Sequential algorithms and strongly stable functions

Paul-André Melliès

Sequential algorithms and strongly stable functions

Paul-André Melliès [1]

Abstract

Intuitionistic proofs (or PCF programs) may be interpreted as functions between domains, or as strategies on games. The two kinds of interpretation are inherently different: static vs. dynamic, extensional vs. intentional. It is extremely instructive to compare and to connect them. In this article, we investigate the extensional content of the *sequential algorithm* hierarchy $[-]_{\mathrm{SDS}}$ introduced by Berry and Curien two decades ago. We equip every sequential game $[T]_{\mathrm{SDS}}$ of the hierarchy with a *realizability relation* between *plays* and *extensions*. In this way, the sequential game $[T]_{\mathrm{SDS}}$ becomes a directed acyclic graph, instead of a tree. This enables to define a hypergraph $[T]_{\mathrm{HC}}$ on the extensions (or terminal leaves) of the game $[T]_{\mathrm{SDS}}$. We establish that the resulting hierarchy $[-]_{\mathrm{HC}}$ coincides with the *strongly stable* hierarchy introduced by Bucciarelli and Ehrhard. We deduce from this a game-theoretic proof of Ehrhard's collapse theorem, which states that the strongly stable hierarchy coincides with the extensional collapse of the sequential algorithm hierarchy.

1 Introduction

A spectacular number of game semantics have been introduced in the last decade, in order to capture the interactive essence of various logical systems or programming languages. Comparatively, the number of interactive paradigms underlying these models has remained desesparately low. Today, mainstream game semantics is

- sequential,
- played on trees [Joyal 1977][Abramsky, Jagadeesan 1994][Lamarche 1992] or on arenas [Hyland, Ong 2000][Nickau 1994][Abramsky, McCusker 1999].

In this article, we champion a more *concurrent* or *graph-theoretic* style of game semantics, which we see pervading a series of recent contributions:

[1] Postal address: Equipe Preuves Programmes Systèmes, CNRS UMR 7126, Université Paris VII, 2 place Jussieu, Case 7014, 75251 Paris Cedex 05, FRANCE. Email address: `mellies@pps.jussieu.fr`

Preprint submitted to Theoretical Computer Science

- *money games* [Joyal 1997] are positional games played on graphs, instead of trees. Joyal introduces them in order to recast Whitman's characterization of the free lattice. This comes as a preliminary step toward the intended construction of the bifree completion of a category. See also the later connection between μ-bicomplete categories and parity games, established in [Santocanale 2002].
- *graph games* [Hyland, Schalk 2002] are positional games played on graphs, instead of trees. The resulting model of PCF is shown to be the sequential algorithm model [Berry, Curien 1982].
- *concurrent games* [Abramsky, Melliès 1999] are positional games played on domains, instead of trees. The model is shown to be fully complete for multiplicative additive linear logic. See also [Abramsky 2001] for a discussion about sequentiality and concurrency in games and logic.

Either sequential or concurrent, these game semantics have one thing in common: they are *positional*.

Interleaving vs. true concurrency

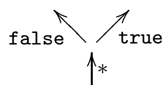
Playing on a *positional game* (instead of a tree or an arena) means that two different sequences of moves starting from the root may lead to the same position. This is the game-theoretic avatar of *true concurrency* in process calculus. Think of a process π and two transitions a and b starting from π. The two transitions a and b are declared *independent* when they may be emitted or received by π in any order, without interference. Independence of the two transitions is generally represented by *tiling* the two sequences $a \cdot b$ and $b \cdot a$ in the transition system:

$$\begin{tikzcd}[column sep=small, row sep=small] & \pi' & \\ \pi_1 \arrow[ur, "b"] & & \pi_2 \arrow[ul, "a"'] \\ & \pi \arrow[ul, "a"] \arrow[ur, "b"'] & \end{tikzcd} \tag{1}$$

The *homotopy equivalence* between transition paths is then defined in the expected way: two paths are called *homotopic* when they are equal modulo a series of permutations (1) of independent transitions. This 2-dimensional grammar of independence provides a "geometry" where the *interleaving semantics* and the *true concurrency semantics* of processes coexist, formulated respectively as transition paths and homotopy classes [Pratt 1991]. The author experienced the relevance of this diagrammatic vision in rewriting theory: the 2-dimensional paradigm leads to a syntax-free theory of causality and neededness, including a standardization theorem, and the characterization of head-reductions in a wide class of calculi [Melliès 1998].

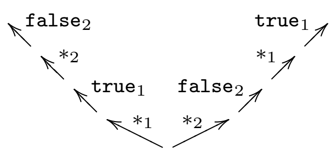
Mainstream game semantics has not reached that stage of refinement yet. It is still very much 1-dimensional. We advocate that bringing out 2-dimensional structures on sequential games will clarify their structure, and their relationship to other models of computation. In this article, we provide evidence for that thesis, with a limited but striking illustration of how concurrency ideas may explicate the *extensional* (we also say *static*) content of sequential game semantics.

True concurrency in games: dynamic plays realize static extensions
We start from the elementary intuition that sequential game semantics provides an *interleaving semantics* of proofs and programs. Suppose that $\mathbb{B}$ is the boolean game starting with Opponent's question $*$ followed by Player's answer `true` or `false`:



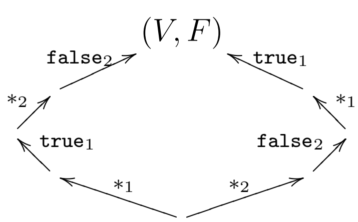
(2)

Each play of the tensor product $\mathbb{B} \otimes \mathbb{B}$ is an interleaving of plays of the two instances $\mathbb{B}_1$ and $\mathbb{B}_2$ of the sequential game $\mathbb{B}$. We draw below a fragment of the resulting tree:



(3)

The two plays drawn in (3) are different from a procedural point of view, but equivalent from an extensional point of view, since both plays answer the same *extensional* pair (V, F) to Opponent's questions — where by V we mean "true" (*vrai* in french) and by F we mean "false".
So, it is tempting to bend the two paths (3) and to tile them as in the diagram below:



(4)

After this plastic surgery, $\mathbb{B} \otimes \mathbb{B}$ becomes a *directed acyclic graph* (dag) instead of a tree. The terminal leaf (V, F) is added on top to indicate that the two plays *realize* the same *extension* (V, F). The resulting diagram (4) is the game-theoretic counterpart of diagram (1). It relates the *interleaving semantics* expressed by the plays to the *true concurrency semantics* expressed by the extension (V, F).
We will see that shifting from a tree in (3) to a dag in (4) clarifies much about how the "implicit/static" and "explicit/dynamic" presentations of sequentiality are

connected at higher types. More precisely, we establish in the course of the article that, for every simple type T, the *extensions* of the sequential game associated to T are precisely the *atoms* of the dI-domain with coherence associated to T in the strongly stable model [Bucciarelli, Ehrhard 1991].

Ehrhard's collapse theorem

This leads to the second motivation of this work: a key result by Ehrhard states that the *sequential algorithm* model of PCF [Berry, Curien 1982] collapses extensionally to the *strongly stable* hierarchy [Bucciarelli, Ehrhard 1991]. The theorem is remarkable, because it links for the first time a *static* and a *dynamic* model of sequentiality. Ehrhard's original proof [Ehrhard 1997] is a domain-theoretic *tour de force* based on the observation that every strongly stable function is definable in PCF enriched with the strongly stable functions of degree 2.

Here, we want to establish the same result another time, using game-theoretic ideas. More specifically, we want to characterize *dynamically* the classes of strategies generated by the extensional collapse. Instead of working directly on Berry-Curien and Bucciarelli-Ehrhard models of PCF, which would be extremely difficult technically, we take advantage of the fact that both hierarchies can be "linearized", that is, derived from models of (intuitionistic) linear logic, using a kleisli construction:

- The sequential algorithm model is linearized by Lamarche as a game model of intuitionistic linear logic, based on *sequential data structures* (sds). Recall that a sds A is defined as (1) a polarized alphabet (M_A, λ_A) of *moves* and (2) a prefix-closed set P_A of alternating *plays* in which Opponent starts. The distinctive feature of the model lies in the interpretation of the exponential modality of linear logic. The sds $!A$ is defined by a *backtrack interleaving* of the plays of the sds A. This departs from the usual definition based on a *repetitive interleaving* of plays given in [Abramsky et al. 1994]. Lamarche shows in [Lamarche 1992] that the model linearizes the sequential algorithm model of PCF. The construction is then reformulated and clarified by Curien in [Curien 1993][Amadio, Curien 1998].
- The strongly stable model is linearized by Ehrhard as a *hypercoherence space* model of linear logic. The model refines Girard *coherence space* model, just like strong stability refines stability. Recall that a hypercoherence space X is just a *hypergraph*, that is (1) a set $|X|$ of atoms (called the *web*) and (2) a set $\Gamma(X) \subset^*_{\mathrm{fin}} |X|$ of nonempty finite subsets of atoms (called the *coherence*) in which every singleton $\{x\}$ is element of $\Gamma(X)$, for $x \in |X|$. Ehrhard shows in [Ehrhard 1993] that the hypercoherence space model linearizes the strongly stable model of PCF.

Extensional data structures
As advocated above, the extensional content of a sequential game is revealed by its 2-dimensional structure. The author is currently developing a theory of *asynchronous games* in which only *local* tiles (1×1) are admitted. In this framework, the usual lexicon of arena games is formulated in a *truely concurrent* fashion: justification pointers and views are reconstructed by permuting moves in a play, and innocent strategies turn out to be positional strategies enjoying forward and backward confluence properties [Melliès 2004b].
The resulting theory is pretty involved though, and we will not develop it here. We take a short cut instead, and demonstrate that only a small amount of homotopy or asynchrony is necessary to capture the extensional content of sequential games: the tiles considered in this article are *global* and expressed by a *realizability relation* between plays (= the interaction paths) and extensions (= their homotopy classes). We write E_A for the set of extensions, and $\|x\|_A \subset^*_{\text{fin}} P_A^{\mathbf{even}}$ for the (nonempty finite) set of (even-length) plays which realize an extension $x \in E_A$. A sequential data structure (sds) $A = (M_A, \lambda_A, P_A)$ equipped with such a realizability relation defines what we call an *extensional data structure* (eds).
The realizability relation enables to *visualize* every extensional data structure as a directed acyclic graph (dag) labelled by extensions x on nodes — at least the graphic edss, see the definition given in Section 6 (definition 6.1). For instance, the graphic eds $!\mathbb{B}$ has three extensions $\perp$, F and V, and is represented as the tree:

$$\begin{tikzcd} F & & V \\ & {} \arrow[ul, "\texttt{false}"'] \arrow[ur, "\texttt{true}"] & \\ & \perp \arrow[u, "*"'] & \end{tikzcd} \tag{5}$$

The extension $\perp$ at the root and the extensions F and V at the leaves indicate that:

$$\epsilon \in \|\perp\|_{!\mathbb{B}} \qquad * \cdot \texttt{false} \in \|F\|_{!\mathbb{B}} \qquad * \cdot \texttt{true} \in \|V\|_{!\mathbb{B}}$$

where ϵ denotes the empty play. The advantage of the graph-theoretic notation becomes clear when one tensors the eds $!\mathbb{B}$ three times, and draws the graphic eds $!\mathbb{B} \otimes !\mathbb{B} \otimes !\mathbb{B}$ as illustrated in figure 1.

Extracting hypercoherence spaces from extensional data structures

We mentioned earlier the coincidence between (1) the *extensions* of the eds interpreting a simple type T in the sequential algorithm hierarchy, and (2) the *atoms* of the dI-domain with coherence interpreting T in the strongly stable hierarchy. We clarify and illustrate this point briefly. Recall that the atoms of the dI-comain with

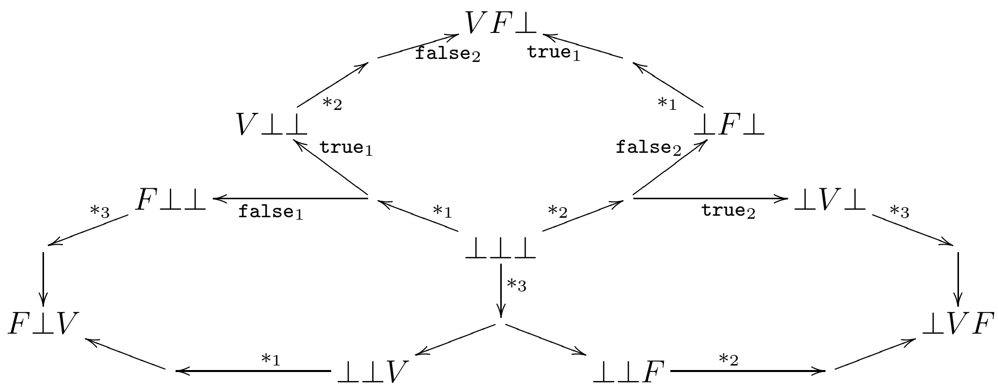


Fig. 1. A fragment of the extensional data structure $!\mathbb{B}\otimes!\mathbb{B}\otimes!\mathbb{B}$.

coherence associated to the simple type T form a *hypercoherence space*. This observation is at the heart of [Ehrhard 1993]. We will see in Section 7 how to *extract* a hypercoherence space $U(A)$ from every eds A — at least when the eds A is *regular*, see definition 7.3. The web of this hypercoherence space $U(A)$ is precisely the set of extensions of A:

$$|U(A)| = E_A.$$

Typically, one deduces from the graph-theoretic presentation of the sequential game $!\mathbb{B}\otimes!\mathbb{B}\otimes!\mathbb{B}$ produced in Figure 1 that:

- the triple $w = \{(V,F,\bot),(F,\bot,V),(\bot,V,F)\}$ is coherent in $!\mathbb{B}\otimes!\mathbb{B}\otimes!\mathbb{B}$ because (informally) Opponent has to choose between one of the extensions of w when she plays the first move $*_i$ for $i \in \{1,2,3\}$. For instance, Opponent plays $*_1$ and thus rejects the extension $(\bot,V,F)$ as possible outcome of the interaction,
- the pair $v = \{(V,F,\bot),(F,\bot,V)\}$ is incoherent in $!\mathbb{B}\otimes!\mathbb{B}\otimes!\mathbb{B}$ because (informally again) Player has to choose between one of the extensions of w after Opponent plays the move $*_1$. For instance, Player plays $*_1 \cdot \texttt{true}_1$ and thus rejects the extension $(F,\bot,V)$ as possible outcome of the interaction.

There are historical reasons for illustrating our ideas with the eds $!\mathbb{B}\otimes!\mathbb{B}\otimes!\mathbb{B}$ and the subset $w = \{(\bot,V,F),(F,\bot,V),(V,F,\bot)\}$ of extensions. The example stems from [Berry 1979] in which the stable hierarchy $[-]_{\mathrm{S}}$ of simple types is introduced. There, Berry describes a *stable* but *non-sequential* function G at the simple type $(o\times o\times o)\Rightarrow o$:

$$G(x,V,F)=V \qquad G(F,x,V)=V \qquad G(V,F,x)=V$$

and $G(x,y,z)=\bot$ otherwise. This function is often called the Gustave function in the litterature. The fact that the triple w is not bounded (and thus "incoherent") in the dI-domain

$$[o \times o \times o]_{\mathrm{S}} = \{\perp, V, F\} \times \{\perp, V, F\} \times \{\perp, V, F\}$$

interpreting the simple type $o \times o \times o$ in the stable model, is the starting point of the theory of strong stability in dI-domains with coherence [Bucciarelli, Ehrhard 1991]. The point of strong stability is precisely that the triple w becomes *coherent* in the dI-domain with coherence interpreting $o \times o \times o$ in the strongly stable hierarchy.

Technical contributions of the article

The first contribution of the article is to clarify the dynamic content of hypercoherence spaces, as follows:

(1) we define when a strategy σ of an eds A implements a set $f \subset E_A$ of extensions of A; and call *configuration* any set $f \subset E_A$ implemented by a strategy,

(2) we extract from any regular eds A a hypercoherence space $U(A)$ with web the set E_A of extensions of A,

(3) we show that in any regular eds A, the finite configurations of A are exactly the finite cliques of $U(A)$.

We consider in this article *two* different interpretations of the base type ι as a sequential game:

- either as the *flat* natural number eds noted $\mathbb{N}_{\mathbf{flat}}$,
- or as the *lazy* natural number eds noted $\mathbb{N}_{\mathbf{lazy}}$,

Each interpretation induces a sequential algorithm hierarchy of simple types $[-]^{\mathbf{flat}}_{\mathrm{SDS}}$ (also noted $[-]_{\mathrm{SDS}}$) and $[-]^{\mathbf{lazy}}_{\mathrm{SDS}}$.

The second contribution of the article is to extract the strongly stable hierarchy from the game-theoretic hierarchies $[T]^{\mathbf{flat}}_{\mathrm{SDS}}$ and $[T]^{\mathbf{lazy}}_{\mathrm{SDS}}$. More precisely, we show that the hypercoherence space $[T]_{\mathrm{HC}}$ associated to a simple type T in [Ehrhard 1993] is precisely the hypercoherence space computed by U from the edss $[T]^{\mathbf{flat}}_{\mathrm{SDS}}$ and $[T]^{\mathbf{lazy}}_{\mathrm{SDS}}$:

$$[T]_{\mathrm{HC}} = U([T]^{\mathbf{flat}}_{\mathrm{SDS}}) = U([T]^{\mathbf{lazy}}_{\mathrm{SDS}}) \tag{6}$$

From that concrete connection between the sequential algorithm and the strongly stable hierarchies, we deduce a game-theoretic proof of Ehrhard's collapse theorem. Surprisingly, this last part is far from easy — despite the equalities (6). We proceed in three steps.

First, (★) we show that the *flat* and the *lazy* sequential algorithm hierarchies collapse to the same hierarchy of types. The argument imported from [Melliès 2004a] is based on the existence of a retraction between $\mathbb{N}_{\mathbf{flat}}$ and $\mathbb{N}_{\mathbf{lazy}}$ in the category of edss:

$$\mathbb{N}_{\mathbf{flat}} \xrightarrow{\mathbf{for}} \mathbb{N}_{\mathbf{lazy}} \xrightarrow{\mathbf{count}} \mathbb{N}_{\mathbf{flat}} \;=\; \mathbb{N}_{\mathbf{flat}} \xrightarrow{\mathbf{id}_{\mathbb{N}_{\mathbf{flat}}}} \mathbb{N}_{\mathbf{flat}}$$

Then, (★★) we prove by a non-constructive compactness argument that the (possibly infinite) configurations of a *finitely branching* eds A are precisely the cliques of $U(A)$. This is precisely the reason why we work with the *lazy* hierarchy instead of the *flat* one: the interpretation $[T]^{\mathbf{lazy}}_{\mathrm{SDS}}$ of every simple type T is finitely branching, and the compactness argument works only on finitely branching games.
Finally, (★★★) we characterize the partial equivalence relation $\sim_T$ on the strategies of $[T]^{\mathbf{lazy}}_{\mathrm{SDS}}$ induced by the extensional collapse, for every simple type T. We show that $\sim_T$ relates two strategies σ and τ of $[T]^{\mathbf{lazy}}_{\mathrm{SDS}}$ precisely when:

(1) the strategies σ and τ are *extensional* in a sense explained in Section 11,
(2) the strategies σ and τ implement the same configuration.

We deduce that the set of strategies of $[T]^{\mathbf{lazy}}_{\mathrm{SDS}}$ quotiented by $\sim_T$ is in a one-to-one relationship with the *configurations* of $[T]^{\mathbf{lazy}}_{\mathrm{SDS}}$.
Putting the three steps (★) and (★★) and (★★★) together, we conclude that the *flat* sequential algorithm hierarchy collapses extensionally to Bucciarelli-Ehrhard strongly stable hierarchy. This is precisely the statement of Ehrhard's theorem in [Ehrhard 1997].

