Games for recursive types

Samson Abramsky and Guy McCusker
Imperial College

13 October 1994

Abstract

We present results concerning the solution of recursive domain equations in the category $\mathcal{G}$ of games, which is a modified version of the category presented in [AJM94]. New constructions corresponding to lifting and separated sum for games are presented, and are used to generate games for two simple recursive types: the vertical and lazy natural numbers.

Recently, the "game semantics" paradigm has been used to model the multiplicative fragment of linear logic [AJ94], and to provide a solution to the full abstraction problem for PCF [AJM94, HO94], where the intensional structure of the games model captures both the sequential and functional nature of the language. In the light of these results, it is natural to ask whether recursive types can be handled in this setting. Here we show that they can: for a wide class of functors Φ, including all of the usual type constructors, the equation $D \simeq \Phi(D)$ has a (canonical) solution. In fact we solve this equation up to identity, and the solution can be constructed in the usual way by iterating the action of the functor on (the object corresponding to) the empty type.

1 Games and strategies

We define here a category of games and (equivalence classes of) history-free strategies which is almost identical to that used in [AJM94], which goes on to define the linear logic connectives $\otimes$ (tensor), $\multimap$ (linear implication), ! (the 'of course' exponential) and & (the 'with' product). The co-Kleisli category for the comonad ! is then a model of intuitionistic linear logic. We do not give the details of these constructions here, merely presenting those definitions which are essential to the rest of the paper. In particular we define $\multimap$, since it is central to the definition of the category.

1.1 Games

A game has two participants, Player (P) and Opponent (O). A *play* of the game consists of a finite or infinite sequence of moves, alternately by O and P. In the games we consider, O always moves first.

Before defining games, we need some notation for sequences and operations on sequences. We shall use $s, t, \ldots$ to range over sequences and $a, b, \ldots$ to range over the elements of these sequences. We shall write as to mean the sequence whose first element is a and whose tail is s; and st for the concatenation of sequences s and t. $|s|$ denotes the length of s, and s_i is the ith element of s. If S is a set, $s \upharpoonright S$ is the restriction of s to elements of S, i.e. the sequence s with all elements not in S deleted. Finally, if S is a set of sequences, then S^{even} is the subset of all even length sequences in S.

A game is specified by a structure $A = (M_A, \lambda_A, P_A, \approx_A)$, where

- M_A is a set of moves.
- $\lambda_A : M_A \rightarrow \{P, O\} \times \{Q, A\}$ is the labelling function.
  The labelling function indicates if a move is by P or by O, and if a move is a question (Q) or an answer (A).
  We write

$$\begin{array}{rcl} \{P,O\} \times \{Q,A\} & = & \{PQ, PA, OQ, OA\} \\ \lambda & = & \langle \lambda_A^{PO}, \lambda_A^{QA} \rangle \\ M_A^P & = & \lambda_A^{-1}(\{P\} \times \{Q,A\}) \\ M_A^O & = & \lambda_A^{-1}(\{O\} \times \{Q,A\}) \\ M_A^Q & = & \lambda_A^{-1}(\{P,O\} \times \{Q\}) \\ M_A^A & = & \lambda_A^{-1}(\{P,O\} \times \{A\}) \end{array}$$

  and so on, and define

$$\overline{P} = O, \quad \overline{O} = P,$$
$$\overline{\lambda_A^{PO}}(a) = \overline{\lambda_A^{PO}(a)}, \quad \overline{\lambda_A} = \langle \overline{\lambda_A^{PO}}, \lambda_A^{QA} \rangle.$$

- Let $M_A^{\circledast}$ be the set of all finite sequences s of moves satisfying:
  1. *O moves first:*
  $$s = at \Rightarrow a \in M_A^O$$
  2. *Alternation:*
  $$(\forall i : 1 \leq i < |s|)\ [\lambda_A^{PO}(s_{i+1}) = \overline{\lambda_A^{PO}(s_i)}]$$

3. *Bracketing condition:*

$$(\forall t \sqsubseteq s)(|t \upharpoonright M_A^A| \leq |t \upharpoonright M_A^Q|)$$

where $\sqsubseteq$ is the prefix ordering on sequences.

Then P_A, the set of valid *positions* of the game, is a non-empty prefix-closed subset of $M_A^{\circledast}$.

The bracketing condition ensures that when an answer is given, there is at least one unanswered question in the position. We think of the answer as being to the most recently asked, as yet unanswered question, so that a question asked by Opponent must be answered by Player and vice versa.

- $\approx_A$ is an equivalence relation on P_A satisfying:

  1. *Preserves labels:*
  $$s \approx_A s' \Rightarrow \lambda_A^*(s) = \lambda_A^*(s')$$
  Notice that this implies that if $s \approx_A s'$ then $|s| = |s'|$.

  2. *Prefix closure:*
  $$st \approx_A s't' \wedge |s| = |s'| \Rightarrow s \approx_A s'$$

  3. *Extendibility:*
  $$s \approx_A s' \wedge sa \in P_A \Rightarrow (\exists a')[sa \approx_A s'a']$$

For example, a game for **Bool** has one possible opening move $*$, which is a request for data, and $\lambda_{\mathbf{Bool}}(*) = OQ$; there are then two possible responses for Player, $\mathbf{tt}$ and $\mathbf{ff}$, with $\lambda_{\mathbf{Bool}}(\mathbf{tt}) = \lambda_{\mathbf{Bool}}(\mathbf{ff}) = PA$. The equivalence relation is just the identity relation on the four possible positions of the game, namely ϵ, $*$, $*\mathbf{tt}$ and $*\mathbf{ff}$. A game for **Nat** can be defined similarly. Another important game is the empty game $I = (\varnothing, \varnothing, \{\epsilon\}, \{(\epsilon, \epsilon)\})$, which turns out to be both a terminal object in the category and the unit of the tensor product. It also plays a key role in the construction of solutions to recursive equations in games, as we will see.

The equivalence relation will play no part in the results presented here. It is, however, vital for the definition of !, which is in turn essential to the definition of the ordinary intuitionistic implication, which gives us our 'function space' types.

1.2 Strategies

A strategy for Player in a game A can be thought of as a function telling Player what move to make in a given position. Since a position in which Player is about to move is always an odd-length sequence of moves, we can define a strategy as a set of even-length positions thus:

A *strategy* for Player in a game A is a non-empty set $\sigma \subseteq P_A^{\mathtt{even}}$ such that $\overline{\sigma} \stackrel{\mathrm{def}}{=} \sigma \cup \mathbf{dom}(\sigma)$ is prefix-closed, where

$$\mathbf{dom}(\sigma) \stackrel{\mathrm{def}}{=} \{sa \in P_A^{\mathtt{odd}} \mid (\exists b)[sab \in \sigma]\}$$

We are interested only in history-free strategies, i.e. those strategies whose responses depend only on the last move made, rather than on the whole position. A strategy σ is *history-free* if it satisfies

- $sab, tac \in \sigma \Rightarrow b = c$
- $sab, t \in \sigma, ta \in P_A \Rightarrow tab \in \sigma$

Note that history-free strategies are closed under directed unions.

