Games on Graphs and Sequentially Realizable Functionals
Extended Abstract

Martin Hyland
DPMMS
Centre for Mathematical Sciences
Wilberforce Road, Cambridge, CB3 0WB, UK
M.Hyland@dpmms.cam.ac.uk

Andrea Schalk
Department of Computer Science
University of Manchester
Oxford Road, Manchester, M13 9PL, UK
A.Schalk@cs.man.ac.uk

Abstract

We present a new category of games on graphs and derive from it a model for Intuitionistic Linear Logic. Our category has the computational flavour of concrete data structures but embeds fully and faithfully in an abstract games model. It differs markedly from the usual Intuitionistic Linear Logic setting for sequential algorithms. However, we show that with a natural exponential we obtain a model for PCF essentially equivalent to the sequential algorithms model. We briefly consider a more extensional setting and the prospects for a better understanding of the Longley Conjecture.

1 Introduction

Ehrhard's striking result that his hypercoherence model for PCF can be obtained as the extensional collapse of the sequential algorithms [6] was the first evidence that there is a natural class of those extensional functionals of higher type which can be computed by a sequential program. Other characterizations of this class have been given by van Oosten [12] and by Longley whose substantial paper [10] describes equivalences which support the identification of this canonical class. We call the idea that there is just one natural such class (and its identification) the Longley Conjecture[1] and in this paper we present some further evidence for it. We provide a new approach to the sequentially realizable functionals via a new notion of computation. We hope this latter will serve to clarify the relation between other characterizations.

We seek a notion of higher type computation which on the one hand is not too far from the sequential algorithm model and on the other relates well to a variety of game models, both concrete and abstract. The hypercoherence model is an example of an abstract games model, but we can make the connections (more directly) between our models and the versions of abstract games described in our [7] and [8]. Observe that a related model is described from a different point of view at the end of [1]. The graph games model we analyse here embeds fully and faithfully into these abstract games models. On the other hand, in recent related work Melliés [11] has given a direct analysis of the relation between the sequential algorithms model and the hypercoherence model. It seems likely that his insights can be exploited to round out the picture we present here.

The notion of computation we use is given by *games on graphs*. This is a very concrete model which differs from previous models in that the strategies (programs) operate on the position (data) reached and not on the details of how the data was obtained. In this it is faithful to the concrete data structure conception of computation (see [2, 9]). In [7] we explained the connections with abstract games. Here we concentrate on the sequential algorithms point of view, and connections with the Longley Conjecture.

2 Graph Games

While traditionally games have been formalized via game *trees*, there is a way of using *graphs* as the underlying structure codifying the rules of playing. This allows there to be different routes (sequences of moves) to the same position. If we restrict the graphs slightly we obtain a category which is a model for Linear Logic. We assume that there are two entities playing games, referred to as Player (P) and Opponent (O).

Definition 1 *A* graph game A *is given by*

- *a set* $A = A_P + A_O$ *of* positions *together with an* initial position $*_A \in A_P$*;*
  - A_P *is the set of* Player positions *(where Opponent is to move) and*

[1] See Sections 12.2 and 12.3 in [10] for a discussion.

  - A_O *is the set of* Opponent positions *(where Player is to move);*

- *the structure $a \longrightarrow a'$ on A of an acyclic directed (A_P, A_O)-bipartite graph (so $a \longrightarrow a'$ implies that $a \in A_P$ if and only if $a' \in A_O$) such that for any $a \in A$, $\{a' \in A \mid a' \longrightarrow^* a\}$ is well-founded with respect to $(\longrightarrow^*)^{\mathsf{op}}$, that is there are no infinite sequences $*_A \longrightarrow a_1 \longrightarrow a_2 \longrightarrow \cdots$ satisfying $a_i \longrightarrow a$ for all $i \in \mathbb{N}$.*

The graph structure describes the valid moves from each position. We think of a game A as being played from $*_A$ with Opponent making the first move: positions which are not reachable from $*_A$ play no part in the game and we could assume that they have been deleted. (But then we would have to insist on reachability in all constructions on games, and that obscures the connection with abstract games.) A P-move is a move to a P-position, an O-move is one to an O-position. We sometimes speak of the 'colour' of positions or moves in this context. A *finite play* of A consists of a sequence of positions $*_A = a_0 a_1 \cdots a_n$ such that for all $0 \leq i < n$, $a_i \longrightarrow a_{i+1}$, in other words it is a sequence of valid moves. We sometimes use $a \longrightarrow^* a'$ to indicate that there is a valid sequences of moves leading from a to a', and we use $a \longrightarrow^+ a'$ if that sequence consists of at least one move, thus ruling out $a = a'$.

