Sequential algorithms as bistable maps

Pierre-Louis Curien

October 18, 2002

Abstract

In this short note, triggered by [5, 6], I exhibit the model of sequential algorithms as a full subcategory of Laird's bistable bi-orders (without reference to the simple type hierarchy over a specific base type). All the structures considered here are finite, for simplicity. These notes try to be self-contained.

1 Background

Recall that a sds [1] is a triple $\mathbf{S} = (C, V, P)$, where C is a set of cells, V is a set of values (C and V disjoint), P a prefix-closed set of non-empty alternating sequences $c_1 v_1 c_2 \dots$ (with the c_i's in C and the v_i's in V). Q (resp. R) is the subset of sequences of odd length in P, called queries (resp. responses). One moreover assumes two special other values $\bot$ and $\top$, not in V (nor in C) (in earlier work [2], $\top$ was called error, and in ludics [4] it is called Demon). Here is a syntax for the strategies of an sds (a strategy is a set of terms of the syntax of type $\mathit{strat}(c)$ for some initial cells):

$$\frac{rc \in Q}{c\bot : \mathit{strat}(rc)} \qquad \frac{rc \in Q}{c\top : \mathit{strat}(rc)}$$

$$\frac{rc_0v_0 \in R \qquad \forall\, c \text{ s.t. } rc_0v_0c \in Q \;\; x_c : \mathit{strat}(rc_0v_0c)}{c_0v_0\{x_c \mid rc_0v_0c \in Q\} : \mathit{strat}(rc_0)}$$

We shall denote the set of strategies by $D^{\top}(\mathbf{S})$, and we use $D(S)$ for the set of strategies obtained by removing the rule introducing $c\top$. We define the predicate $q \in A(x)$ and $r \in x$ (where $r \in R$ or $r = q\top$ for some $q \in Q$) inductively as follows:

$$\frac{}{c \in A(c\bot)} \qquad \frac{q_1 \in A(x_{c_1})}{c_0v_0q_1 \in A(c_0v_0\dot{\{}x_c \mid rc_0v_0c \in Q\})}$$

$$\frac{}{c\top \in c\top} \qquad \frac{r_1 \in x_{c_1}}{c_0v_0r_1 \in c_0v_0\dot{\{}x_c \mid rc_0v_0c \in Q\}} \qquad \frac{}{c_0v_0 \in c_0v_0\dot{\{}x_c \mid rc_0v_0c \in Q\}}$$

We shall only use this predicate for $x : \mathit{strat}(c)$ with c initial, and if x is a collection $\{x_{c_1}, \ldots, x_{c_n}\}$ of such trees (like in the product of two sds's, not recalled here), then we write $q \in A(x)$ for $q \in A(x_{c_i})$ for some i, and similarly for $r \in x$. We write $q \in F(x)$ if $qv \in x$ for some $v \in V$ or if $q\top \in x$. And we write $x <_q y$ when $q \in A(x)$ and $q \in F(y)$.

If $q_1, \ldots, q_n \in A(x)$, we denote by $x[q_1 \leftarrow x_1, \ldots, q_n \leftarrow x_n]$ the strategy obtained by replacing $q_i\bot$ by x_i for all i, in x. We shall abbreviate $q \leftarrow q\bot$ as $q \leftarrow \bot$, and similarly with $\top$. We shall use the convention that writing $x = x[q_1 \leftarrow \bot, \ldots, q_n \leftarrow \bot]$ means that $A(x) = \{q_1, \ldots, q_n\}$.

