-sorted structure $(\mathrm{ob}(\mathcal{C}), \mathcal{C}[\Omega], *, \mathrm{id}^\Omega)$ is a category and
3. if Ω also denotes the function which leaves objects of $\mathcal{C}$ fixed and maps morphisms $f \in \mathcal{C}(A, B)$ to $\Omega_{\langle A,B \rangle} f$, then Ω is a functor

$$\Omega : \mathcal{C} \to \mathcal{C}[\Omega]. \tag{28}$$

□

We will now show that the class of Scott-continuous closure operators of Theorem 2 is an associative structure over the category $\mathcal{BD}$. This will make Lagois Connections into a category. First, we will reduce the complexity of such a proof; note that the operator Ω occurs *nested* in the equations (23)–(24). We can use the concrete poset structure of the category $\mathcal{BD}$ to reduce these equations to equivalent inequalities in which Ω only appears at one level.

Proposition 3 Let $\mathcal{C}$ be a category such that all hom-sets $\mathcal{C}(A, B)$ are posets and composition is monotone. If Ω is a class of closure operators $\Omega_{\langle A,B \rangle}$ on $\mathcal{C}(A, B)$ for all objects A and B in $\mathcal{C}$, then Ω is an associative structure over $\mathcal{C}$ if for all morphisms $f \in \mathcal{C}(B, C)$ and $g \in \mathcal{C}(A, B)$, we have

$$f \circ (\Omega_{\langle A,B \rangle} g) \sqsubseteq \Omega_{\langle A,C \rangle}(f \circ g) \text{ and} \tag{29}$$
$$(\Omega_{\langle B,C \rangle} f) \circ g \sqsubseteq \Omega_{\langle A,C \rangle}(f \circ g). \tag{30}$$

In that case, $*$ is monotone. □

Theorem 5 The class of Scott-continuous closure operators Ω of Theorem 2 is an associative structure over the category $\mathcal{BD}$. In particular,

$$\mathcal{SD} := \mathcal{BD}[\Omega] \tag{31}$$

is a category, the *category of Lagois Connections*. The composition in $\mathcal{SD}$ is Scott-continuous and is given by the *feed-back formula*

$$\langle f, g \rangle * \langle i, j \rangle = \sqcup_{n \geq 0} \langle (ifgj)^n (if), (gjif)^n (gj) \rangle. \tag{32}$$

PROOF. The assumptions of Proposition 3 apply, so it suffices to show the inequalities (29)–(30). For $\langle f, g \rangle \in \mathcal{BD}(D, E)$ and $\langle i, j \rangle \in \mathcal{BD}(E, F)$, we compute the first coordinates of expressions occurring in these inequalities. The proof for the second coordinates is similar and we omit it. The first coordinate of $\Omega_{\langle D,F \rangle}(\langle f, g \rangle \circ \langle i, j \rangle)$ is $\sqcup_{n \geq 0} (ifgj)^n (if)$, the first coordinate of $\langle f, g \rangle \circ (\Omega_{\langle E,F \rangle} \langle i, j \rangle)$ equals $\sqcup_{n \geq 0} (ij)^n (if)$ and the first coordinate of $(\Omega_{\langle D,E \rangle} \langle f, g \rangle) \circ \langle i, j \rangle$ is $\sqcup_{n \geq 0} i(fg)^n f$, so we are done if $i(fg)^n f, (ij)^n (if) \sqsubseteq (ifgj)^n (if)$ for all $n \geq 1$; this is an easy induction. □

One could now define an equivalence relation $\approx_d$ such that $D \approx_d E$ in ASS iff $\mathcal{SD}(D,E) \neq \emptyset$. But since $\Omega_{\langle D,E\rangle}(\mathcal{BD}(D,E)) = \mathcal{SD}(D,E)$, this relation equals $\sim_d$. Thus, these relations render the same concept of equivalence even in the presence of divergence. What can we say in the absence of divergence? If TERM denotes the class of all D in ASS such that $\mathrm{K}(D) = D$, then we have the set equalities $\mathcal{BF}(D,E) = \mathcal{BD}(D,E) = \mathcal{B}(D,E)$ as there are no states of divergence for D and E in TERM.