Assume we have a partial function $\alpha\colon A_O \rightharpoonup A_P$ satisfying $a \longrightarrow \alpha(a)$ whenever $\alpha(a)$ is defined. We inductively define the set $R(\alpha)$ of *reachable positions* for α:

- $*_A \in R(\alpha)$;
- if $a \in R(\alpha) \cap A_P$ and $a \longrightarrow a'$ then $a' \in R(\alpha)$;
- if $a \in R(\alpha) \cap A_O$ and $\alpha(a) = a'$ (and thus $a \longrightarrow a'$) then $a' \in R(\alpha)$.

Since we often need to distinguish between the Player and the Opponent positions occurring in $R(\alpha)$ we use $R_P(\alpha)$ to denote the former and $R_O(\alpha)$ for the latter.

Definition 2 *A* pre-strategy on *A (for Player) is given by a partial map $\alpha\colon A_O \rightharpoonup A_P$ such that $a \longrightarrow \alpha(a)$ whenever $\alpha(a)$ is defined and such that the domain of definition of α is a subset of $R(\alpha)$.*

A strategy *on a graph game is a* conflict-free[2] *pre-strategy, that is a pre-strategy $\alpha\colon A_O \rightharpoonup A_P$ such that whenever $a' \in R_P(\alpha)$ is reachable from $a \in R_O(\alpha)$ then $\alpha(a)$ is defined and a' is reachable from $\alpha(a)$.*

The idea is that if Player, when playing according to some strategy α, is prepared to reach a position $a \in R_P(\alpha)$ then he should be prepared to do so from every position that might come up in a play in accord with α. We briefly describe a concrete data structure intuition. A P-position a' can be thought of as representing a situation in which certain specified cells have all been filled by Player (in an order determined by Opponent). The conflict-freeness condition ensures that Player will fill these cells with the same values no matter in which order Opponent presents them. For if a is an O-position with $a \longrightarrow^* a'$ then a describes a situation where some of the cells filled in a' have been filled already (with the correct values), and Opponent has just presented a further cell (which is also filled in a'). Now $\alpha(a) \longrightarrow^* a'$ means that Player has to fill that cell with the value prescribed by a'.

If α is a strategy for a game A then a *finite play p in accord with* α is a sequence of positions $p = a_0 a_1 a_2 \cdots a_n$ with $a_0 = *_A$ such that

- for $0 \leq i < n$ it is the case that $a_i \longrightarrow a_{i+1}$ and
- for $1 \leq i < n$ odd it is the case that $\alpha(a_i) = a_n$.

We write Π_α for the set of plays in accord with α and $p \in \Pi_\alpha$ for such a play. It is then the case that $\alpha(a) = a'$ if and only if there exists $p \in \Pi_\alpha$ ending with these two positions, that is $p = \cdots a a'$.

2.1 Multiplicative structure

In order to describe the category of graph games we describe the multiplicative structure. The tensor unit $\mathbf{I}$ is the game with just one position which is initial.

The tensor $A \otimes B$ is the game with

- P-positions $A_P \times B_P$;
- O-positions $(A_P \times B_O) + (A_O \times B_P)$.

The initial position is $(*_A, *_B)$ and there are moves $(a, b) \longrightarrow (a', b')$ just when

either $a \longrightarrow a'$ and $b = b'$ 'a move in A'
or $a = a'$ and $b \longrightarrow b'$ 'a move in B'.

The linear function space $A \multimap B$ is the game with

- P-positions $(A_P \times B_P) + (A_O \times B_O)$;
- O-positions $A_P \times B_O$.

The initial position is $(*_A, *_B)$ and there are moves $(a, b) \longrightarrow (a', b')$ just when

either $a \longrightarrow a'$ and $b = b'$ 'a move in A'
or $a = a'$ and $b \longrightarrow b'$ 'a move in B'.

As a consequence of this definition, a play of $A \multimap B$ consists of the merging of a play from A and B subject to the following rules:

[2] The terminology hints at a connection with concrete data structures [9] and event structures [13], see also [3].

- The first move is an O-move in B which is an O-move in $A \multimap B$;
- a P-move in $A \multimap B$ which is given by a P-move in B can only be followed by an O-move in B (which is an O-move in $A \multimap B$);
- a P-move in $A \multimap B$ which is given by an O-move in A can only be followed by a P-move in A (which is an O-move in $A \multimap B$.

Therefore the linear function space for graph games is subject to the usual 'switching' conditions, that is, only Player is allowed to switch between games.

The category of graph games has as objects graph games; the morphisms $A \longrightarrow B$ are the strategies on $A \multimap B$. The critical problem is to show that the composition $\sigma \circ \rho$ of $\rho \colon A \longrightarrow B$ and $\sigma \colon B \longrightarrow C$ defined by parallel composition and hiding is well-defined and associative.