We define four orders $\leq^s$ (stable) , $\leq^c$ (costable) , $\leq$ (bistable), and $\sqsubseteq$ (extensional) as the congruence closures of, respectively:

for $\leq^s$:

$$\frac{x \in \mathit{strat}(rc)}{c\bot \leq^s x}$$

for $\leq^c$:

$$\frac{x \in \mathit{strat}(rc)}{x \leq^c c\top}$$

for $\leq$:

$$\frac{rc \in Q}{c\bot \leq c\top}$$

for $\sqsubseteq$:

$$\frac{x \in \mathit{strat}(rc)}{c\bot \sqsubseteq x} \qquad \frac{x \in \mathit{strat}(rc)}{x \sqsubseteq c\top}$$

Streicher has remarked that $\sqsubseteq$ and $\leq$ make an sds a bistable biorder. This is an easy checking left to the reader. We extend this here to a full and faithful embedding of the category of sds's and sequential algorithms into the category of bistable and $\sqsubseteq$-monotone maps. For completeness, we include the relevant definitions from [1, 5]:

Definition 1.1 *Given sets $A, B \subseteq A$, for any word $w \in A^*$, we define $w\lceil_B$ as follows:*

$$\epsilon\lceil_B = \epsilon \qquad wm\lceil_B = \begin{cases} w\lceil_B & \text{if } m \in A\backslash B \\ (w\lceil_B)m & \text{if } m \in B\,. \end{cases}$$

Definition 1.2 *Given two sds's $\mathbf{S} = (C, V, P)$ and $\mathbf{S}' = (C', V', P')$, we define $\mathbf{S} \multimap \mathbf{S}' = (C'', V'', P'')$ as follows. The sets C'' and V'' are disjoint unions:*

$$\begin{array}{rcl} C'' & = & C' \cup V \\ V'' & = & V' \cup C\,. \end{array}$$

P'' consists of the alternating positions s starting with a c', and which are such that:

$$\begin{array}{l} s\lceil_{\mathbf{S}'} \in P', (s\lceil_{\mathbf{S}} = \epsilon \text{ or } s\lceil_{\mathbf{S}} \in P), \text{ and} \\ s \text{ has no prefix of the form } scc'. \end{array}$$

Definition 1.3 (exponential – sds) *Let $\mathbf{S} = (C, V, P)$ be a sds. The following recursive clauses define a set $P_!$ of alternating words over $Q \cup R = P$:*

$$\frac{(\mathbf{r} = \epsilon \text{ or } \mathbf{r} \in P_!) \quad q \in A(state(\mathbf{r}))}{\mathbf{q} \in P_!} \qquad \frac{\mathbf{q} \in P_! \text{ and } state(\mathbf{q}r) \in D(\mathbf{S})}{\mathbf{q}r \in P_!}$$

where state is the following function mapping responses (or ϵ) of $P_!$ to strategies of $\mathbf{M}$:

$$state(\epsilon) = \{c\bot \mid c \text{ initial}\} \qquad state(\mathbf{r}qr) = state(\mathbf{r})[q \leftarrow r\{rc\bot \mid rc \in Q\}]$$

(In a more traditional terminology of strategies as sets of positions, we have $state(\epsilon) = \emptyset$ and $state(\mathbf{r}qr) = state(\mathbf{r}) \cup \{r\}$. The sds $(Q, R, P_!)$ is called $!\mathbf{S}$.

Definition 1.4 *Given two sds's $\mathbf{S}$ and $\mathbf{S}'$, the strategies of $!\mathbf{S} \multimap \mathbf{S}'$ are called observably sequential algorithms from $\mathbf{S}$ to $\mathbf{S}'$.*

Sequential algorithms can be composed via a simple abstract machine, which can be quite compactly formulated using the above (new) syntax for strategies. We describe here only the special case of the application of a sequential algorithm to an argument:

Definition 1.5 *Let f be an observably sequential algorithm from* $\mathbf{S}$ *to* $\mathbf{S}'$ *and x an observable strategy of* $\mathbf{S}$. *We define $f \cdot x$ (also written $f(x)$) as the normal form of $\langle\, f \parallel x \,\rangle$ for the following rewriting system, which is easily seen to be confluent (no critical pairs) and terminating (for a quite obvious multiset ordering).*