Theorem 6 On the class TERM, the relations $\sim_d$, $\sim_t$ and $\approx_t$ are equal; where we define

$$D \approx_t E \text{ iff } \mathcal{SF}(D,E) \neq \emptyset. \tag{33}$$

In particular, $\approx_t$ is an equivalence relation on TERM. □

One might ask whether $\approx_t$ is also transitive on ASS, or whether its transitive closure equals $\sim_t$ or $\sim_d$; we have not investigated this any further.

The feed-back formula intrinsically contains cases of *nilpotency* which had been noted in [10] as instances of a well-defined composition for Lagois Connections.

Proposition 4 For D and E in ASS and $\langle f,g\rangle \in \mathcal{BD}(D,E)$, we have

1. $\langle fg, fg\rangle \in \mathcal{BD}(E,E)$ and $\langle gf, gf\rangle \in \mathcal{BD}(D,D)$,
2. if $fgf = f$, then
$$\Omega_{\langle D,E\rangle}\langle f,g\rangle = \langle f, gfg\rangle, \tag{34}$$
3. if $gfg = g$, then
$$\Omega_{\langle D,E\rangle}\langle f,g\rangle = \langle fgf, g\rangle, \tag{35}$$
4. if fg or gf is idempotent, then
$$\Omega_{\langle D,E\rangle}\langle f,g\rangle = \langle fgf, gfg\rangle, \tag{36}$$
$$\Omega_{\langle E,E\rangle}\langle fg,fg\rangle = \langle fgfg, fgfg\rangle \text{ and} \tag{37}$$
$$\Omega_{\langle D,D\rangle}\langle gf,gf\rangle = \langle gfgf, gfgf\rangle. \tag{38}$$

□

Let us point out the conceptual gain of these results. By acknowledging the logical level as a sound criterion of equivalence, we can prove two objects D and E in ASS to be equivalent by specifying a basic equivalence $\langle f,g\rangle : D \to E$ in $\mathcal{BD}(D,E)$. If we prefer to work with a Lagois connection, we can simply take $\langle f,g\rangle$ and compute the least fix point $\Omega_{\langle D,E\rangle}\langle f,g\rangle$ which will be an element of $\mathcal{SD}(D,E)$.

There is a larger framework than the categories $\mathcal{BD}$ and $\mathcal{SD}$ which does not model equivalences of abstract state-transition systems, but which does have a mathematical interest in its own right. If $\mathcal{B}$ denotes the four-sorted structure which has all dcpos as objects, dcpo systems as morphisms, $\circ$ as composition and pairs $(\mathrm{id}_D, \mathrm{id}_D)$ as morphisms, then $\mathcal{B}$ is a category which contains $\mathcal{BD}$ as a non-full subcategory. Likewise, if we consider

$$\mathcal{S} := \mathcal{B}[\Omega] \tag{39}$$

then we obtain a corresponding version of Theorem 5 for $\mathcal{B}$ and $\mathcal{S}$.

5 Basic Equivalences in Semantics

We want to relate the concepts and results of the preceding sections to familiar situations in formal semantics of programming languages. As a first example, let us compare *operational semantics* for a simple deterministic, imperative `WHILE`-language [5, 12, 15]. The concrete choice of such a language is irrelevant for the purpose of this discussion. We only need to know that states for such a language come in two flavors. A state is either a pair $\langle S, s\rangle$ where S is a statement in the language and s denotes an *environment* which binds free variables in S, or it is a *terminal state* s'. Intuitively, a state s' models the termination of a program's response to an initial environment. Assuming that our programming language does not produce any *stuck* configurations [12], a state $\langle S, s\rangle$ leads to a successor state which is either terminal, i.e., of the from s', or again of the form $\langle S', s'\rangle$.

There are two prominent operational semantics for such `WHILE`-languages, called *structural operational semantics* and *natural semantics* [12]. The structural operational semantics is defined by a relation $\Rightarrow$ which specifies what the successor states are; then $\Rightarrow^*$ is the transitive closure of $\Rightarrow$. This semantics models a *single-step* computation. The natural semantics models a *big-step* computation. We write $\langle S, s\rangle \to s'$ iff the computation starting at $\langle S, s\rangle$ ends in state s'. Otherwise, the relation $\to$ is undefined. Let $\mathcal{NS}$ be the preorder obtained by forming the reflexive closure of $\to$ on the set of states of our `WHILE`-language, and let $\mathcal{SOS}$ be the preorder obtained by forming the reflexive closure of $\Rightarrow^*$ on the same set. We have functions $f : \mathcal{NS} \to \mathcal{SOS}$ and $g : \mathcal{SOS} \to \mathcal{NS}$ defined by

$$f\langle S,s\rangle := \langle S,s\rangle, \tag{40}$$
$$fs' := s', \tag{41}$$
$$gs' := s' \text{ and} \tag{42}$$
$$g\langle S,s\rangle := \texttt{if } \langle S,s\rangle \Rightarrow^* s' \texttt{ then } s' \texttt{ else } \langle S,s\rangle. \tag{43}$$