Structure of the paper:
We start in Section 2 with preliminaries on models of linear logic, hierarchies of types, and extensional collapse. In Section 3, we introduce a *hypergraph* model which coincides with the hypercoherence space model on simple types, but captures sequentiality more accurately outside the intuitionistic types. In Section 4, we recall the sequential data structure (sds) model of intuitionistic linear logic. In Section 5, we introduce the *extensional data structure* (eds) hierarchy, which is just the original sds hierarchy, equipped with extensional information. We show in Section 6 that every simple type T is interpreted as a *spread* eds $[T]^{\mathbf{flat}}_{\mathrm{SDS}}$ which may be visualized as a directed acyclic graph (dag). In Section 7, we extract from every *regular* eds A a hypercoherence space $U(A)$ and show that the *finite* configurations of A as the *finite* cliques of $U(A)$. We show in Section 8 that the construction U extracts the strongly stable hierarchy $[T]_{\mathrm{HC}}$ from either the *flat* or the *lazy* sequential algorithm hierarchy $[T]^{\mathbf{flat}}_{\mathrm{SDS}}$ and $[T]^{\mathbf{lazy}}_{\mathrm{SDS}}$. In Section 9, we exhibit a retraction between the flat and the lazy hierarchies $[-]^{\mathbf{flat}}_{\mathrm{SDS}}$ and $[-]^{\mathbf{lazy}}_{\mathrm{SDS}}$, and deduce from this that the two hierarchies collapse to the same extensional hierarchy. In Section 10, we use a non-constructive compactness argument to show that the (possibly infinite) configurations of $[T]^{\mathbf{lazy}}_{\mathrm{SDS}}$ are the cliques of $[T]_{\mathrm{HC}}$. The two last sections are the most technical ones. In Section 11, we equip every extensional data structure of the lazy hierarchy with a notion of *alive* plays; and define when a strategy is *extensional* in such a structure. In Section 12 we characterize the self-equivalent strategies of the collapse of $[-]^{\mathbf{lazy}}_{\mathrm{SDS}}$ as the extensional strategies of the hierarchy; and deduce

Ehrhard's collapse theorem from that.

Related works on sequentiality and strong stability

A model of "extensional sequential algorithms" is introduced in [Ehrhard 1997]. The idea is to consider triples (E, X, π) where E is a sequential structure (Ehrhard's own domain-theoretic presentation of *sequential* concrete data structures), X is a hypercoherence space, and π is a strongly stable linear function

$$(E_*, \mathcal{C}^{\mathrm{L}}(E)) \xrightarrow{\pi} qDC(X)$$

between the dI-domains with coherence associated to E and X. The main requirement on the "projection map" π is the following "lifting property" that for any sequential structure F and strongly stable function $f : (F_*, \mathcal{C}^{\mathrm{L}}(F)) \longrightarrow qDC(X)$ there exists a strongly stable function $f' : E_* \longrightarrow F_*$ such that $\pi \circ f' = f$. It follows from the requirement that π is onto, in a uniform way. The category of "extensionally projected sequential structures" (E, X, π) is shown to be cartesian closed. The cartesian product is computed pointwise. The exponentiation $(H, Z, \pi'') = (E, X, \pi) \Rightarrow (F, Y, \pi')$ is computed as follows: Z is the exponentiation $X \Rightarrow Y$ of X and Y in the category of hypercoherence spaces, while G is a *sub-structure* of the exponentiation $E \Rightarrow F$ of E and F in the category of sequential structures, consisting of the *extensional* sequential algorithms between E and F. The lifting property of π plays a remarkable rôle in the proofs.
[van Oosten, 1997] and [Longley 1998] construct independently the same combinatory algebra, and prove that the associated realizability model of modest sets is equivalent to the strongly stable model of [Bucciarelli, Ehrhard 1991]. The combinatory algebra is based on a game-theoretic presentation of sequential evaluation, where strategies are encoded as partial functions from the set of natural numbers to itself. The result is yet another testimony that the strongly stable model of PCF is sequential in nature. [Longley 1998] goes further, and unfolds a comprehensive analysis of the strongly stable model of PCF. Developing ideas of [Ehrhard 1997], Longley establishes a key property of the strongly stable model: that there exists a *universal* simple type $\overline{2}$ of degree 2, universal in the sense that every interpretation $[T]_{\mathrm{HC}}$ of a simple type T is a retract of its interpretation $[\overline{2}]_{\mathrm{HC}}$ in the model. Longley deduces from this universality property an alternative proof that the strongly stable model of PCF is the extensional collapse of the concrete data structure model.
In [Ehrhard 2000], Ehrhard defines the dual categories of *parallel* and *serial* hypercoherence spaces, and proves that every hypercoherence space X may be projected canonically to a parallel (resp. serial) hypercoherence space $P(X)$ (resp. $S(X)$).

Using the two projection maps $\pi_S : S(X) \longrightarrow X$ and $\pi_P : X \longrightarrow P(X)$, one unfolds any hypercoherence space X as a *serial-parallel* hypercoherence space, and expresses this way the sequential game underlying X. By construction, the projections π_S and π_P enjoy the same lifting properties as the projection maps π in [Ehrhard 1996].
In this article, Ehrhard's programme is to *extract* the sequential game from the hypercoherence space, by a series of *parallel* and *serial* unfoldings. In a sense, Ehrhard's direction is just reverse to the direction we take here:

$$\begin{tikzcd}
\text{Sequential games} \arrow[r, bend left, "\text{here}"] & \text{Hypercoherence spaces} \arrow[l, bend left, "\text{[Ehrhard2000][Boudes2002]}"]
\end{tikzcd}$$

Ehrhard's student Boudes has carried on in this direction, and obtained interesting results in his PhD thesis [Boudes 2002].

2 Preliminaries

2.1 Sets

Given two sets E and F, we write $E \subset F$ when E is a subset of F, $E \subset_{\text{fin}} F$ when E is a finite subset of F, $E \subset_{\text{fin}}^{*} F$ when E is a non-empty finite subset of F. We write $\mathcal{P}(E)$ the set of the subsets of E, and $\mathcal{P}_{\text{fin}}^{*}(E)$ the set of the non-empty finite subsets of E.
Definition 2.1 (multisection) *Given a set E and a subset W of $\mathcal{P}(E)$, we call* multisection *of W any set $v \subset E$ such that*
- *for every $w \in W$, $v \cap w$ is non-empty,*
- *for every $e \in v$, there exists $w \in W$ such that $e \in w$.*

2.2 Relations

A relation between E and F is a subset of $E \times F$. The category REL has sets as objects and relations between E and F as morphisms from E to F. The identity of E is the relation

$$\mathbf{id}_E \;=\; \{\ (x, x) \mid x \in E\}$$

and the composite of two relations $f : E \longrightarrow F$ and $g : F \longrightarrow G$ is the relation $f; g : E \longrightarrow G$

$$f; g \;=\; \{(x, z) \mid\ \exists y \in F,\ (x, y) \in f \text{ and } (y, z) \in g\}$$

2.3 Words

For a natural number $k \in \mathbb{N}$, we write:

$$[k] = \{0, 1, ..., k-1\} = \{i \in \mathbb{N} \mid i < k\}$$

We call *alphabet* M any denumerable set, and *word* on this alphabet any finite sequence of elements of M. The set of words on the alphabet M defines a monoid M^* with product concatenation of words denoted "$\cdot$" and unit the empty word ϵ. A word $s \in M^*$ is *prefix* of a word t, what we write $s \sqsubseteq t$ when there exists a word u such that $t = s \cdot u$. We write $s \sqsubseteq^{\text{even}} t$ when s is prefix of t and s is of even-length. We call *polarized alphabet* (M, λ) any alphabet equipped with a function $\lambda : M \longrightarrow \{-1, +1\}$. We say that a word $m_0 \cdots m_k$ is alternating when:

$$\forall i \in [k], \lambda(m_{i+1}) = -\lambda(m_i)$$

We note $M_A^{\circledast}$ the set of alternating words on the polarized alphabet $A = (M_A, \lambda_A)$ which are either empty or start with a negative letter.

2.4 Models of intuitionistic linear logic

Linear logic (LL) is exposed in [Girard 1995]. Here, we restrict ourselves to intuitionistic linear logic (ILL) because it is sufficient to construct hierarchies of simple types, see next Section 2.5. The formulas of ILL are given by the grammar:

$$T \;=\; T \otimes T \mid T \multimap T \mid T \& T \mid !T \mid \mathbf{1} \mid \top$$

The sequent calculus of ILL is recalled in figure (2).

There exist several categorical definitions of a model of ILL. The definition 2.3 below is introduced in [Melliès 2003]. The axiomatization ensures that the equalities required by a model of ILL, recalled in [Bierman 1995], are satisfied in the category.

Definition 2.2 (exponential modality) *An exponential modality over a symmetric monoidal category* $(\mathcal{C}, \otimes, 1)$ *with finite products* $(\&, \top)$ *is given by the following data:*

- *for every object* A*, a commutative comonoid* $(!A, d_A, e_A)$ *with respect to the tensor product,*
- *for every object* A*, a morphism* $\mathbf{der}_A :!A \longrightarrow A$*, such that for every morphism*

$$f :!A \longrightarrow B$$

axiom	$\dfrac{}{A \vdash A}$	cut	$\dfrac{\Delta \vdash A \qquad \Gamma, A \vdash B}{\Gamma, \Delta \vdash B}$
$\otimes$ left	$\dfrac{\Gamma, A, B \vdash C}{\Gamma, A \otimes B \vdash C}$	$\otimes$ right	$\dfrac{\Gamma \vdash A \qquad \Delta \vdash B}{\Gamma, \Delta \vdash A \otimes B}$
$\multimap$ left	$\dfrac{\Delta \vdash A \qquad \Gamma, B \vdash C}{\Gamma, \Delta, A \multimap B \vdash C}$	$\multimap$ right	$\dfrac{\Gamma, A \vdash B}{\Gamma \vdash A \multimap B}$
1 left	$\dfrac{\Gamma \vdash A}{\Gamma, \mathbf{1} \vdash A}$	1 right	$\dfrac{}{\vdash \mathbf{1}}$
& right	$\dfrac{\Gamma \vdash A \qquad \Gamma \vdash B}{\Gamma \vdash A \& B}$	true	$\dfrac{}{\Gamma \vdash \top}$
& left-1	$\dfrac{\Gamma, A \vdash C}{\Gamma, A \& B \vdash C}$	& left-2	$\dfrac{\Gamma, B \vdash C}{\Gamma, A \& B \vdash C}$
dereliction	$\dfrac{\Gamma, A \vdash B}{\Gamma, !A \vdash B}$	promotion	$\dfrac{!\Gamma \vdash A}{!\Gamma \vdash !A}$
weakening	$\dfrac{\Gamma \vdash B}{\Gamma, !A \vdash B}$	contraction	$\dfrac{\Gamma, !A, !A \vdash B}{\Gamma, !A \vdash B}$
	exchange	$\dfrac{\Gamma, A_1, A_2, \Delta \vdash B}{\Gamma, A_2, A_1, \Delta \vdash B}$	

Fig. 2. Sequent calculus of intuitionistic linear logic (ILL)

there exists a unique comonoidal morphism

$$f^\dagger : (!A, d_A, e_A) \longrightarrow (!B, d_B, e_B)$$

making the diagram below commute:

$$\begin{tikzcd} !A \arrow[r, "f^\dagger"] \arrow[dr, "f"'] & !B \arrow[d, "\mathbf{der}_B"] \\ & B \end{tikzcd} \tag{7}$$

- *for every objects A, B, two comonoidal isomorphisms:*

$$(!A, d_A, e_A) \otimes (!B, d_B, e_B) \cong (!A\&B, d_{A\&B}, e_{A\&B})$$

$$(1, \rho_1^{-1} = \lambda_1^{-1}, \mathbf{id}_1) \cong (!\top, d_\top, e_\top)$$

Definition 2.3 (model) *A categorical model of ILL is a symmetric monoidal closed category* $(\mathcal{C}, \otimes, \multimap, 1)$ *with finite products* $(\&, \top)$ *equipped with an* exponential modality. *When in addition, the category* $\mathcal{C}$ *is* $*$*-autonomous, we say that it is a categorical model of LL.*

Remark. We will generally consider models of ILL in which the category $\mathcal{C}$ contains *two* distinguished objects bool and nat. This enables to construct a hierarchy of types over the boolean type o and natural number type ι.

2.5 Hierarchies of types

The class of simple types T over the booleans o and the integers ι is given by the grammar below:

$$T \ ::= \ o \mid \iota \mid T \Rightarrow T$$

A *hierarchy* is a family of sets $[T]$ indexed by simple types T, and a family of functions:

$$\cdot_{T_1 T_2} : [T_1 \Rightarrow T_2] \times [T_1] \ \longrightarrow \ [T_2]$$

Given $f \in [T_1 \Rightarrow T_2]$ and $x \in [T_1]$, we write $f \cdot_{T_1 T_2} x$ or even $f \cdot x$ for the image in $[V]$ of the pair (f, x) by the function $\cdot_{T_1 T_2}$. Every model $(\mathcal{C}, !)$ of intuitionistic linear logic equipped with a pair of objects bool and nat of the category $\mathcal{C}$, induces a hierarchy by Girard's formula:

$$[o] = \mathsf{bool} \qquad [\iota] = \mathsf{nat} \qquad\qquad [T_1 \Rightarrow T_2] \ = \ (!\,[T_1]) \ \multimap \ [T_2]$$

Every object $[T]$ of the category $\mathcal{C}$ is regarded as the hom-set $\mathbf{Hom}_{\mathcal{C}}(1, [T])$ of its elements. The function $\cdot_{T_1 T_2} : [T_1 \Rightarrow T_2] \times [T_1] \longrightarrow [T_2]$ associates the composite $f \cdot x : 1 \longrightarrow [T_2]$

$$1 \xrightarrow{\ f\cdot x\ } [T_2] \qquad = \qquad 1 \xrightarrow{\qquad x^\dagger \qquad} ![T_1] \xrightarrow{\qquad [\![f]\!] \qquad} [T_2]$$

to the pair $x : 1 \longrightarrow [T_1]$ and $f : 1 \longrightarrow [T_1 \Rightarrow T_2]$. Here, the morphism $[\![f]\!]$ denotes the "co-name" of f, that is the morphism $![T_1] \longrightarrow [T_2]$ associated by monoidal closure to the element $f : 1 \longrightarrow (![T_1]) \multimap [T_2]$

2.6 Extensional collapse

A hierarchy is *extensional* when, for every type $T_1 \Rightarrow T_2$ and elements f, g of $[T_1 \Rightarrow T_2]$, one has:

$$(\forall x \in [T_1], f \cdot x = g \cdot x) \quad \Rightarrow \quad f = g$$

Every hierarchy $([T], \cdot_{T_1 T_2})$ and pair of partial equivalence relations $\sim_o$ on $[o]$ and $\sim_\iota$ on $[\iota]$ induces an extensional hierarchy called the *extensional collapse* of $([T], \cdot_{T_1 T_2})$ modulo $\sim_o$ and $\sim_\iota$. The construction goes as follows. Every set $[T]$ is equipped with a partial equivalence relation $\sim_T$ defined by induction:

— $\sim_o$ and $\sim_\iota$ are the partial equivalence relations given on $[o]$ and $[\iota]$,

— $f \ \sim_{T_1 \Rightarrow T_2} \ g \quad \stackrel{defn}{\Longleftrightarrow} \quad \forall x, y \in [T_1], \quad x \ \sim_{T_1} \ y \Rightarrow f \cdot x \ \sim_{T_2} \ g \cdot y.$

The extensional collapse $([T]_{\text{ext}}, \bullet_{T_1 T_2})$ is defined in a straightforward fashion: $[T]_{\text{ext}}$ denotes the set $[T]/\sim_T$ of $\sim_T$-classes in $[T]$; while $\overline{f} \bullet_{T_1 T_2} \overline{a}$ denotes the $\sim_{T_2}$-class of $f \cdot_{T_1 T_2} a$, for every two elements f of the $\sim_{T_1 \Rightarrow T_2}$-class $\overline{f}$ and a of the $\sim_{T_1}$-class $\overline{a}$. We leave the reader check that the definition is correct, and induces an extensional hierarchy $([T]_{\text{ext}}, \bullet_{T_1 T_2})$.

3 Hypergraphs: a polarized variant of hypercoherence spaces

We introduce the hypergraph model of linear logic, a polarized variant of the hypercoherence space model presented in [Ehrhard 1993]. We show that the hypergraph and the hypercoherence space models coincide on simple types, and thus deliver alternative “linearizations” of the strongly stable hierarchy. We also indicate briefly why the hypergraph model is closer to sequentiality than the hypercoherence space model when one considers formulas *outside* the intuitionistic fragment.

3.1 Two equivalent definitions of hypergraphs

A hypergraph X may be seen alternatively:

(1) as a relaxed notion of hypercoherence space in which an element $x \in |X|$ is not necessarily equivalent to itself (definition 3.2),

(2) as a hypercoherence space equipped with a function $\lambda_X : |X| \longrightarrow \{-1, +1\}$ which polarizes every element of the web (definition 3.3.)

Before discussing the two definitions of hypergraphs, we recall the definition of hypercoherence space in [Ehrhard 1993].

Definition 3.1 (Ehrhard) *A hypercoherence space $X = (|X|, \Gamma(X))$ is a pair consisting of:*

(1) an enumerable set $|X|$ called the web *of X, whose elements are called the* atoms *of X,*

(2) a subset $\Gamma(X)$ of $\mathcal{P}^*_{\text{fin}}(|X|)$, called the atomic coherence *of A, such that for any $x \in |X|$, $\{x\} \in \Gamma(X)$.*

A hypercoherence space X with web $|X|$ is also characterized by its strict *atomic coherence, the set $\Gamma^*(X)$ of all sets $u \in \Gamma(X)$ not singleton.*

The first definition of hypergraph, as a relaxed notion of hypercoherence space, is given below:

Definition 3.2 (hypergraph (1)) *A hypergraph $X = (|X|, \breve{\Gamma}(X))$ is a pair consisting of:*

(1) an enumerable set $|X|$ called the web *of X,*

(2) a subset $\breve{\Gamma}(X)$ of $\mathcal{P}^*_{\text{fin}}(|X|)$, called the polarized atomic coherence *of A.*

Every hypergraph $X = (|X|, \breve{\Gamma}(X))$ induces a hypercoherence space $(|X|, \Gamma(X))$:

$$v \in \Gamma(X) \stackrel{defn}{\Longleftrightarrow} v \text{ is singleton or } v \in \breve{\Gamma}(X)$$

and a function $\lambda_X : |X| \longrightarrow \{-1, +1\}$ associating a polarity to every atom of the web:

$$\lambda_X(x) = +1 \stackrel{defn}{\Longleftrightarrow} \{x\} \in \breve{\Gamma}(X)$$

Conversely, every hypercoherence space $X = (|X|, \Gamma(X))$ equipped with a function $\lambda_X : |X| \longrightarrow \{-1, +1\}$ induces a hypergraph $(|X|, \breve{\Gamma}(X))$:

$$v \in \breve{\Gamma}(X) \stackrel{defn}{\Longleftrightarrow} \begin{cases} v \in \Gamma(X) & \text{if } v \text{ is not singleton} \\ \lambda_X(x) = +1 & \text{if } v \text{ is the singleton } \{x\}. \end{cases}$$

This leads to the second definition of hypergraph, as a polarized hypercoherence space:

Definition 3.3 (hypergraph (2)) *A hypergraph $X = (|X|, \Gamma(X), \lambda_X)$ is a hypercoherence space equipped with a function $\lambda_X : |X| \longrightarrow \{-1, +1\}$. An atom $x \in |X|$ is called* positive *or* negative *depending on the sign of $\lambda_X(x)$.*

Remark. From now on, we shall consider all hypergraphs X as either presented by a pair $(|X|, \breve{\Gamma}(X))$ or by a triple $(|X|, \Gamma(X), \lambda_X)$. Note that a hypergraph with web $|X|$ is characterized by its polarity function $\lambda_X : |X| \longrightarrow \{-1, +1\}$ and its *strict* atomic coherence, the set $\Gamma^*(X)$ of all sets $u \in \Gamma(X)$ not singleton.

