Sequentiality, Games and Linear Logic

François Lamarche
(The final version of this paper will contain an additional chapter on variants of the exponentials)

March 25, 1994

Abstract

We construct a category whose objects can be considered as either generalized filiform concrete data structures or as games, for which we give a notion of linear sequential algorithm. This allows us to describe a model of linear logic and to recover the ususal sequential algorithms as the Kleisli category of a comonad.

In versions of the lambda-calculus like PCF, where reduction is deterministic, one can prove an "activity lemma", that says that given a reduced term $t(x_1, \ldots, x_n)$ where the only free variables in t are among $x_1, \ldots, x_n$ then there is a unique i such that "t looks at x_i first". The efforts to mimic this syntactical phenomenon in denotational semantics have given rise to the concept of sequentiality. First one had to come up with a suitable kind of semantical domain, an algebraic poset that has a subdivision in "areas", or *cells*. Such a poset is called a *concrete data structure* or *cds* [11]. The natural notion of sequential function for these domains disappointingly turns out not to pass to higher order, that is, the category of cdses and sequential functions is not cartesian closed. Berry and Curien [2, 8] succeeded in correcting this situation by constructing a category where the morphisms between cdses are more than functions but actually contain some additional, intensional information. These morphisms are called *sequential algorithms*. Bucciarelli-Ehrhard [4] noticed that a cell could be thought of as a linear map to the set $\{\perp, \top\}$ and give a synthetic theory of sequential algorithms for a much wider class of structures than cdses.

In this paper we give a definition for a concept of *linear* sequential algorithm between cdses and get the standard decomposition $X \Rightarrow Y = !X \multimap Y$ of linear logic [9]. We restrict ourselves to what is known as *filiform* cdes, whose structure is simpler than general cdses; actually, in order to model all the connectives of linear logic we have to generalize a little the definition of a filiform cds, to allow for cells that cannot be filled. Our approach to cdses is simpler than the one given in [2, 8] in that the presentations that we come up with are invariant under isomorphism: our generalized cdses are canonical presentations for the objets of the category we construct. In that category we get a model of full linear logic, except that the dereliction rule for the exponentials is not defined for all objects of the category; we show briefly how to correct this defect by enlarging our category. Notice that these enlargements of the concept of cds are not necessary if one does not care about modelling all the connectives of linear logic: in [5] there is a different decomposition $X \Rightarrow Y = !X \multimap Y$, which has the advantage of staying inside the universe of cdses as originally defined, but does not have the necessary symmetry to allow for full linear logic.

The results above were presented at the 1992 CLICS conference in Århus [12]; it is there that we learned that there was another dimension that could be given to this work (of which we were aware, but had not succeeded in formalizing), that our generalised cdses could also be considered as *games*, of the kind used say in set theory [17], and that the "elements" or "states" could be seen as *partial playing strategies* for one player. It then becomes unavoidable to ask if there is a model of linear logic that uses *winning* playing strategies. It turns out that this question has already been given an answer by Blass [3] but there are many possible variations to such models (another one is given by Abramsky-Jagadeesan [1], who make composition associative and get a stronger completeness theorem for the multiplicatives), and that the formalism of our own approach can also be adapted to give a theory of proof nets [14] for *affine* logic, i.e. linear logic with the rule of weakening added. It seems to us that the most distinctive feature of our own work is that it gives a theory of *traces* for game models, which relates them to the more classical field of continuous and (especially) stable semantics for type theories. We find the game-theoretic interpretation very evocative as a paradigm for the study of computation in general and linear logic in particular.

This paper owes a lot to a number of stimulating conversations with Thomas Ehrhard. In particular his (unpublished) description of the function space construction for filiform cdses was the one clue that finally brought the underlying linear structure within reach. An important insight was gained during André Joyal's lecture at CLICS '92 in Århus, when he described his categorical approach to Conway games, a little known, fifteen-year-old piece of work [10]. Finally I would like to thank Pierre-Louis Curien for his enthusiastic support, material as well as moral, and with helping me with suggestions regarding this article.

1 The definitions

A *sequential domain* X is defined by the quadruple $(|X|, \leq, \mathrm{Col}, \frown)$, where

1. $(|X|, \leq)$ is an ordered set (called the *web*, and whose elements we call the *moves*) which is in fact a *forest*: for every $p \in |X|$ the set
$$p{\downarrow} = \left\{ q \in |X| \;\middle|\; q \leq p \right\}$$
is finite and totally ordered. Thus there is a height function which to every move p associates $\mathrm{height}(p) = 1 - \mathrm{Card}(p{\downarrow})$. We denote the subset of all moves of height n by $|X|_n$. For example $|X|_0$ is the set of all minimal elements of $|X|$. It is in bijective correspondence with the set of connected components of $(|X|, \leq)$. Given a move p of height $n > 0$ its *predecessor* p^- is the greatest move $q < p$ and it obviously has height $n - 1$. It is well known that the order structure on a forest can be recovered form the predecessor function and vice versa. Notice that two moves p, q have an inf $p \wedge q$ iff they are in the same connected component. Finally for any $p \in |X|$ we write p_r to denote its *root*: the unique $q \leq p$ of height 0.

2. Col is a function $|X| \to \{\circ, \bullet\}$ which is alternating: $\mathrm{Col}(p^-) \neq \mathrm{Col}(p)$ whenever p^- is defined. A consequence of this is that $\mathrm{Col}(p)$ is determined by $\mathrm{Col}(p_r)$. When $\mathrm{Col}(p) = \bullet$ we say p is a *value*; when $\mathrm{Col}(p) = \circ$ we say it is a *cell*. We denote the set of values by $|X|^\bullet$ and the set of cells by $|X|^\circ$. Thus we can write $|X|^\bullet_3$ to denote the values of height 3, etc..

3. $\frown$ is a coherent domain structure on $|X|_0$. That is, it is a symmetric *anti*reflexive relation on $|X|_0$.

Given a sequential domain $X = (|X|, \leq, \mathrm{Col}, \frown)$ its *orthogonal* $X^\perp$ is the sequential domain $(|X|, \leq, \mathrm{Col} \circ \mathrm{exch}, \smile)$ where exch is the nontrivial permutation on $\{\circ, \bullet\}$ and $\smile$ is the orthogonal coherent domain structure to $\frown$ on $|X|_0$, that is

$$p \smile q \quad \text{iff} \quad p \neq q \quad \text{and} \quad p \not\frown q \quad .$$

If X is as above then an *answer*[1] of X is a subset $x \subset |X|^\bullet$ such that

a) x is down closed:

$$\text{if} \quad p \in x \quad \text{and} \quad p^{--} \quad \text{is defined} \quad \text{then} \quad p^{--} \in x.$$

b) if $p, q \in x$ are in the same component then $\mathrm{Col}(p \wedge q) = \bullet$

c) if $p, q \in x$ are in different components then $p_r \frown q_r$.

Notice that an answer inherits the poset structure on $|X|$ and that this poset turns out to be a forest too.

Proposition 1.1 *In the presence of condition* a) *above, condition* b) *is equivalent to the following:*

b$'$) If $p, q \in x$ are such that $p^- = q^-$ then $p = q$.

Assume a) and b) and let $p, q \in x$ be such that $p^- = q^-$. Then if $p \neq q$ we have that $p \wedge q$ is defined and $p \wedge q = p^- = q^-$. But this is a cell, contradiction. For the converse assume a) and b$'$) and let $p, q \in x$ have an inf $p \wedge q$. If $p \wedge q$ is equal to either p or q there is nothing to prove. If $p \neq p \wedge q \neq q$, assume for a contradiction that $p \wedge q$ is a cell. There is a *value* $p' \leq p$ immediately above $p \wedge q$ (that is $p'^- = p \wedge q$) and a value $q' \leq q$ immediately above $p \wedge q$. By condition a) $p', q' \in x$. Then $p' \wedge q' = p'^- = q'^- = p \wedge q$ and this contradicts b$'$), since $p \neq q$ by assumption. $\square$

[1]This use of the word answer clashes with the one chosen by Abramsky-Jagadeesan, for whom an answer is what we call a value. In subsequent papers we will try to follow the usage that (we hope) will be established by then.

Let x be an answer of X, and $q \in |X|^\circ$ a cell. We say q is *filled in* x if $q \in x\downarrow$. In particular there is $p \in x$ with $p^- = q$; p is the actual value that fills q. Thus we see that the values can be thought of as pieces of information and that an answer is obtained by putting them together in a "consistent" manner; in particular the proposition above says that a cell can only be occupied by a single value at a time.

At a point in the paper we will need to consider sets of the form $x\downarrow$, where x is a value; let us characterize these immediately. We say that a set $u \subset |X|$ is *consistent* if it is downward closed, if $u \cap |X|_0$ is coherent for $\frown$ ($p \neq q \in u \cap |X|_0$ implies $p \frown q$) and if the following holds:

$$p, p' \in u \text{ and } p \neq p \wedge p' \neq p' \quad \Rightarrow \quad p \wedge p' \text{ is a value.}$$

This is the adaptation of condition b) of to the case of sets that are downwards-closed in $|X|$ instead of $|X|^\bullet$. Then $x \subset |X|$ is an answer iff $x\downarrow$ is consistent and all its maximal elements are values, as the reader can easily prove.

A *question* of X is an answer of $X^\perp$. In other words it is a set of cells which is down closed, such that cells in the same component meet over a cell, and cells in distinct components are related by $\smile$.

We denote the poset of answers of X by X_*, the order being inclusion, and its poset of questions by X^*. For the initiates, we should mention that both these posets are dI-domains [6], that is, distributive, prime-generated Scott domains such that the down-set of any compact elements is finite (this last condition along with distributivity implies prime-generatedness). There are obvious embeddings

$$\begin{aligned} |X|^\bullet \to X_* \quad &, \quad p \mapsto p\downarrow \cap |X|^\bullet \\ |X|^\circ \to X^* \quad &, \quad q \mapsto q\downarrow \cap |X|^\circ \end{aligned}$$

and the direct images of these embeddings are exactly the sets of *primes* (also called complete coprimes) of X_*, X^*. Remember an element x of a poset $(X, \leq)$ is prime iff given $A \subset X$ then

$$x \leq \bigvee A \quad \Rightarrow \quad \text{there is } a \in A \text{ with } x \leq a \; .$$

We will sometimes identify a move $p \in |X|$ with the prime it determines in X_* or X^*. For example if $p \in |X|^\bullet$ we will write $p\uparrow$ for the set of all $x \in X_*$ such that $p \in x$. Let X, Y be sequential domains. We recall that a *linear* *map* [9] $f : X_* \to Y_*$ is a monotone function that preserves all pullbacks (finite infs bounded from above) and all the sups that exist. This can be easily shown to be equivalent [13, 16] to the condition

- For every $q \in |Y|^\bullet$ the inverse image $f^{-1}(q\uparrow)$ is the *disjoint* union of a family $(p_i\uparrow)_{i\in I}$ where $p_i \in |X|^\bullet$.

From this we can define the prime trace $\mathrm{TrP}(f)$ of f: it is the set of all pairs $(p, q) \in |X|^\bullet \times |Y|^\bullet$ such that p is minimal in the set $f^{-1}(q\uparrow)$. We also recall that a *stable map* $f : X_* \to Y_*$ is a monotone function that preserves all pullbacks and directed sups. Stability is equivalent to the condition

- For every $q \in |Y|^\bullet$ the inverse image $f^{-1}(q\uparrow)$ is the *disjoint* union of a family $(x_i\uparrow)_{i\in I}$ where $x_i \in X_*$ is finite.

The natural notion of trace for a stable map (which we will call its *ordinary trace*) is the set

$$\mathrm{Tr}(f) \quad = \quad \{\, (x, q) \in X_* \times |Y|^\bullet \mid \; x \text{ is minimal in } f^{-1}(q\uparrow). \;\}$$

It is well known that a linear (resp. stable) map is entirely determined by its prime (resp. ordinary) trace.

A *filiform concrete data structure* is a sequential domain such that

- All the moves of height zero are cells.
- For every cell p there is a value q that fills it: $q^- = p$.
- The coherent structure on the cells is the full structure:

$$p \frown q \quad \text{iff} \quad p \neq q \quad .$$

As an example of a cds we give the lazy natural numbers:

From now on when we say cds we mean filiform cds.

