# Transport of finiteness structures and applications

Christine Tasson, Lionel Vaux

# Transport of finiteness structures and applications

Christine Tasson[1]* and Lionel Vaux[2]*
[1] Preuves, Programmes et Systèmes, CNRS UMR 7126, Paris, France.
[2] Institut de Mathématiques de Luminy, CNRS UMR 6206, Marseille, France.

December 31, 2010

**Abstract**

We describe a general construction of finiteness spaces which subsumes the interpretations of all positive connectors of linear logic. We then show how to apply this construction to prove the existence of least fixpoints for particular functors in the category of finiteness spaces: these include the functors involved in a relational interpretation of lazy recursive algebraic datatypes along the lines of the coherence semantics of system T.

## 1 Introduction

Finiteness spaces were introduced by Ehrhard (2005), refining the purely relational model of linear logic. A finiteness space is a set equipped with a finiteness structure, i.e. a particular set of subsets which are said to be finitary; and the model is such that the relational denotation of a proof in linear logic is always a finitary subset of its conclusion. Applied to this finitary relational model of linear logic, the usual co-Kleisli construction provides a cartesian closed category, hence a model of the simply typed $\lambda$-calculus (see, e.g., Bierman (1995)). The crucial property of finiteness spaces is that the intersection of two finitary subsets of dual types is always finite. This feature allows to reformulate the quantitative semantics of Girard (1988) in a standard algebraic setting, where morphisms interpreting typed $\lambda$-terms are analytic functions between the topological vector spaces generated by vectors with finitary supports. This provided the semantic foundations of the differential $\lambda$-calculus of Ehrhard and Regnier (2003) and motivated the general study of a differential extension of linear logic (Ehrhard and Regnier; 2005, 2006; Ehrhard and Laurent; 2007; Tranquilli; 2008; Vaux; 2009b; Tasson; 2009; Pagani and Tasson; 2009, *etc.*).

The fact that finiteness spaces form a model of linear logic can be understood as a property of the relational interpretation: as we have already mentioned, the relational semantics of a proof is always finitary. The present paper studies the connexion between the category Rel of sets and relations and the category Fin of finiteness spaces and finitary relations, while maintaining a similar standpoint: we investigate whether and how some of the most distinctive features of Rel can be given counterparts in Fin.

Our primary contribution is a very general construction of finiteness spaces: given a relation from a set $A$ to a finiteness space such that the relational image of every element is finitary, we can form a new finiteness space on $A$ whose finitary subsets are exactly those with finitary image. We refer to this result as the *transport lemma*. Although simple in its formulation, the transport lemma subsumes many constructions in finiteness spaces and in particular those interpreting the positive connectives of linear logic (multiplicative "$\otimes$", additive "$\oplus$" and exponential "!") whose action on sets is given by the corresponding relational interpretations. We moreover provide sufficient conditions for a functor in Rel to give rise to a functor in Fin *via* the transport lemma: again, this generalizes the functoriality of the positive connectives of linear logic.

*This work has been partially funded by the French ANR projet blanc "Curry Howard pour la Concurrence" CHOCO ANR-07-BLAN-0324.

The category $\underline{\mathrm{Rel}}$, endowed with inclusion on sets and relations, is enriched on complete partial orders (cpo). This structure was studied in a more general 2-categorical setting (Carboni et al.; 1984, 1991) and the properties of monotonic functors allowed for an abstract description of datatypes (Backhouse et al.; 1991; Hoogendijk and De Moor; 2000; Backhouse and Hoogendijk; 2003). In such a setting, it is standard to define recursive datatypes, such as lists or trees, as the least fixpoints of particular Scott-continuous functors (Smyth and Plotkin; 1982). This prompted us to consider two orders on finiteness spaces derived from set inclusion: the most restrictive one, *finiteness extension*, was used by Ehrhard (2005, unpublished preliminary version) to provide an interpretation of second order linear logic, while the largest one, *finiteness inclusion*, is a cpo on finiteness spaces. We study various notions of continuity for functors in finiteness spaces, and relate them with the existence of fixpoints. A striking feature of this development is that we are led to consider the properties of functors w.r.t. both orders simultaneously: continuity for finiteness inclusion, and monotonicity for finiteness extension. We prove in particular that every functor obtained by applying the transport lemma to a continuous relational functor satisfies these properties, and admits a least fixpoint for finiteness inclusion.

The remaining of the paper is dedicated to the application of these results to the relational semantics of functional programming with recursive datatypes. Indeed, the co-Kleisli construction applied to the relational model of linear logic gives rise to the cartesian closed category $\underline{\mathrm{Rel}}^!$. The fact that the already mentioned co-Keisli $\underline{\mathrm{Fin}}^!$ of $\underline{\mathrm{Fin}}$ provides a model of the $\lambda$-calculus can again be understood as a property of the interpretation in $\underline{\mathrm{Rel}}^!$: the relational semantics of a *simply typed* $\lambda$-term is always finitary. It is however worth noticing that, whereas the relational model can accomodate untyped $\lambda$-calculi (de Carvalho; 2008; Bucciarelli et al.; 2007), finiteness spaces are essentially a model of termination. The whole point of the finiteness construction is to reject infinite computations, ensuring that the intermediate sets involved in the relational interpretation of a cut are all finite. In particular, the relational semantics of fixpoint combinators is finitary only on empty types: general recursion is ruled out from this framework. This is to be related with the fact that finitary relations are *not* closed under arbitrary unions: in contrast with the cpo defined on objects by set inclusion, the category $\underline{\mathrm{Fin}}^!$ (and thus $\underline{\mathrm{Fin}}$) is not enriched on complete partial orders.

Despite this restrictive design, Ehrhard (2005, Section 3) was able to define a finitary interpretation of tail-recursive iteration: this indicates that the finiteness semantics can accomodate a form of typed recursion. This interpretation, however, is not completely satisfactory: tail recursive iteration is essentially linear, thus it does not provide a type of natural numbers (Thibault; 1982; Lambek and Scott; 1988) in the associated model of the $\lambda$-calculus. This is essentially due to the fact that the interpretation of natural numbers is *flat* (in the sense of domains). In fact, a similar effect was already noted by Girard in the design of his coherence semantics of system $T$ (Girard et al.; 1989): his solution was to propose a *lazy* interpretation of natural numbers, where laziness refers to the possibility of pattern matching on non normal terms. The second author remarked that the same solution could be adapted in the relational model and provided a type of natural numbers with finitary recursor, hence a model of system $T$ (Vaux; 2009c).

Our previous developments allow us to generalize this construction: after introducing a finitary relational interpretation of sum types, we consider the fixpoints of particular functors and show that they provide a relational semantics of the typed $\lambda$-calculus with lazy recursive algebraic datatypes by exhibiting their constructors and destructors. Adapting the techniques already employed by the second author in the case of system $T$, we moreover show that these operators are finitary.

**Related and future work.**

Our first interest in the semantics of datatypes in finiteness spaces was the possibility of extending the quantitative semantics of the simply typed $\lambda$-calculus in vectorial finiteness spaces to functional programming with base datatypes. This would broaden the scope of the already well developped proof theory of differential linear logic: the quantitative semantics provides more precise information on cut elimination, and is thus a better guide in the design of syntax than the

plain relational interpretation. Earlier achievements in this direction include the extension of the algebraic $\lambda$-calculus (Vaux; 2009a) with a type of booleans, for which the first author established a semantic characterization of total terms: this is moreover proved to be complete on boolean functions (Tasson; 2009). In previous unpublished work, we also proposed a quantitative semantics of tail recursive iteration. As we mentioned before, this did not provide a semantics of system $T$, which prompted us to investigate the general structure of standard datatypes in finiteness spaces. In this regard, our present contribution is an important step.

Notice that another standard approach to recursive datatypes is to consider the impredicative encoding of inductive datatypes in system $F$. In an unpublished preliminary version of his paper on finiteness spaces, Ehrhard proposed an interpretation of second order linear logic. This is based on a class of functors which, in particular, are monotonic for the finiteness extension order. Notice however that this does not provide a denotational semantics *stricto sensu*: in general, the interpretation decreases under cut elimination. Moreover, the possibility of a quantitative semantics in this setting is not clear.

Other accounts of type fixpoints in linear logic include the system of linear logic proof nets with recursion boxes of Gimenez (2009), which allows to interpret, e.g., PCF. As such this system can be seen as a graphical syntax for general recursion. Along similar lines, Fernández et al. (2009) have proposed a system of interaction nets which models iteration on recursive datatypes. In both cases, no particular denotational semantics is considered. Let us also mention Baelde and Miller's $\mu$MALL (2007) which replaces the exponential modalities of linear logic with least and greatest fixpoints: less close to our contribution, this work is mainly oriented towards proof search. It however introduces the system $\mu$LJ of intuitionistic logic with fixpoints, for which Clairambault (2010) later proposed a cut elimination procedure allowing to encode system $T$, together with a game semantics accounting for typed recursion.

The notion of transport functor we use to describe how functors in Fin can be derived from functors in Rel is similar to the categorical characterization of container types as relators with membership, by Hoogendijk and De Moor (2000): relators are functors in Rel which are monotonic for inclusion of relations; membership relations are particular lax natural transformations associated with these functors. The hypotheses we consider on the functors in Rel underlying transport functors in Fin are weaker than those on relators with membership. On the other hand, in order to ensure the functoriality of transport in Fin, we are led to refer to a *shape* relation: this side condition is essential for some important instances, such as linear logic exponentials.

The relationship we establish between Rel and Fin might profitably be recast in a more general setting. At least the transport lemma can be reformulated for coherence spaces rather than finiteness spaces. Further results of the paper might follow as well, up to some local tweaking of the definitions (e.g., that of shape relations). It is still unclear to us whether the approach we developped is limited to the restricted setting of finiteness spaces, coherence spaces and maybe other web-based models (Ehrhard's hypercoherences (1993), Loader's totality spaces (1994)), or if it can be generalized in the spirit of the glueing and orthogonality techniques studied by Hyland and Schalk (2003).

**Outline of the paper and main results.**

In section 2, we review the structure and properties of Rel. We establish the transport lemma in section 3, and derive the interpretations of the positive connectives of linear logic in Fin from those in Rel. Section 4 introduces two orders on finiteness spaces and associated properties. In particular we provide sufficient conditions for the existence of fixpoints of functors. We moreover prove these conditions are automatically satisfied by transport functors. The last two sections are dedicated to the finitary relational semantics of $\lambda$-calculi: we first recall the semantics of the simply typed $\lambda$-calculus in section 5, and then detail the semantics of recursive algebraic datatypes in section 6.

# 2 Sets and relations

## 2.1 Notations

We write $\mathbf{N}$ for the set of all natural numbers. Let $A$ and $B$ be sets. We write $A \subseteq B$ if $A$ is a subset of $B$ (not necessarily a strict one), and $A \subseteq_{\mathrm{f}} B$ if moreover $A$ is finite. We write $\# A$ for the cardinality of $A$, $\mathfrak{P}(A)$ for the powerset of $A$ and $\mathfrak{P}_{\mathrm{f}}(A)$ for the set of all finite subsets of $A$. We identify multisets of elements of $A$ with functions $A \longrightarrow \mathbf{N}$. If $\mu$ is such a multiset, we write $supp(\mu)$ for its support set $\{\alpha \in A;\ \mu(\alpha) \neq 0\}$. A finite multiset is a multiset with a finite support. We write $\mathfrak{M}_{\mathrm{f}}(A)$ for the set of all finite multisets of elements of $A$. Whenever $(\alpha_1, \ldots, \alpha_n) \in A^n$, we write $[\alpha_1, \ldots, \alpha_n]$ for the corresponding finite multiset: $\alpha \in A \mapsto \#\{i;\ \alpha_i = \alpha\}$. We also write $\#[\alpha_1, \ldots, \alpha_n] = n$ for the cardinality of multisets. The empty multiset is $[]$ and we use the additive notation for multiset union, i.e. $\mu + \mu' : \alpha \in A \mapsto \mu(\alpha) + \mu'(\alpha)$.