If σ is a history-free strategy for a game A, we shall write $\sigma : A$.

We extend $\approx_A$ to a partial equivalence relation (i.e. a symmetric, transitive relation), which we write as $\approx$, on strategies for A thus:

$\sigma \approx \tau$ iff

1. $sab \in \sigma, s'a'b' \in \tau, sa \approx_A s'a' \Rightarrow sab \approx_A s'a'b'$
2. $s \in \sigma, s' \in \tau, sa \approx_A s'a' \Rightarrow sa \in \mathbf{dom}(\sigma)$ iff $s'a' \in \mathbf{dom}(\tau)$

Proposition 1

1. $\approx$ is a partial equivalence relation on strategies.
2. $\sigma \approx \tau$ iff
   - $\sigma \approx \sigma, \tau \approx \tau$
   - $(\forall s \in \sigma)(\exists t \in \tau)[s \approx_A t]$
   - $(\forall t \in \tau)(\exists s \in \sigma)[s \approx_A t]$

The proof of these facts is straightforward, and is omitted. From now on we shall freely write $s \approx s'$ for $s \approx_A s'$ where it is clear what is meant from the context.

1.3 Linear implication

We define the game $A \multimap B$, given games A and B. We refer to A and B as the *component* games.

The game $A \multimap B$ is defined as follows:

- $M_{A \multimap B} = M_A + M_B$.
- $\lambda_{A \multimap B} = [\overline{\lambda_A}, \lambda_B]$
- $P_{A \multimap B}$ is the set of all $s \in M^{\circledast}_{A \multimap B}$ satisfying
  1. *Projection condition:* $s \upharpoonright M_A \in P_A$ and $s \upharpoonright M_B \in P_B$. Here we use the notation $s \upharpoonright M_A$ as a suggestive shorthand for the projection of s onto its first component i.e. M_A.
  2. *Stack discipline:* Every answer is in the same component as the corresponding question.
- $s \approx_{A \multimap B} s'$ iff $s \upharpoonright M_A \approx_A s' \upharpoonright M_A, s \upharpoonright M_B \approx_B s' \upharpoonright M_B$ and $(\forall i : 1 \leq i \leq |s|)[s_i \in M_A \iff s'_i \in M_A]$

Notice that an immediate consequence of the projection condition described above is the *switching condition:* if two successive moves are in different components, i.e. one is in A and the other is in B, it is the Player who has switched components, i.e. the second of the two moves is a P-move. This is because when it is Opponent's turn to move, the total number of moves so far played must be even, so that the numbers of moves in the two components are either both even or both odd. In the former case, opponent can only play in B, and in the latter, only in A. In either case, a simple argument shows that this is the component where Player last moved.

1.4 The category of games

First, some notation: if σ is a history-free strategy for a game A with $\sigma \approx \sigma$ write $[\sigma] = \{\tau \mid \tau \approx \sigma\}$. Let $\hat{A}$ be the set of all such equivalence classes.

Define a category $\mathcal{G}$:

$$\begin{array}{rcl} \text{Objects} & : & \text{Games} \\ \text{Morphisms} & : & [\sigma] : A \to B \text{ is a partial equivalence class } [\sigma] \in \widehat{A \multimap B} \end{array}$$

Identity For any game A, the identity morphism $[\mathtt{id}_A]$ is the equivalence class of the "copycat" strategy, $\mathtt{id}_A$ on the game $A \multimap A$, defined by

$$\mathtt{id}_A = \{s \in P^{\mathtt{even}}_{A_1 \multimap A_2} \mid s \upharpoonright A_1 = s \upharpoonright A_2\}$$

We use subscripts on the A's to distinguish the two occurences.

Composition Here we define the composition of *strategies* $\sigma : A \multimap B$ and $\tau : B \multimap C$. This construction is then lifted to equivalence classes, to give a definition of composition of morphisms.

Given $\sigma : A \multimap B$ and $\tau : B \multimap C$, define their composition $\sigma;\tau : A \multimap C$ by

$$\sigma;\tau = \{s \restriction A, C \mid s \in (M_A + M_B + M_C)^*, s \restriction A, B \in \overline{\sigma}, s \restriction B, C \in \overline{\tau}\}^{\mathsf{even}}$$

For a full proof that this is well-defined and associative, see [AJ94].

Proposition 2 Composition is compatible with $\approx$:

$$\sigma, \sigma' : A \multimap B,\ \tau, \tau' : B \multimap C,\ \sigma \approx \sigma', \tau \approx \tau' \Rightarrow \sigma;\tau \approx \sigma';\tau'$$

In the light of the above Proposition, we can now define composition of morphisms via composition of strategies: $[\sigma];[\tau] \stackrel{\text{def}}{=} [\sigma;\tau]$ assuming the strategies σ and τ are of suitable types. This is the general pattern for defining constructions on morphisms: first a concrete construction on strategies is given, then it is shown to be compatible with $\approx$.

$\mathcal{G}$ as an autonomous category As in [AJM94], it is possible to define the tensor product of two games, and extend this to morphisms so that it becomes a bifunctor. The same can be done for linear implication, and $\mathcal{G}$ then becomes an autonomous category. This is not important for us here, but it will be informative to see how linear implication acts on morphisms. Given $\sigma : A \multimap B$, $\tau : A' \multimap B'$, we define $\sigma \multimap \tau : (B \multimap A') \multimap (A \multimap B')$ to be

$$\{s \in P^{\mathsf{even}}_{(B \multimap A') \multimap (A \multimap B')} \mid s \restriction A, B \in \sigma, s \restriction A', B' \in \tau\}$$

It can be shown that this is indeed a well-defined history-free strategy, and that the action of $\multimap$ on strategies is compatible with $\approx$, so that it defines an action on morphisms. This in fact makes linear implication into a bifunctor $(-) \multimap (+) : \mathcal{G}^{\mathsf{op}} \times \mathcal{G} \to \mathcal{G}$, as expected.

2 Solving recursive equations over games

We present here results concerning the solution of equations of the form $G = \Phi(G)$ where Φ is some endofunctor on the category of games. In fact, because in general one needs to work with 'functors' of mixed variance, we solve equations of the form $G = \Phi(G, G)$ where $\Phi : \mathcal{G}^{\mathsf{op}} \times \mathcal{G} \to \mathcal{G}$ in the style of, for example, [Pit93]. Our approach is similar to the I-categories and IP-categories of Edalat and Smyth [ES93, Eda93]. We define two partial orders: one on games and one on strategies, the latter of which is used to give an order on each hom-set; and we specify a collection of distinguished *inclusion* and *projection* morphisms between certain games which make $\mathcal{G}$ into an I-category. However, we have been unable to establish whether this is a complete I-category, and as such cannot use results from *op. cit.* directly; in fact the results here stand alone, and familiarity with I-categories is neither assumed nor necessary.

2.1 The order on games

Here we present the details of the 'restriction' ordering on games, similar to the usual treatment of information systems, and demonstrate that it is indeed a complete partial order.