We readily check that f is monotone; this means that $\langle S, s\rangle \to s'$ implies $\langle S, s\rangle \Rightarrow^* s'$, so the big-step deduction is sound with respect to the single-step deduction. We also have $\mathrm{id}_{\mathcal{NS}} \sqsubseteq gf$ because $\langle S, s\rangle \Rightarrow^* s'$ guarantees $\langle S, s\rangle \to s'$. Further, we obtain $\mathrm{id}_{\mathcal{SOS}} \sqsubseteq fg$ and the equations $fgf = f$ and $gfg = g$. So it looks as if

$$\langle f,g\rangle : \mathcal{NS} \to \mathcal{SOS} \tag{44}$$

is a Lagois Connection between the preorders $\mathcal{NS}$ and $\mathcal{SOS}$. Yet, we did not prove the *monotonicity* of g; but g simply is not monotone. Consider a transition $\langle S, s\rangle \Rightarrow^* \langle S', s'\rangle$ such that the computation according to $\Rightarrow^*$ is diverging after $\langle S, s\rangle$, and therefore after $\langle S', s'\rangle$ as well. By definition, $g\langle S, s\rangle = \langle S, s\rangle$ and $g\langle S', s'\rangle = \langle S', s'\rangle$, but $\to$ is a discrete order on the set of states $\langle S, s\rangle$ which give rise to diverging behavior. So our example is really a good *non-example*. The fact that g is not monotone captures the essential difference between these two operational semantics: the one-step ordering on diverging streams has no equivalent in the big-step computation. It is worth pointing out that the mere existence of f and g suffice to prove the usual semantic correspondence theorems [12]; the non-monotone behavior of g does not affect this.

It is possible to state another big-step operational semantics such that we do have a Lagois Connection between it and the structural operational semantics. For that, let D be the ideal completion of $\mathcal{SOS}$; so we just add one limit point for each diverging stream $\langle S_0, s_0\rangle \Rightarrow \langle S_1, s_1\rangle \Rightarrow \cdots$. Since our language is assumed to be deterministic, the relation $\Rightarrow$ can be viewed as a Scott-continuous function $h\colon D \to D$ such that

$$\begin{aligned} h\langle S,s\rangle &:= s' \texttt{ if } \langle S,s\rangle \Rightarrow s', && (45)\\ h\langle S,s\rangle &:= \langle S',s'\rangle \texttt{ if } \langle S,s\rangle \Rightarrow \langle S',s'\rangle \text{ and} && (46)\\ hs' &:= s'. && (47) \end{aligned}$$

Since $\mathrm{id}_D \sqsubseteq h$ is obvious, we conclude $\langle h,h\rangle \in \mathcal{BF}(D,D)$. Therefore, we can define

$$\langle \texttt{bigstep}, \texttt{bigstep}\rangle := \Omega_{(D,D)}\langle h,h\rangle \tag{48}$$

which is an element in $\mathcal{SD}(D,D)$. Moreover, $\texttt{bigstep}$ is a Scott-continuous closure operator on D. Its image consists of all elements of D^∞ and all terminal states s'. The n-th approximation $\langle h_n, h_n\rangle$ of the fix point $\langle \texttt{bigstep}, \texttt{bigstep}\rangle$ computes $2n+1$ steps from a given state $\langle S,s\rangle$ as

$$\langle h_n, h_n\rangle = \langle (hh)^n h, (hh)^n h\rangle = \langle h^{2n+1}, h^{2n+1}\rangle. \tag{49}$$

It would be interesting to relate operational semantics and *abstract machines* [12] using the concepts of basic and strong equivalences. We will not do this here for lack of space. However, let us remark that certain configurations on abstract machines might not correspond to anything meaningful in a given operational semantics. Thus, one should study the mathematical framework proposed in this paper in a setting of *partial* equivalences. We have not yet investigated this any further.