Since we will often consider numerous notions associated with a fixed set, we introduce the following typographic conventions: we will in general use latin majuscules for reference sets (e.g. $A$), greek minuscules for their elements (e.g. $\alpha, \alpha' \in A$), latin minuscules for subsets (e.g. $a \subseteq A$), gothic majuscules for sets of subsets (e.g. $\mathfrak{A} \subseteq \mathfrak{P}(A)$), and script majuscules for finiteness spaces (e.g. $\mathcal{A} = (A, \mathfrak{A})$). In general, if $T$ is an operation on sets we derive the notations for elements, subsets, *etc.* of $TA$ from those for elements, subsets, *etc.* of $A$ by the use of various overscripts (e.g. $\widetilde{\alpha} \in \widetilde{a} \subseteq TA$). We reserve overlining for multisets (e.g. $\overline{\alpha} = [\alpha_1, \ldots, \alpha_n] \in \mathfrak{M}_{\mathrm{f}}(A)$).

We will also consider families of objects (sets, elements, finiteness spaces, *etc.*) and thus introduce the following conventions. Unless stated otherwise, all families considered in the same context are based on a common set of indices, say $I$. We then write e.g. $\overrightarrow{A}$ for the family $(A_i)_{i \in I}$. We moreover use generic notations for componentwise operations on families: for instance if $\overrightarrow{A}$ and $\overrightarrow{B}$ are two families of sets, we may write $\overrightarrow{A \cup B}$ for $(A_i \cup B_i)_{i \in I}$, and $\overrightarrow{\mathfrak{P}(A)}$ for $(\mathfrak{P}(A_i))_{i \in I}$. We may also write, e.g., $\overrightarrow{A \subseteq B}$ for $A_i \subseteq B_i$ for all $i \in I$.

Assume $\overrightarrow{A}$ is a family of sets. We write $\prod \overrightarrow{A}$ for the cartesian product of the $A_i$'s and $\sum \overrightarrow{A}$ for their coproduct ($I$-indexed disjoint union): $\prod \overrightarrow{A} = \{\overrightarrow{\alpha};\ \forall i \in I,\ \alpha_i \in A_i\}$ and $\sum \overrightarrow{A} = \{(i, \alpha);\ i \in I \wedge \alpha \in A_i\}$. We may of course denote finite products and coproducts of sets as usual, e.g. $A \times B$ and $A + B$: in that case we assume indices are natural numbers starting from 1, e.g. $A + B = \{(1, \alpha);\ \alpha \in A\} \cup \{(2, \beta);\ \beta \in B\}$.

## 2.2 The category of sets and relations

Let $A$ and $B$ be sets and $f$ be a relation from $A$ to $B$: $f \subseteq A \times B$. We then write ${}^t f$ for the transpose relation $\{(\beta, \alpha) \in B \times A;\ (\alpha, \beta) \in f\}$. For all subset $a \subseteq A$, we write $f \cdot a$ for the *direct image* of $a$ by $f$: $f \cdot a = \{\beta \in B;\ \exists \alpha \in a,\ (\alpha, \beta) \in f\}$. If $\alpha \in A$, we will also write $f \cdot \alpha$ for $f \cdot \{\alpha\}$. We say that a relation $f$ is *quasi-functional* if $f \cdot \alpha$ is finite for all $\alpha$. If $b \subseteq B$, we define the *division* of $b$ by $f$ as $f \setminus b = \{\alpha \in A;\ f \cdot \alpha \subseteq b\}$. This is the greatest subset of $A$ that $f$ maps to a subset of $b$: $f \setminus b = \bigcup \{a \subseteq A;\ f \cdot a \subseteq b\}$. Notice that in general $f \cdot (f \setminus b)$ may be a strict subset of $b$, and $f \setminus (f \cdot a)$ may be a strict superset of $a$.

When $f \subseteq A \times B$ and $g \subseteq B \times C$, we denote by $g \circ f$ their composite: $(\alpha, \gamma) \in g \circ f$ iff there exists $\beta \in B$ such that $(\alpha, \beta) \in f$ and $(\beta, \gamma) \in g$. Notice that this definition does not actually depend on the *types* of $f$ and $g$ (namely the pairs of sets $(A, B)$ and $(B, C)$, respectively). The identity relation on $A$ is the diagonal: $\mathsf{id}^A = \{(\alpha, \alpha);\ \alpha \in A\} \subseteq A \times A$.

**Proposition 2.1** *Let $f \subseteq A \times B$ be a relation. Then:*

- *$f = v \circ {}^t u$ where $u$ and $v$ are (graphs of) functions, namely $u = \{((\alpha, \beta), \alpha);\ (\alpha, \beta) \in f\}$ and $v = \{((\alpha, \beta), \beta);\ (\alpha, \beta) \in f\}$, i.e. the projections from $f$ onto its domain and image respectively;*
- *$f$ is a function from $A$ to $B$ iff there exists $g \subseteq B \times A$ such that $f \circ g \subseteq \mathsf{id}^B$ and $\mathsf{id}^A \subseteq g \circ f$, and we then have $g = {}^t f$.*

Equipped with the above relational composition, relations form a category $\underline{\mathrm{Rel}}$ whose objects are sets. More precisely, morphisms in $\in \underline{\mathrm{Rel}}(A,B)$ are triples $(A,B,f)$ such that $f \subseteq A \times B$: we use this trick only to ensure that homsets in $\underline{\mathrm{Rel}}$ are pairwise disjoint, which is part of the definition of a category (see, e.g., Mac Lane (1998)). Then the identity morphism on set $A$ is $(A,A,\mathsf{id}^A)$ and, if $(A,B,f) \in \underline{\mathrm{Rel}}(A,B)$ and $(B,C,g) \in \underline{\mathrm{Rel}}(B,C)$ then their composite in $\underline{\mathrm{Rel}}$ is $(A,C,g \circ f) \in \underline{\mathrm{Rel}}(A,C)$. Most of the time, we will abuse notations and identify morphisms in $\underline{\mathrm{Rel}}$ with the underlying relations, especially when types are irrelevant for the discussion or clear from the context: we may then simply write, e.g., $f$ for $(A,B,f)$. It is however important to notice that the action of functors on relations may in general depend on their types:

**Definition 2.2** *A* functor $T$ *in* $\underline{\mathrm{Rel}}$ *is the data of a set* $TA$ *for all set* $A$ *and a relation* $T^{A,B}f \subseteq TA \times TB$ *for all* $f \subseteq A \times B$*, so that* $T(A,B,f) = (TA, TB, T^{A,B}f) \in \underline{\mathrm{Rel}}(TA,TB)$*, preserving identities and composition:* $T^{A,A}\mathsf{id}^A = \mathsf{id}^{TA}$ *for all set* $A$*, and* $T^{A,C}(g \circ f) = T^{B,C}g \circ T^{A,B}f$ *for all relations* $f \subseteq A \times B$ *and* $g \subseteq B \times C$*.* Cofunctors *are defined similarly, except for being contravariant, i.e.* $T^{A,B}f \subseteq TB \times TA$ *and* $T^{A,C}(g \circ f) = T^{A,B}f \circ T^{B,C}g$ *for all relations* $f \subseteq A \times B$ *and* $g \subseteq B \times C$*.*

The simplest example of a functor (resp. cofunctor) is the identity functor (resp. the transpose functor $({}^t\cdot)$), which is the identity on sets and maps every morphism $(A,B,f)$ to itself (resp. to its transpose $(B,A,{}^tf)$). For all the care we took in making this definition precise, most of the time we will leave out the superscripts and simply write $Tf$ both for $T^{A,B}f$ and the associated morphism. Moreover, we will sometimes restrict our study to classes of functors for which such scripts are not actually relevant: we say a functor $T$ is *type blind* if $T^{A,B}f = T^{A',B'}f$ for all $f \subseteq (A \times B) \cap (A' \times B')$.

A functor $T$ is *monotonic on sets* if $TA \subseteq TB$ for all sets $A \subseteq B$. If moreover $T^{A,B}\mathsf{id}^A = \mathsf{id}^{TA}$ (resp. $T^{B,A}\mathsf{id}^A = \mathsf{id}^{TA}$) for all sets $A \subseteq B$, we say $T$ *preserves inclusions* (resp. *reverse inclusions*).

**Lemma 2.3** *Let* $T$ *be a functor. Then* $T$ *preserves inclusions (resp. reverse inclusions) iff* $T$ *is monotonic on sets and, for all* $A \subseteq A'$*,* $B \subseteq B'$ *and* $f \subseteq A \times B$*, we have* $T^{A,B}f = T^{A',B'}f \circ \mathsf{id}^{TA}$ *(resp.* $T^{A,B}f = \mathsf{id}^{TB} \circ T^{A',B'}f$*). Moreover* $T$ *preserves both inclusions and reverse inclusions iff* $T$ *is type blind.*

**Proof** Assume $T$ preserves inclusions, $A \subseteq A'$, $B \subseteq B'$ and $f \subseteq A \times B$. Then $T^{A',B'}f \circ \mathsf{id}^{TA} = T^{A',B'}(\mathsf{id}^B \circ f) \circ T^{A,A'}\mathsf{id}^A = T^{B,B'}\mathsf{id}^B \circ T^{A,B}(f \circ \mathsf{id}^A) = \mathsf{id}^{TB} \circ T^{A,B}f = T^{A,B}f$. For the converse, take $A = A' = B \subseteq B'$ and $f = \mathsf{id}^A$. The case of reverse inclusion preserving functors is similar. We conclude since type blindness is just the conjunction of both characterizations. □

A functor $T$ is called a *relator* (Backhouse et al.; 1991) if it is monotonic for relation inclusion: $T^{A,B}f \subseteq T^{A,B}f'$ as soon as $f \subseteq f' \subseteq A \times B$. We say a functor $T$ is *symmetric* if $T\left({}^tf\right) = {}^t(Tf)$ for all $f$.

**Lemma 2.4** *A type blind functor is always a relator. Moreover, every relator is a symmetric functor.*

**Proof** Assume $T$ is type blind and $f \subseteq f' \subseteq A \times B$. By the first item of Proposition 2.1, we can write $f = v \circ {}^tu$ and $f' = v' \circ {}^tu'$ where $u$, $v$, $u'$, $v'$ are the graphs of functions $u : C \longrightarrow A$, $v : C \longrightarrow B$, $u' : C' \longrightarrow A$ and $v' : C' \longrightarrow B$ with $C \subseteq C'$ (in fact $C = f$ and $C' = f'$) and such that $u = u'\big|_C$ and $v = v'\big|_C$, or equivalently $u = u' \circ \mathsf{id}^C$ and $v = v' \circ \mathsf{id}^C$. Since $T$ is type blind, $TC \subseteq TC'$ and we can write: $Tf = T(v \circ {}^tu) = Tv \circ T\,{}^tu = T(v' \circ \mathsf{id}^C) \circ T(\mathsf{id}^C \circ {}^tu') = Tv' \circ T\mathsf{id}^C \circ T\mathsf{id}^C \circ T\,{}^tu' \overset{(*)}{=} Tv' \circ \mathsf{id}^{TC} \circ T\,{}^tu' \subseteq Tv' \circ \mathsf{id}^{TC'} \circ T\,{}^tu' = Tv' \circ T\,{}^tu' = Tf'$. The crucial step is $(*)$, which refers to Lemma 2.3: $T^{C,C'}\mathsf{id}^C = T^{C',C}\mathsf{id}^C = \mathsf{id}^{TC}$.