To give the definition we consider *combined sequences* of triples of positions of the form (a, b, c) where

$$(a, b) \in A \multimap B \text{ and } (b, c) \in B \multimap C;$$

we can view these as 'combined positions'. The possible patterns therefore are

$$(P, P, P), \qquad (P, P, O), \qquad (P, O, O), \qquad (O, O, O).$$

The sequences under consideration are of the form $u = u_0 u_1 \cdots u_n$ where each u_i is of the form (a_i, b_i, c_i) a triple as described above, where $u_0 = (*_A, *_B, *_C)$ and each u_{i+1} differs from u_i by just one move in either A, B, or C, that is

$$\begin{array}{ll} \text{either} & a_i \longrightarrow a_{i+1} \text{ and } b_i = b_{i+1} \text{ and } c_i = c_{i+1}; \\ \text{or} & b_i \longrightarrow b_{i+1} \text{ and } a_i = a_{i+1} \text{ and } c_i = c_{i+1}; \\ \text{or} & c_i \longrightarrow c_{i+1} \text{ and } a_i = a_{i+1} \text{ and } b_i = b_{i+1}. \end{array}$$

The pattern of possible changes is then

$$(P, P, P) \longleftrightarrow (P, P, O) \longleftrightarrow (P, O, O) \longleftrightarrow (O, O, O).$$

Given a sequence we write $u|_{A \multimap B}$ for the sequence of projections of the u_i to $A \times B$ with repetitions deleted (creating an effective projection, just onto $A \multimap B$ this time) and similarly for $u|_{B \multimap C}$: so the moves in $u|_{A \multimap B}$ ($u|_{B \multimap C}$) are the moves in u which are moves in $A \multimap B$ ($B \multimap C$). In particular $u|_{A \multimap B}$ is a play in $A \multimap B$ and $u|_{B \multimap C}$ is a play in $B \multimap C$. Furthermore

$$(u|_{A \multimap B})|_B = (u|_{B \multimap C})|_B$$

is a play in B. Conversely given a play r in $A \multimap B$ and a play s in $B \multimap C$ such that $r|_B = s|_B$ there is a unique combined sequence u as above with

$$u|_{A \multimap B} = r \qquad \text{and} \qquad u|_{B \multimap C} = s.$$

Definition 3 *Let $\rho \colon A \longrightarrow B$ and $\sigma \colon B \longrightarrow C$. A* combined sequence according to ρ and σ *is given by a sequence of triples (a_0, b_0, c_0), (a_1, b_1, c_1), $\ldots$ from $A \times B \times C$ satisfying the following conditions:*

It starts at $(a_0, b_0, c_0) = (*_A, *_B, *_C)$.

We make a case distinction here for (a_i, c_i) in

- *$(A \multimap C)_P$:*
  - *(a_i, b_i, c_i):(P, P, P): then $a_i = a_{i+1}$, $b_i = b_{i+1}$, $c_i \longrightarrow c_{i+1}$;*
  - *(a_i, b_i, c_i):(O, O, O): then $a_i \longrightarrow a_{i=1}$ and $b_i = b_{i+1}$, $c_i = c_{i+1}$;*
- *$(A \multimap C)_O$:*
  - *(a_i, b_i, c_i):(P, P, O): then $a_i = a_{i+1}$ and $\sigma(b_i, c_i) = (b_{i+1}, c_{i+1})$;*
  - *(a_i, b_i, c_i):(P, O, O): then $c_i = c_{i+1}$ and $\rho(a_i, b_i) = (a_{i+1}, b_{i+1})$.*

Remark. (a) In a sequence according to ρ and σ, for all i

$$\begin{array}{lll} (a_i, b_i) \longrightarrow^* (a_{i+1}, b_{i+1}) & \text{in} & A \multimap B \text{ and} \\ (b_i, c_i) \longrightarrow^* (b_{i+1}, c_{i+1}) & \text{in} & B \multimap C. \end{array}$$

(b) If u is a combined sequence according to ρ and σ then $u|_{A \multimap B}$ is a play according to ρ in $A \multimap B$ and $u|_{B \multimap C}$ is a play according to σ in $B \multimap C$. Further the plays $u|_{A \multimap B}|_B$ and $u|_{B \multimap C}|_B$ coincide.

We need to establish that a strategy defined via combined sequences as indicated above is well-defined and conflict-free. For these purposes we need a number of lemmas.

Lemma 2.1 *Let*

$$u = (a_0, b_0, c_0),\ (a_1, b_1, c_1),\ \ldots\ ,\ (a_{n+1}, b_{n+1}, c_{n+1})$$

be a combined sequence according to ρ and σ. Suppose $(a_n, c_n) \in (A \multimap C)_O$, and suppose there is (a', b', c') with

- *$(a', b') \in R(\rho)$,*
- *$(b', c') \in R(\sigma)$ and*
- *$(a', c') \in (A \multimap C)_P$.*

If $a_n \longrightarrow^* a'$ and $c_n \longrightarrow^* c'$ then

$$(a_{n+1}, b_{n+1)} \longrightarrow^* (a', b') \text{ and } (b_{n+1}, c_{n+1}) \longrightarrow^* (b', c').$$

Corollary 2.2 *Let*

$$\begin{array}{c} (a_0, b_0, c_0), \ldots , (a_m, b, c_m),\ (a_{m+1}, b, c_{m+1}) \\ \text{and } (a'_0, b'_0, c'_0), \ldots , (a'_n, b', c'_n),\ (a'_{n+1}, b', c'_{n+1}) \end{array}$$

be sequences according to ρ and σ with

$$(a_m, c_m) = (a'_n, c'_n) = (a, c) \in (A \multimap C)_O.$$

Then

- $(a_{m+1}, c_{m+1}) = (a'_{n+1}, c'_{n+1})$;
- $b = b'$.