3.2 Cliques and augmented cliques of a hypergraph

Definition 3.4 (clique, augmented clique) *Suppose that X is a hypergraph.*
- *a non-empty finite set $v \subset^*_{\text{fin}} |X|$ of atoms is* coherent *in X when $v \in \breve{\Gamma}(X)$,*
- *a set $w \subset |X|$ of atoms is a* clique *of X when:*

$$\forall v \subset^*_{\text{fin}} w, \qquad v \in \breve{\Gamma}(X)$$

- *a set $w \subset |X|$ of atoms is an* augmented clique *of X when:*

$$\forall v \subset^*_{\text{fin}} w, \qquad v \in \Gamma(X)$$

Remark. A clique is an augmented clique containing only *positive* atoms.

3.3 The hypergraph vs. the hypercoherence space models of LL

The hypergraph model of linear logic is defined essentially in the same way as the hypercoherence space model presented in [Ehrhard 1993]. There are three main differences though:
- the coherence $\breve{\Gamma}(X^{\perp})$ of the dual is *exactly* the complement of the coherence $\breve{\Gamma}(X)$. This means that every atom $x \in |X|$ on a hypergraph X changes polarity in the dual hypergraph $X^{\perp}$. Intuitively, an atom $x \in |X|$ of a hypergraph is "sequentially realized" by plays with last Player move when x is positive, and with last Opponent move when x is negative.
- the web of $X \otimes Y$ (and thus of $X \multimap Y$) is not the cartesian product of the web of X and Y, because it does not contain the pairs $(x, y) \in |X| \times |Y|$ of negative atoms. Intuitively, the web of $X \otimes Y$ picks only the "sequentially realizeable" atoms of the web of $X \otimes Y$.
- the web of $!X$ is not the set of finite cliques of X, but the set of augmented cliques of X with *at most* one negative atom. Again, intuitively, the web of $!X$ picks only the "sequentially realizeable" augmented cliques of X. This definition should be compared with the definition of the exponential $!A$ of a sequential data structure A (see Section 4.) Similarly, a play s of the sds $!A$ "explores" an augmented strategy of A which contains *at most* one odd-length play.

3.4 Duality, multiplicatives and additives

The dual of a hypergraph $X = (|X|, \Gamma(X), \lambda_X)$ is the hypergraph $X^{\perp}$ with web $|X^{\perp}| = |X|$ and polarity $\lambda_{X^{\perp}} = -\lambda_X$ and atomic coherence

$$\Gamma(X^{\perp}) = \mathcal{P}^*_{\text{fin}}(|X|) - \Gamma^*(X) \tag{8}$$

The tensor product of two hypergraphs X and Y is the hypergraph $X \otimes Y$ with web

$$|X \otimes Y| = \{(x,y) \in |X| \times |Y| \mid \lambda_X(x) = +1 \text{ or } \lambda_Y(y) = +1\} \tag{9}$$

polarity function

$$\lambda_{X \otimes Y}(x,y) = \lambda_X(x)\lambda_Y(y)$$

and atomic coherence

$$\Gamma(X \otimes Y) = \{w \in \mathcal{P}^*_{\text{fin}}(|X| \times |Y|) \mid w{\restriction}X \in \Gamma(X) \text{ and } w{\restriction}Y \in \Gamma(Y)\}$$

where $w{\restriction}X$ (resp. $w{\restriction}Y$) is the projection of w on $|X|$ (resp. $|Y|$).
The linear implication of two hypergraphs X and Y is defined by de Morgan:

$$X \multimap Y = (X \otimes Y^{\perp})^{\perp}$$

So, by definition, the hypergraph $X \multimap Y$ has web:

$$|X \multimap Y| = \{(x,y) \in |X| \times |Y| \mid \lambda_X(x) = +1 \text{ or } \lambda_Y(y) = -1\}$$

polarity function

$$\lambda_{X \multimap Y}(x,y) = \lambda_X(x)\lambda_Y(y)$$

and atomic coherence $\Gamma(X \multimap Y)$ the set of all $w \in \mathcal{P}^*_{\text{fin}}(|X| \times |Y|)$ such that

$$w{\restriction}X \in \Gamma(X) \Rightarrow w{\restriction}Y \in \Gamma(Y) \quad \text{and} \quad w{\restriction}X \in \Gamma^*(X) \Rightarrow w{\restriction}Y \in \Gamma^*(Y)$$

where $w{\restriction}X$ (resp. $w{\restriction}Y$) is the projection of w on $|X|$ (resp. $|Y|$).
The product of two hypergraphs X and Y is the hypergraph $X \& Y$ with web

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

and atomic coherence $\Gamma(X \& Y)$ the set of all $w \in \mathcal{P}^*_{\text{fin}}(|X| + |Y|)$ such that

$$w{\restriction}X = \emptyset \Rightarrow w{\restriction}Y \in \Gamma(Y) \quad \text{and} \quad w{\restriction}Y = \emptyset \Rightarrow w{\restriction}X \in \Gamma(X)$$

where $w{\restriction}X$ (resp. $w{\restriction}Y$) is the projection of w on $|X|$ (resp. $|Y|$).
The unit $\top$ is the hypergraph with empty web; and the unit $\mathbf{1}$ is the hypergraph with singleton web $\{*\}$ and atomic coherence $\{\{*\}\}$.

3.5 A $*$-autonomous category of hypergraphs

The category HG has hypergraphs as objects and cliques of $X \multimap Y$ as morphisms from X to Y. Morphisms are composed as relations in the category REL, and the identity $\mathbf{id}_X : X \longrightarrow X$ is the clique $\{(x,x) \mid x \in |X|\}$ of $X \multimap X$.

Lemma 3.5 *The category* $(\mathrm{HG}, \otimes, 1)$ *is* $*$*-autonomous, and has finite products given by* $(\&, \top)$.

3.6 Exponentials

The exponential of a hypergraph X is the hypergraph $!X$

- with web $|!X|$ the set of finite augmented cliques of X, containing a negative atom at most,
- with polarity $\lambda_{!A}(w) = +1$ when w is a clique, and $\lambda_{!A}(w) = -1$ when w is an augmented clique containing one negative atom,
- with atomic coherence $\Gamma(!X)$ the set of all $W \subset^*_{\mathrm{fin}} |!X|$ whose every multisection w is coherent in X.

The hypergraph $!X$ defines a commutative comonoid with comultiplication d_X defined as union of augmented cliques, and counity e_X defined as the empty clique. Given a hypergraph X, the dereliction clique is defined as:

$$\mathbf{der}_X = \{(\{x\}, x) \mid x \in |X| \text{ and } \{x\} \in \Gamma(X)\}.$$

Given a clique $f : (!X \multimap Y)$, the clique $f^\dagger : (!X \multimap !Y)$ is defined as:

$$f^\dagger = \{(u,v) \in |!X| \times |!Y| \mid \exists (u_i, x_i) \in f,\ u = u_1 \cup ... \cup u_n \text{ and } v = \{x_1, ..., x_n\}\}.$$

This clique $f^\dagger$ is the unique comonoidal morphism $!X \longrightarrow !Y$ making diagram (7) commute. Besides, the comonoids $!X \otimes !Y$ and 1 are isomorphic (as comonoids) to the comonoids $!(X \& Y)$ and $!\top$, for every hypergraphs X and Y. It follows that $!$ defines an exponential modality on the $*$-autonomous category HG. We conclude from definition 2.3 that

Lemma 3.6 *The category* HG *equipped with the exponential modality* $!$ *defines a model of linear logic.*

3.7 The strongly stable hierarchy $[-]_{\mathrm{HC}}$ of simple types

We explain briefly why the hypergraph model delivers the same hierarchy of simple types (noted $[T]_{\mathrm{HC}}$) as the hypercoherence space model in [Ehrhard 1993]. First, we note that a hypercoherence space may be seen as a particular kind of hypergraph.

Definition 3.7 (hypercoherence space 2) *A hypergraph* $X = (|X|, \Gamma(X), \lambda_X)$ *is called a* hypercoherence space *when every atom* $x \in |X|$ *has polarity* $\lambda_X(x) = +1$. The hierarchy $[-]_{\mathrm{HC}}$ is induced by the hypergraph model, in which the base types o and ι are interpreted as the hypercoherence spaces $[o]_{\mathrm{HC}} = \mathbb{B}_{\mathrm{HC}}$ and $[\iota]_{\mathrm{HC}} = \mathbb{N}_{\mathrm{HC}}$ with webs:

$$|\mathbb{B}_{\mathrm{HC}}| = \{V, F\} \qquad\qquad |\mathbb{N}_{\mathrm{HC}}| = \mathbb{N}$$

and atomic coherence the set of singletons:

$$\breve{\Gamma}(\mathbb{B}_{\mathrm{HC}}) = \{\{V\}, \{F\}\} \qquad\qquad \breve{\Gamma}(\mathbb{N}_{\mathrm{HC}}) = \{\{n\}, n \in \mathbb{N}\}$$

Lemma 3.8 *The hierarchy of types* $[-]_{\mathrm{HC}}$ *coincides with the strongly stable function.*

PROOF Hypercoherence spaces (in the sense of definition 3.7) are preserved by the connectives of $\otimes$, $\multimap$, $\&$ and $!$ in the hypergraph model of ILL. Besides, the interpretations of these connectives on hypercoherence spaces, as well as the base types o and ι, coincides in the hypergraph model and in the original hypercoherence space model presented in [Ehrhard 1993]. It follows that the hypergraph hierarchy $[-]_{\mathrm{HC}}$ coincides with the strongly stable hierarchy of [Bucciarelli, Ehrhard 1991]. ■

4 Sequential data structures

In this section, we recall the sequential data structure (sds) model of intuitionistic linear logic introduced by Lamarche around 1992. We already mentioned that the hierarchy of simple types it generates coincides with the *sequential algorithm* hierarchy on concrete data structures introduced by Berry and Curien. This game model is described for the first time in [Lamarche 1992]. Our presentation follows the later presentation by Curien in [Curien 1993] [Amadio, Curien 1998].

4.1 Sequential data structures

Definition 4.1 (sds) *A sequential data structure is a triple* $A = (M_A, \lambda_A, P_A)$ *consisting of*

- *a polarized alphabet* (M_A, λ_A) *whose elements are called the* moves *of* A,
- *a set* P_A *of words on the alphabet* M_A*, whose elements are called the* plays *of* A.

A move m *is called a cell when* $\lambda_A(m) = -1$ *and a value when* $\lambda_A(m) = +1$.

Every sds is required to verify:

- the empty play ϵ is a play,
- the prefix of a play is a play,

- every non-empty play is alternating and starts by a cell:

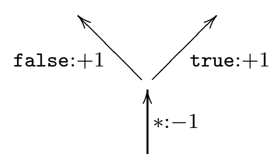
$$\forall m \in M_A, \qquad m \in P_A \Rightarrow \lambda_A(m) = -1,$$

$$\forall s \in P_A, \forall m, n \in M_A, \qquad s \cdot m \cdot n \in P_A \Rightarrow \lambda_A(m) = -\lambda_A(n).$$

A sds may be visualized as a rooted directed tree with plays as vertices and moves as edges. For example, the sds $\mathbb{B}$ defined as:

$$M_{\mathbb{B}} = \{*, \mathtt{false}, \mathtt{true}\} \qquad \lambda_{\mathbb{B}} = \begin{cases} * : -1 \\ \mathtt{false} : +1 \\ \mathtt{true} : +1 \end{cases} \qquad P_{\mathbb{B}} = \{\epsilon\} \cup \left\{ \begin{array}{l} * \\ * \cdot \mathtt{false} \\ * \cdot \mathtt{true} \end{array} \right\}$$

is represented as the labelled tree:



Remark. We often write $m : -1$ when m is a cell, and $m : +1$ when m is a value. We also write $P_A^{\mathbf{even}}$ and $P_A^{\mathbf{odd}}$ for the set of even-length and odd-length plays of a sds A, respectively.

4.2 Strategies and augmented strategies

Definition 4.2 (strategy) *A strategy of a sds A is a set of plays $\sigma \subset P_A^{\mathbf{even}}$ of even-length, which verifies that:*

- *it is closed under even-length prefix:*

$$\forall s, t \in P_A, \quad s \sqsubseteq_A^{\text{even}} t \text{ and } t \in \sigma \;\Rightarrow\; s \in \sigma,$$

- *it is deterministic:*

$$\forall s \in P_A^{\mathbf{even}}, \forall m, n_1, n_2 \in M_A \quad s \cdot m \cdot n_1 \in \sigma \text{ and } s \cdot m \cdot n_2 \in \sigma \;\Rightarrow\; n_1 = n_2,$$

- *it is nonempty: $\epsilon \in \sigma$.*

Definition 4.3 (substrategy) *Let σ and τ be two strategies of a sds A. We say that σ is a substrategy of τ when $\sigma \subset \tau$.*

Definition 4.4 (augmented strategy) *An augmented strategy of A is a set of plays $\sigma \subset P_A$ verifying that:*
- *$\sigma \cap P_A^{\mathbf{even}}$ is a strategy of A,*
- *every odd-length play $t \in \sigma \cap P_A^{\mathbf{odd}}$ factorizes as $t = s \cdot m$ where:*
  - *s is a $\sqsubseteq$-maximal play in the strategy $\sigma \cap P_A^{\mathbf{even}}$,*
  - *$m \in M_A$ is a cell of A.*

We write $\sigma : A$ when σ is a strategy or an augmented strategy of a sds A.

4.3 Multiplicatives and additives

The tensor product of two sdss A and B is the sds $A \otimes B$:
(1) $M_{A\otimes B} = M_A + M_B$,
(2) $\lambda_{A\otimes B}(\mathtt{inl}(m)) = \lambda_A(m)$ and $\lambda_{A\otimes B}(\mathtt{inr}(m)) = \lambda_B(m)$,
(3) $P_{A\otimes B} = \{s \in M_{A\otimes B}^{\circledast},\ s{\upharpoonright}A \in P_A$ and $s{\upharpoonright}B \in P_B\}$.
The linear implication of two sdss A and B is the sds $A \multimap B$:
(1) $M_{A\multimap B} = M_A + M_B$,
(2) $\lambda_{A\multimap B}(\mathtt{inl}(m)) = \lambda_A(m)$ and $\lambda_{A\multimap B}(\mathtt{inr}(m)) = -\lambda_B(m)$,
(3) $P_{A\multimap B} = \{s \in M_{A\multimap B}^{\circledast},\ s{\upharpoonright}A \in P_A$ and $s{\upharpoonright}B \in P_B\}$.
The product of two sdss A and B is the sds $A\&B$:
(1) $M_{A\&B} = M_A + M_B$,
(2) $\lambda_{A\&B} = \lambda_A + \lambda_B$,
(3) $P_{A\&B} = \mathtt{inl}^*(P_A) + \mathtt{inr}^*(P_B)$.
The units 1 and $\top$ are equal to the sds with an empty set of moves.

4.4 A symmetric monoidal closed category of sequential data structures

The category SDS has the sequential data structures as objects, and the strategies of $A \multimap B$ as morphisms from A to B. The identity map $\mathbf{id}_A : A \longrightarrow A$ is the "copycat" strategy $\mathbf{id}_A : A \multimap A$, defined as:

$$\mathbf{id}_A = \{s \in P_{A\multimap A}^{\mathbf{even}} \mid \forall t \sqsubseteq^{\text{even}} s,\ t{\upharpoonright}A_1 = t{\upharpoonright}A_2\}.$$

The composite of two strategies $\sigma : A \longrightarrow B$ and $\tau : B \longrightarrow C$ is the strategy $\sigma;\tau : A \longrightarrow C$ defined by "parallel composition plus hiding":

$$\sigma;\tau \ = \ \{s \in P_{A\multimap C} \mid \forall t \sqsubseteq^{\text{even}} s, \exists u \in \sigma, \exists v \in \tau,$$
$$t{\upharpoonright}A = u{\upharpoonright}A,\ u{\upharpoonright}B = v{\upharpoonright}B,\ v{\upharpoonright}C = t{\upharpoonright}C\}$$

We refer the reader to [Abramsky, Jagadeesan 1994] or [Curien 1993] for a proof that the composition law is associative, and that the strategies $\mathbf{id}_A : A \longrightarrow A$ define

proper identities. Besides, one establishes that:
Lemma 4.5 *The category* SDS *is symmetric monoidal closed, and cartesian.*

4.5 Exploration of augmented strategies

Definition 4.6 ($\sigma\downarrow$) *Suppose that $\sigma : A$ is an augmented strategy. We note*

$$\sigma\downarrow = \{s \in P_A \mid \exists t \in \sigma, s \sqsubseteq_A t\}.$$

Remark. The set $\sigma\downarrow$ may be seen as an Opponent-branching subtree of the sds A. Note that every augmented strategy σ may be recovered from the subtree $\sigma\downarrow$, as its subset of even-length plays and of maximal and odd-length plays.
Definition 4.7 ($\longrightarrow_{\mathbf{btk}}$) *Suppose that σ and τ are augmented strategies of a sds A, and that $t \in P_A \setminus \{\epsilon\}$ is a nonempty play. We write*

$$\sigma \xrightarrow{t}_{\mathbf{btk}} \tau \overset{defn}{\iff} \tau\downarrow = \sigma\downarrow + \{t\}.$$

The notation $+$ means that $\sigma\downarrow = \tau\downarrow \cup \{t\}$ and that t is not element of $\sigma\downarrow$.
Definition 4.8 (exploration) *We say that a word $t = t_0 \cdots t_{n-1}$ on the alphabet $P_A \setminus \{\epsilon\}$* explores *an augmented strategy σ of A when:*

$$\{\epsilon\} \xrightarrow{t_0}_{\mathbf{btk}} \tau_0 \xrightarrow{t_1}_{\mathbf{btk}} \cdots \tau_1 \xrightarrow{t_2}_{\mathbf{btk}} \cdots \xrightarrow{t_{n-1}}_{\mathbf{btk}} \tau_n = \sigma$$

For instance, the two words on the alphabet $P_{\mathbb{B}\otimes\mathbb{B}} \setminus \{\epsilon\}$:

$$*_1 \cdot (*_1 \cdot \mathtt{true}) \cdot *_2 \cdot (*_2 \cdot \mathtt{false}) \quad \text{and} \quad *_2 \cdot (*_2 \cdot \mathtt{false}) \cdot *_1 \cdot (*_1 \cdot \mathtt{true})$$

explore the strategy $\sigma = \{\epsilon, *_1 \cdot \mathtt{true}, *_2 \cdot \mathtt{false}\}$ of the sds $\mathbb{B} \otimes \mathbb{B}$.

4.6 Exponentials

One distinctive feature of Lamarche's model is the interpretation of the exponential modality of linear logic. In this model, the sequential data structure $!A$ is interpreted by interleaving the plays of A *without repetition*, using a clever *backtracking* device in the definition of the contraction map $d_A : !A \longrightarrow !A \otimes !A$. This departs from the mainstream models like [Abramsky et al. 1994] in which the exponential game $!A$ is defined by interleaving and repeating the plays of A as much as Opponent desires. Note that the two styles of exponentials may be compared by exhibiting a retraction between them, see [Melliès 2004a] for details. Formally, Lamarche defines for every sds A the exponential sds $!A$ as follows:

(1) $M_{!A} = P_A \setminus \{\epsilon\}$,
(2) $\lambda_{!A}(s) = +1$ when $s \in P_A^{\mathbf{even}}$ and $\lambda_{!A}(s) = -1$ when $s \in P_A^{\mathbf{odd}}$,
(3) $P_{!A}$ is the set of alternating words $s \in M_{!A}^{\circledast}$ which explore an augmented strategy σ of A.