Now assume $T$ is any relator. Then by the second item of Proposition 2.1, $T\left({}^tf\right) = {}^t(Tf)$ as soon as $f$ is the graph of a function. This extends to all relations by the first item of Proposition 2.1. □

In particular, every inclusion preserving symmetric functor is a relator since it is type blind.[1] Of course, not all functors are type blind (resp. relators, symmetric):

**Counter-example 2.5** *Let $P$ denote the functor of powersets and direct images: $PA = \mathfrak{P}(A)$ and $P^{A,B}f = \{(a, f \cdot a);\ a \subseteq A\}$. Notice that $P^{A,B}f$ is the graph of a function, which is not necessarily injective, hence $P$ is not symmetric. By the previous lemma, $P$ is neither type blind nor a relator.*

A functor is said to be *continuous on sets* if it preserves directed unions of sets: $T\bigcup \overrightarrow{A} = \bigcup \overrightarrow{TA}$ for all family of sets $\overrightarrow{A}$ which is directed for inclusion. Notice the use of our convention for denoting families: $\overrightarrow{TA} = (TA_i)_{i\in I}$. Similarly, we say $T$ is *continuous on relations* if it preserves directed unions of parallel relations: $T\bigcup \overrightarrow{f} = \bigcup \overrightarrow{Tf} \in \underline{\mathrm{Rel}}(TA, TB)$ for all family of relations $\overrightarrow{f} \in \underline{\mathrm{Rel}}(A, B)^I$ which is directed for inclusion.

**Lemma 2.6** *If $T$ is continuous on sets (resp. on relations) then it is monotonic on sets (resp. it is a relator). Moreover, if $T$ is type blind and continuous on relations then it is also continuous on sets.*

**Proof** That continuity implies monotonicity is standard. Assume $T$ is type blind and continuous on relations, and let $\overrightarrow{A}$ be a directed family of sets. Then $\mathsf{id}^{T\bigcup \overrightarrow{A}} = T\mathsf{id}^{\bigcup \overrightarrow{A}} = T\bigcup \overrightarrow{\mathsf{id}^{A}} = \bigcup \overrightarrow{T\mathsf{id}^{A}} = \bigcup \overrightarrow{\mathsf{id}^{TA}} = \mathsf{id}^{\bigcup \overrightarrow{TA}}$ hence $T\bigcup \overrightarrow{A} = \bigcup \overrightarrow{TA}$. □

Notice again our use of the arrow notation for families. This allows to keep our developments concise while remaining self-explanatory and unambiguous: here $\overrightarrow{A}$ is a family of sets, while $T$ is just one functor, hence, e.g., $\overrightarrow{\mathsf{id}^{TA}}$ can only mean $\left(\mathsf{id}^{TA_i}\right)_{i\in I}$. Although this needs some overhead effort to parse the first times, we are confident the reader will quickly become familiar with this convention: we will rely on its conciseness extensively in the remaining of the paper, always taking care that it does not introduce any ambiguity.

If $T$ is continuous on both sets and relations, then we simply say it is *continuous*. Of course, the identity functor is a type blind continuous functor. Another standard example is the multiset functor given by $!A = \mathfrak{M}_{\mathrm{f}}(A)$ and, for all relation $f$,

$$!f = \{([\alpha_1, \ldots, \alpha_n], [\beta_1, \ldots, \beta_n]);\ n \in \mathbf{N} \wedge \forall k,\ (\alpha_k, \beta_k) \in f\}.$$

When $a \subseteq A$, we write $a^! = \mathfrak{M}_{\mathrm{f}}(a) \subseteq !A$ (rather that $!a \subseteq !A$) in order to avoid confusion with the corresponding operation on relations.

Let $T$ and $U$ be two functors from $\underline{\mathrm{Rel}}$ to $\underline{\mathrm{Rel}}$, and let $f$ be the data of a relation $f^A$ from $TA$ to $UA$ for all set $A$: we say $f$ is a *lax natural transformation* from $T$ to $U$ if, for all relation $g$ from $A$ to $B$, $f^B \circ (Tg) \subseteq (Ug) \circ f^A$. We say $f$ is a *natural transformation* if this inclusion is always an equality. In general we omit the annotation and simply write $f$ for $f^A$ when $A$ is clear from the context. Of course, the identities $\mathsf{id}^{TA}$ constitute a natural transformation from each $T$ to itself. For all set $A$, consider the only relation $\mathsf{supp}$ from $!A$ to $A$ such that $\mathsf{supp} \cdot \overline{\alpha} = supp(\overline{\alpha})$ for all $\overline{\alpha} \in !A$. This defines a lax natural transformation from $!$ to the identity functor: notice that in that case, the inclusion $\mathsf{supp} \circ !g \subseteq g \circ \mathsf{supp}$ may be strict. Lax natural transformations between type blind functors enjoy some kind of stability property:

**Lemma 2.7** *Let $T$ and $U$ be type blind functors and let $f$ be a lax natural transformation from $T$ to $U$ Then, if $A \subseteq B$:*

- $f^A = f^B \circ \mathsf{id}^{TA}$;
- *for all $\widetilde{a} \subseteq TA$, $f^A \cdot \widetilde{a} = f^B \cdot \widetilde{a}$;*
- *for all $\widehat{b} \subseteq UB$, $f^A \setminus \widehat{b} = \left(f^B \setminus \widehat{b}\right) \cap TA$.*

[1] This fixes a flawed result by Bird and de Moor (1997, Lemma 5.1), which implicitly relies on every functor preserving inclusions.

**Proof** By applying the naturality condition to the identity $\mathsf{id}^A$ both as a relation from $A$ to $B$ and as a relation from $B$ to $A$, we obtain $f^B \circ \mathsf{id}^{TA} \subseteq \mathsf{id}^{UA} \circ f^A$ and $f^A \circ \mathsf{id}^{TA} \subseteq \mathsf{id}^{UA} \circ f^B$, hence $f^A = f^B \circ \mathsf{id}^{TA} = f^B \cap (TA \times UB)$, from which both other properties follow. □

We shall not restrict our study to unary functors, hence we need to generalize the above notions to families of relations indexed by a fixed set $I$. If $\overrightarrow{A}$ and $\overrightarrow{B}$ are families of sets, we call *relation from* $\overrightarrow{A}$ *to* $\overrightarrow{B}$ any family $\overrightarrow{f}$ of componentwise relations: $\overrightarrow{f \subseteq A \times B}$, i.e. for all $i \in I$, $f_i \subseteq A_i \times B_i$. We denote by $\underline{\mathrm{Rel}}^I$ the category of $I$-indexed families of sets and relations, with componentwise identities and composition: $\mathsf{id}^{\overrightarrow{A}} = \overrightarrow{\mathsf{id}^A}$ and $\overrightarrow{g} \circ \overrightarrow{f} = \overrightarrow{g \circ f}$. Again, we must precise that morphisms in $\underline{\mathrm{Rel}}^I(\overrightarrow{A}, \overrightarrow{B})$ are triples $(\overrightarrow{A}, \overrightarrow{B}, \overrightarrow{f})$ such that $\overrightarrow{f}$ is a relation from $\overrightarrow{A}$ to $\overrightarrow{B}$, although we simply write $\overrightarrow{f}$ for $(\overrightarrow{A}, \overrightarrow{B}, \overrightarrow{f})$ whenever $\overrightarrow{A}$ and $\overrightarrow{B}$ are clear from the context.

An $I$-ary functor in $\underline{\mathrm{Rel}}$ is a functor from $\underline{\mathrm{Rel}}^I$ to $\underline{\mathrm{Rel}}$, i.e. the data of a set $T\overrightarrow{A}$ for all $I$-indexed family $\overrightarrow{A}$ of sets, and of a relation $T^{\overrightarrow{A},\overrightarrow{B}}\overrightarrow{f}$ from $T\overrightarrow{A}$ to $T\overrightarrow{B}$ for all relation $\overrightarrow{f}$ from $\overrightarrow{A}$ to $\overrightarrow{B}$, preserving identities and composition: $T^{\overrightarrow{A},\overrightarrow{A}}\mathsf{id}^{\overrightarrow{A}} = \mathsf{id}^{T\overrightarrow{A}}$ and $T^{\overrightarrow{A},\overrightarrow{C}}\overrightarrow{g \circ f} = T^{\overrightarrow{B},\overrightarrow{C}}\overrightarrow{g} \circ T^{\overrightarrow{A},\overrightarrow{B}}\overrightarrow{f}$. Whenever $\overrightarrow{A}$ and $\overrightarrow{B}$ are clear from the context, we just write $T\overrightarrow{f}$ for both $T^{\overrightarrow{A},\overrightarrow{B}}\overrightarrow{f}$ and $T(\overrightarrow{A}, \overrightarrow{B}, \overrightarrow{f}) = (T\overrightarrow{A}, T\overrightarrow{B}, T^{\overrightarrow{A},\overrightarrow{B}}\overrightarrow{f})$.

Let $T$ be an $I$-ary functor. We say:

- $T$ is *type blind* if $T^{\overrightarrow{A},\overrightarrow{B}}\overrightarrow{f} = T^{\overrightarrow{A'},\overrightarrow{B'}}\overrightarrow{f}$ whenever both sides of the equation are defined;
- $T$ is *monotonic on sets* if $T\overrightarrow{A} \subseteq T\overrightarrow{B}$ for all families $\overrightarrow{A}$ and $\overrightarrow{B}$ of sets such that $\overrightarrow{A \subseteq B}$;
- $T$ is an *$I$-ary relator* if $T\overrightarrow{f} \subseteq T\overrightarrow{g}$ for all relations $\overrightarrow{f}$ and $\overrightarrow{g}$ from $\overrightarrow{A}$ to $\overrightarrow{B}$ such that $\overrightarrow{f \subseteq g}$;
- $T$ is *symmetric* if $T^{\overrightarrow{t}}\overrightarrow{f} = {}^t\left(T\overrightarrow{f}\right)$ for all family of relations $\overrightarrow{f}$.

Lemmas 2.3 and 2.4 extend to $I$-ary functors: $T$ is type blind iff $T$ preserves $I$-ary inclusions $\mathsf{id}^{\overrightarrow{A}}$, both from $\overrightarrow{A}$ to $\overrightarrow{B}$ and from $\overrightarrow{B}$ to $\overrightarrow{A}$, for all $\overrightarrow{A \subseteq B}$; every type blind functor is monotonic on sets and is a relator; every relator is symmetric.