Proposition 1.2 *Let $x \in X_*$ and $\alpha \in X^*$. Then $x\!\downarrow \cap\, \alpha\!\downarrow$ is a path in $|X|$ that is, a totally ordered subset.*

Let $p_1, p_2 \in x\!\downarrow \cap\, \alpha\!\downarrow$. Obviously if $q \leq p_i$ then

$$\begin{array}{lcl} \mathrm{Col}(q) = \bullet & \Rightarrow & q \in x \\ \mathrm{Col}(q) = \circ & \Rightarrow & q \in \alpha \end{array} \quad .$$

Suppose that $p_1 \wedge p_2$ is defined. If $p_1 \wedge p_2$ turned out to be different from both p_1 and p_2, then defining q_1, q_2 by

$$q_i \quad = \quad \text{the move } q_i \leq p_i \text{ immediately above } p_1 \wedge p_2$$

it is obvious that $q_1 \wedge q_2 = q_i^- = p_1 \wedge p_2$, and we would have $\mathrm{Col}(q_1 \wedge q_2) \neq \mathrm{Col}(q_i)$, in contradiction with the definition of either questions or answers. Thus we have to have $p_1 \leq p_2$ or $p_2 \leq p_1$.

The other possibility is that p_1, p_2 are in different components. Obviously there have to be $s_i \in x\!\downarrow \cap\, \alpha\!\downarrow$ with $(s_i)^- = p_{i_r}$ (otherwise p_{i_r} would not be in $x\!\downarrow \cap\, \alpha\!\downarrow \ldots$). Then, whatever the coherence relation is between p_{1_r}, p_{2_r} one can find two moves of the same color in distinct components, whose color contradicts the coherence relation; that is, cells if $\frown$, values if $\smile$. Thus this other possibility cannot happen. □

This proposition is what allows us to interpret a sequential domain in game-theoretical terms. Think of an answer x as partial playing strategy for me, and a question α as a partial playing strategy for my opponent. Then the set $x\!\downarrow \cap\, \alpha\!\downarrow$ is the actual account of the play, seen as an ascending sequence of moves, that happens when the strategies are tested on each other. Notice that who plays first is only decided at match time. These games are very similar to Conway games [7], although the Conway concept of winning (one player wins when the opponent has no move to play in the next turn) is of no use in this particular model; such "total" elements become important in related models [3, 1, 14]. The path $x\!\downarrow \cap\, \alpha\!\downarrow$ should also be thought of as an actual computation, a sequence of atomic events.

2 The Multiplicatives and the Additives

We will go from the concrete to the abstract, by first constructing an object of all linear algorithms between two sequential domains (that is, their linear implication), after which we will give as an exercise an equivalent, more abstract definition for a linear algorithm. Given sequential domains X, Y then the set $|X \multimap Y|_n$ of moves of height n is the set of all *nonempty* lists (sequences)

$$(p_0 q_0)(p_1 q_1) \dots (p_n q_n)$$

of pairs $(p_i q_i) \in |X| \times |Y|$ subject to the additional conditions

1. The pairs of the list can only be of the form $(\bullet\bullet), (\circ\circ)$ or $(\bullet\circ)$. When $(p_n q_n)$ is one of the two former cases, the list is a value. When $(p_n q_n)$ is the latter case, the list is a cell.

2. One can only move up by *one* step up on *one* side. That is,

$$\left(p_{i+1}^{-} = p_i \text{ and } q_{i+1} = q_i\right) \quad \text{or} \quad \left(p_{i+1} = p_i \text{ and } q_{i+1}^{-} = q_i\right) \quad .$$

3. Given $(p_0 q_0) \in |X \multimap Y|_0$ we always have $p_0 \in |X|_0$ and $q_0 \in |Y|_0$.

4. The coherent domain structure is the usual one for implication. That is, given $(pq) \neq (p'q')$ elements of $|X \multimap Y|_0$ we have

$$(pq) \frown (p'q') \quad \text{iff} \quad (p \frown p' \text{ or } p = p') \Rightarrow q \frown q' \quad .$$

The predecessor is obtained by taking off the last item of the list; in other words the order structure on $|X \multimap Y|$ is prefix ordering. Notice that this definition is self-dual in the following sense: given X and Y, if we follow the instructions above but exchange left and right in the pairs, along with cells and values while replacing $\frown$ by $\smile$, we will construct $Y^\perp \multimap X^\perp$ which will be trivially isomorphic to $X \multimap Y$. This self-duality will allow us to cut some proofs by half, as we will see below. We define a *linear sequential algorithm* $X \to Y$ to be an answer $f \in (X \multimap Y)_*$. Let us now give some intuition behind the definition above: an answer f of $X \multimap Y$ is a "linear program" from X to Y, which is obtained by putting "atomic program steps" together, that is, values. As usual in domain theory the order structure on $(X \multimap Y)_*$ is a measure of definition, that is, increasing a program is making it more defined by adding "program steps". Suppose $(p_0 q_0) \dots (p_n q_n)$ is a cell, i.e. $(p_n q_n)$ is $(\bullet\circ)$, and that its predecessor $(p_0 q_0) \dots (p_{n-1} q_{n-1})$ is in f. Then filling this cell is adding a "program step" to our linear sequential algorithm, that tells the algorithm what to do with regard to cell q_n when the input contains value p_n. In general the cell can be filled in two possible ways: either add $(p_n q)$ to the sequence, where $q^- = q_n$, or add (pq_n) where $p^- = p_n$. In the first case the "program step" we are adding is saying "if the input contains value p_n then fill cell q_n with value q". In [8] this was given the more explicit notation *output* q. In the second case we are adding a program step that says "in order to get more information on how to fill cell q_n look in cell p", which was denoted *valof* p in [8].

Given a linear algorithm $f : X \to Y$ define its *trace*

$$f^\sharp \;=\; \left\{ (pq) \in |X| \times |Y| \;\middle|\; \begin{array}{l} (pq) \text{ is either } (\bullet\bullet) \text{ or } (\circ\circ) \text{ and} \\ \text{appears in a sequence } \alpha \in f \end{array} \right\}.$$

Equivalently (since the elements of f are all values) $f^\sharp$ is the set of all (pq) that appear as the last pair of a sequence $\alpha \in f$. The set $f^\sharp$ (more generally any $x \subset |X| \times |Y|$) has a natural order structure as a subset of a product poset.

Proposition 2.1 *The function $\ell : f \to f^\sharp$ that maps a sequence α to its last pair reflects the order structures. As a corollary it is injective (and thus bijective); hence $(f^\sharp, \leq)$ is isomorphic to $(f, \leq)$ and is thus a forest too.*

In other words, given $\alpha, \beta \in f$ such that $\ell(\alpha) \leq \ell(\beta)$ we have to show $\alpha \leq \beta$. If $\alpha = (p_0 q_0) \dots (p_n q_n)$ and $\beta = (r_0 s_0) \dots (r_m s_m)$ our assumption is that $(p_n q_n) \leq (r_m s_m)$. Obviously

$$(p_0 q_0) = (p_{n_r} q_{n_r}) = (r_{m_r} s_{m_r}) = (r_0 s_0)$$

and the sequences have an inf $\alpha \wedge \beta$, which their longest common prefix and that we will write as $(p_0 q_0) \dots (p_i q_i)$, $i \leq n$. Since elements of f meet over values, we know that $(p_i q_i)$ is either $(\bullet\bullet)$ or $(\circ\circ)$. Assume for a contradiction that $\alpha \wedge \beta \neq \alpha$, in other words that $i < n$. Then the pair $(p_{i+1} q_{i+1})$ differs from $(p_i q_i)$ on one side only, and this side is determined by the color of $(p_i q_i)$: left side if $(\circ\circ)$, right side if $(\bullet\bullet)$. Because of duality we can decree without loss of generality that $(p_i q_i)$ is $(\bullet\bullet)$, i.e. $q_{i+1}^- = q_i$ and $p_{i+1} = p_i$. We then also have that $s_{i+1}^- = s_i = q_i$, $r_{i+1} = r_i = p_i$. Since $s_{i+1} \neq q_{i+i}$ (or else the sequence $\alpha \wedge \beta$ would not end in $(p_i q_i) = (r_i s_i)$) we get the desired contradiction, since $q_{i+1} \leq q_n \leq s_m$ and $s_{i+1} \leq s_m$, in other words both q_{i+1} and s_{i+1} belong to the totally ordered interval $[q_i, s_m]$, are directly above q_i but yet they are not comparable. Hence $\alpha \wedge \beta = \alpha$, i.e. $\alpha \leq \beta$. The corollary is trivial to prove. □

Proposition 2.2 *If $f \in (X \multimap Y)_*$ then f is entirely determined by $f^\sharp$.*

We will do a bit more and say exactly, given a subset $x \subset |X| \times |Y|$ made only of $(\bullet\bullet)$ and $(\circ\circ)$ pairs, what the conditions are that x has to satisfy to be of the form $f^\sharp$. Given a pair $(pq) \in x$ (or more generally $(pq) \in |X| \times |Y|$) we will sometimes denote the pair $(p_r q_r)$ by $(pq)_r$, which is not necessarily in x even if (pq) is. We claim $x = f^\sharp$ for some linear algorithm f iff

- L1$^\bullet$: Given $(pq) \in x$ and a value $q' \leq q$ there exists a unique value $p' \leq p$ with $(p'q') \in x$.
- L1°: Dually, given $(pq) \in x$ and a cell $p'' \leq p$ there is a unique cell $q'' \leq q$ with $(p''q'') \in x$.
- L2$^\bullet$: Given $(pq), (rs) \in x$ such that $(pq)_r = (rs)_r$, if $p \leq r$ and p is $\bullet$, then the inf $q \wedge s$ in Y is $\bullet$.
- L2°: Dually if $q \leq s$ and q is $\circ$, $p \wedge r$ is $\circ$.
- L3. If $(pq), (p'q') \in x$ then $(p_r q_r), (p'_r q'_r)$ are coherent for the coherent domain structure defined above.

We will do necessity first; notice that conditions L1, L2 need only one of their halves to be proved, since the duals follow by symmetry. Let f be a linear algorithm, $x = f^\sharp$ and $(pq) \in x$. There is $\alpha = (p_0 q_0) \dots (p_n q_n)$ in f such that $(pq) = (p_n q_n)$. Given a value $q' \leq q$ then obviously, by the one-step-at-a-time condition on α, $q' = q_i$ for some i. Thus p_i will be the required p' of condition L1$^\bullet$: $(p_i q') \in x$. We have to show p_i is uniquely defined: let $p' \leq p$ with $(p'q') \in x$. There are sequences $\gamma, \delta \in f$ such that $\ell(\gamma) = (p_i q')$ and $\ell(\delta) = (p'q')$. Since $(p_i q') \leq (pq) \geq (p'q')$ by the previous proposition $\gamma \leq \alpha \geq \delta$. This implies that either $\gamma \leq \delta$ or $\delta \leq \gamma$, but the situations $\gamma < \delta$, $\delta < \gamma$ cannot happen, because for example $\gamma < \delta$ entails that the sequence δ has the form

$$(p_0 q_0) \dots (p_i q') \dots (p'q')$$

which is clearly impossible (anything between $(p_i q')$ and $(p'q')$ has to be $(\circ\bullet)$). Hence $\gamma = \delta$ and by taking ℓ, we get $(p_i q') = (p'q')$. We have proved L1$^\bullet$.

For condition L2$^\bullet$ let $(pq), (rs) \in x$ have the same roots on both sides and assume that $p \leq r$ and p is a value. If $\alpha = (p_0 q_0) \dots (p_n q_n)$, $\beta =$

$(r_0s_0)\dots(r_ms_m)$ are the sequences in f such that $(pq) = (p_nq_n)$, $(rs) = (r_ms_m)$ we know that the longest common prefix $\alpha \wedge \beta$ is defined; let it be denoted by $(p_0q_0)\dots(p_iq_i)$. Since $\alpha\wedge\beta$ is a value, (p_iq_i) is either $(\bullet\bullet)$ or $(\circ\circ)$; but if it were $(\circ\circ)$ we would have $i < n$, p being $\bullet$, and also $i < m$, since $p_n \leq r_m$. We then would have $q_{i+1} = s_{i+1} = q_i$, thus $p_{i+1} \neq r_{i+1}$ (in order to have $\alpha \wedge \beta$ end with (p_iq_i)), and this is impossible since $p_{i+1}, r_{i+1} \leq r_m$ and $p_{i+1}^- = r_{i+1}^- = p_i$ (which cannot happen in a total ordering, by the same argument as in Proposition 2.1). Therefore (p_iq_i) is $(\bullet\bullet)$. We now want to show $q_i = q\wedge s$, which will show $q\wedge s$ is $\bullet$; if either q_{i+1}, s_{i+1} is undefined this means q_i is either q or s and there is nothing to show; hence assume these two are defined, i.e. $i < m, n$. Obviously $q_i \leq q \wedge s$ but since we already know that $q \geq q_{i+1} \neq s_{i+1} \leq s$, since q_{i+1}, s_{i+1} are immediately above q_i we get that $q_i = q \wedge s$ indeed. This completes the proof of L2$^\bullet$. Notice that $(p_iq_i) = (pq) \wedge (rs)$ in x by Proposition 2.1.