Remark. Note that every play $s \in P_{!A}^{\mathbf{even}}$ explores a strategy, and that every play $s \in P_{!A}^{\mathbf{odd}}$ factors as $t \cdot m$ where $t \in P_{!A}^{\mathbf{even}}$ explores a strategy σ and s explores the augmented strategy $\sigma + \{m\}$, where $m \in M_{!A}$ and $m \in P_A^{\mathbf{odd}}$ at the same time. Consequently, only augmented strategies with *at most* one odd-length play are explored by a play in $!A$. This observation on sequential data structures motivates our interpretation of the exponential modality in the hypergraph model, in Section 3.

The sds $(!A, d_A, e_A)$ defines a commutative comonoid in the category SDS. The strategy d_A is defined in two steps. First, one says that a play $s \in P_{!A \multimap (!A \otimes !A)}$ verifies property (*) when the augmented strategies σ_1, σ_2, σ_3 explored by its first, second and third projections s_1, s_2, s_3 verify $\sigma_1 = \sigma_2 \cup \sigma_3$ (set-theoretic union). Then, one defines:

$$d_A = \{s \in P^{\mathbf{even}}_{!A \multimap (!A \otimes !A)} \mid \forall t \in P^{\mathbf{even}}_{!A \multimap (!A \otimes !A)}, t \sqsubseteq s \Rightarrow \ t \text{ verifies property (*)}\}.$$

The strategy $e_A : (!A \multimap 1)$ is defined as the singleton $\{\epsilon\}$.
The strategy $\mathbf{der}_A$ is defined in two steps. First, one says that a play $s \in P_{!A \multimap A}$ verifies property (**) when the augmented strategy σ explored by the first projection $s\restriction !A$, and the second projection $s \restriction A$ verify together: $\sigma \downarrow = \{u \in P_A \mid u \sqsubseteq s_2\}$. Then, one defines:

$$\mathbf{der}_A = \{s \in P^{\mathbf{even}}_{!A \multimap A} \mid \forall t \in P^{\mathbf{even}}_{!A \multimap A}, t \sqsubseteq s \Rightarrow \ t \text{ verifies property (**)}\}.$$

There exists for every strategy $\sigma : !A \longrightarrow B$ a unique comonoidal strategy $(\sigma)^\dagger : !A \longrightarrow !B$ making diagram (7) commute. For instance, when $A = 1$ and the strategy $\sigma : 1 \longrightarrow B$ is just a strategy of B, the comonoidal strategy $(\sigma)^\dagger : 1 \longrightarrow !B$ is defined as:

$$(\sigma)^\dagger = \{s \in P_{!B} \mid s \text{ explores a substrategy of } \sigma\}.$$

Besides, there exists a comonoidal isomorphism between $!A \otimes !B$ and $!(A \& B)$ for every sdss A and B, and a comonoidal isomorphism between 1 and $!\top$.
It follows that ! defines an exponential modality on the $*$-autonomous category SDS, and from definition 2.3, that:
Lemma 4.9 (Lamarche) *Sequential data structures define a model of intuitionistic linear logic.*

4.7 *The flat and the lazy sequential hierarchies* $[-]^{\mathbf{flat}}_{\mathrm{SDS}}$ *and* $[-]^{\mathbf{lazy}}_{\mathrm{SDS}}$

We consider *two* hierarchies of types induced by the sds model of ILL:

- the *flat* hierarchy $[-]^{\mathbf{flat}}_{\mathrm{SDS}}$ (sometimes written $[-]_{\mathrm{SDS}}$) in which o is interpreted as the sds $\mathbb{B}$ and ι is interpreted as the "flat" natural number sds $\mathbb{N}_{\mathbf{flat}}$:

$$M_{\mathbb{N}_{\mathbf{flat}}} = \{*\} \cup \{\mathtt{n} \mid n \in \mathbb{N}\} \qquad \lambda_{\mathbb{N}_{\mathbf{flat}}} = \begin{cases} * : -1 \\ \mathtt{n} : +1 \end{cases}$$

$$P_{\mathbb{N}_{\mathbf{flat}}} \;=\; \{\epsilon\} \;\cup\; \{*\} \;\cup\; \{* \cdot \mathtt{n} \mid n \in \mathbb{N}\}$$

  The *flat* hierarchy is the hierarchy considered in [Berry, Curien 1982].

- the *lazy* hierarchy $[-]^{\mathbf{lazy}}_{\mathrm{SDS}}$ in which o is interpreted as the sds $\mathbb{B}$ and ι is interpreted as the "lazy" natural number sds $\mathbb{N}_{\mathbf{flat}}$:

$$M_{\mathbb{N}_{\mathbf{lazy}}} = \left\{ \begin{array}{c|c} \geq_{\mathtt{n}} & \\ >_{\mathtt{n}} & n \in \mathbb{N} \\ =_{\mathtt{n}} & \end{array} \right\} \qquad \lambda_{\mathbb{N}_{\mathbf{lazy}}} = \begin{cases} \geq_{\mathtt{n}} : -1 \\ >_{\mathtt{n}} : +1 \\ =_{\mathtt{n}} : +1 \end{cases}$$

$$P_{\mathbb{N}_{\mathbf{lazy}}} = \{\epsilon\} \;\cup\; \left\{ \begin{array}{l|c} \geq_{\mathtt{0}} \cdots \geq_{\mathtt{n}} & \\ \geq_{\mathtt{0}} \cdots \geq_{\mathtt{n}} \cdot >_{\mathtt{n}} & n \in \mathbb{N} \\ \geq_{\mathtt{0}} \cdots \geq_{\mathtt{n}} \cdot =_{\mathtt{n}} & \end{array} \right\}$$

The two natural number sdss $\mathbb{N}_{\mathbf{flat}}$ and $\mathbb{N}_{\mathbf{lazy}}$ are represented as trees in figure 3.

5 Extensional data structures

In this section, we equip every sequential data structure with a realizability relation between plays and *extensions*, obtaining what we call an *extensional data structure* (eds). Our ambition is not to define another model of intuitionistic linear logic (we will see that the sds and eds models are equivalent), but to analyze the extensional content of the strategies in the category SDS.

5.1 *Extensional data structures*

Definition 5.1 (eds) *An extensional data structure (eds) is a six-tuple*

$$A = (M_A, \lambda_A, P_A, E_A, \| - \|_A)$$

where:

- (M_A, λ_A, P_A) *is a sequential data structure,*
- E_A *is an enumerable set whose elements are called the* extensions *of* A*,*
- $\| - \|_A$ *associates to every extension* $x \in E_A$ *a* non-empty finite *set* $\|x\|_A \subset P_A^{\mathbf{even}}$.

The plays in $\|x\|_A$ are called the *realizers* of the extension x. We ask that every extensional data structure is *modest* in the sense that:

$$\forall x, y \in E_A, \quad \|x\|_A \cap \|y\|_A \neq \emptyset \Rightarrow x = y.$$

We write R_A for the set of *realizers* of A:

$$R_A = \bigcup_{x \in E_A} \|x\|_A.$$

We illustrate the definition with the boolean eds $\mathbb{B} = (M_{\mathbb{B}}, \lambda_{\mathbb{B}}, P_{\mathbb{B}}, E_{\mathbb{B}}, \| - \|_{\mathbb{B}})$. It is defined as the boolean sds $(M_{\mathbb{B}}, \lambda_{\mathbb{B}}, P_{\mathbb{B}})$ of Section 4, now equipped with the extensional realizability structure:

$$E_{\mathbb{B}} = \{V, F\}, \qquad \|F\|_{\mathbb{B}} = \{* \cdot \mathtt{false}\}, \qquad \|V\|_{\mathbb{B}} = \{* \cdot \mathtt{true}\}.$$

5.2 Strategies

A strategy of the eds A is defined as a strategy of its underlying sds. It follows that the eds and sds models of intuitionistic linear logic are equivalent.

5.3 When does a strategy implement a set of extensions?

Definition 5.2 ($\preceq_A$) *We write $s \preceq_A x$ when $s \in P_A$ is prefix of a play t realizing an extension $x \in E_A$:*

$$s \preceq_A x \quad \stackrel{defn}{\Longleftrightarrow} \quad \exists t \in P_A,\ s \sqsubseteq_A t \text{ and } t \in \|x\|_A.$$

Definition 5.3 (implement, $\vDash_A$) *A strategy $\sigma : A$ implements an extension $x \in E_A$ when, for every play $s \in P_A$ and move $m \in M_A$ such that $s \cdot m \in P_A$, one has:*

$$s \in \sigma \text{ and } s \cdot m \preceq_A x \ \Rightarrow\ \exists n \in M_A,\ s \cdot m \cdot n \preceq_A x \text{ and } s \cdot m \cdot n \in \sigma.$$

In that case, we write:

$$\sigma \vDash_A x.$$

A strategy σ implements a set $v \subset E_A$ of extensions of A, when σ implements every extension of v, what we note $\sigma \vDash_A v$. Thus:

$$\sigma \vDash_A v \quad \overset{defn}{\Longleftrightarrow} \quad \forall x \in v,\ \sigma \vDash_A x.$$

Remark. The definition of implementation of an extension $x \in E_A$ is inspired by the definition of concurrent strategy in [Abramsky, Melliès 1999]. It should be compared with the definition of conflict-free strategy in [Hyland, Schalk 2002] and of forward confluent strategy in [Melliès 2004b]. Its task is to provide an explicit and dynamic formulation of the usual notion of *sequential realizability*, either given by extensional collapse as in Section 2.6, or by observational equivalence as in Section 4.2 of [Abramsky et al. 2000] and Section 3 of [Hyland, Ong 2000].

5.4 Configurations

Definition 5.4 (configuration) *A* configuration *of A is any set $v \subset E_A$ of extensions implemented by a strategy σ.*

5.5 Multiplicatives and additives

We adapt to edss the model of ILL presented in Section 4. The interpretation is conservative on the sds part. This enables us to limit our definitions to the realizability relation attached to each interpretation.

The tensor product of two edss A and B is the sds $A \otimes B$ equipped with the realizability relation:

(1) $E_{A\otimes B} = E_A \times E_B$,

(2) $\|(x, y)\|_{A\otimes B} = \{s \in M^{\circledast}_{A\otimes B} \mid s{\upharpoonright}A \in \|x\|_A$ and $s{\upharpoonright}B \in \|y\|_B\}$.

The linear implication of two edss A and B is the sds $A \multimap B$ equipped with the realizability relation:

(1) $E_{A\multimap B} = E_A \times E_B$,

(2) $\|(x, y)\|_{A\multimap B} = \{s \in M^{\circledast}_{A\otimes B} \mid s{\upharpoonright}A \in \|x\|_A$ and $s{\upharpoonright}B \in \|y\|_B\}$.

The product of two edss A and B is the sds $A\&B$ equipped with the realizability relation:

(1) $E_{A\&B} = E_A + E_B$,

(2) $\|\mathtt{inl}(x)\|_{A\&B} = \mathtt{inl}^*(\|x\|_A)$ and $\|\mathtt{inr}(y)\|_{A\&B} = \mathtt{inr}^*(\|y\|_B)$.

The unit $\top$ is the sds $\top$ equipped with an empty set of extensions; the unit $\mathbf{1}$ is the sds $\mathbf{1}$ equipped with a single extension $*$, realized by the empty play ϵ.

5.6 Exponentials

Definition 5.5 (sub-implement) *A strategy $\sigma : A$ sub-implements an extension $x \in E_A$ when, for every play $s \in P_A$ and moves $m, n \in M_A$:*

$$s \in \sigma \text{ and } s \cdot m \preceq_A x \text{ and } s \cdot m \cdot n \in \sigma \Rightarrow s \cdot m \cdot n \preceq_A x.$$

A strategy σ sub-implements a set $v \subset E_A$ of extensions of A, when σ sub-implements every extension of v.

This enables to define:

The exponential of an eds A is the sds $!A$ equipped with the realizability relation:

(1) $E_{!A}$ is the set of finite configurations of A,

(2) a play $s \in P_{!A}$ realizes a finite configuration $v \in E_{!A}$ when there exists a strategy σ of A such that:
- s explores the strategy σ,
- σ sub-implements the configuration v,
- $v = \{x \in E_A \mid \sigma \cap \|x\|_A \neq \emptyset\}$,
- $\forall t \in \sigma, \exists x \in v, t \preceq_A x$.

5.7 The category EDS

The category EDS has the edss as objects and the strategies of $A \multimap B$ as morphisms from A to B. Identities and composition are defined as in the category SDS. We obtain immediately that:

Lemma 5.6 *The category* EDS *is equivalent to the category* SDS*.*

5.8 The flat and the lazy sequential hierarchies $[-]^{\mathbf{flat}}_{\mathrm{SDS}}$ and $[-]^{\mathbf{lazy}}_{\mathrm{SDS}}$

We equip the flat and the lazy sequential algorithm hierarchies of types with extensional information. The simple type o is interpreted as the eds $\mathbb{B}$; the simple type ι is interpreted either (1) as the natural number sds $\mathbb{N}_{\mathbf{flat}}$ equipped with the realizability relation:

$$E_{\mathbb{N}_{\mathbf{flat}}} = \mathbb{N}, \qquad \|n\|_{\mathbb{N}_{\mathbf{flat}}} = \{* \cdot \mathtt{n}\}.$$

and (2) as the lazy natural number sds $\mathbb{N}_{\mathbf{lazy}}$ equipped with the realizability relation:

$$E_{\mathbb{N}_{\mathbf{lazy}}} = \mathbb{N}, \qquad \|n\|_{\mathbb{N}_{\mathbf{lazy}}} = \{\geq_{\mathtt{0}} \cdots \geq_{\mathtt{n}} \cdot =_{\mathtt{n}}\}.$$

The two edss $\mathbb{N}_{\mathbf{flat}}$ and $\mathbb{N}_{\mathbf{lazy}}$ are represented in figure 3. We write $[T]^{\mathbf{flat}}_{\mathrm{SDS}}$ and $[T]^{\mathbf{lazy}}_{\mathrm{SDS}}$ for the interpretations of a simple type T in the respective hierarchies.

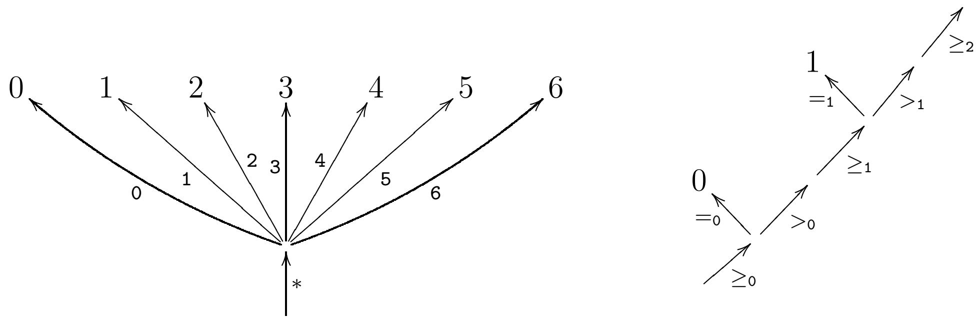


Fig. 3. The *flat* natural number eds $\mathbb{N}_{\mathbf{flat}}$ vs. the *lazy* natural number eds $\mathbb{N}_{\mathbf{lazy}}$.

6 A graphic representation for simple types

In this section, we show that every extensional data structure $[T]^{\mathbf{flat}}_{\mathrm{SDS}}$ and $[T]^{\mathbf{lazy}}_{\mathrm{SDS}}$ interpreting a simple type T, may be represented as directed acyclic graphs (dags). We start by a definition.

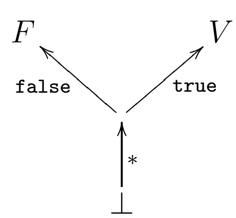
Definition 6.1 (graphic) *Let s and t be any two plays of an eds A. We write $s \sim_A t$, when, for every word u on the alphabet M_A, we have:*

$$s \cdot u \in P_A \iff t \cdot u \in P_A,$$
$$\forall x \in E_A, \qquad s \cdot u \in \|x\|_A \iff t \cdot u \in \|x\|_A.$$

An eds A is called graphic when every two plays s and t realizing the same extension $x \in E_A$ are equivalent:

$$\forall x \in E_A,\ \forall s, t \in P_A, \qquad s, t \in \|x\|_A \Rightarrow s \sim_A t.$$

Every graphic eds A is represented by the dag obtained by identifying all vertices $s \in P_A$ realizing the same extension $x \in E_A$ in the tree associated to the sds (M_A, λ_A, P_A). We have already seen in the introduction that the graphic eds $!\mathbb{B}$ is represented as the labelled tree:



and the graphic eds $!\mathbb{B} \otimes !\mathbb{B} \otimes !\mathbb{B}$ as the labelled dag (of which we only draw a fragment) of figure 1. It turns out that the class of graphic games is closed under the linear connectives: $\otimes$, $\multimap$, $\&$; but not closed under the exponential modality $!(-)$. For instance, the eds $(\mathbb{B} \otimes \mathbb{B})$ is graphic, but the eds $A = !(\mathbb{B} \otimes \mathbb{B})$ is not graphic. Let us explain why. Consider the two plays s and t of A:

$$s \;=\; (*_1) \cdot (*_1 \cdot \mathtt{true}_1) \cdot (*_1 \cdot \mathtt{true}_1 \cdot *_2) \cdot (*_1 \cdot \mathtt{true}_1 \cdot *_2 \cdot \mathtt{false}_2)$$

$$t \;=\; (*_2) \cdot (*_2 \cdot \mathtt{false}_2) \cdot (*_2 \cdot \mathtt{false}_2 \cdot *_1) \cdot (*_2 \cdot \mathtt{false}_2 \cdot *_1 \cdot \mathtt{true}_1)$$

The two plays realize the same extension $(V, F) \in E_{!(\mathbb{B}\otimes\mathbb{B})}$. The only difference is that the play s interrogates its arguments left-to-right, while the play t interrogates its arguments right-to-left. The sds model is too "sequential" to detect that s and t are just doing the same thing, and thus, the word $s \cdot t$ is accepted as a play of $!(\mathbb{B}\otimes\mathbb{B})$ and as a realizer of (V, F). So, the play $s \cdot t$ interrogates its arguments twice, the first time from left-to-right, the second time from right-to-left. It follows immediately that s and t are not $\sim_A$ equivalent (since $t \cdot t$ is not a play) and consequently, that $!(\mathbb{B}\otimes\mathbb{B})$ is not graphic. This is an interesting pathology of sequential data structures, which our analysis uncovers.

Fortunately, the defect is harmless on simple types T, which are interpreted as graphic edss $[T]^{\mathbf{flat}}_{\mathrm{SDS}}$ and even more than that: every eds $[T]^{\mathbf{flat}}_{\mathrm{SDS}}$ is *spread* in the sense given below.

Definition 6.2 (spread) *An eds A is spread when*

$$\forall x \in E_A, \forall s \in \|x\|_A, \forall t \in P_A, \quad s \sqsubseteq t \Rightarrow s = t.$$

and no extension $x \in E_A$ is realized by the empty play.

Note that every spread eds is graphic, and represented by a dag whose extensions are at the leaves. We prove that:

Lemma 6.3 *The interpretations $[T]^{\mathbf{flat}}_{\mathrm{SDS}}$ and $[T]^{\mathbf{lazy}}_{\mathrm{SDS}}$ of every simple type T is spread.*

PROOF The property follows from the three observations that,

(1) the edss $\mathbb{B}$ and $\mathbb{N}_{\mathbf{flat}}$ and $\mathbb{N}_{\mathbf{lazy}}$ are spread,

(2) the eds $!A$ is not necessarily spread when A is spread...