In order to define the continuity of $I$-ary functors, we have to consider families of families. We thus introduce the following conventions: by $\overleftrightarrow{A}$, we denote an $I$-indexed family $\left(\overleftarrow{A}_i\right)_{i\in I}$ of families of sets, where each $\overleftarrow{A}_i = (A_{i,j})_{j\in J_i}$ takes indices in some variable set $J_i$. If $\overrightarrow{j \in J}$ (i.e. $j_i \in J_i$ for all $i \in I$), we also write $\overrightarrow{A}_{\overrightarrow{j}}$ for the $I$-indexed family $(A_{i,j_i})_{i\in I}$. We use leftwards arrows to distinguish families indexed by some variable set from $I$-indexed families. When the order of application of arrows is reversed, as in $\overleftrightarrow{A}$, the leftwards arrow stands for quantifying over all families $\overrightarrow{j} \in \prod \overrightarrow{J}$ of indices, i.e. $\overleftrightarrow{A} = \left(\overrightarrow{A}_{\overrightarrow{j}}\right)_{\overrightarrow{j\in J}}$.

We say $\overleftrightarrow{A}$ is *directed* if each $\overleftarrow{A}_i$ is directed for inclusion. The family $\overrightarrow{\bigcup \overleftrightarrow{A}} = \left(\bigcup \overleftarrow{A}_i\right)_{i\in I}$ is the componentwise union of $\overleftrightarrow{A}$. Then we say $T$ is *continuous on sets* if it commutes to directed unions: $T\overrightarrow{\bigcup \overleftrightarrow{A}} = \bigcup \overleftarrow{T\overrightarrow{A}}$ as soon as $\overleftrightarrow{A}$ is directed. Similarly, we say $T$ is *continuous on relations* if, for all directed family $\overleftrightarrow{f}$, with $f_{i,j} \subseteq A_i \times B_i$ for all $i \in I$ and $j \in J_i$, we have $T\overrightarrow{\bigcup \overleftrightarrow{f}} = \bigcup \overleftarrow{T\overrightarrow{f}}$. If both properties hold, we simply say $T$ is continuous. Again, Lemma 2.6 extends to $I$-ary functors: every type blind functor which is continuous on relations is continuous.

We denote by $\Pi_i$ the $i$-th *projection functor* from $\underline{\mathrm{Rel}}^I$ to $\underline{\mathrm{Rel}}$: for all family of sets $\overrightarrow{A}$, $\Pi_i\overrightarrow{A} = A_i$ and, for all relation $\overrightarrow{f}$ from $\overrightarrow{A}$ to $\overrightarrow{B}$, $\Pi_i\overrightarrow{f} = f_i$. Projection functors are continuous type blind relators. Other standard examples include: the *cartesian product* functor, given by $\bigotimes\overrightarrow{A} = \prod\overrightarrow{A}$ and $\bigotimes\overrightarrow{f} = \left\{\left(\overrightarrow{\alpha}, \overrightarrow{\beta}\right);\ \overrightarrow{(\alpha,\beta) \in f}\right\}$; and the *disjoint union* functor, given by $\bigoplus\overrightarrow{A} = \sum\overrightarrow{A}$ and

$\bigoplus \overrightarrow{f} = \{((i,\alpha),(i,\beta))\,;\ i \in I \wedge (\alpha,\beta) \in f_i\}$. Notice that $\bigoplus$ defines both products and coproducts in $\underline{\mathrm{Rel}}$: we may also write it $\&$ when we refer to it as the functor of products.

Let $T$ and $U$ be two functors from $\underline{\mathrm{Rel}}^I$ to $\underline{\mathrm{Rel}}$, and let $f$ be the data of a relation $f^{\overrightarrow{A}}$ from $T\overrightarrow{A}$ to $U\overrightarrow{A}$ for all $\overrightarrow{A}$: we say $f$ is a *lax natural transformation* from $T$ to $U$, if, for all relation $\overrightarrow{g}$ from $\overrightarrow{A}$ to $\overrightarrow{B}$, $f^{\overrightarrow{B}} \circ (T\overrightarrow{g}) \subseteq (U\overrightarrow{g}) \circ f^{\overrightarrow{A}}$. We say $f$ is a *natural transformation* if moreover this inclusion is always an equality. Again, the identities $\mathsf{id}^{T\overrightarrow{A}}$ define a natural transformation from $T$ to itself. Other basic examples of natural transformations are the following *projection*, *restriction* and *index* relations:

- for all $i \in I$, the projection from $\bigotimes \overrightarrow{A}$ to $A_i$ is $\mathsf{proj}_i = \left\{(\overrightarrow{\alpha},\alpha_i);\ \overrightarrow{\alpha} \in \bigotimes \overrightarrow{A}\right\}$;
- for all $i \in I$, the restriction from $\bigoplus \overrightarrow{A}$ to $A_i$ is $\mathsf{rest}_i = \{((i,\alpha),\alpha);\ \alpha \in A_i\}$;
- the index relation $\mathsf{indx}$ from $\bigoplus \overrightarrow{A}$ to $I$ is given by $\mathsf{indx} = \{((i,\alpha),i);\ i \in I \wedge \alpha \in A_i\}$.

Then: each $\mathsf{proj}_i$ is a natural transformation from $\bigotimes$ to $\Pi_i$; each $\mathsf{rest}_i$ is a natural transformation from $\bigoplus$ to $\Pi_i$; and $\mathsf{indx}$ is a natural transformation from $\bigoplus$ to $E_I$, which is the constant functor $E_I\overrightarrow{A} = I$ and $E_I\overrightarrow{f} = \mathsf{id}^I$. Again, Lemma 2.7 extends to $I$-ary type blind functors and lax natural transformations between them.

# 3 On the transport of finiteness structures

## 3.1 Finiteness spaces

Let $A$ and $B$ be sets, we write $A \perp_{\mathrm{f}} B$ if $A \cap B$ is finite. If $\mathfrak{A} \subseteq \mathfrak{P}(A)$, we define the *predual* of $\mathfrak{A}$ on $A$ as $\mathfrak{A}^{\perp_A} = \{a' \subseteq A;\ \forall a \in \mathfrak{A},\ a \perp_{\mathrm{f}} a'\}$. By standard arguments on closure operators and orthogonality constructions, we have the following properties:

- $\mathfrak{P}_{\mathrm{f}}(A) \subseteq \mathfrak{A}^{\perp_A}$;
- $\mathfrak{A} \subseteq \mathfrak{A}^{\perp_A\perp_A}$;
- if $\mathfrak{A} \subseteq \mathfrak{A}'$, then $\mathfrak{A}'^{\perp_A} \subseteq \mathfrak{A}^{\perp_A}$ and $\mathfrak{A}^{\perp_A\perp_A} \subseteq \mathfrak{A}'^{\perp_A\perp_A}$;
- by the previous two items, $\mathfrak{A}^{\perp_A} = \mathfrak{A}^{\perp_A\perp_A\perp_A}$;
- $\mathfrak{A}^{\perp_A}$ is downwards closed for inclusion, i.e. $a \subseteq a' \in \mathfrak{A}^{\perp_A}$ implies $a \in \mathfrak{A}^{\perp_A}$;
- $\mathfrak{A}^{\perp_A}$ is closed under finite unions, i.e. $a, a' \in \mathfrak{A}^{\perp_A}$ implies $a \cup a' \in \mathfrak{A}^{\perp_A}$;
- if $\mathfrak{A}_i^{\perp_A\perp_A} = \mathfrak{A}_i$ for all $i \in I$, then $\left(\bigcap \overrightarrow{\mathfrak{A}}\right)^{\perp_A\perp_A} = \bigcap \overrightarrow{\mathfrak{A}}$.

A *finiteness structure* on $A$ is a set $\mathfrak{A}$ of subsets of $A$ such that $\mathfrak{A}^{\perp_A\perp_A} = \mathfrak{A}$. Then a *finiteness space* is a pair $\mathcal{A} = (|\mathcal{A}|, \mathfrak{F}(\mathcal{A}))$ where $|\mathcal{A}|$ is the underlying set, called the *web* of $\mathcal{A}$, and $\mathfrak{F}(\mathcal{A})$ is a finiteness structure on $|\mathcal{A}|$. We write $\mathcal{A}^\perp$ for the *dual* finiteness space: $\left|\mathcal{A}^\perp\right| = |\mathcal{A}|$ and $\mathfrak{F}\left(\mathcal{A}^\perp\right) = \mathfrak{F}(\mathcal{A})^{\perp_{|\mathcal{A}|}}$. The elements of $\mathfrak{F}(\mathcal{A})$ are called the *finitary subsets* of $\mathcal{A}$.

For every set $A$, $(A, \mathfrak{P}_{\mathrm{f}}(A))$ is a finiteness space and $(A, \mathfrak{P}_{\mathrm{f}}(A))^\perp = (A, \mathfrak{P}(A))$. In particular, each finite set $A$ is the web of exactly one finiteness space: $(A, \mathfrak{P}_{\mathrm{f}}(A)) = (A, \mathfrak{P}(A))$. We introduce the empty finiteness space $\top$ with web $\emptyset$ and the singleton finiteness space $\mathbf{1}$ with web $\{\emptyset\}$. Having finite webs, $\top$ and $\mathbf{1}$ are identified with their respective duals: $\mathbf{0} = \top^\perp = \top$ and $\bot = \mathbf{1}^\perp = \mathbf{1}$. We moreover introduce the space of *flat natural numbers* $\mathcal{N} = (\mathbf{N}, \mathfrak{P}_{\mathrm{f}}(\mathbf{N}))$.

The following reformulation of bidual closure is given by Ehrhard (2005):

**Lemma 3.1** *If $\mathfrak{A} \subseteq \mathfrak{P}(A)$ is downwards closed for inclusion, then $a \in \mathfrak{A}^{\perp_A\perp_A}$ iff, for all infinite subset $a' \subseteq a$, there is an infinite subset $a'' \subseteq a'$ such that $a'' \in \mathfrak{A}$.*

In particular, the following *does not* define a finiteness structure:

**Counter-example 3.2 (Communicated to us by Laurent Regnier)** *We say* $t \subseteq \mathbf{N}$ *is* thin *if the sequence* $\left(\frac{\#\, t \cap \{0,\ldots,n-1\}}{n}\right)_{n\in\mathbf{N}}$ *converges to* $0$. *Let* $\mathfrak{T}$ *be the set of all thin subsets of* $\mathbf{N}$. *Examples of infinite thin subsets are* $\{n^2;\ n \in \mathbf{N}\}$ *and* $\{n^n;\ n \in \mathbf{N}\}$. *Of course,* $\mathbf{N}$ *itself is not thin. Notice that every infinite subset* $a \subseteq \mathbf{N}$ *contains an infinite thin subset: let* $(\alpha_n)_{n\in\mathbf{N}}$ *be the ordered sequence of the elements of* $a$*; then, for instance,* $\{\alpha_{n^2};\ n \in \mathbf{N}\} \in \mathfrak{T}$. *Notice moreover that* $\mathfrak{P}_{\mathrm{f}}(\mathbf{N}) \subseteq \mathfrak{T}$, *and that* $\mathfrak{T}$ *is downwards closed for inclusion and closed under finite unions. By the previous lemma,* $\mathbf{N} \in \mathfrak{T}^{\perp\perp_A}$ *and then* $\mathfrak{T}^{\perp\perp_{\mathbf{N}}} = \mathfrak{P}(\mathbf{N}) \neq \mathfrak{T}$.