Now we have enough information in place to make the following definition. If $\rho\colon A \longrightarrow B$ and $\sigma\colon B \longrightarrow C$ then their composite, $\sigma \circ \rho$, is defined by

$$(\sigma \circ \rho)(a, c) = (a', c')$$

if and only if there is a sequence according to ρ and σ ending in

$$\ldots\ ,\ (a, b, c),\ (a', b, c')$$

for some b.

Corollary 2.3 *Let (a_0, c_0), (a_1, b_1), $\ldots$, (a_m, c_m) be a play in $A \multimap C$ in accord with $\sigma \circ \rho$. Then there is a unique sequence according to ρ and σ of the form*

$$(a'_0, b'_0, c'_0), \ldots\ ,\ (a'_{n-1}, b', c'_{n-1}),\ (a'_n, b', c'_n)$$

whose effective projection to $A \multimap C$ is the given play. Thus $\sigma \circ \rho$ is a well-defined pre-strategy.

Finally we check conflict-freeness for which we use the well-foundedness condition.

Lemma 2.4 *The composite of two conflict-free strategies is conflict-free.*

We explain associativity of composition as follows. If $\rho\colon A \longrightarrow B$ and $r \in \Pi_\rho$ then r is an interleaving of sequences $r|_A$ of positions in A and $r|_B$ of positions in B. We can then regard this interleaving $r\colon r|_A \longrightarrow r|_B$ as a map in a category $\mathbf{DTI}$ described in more detail in Appendix A. Strategies are determined by the set of plays to which they give rise. Thus we can identify ρ with a set of maps

$$\Pi_\rho = \{r\colon r|_A \longrightarrow r|_B\}$$

with various domains and codomains.

For $\rho\colon A \longrightarrow B$ and $\sigma\colon B \longrightarrow C$ we define the plays in accord with $\sigma \circ \rho$ via composition and hiding; we show that $\sigma \circ \rho$ is a strategy and moreover

$$\begin{aligned}\Pi_{\sigma\circ\rho} \ = \{s \circ r \mid\ \ & r \in \Pi_\rho,\ s \in \Pi_\sigma, \text{ and} \\ & \mathsf{dom} s = \mathsf{cod} r\}\end{aligned}$$

is the set of compositions of interleavings, thought of as maps in $\mathbf{DTI}$.

Since composition in $\mathbf{DTI}$ is associative it is now immediate that composition of strategies is associative. The usual copy-cat strategy is identified with a full set of identities and so acts as the identity under composition. Now the explicit description of the multiplicative structure makes the following routine.

Proposition 2.5 *The category of graph games is symmetric monoidal closed.*

2.2 Additive structure

There is more categorical structure to be found in the category of graph games. The additive structure does not bring any surprises.

The terminal object $\mathbf{1}$ is (again) the game $\mathbf{I}$ with just one (initial) position.

The product $A \times B$ is the 'coalesced sum' of the games A and B, identifying $*_A$ with $*_B$ to give the new initial position. Player and Opponent positions and moves are all inherited from A and B. Projections are given by copy-cat type strategies.

Proposition 2.6 *The category of graph games has arbitrary products.*

2.3 Exponential structure

To obtain a model of (Intuitionistic) Linear Logic[3] we have to restrict ourselves to a full subcategory.

Definition 4 *We say that a graph game is* regulated *if and only if the following conditions are satisfied. If a and a' are of the same colour then $a \longrightarrow^* a' \longrightarrow \bar{a}'$ and $a \longrightarrow \bar{a} \longrightarrow^+ \bar{a}'$ together imply $\bar{a} \longrightarrow^* a'$. Let $\mathbf{GGam}$ be the full subcategory of regulated graph games.*

Being regulated is a condition on how plays split and merge in a graph game: If a from the definition is a P-position then any two plays splitting at a merge for the first time at a P-position (either a' or an earlier P-position). Again there is a concrete data structure intuition. From a P-position play continues with Opponent presenting Player with another cell to fill. Plays split if Opponent presents different cells. Plays merge if the same cells have been presented by Opponent in different orders, but have been filled by Player with the same values. Hence the last move before the plays merge once more must be a value placed by Player.

From now on when we refer to a 'graph game' we will tacitly assume that we speak of a regulated one.