The proof of L3 is obvious.

For the converse let x satisfy conditions L1–L3. Let $(pq) \in x$, and assume it is $(\bullet\bullet)$. We claim that if there is $(rs) < (pq)$ then (pq) has a predecessor (denoted $(pq)^{--}$, for reasons that will soon be obvious) for the order on x. In other words $(pq)^{--}$ has the defining property that

$$(rs) \in x,\ (rs) < (pq) \text{ implies } (rs) \leq (pq)^{--} \quad .$$

Two possibilities may arise:

- $(pq^{--}) \in x$. Then this has to be the predecessor: in other words there cannot be $(rs) \in$ with $(rs) < (pq), (rs) \not\leq (pq^{--})$. Assuming there is such a (rs) then obviously it is (p^-q^-). But by L1° there is a cell $q' \leq q^{--}$ with $(p^-q') \in x$ and we get distinct $(\circ\circ)$ pairs $(p^-q^-), (p^-q')$ and this is impossible by the uniqueness clause of condition L1°.

- $(pq^{--}) \notin x$. Then p^- is defined (otherwise $(pq){\downarrow}$ would be a singleton) and since it is $\circ$ there exists q' with $(p^-q') \in x$. We claim $q' = q^-$; this is because otherwise $q' \leq q^{---}$, hence q^{--} would be defined, and by condition L1$^\bullet$ there would be $p'' < p$ with $(p''q^{--}) \in x$, and since by assumption $p'' \neq p$, the pairs $(p^-q'), (p''q^{--})$ would contradict L2$^\bullet$ (because $p'' \leq p$ is $\bullet$ but $q^{--} \wedge q' = q'$ is $\circ$). Thus $(p^-q^-) \in x$ and we claim it is the predecessor of (pq) in x. This is because if $(rs) < (pq), (rs) \not\leq (p^-q^-)$ then either $r = p$ or $q = s$ and these are impossible:

in the case $r = p$ we can apply condition L1$^\bullet$ to $(p^- q^-)$, $s \leq q^-$ and get $r' \leq p^-$ with $(r's) \in x$, but then $(rs), (r's)$ contradict the uniqueness clause of L1$^\bullet$. In the case $q = s$, the pairs $(pq), (rs)$ also contradict the uniqueness clause (p is the unique $p \leq p$ with $(pq) \in x$!).

If (pq) is $(\circ\circ)$ we get by dualising the proof above that it also has a predecessor in x. Thus for every $(pq) \in x$ that dominates other elements there is a predecessor $(pq)^{--}$ satisfying

$$(pq)^{--} \quad = \quad \begin{cases} (pq^{--}) & \text{if } (pq), (pq)^{--} \text{ are } (\bullet\bullet) \\ (p^{--}q) & \text{if } (pq), (pq)^{--} \text{ are } (\circ\circ) \\ (p^- q^-) & \text{if } (pq), (pq)^{--} \text{ are different colors.} \end{cases}$$

If (pq) has a predecessor then it also has a "half predecessor" $(pq)^-$, not in x, given by

$$(pq)^- \quad = \quad \begin{cases} (pq^-) & \text{if } (pq) \text{ is } (\bullet\bullet) \\ (p^- q) & \text{if } (pq) \text{ is } (\circ\circ) \end{cases}$$

and given any $(pq) \in x$ we can look at the sequence

$$(pq), (pq)^-, (pq)^{--}, ((pq)^{--})^- \ \ldots \ ,$$

stopping only when it is impossible to do so. Let (rs) be the last pair of the sequence which belongs to x and $(r's')$ the last pair of the sequence, period. Thus $(r's') = (rs)$ or $(r's') = (rs)^-$. It is easy to see that if (rs) is $(\bullet\bullet)$ then r has height zero, and s has height ≤ 1 because otherwise we can apply L1. In the same way if (rs) is $(\circ\circ)$ then s has height zero and r has height ≤ 1. But this shows that $(r's') \in |X|_0 \times |Y|_0$ and we get that the reverse of the sequence $(pq), (pq)^-, (pq)^{--} \ldots, (r's')$ is an element of $|X \multimap Y|$. We have thus defined an injective, order-preserving map $x \to |X \multimap Y|$. It is left to show that the image of this map is an answer f of $X \multimap Y$; it will obviously be an inverse to ℓ on f. Going back to the definition of an answer, the closure from below (condition a)) is obvious, and so is the coherence condition c). Thus all is left to prove is condition b$'$) of Proposition 1.1. Clearly we only have to show the following: if $(pq), (rs) \in x$ are such that $(pq)^- = (rs)^-$ then $(pq) = (rs)$. Assume $(pq)^- = (rs)^-$; if both pairs are $(\bullet\bullet)$ then $p = r$, $q^- = s^-$. By condition L2$^\bullet$ we have that $q \wedge s$ is $\bullet$. But this can only happen if $q = r$. If $(pq), (rs)$ are $(\circ\circ)$, just apply the dual of this argument.

Finally if the pairs are different colors, say (pq) is $(\bullet\bullet)$ and (rs) $(\circ\circ)$ then the hypothesis boils down to $q^- = s$, $r^- = p$. But this cannot happen since it contradicts condition L2$^\bullet$: $p \leq r$ is $\bullet$ but $q \wedge s = s$ is $\circ$. This completes the proof. $\square$

Exercise 1 *Let x satisfy the conditions above. If $(pq), (p'q') \in x$ have an inf (rs) in x show that if (rs) is $(\bullet\bullet)$ then $q \wedge q' = s$, and dually if (rs) is $(\circ\circ)$ then $p \wedge p' = r$. The converse is also true, being trivial.*

The following proposition is an extension of condition L2$^\bullet$, although we restrict ourselves to $(\bullet\bullet)$ pairs (the first half of the statement is still true without this restriction).

Proposition 2.3 *Let $f : X \to Y$ be a linear sequential algorithm, and $(p_1q_1), (p_2q_2) \in |X|$ be $(\bullet\bullet)$ pairs such that $(p_1q_1)_r = (p_2q_2)_r$ and $p_1 \wedge p_2$ is a value. Then $q_1 \wedge q_2$ is a value; moreover if $q_1 = q_2$ then $p_1 = p_2$. The dual also holds and is left for the reader to state.*

Let $\alpha_1, \alpha_2 \in f$ be the sequences such that $\ell(\alpha_i) = (p_iq_i)$, and let (pq) be the inf $(p_1q_1) \wedge (p_2q_2)$ in $f^\sharp$. We claim (pq) is $(\bullet\bullet)$. Once again this is proved by contradiction: assume (pq) is $(\circ\circ)$; since $p \leq p_1, p_2$ and $p_1 \wedge p_2$ is $\bullet$ we know that neither α_i ends with (pq); then the pairs immeditately after (pq) in α_i are of the form (r_iq), with $r_i^- = p$, $r_i \leq p_i$. We have to have $r_1 \neq r_2$ (otherwise (pq) would not be the inf $(p_1q_1) \wedge (p_2q_2)$) but this means $r_1 \wedge r_2 = p$ which is a cell, and we also have $r_1 \wedge r_2 = p_1 \wedge p_2$ which is a value, contradiction. Hence (pq) is $(\bullet\bullet)$; if (pq) happens to be one of the (p_iq_i) then $q = q_1 \wedge q_2$ is indeed $\bullet$. Otherwise we do the argument we just did for the other side: there are pairs (ps_i) in α_i immediately after (pq); we have $q_1 \geq s_1 \neq s_2 \leq q_2$ and $s_i^- = q$. Hence $q_1 \wedge q_2 = q$ and is a value.

For the second half assume $q_1 = q_2$. There are for possibilities: $p_1 = p_2$, $p_1 < p_2$, $p_1 > p_2$ and $p_1 \neq p_1 \wedge p_2 \neq p_2$. The second and third possibilities are impossible since they contradict the uniqueness part of L1$^\bullet$. For a contradiction assume the fourth possibility. By condition L2° there are $r_1, r_2 \leq q_i$ with $(p_i^- r_i) \in f^\sharp$. But r_1, r_2 are related and L2° entails that $p_1^- \wedge p_2^-$ is a cell, which is the desired contradiction since $p_1^- \wedge p_2^- = p_1 \wedge p_2$ which is a value by assumtion. $\square$

The assumption that $p_1 \wedge p_2$ is a value amounts to saying that they can appear in the same answer, in other words that they are consistent. Thus the conclusion is natural: q_1 and q_2 also have to be consistent. Condition L2$^\bullet$ in Proposition 2.2 is a simplification of (the first half of) 2.3 which is equivalent to it in the presence of the other L-conditions. A consequence of 2.3, left as an exercise, is that given $f \in (X \multimap Y)_*$ the subset of $f^\sharp$ composed of all the $(\bullet\bullet)$ pairs is the prime trace $\mathrm{TrP}(f_*)$ of a linear map $f_* : X_* \to Y_*$, and its complement the prime trace of a linear map $f^* : Y^* \to X^*$. Actually the requirement of f_* being linear is *equivalent* to the conjunction of conditions L1$^\bullet$, L2$^\bullet$ and L3. Thus in our approach (linear) sequentiality can be seen to be a generalized form of traditional linearity on dI-domains, a "two-way" form of linearity. Thus we could define a sequential algorithm $X \to Y$ as a pair (f_*, f^*) of linear maps whose traces "merge well", the formalization of this notion being as follows:

Exercise 2 *Show that the concrete definition we have given of a linear algorithm $X \to Y$ is equivalent to the following abstract one: a linear algorithm is given by a pair (f_*, f^*) as above subject to the additional conditions:*

- *If $(pq) \in \mathrm{TrP}(f^*)$ and $q' \leq q$ is $\bullet$ then there exists a unique $(p'q') \in \mathrm{TrP}(f_*)$ with $p' \leq p$.*
- *If $(pq) \in \mathrm{TrP}(f_*)$ and $p'' \leq p$ is $\circ$ then there exists a unique $(p''q'') \in \mathrm{TrP}(f^*)$ with $q'' \leq q$.*

Thus the half-condition L1$^\bullet$ can be further split in two: one half for when the pair (pq) is $(\bullet\bullet)$, which is necessary for the linearity of f_*, and the other part for when (pq) is $(\circ\circ)$, which is necessary for the"good merging" of the traces of f_*, f^*; and the same goes for L1°. This discussion suggests how to compose linear sequential algorithms: compose their traces seen as graphs.

Theorem 1 *Given $f \in (X \multimap Y)_*$ and $g \in (Y \multimap Z)_*$ then there exists $gf \in (X \multimap Z)_*$ such that $(gf)^\sharp = g^\sharp \circ f^\sharp$, the latter set being the composition of the graphs.*

We only have to verify conditions L1–L3. For L1$^\bullet$ let $(pr) \in g^\sharp \circ f^\sharp$ and $r' \leq r$ be a value. There is q with $(pq) \in f^\sharp$, $(qr) \in g^\sharp$. Thus there is $q' \leq q$ with $(q'r') \in g^\sharp$, then $p' \leq p$ with $(p'q') \in f^\sharp$, and we get $(p'r') \in g^\sharp \circ f^\sharp$. The value p' is uniquely defined: suppose there is p'' with $(p''r') \in g^\sharp \circ f^\sharp$. Then there is q'' with $(p''q'') \in f^\sharp$ and $(q''r') \in g^\sharp$. Since $p' \leq p \geq p''$, p' and p'' are related (that is, either $p' \leq p''$ or $p'' \leq p'$), and also ${q'}_r = {q''}_r$, because if the roots were different the coherence rule for $|X \multimap Y|$ would force ${q'}_r \frown {q''}_r$ while the coherence rule for $|Y \multimap Z|$ would force the orthogonal relation. Hence we can apply L2$^\bullet$ to $f^\sharp$ and get that $q' \wedge q''$ is a value. Then we can apply the second part of Proposition 2.3 to $(q'r'), (q''r')$ and get that $q'' = q'$, which implies $p'' = p'$ by the uniqueness clause of condition L1$^\bullet$ applied to f.