All along the text, we provide relevant counter-examples in order to motivate the various notions we introduce, and also to emphasize the complex structure of finiteness spaces. These will often refer to a situation like the above one: we say $\mathfrak{A} \subseteq \mathfrak{P}(A)$ is a *fake finiteness structure* on $A$ if $\mathfrak{A}$ is downwards closed for inclusion, closed under finite unions, and contains $\mathfrak{P}_{\mathrm{f}}(A)$, but $\mathfrak{A} \neq \mathfrak{A}^{\perp\perp_A}$. Below we present another fake finiteness structure, the properties of which will be useful in some of our arguments.

**Counter-example 3.3** *For all* $n \in \mathbf{N}$, *write* $\dagger_n = \{(p,q);\ p = n \vee q = n\}$. *Then, for all* $n \in \mathbf{N}$, *write* $\mathfrak{C}_n = \{\dagger_p;\ p \geq n\}^{\perp_{\mathbf{N}\times\mathbf{N}}}$. *Being a dual set, each* $\mathfrak{C}_n$ *is a finiteness structure on* $\mathbf{N} \times \mathbf{N}$. *Moreover,* $\mathfrak{C}_n \subseteq \mathfrak{C}_{n'}$ *as soon as* $n \leq n'$. *As a consequence,* $\mathfrak{C} = \bigcup \overrightarrow{\mathfrak{C}}$ *is downwards closed for inclusion, closed under finite unions and contains all finite subsets, but not every subset. However,* $\mathfrak{C}^{\perp_{\mathbf{N}\times\mathbf{N}}} = \mathfrak{P}_{\mathrm{f}}(\mathbf{N} \times \mathbf{N})$ *whose dual is* $\mathfrak{P}(\mathbf{N} \times \mathbf{N})$.

## 3.2 Transport of finiteness structures

The following lemma will be used throughout the paper. It allows to transport a finiteness structure on set $B$, along any relation $f$ from $A$ to $B$, provided $f$ maps finite subsets of $A$ to finitary subsets of $B$.

**Lemma 3.4 (Transport)** *Let* $A$ *be a set,* $\mathcal{B}$ *a finiteness space and* $f$ *a relation from* $A$ *to* $|\mathcal{B}|$ *such that* $f \cdot \alpha \in \mathfrak{F}(\mathcal{B})$ *for all* $\alpha \in A$.[2] *Then* $\mathfrak{F}_{\mathcal{B},f} = \{a \subseteq A;\ f \cdot a \in \mathfrak{F}(\mathcal{B})\}$ *is a finiteness structure on* $A$ *and, more precisely,* $\mathfrak{F}_{\mathcal{B},f} = \{f \setminus b;\ b \in \mathfrak{F}(\mathcal{B})\}^{\perp\perp_A}$.

**Proof** Write $\mathfrak{A} = \{f \setminus b;\ b \in \mathfrak{F}(\mathcal{B})\}$. The first inclusion is easy: $\mathfrak{F}_{\mathcal{B},f} \subseteq \mathfrak{A}^{\perp\perp_A}$ because, for all $a \in \mathfrak{F}_{\mathcal{B},f}$ and $a' \in \mathfrak{A}^{\perp_A}$, $a \cap a'$ is finite. Indeed, $f \cdot a \in \mathfrak{F}(\mathcal{B})$ hence $a' \cap (f \setminus (f \cdot a))$ is finite; moreover $a \subseteq f \setminus (f \cdot a)$.

We now prove the reverse inclusion: let $a \in \mathfrak{A}^{\perp\perp_A}$, we establish that $a \in \mathfrak{F}_{\mathcal{B},f}$, i.e. $f \cdot a \in \mathfrak{F}(\mathcal{B})$. It is sufficient to show that, for all $b' \in \mathfrak{F}(\mathcal{B}^{\perp})$, $b'' = (f \cdot a) \cap b'$ is finite. Since $b'' \subseteq f \cdot a$, for all $\beta \in b''$ there is $\alpha \in a$ such that $\beta \in f \cdot \alpha$: by the axiom of choice, we obtain a function $\phi : b'' \longrightarrow a$ such that $\beta \in f \cdot \phi(\beta)$ for all $\beta \in b''$, which entails $b'' \subseteq f \cdot \phi(b'')$. Now it is sufficient to show that $\phi(b'')$ is finite. Indeed, in that case, $f \cdot \phi(b'') = \bigcup_{\alpha\in\phi(b'')} f \cdot \alpha$ is a finite union of finitary subsets of $\mathcal{B}$: recall that by our hypothesis on $f$, $f \cdot \alpha \in \mathfrak{F}(\mathcal{B})$ for all $\alpha \in A$. Hence $b'' \in \mathfrak{F}(\mathcal{B})$ and, since we also have $b'' \subseteq b' \in \mathfrak{F}(\mathcal{B}^{\perp})$, $b''$ is finite.

Since $\phi(b'') \subseteq a \in \mathfrak{A}^{\perp\perp_A}$, it will be sufficient to prove that $\phi(b'') \in \mathfrak{A}^{\perp_A}$ also. For that purpose, we consider $b \in \mathfrak{F}(\mathcal{B})$ and prove that $a'' = \phi(b'') \cap f \setminus b$ is finite. If $\alpha \in a''$, there exists $\beta \in b''$ such that $\alpha = \phi(\beta)$ and moreover $f \cdot \alpha \subseteq b$; since $\beta \in f \cdot \phi(\beta) = f \cdot \alpha$, we obtain that $\beta \in b'' \cap b$. Hence $a'' \subseteq \phi(b'' \cap b)$, which is finite because $\phi$ is a function and $b'' \cap b \subseteq b' \cap b$ is finite as $b' \in \mathfrak{F}(\mathcal{B}^{\perp})$ and $b \in \mathfrak{F}(\mathcal{B})$. □

[2] Following the terminology of Hyland and Schalk (2003), this condition can be rephrased as $f$ being negative from $(A, \mathfrak{P}(A), \mathfrak{P}_{\mathrm{f}}(A))$ to $(|\mathcal{B}|, \mathfrak{F}(\mathcal{B}), \mathfrak{F}(\mathcal{B}^{\perp}))$, i.e. for all $a \subseteq A$ and $b' \in \mathfrak{F}(\mathcal{B}^{\perp})$, $a \perp_{\mathrm{f}} {}^t f \cdot b'$ implies $f \cdot a \perp b'$. It is however unclear, at the time of writing, under which hypotheses the transport lemma could be recast in this more general setting.

The reader should remark that the structure of this proof is very similar to that of the characterization of the exponential modality, given by Ehrhard (2005, Lemma 4). Actually, we obtain this characterization as a straightforward application of transport:

**Example 3.5** *Let $\mathcal{A} = (A, \mathfrak{A})$ be a finiteness space, and recall that $\mathsf{supp}^A$ is the only relation from $!A$ to $A$ such that $\mathsf{supp}^A \cdot \overline{\alpha} = supp(\overline{\alpha})$ for all $\overline{\alpha} \in !A$. Notice in particular that $supp(\overline{\alpha}) \in \mathfrak{P}_{\mathrm{f}}(A) \subseteq \mathfrak{A}$. By the transport lemma, $(!A, \mathfrak{F}_{\mathcal{A},\mathsf{supp}^A})$ is a finiteness space that we denote by $!\mathcal{A}$. We moreover have that $\mathsf{supp}^A \setminus a = \mathfrak{M}_{\mathrm{f}}(a) = a^!$, and we obtain:*

$$\mathfrak{F}(!\mathcal{A}) = \left\{\overline{a} \subseteq !|\mathcal{A}|;\ \mathsf{supp}^{|\mathcal{A}|} \cdot \overline{a} \in \mathfrak{F}(\mathcal{A})\right\} = \left\{a^!;\ a \in \mathfrak{F}(\mathcal{A})\right\}^{\perp\perp_{|\mathcal{A}|}}.$$

The transport lemma is easily generalized to families of finiteness structures. If we write $\overrightarrow{f} \setminus \overrightarrow{b}$ for $\bigcap \overrightarrow{f \setminus b} = \bigcap_{i\in I}(f_i \setminus b_i)$, we obtain:

**Corollary 3.6** *Let $A$ be a set, $\overrightarrow{\mathcal{B}}$ a family of finiteness spaces and $\overrightarrow{f}$ a family of relations such that, for all $\alpha \in A$ and all $i \in I$, $f_i \cdot \alpha \in \mathfrak{F}(\mathcal{B}_i)$. Then $\mathfrak{F}_{\overrightarrow{\mathcal{B}},\overrightarrow{f}} = \{a \subseteq A;\ \forall i \in I,\ f_i \cdot a \in \mathfrak{F}(\mathcal{B}_i)\}$ is a finiteness structure on $A$ and, more precisely, $\mathfrak{F}_{\overrightarrow{\mathcal{B}},\overrightarrow{f}} = \left\{\overrightarrow{f} \setminus \overrightarrow{b};\ \overrightarrow{b \in \mathfrak{F}(\mathcal{B})}\right\}^{\perp\perp_A}$.*

**Proof** By Lemma 3.4, each $\mathfrak{F}_{\mathcal{B}_i,f_i}$ is a finiteness structure on $A$. As bidual closure commutes to intersections of finiteness structures, $\mathfrak{F}_{\overrightarrow{\mathcal{B}},\overrightarrow{f}} = \bigcap_{i\in I}\mathfrak{F}_{\mathcal{B}_i,f_i}$ is a finiteness structure. Let us prove that $\mathfrak{F}_{\overrightarrow{\mathcal{B}},\overrightarrow{f}} = \left\{\bigcap_{i\in I}(f_i \setminus b_i);\ \overrightarrow{b \in \mathfrak{F}(\mathcal{B})}\right\}^{\perp\perp_A}$. Let $a \in \mathfrak{F}_{\overrightarrow{\mathcal{B}},\overrightarrow{f}}$: for all $i \in I$, $a \in \mathfrak{F}_{\mathcal{B}_i,f_i}$, hence setting $b_i = f_i \cdot a$ we obtain $b_i \in \mathfrak{F}(\mathcal{A}_i)$ and $a \subseteq f_i \setminus b_i$. We have thus found $\overrightarrow{b \in \mathfrak{F}(\mathcal{B})}$ such that $a \subseteq \bigcap_{i\in I}(f_i \setminus b_i)$, which proves one inclusion. For the reverse, let $\overrightarrow{b \in \mathfrak{F}(\mathcal{B})}$: for all $j \in I$, $\bigcap_{i\in I}(f_i \setminus b_i) \subseteq f_j \setminus b_j$. Now, observe that $f_j \setminus b_j \in \mathfrak{F}_{\mathcal{B}_j,f_j}$ which is downwards closed for inclusion, hence $\bigcap_{i\in I}(f_i \setminus b_i) \in \mathfrak{F}_{\mathcal{B}_j,f_j}$. We have just proved that $\left\{\overrightarrow{f} \setminus \overrightarrow{b};\ \overrightarrow{b \in \mathfrak{F}(\mathcal{B})}\right\} \subseteq \mathfrak{F}_{\overrightarrow{\mathcal{B}},\overrightarrow{f}}$, and we conclude since bidual closure is monotonic and idempotent. □