Given two games A and B, write $A \trianglelefteq B$ iff

- $M_A \subseteq M_B$
- $\lambda_A = \lambda_B \restriction M_A$
- $P_A = P_B \cap M_A^{\circledast}$
- $s \approx_A s'$ iff $s \approx_B s'$ and $s \in M_A^{\circledast}$

Here $\lambda_B \restriction M_A$ denotes the restriction of the labelling function on B to the moves of A. So $A \trianglelefteq B$ means that A is a 'full subgame' of B. The lopsidedness of the last condition is important; it ensures that in B, no position of A is equivalent to a position not in A, thus allowing us to consider a strategy for A as a strategy for B:

Lemma 3 If $\sigma : A$, $\sigma \approx_A \sigma$, and $A \trianglelefteq B$, then σ, considered as a subset of P_B, is a well defined history-free strategy for B, and $\sigma \approx_B \sigma$.

Proof σ is clearly a history-free strategy for B. The crucial step is in the second condition for $\sigma \approx_B \sigma$. Suppose $sab, s' \in \sigma$ and $sa \approx_B s'a'$. We must show that $s'a' \in \mathbf{dom}(\sigma)$. But since $sab \in \sigma$, we have $sa \in M_A^{\circledast}$, so since $A \trianglelefteq B$, $sa \approx_A s'a'$, and so the fact that $\sigma \approx_A \sigma$ tells us that $s'a' \in \mathbf{dom}(\sigma)$. ∎

Proposition 4 $\trianglelefteq$ defines a complete partial order on the objects of $\mathcal{G}$, with least element I.

Proof That $\trianglelefteq$ is a partial order with least element I is trivial. To see that it is complete, suppose Δ is a directed set of games. Let

$$A^* = (M, \lambda, P, \approx)$$

where

$$\begin{aligned} M &= \bigcup\{M_A \mid (M_A, \lambda_A, P_A, \approx_A) \in \Delta\} \\ \lambda &= \bigcup\{\lambda_A \mid (M_A, \lambda_A, P_A, \approx_A) \in \Delta\} \\ P &= \bigcup\{P_A \mid (M_A, \lambda_A, P_A, \approx_A) \in \Delta\} \\ \approx &= \bigcup\{\approx_A \mid (M_A, \lambda_A, P_A, \approx_A) \in \Delta\} \end{aligned}$$

The directedness of Δ easily gives us that this is a well-defined game. We shall show that it is the least upper bound of Δ.

Upper Bound Suppose $A \in \Delta$. Then $M_A \subseteq M$ by the definition of M, and it is easy to see that $\lambda_A = \lambda \upharpoonright M_A$.

It is clear that $P_A \subseteq P \cap M_A^{\circledast}$. For the converse, suppose $s \in P \cap M_A^{\circledast}$. Then for some $B \in \Delta$, $s \in P_B$. Since Δ is directed, there exists a $C \in \Delta$ such that $A, B \trianglelefteq C$. Then $s \in P_C$, and $s \in M_A^{\circledast}$, so $s \in P_A$.

The case of $\approx$ is similar.

Least Upper Bound Let D be any upper bound for Δ. Then for any $A \in \Delta$, $M_A \subseteq M_D$ so $M \subseteq M_D$. Similarly $\lambda \subseteq \lambda_D$, so $\lambda = \lambda_D \upharpoonright M$.

Clearly $P \subseteq P_D$. To see that $P = P_D \cap M_{A^*}^{\circledast}$, suppose $s \in P_D \cap M_{A^*}^{\circledast}$. Then $s \in M_A^{\circledast}$ for some $A \in \Delta$, so $s \in P_D \cap M_A^{\circledast}$, so $s \in P_A$, so $s \in P$.

The case of $\approx$ is similar. ■

Lemma 5 If $A \trianglelefteq B$ and $M_A = M_B$ then $A = B$.

Proof $M_A = M_B$, so $\lambda_A = \lambda_B \upharpoonright M_A = \lambda_B$. This means that $M_A^{\circledast} = M_B^{\circledast}$, so $P_A = P_B \cap M_A^{\circledast} = P_B$, and similarly $\approx_A = \approx_B$. ■

Lemma 6 If F is a function on games, then F is continuous with respect to $\trianglelefteq$ iff F is monotone w.r.t. $\trianglelefteq$ and continuous on move sets (i.e. the action of F on move sets preserves directed unions.)

Proof This is a direct consequence of Lemma 5. ■

Using these results, together with the usual fixpoint theorem for CPOs, we can construct solutions to recursive equations $G = \Phi(G, G)$ whenever the action of $\Phi : \mathcal{G}^{\mathsf{op}} \times \mathcal{G} \to \mathcal{G}$ on objects is continuous with respect to $\trianglelefteq$. For linear implication, we have:

Proposition 7 The action of $\multimap$ on games is continuous with respect to $\trianglelefteq$

Proof We shall consider continuity in the left argument only. By Lemma 6, we just need to show that $\multimap$ is monotone on games and continuous on move sets. For monotonicity, suppose $A \trianglelefteq B$ and C is any other game. We must show that $A \multimap C \trianglelefteq B \multimap C$. For the first two conditions, notice that

$$M_{A\multimap C} = M_A + M_C \subseteq M_B + M_C = M_{B\multimap C}$$

and that

$$\lambda_{A\multimap C} = [\overline{\lambda_A}, \lambda_C] = [\overline{\lambda_B} \upharpoonright M_A, \lambda_C] = [\overline{\lambda_B}, \lambda_C] \upharpoonright (M_A + M_C) = \lambda_{B\multimap C} \upharpoonright M_{A\multimap C}$$

For the third condition, suppose $s \in P_{A\multimap C}$. Then $s \in M^{\circledast}_{A\multimap C} \subseteq M^{\circledast}_{B\multimap C}$ because of the first two conditions, and $s \upharpoonright M_A \in P_A \subseteq P_B$, $s \upharpoonright M_C \in P_C$, and the stack discipline for $\multimap$ is satisfied by s, so $s \in P_{B\multimap C}$.

Conversely, let $s \in P_{B\multimap C} \cap M^{\circledast}_{A\multimap C}$. Then $s \upharpoonright M_C \in P_C$ and $S \upharpoonright M_B \in P_B \cap M^{\circledast}_A = P_A$ and the stack discipline is satisfied by s, so $s \in P_{A\multimap C}$.

For the last condition, the left to right implication is trivial. So suppose $s \approx_{B\multimap C} s'$ and $s \in M^{\circledast}_{A\multimap C}$. Then $s \upharpoonright M_B \in M^{\circledast}_A$ and $s \upharpoonright M_B \approx_B s' \upharpoonright M_B$. Since $A \trianglelefteq B$, we get $s' \in M^{\circledast}_A$ and $s \upharpoonright M_B \approx_A s' \upharpoonright M_B$. From this it easily follows that $s \approx_{A\multimap C} s'$.