For L2$^\bullet$ let $(p_1r_1), (p_2r_2) \in g^\sharp \circ f^\sharp$ be such that p_1 is a value, $p_1 \leq p_2$ and ${r_1}_r = {r_2}_r$. We want to show $r_1 \wedge r_2$ is $\bullet$. There are $q_1, q_2 \in |X|$ with $(p_iq_i) \in f^\sharp$, $(q_ir_i) \in g^\sharp$. An argument used just above tells us that ${q_1}_r = {q_2}_r$; thus $q_1 \wedge q_2$ is a value by condition L2$^\bullet$ applied to $f^\sharp$. Then we apply Proposition 2.3 to (q_ir_i) and get the desired result. The duals are proved by symmetry, and L3 is a standard fact of the theory of linear mappings. $\square$

Notice that we immediately get that composition is associative and that the identity morphism on an object X is the answer 1_X of $X \multimap X$ defined by

$$1_X^\sharp = \{(p,p) \mid p \in |X|\} \quad .$$

Thus we do get a category $\mathcal{LA}$. Composition in categories of games such as ours is a dynamical process, very close to cut-elimination in syntax. The reader should consult [1, 10] to get the full illustration of this. It is remarkable that in our case this dynamical process can be recovered from the very static one of composing graphs. In a subsequent paper [14] we will show the exact correspondence between composisition of morphisms in our category and the cut-elimination process for proof nets [9].

Now that the connectives of linear negation and implication have been defined we get the other multiplicatives automatically. Given sequential domains X and Y the sets of moves $|X \otimes Y|, |X \odot Y|$ (this is the symbol we use for "par") are constructed just as above as sets of sequences of pairs in $|X| \times |Y|$ except that the coloring rules and the coherent domain structures are different. The coloring rules are as follows:

- If (pq) is in a sequence of $|X \otimes Y|$ then it is either $(\bullet\circ)$ or $(\circ\bullet)$ (these being cells) or $(\bullet\bullet)$, this being a value.

- If (pq) is in a sequence of $|X \odot Y|$ then it is either $(\bullet\circ)$ or $(\circ\bullet)$ (these being values) or $(\circ\circ)$, this being a cell[2] .

The coherent domain structures on $|X \otimes Y|_0$, $|X \odot Y|_0$ follow the pattern of ordinary coherent domains: first, define

$$\begin{aligned} p \approx q &\quad \text{if} \quad p = q \text{ or } p \frown q \\ p \asymp q &\quad \text{if} \quad p = q \text{ or } p \smile q \end{aligned}$$

Then, given $(pq), (p'q') \in |X \otimes Y|_0$ we have

$$(pq) \approx (p'q') \quad \text{iff} \quad p \approx p' \text{ and } q \approx q'$$

and if $(pq), (p'q') \in |X \odot Y|_0$ instead

$$(pq) \asymp (p'q') \quad \text{iff} \quad p \asymp p' \text{ and } q \asymp q' \quad .$$

The game-theoretical interpretation of these connectives may seem a little bit lopsided: the game $X \otimes Y$ is played on both boards for X and Y at the same time but I can only play on the board that my opponent has just played on, while he can play on any one he please. This makes perfect sense if we think in terms of data structures: an answer is an element of a data type, seen as an answering strategy. That is, the identity of that data element is determined by the way it answers the querying strategies (combinations of atomic questions) of the opponent. Then since a tensor type $X \otimes Y$ is a parallel product whose data elements are pairs $(x, y), x \in X, y \in Y$ where both components have to be used by the receiving device, the natural querying strategies for these pairs are interleaved combinations of strategies for the left and the right component.

There is a simple description of iterated multiplicatives. For example given a finite sequence $X^1, \ldots, X^m$ of sequential domains the web

$$\left| X^1 \otimes \cdots \otimes X^m \right|$$

is in bijective correspondence with the set of all nonfinite lists

$$(p_0^1 p_0^2 \cdots p_0^m)(p_1^1 p_1^2 \cdots p_1^m) \ \cdots \ (p_n^1 p_n^2 \cdots p_n^m)$$

[2] These rules show that our coloring is exactly the so-called Danos-Régnier system of polarities [15] with $\bullet$ = Output, $\circ$ = Input.

of m-uples in $|X^1| \times \cdots \times |X^m|$ subject to the rule that given $i \leq n$ either all the p_i^j are values (in which case the prefix sequence ending in $(p_i^1 \dots p_i^m)$ is a value) or else there is a single p_i^k which is a cell, all the other p_i^j being values. In that case the prefix sequence is a cell. The "move up by only one move at a time" rule translates as

$$p_{i+1}^j = p_i^j \text{ for all } j \text{ except a unique } k \text{ where } (p_{i+1}^k)^- = p_i^k \quad .$$

The coherence rule is as expected: given

$$(p^1 \dots p^m), (q^1 \dots q^m) \in \left|X^1 \otimes \cdots \otimes X^m\right|_0$$

then

$$(p^1 \dots p^m) \approx (q^1 \dots q^m) \quad \text{iff} \quad p^j \approx q^j, \text{ all } j \quad .$$

For example this makes tivial to check associativity for $\otimes$ and $\odot$. One can mix the multiplicatives in this iterated construction, but the coloring rules can become tricky. The unit $\mathbf{I}$ of the tensor is defined by $|\mathbf{I}| = \{\bullet\}$ while the dualizing object $\perp$ has $|\perp| = \{\circ\}$. It is trivial that

$$\mathcal{LA}(\mathbf{I}, X) \cong X_* \quad \text{and} \quad \mathcal{LA}(X, \perp) \cong X^* \quad .$$

Given associativity the natural isomorphism $\mathcal{LA}(X \otimes Y, Z) \cong \mathcal{LA}(X, Y \multimap Z)$ is trivial to prove, since

$$\begin{aligned} \mathcal{LA}(X \otimes Y, Z) &\cong ((X \otimes Y) \multimap Z)_* \cong ((X \otimes Y)^\perp \odot Z)_* \cong (X^\perp \odot Y^\perp \odot Z)_* \\ &\cong (X \multimap (Y \multimap Z))_* \cong \mathcal{LA}(X, Y \multimap Z) \quad . \end{aligned}$$

The construction of the additives should look familiar: given sequential domains X, Y their product $X \& Y$ and coproduct $X + Y$ share the same web

$$|X \& Y| = |X + Y| = |X| + |Y| \ ,$$

the disjoint sum of the posets. The difference in the operations is in the coherent structure. For both additives the coherent structure is the same as for the operands on the subsets $|X|_0 \subset |X|_0 + |Y|_0$ and $|Y|_0 \subset |X|_0 + |Y|_0$, while given $p \in |X|_0$ and $q \in |Y|_0$ we have $p \frown q$ always in $|X \& Y|_0$ and $p \smile q$ always in $|X + Y|_0$. It is trivial to prove these are indeed the product and coproduct; for example given linear algorithms $f{:}W \to X$ and $g{:}W \to Y$ then the trace of $\langle f, g\rangle : W \to X \& Y$ is the union of $f^\sharp$ and $g^\sharp$. The terminal and coterminal object are identical: they are the unique structure you can put on the empty web.

3 The Exponentials

Let X be a sequential domain, $p \in X_*$, $q \in |X|^\circ$. We say that q is *enabled by* x, and write $x \vdash q$, if $q \notin x\downarrow$ and either $q^- \in x$ or q has height zero and $\{q\} \cup (x\downarrow \cap |X|_0)$ is a coherent set. In other words, "if there were p with $p^- = q$ then the set $x \cup \{p\}$ would still be an answer". Given X we now define its domain of repetitions $!X$ as follows. First

$$|!X|_0 = \{u \subset |X|_0^\bullet \mid u \text{ is finite and coherent}\}$$

and it is given the standard coherent domain structure:

$$u \frown v \quad \text{iff} \quad u \neq v \text{ and } u \cup v \text{ is coherent} \quad .$$

For example, if X is a cds $|!X|_0$ has a unique element $\emptyset$. All the moves in $|!X|_0$ are decreed to be values; as a consequence, all moves of the same height n in $|!X|$ have the same color: values for n even, cells for n odd. For $n > 0$ a move $p \in |!X|_n$ is a list $(p_0 p_1 \ldots p_n)$ where $p_0 \in |!X|_0$ and $(p_1 \ldots p_n)$ is a color-alternating list of moves $p_i \in |X|$, starting with p_1 a cell, subject to the (slightly redundant) conditions:

- for every even $i \leq n$ the set $p_0 \cup \{p_2, p_4 \ldots, p_{i-2}, p_i\}$ is an answer and $p_i \notin p_0 \cup \{p_2, p_4 \ldots, p_{i-2}\}$.
- for every odd $i \leq n$ we have $p_0 \cup \{p_2, p_4 \ldots, p_{i-3}, p_{i-1}\} \vdash p_i$.
- $p_i^- = p_{i-1}$ for $i \geq 2$ even.

Given $p \in |!X|$ just as above define

$$\overline{p} = p_0 \cup \{p_1, \ldots, p_n\} \subset |X| \quad .$$

Note that if $n \geq 1$ the last element of the sequence $(p_0 \ldots p_n)$ is of the same color as p. If $p \in |!X|$ is a value it can be thought of as a finite answer $\overline{p} \cap |X|^\bullet$ along with a "seed" subanswer $p_r = p_0 \subset \overline{p}$ of height zero and a bijective path $p_2, p_4 \ldots p_n$ of $\overline{p} - p_r$ which is always consistent. The predecessor function is as usual: remove the last item of the list; in other words the ordering is prefix ordering.

We define a *sequential algorithm* $f : X \to Y$ to be an answer $f \in (!X \multimap Y)_*$. The difference between a general sequential algorithm an a linear one should be clear: when adding a "program step" to a general sequential algorithm we have the right to explore a cell which is not necessarily directly above the last value we have looked at; a new cell we explore only has to be enabled by the values that have been previously encountered.

Define an embedding $|X| \to |!X|$ as follows, for $p \in |X|$ (by p_1 we denote the unique move $p_1 \leq p$ such that $p_1^- = p_r$, if it exists)

$$\underline{p} \quad = \quad \begin{cases} (\{p_r\}p_1 \dots p^{--}p^-p) & \text{if } p_r \text{ is } \bullet \\ (\emptyset p_r p_1 \dots p^{--}p^-p) & \text{if } p_r \text{ is } \circ \end{cases}$$

The function $p \mapsto \underline{p}$ obviously preserves the predecessor function.

Proposition 3.1 *The graph*

$$\varepsilon_X^\sharp \quad = \quad \Big\{(\underline{p}\, p) \,\Big|\, p \in |X|\Big\}$$

determines a linear algorithm $\varepsilon_X : !X \to X$ iff the sequential domain X has the property that for every $p, q \in |X|_0$

$$p \smile q \textit{ implies both } p, q \textit{ are } \bullet \quad .$$

When this happens we say X is acceptable.

Assume ε_X is a linear algorithm and let $p \smile q$. If one of p, q, say p is $\circ$ then $\underline{p} = (\emptyset)$, thus $((\emptyset)p) \in \varepsilon_X^\sharp$ and since $\emptyset \frown \underline{q}$ always this contradicts the coherence rule for linear implication. The converse is proved by verifying that ε_X is indeed a linear algorithm when X is acceptable. $\square$

Now if ε_X is a linear algorithm it determines a sequential algorithm $X \to X$, which unsurprisingly will turn out to be the identity in the universe of general sequential algorithms. We will use the notation $\varepsilon_X^\sharp$ for the graph defined above, even if ε is not a linear algorithm. We will show how to define composition for *any* composable pair of linear algorithms, but as we have shown above the identity algorithm will exist only on acceptable objects[3].

[3]The following generalization allows for the construction of a category where ! is a true comonad: given a generalized sequential domain X the coherence relation $\frown$ is now required to be defined for all sets of the form $\{q \mid q^- = p\}$ for *all* $p \in |X|$. Thus an ordinary sequential domain is one for which $\frown$ is the full relation when p is a value, end the empty one when p is a cell. An answer of a generalized domain id defined as a down-closed set x all whose maximal elements are values and which is coherent in the sense that if $p, q \in x$ are such that $p^- = q^-$ then either $p = q$ or $p \frown q$; thus this coherence condition is the natural generalization of b') (and c)). The theory given in this paper (in particular the S- and L-conditions) can be adapted to allow for these more general domains.