$$
\begin{array}{lcll}
\langle\, c'v'\{x''_i \mid i \in I\} \parallel x \,\rangle & \to & c'v'\{\langle\, x''_i \parallel x \,\rangle \mid i \in I\} & \\
\langle\, c'\bot \parallel x \,\rangle & \to & c'\bot & \\
\langle\, c'\top \parallel x \,\rangle & \to & c'\top & \\
\langle\, c'q\{x''_r \mid r''c'qr \in Q''\} \parallel x \,\rangle & \to & c'?(\langle\, x''_{r_0} \parallel x \,\rangle) & (r_0 \in x) \quad (\mathit{in}\ \mathit{strat}(r''c')) \\
\langle\, c'q\{x''_r \mid r''c'qr \in Q''\} \parallel x \,\rangle & \to & c'\bot & (q\bot \in x) \quad (\mathit{in}\ \mathit{strat}(r''c')) \\
\langle\, c'q\{x''_r \mid r''c'qr \in Q''\} \parallel x \,\rangle & \to & c'\top & (q\top \in x) \quad (\mathit{in}\ \mathit{strat}(r''c')) \\
\langle\, r_0q\{x''_r \mid r''r_0qr \in Q''\} \parallel x \,\rangle & \to & \langle\, x''_{r_1} \parallel x \,\rangle & (r_1 \in x) \quad (\mathit{in}\ \mathit{strat}(r''r_0)) \\
?(\langle\, r_0q\{x''_r \mid r''r_0qr \in Q''\} \parallel x \,\rangle) & \to & \bot & (q\bot \in x) \quad (\mathit{in}\ \mathit{strat}(r''r_0)) \\
?(\langle\, r_0q\{x''_r \mid r''r_0qr \in Q''\} \parallel x \,\rangle) & \to & \top & (q\top \in x) \quad (\mathit{in}\ \mathit{strat}(r''r_0)) \\
?(\langle\, rv'\{x''_i \mid i \in I\} \parallel x \,\rangle) & \to & v'\{\langle\, x''_i \parallel x \,\rangle \mid i \in I\} & \\
?(\langle\, r\bot \parallel x \,\rangle) & \to & \bot & \\
?(\langle\, r\top \parallel x \,\rangle) & \to & \top &
\end{array}
$$

Observable sequential algorithms can also be defined as observably sequential functions [2, 1].

Definition 1.6 (sequential function (Kahn-Plotkin)) *Let* $\mathbf{S}$ *and* $\mathbf{S}'$ *be two sds's. A $\leq^s$-monotone function $f : D^\top(\mathbf{S}) \to D^\top(\mathbf{S}')$ is called sequential at x if for any $q' \in A(f(x))$ one of the following properties hold:*

1. $\forall\, y \geq^s x \;\; q' \notin F(f(y))$.
2. $\exists\, q \in A(x) \quad \forall\, y > x \;\; (f(x) <_{c'} f(y) \;\Rightarrow\; x <_c y)$.

A query q satisfying condition (2) is called a sequentiality index *of f at (x, q'). The index is called strict if (1) does not hold. If (1) holds, then any q in $A(x)$ is a (vacuous) sequentiality index. If q is a sequentiality index at (x, q') and if moreover $q'\top \in f(x[q \leftarrow q\top])$, then we say that f is observably sequential at x, q', with index q (note that there can then be no other sequentiality index).*

The function f is called sequential (resp. observably sequential) from $\mathbf{S}$ to $\mathbf{S}'$ if it is sequential (resp. observably sequential) at all points.

With the notation introduced above and using the $\leq^s$ monotonicity, we can rephrase the definition more symetrically as follows, at $x = x[q_1 \leftarrow \bot, \ldots, q_n \leftarrow \bot]$:

$$\begin{array}{l} \forall x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n \\ \qquad q' \notin F(x[q_1 \leftarrow x_1, \ldots q_{i-1} \leftarrow x_{i-1}, q_i \leftarrow \bot, q_{i+1} \leftarrow x_{i+1}, \ldots, q_n \leftarrow x_n]) \end{array}$$

$$\begin{array}{l} \forall x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n \\ \qquad q' \in F(x[q_1 \leftarrow x_1, \ldots q_{i-1} \leftarrow x_{i-1}, q_i \leftarrow \top, q_{i+1} \leftarrow x_{i+1}, \ldots, q_n \leftarrow x_n]) \end{array}$$