**Example 3.7** *For all family $\overrightarrow{\mathcal{A}}$ of finiteness spaces, we denote by $\bigotimes \overrightarrow{\mathcal{A}}$ the finiteness space $\left(\prod \overrightarrow{|\mathcal{A}|}, \mathfrak{F}_{\overrightarrow{\mathcal{A}},\overrightarrow{\mathsf{proj}}}\right)$: for all $\widetilde{a} \subseteq \prod \overrightarrow{|\mathcal{A}|}$, $\widetilde{a} \in \mathfrak{F}\left(\bigotimes \overrightarrow{\mathcal{A}}\right)$ iff $\mathsf{proj}_i \cdot \widetilde{a} \in \mathfrak{F}(\mathcal{A}_i)$ for all $i \in I$. We moreover obtain $\mathfrak{F}\left(\bigotimes \overrightarrow{\mathcal{A}}\right) = \left\{\prod \overrightarrow{a};\ \overrightarrow{a \in \mathfrak{F}(\mathcal{A})}\right\}^{\perp\perp_{\prod|\overrightarrow{\mathcal{A}}|}}$.*

*Similarly, let $\mathop{\&} \overrightarrow{\mathcal{A}}$ be the finiteness space $\left(\sum \overrightarrow{|\mathcal{A}|}, \mathfrak{F}_{\overrightarrow{\mathcal{A}},\overrightarrow{\mathsf{rest}}}\right)$: for all $\widetilde{a} \subseteq \sum \overrightarrow{|\mathcal{A}|}$, $\widetilde{a} \in \mathfrak{F}\left(\mathop{\&} \overrightarrow{\mathcal{A}}\right)$ iff $\mathsf{rest}_i \cdot \widetilde{a} \in \mathfrak{F}(\mathcal{A}_i)$ for all $i \in I$. Notice that this implies $\mathfrak{F}\left(\mathop{\&} \overrightarrow{\mathcal{A}}\right) = \left\{\sum \overrightarrow{a};\ \overrightarrow{a \in \mathfrak{F}(\mathcal{A})}\right\}$, hence the bidual closure is optional in that case.*

*Finally, we define the finiteness space $\bigoplus \overrightarrow{\mathcal{A}} = \left(\sum \overrightarrow{|\mathcal{A}|}, \mathfrak{F}_{(\overrightarrow{\mathcal{A}},\mathcal{I}),(\overrightarrow{\mathsf{rest}},\mathsf{indx})}\right)$ where $\mathcal{I} = (I, \mathfrak{P}_{\mathrm{f}}(I))$: $\widetilde{a} \in \mathfrak{F}\left(\bigoplus \overrightarrow{\mathcal{A}}\right)$ iff $\mathsf{indx} \cdot \widetilde{a}$ is finite and $\mathsf{rest}_i \cdot \widetilde{a} \in \mathfrak{F}(\mathcal{A}_i)$ for all $i \in I$. We obtain $\mathfrak{F}\left(\bigoplus \overrightarrow{\mathcal{A}}\right) = \left\{\sum_{i\in J} a_i;\ J \subseteq_{\mathrm{f}} I \wedge \forall i \in J,\ a \in \mathfrak{F}(\mathcal{A}_i)\right\}$, the bidual closure being optional. We have $\left(\bigoplus \overrightarrow{\mathcal{A}}\right)^\perp = \mathop{\&} \overrightarrow{\mathcal{A}^\perp}$, and moreover $\bigoplus \overrightarrow{\mathcal{A}} = \mathop{\&} \overrightarrow{\mathcal{A}}$ when $I$ is finite.*

Finally we introduce two other constructions on finiteness spaces which are not directly obtained by transport. If $\overrightarrow{\mathcal{A}}$ is a family of finiteness spaces, we set $\parr \overrightarrow{\mathcal{A}} = \left(\bigotimes \overrightarrow{\mathcal{A}^\perp}\right)^\perp$. From this, we derive $\mathcal{A} \multimap \mathcal{B} = \mathcal{A}^\perp \parr \mathcal{B} = \left(\mathcal{A} \otimes \mathcal{B}^\perp\right)^\perp$ for all finiteness spaces $\mathcal{A}$ and $\mathcal{B}$.

## 3.3 Finitary relations

Let $\mathcal{A}$ and $\mathcal{B}$ be two finiteness spaces: we say a relation $f$ from $|\mathcal{A}|$ to $|\mathcal{B}|$ is *finitary* from $\mathcal{A}$ to $\mathcal{B}$ if: for all $a \in \mathfrak{F}(\mathcal{A})$, $f \cdot a \in \mathfrak{F}(\mathcal{B})$, and for all $b' \in \mathfrak{F}\left(\mathcal{B}^\perp\right)$, ${}^t f \cdot b' \in \mathfrak{F}\left(\mathcal{A}^\perp\right)$. The following characterization of finitary relations is given by Ehrhard (2005, Section 1.1):

**Lemma 3.8** *Let $f \subseteq |\mathcal{A}| \times |\mathcal{B}|$. The following propositions are equivalent:*

*(a) $f$ is finitary from $\mathcal{A}$ to $\mathcal{B}$;*

*(b) ${}^t f$ is finitary from $\mathcal{B}^\perp$ to $\mathcal{A}^\perp$;*

*(c) for all $a \in \mathfrak{F}(\mathcal{A})$, $f \cdot a \in \mathfrak{F}(\mathcal{B})$ and, for all $\beta \in |\mathcal{B}|$, ${}^t f \cdot \beta \in \mathfrak{F}(\mathcal{A}^\perp)$;*

*(d) $f \in \mathfrak{F}(\mathcal{A} \multimap \mathcal{B})$.*

Notice that the identity relation $\mathsf{id}^{|\mathcal{A}|}$ is finitary from $\mathcal{A}$ to itself, and that finitary relations compose: we thus introduce the category $\underline{\mathrm{Fin}}$ whose objects are finiteness spaces and morphisms are finitary relations. Again, although we should precise that a morphism in $\underline{\mathrm{Fin}}(\mathcal{A}, \mathcal{B})$ is a triple $(\mathcal{A}, \mathcal{B}, f)$ such that $f$ is a finitary relation from $\mathcal{A}$ to $\mathcal{B}$, we will in general identify $\underline{\mathrm{Fin}}(\mathcal{A}, \mathcal{B})$ with $\mathfrak{F}(\mathcal{A} \multimap \mathcal{B})$. Functors in $\underline{\mathrm{Fin}}$ are defined similarly to those in $\underline{\mathrm{Rel}}$: a functor $\mathcal{T}$ is the data of a finiteness space $\mathcal{TA}$ for all finiteness space $\mathcal{A}$ and of a finitary relation $\mathcal{T}^{\mathcal{A},\mathcal{B}} f$ from $\mathcal{TA}$ to $\mathcal{TB}$ for all $f \in \underline{\mathrm{Fin}}(\mathcal{A}, \mathcal{B})$, preserving identities and composition.

Some functors in $\underline{\mathrm{Fin}}$ give rise to functors in $\underline{\mathrm{Rel}}$: we say a functor $\mathcal{T}$ in $\underline{\mathrm{Fin}}$ *has a web* if there exists a functor $T$ in $\underline{\mathrm{Rel}}$, such that $|\mathcal{TA}| = T|\mathcal{A}|$ for all finiteness space $\mathcal{A}$, and $\mathcal{T}^{\mathcal{A},\mathcal{B}} f = T^{|\mathcal{A}|,|\mathcal{B}|} f$ for all $f \in \underline{\mathrm{Fin}}(\mathcal{A}, \mathcal{B})$. We then say $T$ is the web of $\mathcal{T}$ and write $T = |\mathcal{T}|$. Notice that in that case, $T^{A,B} f$ must be finitary from $\mathcal{T}(A, \mathfrak{A})$ to $\mathcal{T}(B, \mathfrak{B})$ whenever $f$ is finitary from $(A, \mathfrak{A})$ to $(B, \mathfrak{B})$. We say $\mathcal{T}$ is type blind if $\mathcal{T}^{\mathcal{A},\mathcal{B}} f = \mathcal{T}^{\mathcal{A}',\mathcal{B}'} f$ whenever both sides of the equation are defined, i.e. $f \in \mathfrak{F}(\mathcal{A} \multimap \mathcal{B}) \cap \mathfrak{F}(\mathcal{A}' \multimap \mathcal{B}')$. Clearly, if $\mathcal{T}$ has a web, then $\mathcal{T}$ is type blind iff $|\mathcal{T}|$ is type blind. Of course, not all functors in $\underline{\mathrm{Fin}}$ have a web:

**Counter-example 3.9** *Let $\mathcal{F}$ denote the functor of finiteness structures and direct images: $\mathcal{FA} = (\mathfrak{F}(\mathcal{A}), \mathfrak{P}_{\mathrm{f}}(\mathfrak{F}(\mathcal{A})))$ and $\mathcal{F}^{\mathcal{A},\mathcal{B}} f = \{(a, f \cdot a);\ a \in \mathfrak{F}(\mathcal{A})\}$. The functoriality of $\mathcal{F}$ is clear as soon as we show that $\mathcal{F}^{\mathcal{A},\mathcal{B}} f \in \mathfrak{F}(\mathcal{FA} \multimap \mathcal{FB})$ when $f \in \mathfrak{F}(\mathcal{A} \multimap \mathcal{B})$. First, $\mathcal{F}^{\mathcal{A},\mathcal{B}} f \subseteq \mathfrak{F}(\mathcal{A}) \times \mathfrak{F}(\mathcal{B})$, since $f$ is finitary. Moreover, $\mathcal{F}^{\mathcal{A},\mathcal{B}} f$ is the graph of a function, hence it sends $\mathfrak{P}_{\mathrm{f}}(\mathfrak{F}(\mathcal{A}))$ to $\mathfrak{P}_{\mathrm{f}}(\mathfrak{F}(\mathcal{B}))$. That ${}^t(\mathcal{F}^{\mathcal{A},\mathcal{B}} f)$ sends $\mathfrak{F}(\mathcal{FB})^\perp$ to $\mathfrak{F}(\mathcal{FA})^\perp$ is automatic since $\mathfrak{F}(\mathcal{FA})^\perp = \mathfrak{P}(\mathfrak{F}(\mathcal{A}))$.*

The definition of $\underline{\mathrm{Fin}}^I$ and of $I$-ary functors in $\underline{\mathrm{Fin}}$ is straightforward, and matches exactly that of $\underline{\mathrm{Rel}}^I$ from $\underline{\mathrm{Rel}}$. Order relations on finiteness spaces, together with associated notions of monotonicity and continuity, will be discussed thoroughly in Section 4. Let us just remark that $\underline{\mathrm{Fin}}$ is not cpo-enriched. Indeed, if $f \subseteq |\mathcal{A}| \times |\mathcal{B}|$ is not finitary from $\mathcal{A}$ to $\mathcal{B}$, then $\mathfrak{P}_{\mathrm{f}}(f)$ is a set of finitary relations but it has no finitary upper bound.

On a side note, remark that the construction of a finiteness space by the transport lemma is not initial, in the sense that the relation $f$ from $A$ to $\mathcal{B}$ through which we transport the finiteness structure of $\mathcal{B}$ is not finitary from $(A, \mathfrak{F}_{\mathcal{B},f})$ to $\mathcal{B}$ in general: although the condition — $f$ sends every element of $A$ to a finitary subset of $\mathcal{B}$ — is necessary for $f$ to be finitary, it is not sufficient.