Thus we are defining something which is a little less than a full category, not having all identity morphisms, but that nonetheless will have the great majority of the features of the Kleisli category for a "comonad", whose counit is only partially defined.

Subsets of the form $\overline{p}$ ($p \in |!X|$ a cell of a value) play an important role in what follows. Let us define the set of all such things:

$$X_\natural \quad = \quad \left\{ x{\downarrow} \subset |X| \;\middle|\; \begin{array}{l} x \text{ is a finite answer or } x = y \cup \{q\}, \\ y \text{ a finite answer and } q \text{ a cell with } y \vdash q \end{array} \right\} \quad .$$

Let $u \in X_\natural$. We say that it is *value-like* (or just say that it is $\bullet$) if $u = x{\downarrow}$ for a finite answer x, in other words if u is a finite consistent set (recall that the definition of *consistent* is in the first chapter) all whose maximal elements are values. Otherwise, if u is finite and consistent and exactly one of its maximal elements is a cell, we say it is *cell-like*, or that it is $\circ$. Also we define the *root* of $u \in X_\natural$ to be the value-like subset $u_r = u \cap |X|_0^\bullet$. Notice that we always have $\overline{p}_r = p_r$ for $p \in |!X|$; this is because if $p = (p_r p_1 \ldots p_n)$ then the p_i, $i \geq 1$ cannot be elements of $|X|_0^\bullet$. Given $p \in |!X|$, if we think of $\overline{p}$ as the "underlying set" of p, we can turn the tables and thing of p as a *structure of path* over the consistent set $\overline{p}$.

The right order structure on $X_\natural$ is as follows: given $u, v \in X_\natural$ we write $u \leq v$ when $u \subset v$ *and* $u_r = v_r$. Note that a cell-like set $(u \cup \{q\})$, q the maximal cell, has a predecessor u for that ordering, but otherwise the order structure is much weaker than that of a forest: infs do not exist in general (for a counterexample construct two cell-like sets u, u' such that $u \cap u'$ has two maximal cells). There are some forest-like features, though. For example $(X_\natural, \leq)$ is well-founded: $u{\downarrow}$ is always finite. Also every connected component of $X_\natural$ has a bottom: if u, v are in the same component (if they can be joined by a zigzag) then $u_r = v_r$, this obviously being the bottom of their component.

Let $f \in (!X \multimap Y)_*$. Thus an element α of f is a sequence $(p_0 q_0) \ldots (p_n q_n)$ where $p_i \in |!X|$, $q_i \in |Y|$. Let

$$\iota(\alpha) = (\overline{p_n} q_n) \in X_\natural \times |Y| \quad .$$

We then define the *trace* of f to be

$$f^\natural \quad = \quad \{\, \iota(\alpha) \mid \alpha \in f \} \quad .$$

$f^\natural \subset X_\natural \times |Y|$ inherits the product order structure.

Proposition 3.2 *If $f \in (!X \multimap Y)_*$ is a sequential algorithm then the function $\iota : f \to f^\natural$ reflects the order on $f^\natural$. In particular ι is an isomorphism for the order structures.*

Let $\alpha, \beta \in f$ be such that $\iota(\alpha) \leq \iota(\beta)$. If $\alpha = (p_0 q_0) \ldots (p_n q_n)$ and $\beta = (r_0 s_0) \ldots (r_m s_m)$ our assumption is that $(\overline{p_n} q_n) \leq (\overline{r_m} s_m)$. From the remark above on connected components we get that $\overline{p_n}_r = \overline{r_m}_r$ and since $\overline{p_n}_r = p_{nr} = p_0$, $\overline{r_m}_r = r_{mr} = r_0$ (because $p_0, r_0 \in |X|_0$) we get $p_0 = r_0$, while the equation $q_0 = s_0$ is now a standard result. Thus the sequences α, β have a longest common prefix $\alpha \wedge \beta$, which can be written as $(p_0 q_0) \ldots (p_i q_i)$, $i \leq n$. Assume for a contradiction that $\alpha \wedge \beta \neq \alpha$, equivalently $i < n$. We know that $(p_i q_i)$ is either $(\bullet\bullet)$ or $(\circ\circ)$. In the first case the proof is just like 2.1. In the second case p_i is a sequence $(p_{i0} p_{i1} \ldots p_{ik})$ of moves $p_{ij} \in |X|$, and p_{ik} is a cell. By assumption and the color rules for $\multimap$ we have that $p_{i+1} \neq q_{i+1}$ and this implies that the last (in fact $k + 1^{\text{th}}$) moves in the sequences p_{i+1}, q_{i+1}, call them p, q, are different values. But this means that we have values $p \neq q \in \overline{r_m} \in X_\natural$ such that $p^- = q^-$, showing $\overline{r_m}$ is inconsistent, the desired contradiction. Hence $\alpha \leq \beta$. $\square$

Here is an immediate corollary of this:

Proposition 3.3 *A sequential algorithm $f : X \to Y$ is entirely determined by its trace $f^\natural$.*

Let us first give a quickie proof of this result, which will immediately be superseded by one that contains more information. Our claim is that if f is as above and $(\overline{p}q) \in f^\natural$ then p can be recovered by examining the structure of $f^\natural$. Look at the set

$$A \quad = \quad \{u \in X_\natural \mid \text{there exists } q' \text{ with } (uq') \in f^\natural \text{ and } (uq') \leq (\overline{p}q)\} \quad .$$

By the previous proposition this set, with the order induced from $X_\natural$, will be isomorphic to a subset of $p\!\downarrow$, thus a total ordering. In fact if $u \leq v \in A$ are such that there in nothing in A strictly between them then the proof of Proposition 2.2 tells us that only two things can happen: either $v - u$ is a singleton (meaning they are different colors) or $v - u$ has two elements (meaning they are both cell-like). In the second case it is trivial to find the value-like w in between, and then we can construct a filtration

$$\overline{p}_r = u_0 \subset u_1 \cdots \subset u_n = \overline{p}$$

where $u_i - u_{i-1}$ is a singleton $\{p_i\}$, and we see that $p = (u_0 p_1 \dots p_n)$, which proves our claim. $\square$

This result allows us to define Kleisli composition for sequential algorithms; if $f : !X \to Y$ and $g : !Y \to Z$ are linear algorithms, define $g * f : !X \to Z$ by

$$(g * f)^\natural = \left\{ (ur) \,\middle|\, \begin{array}{l} \text{there exist } v \text{ and a family } (u_q)_{q \in v} \text{ with } u = \bigcup_q u_q \\ \text{and } (u_q q) \in f^\natural,\ (vr) \in g^\natural,\ u, v \text{ the same color} \end{array} \right\}$$

Naturally, in order to prove that the graph thus obtained is the trace of a sequential algorithm we need explicit conditions on trace sets:

Proposition 3.4 *Let $x \subset X_\natural \times |Y|$ be a set whose elements are all $(\circ\circ)$ or $(\bullet\bullet)$ pairs. Then $x = f^\natural$ for a sequential algorithm f iff the conditions below hold.*

- S1$^\bullet$. Given $(uq) \in W$ for every value $q' \leq q$ there is a unique $u' \leq u$ with $(u'q') \in W$.
- S1°. Given $(uq) \in x$ for every cell $c \in u - u_r$ there is a unique $(vq') \in x$ such that $v \leq u$ is cell-like and c is its maximal cell, and $q' \leq q$.
- S2$^\bullet$. Given $(u_1 q_1), (u_2 q_2) \in x$ such that $(u_1 q_1)_r = (u_2 q_2)_r$, if $u_2 \leq u_1$ and u_2 is $\bullet$, then the inf $q_1 \wedge q_2$ in Y is $\bullet$.
- S2°. Dually if $q_2 \leq q_1$ and q_2 is $\circ$, then the set $u_1 \cap u_2$ has a maximal cell c such that u_i fills c iff $u_i \neq u_1 \cap u_2$.
- S3. Given $(uq), (u'q') \in x$, the pairs $(u_r q_r), (u'_r q'_r)$ are coherent in the coherent domain $|!X \multimap Y|_0$.

Notice that the numbering is consistent with 2.2, despite the loss of perfect self-duality.

The proof of necessity will be left to the reader as an exercise (hint: Props. 2.3 and 3.2 are more than handy). The proof of sufficiency is a corollary of a more general result that is to follow soon. $\square$

The next proposition is the equivalent of Proposition 2.3 for general sequential algorithms.

Proposition 3.5 *Let $x \subset X_\natural \times |Y|$ satisfy conditions S1–S3 of 3.4 and $(u_1q_1), (u_2q_2) \in x$ be $(\bullet\bullet)$ pairs such that $u_1 \cup u_2$ is consistent and $q_{1_r} = q_{2_r}$. Then $q_1 \wedge q_2$ is a value. Moreover if $q_1 = q_2$ then $u_1 = u_2$.*

First assume $q_1 = q_2$. Since $u_{1_r} \cup u_{2_r}$ is a coherent set by the coherence rule for $\multimap$ we have to have $u_{1_r} = u_{2_r}$. Then there are four possibilities: $u_1 = u_2$, $u_1 < u_2$, $u_1 > u_2$ and $u_1 \neq u_1 \cap u_2 \neq u_2$. The second and third possibilities are impossible because of the uniqueness condition of $S1^\bullet$. Assume the fourth one; this means there are cells $c_i \in u_i$ with $c_i \notin u_1 \cap u_2$. By S1° there are $(w_ir_i) \in x$ such that c_i is the maximal cell of w_i and $r_i \leq q_i$. Then r_1, r_2 are related and by S2° $u_2 \cap u_2$ has a maximal cell c' which is distinct from both c_1 and c_2. But this means u_1, u_2 fill the cell c' with different values (or else c' would not be maximal in $u_1 \cap u_2$), contradicting the consistency of u. Therefore $u_1 = u_2$ is the only possibility.

We can now prove the first part of the statement. If u_1, u_2 are related we can apply condition $S2^\bullet$ to get the desired conclusion. Hence here again the only possibility worth examining is $u_1 \neq u_1 \cap u_2 \neq u_2$. It is easy to see $u_1 - u_1 \cap u_2$ contains some cells and by S2° there is a cell $r_1 \leq q_1$ along with a cell-like w_1 such that $(w_1r_1) \in x$, w_1's maximal cell is in $u_1 - u_1 \cap u_2$, and moreover c_1 is the least cell below q_1 with such a w_1. By symmetry we get $c_2 \leq q_2$ minimal with w_2 whose maximal cell is in $u_2 - u_1 \cap u_2$. Obviously $w_1 \cup w_2$ is consistent, and therefore by the result just proved we cannot have $r_1 = r_2$ since it would imply $w_1 = w_2$, a contradiction. But this shows

$$q_1 \wedge q_2 = r_1 \wedge r_2 = r_1^- = r_2^-$$

which is a value, because if $r_1 \wedge r_2$ were any smaller the r_i would not be the minimal cells with their defining properties. $\square$

Proposition 3.6 *Let $x \subset X_\natural \times |Y|$ satisfy conditions S1–S3. Then there exists a linear algorithm $\Psi_x : !X \to !Y$ such that the set*

$$\left\{(pq) \mid q \in |Y|, (p\underline{q}) \in \Psi_x^\sharp\right\}$$

is the trace $f^\sharp$ of a linear algorithm $f : !X \to Y$ such that $f^\natural = x$.

Thus, from a (co)monadological point of view, Ψ_x turns out to be the linear algorithm $(!f)\delta_X : !X \to !Y$, and conversely $f^\sharp$ can be thought of as the composite graph $(\varepsilon_Y^\sharp) \circ \Psi_x$ which is always defined, even if $\varepsilon_Y^\sharp$ does not define an algorithm. We first extend x to $x^\dagger \subset X_\natural \times Y_\natural$ by

$$x^\dagger \quad = \quad \left\{ (uv) \left| \begin{array}{l} v \in Y_\natural \text{ and there is a } v\text{-indexed family } (u_q)_{q \in v} \\ \text{of elements } u_q \in X_\natural \text{ such that } (u_q q) \in f^\natural \text{ and} \\ u = \bigcup_q u_q \in X_\natural \text{ with } u, v \text{ the same color.} \end{array} \right. \right\} .$$

Before we can prove 3.6 we need the following.