Proposition 2.7 *The category* $\mathbf{GGam}$ *(of regulated graph games) is closed under tensor, linear function space and products.*

Regulated[4] games have an interesting property. Let a strategy on $A \multimap B$ be given by its reachable positions (which are pairs consisting of a position in A and a position in B), and read these as a relation from positions in A to positions in B. Then composition of strategies is given by relational composition. Thus there is a good connection with the relational model of Linear Logic and so with categories of abstract games based on it.

[3] See for example [8] for a precise definition of what we mean by that.

[4] There is in fact a weaker condition with this effect.

Proposition 2.8 *There is a faithful linearly distributive functor* $\mathbf{GGam} \longrightarrow \mathbf{Rel}$.

As we explain in Section 4 this leads us to hope that we can in some sense borrow an exponential from the relational model.

The linear exponential we introduce here is somewhat unusual in that it has the property that applied to a finite game, it results in a finite game. We say that a strategy is finite if its image (when viewed as partial function) is finite. In other words α on A is finite if and only if $R_P(\alpha)$ is finite. We order strategies in the usual way as partial functions, that is a strategy can be increased by extending its domain of definition. Note that because of conflict-freeness, extending the domain of definition by just one element may not be possible with strategies on tree-games—a number of further elements may have to be added.

The P-position of $!A$ are the finite strategies on A. From such a finite strategy α, a valid move consists of pairing α with one of the elements of $R_O(\alpha)$ on which α is undefined. In keeping with the concrete data structure intuition we can view this as Opponent now asking Player to fill another cell given by a. From such a pair (α, a) there is a move to a (finite) P-strategy α' if

- α' extends α (as partial functions);
- α' is defined at a;
- α' is minimal with that property.

Note that for tree games (where strategies are automatically conflict-free) one can always extend the domain of definition of a strategy by just one element; the conflict-freeness condition means that this need not be possible for general graph games. This means that a 'one step extension' $\alpha \longrightarrow (\alpha, a) \longrightarrow \alpha'$ need not be uniquely defined, even when a is given, but below each extension there is a unique least one. In terms of concrete data structures this kind of situation might apply if, in order to fill the cell Opponent asked for (given by a), Player first needs to fill other cells, but those 'other cells' are not uniquely defined.

The result is a regulated graph game $!A$, and ! can be extended to an endofunctor on $\mathbf{GGam}$. All the structure maps for the comonad can be viewed as copy-cat type strategies, but the details are quite delicate and we lack space to give them here.

Theorem 2.9 *The category* $\mathbf{GGam}$ *of (regulated) graph games is a model of Intuitionistic Linear Logic.*

This provides us with a cartesian closed structure on $\mathbf{GGam}$ in the usual way. The product remains the old product, and the function space is given by the Girard formula

$$A \Rightarrow B = !A \multimap B.$$

3 Graph games and tree games

There are categories of traditional tree games which also give models for Linear Logic. One such category, $\mathbf{TGam}$ (on which most categories of games in semantics are based), is identified in [7], and we will describe here how it is connected to the category of graph games. A tree game (in the traditional sense) is just a game whose underlying graph is a tree. Let A and B be tree games. Then the positions of their linear function space $A \multimap_t B$ (in the traditional sense, as tree games), can be mapped to pairs of positions, one in A and one in B in an obvious way. This turns out to be a position in $A \multimap B$. We can now interpret a morphism $A \longrightarrow B$ in $\mathbf{TGam}$, that is a strategy on $A \multimap_t B$, as a strategy on $A \multimap B$. For any pair of positions (a, b) in the trees for A and B occurs in a play in $A \multimap_t B$ according to a tree strategy ρ in at most one way, and therefore given a O-position (a, b) in $A \multimap B$ we can use ρ to tell us how the corresponding graph game strategy replies. That this gives a valid conflict-free pre-strategy is somewhat surprising, the reason for this is given by the 'switching conditions' in $A \multimap B$: only Player is allowed to switch between the component games. We thus obtain an embedding $\mathbf{TGam} \longrightarrow \mathbf{GGam}$.

Proposition 3.1 *There exists a full monoidal embedding* $J\colon \mathbf{TGam} \longrightarrow \mathbf{GGam}$ *which preserves all products.*

Given a graph game A, we can 'treeify' it. The positions of the resulting game TA are plays of the graph game. Plays of even length are O-positions while plays of odd length become P-positions.[5] There is a move in the treeified game from one such position to another if and only if the latter extends the former by one move. It then turns out that $TA \multimap_t TB$ and $T(A \multimap B)$ are naturally isomorphic, and functoriality of T comes for free, as does monoidality of T.

Proposition 3.2 *The functor* $T\colon \mathbf{GGam} \longrightarrow \mathbf{TGam}$ *preserves products, tensor products and linear function spaces.*

It is worth pointing out that these two functors are *not* adjoint to each other. The reason for this is that the obvious copy-cat strategies $A \longrightarrow JTA$ and $JTA \longrightarrow A$ are not conflict-free.