(3) but the eds $A \multimap B$ is spread when B is spread.

■

7 Extracting hypercoherence spaces from extensional data structures

In this section, we associate to every (regular) extensional data structure A a hypercoherence space $U(A)$ whose finite cliques are the configurations of A.

Definition 7.1 (frontier) *The* cone *of a non-empty finite set $v \subset^*_{\mathrm{fin}} E_A$ of extensions* *is defined as follows:*

$$\mathbf{cone}(v) = \{s \in P_A \mid \forall x \in v, s \preceq_A x\}.$$

The frontier of v *is the set of* $\sqsubseteq_A$*-maximal plays in the cone of* v*:*

$$\mathbf{frontier}(v) = \mathsf{max}_{\sqsubseteq_A}(\mathbf{cone}(v)).$$

Remark. The cone of a set $v \subset^*_{\text{fin}} E_A$ of extensions is always *finite*. It follows that $\mathbf{frontier}(v)$ is nonempty.

Definition 7.2 (coherence) *A non-empty finite subset $v \subset^*_{\text{fin}} E_A$ of extensions is declared:*

- coherent *in A when $\mathbf{frontier}(v) \subset P_A^{\mathbf{even}}$,*
- incoherent *in A when $\mathbf{frontier}(v) \subset P_A^{\mathbf{odd}}$.*

Definition 7.3 (regular) *An eds is regular when every non-empty finite subset $v \subset^*_{\text{fin}} E_A$ of extensions is either coherent or incoherent.*

To every regular eds A we associate the hypergraph

$$U(A) = (|U(A)|, \breve{\Gamma}(U(A))).$$

defined as follows:

- $|U(A)| = E_A$,
- $\breve{\Gamma}(U(A))$ contains the coherent subsets of E_A.

Remark. We required in our definition of an eds A that $\|x\|_A \subset P_A^{\mathbf{even}}$ for every extension $x \in E_A$. It follows that every extension $x \in E_A$ is coherent in an eds A; and from this, that the hypergraph $U(A)$ is a hypercoherence space in the sense of definition 3.7: every atom $x \in |U(A)|$ is positive. So, the advantage of using hypergraphs instead of hypercoherence spaces is only visible when one moves outside the intuitionistic hierarchy, with a less constrained notion of sds and eds.

The definition of the hypercoherence space $U(A)$ is motivated by the result below:

Lemma 7.4 (configuration=clique (finite case)) *Suppose that A is a regular eds and that v is a finite subset of E_A. Then, the following are equivalent:*

(1) v is a configuration of A,

(2) v is a clique of $U(A)$.

PROOF $(1 \Rightarrow 2)$ Let w be any nonempty finite subset of v. We claim that w is coherent in $U(A)$. Let σ be a strategy implementing v. The strategy σ implements $w \subset v$ as well. Besides, the set $\mathbf{cone}(w) \subset P_A$ is finite and contains the empty play ϵ. It follows that $\sigma \cap \mathbf{cone}(w)$ is finite and nonempty; and that there exists a $\sqsubseteq_A$-maximal play s in $\sigma \cap \mathbf{cone}(w)$. We claim that $s \in \mathbf{frontier}(w)$. Suppose not.

Then, there would exist a cell $m \in M_A$ such that $s \cdot m \in \mathbf{cone}(w)$; by definition of $\sigma \vDash_A w$, there would exist a value n such that $s \cdot m \cdot n \in \sigma$ and $s \cdot m \cdot n \in \mathbf{cone}(w)$; and this would contradict maximality of s in $\sigma \cap \mathbf{cone}(w)$. We conclude that $s \in \mathbf{frontier}(w)$. Now, as an element of σ, the play s is of even-length. And the eds A is regular. It follows that every play $\mathbf{frontier}(w)$ is of even-length. We conclude that w is coherent, and that v is a clique in $U(A)$.
$(2 \Rightarrow 1)$ is by finiteness. We write $s \xrightarrow{m,n}_v t$ when
- $s, t \in P_A^{\mathbf{even}}$ and $m, n \in M_A$ and $t = s \cdot m \cdot n$,
- $\forall x \in v, s \cdot m \preceq_A x \;\Rightarrow\; s \cdot m \cdot n \preceq_A x$.

The relation $\longrightarrow_v$ defines a tree T_v on the even-length plays of A, labelled with pairs of moves (m, n). Let σ be maximal among the subtrees of T_v closed under even-length prefix, and verifying

$$\forall m, n_1, n_2 \in M_A, \forall s, t_1, t_2 \in \sigma, \qquad s \xrightarrow{m,n_1}_v t_1 \text{ and } s \xrightarrow{m,n_2}_v t_2 \Rightarrow n_1 = n_2 \quad (10)$$

Clearly, σ is a strategy of A. We claim that this strategy σ implements v. Indeed, suppose that $x \in v$, that $s \in \sigma$, that $m \in M_A$, and that $s \cdot m \preceq_A x$. We prove that

$$\exists n \in M_A, \qquad s \cdot m \cdot n \preceq_A x \text{ and } s \cdot m \cdot n \in \sigma.$$

Let $w = \{x \in v \mid s \cdot m \preceq_A x\}$. As a finite subset of the clique v, the set w is coherent in $U(A)$. By definition of coherence, this means that all the plays in $\mathbf{frontier}(w)$ are of even-length. On the other hand, $s \cdot m$ is element of $\mathbf{cone}(w)$ and of odd-length. Thus, $s \cdot m$ is strict prefix of a play $t \in \mathbf{frontier}(w)$. Let $p \in M_A$ be the value such that $s \cdot m \cdot p \sqsubseteq t$. Note that $s \cdot m \cdot p \in w$, and thus $s \xrightarrow{m,p}_v s \cdot m \cdot p$. So, by maximality of σ, there exists a move $n \in M_A$ such that $s \xrightarrow{m,n}_v s \cdot m \cdot n$. By definition of $\xrightarrow{m,n}_v$ and $s \cdot m \preceq_A x$, the inequality $s \cdot m \cdot n \preceq_A x$ holds. We conclude.
■

The definition of U is nicely illustrated by the regular eds $!\mathbb{B} \otimes !\mathbb{B} \otimes !\mathbb{B}$ discussed in the introduction, and presented in figure 1. Consider the two subsets v, w of $E_{!\mathbb{B} \otimes !\mathbb{B} \otimes !\mathbb{B}}$:

$$w = \{(\bot, V, F), (F, \bot, V), (V, F, \bot)\}, \qquad v = \{(F, \bot, V), (V, F, \bot)\}.$$

The frontier of v and w are given by singletons:

$$\mathbf{frontier}(w) = \{\epsilon\}, \qquad \mathbf{frontier}(v) = \{*_1\}.$$

The empty play ϵ is of even-length and the play $*_1$ is of odd-length. It follows that w is coherent and that v is incoherent in the eds $!\mathbb{B} \otimes !\mathbb{B} \otimes !\mathbb{B}$.

8 The strongly stable vs. the sequential algorithm hierarchies

In this section, we prove

(1) that the extensional data structures $[T]^{\mathbf{flat}}_{\mathrm{SDS}}$ and $[T]^{\mathbf{lazy}}_{\mathrm{SDS}}$ are regular for every simple type T, and

(2) that the hypercoherence space $[T]_{\mathrm{HC}}$ interpreting T in the strongly stable model, is precisely the hypercoherence space extracted from the edss $[T]^{\mathbf{flat}}_{\mathrm{SDS}}$ and $[T]^{\mathbf{lazy}}_{\mathrm{SDS}}$:

$$[T]_{\mathrm{HC}} \;=\; U([T]^{\mathbf{flat}}_{\mathrm{SDS}}) \;=\; U([T]^{\mathbf{lazy}}_{\mathrm{SDS}}).$$

This is the reconstruction theorem 8.3 established in Section 8.3. The theorem is proved by induction on the type T, after the lemmas of Sections 8.1 and 8.2.

8.1 The linear implication $\multimap$

Lemma 8.1 *Suppose that A, B are regular edss, and that B is spread. Then, the eds $A \multimap B$ is regular, and verifies the equality:*

$$U(A \multimap B) = U(A) \multimap U(B).$$

PROOF The two hypergraphs $U(A)$ and $U(B)$ are hypercoherence spaces. The web $U(A) \multimap U(B)$ is therefore equal to the cartesian product of the webs of $U(A)$ and $U(B)$, that is: $E_A \times E_B$. It follows that the hypergraphs $U(A) \multimap U(B)$ and $U(A \multimap B)$ have the same web. Besides, we know that every atom of $U(A) \multimap U(B)$ and $U(A \multimap B)$ is of polarity $+1$. We prove now that the strict coherence of $U(A) \multimap U(B)$ and $U(A \multimap B)$ coincide.

Suppose that $v \subset^*_{\mathrm{fin}} E_{A\multimap B}$ is strictly coherent in the hypergraph $U(A) \multimap U(B)$. That means that v is not a singleton, and that both assertions hold:

$$v{\restriction}A \in \Gamma(U(A)) \Rightarrow v{\restriction}B \in \Gamma(U(B)), \tag{11}$$

$$v{\restriction}A \in \Gamma^*(U(A)) \Rightarrow v{\restriction}A \in \Gamma^*(U(B)). \tag{12}$$

We claim that v is coherent in the eds $A \multimap B$. Indeed, let $s \in P^{\mathbf{odd}}_{A\multimap B}$ be a play of odd-length in $\mathbf{cone}(v)$. We prove that there exists $m \in M_{A\multimap B}$ such that $s \cdot m \in \mathbf{cone}(v)$.

Note that the projection $s{\restriction}A$ is of even-length and in $\mathbf{cone}(v{\restriction}A)$; and that the projection $s{\restriction}B$ is of odd-length and in $\mathbf{cone}(v{\restriction}B)$.

We proceed by case analysis. First case: when $s{\restriction}A \notin \mathbf{frontier}(v{\restriction}A)$. Then, there exists a cell $m \in M_A$ such that $(s{\restriction}A)\cdot m \in \mathbf{cone}(v{\restriction}A)$. It follows that $s \cdot \mathtt{inl}(m) \in \mathbf{cone}(v)$, and we conclude.

Second case: when $s\restriction A \in \mathbf{frontier}(v\restriction A)$. Then, it follows from regularity that $v\restriction A$ is coherent in the eds A. From this, and (11), it follows that $v\restriction B$ is coherent in the eds B. It is worth noting here that $v\restriction B$ is not singleton, because, otherwise, $v\restriction A$ would be singleton by (12) and thus v would be singleton — which contradicts our hypothesis.
So, the set $v\restriction B$ is coherent. And the play $s\restriction B$ is of odd-length and in $\mathbf{cone}(v\restriction B)$. It follows that there exists a value $m \in M_B$ such that $(s\restriction B) \cdot m \in \mathbf{cone}(v\restriction B)$. The set $v\restriction B$ is also non singleton. It follows that $\mathbf{cone}(v\restriction B)$ does not contain any $\sqsubseteq$-maximal play in the eds B. So, the play $(s\restriction B) \cdot m$ is not maximal. From this, it follows easily that $s \cdot \mathtt{inr}(m) \in \mathbf{cone}(v)$.
We have just proved that every play $s \in P^{\mathbf{odd}}_{A \multimap B}$ of odd-length in $\mathbf{cone}(v)$ may be extended by a value $m \in M_{A \multimap B}$ such that $s \cdot m \in \mathbf{cone}(v)$. We conclude that v is coherent in the eds $A \multimap B$.
Now, suppose that $v \subset^*_{\mathrm{fin}} E_{A \multimap B}$ is strictly incoherent in the hypergraph $U(A) \multimap U(B)$. That means that v is not a singleton, and that:

$$v\restriction A \in \Gamma(U(A)) \text{ and } v\restriction B \notin \Gamma^*(U(B)). \tag{13}$$

We claim that v is incoherent in the eds $A \multimap B$. Indeed, let $s \in P^{\mathbf{even}}_{A \multimap B}$ be a play of even-length in $\mathbf{cone}(v)$. We prove that there exists a cell $m \in M_{A \multimap B}$ such that $s \cdot m \in \mathbf{cone}(v)$.
We proceed by case analysis. First case: when the last move of s is played in the component A. Then, the projection $s\restriction A$ is of odd-length and in $\mathbf{cone}(v\restriction A)$. We know from (13) that $v\restriction A$ is coherent in the eds A. It follows that there exists a value $m \in M_A$ such that $(s\restriction A) \cdot m \in \mathbf{cone}(v\restriction A)$. It follows that $s \cdot \mathtt{inl}(m) \in \mathbf{cone}(v)$, and we conclude.
Second case: when the last move of s is played in the component B, or when s is the empty play. Then, the projection $s\restriction B$ is of even-length and in $\mathbf{cone}(v\restriction B)$. By (13) the set of extensions $v\restriction B$ is either (a) singleton $v\restriction B = \{y\}$, or (b) non singleton and incoherent in the eds B. We claim that in both cases (a) and (b) there exists a cell $m \in M_B$ such that $(s\restriction B) \cdot m \in \mathbf{cone}(v\restriction B)$. This is immediate in case (b) when v is incoherent in B. This is also true in case (a) because, we claim, the play $s\restriction B$ is not element of $\|y\|_B$. Indeed, if this was the case, then $s\restriction B$ would be $\sqsubseteq$-maximal in the eds B, because B is spread; and in turn, the play s would be $\sqsubseteq$-maximal in the eds $A \multimap B$; this maximality and $s \in \mathbf{cone}(v)$ would imply that v is singleton, which contradicts our hypothesis. We conclude that there exists a cell $m \in M_B$ such that $(s\restriction B) \cdot m \in \mathbf{cone}(v\restriction B)$. It follows easily that $s \cdot \mathtt{inr}(m) \in \mathbf{cone}(v)$.
We have just proved that every play $s \in P^{\mathbf{even}}_{A \multimap B}$ of even-length in $\mathbf{cone}(v)$ may be extended by a cell $m \in M_{A \multimap B}$ in such a way that $s \cdot m \in \mathbf{cone}(v)$. We conclude that v is incoherent in the eds $A \multimap B$.

Now, observe that every non-empty finite subset v of $E_{A\multimap B}$ is either coherent or incoherent in the hypergraph $U(A) \multimap U(B)$. We have just proved that the subset v is coherent in the eds $A \multimap B$ in the first case, and incoherent in the eds $A \multimap B$ in the second case. We conclude that $A \multimap B$ is regular, and that $U(A \multimap B) = U(A) \multimap U(B)$. ■

8.2 The exponentials

Lemma 8.2 (exponential) *Suppose that A is a regular eds. Then, the eds $!A$ is regular and verifies the equality:*

$$U(!A) = !U(A).$$

PROOF Suppose that A is regular. By lemma 7.4 the two webs of $U(!A)$ and $!U(A)$ are equal. We prove that the hypercoherence structure on $|U(!A)| = |!U(A)|$ are the same. Let $\{v_0, ..., v_{j-1}\}$ be a non-empty finite subset of $|U(!A)| = |!U(A)|$.
Suppose that $\{v_0, ..., v_{j-1}\}$ is coherent in the hypergraph $!U(A)$, or more explicitly that every section w of $\{v_0, ..., v_{j-1}\}$ is coherent in $U(A)$. We claim that $\{v_0, ..., v_{j-1}\}$ is coherent in the eds $!A$. Indeed, let s be an odd-length play of $!A$ verifying $s \in \mathbf{cone}(\{v_0, ..., v_{j-1}\})$, or more explicitly $\forall i \in [j], s \preceq_{!A} v_i$. By definition, the word s explores an augmented strategy σ of A with exactly one odd-length play $t \in P_A$. Note that the play $t \in P_A$ is at the same time the last move of s in the eds $!A$. Define v as the set of extensions $x \in \bigcup_{i\in[j]} v_i$ such that $t \preceq_A x$. It follows from $\forall i \in [j], s \preceq_{!A} v_i$ that the set v defines a finite section of $\{v_0, ..., v_{j-1}\}$. By hypothesis, the section v is coherent in A. This implies that the play $t \in \mathbf{cone}(v)$ may be extended with a value $m \in M_A$ into an even-length play $t \cdot m \in \mathbf{cone}(v)$. The play $t \cdot m$ is also a move of $!A$. The play $s \in P_{!A}$ extended with that move $(t \cdot m)$ defines a play $s \cdot (t \cdot m) \in P_{!A}$ which verifies

$$\forall i \in [j], s \cdot (t \cdot m) \preceq_{!A} v_i.$$

We conclude that $\{v_0, ..., v_{j-1}\}$ is coherent in the eds $!A$.
Suppose now that $\{v_0, ..., v_{j-1}\}$ is not coherent in the hypergraph $!U(A)$, or more explicitly that there exists an incoherent section w of $\{v_0, ..., v_{j-1}\}$ in the hypergraph $U(A)$. We claim that $\{v_0, ..., v_{j-1}\}$ is incoherent in the eds $!A$. Indeed, let s be an even-length play of $!A$ such that $s \preceq_{!A} \{v_0, ..., v_{j-1}\}$, or more explicitly $\forall i \in [j], s \preceq_{!A} v_i$. By definition, the word s explores a strategy σ of A. Let $t \in P_A$ be maximal (wrt. $\sqsubseteq$) among the plays verifying $t \preceq_A w$ in the prefix-closed set of plays $\sigma \downarrow = \{t \in P_A \mid \exists t' \in \sigma, t \sqsubseteq_A t'\}$. We deduce from $s \preceq_{!A} \{v_0, ..., v_{k-1}\}$ that t is of even-length. By hypothesis, w is incoherent in the eds A. Opponent may

therefore extend the play t into $t \cdot m$ in such a way that $t \cdot m \preceq_A w$, or more explicitly that $\forall x \in w, t \cdot m \preceq_A x$. The definition of w as a section of $\{v_0, ..., v_{j-1}\}$ implies that $\forall i \in [j], \exists x \in v_i, t \cdot m \preceq_A x$. The word $s \cdot (t \cdot m)$ is a play of $!A$ and verifies $\forall i \in [j], s \cdot (t \cdot m) \preceq_{!A} v_j$. We conclude that $\{v_0, ..., v_{j-1}\}$ is incoherent in the eds $!A$.
Observe that every non-empty finite subset v of $E_{!A}$ is either coherent or incoherent in the hypergraph $!U(A)$. By the previous arguments, the subset v is coherent in $!A$ in the first case, and incoherent in $!A$ in the second case. We conclude that $!A$ is regular, and that $U(!A) = !U(A)$. ■

8.3 Reconstruction theorem

Theorem 8.3 (reconstruction) *Every simple type T is interpreted as a spread regular eds $[T]^{\mathbf{flat}}_{\mathrm{SDS}}$ or $[T]^{\mathbf{lazy}}_{\mathrm{SDS}}$, with associated hypergraph $U[T]^{\mathbf{flat}}_{\mathrm{SDS}} = U[T]^{\mathbf{lazy}}_{\mathrm{SDS}}$ the interpretation $[T]_{\mathrm{HC}}$ of T in the hypercoherence space model. Thus:*

$$[-]_{\mathrm{HC}} \;=\; U \circ [-]^{\mathbf{flat}}_{\mathrm{SDS}} \;=\; U \circ [-]^{\mathbf{lazy}}_{\mathrm{SDS}}. \tag{14}$$

PROOF By induction on the simple type T. The regularity property as well as the equality (14) are verified at the simple types ι and o, and it follows from lemmas 8.1 and 8.2 that they are preserved by the arrow construction $T_1 \Rightarrow T_2 = (!T_1) \multimap T_2$. We conclude. ■