Proposition 3.7 *Let $v \in Y_\natural$ and $u \subset |X|$ be a consistent set such that there is a family $(u_q)_{q \in v}$ with $(u_q q) \in x$ such that $u = \bigcup_q u_q$. Let $(wq) \in x$ be a $(\circ\circ)$ pair such that $w \leq u$, $q \in v$, and let c be the maximal cell of w. Then if the cell q is filled in v, c is filled in u.*

Assume that q is filled, i.e. that there is $q' \in v$ with $q'^- = q$. We claim that $w \subset u_{q'}$; for a contradiction assume it is not the case, i.e. that $w \cap q_{q'} \neq w$. We know by condition S2° that $w \cap u_{q'}$ has a maximal cell c' and S2° also means that c' is filled by w. But then $u \notin X_\natural$ because it is an inconsistent set since the value in w immediately above c' is different from the value in $u_{q'}$ immediately above c' (otherwise c' would be filled in $w \cap u_{q'}$) and $w, u_{q'} \subset u$. Therefore $w \subset u_{q'}$, which implies $c \in u_{q'}$ and c is filled in u, since $u_{q'}$ is value-like. $\square$

As a consequence of this, if v is value-like then u also has to be value-like, since it is consistent by assumption. Hence if $u \in X_\natural$ as well as $v \in Y_\natural$, if u is cell-like v has to be cell-like too, trivially; moreover if c is u's maximal cell and q is v's, then c has to be u_q's maximal cell, since there is not other way for c to be left unfilled. This also shows that if v is value-like then

$$u = \bigcup_{q \in v \cap |X|^\bullet} u_q \quad ,$$

and if v is cell-like

$$u = \bigcup_{q \in (v \cap |X|^\bullet) \cup \{c\}} u_q$$

c being the maximal cell of v.

Proposition 3.8 *Let $x \subset |!X| \times |Y|$ satisfy the S-conditions, $(w_1 q)(w_2 q) \in x$ be such that $w_1 \cup w_2$ is consistent. Then either $w_1 \leq w_2$ or $w_2 \leq w_1$.*

The roots of w_1, w_2 are equal because of the coherence rule for $\multimap$. It cannot be that $w_1 \neq w_1 \cap w_2 \neq w_2$ because as in the proposition above (using S2° again) $w_1 \cap w_2$ would have a maximal cell that would be filled by two different values, contradicting the consistency of $w_1 \cup w_2$. $\square$

We can now start the proof of 3.6. $x \subset |!X| \times |Y|$ is assumed to satisfy the S-conditions. Let $(uv) \in x^\dagger$ (with $(u_q)_{q \in v}$ as above) be such that v is of the form $\overline{q}$ for $q \in |!X|$. That is, there is a bijective enumeration $q_1, \ldots q_n$ of the set $v - v_r$ such that $q = (v_r q_1 \ldots q_n)$. If c is a cell of u by definition there is i with $c \in u_{q_i}$, hence by 1° there is j and w cell-like with $(wq_j) \in x$ and c the maximal cell of w. There might be more than one of these pairs (wq_j); let $[u,q]c$ denote the least j for which such a pair exists. Thus for every cell $c \in u$ there is also a uniquely defined w^c such that $(w^c q_{[u,q]c}) \in x$ and c is w^c's maximal cell. Define

$$\langle u, q \rangle c = w^c \cup \bigcup_{i \leq [u,q]c} u_{q_i}$$

and the ordering on cells of u

$$c \preceq c' \quad \text{iff} \quad \langle u, q \rangle c \subset \langle u, q \rangle c' \,. \tag{1}$$

We claim this is a *total* ordering. This is because if $[u,q]c < [u,q]c'$ then trivially (by way of 3.7) $\langle u,q \rangle c \subset \langle u,q \rangle c'$ and they are different sets. On the other hand if $[u,q]c = [u,q]c'$ then $\langle u,q \rangle c$ and $\langle u,q \rangle c'$ differ only by w^c, $w^{c'}$ and by 3.8 we have to have either $\langle u,q \rangle c \subset \langle u,q \rangle c'$ or $\langle u,q \rangle c' \subset \langle u,q \rangle c$. We can extend the function $\langle u,q \rangle$ to the whole of $u - u_r$, for if p is a value we can define

$$\langle u,q \rangle p = \langle u,q \rangle p^- \cup \{p\} \,.$$

It is easy to see that the order obtained on the full set $u - u_r$ by applying definition (1)is still total, since u is a consistent set and a cell can be filled in only one way. Out of this total ordering of the cells we get a move $p = (p_0 \ldots p_m)$ in $|!X|$ such that $\overline{p} = u$: for p_0 take u_r; then the list $p_1 \ldots p_m$ is simply the enumeration of $u - u_r$ in the ascending $\preceq$-order.

Thus, given $(uv) \in x$ and a path structure $q \in |!Y|$ on q we have constructed a path structure $p \in |!X|$ on u. We say that p was obtained by applying the *Ordering Process* on $(u, v), q$. The Ordering Process has an important property:

Proposition 3.9 *Let p be obtained by applying the Ordering Process on $(uv) \in x$, $q \in |!Y|$; in particular $\overline{q} = v$. Let $q' \leq q$ and $u' \leq u$ be such that $(u'\overline{q'}) \in x$. Then applying the Ordering Process on $(u\overline{q'})$ yields a move $p' \leq p$.*

The proof is an easy consequence of the obvious observation that the function $[u', q']$ is $[u, q]$ restricted to u', and thus the function $\langle u', q' \rangle$ is also the restriction of $\langle u, q \rangle$ to u. $\square$

We can now define $\Psi_x^\sharp \subset |!X| \times |!Y|$: it is the set of all pairs $(pq) \in |!X| \times |!Y|$ such that $(\overline{p}\,\overline{q}) \in x^\dagger$ and p is obtained from $(\overline{p}\,\overline{q})$ and q by Ordering Process. We have to prove $\Psi_x^\sharp$ satisfies condiditions L1–L3.

First condition L1$^\bullet$. Let $(pq) \in \Psi_x^\sharp$ and $q' \leq q$ be $\bullet$. Let $q = (q_0 q_1 \ldots q_n)$; thus $q' = (q_0 q_1 \ldots q_m)$ for some $0 \leq m \leq n$, where $q_0 = q_r$, and $\overline{p} = \bigcup_i (u_i q_i)$ for a family of pairs $(u_i q_i) \in x$. Let

$$u' = \bigcup_{i \leq m} u_i \quad .$$

We can apply the Ordering Process on $(u'\overline{q'})$, q' and get $p' \in |!X|$ and the previous proposition tells us $p' \leq p$. We have to show p' thus defined is unique. Given $r \leq p$ with $(rq') \in \Psi_x^\sharp$, we know by definition of $\Psi_x^\sharp$ that there is a family $(v_i)_{i \leq m}$ with $(v_i q_i) \in x$ such that $\overline{r} = \bigcup_i v_i$, and then since $u_i, v_i \leq \overline{p}$ we can apply 3.5 and get $u_i = v_i$ whenever q_i is a value, proving (by the remark right after 3.7) that

$$\overline{r} = \bigcup_{i \leq m,\, q_i \text{ is } \bullet} v_i = \bigcup_{i \leq m,\, q_i \text{ is } \bullet} u_i = \overline{p'} \quad ,$$

and the result $r \leq p'$, $p' \leq r$ follows by 3.9 applied twice.

Now for L1°: if (pq) is as in the paragraph above, with $q = (q_1 \ldots q_n)$ let $p'' \leq p$ be a cell in $|!X|$. Hence $p = (p_0 \ldots p_l)$ for some l and p'' is the prefix $(p_0 \ldots p_j)$, with p_j a cell. By condition S1° there exists a unique cell q_i and unique cell-like $w \leq \overline{p}$ having p_l as its maximal cell with $(wq_j) \in x$. Clearly we have $\overline{p''} = \{w\} \cup \bigcup_{i<l} u_i$. Thus if we take $q'' = (q_0 \dots q_j)$, Proposition 3.9 tells us that p'' can be obtained by applying the Ordering Process to $(\overline{p''}\,\overline{q''})$, q'' and thus $(p''q'') \in \Psi_x$. It is left to show that q'' is uniquely defined and this will be left to the reader.

For condition L2$^\bullet$ let now $(p_1q_1), (p_2q_2) \in \Psi_x^\sharp$ have common roots and $p_1 \leq p_2$, with p_1 value-like. Our goal is to prove that the greates common prefix $q_1 \wedge q_2$ is a value; let $q_i = (q_{i,0} \dots q_{i,n_i})$, for $i = 1, 2$ and let $u_{i,j} \in X_\natural$ be such that $(u_{i,j}q_{i,j}) \in x$ and $\overline{p_i} = \bigcup_{j \leq n_i} u_{i,j}$. Let also $r = (r_0 \dots r_m)$ be the inf $q_1 \wedge q_2$; our goal is equivalent to showing r_m is a value in $|Y|$. Obviously $r_j = q_{1,j} = q_{2,j}$ for $j \leq m$. If there is no r_j which is a cell, then we are done since $r = (r_0)$ is a value. Otherwise let r_k be the last cell in r, in particular k is either m or $m-1$. The value $q_{1,k+1}$ is defined because q_{1,n_1} is a value, forcing $q_{1,k} = r_k < q_{1,n_1}$ to be filled. The value $q_{2,k+1}$ is also defined because otherwise we would have the cell $q_2 = r \leq q_1$, which is impossible because by condition S2° it entails that $\overline{p_1} \cap \overline{p_2}$ has a maximal cell, contradicting our assumption. We claim that $q_{1,k+1} = q_{2,k+1}$, which shows that $m = k+1$ and r is a value. But this follows from the fact that $u_{1,k+1} \cap u_{2,k+1}$ is consistent, being a subset of $\overline{p_2}$, and then by Proposition 3.5 $q_{1,k+1} \wedge q_{2,k+1}$ is a value, which forces $q_{1,k+1} = q_{2,k+1}$.

For L2°: let (p_iq_i), $(u_{i,j})$ be just as above but this time we assume q_1 is a cell and $q_1 \leq q_2$; now we want to show the prefix $p_1 \wedge p_2$ is a cell of $|!X|$. Let $Q = \overline{p_1 \wedge p_2}$. Obviously

$$Q \subset \overline{p_1} \cap \overline{p_2}\,.$$

Moreover, if for $i = 1, 2$ we denote by $\preceq_i$ the $\preceq$-order determined by the Ordering Process on $\overline{p_i} - \overline{p_i}_r$ by q_i, then Q is the largest subset of $\overline{p_1} \cap \overline{p_2}$ such that the $\preceq_i$-orders coincide (so we can denote it by $\preceq$) and which is $\preceq_i$-down-closed. Thus our goal is to show that the $\preceq_i$-last element of Q (actually $Q - Q_r$) is a cell. Because $q_1 \leq q_2$ the functions $[\overline{p_1}, q_1]$ and $[\overline{p_2}, q_2]$ coincide on $Q - Q_r$, and thus also $\langle \overline{p_i}, q_i \rangle$.

We first show that Q does contain cells: if $q_{1,l}$ is the first cell of the sequence q_1 then it is also the first cell of q_2 and because S2° ensures that $u_{1,l} \cap u_{2,l}$ has at least one cell, the uniqueness clause of S1° guarantees that the $\preceq_1$- and $\preceq_2$-least cells in $u_{1,l} \cap u_{2,l}$ are identical, hence that this least cell is in Q. As a consequence $Q - Q_r$ is nonempty; let p be its $\preceq$-last element, and assume for a contradiction that it is a value. Let $j = [\overline{p_1}, q_1]p^- = [\overline{p_2}, q_2]p^-$. Since $q_1 \leq q_2$ we have $q_{1,j} = q_{2,j}$ and we call this cell r. For $i = 1, 2$ define

$(w_i r_i) \in x$ as follows:

I– If the $\preceq_i$-next cell c_i after p in $\overline{p_i}$ exists and is such that $[\overline{p_i}, q_i]c_i = r$ then $r_i = r$ and w_i is the (uniquely defined) cell-like with $(w_i r_i) \in x$ having c_i as a maximal cell. Hence there is a single cell $c_i \in w_i$ not in Q.