We get these results for general reasons. The functor JT is a monoidal endofunctor on $\mathbf{GGam}$ which can be equipped with coherent isomorphisms

$$\begin{array}{rcll} JT(JTA \otimes B) & \cong & JT(A \otimes B) & \quad\text{and} \\ JT(JTA \multimap B) & \cong & JT(A \multimap B) & \\ & \cong & JT(A \multimap JTB). & \end{array}$$

[5] If this sounds different from the usual conventions it is because our games are based on the notion of *position* rather than that of *move*.

Using this the multiplicative structure on the image $\mathbf{TGam}$ of JT can be defined by

$$\begin{aligned} A \otimes_t B &= T(JA \otimes JB) \quad \text{and} \\ A \multimap_t B &= T(JA \multimap JB). \end{aligned}$$

More significantly we have natural transformations

$$\begin{aligned} \phi &\colon ! \longrightarrow !JT \quad \text{and} \\ \psi &\colon ! \longrightarrow JT! \end{aligned}$$

satisfying natural coherence conditions. Those relevant for the exponential on $\mathbf{TGam}$ are

$$\begin{aligned} (T! \xrightarrow{T\psi} TJT! \cong T!) &= \mathsf{id}_{T!} \\ (!J \xrightarrow{\phi J} !JTJ \cong !J) &= \mathsf{id}_{!J} \\ (!! \xrightarrow{!\psi} !JT!) &= (!! \xrightarrow{\phi !} !JT!) \end{aligned}$$

These conditions allow us to define an exponential on $\mathbf{TGam}$ within $\mathbf{GGam}$ by

$$!_t = T!J.$$

We omit the full definition of the exponential structure for lack of space. One can deduce abstractly from our Theorem 2.9 the following result.

Theorem 3.3 $\mathbf{TGam}$ *is a model of Intuitionistic Linear Logic.*

The exponential we obtain in Theorem 3.3 is the Curien exponential [4] so this result is already known. We note however that it appears that one cannot obtain other known exponentials on $\mathbf{TGam}$ in this way. Other well-known exponentials on $\mathbf{TGam}$ do not seem to give rise to exponentials on $\mathbf{GGam}$, so the connection between $\mathbf{TGam}$ and $\mathbf{GGam}$ privileges the Curien exponential on the former.

We now show that the type structure $\mathbb{N}_g^{(\sigma)}$ and $\mathbb{N}_t^{(\sigma)}$ are closely related. Say that a regulated game A is *P-tree-like* if and only if whenever two plays merge they do so at an O-position (and so the two plays diverged also from an O-position). Dually there is a notion of O-tree-like. We have the following.

Lemma 3.4 *For any A, $!A$ is O-tree-like.*

Lemma 3.5 *Suppose A is P-tree-like.*

(i) Strategies in A correspond exactly to strategies in TA, that is we have

$$T\colon \mathbf{GGam}(\mathbf{1}, A) \cong \mathbf{TGam}(\mathbf{1}, TA).$$

(ii) The functor T commutes with $!$ on A, that is,

$$T!A \cong T!JTA.$$

Lemma 3.6 *Suppose that A is O-tree-like and that B is P-tree-like. Then $A \multimap B$ is P-tree-like.*

Proposition 3.7 *Suppose A and B are P-tree-like. Then so is $A \Rightarrow B$ and moreover*

$$T(A \Rightarrow B) \cong TA \Rightarrow_t TB.$$

We give the final calculation.

$$\begin{aligned} T(A \Rightarrow B) &\cong T(!A \multimap B) \\ &\cong T!A \multimap_t TB \\ &\cong T(JT!JTA \multimap JTB) \\ &\cong TA \Rightarrow_t TB\,. \end{aligned}$$

Let $\mathbb{N}_t$ in $\mathbf{TGam}$ and $\mathbb{N}_g$ in $\mathbf{GGam}$ be the game with one opening O-move to which P may respond by playing any natural number. We define $\mathbb{N}_g^{(\sigma)}$ for higher types σ inductively via

$$\begin{aligned} \mathbb{N}_g^{(0)} &= \mathbb{N}_g \\ \mathbb{N}_g^{(\sigma \times \tau)} &= \mathbb{N}_g^{(\sigma)} \times \mathbb{N}_g^{(\tau)} \\ \mathbb{N}_g^{(\sigma \to \tau)} &= \mathbb{N}_g^{(\sigma)} \Rightarrow \mathbb{N}_g^{(\tau)} \end{aligned}$$

and similarly for $\mathbb{N}_t^{(\sigma)}$. Note that the type structure $\mathbb{N}_t^{(\sigma)}$ is the sequential algorithms model of Berry and Curien (see [2]).

As models for this type structure, the category of regulated graph games and that of tree games are very closely connected.

Theorem 3.8 *The functor T induces an isomorphism*

$$T\mathbb{N}_g^{(\sigma)} \cong \mathbb{N}_t^{(\sigma)}$$

for all types σ.