**Counter-example 3.10** *The relation $\mathsf{supp}^{|\mathcal{A}|}$ from $|!\mathcal{A}|$ to $|\mathcal{A}|$ is not finitary from $!\mathcal{A}$ to $\mathcal{A}$ whenever $|\mathcal{A}|$ is non-empty: let $\alpha \in |\mathcal{A}|$, then ${}^t\mathsf{supp}^{|\mathcal{A}|} \cdot \alpha \supseteq \{\alpha\}^! \setminus \{[]\} \in \mathfrak{F}(!\mathcal{A})$ which is an infinite finitary subset, hence ${}^t\mathsf{supp}^{|\mathcal{A}|} \cdot \alpha \notin \mathfrak{F}(!\mathcal{A})^{\perp_{|!\mathcal{A}|}}$; we conclude by Lemma 3.8.*

The following section explains how functors in $\underline{\mathrm{Fin}}$ may be derived from functors in $\underline{\mathrm{Rel}}$ *via* the transport lemma.

## 3.4 Transport functors

Let $I$ be a fixed set of indexes. Let $T$ be a functor from $\underline{\mathrm{Rel}}^I$ to $\underline{\mathrm{Rel}}$. We call *ownership relation* on $T$ the data of a quasi-functional lax natural transformation $\mathsf{own}_i$ from $T$ to the projection functor $\Pi_i$, for all $i \in I$. Notice that any ownership relation on $T$ satisfies the hypotheses of Corollary 3.6. Indeed, for any family $\overrightarrow{\mathcal{A}}$ of finiteness spaces, we have $\mathsf{own}_i^{|\overrightarrow{\mathcal{A}}|} \in \underline{\mathrm{Rel}}(T|\overrightarrow{\mathcal{A}}|, |\mathcal{A}_i|)$ and, since $\mathsf{own}_i^{|\overrightarrow{\mathcal{A}}|}$ is quasi-functional, $\mathsf{own}_i^{|\overrightarrow{\mathcal{A}}|} \cdot \widetilde{\alpha}$ is finite for all $\widetilde{\alpha} \in T|\overrightarrow{\mathcal{A}}|$, hence it is finitary in $\mathcal{A}_i$. Therefore, $\mathfrak{F}_{\overrightarrow{\mathcal{A}},\overrightarrow{\mathsf{own}}}$

is always a finiteness structure on $T|\overrightarrow{\mathcal{A}}|$. We call *transport situation* the data of a functor $T$ and an ownership relation $\overrightarrow{\mathsf{own}}$ on $T$. In such a situation, for all family $\overrightarrow{\mathcal{A}}$ of finiteness spaces, we write $T_{\overrightarrow{\mathsf{own}}}\overrightarrow{\mathcal{A}}$ for the finiteness space $\left(T|\overrightarrow{\mathcal{A}}|, \mathfrak{F}_{\overrightarrow{\mathcal{A}},\overrightarrow{\mathsf{own}}}\right)$ and, for all finitary relation $\overrightarrow{f}$ from $\overrightarrow{\mathcal{A}}$ to $\overrightarrow{\mathcal{B}}$, we write $T_{\overrightarrow{\mathsf{own}}}\overrightarrow{f} = T\overrightarrow{f}$. Notice that $T_{\overrightarrow{\mathsf{own}}}$ defines a functor from $\underline{\mathrm{Fin}}^I$ to $\underline{\mathrm{Fin}}$ iff $T\overrightarrow{f}$ is finitary from $T_{\overrightarrow{\mathsf{own}}}\overrightarrow{\mathcal{A}}$ to $T_{\overrightarrow{\mathsf{own}}}\overrightarrow{\mathcal{B}}$ as soon as $\overrightarrow{f}$ is finitary from $\overrightarrow{\mathcal{A}}$ to $\overrightarrow{\mathcal{B}}$. In that case, we say $T_{\overrightarrow{\mathsf{own}}}$ is the *transport functor* deduced from the transport situation $(T, \overrightarrow{\mathsf{own}})$.

We now provide sufficient conditions for a transport situation to give rise to a transport functor. A *shape relation* on $(T, \overrightarrow{\mathsf{own}})$ is the data of a fixed set $S$ of *shapes* and a quasi-functional lax natural transformation $\mathsf{shp}$ from $T$ to the constant functor $E_S$ which sends every set to $S$ and every relation to $\mathsf{id}^S$, subject to the following additional condition: for all $\widetilde{a} \subseteq T\overrightarrow{A}$, if $\mathsf{shp} \cdot \widetilde{a}$ is finite and, for all $i \in I$, $\mathsf{own}_i \cdot \widetilde{a}$ is finite, then $\widetilde{a}$ is itself finite.

In other words, with every $T$-element $\widetilde{\alpha} \in T\overrightarrow{A}$ is associated a set of shapes $\mathsf{shp}\cdot\widetilde{\alpha}$, which is finite (because $\mathsf{shp}$ is quasi-functional). Moreover shapes are preserved by $T$-relations; more precisely, if $(\widetilde{\alpha}, \widetilde{\beta}) \in T\overrightarrow{f}$ then every shape of $\widetilde{\beta}$ is a shape of $\widetilde{\alpha}$ (because $\mathsf{shp}$ is a lax natural transformation). Notice that when $T$ is symmetric, ${}^tT\overrightarrow{f} = T{}^t\overrightarrow{f}$, and we actually obtain $\mathsf{shp} \cdot \widetilde{\alpha} = \mathsf{shp} \cdot \widetilde{\beta}$. The additional condition states that any $T$-subset $\widetilde{a} \subseteq T\overrightarrow{A}$ which involves finitely many shapes and has a finite support in each component is itself finite.

**Lemma 3.11** *A transport situation on a symmetric functor defines a transport functor as soon as it admits a shape relation.*

**Proof** Let $(T, \overrightarrow{\mathsf{own}})$ be a transport situation with $T$ a symmetric functor, and let $\mathsf{shp}$ be a shape relation for this situation. By the above discussion on transport situations, we only have to prove that $\widetilde{f} = T\overrightarrow{f}$ is a finitary relation from $T_{\overrightarrow{\mathsf{own}}}\overrightarrow{\mathcal{A}}$ to $T_{\overrightarrow{\mathsf{own}}}\overrightarrow{\mathcal{B}}$ as soon as, for all $i \in I$, $f_i$ is a finitary relation from $\mathcal{A}_i$ to $\mathcal{B}_i$.

First, let us show that if $\widetilde{a} \in \mathfrak{F}\left(T_{\overrightarrow{\mathsf{own}}}\overrightarrow{\mathcal{A}}\right)$, then $\widetilde{f} \cdot \widetilde{a} \in \mathfrak{F}\left(T_{\overrightarrow{\mathsf{own}}}\overrightarrow{\mathcal{B}}\right)$. Indeed, for all $i \in I$, $\mathsf{own}_i \cdot \widetilde{f} \cdot \widetilde{a} \subseteq f_i \cdot \mathsf{own}_i \cdot \widetilde{a}$ because $\mathsf{own}_i$ is a lax natural transformation from $T$ to $\Pi_i$. Moreover, by the definition of $\mathfrak{F}\left(T_{\overrightarrow{\mathsf{own}}}\overrightarrow{\mathcal{A}}\right)$, $\mathsf{own}_i \cdot \widetilde{a} \in \mathfrak{F}(\mathcal{A}_i)$ and then $f_i \cdot \mathsf{own}_i \cdot \widetilde{a} \in \mathfrak{F}(\mathcal{B}_i)$, because $f_i$ is a finitary relation.

We are left to prove that for all $\widetilde{\beta} \in \left|T_{\overrightarrow{\mathsf{own}}}\overrightarrow{\mathcal{B}}\right|$, $\widetilde{a}' = {}^t\widetilde{f} \cdot \widetilde{\beta} \in \mathfrak{F}\left(\left(T_{\overrightarrow{\mathsf{own}}}\overrightarrow{\mathcal{A}}\right)^{\perp}\right)$, i.e. for all $\widetilde{a} \in \mathfrak{F}\left(T_{\overrightarrow{\mathsf{own}}}\overrightarrow{\mathcal{A}}\right)$, $\widetilde{a} \cap \widetilde{a}'$ is finite. By the properties of shape relations, it is sufficient to prove that $\mathsf{shp} \cdot (\widetilde{a} \cap \widetilde{a}')$ is finite and, for all $i \in I$, $\mathsf{own}_i \cdot (\widetilde{a} \cap \widetilde{a}')$ is finite. Notice that $T$ being symmetric, we have ${}^t\widetilde{f} = T{}^t\overrightarrow{f}$. Then, since $\mathsf{shp}$ is a lax natural transformation, $\mathsf{shp} \circ {}^t\widetilde{f} \subseteq \mathsf{shp}$. We obtain that $\mathsf{shp} \cdot (\widetilde{a} \cap \widetilde{a}') \subseteq \mathsf{shp} \cdot \widetilde{a}' = \mathsf{shp} \cdot {}^t\widetilde{f} \cdot \widetilde{\beta} \subseteq \mathsf{shp} \cdot \widetilde{\beta}$ which is finite, since $\mathsf{shp}$ is quasi-functional. Similarly, for all $i \in I$, $\mathsf{own}_i$ is a lax natural transformation from $T$ to $\Pi_i$, hence $\mathsf{own}_i \circ {}^t\widetilde{f} \subseteq {}^tf_i \circ \mathsf{own}_i$: we obtain $\mathsf{own}_i \cdot \widetilde{a}' \subseteq {}^tf_i \cdot \mathsf{own}_i \cdot \widetilde{\beta}$. Since $\mathsf{own}_i$ is quasi-functional $\mathsf{own}_i \cdot \widetilde{\beta}$ is finite and in particular $\mathsf{own}_i \cdot \widetilde{\beta} \in \mathfrak{F}\left(\mathcal{B}_i^{\perp}\right)$: $f_i$ being a finitary relation, we obtain that ${}^tf_i \cdot \mathsf{own}_i \cdot \widetilde{\beta} \in \mathfrak{F}\left(\mathcal{A}_i^{\perp}\right)$, and thus $\mathsf{own}_i \cdot \widetilde{a}' \in \mathfrak{F}\left(\mathcal{A}_i^{\perp}\right)$. By the definition of $\mathfrak{F}\left(T_{\overrightarrow{\mathsf{own}}}\overrightarrow{\mathcal{A}}\right)$, we also have $\mathsf{own}_i \cdot \widetilde{a} \in \mathfrak{F}(\mathcal{A}_i)$, and we conclude that $\mathsf{own}_i \cdot (\widetilde{a} \cap \widetilde{a}') \subseteq (\mathsf{own}_i \cdot \widetilde{a}) \cap (\mathsf{own}_i \cdot \widetilde{a}')$ is finite. □

We do not claim the hypotheses of Lemma 3.11 are minimal. Notice however that the symmetry of $T$ is essential in the proof, since it allows $\overrightarrow{\mathsf{own}}$ to control the behaviour of ${}^tT\overrightarrow{f}$ as well as of $T\overrightarrow{f}$. Moreover, the existence of a shape relation is crucial, since some transport situations with symmetric functor do not preserve finitary relations:

**Counter-example 3.12** *Consider the symmetric functor $S$ of $\mathbf{N}$-indexed sequences: for all set $A$, $SA = A^{\mathbf{N}}$ and, for all relation $f \subseteq A \times B$, $Sf = \left\{(\overrightarrow{\alpha}, \overrightarrow{\beta});\ \forall n \in \mathbf{N},\ (\alpha_n, \beta_n) \in f\right\}$. The projections $\mathsf{proj}_n = \left\{(\overrightarrow{\alpha}, \alpha_n);\ \overrightarrow{\alpha} \in A^{\mathbf{N}}\right\}$ define an ownership relation $\overrightarrow{\mathsf{proj}}$ on $S$. Now consider*