Continuity on move sets reduces to the set-theoretic statement that if Δ is a set of sets directed under $\subseteq$, $\bigcup\{A + C \mid A \in \Delta\} = \bigcup \Delta + C$. ■

It can also be shown that tensor, the 'of course' exponential and the 'with' product of linear logic are continuous with respect to $\trianglelefteq$, so we have solutions for equations involving functors built up from these constructs. It remains to show that the solution thus constructed is the canonical solution, i.e. that it is a *minimal invariant* [Fre91]. For this we need more structure on our category.

2.2 The order on morphisms

We define here a partial order on the set $\hat{A}$ for a game A. It has not yet been established whether this is in general a complete partial order, but we describe the completeness properties that it does have; these are enough for our purposes.

For history-free strategies $\sigma, \tau : A$, we define

$$\sigma \lesssim_A \tau \quad \Longleftrightarrow \quad \sigma \approx \sigma \wedge \tau \approx \tau \wedge (\forall s \in \sigma)(\exists t \in \tau)[s \approx_A t]$$

This is clearly a preorder on those strategies σ for which $\sigma \approx \sigma$, and Proposition 1 tells us that the associated equivalence relation is $\approx_A$. Therefore we obtain a partial order on $\hat{A}$, which we write as $\sqsubseteq_A$. It is easy to see that $[\{\epsilon\}]$ is the least element of this order.

Now suppose $[\sigma_1] \sqsubseteq_A [\sigma_2] \sqsubseteq_A \ldots$ is an ω-chain in $\hat{A}$. Say that it is a *strong* chain if and only if there exist strategies $\sigma'_1, \sigma'_2, \ldots$ such that for each $i \in \omega$, $\sigma'_i \approx \sigma_i$ and $\sigma'_i \subseteq \sigma'_{i+1}$.

Proposition 8 If $[\sigma_1] \sqsubseteq [\sigma_2] \sqsubseteq \ldots$ is a strong ω-chain in $\hat{A}$, then its $\sqsubseteq$-least upper bound $\bigsqcup_i [\sigma_i]$ exists, and is given by $[\bigcup \sigma'_i]$, for any chain $\sigma'_1 \subseteq \sigma'_2 \subseteq \ldots$ with $\sigma'_i \in [\sigma_i]$.

Proof Let $\sigma'_1 \subseteq \sigma'_2 \subseteq \ldots$ be an ω-chain with $\sigma'_i \approx \sigma_i$ for each i. Then it is easy to see that $\bigcup_i \sigma'_i$ is a well-defined history-free strategy, and that $\bigcup_i \sigma'_i \approx \bigcup_i \sigma'_i$. It just remains to show that $[\bigcup_i \sigma'_i]$ is indeed the lub of the chain $[\sigma_1] \sqsubseteq [\sigma_2] \sqsubseteq \ldots$.

Let $j \in \omega$, and suppose $s \in \sigma_j$. Since $\sigma_j \approx \sigma'_j$, $(\exists s' \in \sigma'_j)[s \approx_A s']$ so

$$(\forall s \in \sigma_j)(\exists s' \in \bigcup_i \sigma'_i)[s \approx_A s']$$

i.e. $[\sigma_j] \sqsubseteq [\bigcup_i \sigma'_i]$, so it is an upper bound.

Suppose $[\tau]$ is another upper bound. Let $s \in \bigcup_i \sigma'_i$. Then for some $j \in \omega$, $s \in \sigma'_j$, and $\sigma'_j \approx \sigma_j$, so $(\exists s' \in \sigma_j)[s \approx_A s']$. But we know that $[\sigma_j] \sqsubseteq [\tau]$, so $(\exists t \in \tau)[t \approx_A s' \approx_A s]$. So we have $(\forall s \in \bigcup_i \sigma'_i)(\exists t \in \tau)[s \approx_A t]$ i.e. $[\bigcup_i \sigma'_i] \sqsubseteq [\tau]$, so it is the least upper bound. ■

2.3 Inclusions and projections

Given games A and B with $A \trianglelefteq B$, define a strategy $\mathtt{in}_{A,B} : A \multimap B$ as follows:

$$\mathtt{in}_{A,B} \stackrel{\text{def}}{=} \{s \mid s \upharpoonright B = s \upharpoonright A \in P_A\}$$

Define a strategy $\mathtt{proj}_{B,A} : B \multimap A$ in the same way:

$$\mathtt{proj}_{B,A} \stackrel{\text{def}}{=} \{s \mid s \upharpoonright B = s \upharpoonright A \in P_A\}$$

So $\mathtt{in}_{A,B}$ and $\mathtt{proj}_{B,A}$ are both just $\mathtt{id}_A$ considered as strategies for $A \multimap B$ and $B \multimap A$ respectively. Proposition 7 tells us that $A \multimap A \trianglelefteq A \multimap B$ and $A \multimap A \trianglelefteq B \multimap A$, so applying Lemma 3 gives the following:

Proposition 9 If $A \trianglelefteq B$ then $\mathtt{in}_{A,B}$ and $\mathtt{proj}_{B,A}$ are well-defined history-free strategies and $\mathtt{in}_{A,B} \approx \mathtt{in}_{A,B}$, $\mathtt{proj}_{B,A} \approx \mathtt{proj}_{B,A}$.

So if $A \trianglelefteq B$, we can define a canonical *inclusion* morphism $\iota_{A,B} \stackrel{\text{def}}{=} [\mathtt{in}_{A,B}] : A \to B$, and a canonical *projection* morphism $p_{B,A} \stackrel{\text{def}}{=} [\mathtt{proj}_{B,A}] : B \to A$.

The fact that $\mathtt{in}_{A,B}$ and $\mathtt{proj}_{B,A}$ are really just $\mathtt{id}_A$ considered as a strategy for a different game also gives us:

Proposition 10 For games A and B with $A \trianglelefteq B$,

$$\begin{aligned} \mathtt{in}_{A,B};\mathtt{proj}_{B,A} &= \mathtt{id}_A \\ \mathtt{proj}_{B_1,A};\mathtt{in}_{A,B_2} &= \{s \mid s \upharpoonright B_1 = s \upharpoonright B_2 \in P_A\} \end{aligned}$$

We extend $\trianglelefteq$ to $\mathcal{G}^{\mathrm{op}} \times \mathcal{G}$ using the pointwise ordering, and define the canonical inclusion and projection for $(A, B) \trianglelefteq (A', B')$ thus:

$$\begin{aligned} \iota_{(A,B),(A',B')} &\stackrel{\text{def}}{=} (p_{A',A}, \iota_{B,B'}) \\ p_{(A',B'),(A,B)} &\stackrel{\text{def}}{=} (\iota_{A,A'}, p_{B',B}) \end{aligned}$$

We also extend the ordering on hom-sets of $\mathcal{G}$, and with it the notion of a strong chain, to hom-sets of $\mathcal{G}^{\mathrm{op}} \times \mathcal{G}$ using the pointwise ordering.

2.4 Minimal invariant property

Given a functor $\Phi : \mathcal{G}^{\mathrm{op}} \times \mathcal{G} \to \mathcal{G}$, an *invariant object* for Φ is a game D equipped with an isomorphism $i : \Phi(D, D) \cong D$.