4 An extensional exponential

By working with graph games we aim to force a degree of extensionality into our strategies. (This idea goes back to approaches to PCF sequentiality developed by Gandy and Pani in unpublished work.) The analysis we have just given shows us that our simple exponential subverts that purpose (at least for PCF types).

We believe that there is a natural more extensional exponential which better serves this purpose and is consonant with the approach of Gandy and Pani. This is based in effect on sets of (maximal) positions rather than on the intensional strategies themselves (so it derives from the exponential on abstract games). If we are right this gives a new model for Intuitionistic Linear Logic and a new cartesian closed type structure $\mathbb{N}_e^{(\sigma)}$. We comment on the relation we expect between the type structures $\mathbb{N}_e^{(\sigma)}$ and $\mathbb{N}_t^{(\sigma)}$. Clearly $\mathbb{N}_e$ and $\mathbb{N}_t$ will have the same elements (global sections) and the same should be essentially true for any type of the form $(0 \times \cdots \times 0 \to 0)$. In particular both $\mathbb{N}_e^{(0\times 0\to 0)}$ and $\mathbb{N}_t^{(0\times 0\to 0)}$ will contain left and right algorithms ζ_l and ζ_r for a function which takes $(0,0)$ to 0 and is otherwise undefined. So ζ_l and ζ_r are approximations to the familiar left and right algorithms $+_l$ and $+_r$ for addition. In the sequential algorithms model $\mathbb{N}_t^{(\sigma)}$ there is an algorithm δ which distinguishes between ζ_l and ζ_r, that is with $\delta(\zeta_l) = 0$ and $\delta(\zeta_r) = 1$. But since running ζ_l and ζ_r in the graph game model will end in the same position, δ should not exist in $\mathbb{N}_e^{(\sigma)}$. Generally at the type 2 level the elements of $\mathbb{N}_e^{(\sigma)}$ should be included in those of $\mathbb{N}_t^{(\sigma)}$. At higher levels this ceases even to make sense; at type 3 level the positions of $\mathbb{N}_e^{(\sigma)}$ should be identifiable with equivalence classes of elements of $\mathbb{N}_t^{(\sigma)}$.

Recall that Ehrhard identified the extensional collapse of the sequential algorithms model of higher types with the hypercoherence model [5]. In other words if one starts with $\sim^0$ the identity relation on $\mathbb{N}_t^{(0)}$ and defines the partial equivalence relations $\sim^\sigma$ inductively in the usual realizability style (see for example [10]) then the equivalence classes $\mathbb{N}_t^{(\sigma)}/_{\sim^\sigma}$ are the sequentially realizable functionals and correspond to the points of the hypercoherence model. Despite the marked differences between $\mathbb{N}_t^{(\sigma)}$ and $\mathbb{N}_e^{(\sigma)}$ we expect, in view of results in Longley [10], that the same will be true of the equivalence classes in the extensional collapse of $\mathbb{N}_e^{(\sigma)}$.

5 Conclusions and Further Work

We have provided a new concrete data structure-style model for sequential computation and, using a simple exponential, we have shown that it gives rise to the sequential algorithms model. This provides rather modest support for the Longley conjecture. We hope that we can provide more telling evidence for it using a more extensional exponential.

We make two further observations. First our model is computational yet there is a good connection with a category of abstract games. This raises the possibility, not analysed here, that one could use our model to provide cleaner proofs of the harder equivalences in Longley's [10]. In particular we observe that Melliés in [11] has independently considered issues connected with equivalences of plays in the sequential algorithms model. While he deals only with that setting, he is able to use his notion of *extensional data structure* to provide a direct link with the general notion of coherence on which the hypercoherence model is based, allowing him to give a computational interpretation of same. There should be connections with our work and one might hope to exploit Melliés' insights further in exploring the world of sequentially realizable functionals.

Secondly if our suggestion for a more extensional exponential works it provides a model with less 'junk' (that is non-extensional programs/strategies) than the sequential algorithms model. It therefore might be a first step towards the answer to a question posed in [10]: Is there a 'games style' model for the sequentially realizable functionals without junk? This is a modern reading of a question which Gandy was constantly addressing in his last years.

Appendix

A Merging sequences

Let $\mathbf{TI}$ be the category whose objects are finite totally ordered sets (which we think of as plays) and whose morphisms are 'merges', or 'total interleavings' of the two, that is, if A and B are finite totally ordered sets then a morphism $A \longrightarrow B$ is a total order on $A + B$ respecting the given orders on A and B. These morphisms are uniquely represented by relations $F\colon A \relbar\joinrel\mapstochar\joinrel\rightarrow B$ such that

$$a' \geq a \; F \; b \geq b' \qquad \text{implies} \qquad a' \; F \; b'.$$

The total order on $A + B$ then is the disjoint union of the orders on A and B with

$$\begin{array}{lll} b < a & \text{if and only if} & a \; F \; b \quad \text{and} \\ a < b & \text{if and only if} & \neg(a \; F \; b) \end{array}$$

Composition is given by relational composition, and the identity on A is given by the $>$ relation on A.