Given an abstract machine for our WHILE-language, one usually proves that the semantic function induced by this machine is equivalent to the one induced by the natural semantics. However, one can also prove this using the structural operational semantics [12]. This is being done by introducing a *bisimulation relation* [12]. It seems as if the specification of such a relation is nothing else than the definition of a Lagois Connection between the respective preorders of states, for Lagois Connections $\langle f,g\rangle\colon P \rightharpoonup Q$ can be characterized as certain equivalence relations on P and Q [10]. This apparent analogy needs further study.

A simple example of a strong equivalence between an abstract machine and an operational semantics can be found in [10]. It relates an operational semantics of *marked infix arithmetic expressions* to a stack machine for evaluating *postfix arithmetic expressions*.

In [6], an abstract semantics for a higher-order functional language with logic variables has been given by *solving simultaneous fix-point equations of closure operators* over Scott-domains. Since constraints are modeled by closure operators [6], one wants to associate a closure operator h with two constraints modeled by f and g. It turns out that this can be viewed as

$$\langle h,h\rangle := \langle f,f\rangle * \langle g,g\rangle. \tag{50}$$

There is an underlying symmetry for the composition $*$ if all participants are closure operators represented as in equation (50). This reflects the permutation symmetry of a finite set of constraints as the composition $*$ is then *commutative*. Our mathematical framework could serve as a 'clean' approach to this kind of constraint semantics.

Figure 2: dcpos D_0 and D_1 in ASS with $\max(D_0) \cong \max(D_1)$ such that there is no basic equivalence $\langle f,g\rangle\colon D_0 \rightharpoonup D_1$

We saw above that the existence of a Lagois Connection between two operational semantics, where one map was only partially monotone, brought about a correspondence theorem between them: the semantic functions induced by them are extensionally equal. The existence of a basic equivalence is actually stronger than the extensional equality of semantics functions between the respective state-transition systems. For example, let D denote the dcpo in ASS which has one initial state 0 and then two possible successor states a_0 and b_0 which initiate diverging streams of computation $a_0 < a_1 < \cdots$ and $b_0 < b_1 < \cdots$. The semantic function of D equals the one of $\omega + 1$, namely it is $\lambda s.\bot$ as no computation terminates. A basic equivalence $\langle f,g\rangle\colon D \rightharpoonup (\omega+1)$ would induce bijections between $\max(D)$ and $\max(\omega+1)$, but the first set has two elements and the second one is a singleton. Therefore, basic equivalences not only control terminating behavior, they also guarantee that the pattern of diverging streams coincide.

Also, having a bijection between $\max(D)$ and $\max(E)$ is not sufficient to conclude $D \sim_{\mathrm{d}} E$. As an example, take the domains D_0 and D_1 in Figure 2 which have each two maximal elements. It is readily seen that there is no basic equivalence $\langle f,g\rangle\colon D_0 \rightharpoonup D_1$.

6 Conclusions

We established a category $\mathcal{BD}$ with abstract state-transition systems as objects and basic equivalences as morphisms. We introduced the concept of an associative structure on a category which lead to the category $\mathcal{SD}$ with abstract state-transition systems as objects and Lagois Connections as morphisms. The categorical structures gave us two equivalence relations; since basic equivalences are transformed into Lagos Connections by a least fix-point operator, these two equivalence relations coincide. We discussed structural operational semantics and natural semantics in light of these results. The supremum of closure operators is the composition in $\mathcal{SD}$. This gives an interactive account of a semantics of constraint programming.

Acknowledgements

Thanks to Austin Melton for sending me a copy of [10] so fast. Samson Abramsky made most valuable comments throughout the progress of this work. Olivier Danvy was a critical and helpful reader of this article. Radha Jagadeesan pointed out to me the relevance of the composition $*$ to his work in [6]. Geoffrey Burn and Abbas Edalat helped shaping this paper as reviewers.