Or, alternatively, using instead the $\sqsubseteq$-monotonicity (which is established below), we get the following quantifier free definition:

$$q' \notin F(x[q_1 \leftarrow \top, \ldots q_{i-1} \leftarrow \top, q_i \leftarrow \bot, q_{i+1} \leftarrow \top, \ldots, q_n \leftarrow \top])$$

$$q' \in F(x[q_1 \leftarrow \bot, \ldots q_{i-1} \leftarrow \bot, q_i \leftarrow \top, q_{i+1} \leftarrow \bot, \ldots, q_n \leftarrow \bot])$$

Now we recall Laird's definitions:

Definition 1.7 *A bistable biorder is a set $(D, \leq, \sqsubseteq)$ equipped with two partial orders, such that whenever x, y have a lower or an upper bound for $\leq$ then $x \vee y$ and $x \wedge y$ exist (for $\leq$), and then $x \vee y = x \sqcup y$ and $x \wedge y = x \sqcap y$, i.e. the two orders coincide for $\leq$ connected components. Moreover $\wedge$ and $\vee$ distribute over each other in each component. One writes $x \updownarrow y$ when x, y have a $\leq$-upper bound (or equivalently a $\leq$-lower bound). A bistable function is a $\leq^s$-monotone function such that for any $x \updownarrow y$:*

$$\begin{array}{c} f(x \wedge y) = f(x) \wedge f(y) \\ f(x \vee y) = f(x) \vee f(y) \end{array}$$

Note, in view of the sds example of bistable biroders, that the preservation of $\vee$ in the above definition is not a requirement of linearity, as it is taken with respect to $\leq$, not $\leq^s$. In [5] the following theorem is proved:

Proposition 1.8 *The category of bistable biorders and bistable and $\sqsubseteq$-monotone functions is cartesian closed.*

2 Full embedding

In this section, we show that the category of observably sequential algorithms embeds fully in Laird's category. This amounts to the following proposition:

Proposition 2.1 *Given two sds's* $\mathbf{S}$ *and* $\mathbf{S'}$*, a function* $f : D^\top(\mathbf{S})$ *to* $D^\top(\mathbf{S}')$ *is observably sequential if and only if it is bistable and* $\subseteq$*-monotone.*

Proof. The proof that if f is $\sqsubseteq$-monotone and bistable then it is sequential follows the steps of Laird's proof at $\bot$. Suppose that q' is accessible from $f(x)$ and suppose moreover that none of the q_i's is a sequentiality index. Then by $\sqsubseteq$ monotonicity we have that q' is filled in all of the $f(x[q_0,,\ldots,q_{i-1},q_{i+1},q_n \leftarrow \top,\ldots,\top])$. It is filled with the same value, because of the common reference $f(x[q_0,,\ldots,q_n \leftarrow \top,\ldots,\top])$. Now by stability, it should be also filled in $f(x)$ since

$$x = \bigwedge_{i=1\ldots n} x[q_0,,\ldots,q_{i-1},q_{i+1},q_n \leftarrow \top,\ldots,\top]\ .$$

We show that if f is $\sqsubseteq$-monotone and bistable, then it is observably sequential. That is, we have to show that if f has q_0 as sequentiality index at (x,q'), then $q'\top \in f(x[q_0 \leftarrow \top])$. Suppose not. Then, if y is such that $x \leq^s y$ and q' is filled in $f(y)$, we have $y \sqsubseteq y[q_0 \leftarrow \top]$ and $x[q_0 \leftarrow \top] \leq y[q_0 \leftarrow \top]$. Since by extensionality q' is filled in $f(y[q_0 \leftarrow \top])$, there exists a sequentiality index q_1 for f at $(x[q_0 \leftarrow \top],q')$. Continuing in this way, we would exhaust all the $\bot$'s of x. So we get q' filled in, say, $f(x[q_0,q_1,\ldots,q_n \leftarrow \top,\ldots,\top])$, and hence by costability in one of the $f(x[q_i \leftarrow \top])$. But it cannot be an $i > 0$ since q_0 is a sequentiality index at x, hence it is $i = 0$, which contradicts the assumption we have made.