The order obtained on $A + A$ by the identity 'interleaves' the two copies of A in such a way that identical elements become neighbours and the right copy always comes first.

The order obtained on $A + C$ obtained by composing $F\colon A \relbar\joinrel\mapstochar\joinrel\rightarrow B$ and $G\colon B \relbar\joinrel\mapstochar\joinrel\rightarrow C$ is identical with the one obtained as follows. View $A + C$ as embedded in $A + B + C$ with the order given by the transitive closure of those defined on $A + B$ and $B + C$ via F and G respectively. So $c < a$ if and only if there exists $b \in B$ with $c < b < a$.

For our application we are actually interested in merging plays, where moves come in two flavours: P-moves and O-moves. For that purpose we consider the 'dualized' category

$$\mathbf{DTI} = \mathbf{TI} \times \mathbf{TI}^{\mathrm{op}}.^{6}$$

This category has objects of the form (A^+, A^-) and we think of the elements of A^+ as P-moves, and of the elements of A^- as O-moves. Clearly if we split a finite play into these two sets then the size of the former is either equal

[6] See [8] for a sketch of the properties of categories formed in this way.

to the size of the latter, or it is smaller by one, and the merge is 'perfect', starting with an element of $^-$. Hence not all objects in the category can be viewed as plays.

The identity on (A^+, A^-) gives rise to a total order on $(A^+ + A^+) + (A^- + A^-)$ as follows. The least element of 'the right' copy of A^- is followed by the least element of 'the left' copy of A^-, then we get the least element of 'the left' copy of A^+ followed by the least element of 'the right' copy of A^+, and the circle starts over. This is the order of moves carried out by the standard copy-cat strategy.

The order given on $(A^+ + C^+, A^-, C^-)$ given by the composite of the morphisms

$$(F^+, F^-)\colon (A^+, A^-) \longrightarrow (B^+, B^-) \qquad \text{and}$$
$$(G^+, G^-)\colon (B^+, B^-) \longrightarrow (C^+, C^-)$$

is the one inherited from that on

$$(A^+ + B^+ + C^+) + (A^- + B^- + C^-)$$

which is given by the transitive closure of the orders given on $(A^+ + B^+) + (A^- + B^-)$ and $(B^+ + C^+) + (B^- + C^-)$ by the two morphisms. For objects which arise from plays this corresponds to 'parallel composition and hiding'—the hidden positions are those in B.

References

[1] P. Baillot, V. Danos, T. Ehrhard, and L. Regnier. Believe it or not, AJM's games model is a model of classical linear logic. In *Proc. Logic in Computer Science 1997*, pages 68–75. IEEE Computer Society Press, 1997.

[2] G. Berry and P.-L. Curien. Sequential algorithms on concrete data structures. *Theoretical Computer Science*, 20:265–321, 1982.

[3] P.-L. Curien. *Categorical Combinators, Sequential Algorithms and Functional Programming*. Birkhäuser, 2nd edn. edition, 1993.

[4] P.-L. Curien. On the symmetry of sequentiality. In *Proceedings of MFPS93*, volume 802 of *Lecture Notes in Computer Science*, pages 29–71. Springer-Verlag, 1994.

[5] T. Ehrhard. Hypercoherences: a strongly stable model of linear logic. *Math. Struct. in Comp. Science*, 3:365–385, 1993.

[6] T. Ehrhard. A relative definability result for strongly stable functions and some corollaries. *Information and Computation*, 152:111–137, 1999.

[7] M. Hyland and A. Schalk. Abstract games for linear logic. extended abstract. In M. Hofmann, G. Rosolini, and D. Pavlovic, editors, *Proceedings of CTCS '99, Conference on Category Theory and Computer Science*, volume 29 of *Electronic Notes on Theoretical Computer Science*, 1999.

[8] M. Hyland and A. Schalk. Glueing and orthogonality for models of linear logic. *Theoretical Computer Science*, 2002.

[9] G. Kahn and G. Plotkin. Concrete domains. *Theoretical Computer Science*, 121(1–2):187–277, 1993.

[10] J. Longley. The sequentially realizable functionals. *Annals of Pure and Applied Logic*. To appear.

[11] P. A. Melliés. On the extensional content of sequential games. Available electronically at the url http://www.pps.jussieu.fr/~mellies/extensionalgames.ps.gz.

[12] J. van Oosten. A combinatory algebra for sequential functionals of finite type. Technical Report 996, Utrecht University, 1997.

[13] G. Winskel. Event structures. In *Petri Nets: Applications and Relationships to Other Models of Concurrency, Advances in Petri Nets 1986*, volume 255 of *Lecture Notes in Computer Science*, pages 325–392. Springer-Verlag, 1986.