II– If there is no such cell c then either p is the last move of p_i or for the $\preceq_i$-next cell c' we have $[\overline{p_i}, q_i]c' > [\overline{p_i}, q_i]c$. In both cases there has to be a value $q_{i,j+1}$ filling $q_{i,j} = r$ and we take $(w_i r_i) = (u_{i,j+1} q_{i,j+1})$. Clearly $w_i \subset Q$.

The contradiction is obtained by showing that no combination of cases I,II can be assigned to $i = 1, 2$. Case I cannot hold for both $(w_1 r_1)$, $(w_2 r_2)$ because then $r_1 = r_2$ are cells and 3.8 forces $w_1 \leq w_2$ or $w_2 \leq w_1$; since the maximal cell c_i of w_i cannot be filled this forces $w_1 = w_2$, thus $c_1 = c_2$, contradicting the assumption that p is the $\preceq$-last move of $\overline{p_1 \wedge p_2}$. The combination (say) I for $(w_1 r_1)$, II for $(w_2 r_2)$ is also impossible: it implies $r_1 < r_2$ is a cell, therefore by S2° $w_1 \cap w_2$ has a maximal cell which cannot be filled, but $w_1 \cap w_2 \subset Q$ and by 3.7 all the cells of Q are filled. Finally if case II holds for both pairs $(w_i r_i)$, since q_1 is a cell both $\overline{p_1}, \overline{p_2}$ contain additional cells, and we can start again by finding a $\preceq_i$-next cell d_i, that has to satisfy $[\overline{p_i}, q_i]d_i = j + 2$, and we can use 3.8 again to show $d_1 = d_2$, and proceed just as above.

The proof L3 is absolutely obvious. Notice that if $(uv) \in x^\dagger$ is such that $v \subset |Y|_0$, then $u \subset |X|_0$.

We have constructed the linear algorithm $\Psi_x : !X \to !Y$. It is left to show that the graph $(\varepsilon_Y^\sharp) \circ \Psi_x^\sharp$ is the trace $f^\sharp$ of a sequential algorithm $f : !X \to Y$ such that $f^\natural = x$. Look at the subgraph

$$G \;=\; \left\{ (p\underline{q}) \in \Psi_x^\sharp \mid q \in |Y| \right\}$$

which is obviously isomorphic to $(\varepsilon_Y^\sharp) \circ \Psi_x^\sharp$ because the map $q \mapsto \underline{q}$ is injective. Since G is down-closed and $\Psi_x^\sharp$ satisfies L1–L3, G trivially also satisfies these conditions and G is (isomorphic to) the linear trace of a sequential algorithm $f : !X \to Y$. Thus the graph $f^\natural$ is

$$f^\natural \;=\; \left\{ (\overline{p}q) \in X_\natural \times |Y| \mid (p\underline{q}) \in \Psi_x^\sharp \right\} ;$$

from the definition of Ψ_x we know that given a pair $(\overline{p}q) \in f^\natural$ there is a family $(u_r)_{r\in\underline{\overline{q}}}$ with $(u_r r) \in x$ such that $\overline{p} = \bigcup_r u_r$; notice that $\underline{\overline{q}} = q\downarrow$, hence we get

$$\overline{p} \;=\; \bigcup_{r\leq q} u_r \;=\; u_q \cup \bigcup_{r\leq q,\, r \text{ a value}} u_r$$

and it is easy to see that $\overline{p} = u_q$, since all the $(\bullet\bullet)$ pairs $(u_r r)$ have $u_r \leq u_q$, which shows $(\overline{p}q) \in x$, hence $f^\natural = x$. This completes the proof of 3.6. $\square$

We can now officially define composition for sequential algorithms. If $f : !X \to Y$, $g : !Y \to Z$ are linear algorithms we define their Kleisli composition $g * f : !X \to Z$ as the linear composite

$$g * f \;=\; g(\Psi_{f^\natural}) \quad .$$

Out of the definition above we get that $(g * f)^\natural = g^\natural \circ f^{\natural\dagger}$, which is also equal to the result of the equation just above 3.6. This enables us to show easily that Kleisli composition is associative: given another linear $h : !Z \to W$ we get

$$\begin{aligned} &((h * g) * f)^\natural \;=\; (h * (g * f))^\natural \;= \\ &= \left\{ (ur) \mid \text{ there exist } v, w \text{ with } (uv) \in f^{\natural\dagger}, (vw) \in g^{\natural\dagger}, (wr) \in h^\natural \right\} \end{aligned}$$

Given an acceptable object X we also have to check that $\varepsilon_X : !X \to X$ is a unit for Kleisli composition both to the left and the right; this will be left to the reader.

Given sequential domains X, Y we denote the set of linear algorithms $X \to Y$ by $\mathcal{SA}(X, Y) = \mathcal{LA}(!X, Y)$. As noted above we get something which is "almost a category". We use $X \Rightarrow Y$ to denote the sequential domain $!X \multimap Y$.

Theorem 2 *There is a natural isomorphism* $\mathcal{SA}(X \& Y, Z) \cong \mathcal{SA}(X, Y \Rightarrow Z)$

We only have to show that $!(X\&Y)$ is naturally isomorphic to $!X \otimes !Y$. Then we get the well-known sequence of isomorphisms

$$\begin{aligned} \mathcal{LA}(X\&Y, Z) \;&\cong\; (!(X\&Y) \multimap Z)_* \;\cong\; ((!X \otimes !Y) \multimap Z)_* \\ &\cong\; (!X \multimap (!Y \multimap Y))_* \;\cong\; \mathcal{LA}(X, Y \Rightarrow Z) \quad . \end{aligned}$$

Let p be a value in $|!(X\&Y)|_n$. Thus $p = (p_0 p_1 \dots p_n)$ where $p_0 \subset |X\&Y|_0^\bullet$ and $p_i \in |X\&Y|$. But $|X\&Y|$ is the disjoint union of $|X|$ and $|Y|$ and this allows us to construct a value $\chi(p) = ((q_0 r_0)(q_1 r_1)\dots(q_n r_n))$ of $|!X \otimes !Y|$ (where $q_i \in |!X|$, $r_i \in |!Y|$) as follows:

$$(q_0 r_0) = ((p_0 \cap |X|)(p_0 \cap |Y|)) \in |!X|_0 \times |!Y|_0 = |!X \otimes !Y|_0 \quad .$$

$(q_{i+1} r_{i+1})$ is constructed by induction: if $(q_0 r_0)\dots(q_i r_i)$ have been constructed then

$$(q_{i+1} r_{i+1}) \quad = \quad \begin{cases} ((q_i \cdot p_{i+1}) r_i) \text{ if } p_{i+1} \text{ is in } |X| \\ (q_i (r_i \cdot p_{i+1})) \text{ if } p_{i+1} \text{ is in } |Y| \end{cases}$$

where the operation $\cdot$ is concatenation of a sequence of moves with a single move. It is very simple to see that χ is an isomorphism $|!(X\&Y)| \to |!X| \otimes |!Y|$ and this isomorphism is natural in both variables in the linear category. It is also easy to see that χ is compatible with the associativity and symmetry isomorphisms associated with the two symmetrical monoidal structures it links: the monoidal structure of additive conjunction and the one of multiplicative conjunction. □

Now to get something like a cartesian closed category evaluation has to be defined; its being the transpose of the identity morphism, say in $\mathcal{SA}(X \Rightarrow Y, X \Rightarrow Y)$, we have to know when $(X \Rightarrow Y)$ is acceptable. There are two important classes of acceptable domains that define cartesian closed categories. First, the class of domains X such that $|X|_0 = |X|_0^\bullet$ is obviously closed under the additives, $\otimes$, $\multimap$ and $!$, and thus cartesian closed (notice that this class is also closed under $\odot$, which is a trivial operation in it). Also if $|X|_0 = |X|_0^\circ$ and Y is a cds, then since $|!X|_0$ is a $\bullet$ singleton we get that $X \Rightarrow Y$ is a cds since the coherent spaces $|X \Rightarrow Y|_0$ and $|Y|_0$ are isomorphic. Hence the category of cdses and sequential algorithms is cartesian closed, being obviously closed under $\&$ (it is also closed under $\odot$).

4 Abstract sequential algorithms

Let X be a sequential domain. Recall that at the very beginning of this paper we set a bijective correspondence between the sets

$$\left\{ \begin{array}{l} \text{subsets of } |X|^\bullet \\ \text{satisfying a), b), c)} \end{array} \right\} \cong \left\{ \begin{array}{l} \text{consistent subsets of } |X| \\ \text{whose max. elts. are values} \end{array} \right\} .$$

This bijection was simple: if x was in the set to the left then $x\!\downarrow$ was its corresponding element to the right. Conversely if u was in the set to the right then $u \cap |X|^\bullet$ was its correspondent in the left set. The elements in the left set we called answers and we denoted the set of all such by X_*. In this chapter we would now like to consider an answer as an element in the right set: from now on an element of X_* is considered as a consistent subset of $|X|$. An element in the left set we will call an *answer in the old sense*. The order $\leq$ on X_* is inclusion as before, but we tend to use the more concrete notation $\cap$ for the $\leq$-inf.

Let X, Y be cdses. Given $x \in X_*$ we define

$$X_x \quad = \quad \{\, p \in |X|^\circ \mid x \vdash p \,\} \quad .$$

We use [4]'s adaptation of the axioms in [2], according to which a Sequential Algorithm $f : X \to Y$ (we use capitals to distinguish this definition from ours) is a pair (f_*, ϕ) where $f : X_* \to Y_*$ is a monotone function and $\phi = (\phi_x)_{x \in X_*}$ a family of *partial* functions $\phi_x : Y_{f_*(x)} \to X_x$ satisfying the following (in what follows $x \in X_*$ and $q \in Y_{f_*(x)}$; by $a \doteq b$ we mean "either $a = b$ or both are undefined"):

- (*Sequentiality*) Given $x' \geq x$, if q is filled by $f(x')$ then $\phi_x(q)$ is defined and filled by x'.
- (*Permanence*) Given $x' \geq x$ such that $\phi_x(q)$ is not filled by x (in particular $\phi_x(q)$ can be undefined), then $\phi_{x'}(q) \doteq \phi_x(q)$.
- (*Continuity*) If $\phi_x(q)$ is defined there is a finite $y \leq x$ such that $\phi_y(q) = \phi_x(q)$ is defined.

Proposition 4.1 *If $(f_*, \phi) : X \to Y$ is a Sequential Algorithm then f_* is stable.*

The proof is in [4], and is closely related to the next result. □

Proposition 4.2 *Let $(f_*, \phi) : X \to Y$ be a Sequential Algorithm, $x \in X_*$, $q \in Y_{f_*(x)}$ such that $\phi_x(q)$ is defined. Then the set of answers*

$$A = \{\, y \leq x \mid \phi_y(q) = \phi_x(q) \,\}$$

has a least element.

First we know by Continuity that there is $y \in A$ finite. Hence it suffices to show that the subset $A_{\mathrm{f}} \subset A$ made of finite elements is closed under binary intersections. If $y_1, y_2 \in A_{\mathrm{f}}$, the only interesting case in establishing $y_1 \cap y_2 \in A_{\mathrm{f}}$ is when $y_1 \neq y_1 \cap y_2 \neq y_2$; thus assume this, and denote $y_1 \cap y_2$ by z. If $q \notin Y_{f_*(z)}$ let $q' \in Y_{f_*(z)}$ be the unique such cell with $q' \leq q$. Since $f(y_1)$, $f(y_2)$ fill q' by Sequentiality we have both $\phi_z(q') \in y_1 - z$ and $\phi_z(q') \in y_2 - z$, a contradiction. Thus $q \in Y_{f_*(z)}$, and since $z = y_1 \cap y_2$ and $y_1 \neq z \neq y_2$ there is one of y_1, y_2 that does not fill $\phi_z(q)$ (whether it's defined or not); call it y_i. We can now apply Permanence to $y_i \geq z$ to get that $\phi_z(q) = \phi_z(y_i)$. □

Given $x \in X_*$, $q \in Y_{f(x)}$, we say (x, q) *is ϕ-minimal* if $\phi_x(q)$ is defined and there is no answer $y < x$ such that $\phi_y(q) = \phi_x(q)$. Thus the previous proposition says that for every $x \in X_*$ such that $\phi_x(q)$ is defined there is a unique ϕ-minimal (y, q) with $y \leq x$.

Let $u \in X_\natural$. By $\lfloor u \rfloor$ we denote the answer obtained by removing the maximal cell of u if it is cell-like, and doing nothing if it is value-like.