Such an invariant is said to be *minimal* if the function

$$\delta : \mathcal{G}(D, D) \to \mathcal{G}(D, D)$$

given by

$$\delta([\sigma]) = i\Phi([\sigma], [\sigma])i^{-1}$$

has the identity morphism $[\mathtt{id}_D]$ as its unique fixed point. We show here that every such functor which obeys certain properties has such a minimal invariant, and that in those cases the minimal invariant is the one constructed above. Since we are solving up to identity in those cases, the isomorphism i is in fact the identity, which simplifies matters.

We say that Φ *preserves inclusions and projections* if, given $(A, B) \trianglelefteq (A', B')$ in $\mathcal{G}^{\mathrm{op}} \times \mathcal{G}$, $\Phi(p_{A',A}, \iota_{B,B'}) = \iota_{\Phi(A,B),\Phi(A',B')}$ and $\Phi(\iota_{A,A'}, p_{B',B}) = p_{\Phi(A',B'),\Phi(A,B)}$. Notice that this implies that Φ is monotone with respect to $\trianglelefteq$.

If Φ is monotone with respect to $\sqsubseteq$, say that Φ is *locally strong-continuous* if for any *strong* chain $([\sigma_1], [\tau_1]) \sqsubseteq ([\sigma_2], [\tau_2]) \sqsubseteq \ldots$ in a hom-set of $\mathcal{G}^{\mathrm{op}} \times \mathcal{G}$, the chain $\Phi([\sigma_1], [\tau_1]) \sqsubseteq \Phi([\sigma_2], [\tau_2]) \sqsubseteq \ldots$ is also a strong chain, and

$$\Phi(\bigsqcup_i \{([\sigma_i], [\tau_i])\}) = \bigsqcup_i \{\Phi([\sigma_i], [\tau_i])\}$$

We are now in a position to state the main theorem:

Theorem 11 Suppose $\Phi : \mathcal{G}^{\mathrm{op}} \times \mathcal{G} \to \mathcal{G}$ is a functor which is

– continuous with respect to $\trianglelefteq$

– locally strong-continuous, and

– preserves inclusions and projections.

Then Φ has a minimal invariant D given by $D = \bigsqcup_{\trianglelefteq} D_n$ where

$$\begin{aligned} D_0 &= I \\ D_{n+1} &= \Phi(D_n, D_n) \end{aligned}$$

and in fact $\Phi(D, D) = D$.

Proof Define the D_n as above. Then immediately $D_0 \trianglelefteq D_1$, and if for some n we have that $D_n \trianglelefteq D_{n+1}$, the monotonicity of Φ with respect to $\trianglelefteq$ tells us that $\Phi(D_n, D_n) \trianglelefteq \Phi(D_{n+1}, D_{n+1})$, i.e. $D_{n+1} \trianglelefteq D_{n+2}$; so by induction we have an ω-chain with respect to $\trianglelefteq$. Let $D = \bigsqcup_{\trianglelefteq} D_n$. It follows that $\Phi(D, D) = D$ by the usual argument. We must now verify the minimal invariant property.

Define a function $\delta : \mathcal{G}(D, D) \to \mathcal{G}(D, D)$ by $\delta([\sigma]) = \Phi([\sigma], [\sigma])$. We must show that the unique fixpoint of δ is $[\mathtt{id}_D]$. Let the inclusion and projection between D_n and D be i_n and p_n respectively. Then for any n,

$$\begin{aligned} \delta(p_n; i_n) &= \Phi(p_n; i_n, p_n; i_n) \\ &= \Phi(i_n, p_n); \Phi(p_n, i_n) \\ &= \Phi(p_{(D,D),(D_n,D_n)}); \Phi(\iota_{(D_n,D_n),(D,D)}) \\ &\qquad \text{by the definition of inclusions and projections in } \mathcal{G}^{\mathrm{op}} \times \mathcal{G} \\ &= p_{n+1}; i_{n+1} \quad \text{since } \Phi \text{ preserves inclusions and projections.} \end{aligned}$$

Then by Proposition 10, $p_0; i_0 = [\{\epsilon\}]$, so $\delta^n([\{\epsilon\}]) = p_n; i_n$ for each $n \in \omega$. This, together with the fact that Φ is monotone with respect to $\sqsubseteq$, tells us that we have a chain

$$[\{\epsilon\}] \sqsubseteq \delta([\{\epsilon\}]) \sqsubseteq \delta^2([\{\epsilon\}]) \sqsubseteq \cdots$$

By Proposition 10, for each n, $\mathtt{proj}_{D,D_n}; \mathtt{in}_{D_n,D} \subseteq \mathtt{proj}_{D,D_{n+1}}; \mathtt{in}_{D_{n+1},D}$ and by Proposition 2, $\mathtt{proj}_{D,D_n}; \mathtt{in}_{D_n,D} \in \delta^n([\{\epsilon\}])$, so this is in fact a strong chain, so by Proposition 8 it has a least upper bound given by $[\bigcup_n(\mathtt{proj}_{D,D_n}; \mathtt{in}_{D_n,D})]$. The local strong-continuity of Φ together with the usual considerations now tell us that this is a fixpoint for δ. By Proposition 10, for each $n \in \omega$,

$$\mathtt{proj}_{D,D_n}; \mathtt{in}_{D_n,D} \subseteq \mathtt{id}_D$$

For the other inclusion, suppose that $s \in \mathtt{id}_D$. Then $s \in P_{D \multimap D}$ and so for some n, $s \in P_{D_n \multimap D_n}$, and s is part of a copycat strategy, so by Proposition 10, $s \in \mathtt{proj}_{D,D_n}; \mathtt{in}_{D_n,D}$. Therefore

$$[\bigcup_n (\mathtt{proj}_{D,D_n}; \mathtt{in}_{D_n,D})] = [\mathtt{id}_D]$$

and so $[\mathtt{id}_D]$ is a fixpoint of δ (the least fixpoint with respect to $\sqsubseteq$).

To see that this is the unique fixpoint of δ, suppose that $[\sigma]$ is another fixpoint. Then $[\mathtt{id}_D] \sqsubseteq [\sigma]$. Suppose $[\sigma] \not\sqsubseteq [\mathtt{id}_D]$. Then for some $sab \in \sigma$ we have that $(\neg\exists tcd \in \mathtt{id}_D)[sab \approx tcd]$. Let sab be such a string of minimal length. Then for some $s' \in \mathtt{id}_D$, $s \approx s'$. By extendibility, $sa \approx s'a'$ for some a', and $s'a'a' \in \mathtt{id}_D$ (where the two occurences of a' are in different components of $D \multimap D$). So since $[\mathtt{id}_D] \sqsubseteq [\sigma]$, there exists $tcd \in \sigma$ with $s'a'a' \approx tcd$. But then both sab and tcd are in σ, and $sa \approx s'a' \approx tc$, so since $\sigma \approx \sigma$, $sab \approx tcd$. This gives us that $sab \approx s'a'a' \in \mathtt{id}_D$, contradicting our assumption. So $[\sigma] \sqsubseteq [\mathtt{id}_D]$, and therefore $[\sigma] = [\mathtt{id}_D]$. ■

This completes the main theorem. Of course, for it to be useful, we must show that the functors corresponding to the type constructors we wish to use satisfy the hypotheses of the theorem. This is indeed the case for all the constructors which have so far been translated into this setting. As before, we shall use linear implication as an example. The continuity of $\multimap$ with respect to $\trianglelefteq$ has already been established (Proposition 7), so it just remains to show that it is locally strong-continuous and preserves inclusions and projections.

