Stateful values as finitely supported paths (work in progress)

Paul Blain Levy, University of Birmingham

"Full ground references" means references to integers and to other references, but not functions or thunks. A game model for full ground references was presented in [6], and a possible world model in [3]. In this note we propose to represent a full ground value with hidden state as an finitely supported infinite path through the so-called *heap possibility graph*. (Not to be confused with the usual representation of a heap as a graph.) This gives a semantics of stateful full ground values that is *explicit* in the sense that denotations are not equivalence classes.

Let $\mathcal{S}$ be a set of *reference sorts*; the intention is that every reference (memory address) has a sort.

- The *full ground types* are as follows. $A ::= \; 0 \mid A + A \mid \sum_{i\in\mathbb{N}} A_i \mid 1 \mid A \times A \mid \mathsf{Ref}_c \; (c \in \mathcal{S})$
- A *sorted set* is finite set of *references* l, each with an associated sort $\mathsf{sort}(l)$. A *world* is a finite sorted set.
- For a world w and full ground type A, the judgement $w \vdash^{\mathsf{v}} V : A$ says that V is value of type A in world w. It is inductively given as follows.

$$\frac{w \vdash^{\mathsf{v}} V : A}{w \vdash^{\mathsf{v}} \texttt{inl}\ V : A + B} \qquad \frac{w \vdash^{\mathsf{v}} V : B}{w \vdash^{\mathsf{v}} \texttt{inr}\ V : A + B} \qquad \frac{w \vdash^{\mathsf{v}} V : A_{\hat{\imath}}}{w \vdash^{\mathsf{v}} \texttt{in}_{\hat{\imath}}\ V : \sum_{i\in\mathbb{N}} A_i}\ \hat{\imath} \in \mathbb{N}$$

$$\frac{}{w \vdash^{\mathsf{v}} \langle\rangle : 1} \qquad \frac{w \vdash^{\mathsf{v}} V : A \quad w \vdash^{\mathsf{v}} W : B}{w \vdash^{\mathsf{v}} \langle V, W\rangle : A \times B} \qquad \frac{}{w \vdash^{\mathsf{v}} l : \mathsf{Ref}_{\mathsf{sort}(l)}}\ l \in w$$

.

A *syntactic full ground signature* is a set $\mathcal{S}$ of reference sorts, and for each $c \in \mathcal{S}$ a full ground type D_c over $\mathcal{S}$ called the *content type* of c. The intention is that a reference of sort c stores a value of type D_c. Given such a signature, a *stateful value* $w \vdash^{\mathsf{sv}} (x, s, V) : A$ consists of

- a finite sorted set x—we think of cells in x as local/private, whereas those in w are global/public
- for each local cell l in x, a value $w + x \vdash^{\mathsf{v}} s_l : D_{\mathsf{sort}(l)}$
- and a value $w + x \vdash^{\mathsf{v}} V : A$.

Intuitively, adding local cells to a stateful value should not make a material difference, and neither should permuting the cells. We therefore write $w \vdash^{\mathsf{v}} (x, s, V) \equiv (y, t, W) : A$ when there is a sort-preserving partial injection $\alpha : x \rightarrowtail y$ such that (x, s, V) extends a stateful value $(\mathrm{dom}(\alpha), s', V')$ by some additional local cells, and (y, t, W) extends the corresponding stateful value $(\mathrm{range}(\alpha), t', W')$ by some additional local cells. This is an equivalence relation. Our task is to give an explicit description of stateful values modulo equivalence.

For a fixed $\mathcal{S}$, we consider two denotational semantics of full ground types: strong nominal sets and strong named sets. Our focus is on the latter, but the former may provide helpful intuition.

1. **Strong nominal sets** For each sort c, let $\mathbb{A}_c$ be an infinite set; its elements are called *atoms*, and in our setting represent references. Let $\mathrm{Perm}(\mathbb{A}_c)$ be the group of permutations of $\mathbb{A}_c$, so $\prod_{c\in\mathcal{S}} \mathrm{Perm}(\mathbb{A}_c)$ is the group of sort-preserving permutations. Let G be the subgroup consisting of finitely supported permutations. A *nominal set* [1] is a set X equipped with an action $\cdot$ of G, such that for every $x \in X$ some finite set U *supports* x i.e. every $\pi \in G$ that fixes every $a \in U$ fixes x. Hence (it may be shown) every $x \in X$ has a least finite support, called $\mathsf{supp}(x)$. A nominal set $(X, \cdot)$ is *strong* [8, 10] when for every $x \in X$, every $\pi \in G$ that fixes x fixes every $a \in \mathsf{supp}(x)$. This property is preserved by sum and finite product, so every type denotes a strong nominal set. In particular $\texttt{Ref}_c$ denotes the set of c-sorted atoms with the evident action.

2. **Strong named sets** For any finite sorted set w, we write $P(w)$ for the strong nominal set consisting of sortwise injective maps sending each element of w to an atom of the same sort. Any strong nominal set is of the form $\sum_{i\in I} P(w_i)$ for a family $(w_i \mid i \in I)$ of finite sorted sets. More precisely, we have an equivalence from $\mathrm{Fam}((\mathbf{Set}/\mathcal{S})^{\mathsf{op}})$ to the category of strong nominal sets and equivariant functions, sending $(w_i \mid i \in I)$ to $\sum_{i\in I} P(w_i)$. (This is a special case of the equivalence between named sets and nominal sets [2, 9].) We thus say that a *strong named set* is a family of finite sorted sets; and each full ground type A denotes a strong named set $[\![A]\!]$.