9 Intermezzo: a retraction between the flat and the lazy hierarchies

In this section, we prepare our alternative proof of Ehrhard's theorem in Section 12. We show that the flat and the lazy sequential algorithm hierarchies (introduced at the end of Section 4) collapse to the same extensional hierarchy of types. The proof is based on a back-and-forth translation technique introduced in [Melliès 2004a]. The key step is to exhibit a retraction in the category EDS (or equivalently SDS) between the flat and the lazy natural numbers edss $\mathbb{N}_{\mathbf{flat}}$ and $\mathbb{N}_{\mathbf{lazy}}$:

$$\mathbb{N}_{\mathbf{flat}} \xrightarrow{\mathbf{for}} \mathbb{N}_{\mathbf{lazy}} \xrightarrow{\mathbf{count}} \mathbb{N}_{\mathbf{flat}} \;=\; \mathbb{N}_{\mathbf{flat}} \xrightarrow{\mathbf{id}_{\mathbb{N}_{\mathbf{flat}}}} \mathbb{N}_{\mathbf{flat}}\,. \tag{15}$$

The strategies **for** and **count** are defined as follows:

$$\mathbf{for} = \{s \in P^{\mathbf{even}}_{\mathbb{N}_{\mathbf{flat}} \multimap \mathbb{N}_{\mathbf{lazy}}} \mid \exists n \in \mathbb{N},\; s \sqsubseteq s_n\}$$

$$\mathbf{count} = \{s \in P^{\mathbf{even}}_{\mathbb{N}_{\mathbf{lazy}} \multimap \mathbb{N}_{\mathbf{flat}}} \mid \exists n \in \mathbb{N},\; s \sqsubseteq t_n\}$$

where s_n is the play of $\mathbb{N}_{\mathbf{flat}} \multimap \mathbb{N}_{\mathbf{lazy}}$ defined as:

$$s_n = \geq_{\mathtt{0}} \cdot * \cdot \mathtt{n} \cdot >_{\mathtt{0}} \cdot \geq_{\mathtt{1}} \cdots \geq_{\mathtt{n}} \cdot =_{\mathtt{n}}$$

and t_n is the play of $\mathbb{N}_{\mathbf{lazy}} \multimap \mathbb{N}_{\mathbf{flat}}$ defined as:

$$t_n = * \cdot \geq_{\mathtt{0}} \cdot >_{\mathtt{0}} \cdot \geq_{\mathtt{1}} \cdots \geq_{\mathtt{n}} \cdot =_{\mathtt{n}} \cdot \mathtt{n}.$$

The retraction (15) induces a retraction between the edss $[T]^{\mathbf{flat}}_{\mathrm{SDS}}$ and $[T]^{\mathbf{lazy}}_{\mathrm{SDS}}$ in the category EDS, for every simple type T:

$$[T]^{\mathbf{flat}}_{\mathrm{SDS}} \xrightarrow{[T]^{\mathbf{flat}}_{\mathbf{lazy}}} [T]^{\mathbf{lazy}}_{\mathrm{SDS}} \xrightarrow{[T]^{\mathbf{lazy}}_{\mathbf{flat}}} [T]^{\mathbf{flat}}_{\mathrm{SDS}} \quad = \quad [T]^{\mathbf{flat}}_{\mathrm{SDS}} \xrightarrow{\mathbf{id}_{[T]^{\mathbf{flat}}_{\mathrm{SDS}}}} [T]^{\mathbf{flat}}_{\mathrm{SDS}} \qquad (16)$$

The partial equivalence relations $\sim^{\mathbf{flat}}_T$ and $\sim^{\mathbf{lazy}}_T$ defined by extensional collapse (see Section 2.6) on the sets of strategies of $[T]^{\mathbf{flat}}_{\mathrm{SDS}}$ and $[T]^{\mathbf{lazy}}_{\mathrm{SDS}}$ are given below:

Definition 9.1 ($\sim^{\mathbf{flat}}$ and $\sim^{\mathbf{lazy}}$)

$$\sigma \sim^{\mathbf{flat}}_o \tau \quad \overset{defn}{\Longleftrightarrow} \quad \sigma \sim^{\mathbf{lazy}}_o \tau \quad \overset{defn}{\Longleftrightarrow} \quad \exists x \in \{V, F\},\ \sigma \ \vDash_{\mathbb{B}} x \textit{ and } \tau \ \vDash_{\mathbb{B}} x,$$

$$\sigma \sim^{\mathbf{flat}}_\iota \tau \quad \overset{defn}{\Longleftrightarrow} \quad \exists n \in E_{\mathbb{N}_{\mathbf{flat}}},\ \sigma \ \vDash_{\mathbb{N}_{\mathbf{flat}}} n \textit{ and } \tau \ \vDash_{\mathbb{N}_{\mathbf{flat}}} n,$$
$$\sigma \sim^{\mathbf{lazy}}_\iota \tau \quad \overset{defn}{\Longleftrightarrow} \quad \exists n \in E_{\mathbb{N}_{\mathbf{lazy}}},\ \sigma \ \vDash_{\mathbb{N}_{\mathbf{lazy}}} n \textit{ and } \tau \ \vDash_{\mathbb{N}_{\mathbf{lazy}}} n.$$

We establish now that the retraction morphisms (15) behave well towards the partial equivalence relations $\sim^{\mathbf{flat}}_\iota$ and $\sim^{\mathbf{lazy}}_\iota$.

Lemma 9.2 (preservation) *Suppose that σ and τ are strategies of $\mathbb{N}_{\mathbf{flat}}$. Then:*

$$\sigma \sim^{\mathbf{flat}}_\iota \tau \Rightarrow \sigma; \mathbf{for} \sim^{\mathbf{lazy}}_\iota \tau; \mathbf{for}.$$

Suppose that σ and τ are strategies of $\mathbb{N}_{\mathbf{lazy}}$. Then:

$$\sigma \sim^{\mathbf{lazy}}_\iota \tau \Rightarrow \sigma; \mathbf{count} \sim^{\mathbf{flat}}_\iota \tau; \mathbf{count} \qquad \textit{and} \qquad \sigma \sim^{\mathbf{lazy}}_\iota \sigma; \mathbf{count}; \mathbf{for}.$$

PROOF We prove the first statement. The two remaining statements are proved in a similar fashion. Suppose that $\sigma : \mathbb{N}_{\mathbf{flat}}$ and $\tau : \mathbb{N}_{\mathbf{flat}}$ are strategies and that $\sigma \sim^{\mathbf{flat}}_\iota \tau$. By definition, there exists an extension $n \in E_{\mathbb{N}_{\mathbf{flat}}}$ such that $\sigma \vDash_{\mathbb{N}_{\mathbf{flat}}} n$ and $\tau \vDash_{\mathbb{N}_{\mathbf{flat}}} n$. This implies that $\sigma = \tau$ is the strategy $\{\epsilon, * \cdot \mathtt{n}\}$. The strategies $(\sigma; \mathbf{for})$ and $(\tau; \mathbf{for})$ are equal to the strategy $\mu : \mathbb{N}_{\mathbf{lazy}}$ which contains exactly the even-length prefixes of the play:

$$\geq_{\mathtt{0}} \cdot >_{\mathtt{0}} \cdot \geq_{\mathtt{1}} \cdots \geq_{\mathtt{n}} \cdot =_{\mathtt{n}}$$

This strategy μ is the (unique) strategy of $\mathbb{N}_{\mathbf{lazy}}$ which implements $n \in E_{\mathbb{N}_{\mathbf{lazy}}}$. We conclude that $\sigma; \mathbf{for} \sim_{\iota}^{\mathbf{lazy}} \tau; \mathbf{for}$. ∎

By Lemma 9.2, he family of retractions (16) defines a *back-and-forth translation* between the hierarchies $[-]_{\mathrm{SDS}}^{\mathbf{flat}}$ and $[-]_{\mathrm{SDS}}^{\mathbf{lazy}}$ in the sense of [Melliès 2004a]. The existence of such a back-and-forth translation implies immediately that:

Lemma 9.3 *The two hierarchies $[-]_{\mathrm{SDS}}^{\mathbf{flat}}$ and $[-]_{\mathrm{SDS}}^{\mathbf{lazy}}$ collapse to the same extensional hierarchy.*

Remark. There remains to show that this extensional hierarchy is precisely the strongly stable hierarchy of [Bucciarelli, Ehrhard 1991]. This is precisely what we do from now on, in Sections 10, 11 and 12.

10 Compactness

In this section, we analyze the lazy sequential algorithm hierarchy of types, introduced in Section 5 and recalled in Section 9. In Section 10.1, we show that in this hierarchy, every simple type T is interpreted as a *finitely branching* eds. This departs from the flat hierarchy, where the base type ι is interpreted as the eds $\mathbb{N}_{\mathbf{flat}}$, which is not finitely branching. In Section 10.2, we use a non-constructive compactness argument to extend the characterization lemma 7.4 to possibly *infinite* configurations and cliques — at least when the underlying eds A is regular and finitely branching. We conclude in Section 10.3 that the configurations of the eds $[T]_{\mathrm{SDS}}^{\mathbf{lazy}}$ are the cliques of the hypercoherence space $[T]_{\mathrm{HC}}$.

10.1 The lazy hierarchy $[-]_{\mathrm{SDS}}^{\mathbf{lazy}}$ defines only finitely branching edss

Definition 10.1 (finitely branching) *An eds A is finitely branching when for every play $s \in P_A$, there exists only a finite number of moves $m \in M_A$ such that $s \cdot m \in P_A$.*

Lemma 10.2 *Every simple type T is interpreted as a finitely branching eds $[T]_{\mathrm{SDS}}^{\mathbf{lazy}}$ in the lazy hierarchy.*

PROOF The edss $\mathbb{B}$ and $\mathbb{N}_{\mathbf{lazy}}$ are finitely branching, and the class of finitely branching edss is closed under linear implication $\multimap$ and exponential modality $!(-)$. ∎

10.2 Configurations coincide with cliques (the infinite case)

We extend lemma 7.4 on possibly *infinite* configurations and cliques when the extensional data structure A is finitely branching.

Lemma 10.3 (configuration=clique (infinite case)) *Suppose that A is a regular finitely branching eds and that f is a (possibly infinite) subset of E_A. Then, the following are equivalent:*
(1) f is a configuration of A,
(2) f is a clique of $U(A)$.
PROOF $(1 \Rightarrow 2)$ is established as in lemma 7.4. $(2 \Rightarrow 1)$ is proved by a non-constructive compactness argument. We proceed as in the proof of lemma 7.4, and write $s \xrightarrow{m,n}_f t$ when
- $s, t \in P_A^{\mathbf{even}}$ and $m, n \in M_A$ and $t = s \cdot m \cdot n$,
- $\forall x \in f, s \cdot m \preceq_A x \;\Rightarrow\; s \cdot m \cdot n \preceq_A x$.

The relation $\longrightarrow_f$ defines a tree T_f on the even-length plays of A, labelled with pairs of moves (m, n). Let σ be maximal among the subtrees of T_f closed under even-length prefix, and verifying

$$\forall m, n_1, n_2 \in M_A, \forall s, t_1, t_2 \in \sigma, \qquad s \xrightarrow{m,n_1}_f t_1 \text{ and } s \xrightarrow{m,n_2}_f t_2 \Rightarrow n_1 = n_2 \quad (17)$$

The tree σ defines a strategy of A which, we claim, implements f. Indeed, suppose that $x \in f$, that $s \in \sigma$, that $m \in M_A$, and that $s \cdot m \preceq_A x$. We prove that

$$\exists n \in M_A, \qquad s \cdot m \cdot n \preceq_A x \text{ and } s \cdot m \cdot n \in \sigma$$

Let $g = \{x \in f \mid s \cdot m \preceq_A x\}$, and let $W = \mathcal{P}^*_{\text{fin}}(g)$ be the set of nonempty finite subsets of g. Let w be an element of W. As a finite subset of the clique f, the set w is coherent in $U(A)$. By definition of coherence, this means that all the plays in $\mathbf{frontier}(w)$ are of even-length. On the other hand, $s \cdot m$ is element of $\mathbf{cone}(w)$ and of odd-length. Consequently, the set $P(w) \subset M_A$ of values p such that $s \cdot m \cdot p \in \mathbf{cone}(w)$ is nonempty. Besides, and here comes compactness, the set $P(w)$ is *finite* because the eds A is finitely branching. It follows that the intersection

$$P = \bigcap_{w \in W} P(w)$$

is nonempty. Since every $p \in P$ verifies $s \xrightarrow{m,p}_f s \cdot m \cdot p$, we conclude by maximality of σ that there exists a move $n \in M_A$ such that $s \xrightarrow{m,n}_f s \cdot m \cdot n$. By definition of $\xrightarrow{m,n}_f$ and of $s \cdot m \preceq_A x$, the inequality $s \cdot m \cdot n \preceq_A x$ holds. We conclude. ■

10.3 The configurations are the strongly stable functions

It follows directly from theorem 8.3 and lemma 10.3 that
Corollary 10.4 *Suppose that T is a simple type, interpreted as $[T]^{\mathbf{lazy}}_{\mathrm{SDS}}$ in the lazy sequential algorithm model, and as $[T]_{\mathrm{HC}}$ in the strongly stable model. Then:*

(1) $[T]_{\mathrm{HC}} = U([T]^{\mathbf{lazy}}_{\mathrm{SDS}})$,
(2) the configurations of $[T]^{\mathbf{lazy}}_{\mathrm{SDS}}$ *are the cliques* $[T]_{\mathrm{HC}}$.

11 Collapse data structures

We carry on our analysis of extensionality in sequential games, and equip every extensional data structure A with a set $P_A^{\mathbf{alive}}$ of *alive* plays. The notion of alive play is mainly motivated by the description of the partial equivalence relation generated by extensional collapse on the hierarchy $[-]^{\mathbf{lazy}}_{\mathrm{SDS}}$ — see the anatomic theorem 12.6 in Section 12. For that reason, we call *collapse data structure* (cods) an extensional data structure equipped with a notion of alive play.

11.1 Collapse data structures

Definition 11.1 (cods) *A collapse data structure (cods) is an extensional data structure A equipped with a set $P_A^{\mathbf{alive}} \subset P_A^{\mathbf{even}}$ of even-length plays of A.*
A play $s \in P_A$ is called *alive* when $s \in P_A^{\mathbf{alive}}$.

11.2 Extensional and sub-extensional strategies

We associate to every strategy σ in a cods A a set of extensions $U(\sigma)$ defined as follows:
Definition 11.2 *$U(\sigma)$ denotes the set of extensions $x \in E_A$ "encountered" by the strategy σ, that is:*

$$U(\sigma) = \{x \in E_A \mid \sigma \cap \|x\|_A \neq \emptyset\}.$$

Now, we define a notion of *extensional* and *sub-extensional* strategy in a collapse data structure A. We recall that the notion sub-implementation is introduced in Definition 5.5.
Definition 11.3 (extensional strategy) *A strategy σ is extensional when*
- $\sigma \subset P_A^{\mathbf{alive}}$,
- *σ implements every extension $x \in U(\sigma)$:*

$$\forall x \in E_A, \qquad x \in U(\sigma) \Rightarrow \sigma \vDash_A x.$$

Definition 11.4 (sub-extensional strategy) *A strategy σ is sub-extensional when*
- *$U(\sigma)$ is a configuration of A,*
- $\sigma \subset P_A^{\mathbf{alive}}$,
- *σ sub-implements every extension $x \in U(\sigma)$.*

One proves easily that

Lemma 11.5 *Every substrategy of an extensional strategy is sub-extensional.*
Remark. We will see next section, theorem 12.6, that the extensional strategies of $[T]^{\mathbf{lazy}}_{\mathrm{SDS}}$ are precisely the self-equivalent strategies of the extensional collapse $\sim^{\mathbf{lazy}}_{T}$.

11.3 The hierarchy $[-]^{\mathbf{lazy}}_{\mathrm{CODS}}$ of simple types

The hierarchy $[-]^{\mathbf{lazy}}_{\mathrm{CODS}}$ is just the hierarchy $[-]^{\mathbf{lazy}}_{\mathrm{SDS}}$ in which every eds $[T]^{\mathbf{lazy}}_{\mathrm{SDS}}$ is equipped with the adequate notion of alive play. The base types o and ι are interpreted by the edss $\mathbb{B}$ and $\mathbb{N}_{\mathbf{lazy}}$ in which every even-length play is seen as alive:

$$P^{\mathbf{alive}}_{\mathbb{B}} = P^{\mathbf{even}}_{\mathbb{B}} \qquad\qquad P^{\mathbf{alive}}_{\mathbb{N}_{\mathbf{lazy}}} = P^{\mathbf{even}}_{\mathbb{N}_{\mathbf{lazy}}}$$

The type $T = T_1 \Rightarrow T_2$ is interpreted by Girard formula:

$$[T]^{\mathbf{lazy}}_{\mathrm{CODS}} \;=\; (![T_1]^{\mathbf{lazy}}_{\mathrm{CODS}}) \multimap [T_2]^{\mathbf{lazy}}_{\mathrm{CODS}}$$

where the linear implication and exponentials are defined as follows:
The linear implication of two codss A and B is the eds $A \multimap B$ in which $P^{\mathbf{alive}}_{A\multimap B}$ in which a play $s \in P^{\mathbf{even}}_{A\multimap B}$ is alive precisely when:
- $s{\upharpoonright}A \in P^{\mathbf{alive}}_{A} \Rightarrow s{\upharpoonright}B \in P^{\mathbf{alive}}_{B}$,
- $(s{\upharpoonright}A \in P^{\mathbf{alive}}_{A}$ and $s{\upharpoonright}B \in R_B) \Rightarrow s{\upharpoonright}A \in R_A$.

The exponential of a cods A is the eds $!A$ in which a play $s \in P^{\mathbf{even}}_{!A}$ is alive precisely when it explores a *sub-extensional* strategy $\sigma : A$.

Remark. The definition of $P_{A\multimap B}$ is motivated by Theorem 12.6. Intuitively, a play is “alive” means that it may be “visited” by a self-equivalent strategy. So, the first condition tells that an alive play of A is transported to an alive play of B by an alive play of $A \multimap B$. The second condition tells that an alive play of A transported to a realizer of B by an alive play of $A \multimap B$, is itself a realizer of A.

11.4 Alive collapse data structures

We introduce a notion *alive* cods in which a converse to lemma 11.5 may be established (lemma 11.7).
Definition 11.6 (alive) *A cods is* alive *when:*
- $\forall x \in E_A,\ \|x\|_A \subset P^{\mathbf{alive}}_{A}$,
- $\forall s \in P_A, \forall t \in P_A, \quad s \sqsubseteq^{\mathrm{even}}_{A} t$ *and* $t \in P^{\mathbf{alive}}_{A} \;\Rightarrow\; s \in P^{\mathbf{alive}}_{A}$

Lemma 11.7 *Suppose that a cods A is alive and regular, and that σ is a sub-extensional strategy of A. Then, σ is the substrategy of an extensional strategy τ verifying $U(\sigma) = U(\tau)$.*

PROOF By regularity, the finite configuration $U(\sigma)$ is also a clique of the hypercoherence space $U(!(A))$. The proof then proceeds as lemma 7.4 $(2 \Rightarrow 1)$. It differs only in that the maximal strategy σ considered in (10) is required to contain the strategy τ. ■

We observe moreover that:

Lemma 11.8 *Every interpretation* $[T]^{\mathbf{lazy}}_{\mathrm{CODS}}$ *is alive.*

PROOF By induction on the simple type T. The proof is based on the two observations below:

- a cods $A \multimap B$ is alive when the codss A and B are alive, and B is spread,
- a cods $!A$ is alive when the cods A is alive.

■

11.5 Compositionality of extensional strategies

In this section, we relate the composition laws of extensional strategies in cods and of cliques in hypercoherence spaces.