Proposition 12 $(-) \multimap (+) : \mathcal{G}^{\mathrm{op}} \times \mathcal{G} \to \mathcal{G}$ is locally strong-continuous and preserves inclusions and projections.

Proof We must first show that $\multimap$ is monotone with respect to $\sqsubseteq$. To this end, suppose that $\sigma, \sigma' : A \multimap B$ and $\tau, \tau' : A' \multimap B'$ with $[\sigma] \sqsubseteq [\sigma']$ and $[\tau] \sqsubseteq [\tau']$. Suppose $s \in \sigma \multimap \tau$, and let $s_1 = s \upharpoonright A, B$ and $s_2 = s \upharpoonright A', B'$, so that $s_1 \in \sigma$ and $s_2 \in \tau$. So there exist $t_1 \in \sigma'$ and $t_2 \in \tau'$ with $t_1 \approx s_1$ and $t_2 \approx s_2$. Then interleaving t_1 and t_2 in the same way as s_1 and s_2 are interleaved in s gives us a position $t \in \sigma' \multimap \tau'$ with $t \approx s$. So $[\sigma] \multimap [\tau] \sqsubseteq [\sigma'] \multimap [\tau']$.

We now turn to strong-continuity. Suppose we have two strong chains $[\sigma_1] \sqsubseteq [\sigma_2] \sqsubseteq \ldots$ and $[\tau_1] \sqsubseteq [\tau_2] \sqsubseteq \ldots$, where for each i, $\sigma_i : A \multimap B$ and $\tau_i : A' \multimap B'$. Without loss of generality, we can assume that $\sigma_1 \subseteq \sigma_2 \subseteq \ldots$ and $\tau_1 \subseteq \tau_2 \subseteq \ldots$. Then from the definition of $\multimap$ as a concrete operation on strategies, it is not hard to see that $\sigma_1 \multimap \tau_1 \subseteq \sigma_2 \multimap \tau_2 \subseteq \ldots$, so the chain produced by the application of $\multimap$ is a strong one. Its least upper bound is therefore given by

$$\bigcup_i \{\sigma_i \multimap \tau_i\} = \bigcup_i \{s \mid s \upharpoonright A, B \in \sigma_i,\ s \upharpoonright A', B' \in \tau_i\}$$

whereas applying $\multimap$ to the lubs of the original chains gives

$$\bigcup_i \sigma_i \multimap \bigcup_j \tau_j = \{s \mid s \upharpoonright A, B \in \bigcup_i \sigma_i,\ s \upharpoonright A', B' \in \bigcup_j \tau_j\}$$

But the fact that the σ_i and the τ_j form subset chains tells us that these two sets are equal.

To see that $\multimap$ preserves inclusions and projections, suppose $(A, B) \trianglelefteq (A', B')$. It suffices to show that $\mathtt{proj}_{A',A} \multimap \mathtt{in}_{B,B'} = \mathtt{in}_{(A \multimap B),(A' \multimap B')}$ and that $\mathtt{in}_{A,A'} \multimap \mathtt{proj}_{B',B} = \mathtt{proj}_{(A' \multimap B'),(A \multimap B)}$ (for some constructs, the analogous statement will not be true; in particular, when considering the exponential !, we only have that $!\,\mathtt{in}_{A,A'} \approx \mathtt{in}_{!A,!A'}$, but this is, of course, enough). But these are simple consequences of the facts that the inclusions and projections are (as sets) the same as the identity strategies, and that the action of $\multimap$ on morphisms is defined by a concrete operation on sets which respects $\subseteq$; the functoriality of $\multimap$ tells us that we get a strategy which (as a set) is equal to an identity strategy, which is therefore the required inclusion or projection. ■

Notice that the proof of this result is simple given that the functor is defined by a concrete operation on strategies which, when we consider strategies as sets, preserves $\subseteq$. The same is true of all the constructs which have been defined on games so far, and so similarly simple proofs can be given for these constructs.

3 Lifting and separated sum

We give here simple constructions corresponding to the usual lifting and separated sum operations on games. Lifting is, of course, a special case of the separated sum, but we describe it separately because of its special properties; in particular, it has the structure of a strong monad, as is expected [Mog91].

For lifting, the idea is to add to a game an initial protocol in which the Opponent asks a question, after which Player has only one available move, which is an answer to the question. Once this answer has been given, play continues as if the original game had just begun. A strategy for Player may or may not have a response to the initial question. If it does not, it is the empty strategy and therefore the bottom element of the domain. If it does, it corresponds to a strategy in the original, unlifted game.

Given a game $A = (M_A, \lambda_A, P_A, \approx_A)$, define $A_\perp = (M_{A_\perp}, \lambda_{A_\perp}, P_{A_\perp}, \approx_{A_\perp})$ as follows:

$$\begin{aligned} M_{A_\perp} &= \{\circ, \bullet\} + M_A \\ \lambda_{A_\perp} &= [\{\circ \mapsto OQ, \bullet \mapsto PA\}, \lambda_A] \\ P_{A_\perp} &= \{\epsilon, \circ\} \cup \{\circ \bullet s \mid s \in P_A\} \end{aligned}$$

$$
\begin{aligned}
s \approx_{A_\perp} s' \quad \text{iff} \quad & s = s' = \epsilon \quad \text{or} \\
& s = s' = \circ \quad \text{or} \\
& s = \circ \bullet t \text{ and } s' = \circ \bullet t' \text{ and } t \approx_A t'
\end{aligned}
$$

It is clear how this can be generalised to give the separated sum of n games, $A_1 + A_2 + \ldots + A_n$: there are n possible answers to the first question, one corresponding to each component game. If the answer corresponding to A_i is given, play continues as in A_i.

We now give a strategy $\eta_A : A \multimap A_\perp$ and an operation $(-^*)$ taking a strategy $\sigma : A \multimap B_\perp$ to a strategy $\sigma^* : A_\perp \multimap B_\perp$ which make $(-_\perp, \eta, -^*)$ into a Kleisli triple.

For a game A, $\eta_A : A \multimap A_\perp$ is defined by

$$\eta_A = \{\epsilon\} \cup \{\circ \bullet s \mid \circ \bullet s \in P_{A \multimap A_\perp} \wedge s \upharpoonright A = s \upharpoonright A_\perp\}$$

For a strategy $\sigma : A \multimap B_\perp$, the strategy $\sigma^* : A_\perp \multimap B_\perp$is defined by

$$\sigma^* = \{\epsilon, \circ_B \circ_A\} \cup \{\circ_B \circ_A \bullet_A s \mid \circ_B s \in \sigma\}$$

We now state without proof the facts that we need in order to make this a good definition:

Proposition 13

1. $-^*$ is compatible with $\approx$ and hence defines an operation on morphisms.
2. $(-_\perp, \eta, -^*)$ is a Kleisli triple in $\mathcal{G}$.