Conversely, if f observably sequential, we first show that f is $\sqsubseteq$-monotone. It is enough to check that if $x \leq^c y$ then $f(x) \leq^c f(y)$. To show this – actually, the natural thing to show is that more generally if $f \leq^c g$ and $x \leq^c y$ then $f(x) \leq^c f(y)$, we extend the definition of $\leq^c$ by congruence to all the terms involved in the rewriting system of Definition 1.5. It is straightforward to show, for each of the rules of the rewriting system, that if $t \leq^c t'$ and $t' \to t'_1$ then there exists t_1 such that $t \to t_1$. Informally, the only difference of behaviour is that $\langle\, g \mid y \,\rangle$ might terminate before $\langle\, f \parallel x \,\rangle$, because of a new $\top$ in g or y. Notice that this argument is the same as the one used in the separation theorem of ludics [4, 3] (and, by the way, designs also form a bistable biorder).

Finally, we show that if f is observably sequential then it is bistable. The argument for stability is standard. One proves actually the preservation of all $\leq^s$ compatible binary $\leq^s$ glb's (for which we use the notation $\wedge^s$). Suppose that $x, y \leq^s z$ and $f(x \wedge^s y) < f(x) \wedge^s f(y)$. Let q' be accessible from $f(x \wedge^s y)$ and filled in $f(x)$ and $f(y)$. Let q be a sequentiality index at $x \wedge^s y, c'$. Then by sequentiality q is filled in both x and y, and with the same value since x, y have a $\leq^s$ upper bound. But then q is also filled in $x \wedge^s y$, contrarily to the assumption. This shows a fortiori the first half of bistability since $x \updownarrow y$ implies a fortiori that x, y have a $\leq^s$-upper bound, and that $x \wedge y = x \wedge^s y$. As for costability, suppose that $f(x) \vee f(y) < f(x \vee y)$. Let q' be accessible from $f(x) \vee f(y)$ and filled in $f(x \vee y)$. Then q' must be already accessible from $f(x \wedge y)$, as the order $\leq$ does not add new requests. Let q be the sequentiality index for q' at $x \wedge y$.Then we have that q is filled in $x \vee y$ by sequentiality, i.e. it is filled in either x or y, say, in x. Since q is accessible from $x \wedge y$, q must be filled with $\top$. But then q' is filled (with $\top$) in $f(x)$, and a fortiori in $f(x \vee y)$, by observable sequentiality: contradiction. Let us remark that it does not seem clear whether f is costable in the sense dual to stable, i.e. whether for all x, y with a $\leq^c$ lower bound one has $f(x \vee^c y) = f(x) \vee^c f(y)$. $\square$

From this proposition, and from the fact that both categories are cartesian closed, we automatically get that $f \leq g$ (resp. $f \sqsubseteq g$) as strategies iff $f \leq g$ (resp. $f \sqsubseteq g$) as bistable functions. Checking this directly may be however reinsuring. For example, we show that if f, g are observably sequential, $f \leq g$, and $x \leq y$, then $g(y) = g(x) \vee f(y)$. Suppose that $g(y) > g(x) \vee f(y)$. Let q' filled in $g(y)$ and accessible from $g(x) \vee f(y)$. It follows that q' is accessible from $g(x)$ and from $f(y)$. The only way for g to make the difference w.r.t. f on input y is to use one of its additional $\top$'s, say, $q'\top$, in the interaction with y. More precisely, one of the $\top$'s of g is responsible for the filling of q'. But the interaction of g with y is the same as with x, as we have here error creation, not error propagation. Hence we must have q' filled in $g(x)$, contrary to our assumption.

References

[1] R. Amadio and P.-L. Curien, Domains and Lambda-calculi, Cambridge University Press (1998).

[2] R. Cartwright, P.-L. Curien, and M. Felleisen, Fully abstract semantics for observably sequential languages, Information and Computation 111 (2), 297-401 (1994).

[3] P.-L. Curien, Introduction to Ludics, notes available from http://www.pps.jussieu.fr/ curien.

[4] J.-Y. Girard, Locus Solum, Mathematical Structures in Computer Science (2001).

[5] J. Laird, Notes on Bistability, draft (2002).

[6] T. Streicher, Laird domains, draft (2002).