Lemma 11.9 (compositionality) *Suppose that T_1 and T_2 are simple types. Suppose that τ is an extensional strategy of $[T_1]^{\mathbf{lazy}}_{\mathrm{CODS}}$ and that σ is an extensional strategy of $[T_1 \Rightarrow T_2]^{\mathbf{lazy}}_{\mathrm{CODS}}$. Then, the strategy $(\sigma \cdot_{T_1T_2} \tau)$ is extensional in the cods $[T_1]^{\mathbf{lazy}}_{\mathrm{CODS}}$, and:*

$$U(\sigma \cdot_{T_1T_2} \tau) = U(\sigma) \cdot_{T_1T_2} U(\tau)$$

where the strategy $\sigma \cdot_{T_1T_2} \tau$ is defined in the lazy hierarchy $[-]^{\mathbf{lazy}}_{\mathrm{CODS}}$ and the configuration $U(\sigma) \cdot_{T_1T_2} U(\tau)$ is defined in the strongly stable hierarchy $[-]_{\mathrm{HC}}$.

PROOF We write $A = [T_1]^{\mathbf{lazy}}_{\mathrm{CODS}}$ and $B = [T_2]^{\mathbf{lazy}}_{\mathrm{CODS}}$.

We prove first that $(\sigma \cdot_{T_1T_2} \tau) \subset P^{\mathbf{alive}}_B$. Suppose that $t \in \sigma \cdot_{T_1T_2} \tau$. By definition of composition, there exists a play $s \in \sigma$ such that:

- $s \restriction !A$ is a play in the strategy $(\tau)^\dagger$,
- $s \restriction B = t$.

The play $s \restriction !A$ explores a substrategy μ of the extensional strategy $\tau \subset P^{\mathbf{alive}}_A$. By lemma 11.5, the strategy μ is sub-extensional. It follows that $s \restriction !A$ is alive. We conclude from the definition of $P^{\mathbf{alive}}_{(!A) \multimap B}$ that $t = s \restriction B \in P^{\mathbf{alive}}_B$. We conclude that $(\sigma \cdot_{T_1T_2} \tau) \subset P^{\mathbf{alive}}_B$.

Now, we claim that every time the strategy τ implements a configuration $v \subset^*_{\mathrm{fin}} E_A$ and the strategy σ implements an extension $(v, y) \in E_{(!A) \multimap B}$, then the strategy $(\sigma \cdot_{T_1T_2} \tau)$ implements the extension $y \in E_A$. The proof (not difficult, but lengthy) is not detailed here.

We prove now the inclusion $U(\sigma \cdot_{T_1T_2} \tau) \supset U(\sigma) \cdot_{T_1T_2} U(\tau)$. Suppose that $y \in U(\sigma) \cdot_{T_1T_2} U(\tau)$. By definition of relational composition, this means that there ex-

ists $v \subset E_A$ such that $v \subset U(\tau)$ and $(v, y) \in U(\sigma)$. By extensionality of σ and τ, this means that σ implements the extension (v, y) and that τ implements the configuration v. We conclude that the strategy $(\sigma \cdot_{T_1 T_2} \tau)$ implements the extension $y \in E_B$. It follows from finiteness of $\|y\|_B$ that $y \in U(\sigma \cdot_{T_1 T_2} \tau)$. We conclude.
We prove now the converse inclusion $U(\sigma \cdot_{T_1 T_2} \tau) \subset U(\sigma) \cdot_{T_1 T_2} U(\tau)$. Suppose that $y \in U(\sigma \cdot_{T_1 T_2} \tau)$. By definition of game-theoretic composition, this implies that there exists a play $s \in \sigma$ such that:

- $s \upharpoonright !A$ is a play of the strategy $(\tau)^\dagger$,
- $s \upharpoonright B$ is a play of $\|y\|_B$.

The play s is alive in the cods $!A \multimap B$, as well as its projection $s \upharpoonright !A$. By definition of $P^{\mathbf{alive}}_{(!A) \multimap B}$, $s \upharpoonright B \in R_B$ implies that $s \upharpoonright !A \in R_A$. So, there exists a finite configuration $v \subset_{\text{fin}} E_A$ such that $s \upharpoonright !A \in \|v\|_{!A}$. The definition of $s \upharpoonright !A \in \|v\|_{!A}$ indicates that there exists a strategy μ of A such that:

- $s \upharpoonright !A$ explores the strategy μ,
- μ is sub-extensional,
- $U(\mu) = v$,
- $\forall t \in \mu, \exists x \in v, t \preceq_A x$.

Now, it follows from $s \upharpoonright !A \in (\tau)^\dagger$ that $\mu \subset \tau$, and thus, that $v \subset U(\tau)$. Note also that $(v, y) \in U(\sigma)$. We conclude that $\tau \vDash_A v$ and $\sigma \vDash_{(!A) \multimap B} (v, y)$, and thus, that $U(\sigma \cdot_{T_1 T_2} \tau) \subset U(\sigma) \cdot_{T_1 T_2} U(\tau)$.
We have just proved that

- $\sigma \cdot_{T_1 T_2} \tau \subset P^{\mathbf{alive}}_B$,
- any extension $y \in U(\sigma) \cdot_{T_1 T_2} U(\tau)$ is implemented by $(\sigma \cdot_{T_1 T_2} \tau)$,
- $U(\sigma \cdot_{T_1 T_2} \tau) = U(\sigma) \cdot_{T_1 T_2} U(\tau)$.

We conclude that the strategy $(\sigma \cdot_{T_1 T_2} \tau)$ implements every extension of $U(\sigma \cdot_{T_1 T_2} \tau)$, and that the strategy $(\sigma \cdot_{T_1 T_2} \tau)$ is thus extensional. ■

Remark. It follows from corollary 10.4 that the partial equivalence classes of $\approx^{\mathbf{lazy}}_T$ are in one-to-one relationship with the *configurations* of $[T]^{\mathbf{lazy}}_{\mathrm{CODS}}$ and with the *cliques* of $[T]_{\mathrm{HC}}$. Compositionality (lemma 11.9) ensures that extensional strategies modulo $\approx^{\mathbf{lazy}}_T$ compose as configurations in $[T]_{\mathrm{HC}}$. From this, one concludes that the hierarchy $[-]^{\mathbf{lazy}}_{\mathrm{CODS}}$ quotiented by $\approx^{\mathbf{lazy}}$ coincides with the hierarchy $[-]_{\mathrm{HC}}$.

12 An anatomy of Ehrhard's collapse theorem

In this section, we *characterize* the partial equivalence relation $\sim^{\mathbf{lazy}}$ induced by extensional collapse on the lazy hierarchy $[-]^{\mathbf{lazy}}_{\mathrm{SDS}}$ as the partial equivalence relation $\approx^{\mathbf{lazy}}_T$ below:

Definition 12.1 ($\approx_T^{\mathbf{lazy}}$) *Two strategies σ and τ of the collapse data structure $[T]^{\mathbf{lazy}}_{\mathrm{CODS}}$ verify $\sigma \approx_T^{\mathbf{lazy}} \tau$ precisely when:*
- *σ and τ are extensional,*
- *$U(\sigma) = U(\tau)$.*

12.1 Preliminaries

Before starting off the proof of theorem 12.6, we give two useful definitions and establish two easy lemmas.

Definition 12.2 (big cone) *Suppose that $v \subset E_A$ is a non-empty subset of extensions of a cods A. We write:*

$$\mathbf{bigcone}(v) = \bigcup_{x \in v} \{s \in P_A \mid s \preceq_A x\}$$

Lemma 12.3 *Suppose that A is a spread cods, that σ is an extensional strategy of A, and that $v \subset E_A$ is a non-empty set of extensions of A. Then, $\sigma \cap \mathbf{bigcone}(v)$ is an extensional strategy and $U(\sigma \cap \mathbf{bigcone}(v)) = U(\sigma) \cap v$.*

PROOF Obviously, $\sigma \cap \mathbf{bigcone}(v)$ is a strategy included in $P_A^{\mathbf{alive}}$ which implements every extension in $U(\sigma) \cap v$. It follows that $U(\sigma) \cap v \subset U(\sigma \cap \mathbf{bigcone}(v))$. Conversely, the cods A is spread, and thus, the set $\mathbf{bigcone}(v) \cap \|x\|_A$ is non-empty only for extensions $x \in v$. It follows that $U(\sigma \cap \mathbf{bigcone}(v)) \subset U(\sigma) \cap v$. We obtain that $U(\sigma \cap \mathbf{bigcone}(v)) = U(\sigma) \cap v$ and that every extension in $U(\sigma \cap \mathbf{bigcone}(v))$ is implemented by $\sigma \cap \mathbf{bigcone}(v)$. We conclude that $\sigma \cap \mathbf{bigcone}(v)$ is extensional. ■

Definition 12.4 (compatible) *Two strategies μ_1, μ_2 of a cods A are* compatible *when μ_1, μ_2 are substrategies of a strategy μ_3: that is, $\mu_1 \subset \mu_3$ and $\mu_2 \subset \mu_3$.*

Lemma 12.5 *Suppose that A, B are cods, and that μ_1, μ_2 are compatible strategies of A such that μ_1 is not included in μ_2. Suppose that $\sigma : (!A) \multimap B$ is a strategy and $s \in P_{(!A) \multimap B}$ a play verifying:*
(1) $s \in \sigma$,
(2) $s \upharpoonright !A$ explores the strategy μ_1.
Then, there exists a play $u \in P_B^{\mathbf{even}}$ and a cell $m \in M_B$ verifying:
(1) $u \cdot m \sqsubseteq_B s \upharpoonright B$,
(2) $u \in (\mu_2)^\dagger; \sigma$,
(3) $\forall n \in M_B, u \cdot m \cdot n \notin (\mu_2)^\dagger; \sigma$.

PROOF By hypothesis, there exists a play $t \in P_A^{\mathbf{even}}$ such that $t \in \mu_1$ and $t \notin \mu_2$. Every such play $t \in P_A$ is also a move $n \in M_{!A}$. Let n be the first such move appearing in the play $s \in P_{(!A) \multimap B}$. The play s factors as $s_1 \cdot \mathtt{inr}(m) \cdot s_2 \cdot n \cdot s_3$ where:

- the play $s_1 \in P_{(!A)\multimap B}$ is of even-length,
- the move $\mathtt{inr}(m)$ is a cell of the component B,
- the moves of $s_2 \cdot n$ are played in the component $!A$,

Let $u \in P_A^{\mathbf{even}}$ denote the projection $s_1 \upharpoonright B$. It follows from $s_1 \cdot \mathtt{inr}(m) \sqsubseteq_{(!A)\multimap B} s$ that $u \cdot m \sqsubseteq_B s \upharpoonright B$. Note that the play $s_1 \cdot \mathtt{inr}(m)$ is maximal among the plays t of σ prefix of s and such that $t \upharpoonright !A \in (\mu_2)^\dagger$. We conclude that $u \in (\mu_2)^\dagger; \sigma$ and that $\forall n \in M_B, u \cdot m \cdot n \notin (\mu_2)^\dagger; \sigma$. ∎

12.2 Anatomy of a collapse

We prove now our main theorem which states that, for every simple type T, and strategies σ and τ of the collapse data structure $[T]^{\mathbf{lazy}}_{\mathrm{CODS}}$:

Theorem 12.6 (anatomic) $\sigma \sim^{\mathbf{lazy}}_T \tau \iff \sigma \approx^{\mathbf{lazy}}_T \tau$

PROOF By induction on the type T. The property is obvious for the base types o and ι. Now, suppose that the property is established for the simple types T_1 and T_2. We prove that the property holds for $T = T_1 \Rightarrow T_2$. In order to simplify our notations, we write $A = [T_1]^{\mathbf{lazy}}_{\mathrm{CODS}}$ and $B = [T_2]^{\mathbf{lazy}}_{\mathrm{CODS}}$. Note that $[T]^{\mathbf{lazy}}_{\mathrm{CODS}} = (!A) \multimap B$.

($\Leftarrow$) is nearly immediate by lemma 11.9 (compositionality). Indeed, suppose that $\sigma \approx^{\mathbf{lazy}}_T \tau$ and consider two strategies μ and ν such that $\mu \sim^{\mathbf{lazy}}_{T_1} \nu$. The equivalence $\mu \approx^{\mathbf{lazy}}_{T_1} \nu$ holds by induction hypothesis on T_1. The equivalence

$$(\sigma \cdot_{T_1T_2} \mu) \approx^{\mathbf{lazy}}_{T_2} (\tau \cdot_{T_1T_2} \nu)$$

follows from lemma 11.9. We deduce from this and our induction hypothesis on T_2 that:

$$(\sigma \cdot_{T_1T_2} \mu) \sim^{\mathbf{lazy}}_{T_2} (\tau \cdot_{T_1T_2} \nu).$$

We conclude that:

$$\forall \mu, \nu, \qquad \mu \sim^{\mathbf{lazy}}_{T_1} \nu \Rightarrow (\sigma \cdot_{T_1T_2} \mu) \sim^{\mathbf{lazy}}_{T_2} (\tau \cdot_{T_1T_2} \nu)$$

and thus, that $\sigma \sim^{\mathbf{lazy}}_T \tau$.

($\Rightarrow$) We suppose that $\sigma \sim^{\mathbf{lazy}}_T \tau$ and deduce that $\sigma \approx^{\mathbf{lazy}}_T \tau$. We prove in **Part I** that the strategies σ and τ are extensional and in **Part II** that $U(\sigma) = U(\tau)$.

Part I: We show that $\sigma \sim^{\mathbf{lazy}}_T \sigma$ implies:

(★) that $\sigma \subset P^{\mathbf{alive}}_{(!A)\multimap B}$;

(★★) that two configurations $v \subset w$ are equal when $(v, y) \in U(\sigma)$ and $(w, y) \in U(\sigma)$ for some extension $y \in E_B$;

($\bigstar\bigstar\bigstar$) that $(v, y) \in U(\sigma)$ implies $\sigma \vDash_{(!A) \multimap B} (v, y)$ for any extension $(v, y) \in E_{(!A) \multimap B}$.

($\bigstar$) We proceed by contradiction. Suppose that that there exists a play $s \in \sigma$ such that $\neg(s \in P^{\mathbf{alive}}_{(!A) \multimap B})$. We start a case analysis:

(1) either $s \upharpoonright !A \in P^{\mathbf{alive}}_{!A}$ and $\neg(s \upharpoonright B \in P^{\mathbf{alive}}_{B})$, or

(2) $s \upharpoonright !A \in P^{\mathbf{alive}}_{!A}$ and $s \upharpoonright B \in R_B$ and $\neg(s \upharpoonright !A \in R_{!A})$.

In both cases, $s \upharpoonright !A \in P^{\mathbf{alive}}_{!A}$ means that $s \upharpoonright !A$ explores a sub-extensional strategy μ of the cods A. By definition of a sub-extensional strategy , $v = U(\sigma)$ is a configuration. By lemma 11.7, the strategy μ is included in an extensional strategy $\nu : A$ such that $U(\nu) = v$. Note also that the play $s \upharpoonright !A$ is element of the two comonoidal strategies $(\mu)^\dagger$ and $(\nu)^\dagger$ of the cods $!A$.

(Case 1) The strategy $\nu : A$ is extensional, and thus verifies $\nu \approx^{\mathbf{lazy}}_{T_1} \nu$ by definition of $\approx^{\mathbf{lazy}}$. From this and our induction hypothesis on T_1, it follows that $\nu \sim^{\mathbf{lazy}}_{T_1} \nu$. From $\sigma \sim^{\mathbf{lazy}}_{T} \sigma$, it follows that

$$(\sigma \cdot_{T_1 T_2} \nu) \sim^{\mathbf{lazy}}_{T_2} (\sigma \cdot_{T_1 T_2} \nu).$$

Finally, we deduce from our induction hypothesis on T_2 that

$$(\sigma \cdot_{T_1 T_2} \nu) \approx^{\mathbf{lazy}}_{T_2} (\sigma \cdot_{T_1 T_2} \nu).$$

This establishes that the strategy $(\sigma \cdot_{T_1 T_2} \nu)$ is extensional, and in particular, included in $P^{\mathbf{alive}}_{B}$. This contradicts the fact that $\neg(s \upharpoonright B \in P^{\mathbf{alive}}_{B})$ since $s \upharpoonright B \in (\sigma \cdot_{T_1 T_2} \nu)$. We conclude.

(Case 2) It follows from $\neg(s \upharpoonright !A \in R_{!A})$ that the play $s \upharpoonright !A$ is not element of $\|v\|_{!A}$. By definition of $\|v\|_{!A}$ and of $P^{\mathbf{alive}}_{!A}$, this can only mean that μ is not included in $\mathbf{bigcone}(v))$.

Now, we define the strategy ν' as

$$\nu' = \nu \cap \mathbf{bigcone}(v).$$

By lemma 12.3, the strategy ν' is extensional and verifies $U(\nu') = v = U(\nu)$. It follows from the definition of $\approx$ that $\nu \approx^{\mathbf{lazy}}_{T_1} \nu'$; from our induction hypothesis on T_1, that $\nu \sim^{\mathbf{lazy}}_{T_1} \nu'$; from our hypothesis that $\sigma \sim^{\mathbf{lazy}}_{T} \sigma$, that

$$(\sigma \cdot_{T_1 T_2} \nu) \sim^{\mathbf{lazy}}_{T_2} (\sigma \cdot_{T_1 T_2} \nu')$$

and finally, from our induction hypothesis on T_2, that

$$\sigma \cdot_{T_1 T_2} \nu \approx^{\mathbf{lazy}}_{T_2} \sigma \cdot_{T_1 T_2} \nu'. \tag{18}$$

Recall that the play $s\upharpoonright B$ is element of the strategy $(\sigma \cdot_{T_1T_2} \nu)$ and that $s\upharpoonright B \in R_B$. Let $y \in E_B$ be an extension such that $s\upharpoonright B \in \|y\|_B$. Note that $y \in U(\sigma \cdot_{T_1T_2} \nu)$. Equivalence (18) implies that the strategy $(\sigma \cdot_{T_1T_2} \nu)$ is extensional, and thus, that $(\sigma \cdot_{T_1T_2} \nu) \vDash_B y$. Now, equivalence (18) again implies that $(\sigma \cdot_{T_1T_2} \nu') \vDash_B y$.
We show that we reach a contradiction. Observe that the two strategies μ and ν' are included in the strategy $\nu : A$, and thus compatible. At the same time, the strategy μ is not included in $\mathbf{bigcone}(v)$, and thus not included in $\nu' \subset \mathbf{bigcone}()$. It follows from lemma 12.5 that there exists a play $u \in P_B^{\mathbf{even}}$ and a value $m \in M_B$ such that:

- $u \in (\sigma \cdot_{T_1T_2} \nu')$,
- $u \cdot m \sqsubseteq_B s\upharpoonright B$,
- $\forall n \in M_B, u \cdot m \cdot n \notin (\sigma \cdot_{T_1T_2} \nu')$.

Put together with $s\upharpoonright B \in \|y\|_B$, this contradicts $\sigma \cdot_{T_1T_2} \nu' \vDash_B y$. We conclude from (case 1) and (case 2) that $\sigma \subset P^{\mathbf{alive}}_{(!A)\multimap B}$ when $\sigma \sim_T^{\mathbf{lazy}} \sigma$. This ends part ($\bigstar$).

($\bigstar\bigstar$) Suppose that $\sigma \sim_T^{\mathbf{lazy}} \sigma$, that $(v, y) \in U(\sigma)$ and $(w, y) \in U(\sigma)$ for two configurations $v, w \in E_{!A}$ and an extension $y \in E_B$. We claim that $v = w$ when $v \subset w$. Indeed, suppose that $v \subset w$, and let $s \in P_{(!A)\multimap B}$ be a play in $\sigma \cap \|(w, y)\|_{(!A)\multimap B} \neq \emptyset$. The projection $s\upharpoonright !A$ is element of $\|w\|_{!A}$. By definition, there exists a sub-extensional strategy $\mu_1 : A$ such that:

- $s\upharpoonright !A$ explores the strategy μ_1,
- $U(\mu_1) = w$,
- $\mu_1 \subset \mathbf{bigcone}(w)$.