For example, with one sort, the type $\texttt{Ref} + \texttt{Ref} \times \texttt{Ref}$ denotes a family with three indices: a left index with a singleton set, a right index with a singleton set, and a right index with a doubleton set. This is because there are three kinds of value of this type: $\texttt{inl}\ l$, $\texttt{inr}\ \langle l, l \rangle$, and $\texttt{inr}\ \langle l, l' \rangle$ where $l \neq l'$.

The sum of strong named sets is just the sum of families, and 1 (the unit) is the singleton family of the empty set, but the binary product is more complicated. For finite sorted sets w and x and a sort-preserving partial injection $\alpha : x \rightarrowtail w$, let $\mathsf{Tot}(w, x, \alpha)$ be the finite sorted set obtained from $w + x$ by identifying α-related elements. Then we set $(w_i \mid i \in I) \times (x_j \mid j \in J) \stackrel{\text{def}}{=} (\mathsf{Tot}(w_i, x_j, \alpha) \mid i \in I,\ j \in J,\ \alpha : x_j \rightarrowtail w_i)$.

A *semantic full ground signature* consists of a set $\mathcal{S}$, and for each sort $c \in \mathcal{S}$ an $\mathcal{S}$-sorted strong named set E_c. Thus a syntactic full ground signature $(D_c)_{c \in \mathcal{S}}$ denotes the semantic one $[\![D_c]\!]_{c \in \mathcal{S}}$.

A semantic full ground signature gives rise to the following *heap possibility graph*. A node consists of a world w and a strong named set $(x_i \mid i \in I)$ and is written $w \vdash (x_i \mid i \in I)$. An edge from this node consists of an index $i \in I$ and sort-preserving partial injection $\alpha : x_i \rightarrowtail w$. The target of this edge is the node $\mathsf{Tot}(w, x_i, \alpha) \vdash \prod_{l \in x_i \setminus \mathrm{dom}(\alpha)} E_{\mathsf{sort}(l)}$. The sort-preserving injection $w \rightarrowtail \mathsf{Tot}(w, x_i, \alpha)$ is the *world embedding* associated with this edge.

A stateful value $w \vdash^{\mathsf{sv}} (x, s, V) : A$ will denote an infinite path from the node $w \vdash [\![A]\!]$. Intuitively such a value must firstly provide tags, then several distinct locations, some of which will come from w and some of which are new. The first edge of the path will provide all this data. For the new cells, we have to provide the values stored and they too are stateful values, so we continue through the graph.

Any infinite path gives a sequence of world embeddings $w_0 \rightarrowtail w_1 \rightarrowtail w_2 \rightarrowtail \cdots$ whose colimit (a sorted set) is called the *support* of the path. Since a stateful value provides only finitely many local cells, the infinite path it denotes must have finite support. An infinite path has finite support iff it contains an edge (i, α) with α total; then every subsequent node is of the form $w \vdash 1$.

We thus obtain a bijection from the stateful values $w \vdash^{\mathsf{sv}} (x, s, V) : A$, modulo $\equiv$, to the finitely supported infinite paths from $w \vdash [\![A]\!]$. This gives an explicit semantics of stateful values: denotations are not equivalence classes. We speculate that this will lead to a model of a first-order language that is a fragment of both the game model and the possible world model of higher-order computation; and also to novel presentations of both kinds of model. For possible world models, to an explicit description of partial injections with initialization [7]. For game models of references [5, 6, 4], to a formulation, based on strong named sets rather than strong nominal sets, in which every move includes a finitely supported path through a heap possibility graph.

References

[1] M. J. Gabbay and A. M. Pitts. A new approach to abstract syntax with variable binding. *Formal Aspects of Computing*, 13:341–363, 2001.

[2] Fabio Gadducci, Marino Miculan, and Ugo Montanari. About permutation algebras, (pre)sheaves and named sets. *Higher-Order and Symbolic Computation*, 19(2-3):283–304, 2006.

[3] O. Kammar, P.B.Levy, S.Moss, and S.Staton. A monad for full ground references. to appear in the proceedings of LICS 2017.

[4] J. Laird. A fully abstract trace semantics for general references. In Lars Arge, Christian Cachin, Tomasz Jurdzinski, and Andrzej Tarlecki, editors, *Proceedings, 34th International Colloquium on Automata, Languages and Programming, Wroclaw, Poland*, volume 4596 of *Lecture Notes in Computer Science*, pages 667–679. Springer, 2007.

[5] A. S. Murawski and N. Tzevelekos. Game semantics for good general references. In *LICS*, pages 75–84. IEEE Computer Society, 2011.

[6] A. S. Murawski and N. Tzevelekos. Algorithmic games for full ground references. In *ICALP (2)*, volume 7392 of *LNCS*, pages 312–324, 2012.

[7] P.B.Levy. Kripke semantics for full ground references (work in progress). LOLA 2016.

[8] U. Schöpp. *Names and binding in type theory.* PhD thesis, University of Edinburgh, UK, 2006. British Library, EThOS.

[9] S. Staton. *Name-passing process calculi: operational models and structural operational semantics.* PhD thesis, University of Cambridge, UK, 2007.

[10] N. Tzevelekos. Full abstraction for nominal general references. In *LICS*. IEEE Computer Society, 2007.