Theorem 3 *The category of cdses and sequential algorithms is equivalent to the category of cdses and Sequential Algorithms.*

Let X, Y be cdses and $f : X \to Y$ a sequential algorithm. We define $f_*{:}X_* \to Y_*$ by first taking

$$f_1(x) = \{\, p \in |Y|^\bullet \mid \text{ there exists } u \subset x \text{ with } (up) \in f^\natural \,\} \quad .$$

Let us show that the set $f_1(x) \subset |Y|^\bullet$ is a value in the old sense; it is down-closed because of condition S1$^\bullet$; it is coherent at the roots for trivial reasons, and if $p_1, p_2 \in f_*(x)$ have $p_{1r} = p_{2r}$ then there are $u_i \subset x$ with $(u_i p_i) \in f^\natural$, thus $u_1 \cup u_2$ is consistent and by 3.5 $p_1 \wedge p_2$ is a value. Hence taking

$$f_*(x) = f_1(x){\downarrow}$$

does indeed give a function $X_* \to Y_*$ which is obviously monotone.

Now, given $x \in X_*$ and $q \in Y_{f_*(x)}$ the partial function ϕ_x is defined as

$$\phi_x(q) = \text{the max. cell } c \text{ of } u \text{ with } (uq) \in f^\natural \text{ such that } \lfloor u \rfloor \subset x,\, c \notin x \ .$$

If u above exists then it is unique, because given u' satisfying the same conditions, the requirement $\lfloor u \rfloor \subset x$ force $u \cup u'$ to be consistent, and we can then apply 3.8 to show $u = u'$. We are left to prove the pair (f_*, ϕ) is a Sequential Algorithm, which is straighforward and will be left to the reader.

For the reverse direction assume (f_*, ϕ) is a Sequential Algorithm. Define the set

$$W \quad = \quad \begin{array}{l} \Big\{ (up) \mid p \in |Y|^\bullet \text{ and } u \text{ is minimal in } f_*^{-1}(p\uparrow) \Big\} \cup \\ \Big\{ (uq) \mid q \in |Y|^\circ \text{ and } u = x \cup \{\phi_x(q)\},\, (x,q)\ \phi\text{-minimal} \Big\} \end{array} .$$

Here we consider a value $p \in |Y|^\bullet$ as an answer, in other words we identify p with $p\downarrow$. Since f_* is stable, for every $p \in |Y|^\bullet$ the set $f_*^{-1}(p\uparrow)$ is always guaranteed to be a disjoint union of sets of the form $x\uparrow$, for $x \in X_*$ a finite answer. The definition of W shows that the decomposition of the trace of a sequential algorithm in $(\bullet\bullet)$ and $(\circ\circ)$ pairs takes meaning when looked at from the point of view of Sequential Algorithms: the $(\bullet\bullet)$ pairs form the ordinary trace $\mathrm{Tr}(f_*)$, the other pairs contain the information about ϕ. We now have to show that there is a linear algorithm f such that $W = f^\natural$, in other words that W satisfies conditions S1–S3. Before we start doing this we need a little lemma:

Proposition 4.3 *Let $(uq) \in W$, $x \leq u$ a value and $r \leq q$ a cell such that $f_*(x) \vdash r$. Then $\phi_x(r) \in u$ and is defined.*

Assume there is an answer y between x and u such that $f_*(y)$ fills r; then the conclusion follows by Sequentiality. If $f_*(y)$ does not fill r for any value $x \leq y \leq u$ then u obviously has to be cell-like (since otherwise q would be a filler) and $r = q$, and we know the maximal cell c of u is equal to $\phi_{\lfloor u \rfloor}(q)$; if follows that $\phi_x(r) = \phi_x(q)$ is defined because otherwise by Permanence applied to $\lfloor u \rfloor \geq x$ we would get that $\phi_{\lfloor u \rfloor}(q)$ is also undefined. This way we can also show that $\phi_x(r) \in u$ because otherwise by Permanence we would get $\phi_{\lfloor u \rfloor}(q) \notin u$. □

Conditions S1$^\bullet$ and S2$^\bullet$ are straightforward consequences of the stability of f_* (actually, equivalent to stability), while condition S3 is trivial (since the root sets are). For condition S1° let $(uq) \in W$ and $c \in u$ be a cell. We are looking for $(vr) \leq (uq)$ in W such that c is v's maximal cell. Let $x \leq u$ be the greatest answer $x \in X_*$ that does not fill c; in other words

$$x = \{\, p \in u \mid c \not\leq p \,\} .$$

There is a cell $r \leq q$ with $f(x) \vdash r$; otherwise $q \in f_*(x)$ and is a value, which contradicts the minimality of the pair (uq) since $x \neq u$. We know $\phi_x(r) \in u$ by Proposition 4.3, but the only cell $b \in u$ with $x \vdash b$ is c and this shows $\phi_x(r) = c$.

We define $z \leq x$ to be the unique answer such that (z, r) is ϕ-minimal, and taking $v = z \cup \{c\}$ will give us the required (vr) pair. For uniqueness let $(v'r') \leq (uq)$ be another pair of W with c the maximal cell of v'. r and r' are related so assume $r < r'$. Then $f_*(\lfloor v' \rfloor)$ fills r, and by Sequentiality $\phi_{\lfloor v \rfloor}(r) = c$ is filled by v', a contradiction. By symmetry $r' < r$ is also impossible, and therefore the only possibility is $r = r'$ and the equation $v = v'$ follows by 4.2. This finishes the proof of S1°.

Let $(uq) \in W$, $c, c' \in u$ be cells and $(vr), (v'r') \leq (uq)$ the pairs of W for which c, c' are maximal cells for v, v', respectively.

i) If $c' \in v$ then we have $v' \subset v$ because the assumption $c' \in v$ determines a pair $(v''r'') \leq (vr)$ with c' maximal for v'' and then both $(v'r'), (v''r'')$ are pairs below (uq) that have c' as maximal cells, hence $(v'r') = (v''r'')$ by the uniqueness clause of S1°.

ii) If $r = r'$ it cannot be that both $c \notin v'$, $c' \notin v$, because then, letting $y = \lfloor v \rfloor \cup \lfloor v' \rfloor$ it would imply $f_*(y) \vdash r$ by Sequentiality (r cannot be filled by $f_*(y)$ for one of c, c' would be filled by y too), and then we could apply Permanence to $y \geq \lfloor v \rfloor$ to get $\phi_y(r) = \phi_{\lfloor v \rfloor}(r) = c$, and to $y \geq \lfloor v' \rfloor$ to get $\phi_y(r) = \phi_{\lfloor v' \rfloor}(r) = c'$, thus $c = c'$, which clearly is a contradiction. Thus by i) we get $v \leq v'$ ($c \in v'$) or $v' \leq v$ ($c' \in v$).

iii) If $r < r'$ then $v < v'$ because $f_*(x')$ fills r, thus x' fills c, and the conclusion follows by a).

We can now start proving S2°. Let $(u_1q_1), (u_2q_2) \in W$ be such that $q_1 \leq q_2$ and q_1 is a cell. We are looking for a cell d maximal in $u_1 \cap u_2$ with a certain property. If $i = 1$ or 2 and $c \in u_i$ is a cell, let $r_i(c) \leq q_i$, $v_i(c) \subset u_i$ be such that $(v_i(c)r_i(c)) \in W$ and c is $v_i(c)$'s maximal cell. The discussion right above tells us that the relation $\preceq_i$ on the cells of u_i defined by

$$c \preceq_i c' \quad \text{iff} \quad v_i(c) \subset v_i(c')$$

is a total ordering: first we always have $r_i(c) < r_i(c')$, $r_i(c) > r_i(c')$ or $r_i(c) = r_i(c')$. In the first two cases the conclusion $c \leq c'$ or $c \geq c'$ follows from iii). In the third case the conclusion follows from ii).

Let now $C \subset u_1 \cap u_2$ be the set of cells c such that $v_1(c) = v_2(c)$ and $r_1(c) = r_2(c)$. We claim C is not empty. This is because if c_i is u_i's $\preceq_i$-minimal cell then obviously $c_i \in |X|_0$ and $v_i(c_i) = \{c_i\}$; we have $r_i(c_i) \leq q_1$ and this implies $r_1(c_1) = r_2(c_2)$ (because $r_i(c_i) < r_j(c_j)$ implies c_i is filled by $v_j(c_j)$, by iii) above), and ii) forces $c_1 = c_2$ since it means one singleton $\{c_i\}$ is contained in the other. Obviously the $\preceq_1$ and $\preceq_2$ orderings coincide on C. Let now $d \in C$ be the greatest cell for that common order. We have to show that d is maximal in $u_1 \cap u_2$; assume for a contradiction there is a value $p \in u_1 \cap u_2$ filling d and let $z = v_1(d) \cup \{p\}$. Clearly $z < u_i$, hence $f_*(z)$ does not fill q_1. If $s \leq q_1$ is the uniquely defined cell such that $f_*(z) \vdash s$, then by 4.3 $\phi_z(s) \in u_1$ is defined and it is obviously the $\preceq_1$-next cell in u_1. But $r \leq q_1 \leq q_2$ and we can apply 4.3 again to get $\phi_z(s) \in u_2$, making it the $\preceq_2$-next cell in u_2, and this makes $\phi_z(s)$ an element of C above d, the desired contradiction.

We finally have to show that d has the "certain property"

$$u_i \neq u_1 \cap u_2 \quad \text{iff} \quad d \text{ is filled by } u_i\ .$$

Obviously it is enough to prove necessity: assume $u_i \neq u_1 \cap u_2$. Since $v_i(d) < u_i$ there is $b \in u_i - v_i(d)$. If $b^- = d$ we are done. Otherwise there is a cell $b' \leq b$ in $u_i - v_i(d)$ since all the maximal elements $\neq d$ of v_i are values. Then $d \prec_i b'$ by definition of d, hence d is filled in $v_i(b') \subset u_i$. This concludes the proof of S2°.

We have proved that the hom-sets of the category of sequential algorithms are isomorphic to the hom-sets of the category of Sequential Algorithms; it is left to show that these isomorphisms respect composition, which is quite easy. □

References

[1] S. Abramsky, R. Jagadeesan: Games and full completeness for multiplicative linear logic, preprint.

[2] G. Berry, P.-L. Curien: Sequential algorithms on concrete data structures. *Theoret. Comp. Sci.* 20(1982)265–321.

[3] A. Blass: A games semantics for linear logic. *Annals of Pure and Applied Logic*, 56(1992)183–220.

[4] A. Bucciarelli, Th. Ehrhard: A theory of sequentiality, to appear in *Theoret. Comp. Sci.*

[5] R, Cartwright, P.- L. Curien, M. Felleisen: Fully abstract models of observably sequential languages, manuscript.

[6] T.Coquand, C. Gunter, G. Winskel: dI-domains as models of polymorphism, in Main, Melton, Mislove & Schmidt eds: *Mathematical Foundations of Programming Semantics* Lecture notes in Computer Science vol. 298, pp. 344-363, Springer 1989.

[7] J. H Conway: On numbers and games, *London Mathematical Society Monographs* Vol. 6, Academic Press 1976.

[8] P.- L. Curien: Categorical combinators, sequential algorithms and functional programming.*Research notes in theoretical computer science*, Pitman, London, 1986.

[9] J-Y. Girard: Linear Logic. *Theoret. Comp. Sci* 50(1987)1–20.

[10] A. Joyal: Remarques sur la théorie des jeux a deux personnes, *Gazette des sciences mathématiques du Québec* 1.4(1977).

[11] G Kahn, G. Plotkin: Domaines Concrets, Rapport IRIA-LABORIA 336(1978).

[12] F. Lamarche Sequentiality, games and linear logic (Announcement). in: Proceedings of the CLICS workshop, Aarhus University, 23–27 March 1992. *Publications of the Computer Science department of Aarhus University*, DAIMI PB-397-II(1992).

[13] F. Lamarche: Stable domains are generalized topological spaces. To appear in *Theoretical Computer Science*.

[14] F. Lamarche: Games, additives and correctness criteria, manuscript.

[15] P. Malacaria, L. Régnier: Some results on the interpretation of λ-calculus in operator algebras. Proceedings of LICS 91.

[16] G-Q. Zhang: Stable neighbourhoods (*Note*) *Theoret. Comp. Sci.* 93(1992)143–157

[17] Some sample paper on the use of games in set theory.