By lemma 11.7, the strategy μ_1 is included in an extensional strategy ν_1 which verifies $U(\nu_1) = w$. By induction hypothesis on T_1 and T_2, one deduces from $\sigma \sim_T^{\mathbf{lazy}} \sigma$ that the strategy $(\sigma \cdot_{T_1T_2} \nu_1)$ is extensional in the cods B. It follows from

$$s\upharpoonright B \in (\sigma \cdot_{T_1T_2} \nu_1) \cap \|y\|_B$$

that $y \in U(\sigma \cdot_{T_1T_2} \nu_1)$ and thus, that $(\sigma \cdot_{T_1T_2} \nu_1) \vDash_B y$.
Similarly, one deduces from $\sigma \cap \|(w, y)\|_{(!A)\multimap B} \neq \emptyset$ that there exists an extensional strategy ν_2 which (1) verifies $U(\nu_2) = v$ and (2) induces an extensional strategy $(\sigma \cdot_{T_1T_2} \nu_2)$ which implements y in the cods B.
Now, we define the strategy

$$\nu_3 = \nu_1 \cap \mathbf{bigcone}(v).$$

By lemma 12.3, the strategy ν_3 verifies the equivalence $\nu_2 \approx_{T_1}^{\mathbf{lazy}} \nu_3$. By induction hypothesis on T_1, the equivalence $\nu_2 \sim_{T_1}^{\mathbf{lazy}} \nu_3$ holds. The equivalence $\sigma \sim_T^{\mathbf{lazy}} \sigma$ implies the equivalence

$$(\sigma \cdot_{T_1T_2} \nu_2) \sim_{T_2}^{\mathbf{lazy}} (\sigma \cdot_{T_1T_2} \nu_3)$$

which implies by induction hypothesis on T_2 the equivalence

$$(\sigma \cdot_{T_1 T_2} \nu_2) \approx^{\mathbf{lazy}}_{T_2} (\sigma \cdot_{T_1 T_2} \nu_3).$$

It follows that the strategy $(\sigma \cdot_{T_1 T_2} \nu_3)$ is extensional and implements y.
Here, we reason as in part (★). We observe that the strategies μ_1 and ν_3 are subset of the strategy ν_1, thus compatible. We proceed by contradiction, and suppose that v is stricly included in w. In that case, the strategy μ_1 is not included in the strategy ν_3, and thus one may apply lemma 12.5 to deduce that there exists a play $u \in P_B^{\mathbf{even}}$ and a cell $m \in M_B$ such that:

- $u \in (\sigma \cdot_{T_1 T_2} \nu_3)$,
- $u \cdot m \sqsubseteq_B s{\upharpoonright}B$,
- $\forall n \in M_B, u \cdot m \cdot n \notin (\sigma \cdot_{T_1 T_2} \nu_3)$.

Put together with $s{\upharpoonright}B \in \|y\|_B$, this contradicts the hypothesis that the strategy $(\sigma \cdot_{T_1 T_2} \nu_3)$ implements y. We conclude that $v = w$.
(★★★) We proceed by contradiction, and suppose that there exists an extension $(v, y) \in E_{!A \multimap B}$ such that $(v, y) \in U(\sigma)$ but σ does not implement (v, y). Let s_1 be a play of $\sigma \cap \|(v, y)\|_{(!A) \multimap B}$. We repeat the pattern met in (★★). The projection $s_1{\upharpoonright}!A$ is element of $\|v\|_{!A}$. Thus, there exists a sub-extensional strategy $\mu_1 : A$ such that:

- $s_1{\upharpoonright}!A$ explores the strategy μ_1,
- $U(\mu_1) = v$,
- $\mu_1 \subset \mathbf{bigcone}(v)$.

By lemma 11.7, the strategy μ_1 is included in an extensional strategy ν_1 which verifies $U(\nu_1) = v$. By induction hypothesis on T_1 and T_2 and hypothesis $\sigma \sim^{\mathbf{lazy}}_T \sigma$, one deduces that the strategy $(\sigma \cdot_{T_1 T_2} \nu_1)$ is extensional in the cods B. It follows from $s_1{\upharpoonright}B \in (\sigma \cdot_{T_1 T_2} \nu_1) \cap \|y\|_B$ that $(\sigma \cdot_{T_1 T_2} \nu_1) \vDash_B y$.
On the other hand, we know that the strategy σ does not implement (v, y). This means that there exists a play $t \in \sigma$ and a cell $m \in M_{A \multimap B}$ such that

$$t \cdot m \preceq_{(!A) \multimap B} (v, y) \tag{19}$$

and:

(1) either $\forall n \in M_{A \multimap B}, \neg(t \cdot m \cdot n \in \sigma)$,
(2) or $\exists n \in M_{A \multimap B}, t \cdot m \cdot n \in \sigma$ and $\neg(t \cdot m \cdot n \preceq_{A \multimap B} (v, y))$.

In both cases, the assertion $t \preceq_{(!A) \multimap B} (v, y)$ in (19) means that the play t is prefix of a play $s_2 \in \|(v, y)\|_{(!A) \multimap B}$. We apply another time the pattern met in (★★). By definition, $s_2{\upharpoonright}!A \in \|v\|_{!A}$ means that there exists a sub-extensional strategy $\mu_2 : A$ such that:

- $s_2{\upharpoonright}!A$ explores the strategy μ_2,

- $U(\mu_2) = v$,
- $\mu_2 \subset \mathbf{bigcone}(v)$.

By lemma 11.7, the strategy μ_2 is included in an extensional strategy ν_2 which verifies $U(\nu_2) = v$. By lemma 12.3, we may even choose $\nu_2 \subset \mathbf{bigcone}(v)$. By definition of $\approx_{T_1}^{\mathbf{lazy}}$, the two strategies ν_1 and ν_2 are $\approx_{T_1}^{\mathbf{lazy}}$-equivalent. By applying our induction hypothesis on T_1 and T_2, as well as the hypothesis $\sigma \sim_T^{\mathbf{lazy}} \sigma$, we deduce that $(\sigma \cdot_{T_1T_2} \nu_1) \approx_{T_2}^{\mathbf{lazy}} (\sigma \cdot_{T_1T_2} \nu_2)$. From this, it follows that $(\sigma \cdot_{T_1T_2} \nu_2) \vDash_B y$. We claim that this is not possible. We start from the definition of the play $t \in P^{\mathbf{even}}_{(!A)\multimap B}$ and $m \in M_{!A\multimap B}$ in (19). It follows from $t \cdot m \in P^{\mathbf{odd}}_{(!A)\multimap B}$ that $t \cdot m \upharpoonright B \in P^{\mathbf{odd}}_B$. So, the play $t \cdot m \upharpoonright B$ factors as $t \cdot m = u \cdot p$ where $u \in (\sigma \cdot_{T_1T_2} \nu_2)$ is an even-length play and $p \in M_B$ is a cell. It follows from (19) that $u \cdot p \preceq_B y$. We proceed by case analysis.

- when $\forall n \in M_{A\multimap B}, \neg(t \cdot m \cdot n \in \sigma)$, there is no value $q \in M_B$ such that $u \cdot p \cdot q \in (\sigma \cdot_{T_1T_2} \nu_2)$.
- when $\exists n \in M_{A\multimap B}, t \cdot m \cdot n \in \sigma$ and $\neg(t \cdot m \cdot n \preceq_{A\multimap B} (v, y))$ and n is a move in the component A, then the two hypothesis

  $$t \cdot m \preceq_{(!A)\multimap B} (v, y) \qquad \text{and} \qquad \neg(t \cdot m \cdot n \preceq_{(!A)\multimap B} (v, y))$$

  imply together that the move n, considered as a play of A, is not element of $\mathbf{bigcone}(v)$. We were careful to choose a strategy $\nu_2 : A$ included in $\mathbf{bigcone}(v)$. It follows that the strategy "does not answer" to the move n, in the sense that there exists no move $n' \in M_{!A}$ such that

  $$((t \cdot m \cdot n) \upharpoonright !A) \cdot n' \in (\nu_2)^{\dagger}.$$

  We conclude that there is no value $q \in M_B$ such that $u \cdot p \cdot q \in (\sigma \cdot_{T_1T_2} \nu_2)$.
- when $\exists n \in M_{A\multimap B}, t \cdot m \cdot n \in \sigma$ and $\neg(t \cdot m \cdot n \preceq_{A\multimap B} (v, y))$ and n is a move in the component B, then

  $$(t \cdot m \cdot n) \upharpoonright B \in (\sigma \cdot_{T_1T_2} \nu_2)$$

  and either $\neg((t \cdot m \cdot n) \upharpoonright B \preceq_B y)$ and we are done in that case, or

  $$(t \cdot m \cdot n) \upharpoonright B \in \|y\|_B \quad \text{and} \quad (t \cdot m \cdot n) \upharpoonright !A \notin \|v\|_{!A}.$$

  In that last case, we know that $(t \cdot m \cdot n) \upharpoonright !A \in R_{!A}$ by (★). Thus, there exists a configuration w such that $(t \cdot m \cdot n) \upharpoonright !A \in \|w\|_{!A}$, and $w \subsetneq v$ because

  $$(t \cdot m \cdot n) \upharpoonright !A \sqsubseteq s_2 \quad \text{and} \quad (t \cdot m \cdot n) \upharpoonright !A \notin \|v\|_{!A}.$$

  It follows that $(w, y) \in U(\sigma)$, which contradicts (★★).

In the three cases, we may conclude that the strategy $(\sigma \cdot_{T_1T_2} \nu_2)$ does not implement the extension $y \in E_B$. This concludes **Part I** of the proof, and shows that when two strategies σ and τ verify the equivalence $\sigma \sim_T^{\mathbf{lazy}} \tau$, then the strategies σ and τ are extensional.

Part II: Suppose that $\sigma \cap \|(v, y)\|_{!A\multimap B}$ is nonempty and that $\tau \cap \|(v, y)\|_{!A\multimap B}$ is empty. We know from ($\bigstar\bigstar$) that $\sigma \cap \|(w, y)\|_{!A\multimap B}$ is empty for every strict subset $w \subsetneq v$. We may therefore suppose without loss of generality that $\tau \cap \|(w, y)\|_{!A\multimap B}$ is empty for every subset $w \subset v$.
We know from Part I that the strategies σ and τ are extensional. Besides, there exists an extensional strategy $\nu : A$ such that $U(\nu) = v$. By lemma 11.9 (compositionality), the strategy $\sigma \cdot_{T_1T_2} \nu$ is extensional and implements the extension y. On the other hand, by compositionality again, the strategy $\tau \cdot_{T_1T_2} \nu$ does not implement y. It follows from the definition of $\approx_{T_2}^{\mathbf{lazy}}$ that the equivalence

$$(\sigma \cdot_{T_1T_2} \nu) \approx_{T_2}^{\mathbf{lazy}} (\tau \cdot_{T_1T_2} \nu)$$

does not hold; and by induction hypothesis on T_2, that the equivalence

$$(\sigma \cdot_{T_1T_2} \nu) \sim_{T_2}^{\mathbf{lazy}} (\tau \cdot_{T_1T_2} \nu)$$

does not hold either. We conclude from $\nu \sim_{T_1}^{\mathbf{lazy}} \nu$ that the strategies σ and τ are not $\sim_T^{\mathbf{lazy}}$-equivalent. This concludes Part II.
We deduce from Part I and Part II that two $\sim_T^{\mathbf{lazy}}$-equivalent strategies σ and τ of the collapse data structure $[T]_{\mathrm{CODS}}^{\mathbf{lazy}}$ are also $\approx_T^{\mathbf{lazy}}$-equivalent. This conclude the proof of theorem 12.6. ■

12.3 The collapse theorem

Ehrhard's collapse theorem follows quite immediately from theorem 12.6.
Corollary 12.7 (collapse theorem) *The strongly stable model is the extensional collapse of the sequential algorithm model.*
PROOF We conclude from theorem 12.6 that the hierarchy $[-]_{\mathrm{CODS}}^{\mathbf{lazy}}$ collapses to the strongly stable hierarchy $[-]_{\mathrm{HC}}$. Ehrhard's collapse theorem follows immediately from lemma 9.3. ■

Remark. The proof of theorem 12.6 is quite elaborate. In that respect, it should be compared to the proof in [Barreiro, Ehrhard 1998] that the hierarchy $[-]_{\mathrm{MSET}}$ generated by the coherence space model of LL with multiset exponentials, collapses extensionally to Berry stable hierarchy $[-]_{\mathrm{S}}$. We show in [Melliès 2004a] that Barreiro and Ehrhard's result may be also established by exhibiting a back-and-forth translation between the hierarchies $[-]_{\mathrm{MSET}}$ and $[-]_{\mathrm{S}}$. We leave it as an open question whether a similar translation technique may be applied to establish that the sequential algorithm hierarchy collapses to the strongly stable hierarchy.

13 Conclusion

We analyze the extensional content of Berry-Curien sequential algorithm model by shifting from sequential games plays on trees to sequential games played on graphs. This clarifies the sequential nature of hypercoherence spaces, and the reasons why the sequential algorithm hierarchy collapses extensionally to Bucciarelli-Ehrhard strongly stable hierarchy. These results should advocate more asynchronous and concurrent forms of game semantics — even in the study of sequentiality.

References

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

[Abramsky et al. 1994] S. Abramsky, R. Jagadeesan and P. Malacaria. Full abstraction for PCF (extended abstract). In M. Hagiya and J.C. Mitchell, editors, *Theoretical Aspects of Computer Software*, page 1-15. Springer Verlag, 1994.

[Abramsky, McCusker 1999] S. Abramsky and G. McCusker. Game semantics. In *Computational Logic: Proceedings of the 1997 Marktoberdorf Summer School.* H. Schwichtenberg and U. Berger eds, Springer Verlag, 1999.

[Abramsky, Melliès 1999] S. Abramsky and P.-A. Melliès. Concurrent games and full completeness. In *Proc. of the 14th Int. Symposium on Logic in Computer Science*, (Computer Society Press of the IEEE), 1999.

[Abramsky et al. 2000] S. Abramsky, R. Jagadeesan and P. Malacaria. Full abstraction for PCF. Information and Computation, Volume 163, Number 2, pp 409–470, 2000.

[Abramsky 2001] S. Abramsky. Sequentiality vs. Concurrency in Games and Logic. Oxford University Programming Research Group Research Report RR-01-08

[Amadio, Curien 1998] R. Amadio and P.-L. Curien. *Domains and Lambda-Calculi.* Cambridge University Press, 1998.

[Barreiro, Ehrhard 1998] N. Barreiro and T. Ehrhard. Anatomy of an extensional collapse. *Manuscript*, 1998.

[Berry 1979] G. Berry. *Modèles complètement adéquats et stables des λ-calcul typés.* Thèse de doctorat d'Etat, Université Paris VII, 1979.

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

[Bierman 1995] G. Bierman. What is a categorical model of intuitionistic linear logic? *Typed Lambda-Calculus and Applications*, 1995.

[Boudes 2002] P. Boudes, Hypercohérence et jeux. *PhD thesis,* Marseille, 2002.

[Bucciarelli, Ehrhard 1991] A. Bucciarelli and T. Ehrhard. Sequentiality and strong stability. *Proceedings LICS'91*, Amsterdam, 1991.

[Curien 1993] P.-L. Curien. On the symmetry of sequentiality. In *Proceedings of Mathematical Foundations of Programming Semantics 1993*, LNCS 802, Springer Verlag, 1993.

[Ehrhard 1993] T. Ehrhard. Hypercoherences: a strongly stable model of linear logic. *Mathematical Structures in Computer Science*, 3:365-385, 1993.

[Ehrhard 1996] T. Ehrhard. Projecting sequential algorithms on strongly stable functions. *Annals of Pure and Applied Logic*, 77:201-244, 1996.

[Ehrhard 1997] T. Ehrhard. A relative definability result for strongly stable functions and some corollaries. *Information and Computation*, 1997.

[Ehrhard 2000] T. Ehrhard. Parallel and serial hypercoherences. *Theoretical Computer Science*, North-Holland, 247, 39–81, 2000.

[Girard 1995] J.-Y. Girard. Linear logic: its syntax and semantics. In *Advances in Linear Logic*, J.-Y. Girard, Y. Lafont and L. Regnier, editors, volume 222 of *London Mathematical society Lecture Notes Series*. Cambridge University Press, 1995.

[Hyland, Ong 2000] M. Hyland and L. Ong. On full abstraction for PCF: I, II and III. *Information and Computation.* Volume 163, Number 2, pp. 285-408, December 2000.

[Hyland, Schalk 2002] M. Hyland and A. Schalk. Games on Graphs and Sequentially Realizable Functionals. Extended Abstract. In *Proc. Logic in Computer Science 2002,* IEEE Computer Science Press, pp. 257–264. 2002.

[Joyal 1977] A. Joyal. Remarques sur la théorie des jeux à deux personnes. Gazette des Sciences Mathématiques du Québec, Vol I, no 4, pp.46-52, 1977.

[Joyal 1997] A. Joyal. Free lattices, communication and money games. In M.L. Dalla Chiara et al. (eds.), *Logic and Scientifi c Methods*, 29-68. Kluwer Academic Publishers. 1997.

[Lamarche 1992] F. Lamarche. Sequentiality, games and linear logic. *Manuscript*, 1992.

[Longley 1998] J. Longley. *The sequentialy realizable functionals.* Submitted to Journal of Symbolic Logic, 1998.

[McCusker, 1998] G. McCusker. *Games and full abstraction for a functional metalanguage with recursive types*. Distinguished Dissertation Series, Springer Verlag, 1998.

[Melliès 1998] P.-A. Melliès. Axiomatic Rewriting Theory 4: a stability theorem in rewriting theory. *Proceedings of Logic in Computer Science*, 1998.

[Melliès 2000] P.-A. Melliès. Sequentiality and extensionality. 13 pages. Unpublished notes circulated at the Linear Summer School. Azores, September 2000.

[Melliès 2003] P.-A. Melliès. Categorical models of linear logic revisited. Prépublication PPS n^o 22. September 2003. To appear in Theoretical Computer Science.

[Melliès 2004a] P.-A. Melliès. Comparing hierarchies of types in models of linear logic. *Information and Computation*, Volume 189, Issue 2, Pages 202-234, March 2004.

[Melliès 2004b] P.-A. Melliès. Asynchronous games 2: the true concurrency of innocence. *Proceedings of the 15th International Conference on Concurrency Theory*, London, 2004. Lecture Notes in Computer Science 3170, pp. 448-465, Springer Verlag.

[Nickau 1994] H. Nickau. Hereditarily sequential functionals. *Proceedings of the Symposium on Logical Foundations of Computer Science: Logic at St. Petersburg*, Lecture Notes in Computer ScienceSpringer Verlag, 1994.

[van Oosten, 1997] J. van Oosten. A combinatory algebra for sequential functionals of finite type. *Technical Report 996*, University of Utrecht, 1997. Also published in *Models and Computability*, S. B. Cooper and J. K. Truss eds, Cambridge University Press, pp 389–406, 1999.

[Plotkin, 1977] G. Plotkin. LCF considered as a programming language. *Theoretical Computer Science*, 5:223-255, 1977

[Pratt 1991] V. Pratt. Modeling Concurrency with Geometry, *Proc. 18th Ann. ACM Symposium on Principles of Programming Languages*, 311-322, January, 1991.

[Santocanale 2002] L. Santocanale. Mu-Bicomplete Categories and Parity Games. *Theoretical Informatics and Applications*, 36 (2002) 195-227, September 2002.

[Scott 1969] D. Scott. A type-theoretical alternative to ISWIM, CUCH, OWHY. *Theoretical Computer Science*. 121:411-440, 1993. Manuscript circulated since 1969.