References

[1] S. Abramsky and R. Jagadeesan *New Foundations for the Geometry of Interaction*; in the proceedings of the LICS 1992 conference, pp.211–222

[2] J. Adámek, H. Herrlich and G. Strecker *Abstract And Concrete Categories*, J. Wiley & Sons, Inc., 1990

[3] G. Birkhoff *Lattice Theory*, American Mathematical Society Colloquium Publications, volume 25, third edition, third printing, reprinted with corrections, 1984

[4] G. Gierz, K. H. Hofmann, K. Keimel, J. D. Lawson, M. W. Mislove and D. Scott *A Compendium of Continuous Lattices*, Springer Verlag, New York, 1980

[5] C. A. Gunter *Semantics of Programming Languages*, Foundations in Computing Series, The MIT Press, Cambridge, Massachusetts, 1992

[6] R. Jagadeesan and K. Pingali *Abstract Semantics for a Higher-Order Functional Language with Logic Variables*, in the Conference Record of the Nineteenth Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, Albuquerque, New Mexico, January 19–22, 1992, ACM Press, pp. 355–366

[7] A. Jung *Cartesian Closed Categories of Domains*, CWI Tract **66**, Amsterdam, 110pp., 1989

[8] S. Mac Lane *Categories for the Working Mathematician*, Springer Verlag, New York, 1971

[9] A. Melton, B. S. W. Schröder and G. E. Strecker *Connections*, in the proceedings of the 7th workshop on the Mathematical Foundations of Programming Semantics, LNCS **598**, Springer Verlag, New York, 1992, pp.492–506

[10] A. Melton, B. S. W. Schröder and G. E. Strecker *Lagois Connections—a Counterpart to Galois Connections*, Technical Report, Department of Information and Computing Sciences, Kansas State University, 1993

[11] R. Milner *Communication and Concurrency*, Prentice Hall International Series In Computer Science, London, 1989

[12] H. R. Nielson and F. Nielson *Semantics With Applications*, Wiley Professional Computing, England, 1992

[13] B. C. Pierce *Basic Category Theory for Computer Scientists*, Foundations of Computing Series, MIT Press 1991

[14] G. Plotkin *The category of complete partial orders: a tool of making meaning*, in: *Proceedings of the Summer School on Foundations of Artificial Intelligence and Computer Science*, Instituto di Scienze dell'Informazione, Universita di Pisa, 1978

[15] D. A. Schmidt *Denotational Semantics*, Allyn and Bacon, Inc., 1986

[16] D. Scott *Domains for Denotational Semantics*, in: *ICALP82*, M. Nielsen and E. M. Schmidt (editors), Springer Verlag, LNCS **140**, 1982

[17] G. Q. Zhang *Logic of Domains*, Birkhäuser, Boston, 1991

Towards a Modal Logic of Durative Actions

Stuart Kent

sjk@doc.ic.ac.uk

Department of Computing, Imperial College
180 Queen's Gate, London SW7 2BZ, United Kingdom

Abstract

This paper proposes an extension of modal action logics, which typically make the assumption that an action is atomic, to include durative actions. These logics have been developed to support the formal specification of information systems: we argue, with particular reference to object oriented systems, that assuming atomicity is too restrictive to express many kinds of temporal constraint. In consequence, we propose that actions be regarded as durative, and encode this by assuming that an action occurs over a sequence of atomic transitions, or interval, rather than a single transition. With this as a pre-requisite, the paper continues to redefine and extend operators of atomic action logics to fit the durative case.

1 Introduction

This paper describes work whose aim is to support the formal specification, characterisation and development of object oriented systems. Specifically, we propose an extension to action logics [16, 18, 7, 15] which have already been used with some success in providing axiomatic semantics to object oriented specification languages [8, 9] and thereby providing a basis for reasoning about object oriented systems and their development. The core of our proposal is a reworking of the semantics of such logics to admit *durative* actions.

An action logic distinguishes between terms denoting actions performed in the system and terms denoting values. The value-denoting terms are used to represent the state of the "abstract machine" being specified. Actions identify transitions to change that state. Typically, the semantics of these logics force actions to be atomic:[1] they are regarded as occurring over single atomic transitions and, with regard to concurrency, it is only possible to express restrictions on their synchronisation. However, when constructing an object oriented model of a system one usually makes the assumption that methods have duration, and, in general, may operate concurrently not just synchronously –ie. one method may start whilst another is in progress, and only parts of a method need to synchronise where conflict avoidance is required. This is reflected in the recent development of OO programming languages, such as DRAGOON [4] and POOL [3], in which concurrent method execution is the norm and language

[1] Perhaps an exception to this is [17], which admits compound durative actions (eg. actions constructed using a sequential combinator). However, primitive durative actions (ie. actions whose structure is not determined) are not allowed.