Because we have a Kleisli triple, we can make $-_\perp$ into a functor by defining its action on strategies in the usual way: if $\sigma : A \multimap B$, then $\sigma_\perp \stackrel{\text{def}}{=} (\sigma; \eta_B)^* : A_\perp \multimap B_\perp$. It can be shown that $\sigma_\perp : A_\perp \multimap B_\perp$ is given by

$$\{\epsilon, \circ_B \circ_A\} \cup \{\circ_B \circ_A \bullet_A \bullet_B s \mid s \in \sigma\}$$

As usual, this is a concrete operation on the sets representing strategies, and it preserves $\subseteq$, so it is reasonably simple to establish that the lifting functor satisfies the hypotheses of Theorem 11. The action of separated sum on morphisms is defined in an analogous way. For the binary case, suppose $\sigma : A \multimap B$ and $\tau : A' \multimap B'$. If we denote the initial question of $A + A'$ as $\circ_A$ and the two possible answers as a and a', and similarly introduce moves $\circ_B$, b and b' for $B + B'$, then $\sigma + \tau : A + A' \multimap B + B'$ is given by

$$\{\epsilon, \circ_B \circ_A\} \cup \{\circ_B \circ_A abs \mid s \in \sigma\} \cup \{\circ_B \circ_A a'b's \mid s \in \tau\}$$

Proposition 14 The functors $-_\perp$ and $+$ are locally strong-continuous; their actions on objects are continuous with respect to $\trianglelefteq$, and their actions on morphisms preserve inclusions and projections.

Notice that $+$ is a bifunctor which is covariant in both arguments, so here we are really talking about the family of functors $A + (-)$ and $(-) + A$ for some game A.

It just remains to say what the tensorial strength for the lifting monad is. Given games A and B, the strategy $t_{A,B} : A \otimes B_\perp \multimap (A \otimes B)_\perp$ is the obvious "copycat" strategy between these two games, bearing in mind that the set of moves of each is the disjoint union of M_A, M_B and $\{\circ, \bullet\}$. Verification that this defines a tensorial strength is straightforward, and is omitted.

We can also give a characterisation of $(-)_\perp$ as the left adjoint to a certain functor. Let $\mathcal{G}_\perp$ be the category whose objects are the empty game I and games which have a unique first move, and whose morphisms are equivalence classes of strategies $\sigma : A \multimap B$ which respond to the first move of B with the first move of A. It is not hard to see that this is a well defined category: composition and identity are as in $\mathcal{G}$. Then let $U : \mathcal{G}_\perp \to \mathcal{G}$ be the forgetful functor. It can be shown that $(-)_\perp$, considered as a functor from $\mathcal{G}$ to $\mathcal{G}_\perp$, is left adjoint to U, and in fact $\mathcal{G}_\perp$ is the Eilenberg-Moore category of algebras for $(-)_\perp$.

4 Examples: the natural numbers

Now that lifting and separated sum have been defined, it is possible for us to define the 'vertical' and 'lazy' natural numbers types by the recursive equations $\mathbf{N}_{\mathrm{vert}} = (\mathbf{N}_{\mathrm{vert}})_\perp$ and $\mathbf{N}_{\mathrm{lazy}} = I + \mathbf{N}_{\mathrm{lazy}}$. More precisely, we solve the equations $\mathbf{N}_{\mathrm{vert}} = \Phi_{\mathrm{vert}}(\mathbf{N}_{\mathrm{vert}}, \mathbf{N}_{\mathrm{vert}})$ and $\mathbf{N}_{\mathrm{lazy}} = \Phi_{\mathrm{lazy}}(\mathbf{N}_{\mathrm{lazy}}, \mathbf{N}_{\mathrm{lazy}})$, where the functors $\Phi_{\mathrm{vert}}, \Phi_{\mathrm{lazy}} : \mathcal{G}^{\mathrm{op}} \times \mathcal{G} \to \mathcal{G}$ are defined by

$$\begin{array}{rclrl} \Phi_{\mathrm{vert}}(A,B) & = & B_\perp & \Phi_{\mathrm{vert}}([\sigma],[\tau]) = & [\tau_\perp] \\ \Phi_{\mathrm{lazy}}(A,B) & = & I + B & \Phi_{\mathrm{lazy}}([\sigma],[\tau]) = & \mathtt{id}_I + [\tau] \end{array}$$

for games A and B and strategies σ and τ of suitable types. These functors satisfy the hypotheses of Theorem 11, so we obtain our solutions as the least upper bounds of the sequences $V_0 \trianglelefteq V_1 \trianglelefteq \ldots$ and $L_0 \trianglelefteq L_1 \ldots$ of games, where $V_0 = L_0 = I$ and $V_{i+1} = \Phi_{\mathrm{vert}}(V_i, V_i)$ and $L_{i+1} = \Phi_{\mathrm{lazy}}(L_i, L_i)$, for all $i \in \omega$.

Consider the vertical natural numbers first. We have $V_1 = I_\perp$, $V_2 = I_{\perp\perp}$ and so on. So the set of moves of V_i for a given i consists of i disjoint copies of $\{\circ, \bullet\}$. Refer to these as $\circ_1, \bullet_1, \circ_2, \bullet_2, \ldots$. The labelling function is the obvious one, and the set of plays of V_i then consists of all prefixes of $\circ_1 \bullet_1 \circ_2 \bullet_2 \ldots \circ_i \bullet_i$. The

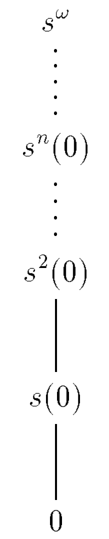

Figure 1: $\mathbf{N}_{\text{vert}}$: the vertical natural numbers

equivalence relation on each V_i is just the identity relation. So the least upper bound of the chain $V_0 \trianglelefteq V_1 \trianglelefteq \ldots$ is the game $\mathbf{N}_{\text{vert}}$ where

$$\begin{array}{rcl} M_{\mathbf{N}_{\text{vert}}} & = & \bigcup_{i=1}^{\omega} \{\circ_i, \bullet_i\} \\ \lambda_{\mathbf{N}_{\text{vert}}} & = & \bigcup_{i=1}^{\omega} \{\circ_i \mapsto OQ, \bullet_i \mapsto PA\} \\ P_{\mathbf{N}_{\text{vert}}} & = & \{s \mid s \text{ is a prefix of } \circ_1 \bullet_1 \circ_2 \bullet_2 \ldots\} \\ s \approx_{\mathbf{N}_{\text{vert}}} s' & \iff & s = s' \end{array}$$

So the strategies for $\mathbf{N}_{\text{vert}}$ are those of the form

$$\{s \mid |s| \text{ even and } s \text{ is a prefix of } \circ_1 \bullet_1 \ldots \circ_n \bullet_n\}$$

for each n, which we call $s^n(0)$ (in particular, the empty strategy is called 0) and the strategy $\{s \mid |s| \text{ even and } s \text{ is a prefix of } \circ_1 \bullet_1 \circ_2 \bullet_2 \ldots\}$, which we call s^ω (this is the only infinite strategy). It is then easy to see that the order $\sqsubseteq$ on these strategies gives us the poset $(\mathbf{N}_{\text{vert}}, \sqsubseteq)$ shown in figure 1.

The diagrams in this paper were produced using Paul Taylor's commutative diagram package, for which the authors are grateful.

Now we come to the lazy natural numbers. We have $L_0 = I$, $L_1 = I + I$, $L_2 = I + (I + I)$ and so on. So for $i \in \omega$, $L_i = I + (I + (I + \ldots + (I + I))\ldots)$, with $i + 1$ occurrences of I. Suppose the initial question of a game $A + B$ is $\circ$, followed by answer $\nwarrow$ for A or $\nearrow$ for B. Then the move set of L_i is given by

$$M_{L_i} = \{\circ, \nwarrow, \nearrow\} + (\{\circ, \nwarrow, \nearrow\} + (\ldots + \{\circ, \nwarrow, \nearrow\}))$$

with i copies of $\{\circ, \nwarrow, \nearrow\}$. For convenience, refer to the leftmost copy as $\{\circ_0, \nwarrow_0, \nearrow_0\}$, the next as $\{\circ_1, \nwarrow_1, \nearrow_1\}$ and so on. Of course $\lambda_{L_i}(\circ_j) = OQ$ and $\lambda_{L_i}(\nwarrow_j) = \lambda_{L_i}(\nearrow_j) = PA$ for any $j < i$. The set of positions of L_i is given by

$$\begin{aligned} P_{L_i} \quad = \quad & \{s \mid s \text{ is a prefix of } \circ_0 \nearrow_0 \circ_1 \nearrow_1 \ldots \circ_{i-1} \nearrow_{i-1}\} \\ & \cup \{\circ_0 \nearrow_0 \circ_1 \nearrow_1 \ldots \circ_{n-1} \nearrow_{n-1} \circ_n \nwarrow_n \mid 0 \leq n < i\} \end{aligned}$$

The equivalence relation is just the identity relation. So it is not hard to see that the least upper bound of the chain $L_0 \trianglelefteq L_1 \trianglelefteq \ldots$ is the game $\mathbf{N}_{\text{lazy}}$ where

$$\begin{aligned} M_{\mathbf{N}_{\text{lazy}}} \quad &= \quad \bigcup_{i \in \omega} \{\circ_i, \nwarrow_i, \nearrow_i\} \\ \lambda_{\mathbf{N}_{\text{lazy}}} \quad &= \quad \bigcup_{i \in \omega} \{\circ_i \mapsto OQ, \nwarrow_i \mapsto PA, \nearrow_i \mapsto PA\} \\ P_{\mathbf{N}_{\text{lazy}}} \quad &= \quad \{s \mid |s| \text{ even and } s \text{ is a prefix of } \circ_0 \nearrow_0 \circ_1 \nearrow_1 \ldots\} \cup \\ & \qquad \{\circ_0 \nearrow_0 \circ_1 \nearrow_1 \ldots \circ_{n-1} \nearrow_{n-1} \circ_n \nwarrow_n \mid n \in \omega\} \\ s \approx_{\mathbf{N}_{\text{lazy}}} s' \quad &\iff \quad s = s' \end{aligned}$$

Therefore a finite strategy for this game is determined by the longest string it contains, and there is only one infinite strategy (consisting of all even length prefixes of $\circ_0 \nearrow_0 \circ_1 \nearrow_1 \ldots$). Furthermore, the ordering $\sqsubseteq$ coincides with the prefix ordering on the longest strings. We denote by $s^n(\perp)$ the strategy whose longest string is $\circ_0 \nearrow_0 \ldots \circ_n \nearrow_n$ and refer to the strategy whose longest string is $\circ_0 \nearrow_0 \ldots \circ_n \nwarrow_n$ as $s^n(0)$. The infinite strategy is denoted by s^ω. Then the poset $(\mathbf{N}_{\text{lazy}}, \sqsubseteq)$ is as depicted in figure 2, and is clearly the same as the traditional lazy natural numbers domain.

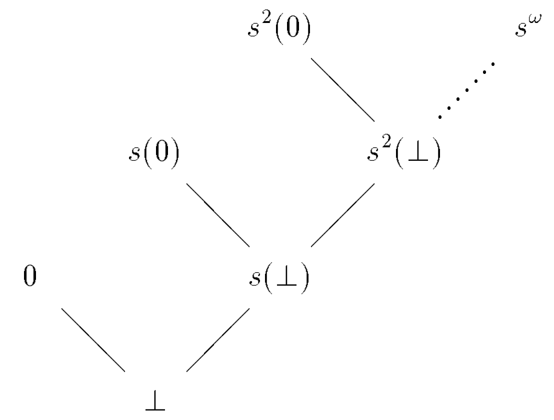

Figure 2: $\mathbf{N}_{\mathrm{lazy}}$: the lazy natural numbers

References

[AJ94] Samson Abramsky and Radha Jagadeesan. Games and full completeness for multiplicative linear logic. *Journal of Symbolic Logic*, 59(2):543 – 574, June 1994.

[AJM94] Samson Abramsky, Radha Jagadeesan, and Pasquale Malacaria. Full abstraction for PCF (extended abstract). In Masami Hagiya and John C. Mitchel, editors, *Theoretical Aspects of Computer Software. International Symposium TACS'94*, number 789 in Lecture Notes in Computer Science, pages 1–15, Sendai, Japan, April 1994. Springer-Verlag.

[Eda93] A. Edalat. Self-duality, minimal invariant objects and Karoubi invariance in information categories. In Geoffrey L. Burn, Simon J. Gay, and Mark D. Ryan, editors, *Theory and Formal Methods 1993: Proceedings of the First Imperial College Department of Computing Workshop on Theory and Formal Methods*. Springer-Verlag Workshops in Computer Science, 1993.

[ES93] A. Edalat and M. Smyth. I-categories as a framework for solving domain equations. *Theoretical Computer Science*, 115:77–106, 1993.

[Fre91] P. J. Freyd. Algebraically complete categories. In A. Carboni et al., editors, *Proc. 1990 Como Category Theory Conference*, pages 95–104, Berlin, 1991. Springer-Verlag. Lecture Notes in Mathemetics Vol. 1488.

[HO94] M. Hyland and C.H.L. Ong. Observational full abstraction for pcf: dialogue games and innocent strategies. Draft, 1994.

[Mog91] E. Moggi. Notions of computation and monads. *Information and Computation*, 93:55–92, 1991.

[Pit93] A. M. Pitts. Relational properties of domains. Technical Report 321, Cambridge Univ. Computer Laboratory, December 1993. 37 pages.