*the unique finiteness space* $2$ *with web* $\{0,1\}$. *Then* $S\,|2| = \{0,1\}^{\mathbf{N}}$ *and*

$$\mathfrak{F}_{2,\overrightarrow{s}} = \{\widetilde{a};\ \forall n \in \mathbf{N}, \mathsf{proj}_n \cdot \widetilde{\alpha} \in \mathfrak{F}\,(2)\} = \mathfrak{P}\,(S\,|2|)\,;$$

*in particular* $\mathfrak{F}^{\perp}_{2,s} = \mathfrak{P}_{\mathrm{f}}\,(S\,|2|)$. *Now let* $f = \{(0,0),(1,0)\}$ *which is a finitary relation from* $2$ *to* $2$: $Sf$ *is not finitary because* ${}^{t}Sf \cdot (0)_{n\in\mathbf{N}} = S\,|2|$ *which is infinite.*

**Example 3.13** *The transport functor* ! *in* Fin *is derived from the transport situation* $(!, \mathsf{supp})$, *with shape relation* $\mathsf{card}^A = \{(\overline{\alpha}, \#\,\overline{\alpha});\ \overline{\alpha} \in\, !A\}$. *The* $I$*-ary transport functor* $\&$ *(resp.* $\bigoplus$*) in* Fin *is derived from the transport situation* $\left(\bigoplus, \overrightarrow{\mathsf{rest}}\right)$ *(resp.* $\left(\bigoplus, \overrightarrow{\mathsf{rest}}, \mathsf{indx}\right)$*) with shape relation* $\mathsf{indx}$ *(resp.* $\emptyset$*). Finally, we only consider finite tensor products: the binary functor* $\otimes$ *in* Fin *is derived from the transport situation* $(\otimes, \mathsf{proj}_1, \mathsf{proj}_2)$ *with empty shape relation. Indeed, infinitary tensor products do not define functors: the functor* $S$ *in the above counter-example is an instance of* $\bigotimes$ *with* $I = \mathbf{N}$.

# 4 Continuity and fixpoints

In the classical setting of Scott domains and more precisely of complete partial orders, continuity is the key property for a function to have a fixpoint, see for instance Amadio and Curien (1998). It is well known that Rel endowed with the inclusion order is a complete partial order, and that Rel is cpo-enriched. Then the continuity of an endofunctor on Rel boils down to the commutation of the functor with directed unions of both sets and relations. Moreover the fixpoint of any $(n+1)$-ary continuous functor exists and is an $n$-ary functor in Rel.

The situation in Fin is more complex, if only because the order relations we consider on finiteness spaces must have something to do with finiteness structures, whose behaviour w.r.t. the inclusion order on webs is not trivial. Our first task is thus to describe the different orders derived from set inclusion that can naturally endow Fin. We put forward two of them: the largest one, *finiteness inclusion*, is a cpo on finiteness spaces; the most restrictive one, *finiteness extension*, reflects more closely the inclusion order on webs. We then show the interest of studying both orders simultaneously: if a sequence $\overrightarrow{\mathcal{A}}$ of finiteness spaces is increasing for finiteness extension, then its supremum for finiteness inclusion is *exact*, i.e. it finiteness structure is obtained as the union of the finiteness structures in the sequence. This property prompts us to introduce various notions of continuity for finiteness inclusion, depending on the exactness of the suprema we consider. We then discuss the continuity of transport functors: type blindness is an essential property, in that it ensures ownership relations are stable under inclusions of webs (Lemma 2.7).

Finally, recall Fin is not cpo-enriched: the least fixpoint of an $(n+1)$-ary functor in one of its variables might not be functorial in the others. In the next section, we will however exhibit a restricted class of $(n+1)$-ary transport functors, the fixpoints of which are $n$-ary functors: the transport technique is again essential in that development. At the time of writing, we do not know whether this could be generalized to a larger class of transport functors.

## 4.1 Three order relations on finiteness spaces

We can consider two natural orders on finiteness spaces, both based on the inclusion of webs:

- *finiteness inclusion*: write $\mathcal{A} \sqsubseteq \mathcal{B}$ if $|\mathcal{A}| \subseteq |\mathcal{B}|$ and $\mathfrak{F}\,(\mathcal{A}) \subseteq \mathfrak{F}\,(\mathcal{B})$ ;
- *finiteness extension*: write $\mathcal{A} \preceq \mathcal{B}$ if $|\mathcal{A}| \subseteq |\mathcal{B}|$ and $\mathfrak{F}\,(\mathcal{A}) = \mathfrak{F}\,(\mathcal{B}) \cap \mathfrak{P}\,(|\mathcal{A}|)$.

Notice that the dual construction is increasing for the extension order: $\mathcal{A} \preceq \mathcal{B}$ iff $\mathcal{A}^{\perp} \preceq \mathcal{B}^{\perp}$. In general, this does not hold for finiteness inclusion: we may have $\mathcal{A} \sqsubseteq \mathcal{B}$ and $\mathcal{A}^{\perp} \not\sqsubseteq \mathcal{B}^{\perp}$. When $|\mathcal{A}| = |\mathcal{B}|$ we even obtain $\mathcal{A} \sqsubseteq \mathcal{B}$ iff $\mathcal{B}^{\perp} \sqsubseteq \mathcal{A}^{\perp}$ (whereas, in that case, $\mathcal{A} \preceq \mathcal{B}$ iff $\mathcal{A} = \mathcal{B}$). Thus we could equivalently consider the order given by the dual inclusion, $\mathcal{A} \sqsubseteq\!\!\!\!\!\!\sim\ \mathcal{B}$ if $\mathcal{A}^{\perp} \sqsubseteq \mathcal{B}^{\perp}$, in place of $\sqsubseteq$. On a side note, observe that $\mathcal{A} \preceq \mathcal{B}$ iff we have $\mathcal{A} \sqsubseteq \mathcal{B}$ and $\mathcal{A} \sqsubseteq\!\!\!\!\!\!\sim\ \mathcal{B}$ simultaneously. Moreover

**0** is the minimum of each of these orders (recall that **0** is the empty finiteness space). From now on, we consider only $\sqsubseteq$ and $\preceq$: the properties of $\sqsupseteq$ are exactly those of $\sqsubseteq$ up to finiteness duality.

**Lemma 4.1** *Every family $\overrightarrow{\mathcal{A}}$ of finiteness spaces admits a least upper bound $\bigsqcup \overrightarrow{\mathcal{A}}$ (their* finiteness supremum*) and a greatest lower bound $\bigsqcap \overrightarrow{\mathcal{A}}$ (their* finiteness infimum*) for the finiteness inclusion order. They are given by $\left|\bigsqcup \overrightarrow{\mathcal{A}}\right| = \bigcup \overrightarrow{|\mathcal{A}|}$, $\left|\bigsqcap \overrightarrow{\mathcal{A}}\right| = \bigcap \overrightarrow{|\mathcal{A}|}$, $\mathfrak{F}\left(\bigsqcup \overrightarrow{\mathcal{A}}\right) = \left(\bigcup \overrightarrow{\mathfrak{F}(\mathcal{A})}\right)^{\perp\perp_{\bigcup \overrightarrow{|\mathcal{A}|}}}$ and $\mathfrak{F}\left(\bigsqcap \overrightarrow{\mathcal{A}}\right) = \bigcap \overrightarrow{\mathfrak{F}(\mathcal{A})}$. In particular, $\sqsubseteq$ is a complete partial order on finiteness spaces.*

**Proof** This is a general fact for bidual closure operators. □

In the following, unless otherwise stated, suprema and infima are always relative to the inclusion order $\sqsubseteq$, as described in the previous lemma.

Notice that, in general, $\bigcup \overrightarrow{\mathfrak{F}(\mathcal{A})}$ is not a finiteness structure on $\left|\bigsqcup \mathcal{A}\right|$ by itself, hence the bidual closure in $\mathfrak{F}\left(\bigsqcup \overrightarrow{\mathcal{A}}\right)$:

**Counter-example 4.2** *Let $\mathfrak{A}$ be any fake finiteness structure on some set $A$, that is such that $\mathfrak{A} \subsetneq \mathfrak{A}^{\perp\perp_A}$. For all $f \in \mathfrak{A}$, let $\mathcal{A}_f = (f, \mathfrak{P}(f))$. Then $\bigcup_{f\in\mathfrak{A}} \mathfrak{F}(\mathcal{A}_f) = \mathfrak{A}$, but $\mathfrak{F}\left(\bigsqcup_{f\in\mathfrak{A}} \mathcal{A}_f\right) = \mathfrak{A}^{\perp\perp_A}$.*

When however $\bigcup \overrightarrow{\mathfrak{F}(\mathcal{A})}$ is a finiteness structure, we have: $\mathfrak{F}\left(\bigsqcup \overrightarrow{\mathcal{A}}\right) = \left(\bigcup \overrightarrow{\mathfrak{F}(\mathcal{A})}\right)^{\perp\perp} = \bigcup \overrightarrow{\mathfrak{F}(\mathcal{A})}$ and we say $\bigsqcup \overrightarrow{\mathcal{A}}$ is an *exact* supremum.

Suprema and infima for $\preceq$ do not exist in general, even considering the variant up to bijections: $\mathcal{A} \precsim \mathcal{B}$ if there is $\mathcal{A}' \cong \mathcal{A}$ such that $\mathcal{A}' \preceq \mathcal{B}$.[3]

**Counter-example 4.3** *Let $\overrightarrow{\mathfrak{F}} = (\mathfrak{F}_n)_{n\in\mathbf{N}}$ be the unique sequence of finiteness spaces such that, for all $n \in \mathbf{N}$, $|\mathfrak{F}_n| = \{0, \dots, n-1\}$: then any finiteness space of web $\mathbf{N}$ is a $\preceq$-upper bound of all the $\mathfrak{F}_n$'s, hence a $\precsim$-upper bound; but, e.g., $\mathcal{N} = (\mathbf{N}, \mathfrak{P}_{\mathrm{f}}(\mathbf{N}))$ and $\mathcal{N}^\perp$ have no common $\precsim$-lower bound.*

Notice however that in that case $\bigsqcup \overrightarrow{\mathfrak{F}} = \mathcal{N}$ is an exact supremum. This remark is actually an instance of a more general fact. Indeed:

**Lemma 4.4** *If $\overrightarrow{\mathcal{A}}$ is an $\preceq$-increasing sequence, then $\bigsqcup \overrightarrow{\mathcal{A}}$ is exact.*

**Proof** Apply the transport lemma in the form of the Corollary 3.6 to $A = \bigcup_{n\in\mathbf{N}} |\mathcal{A}_n|$ and to the following $(\{*\} \cup \mathbf{N})$-indexed families of finiteness spaces and relations:

- $\mathcal{B}_* = \mathcal{N}$ and $\forall n \in \mathbf{N}$, $\mathcal{B}_n = \mathcal{A}_n$
- $f_* = \{(\alpha, n);\ \alpha \notin |\mathcal{A}_n|\}$ and $\forall n \in \mathbf{N}$, $f_n = \mathsf{id}^{|\mathcal{A}_n|} = \{(\alpha, \alpha);\ \alpha \in |\mathcal{A}_n|\}$.

Then, using that $\overrightarrow{\mathcal{A}}$ is increasing, the reader can easily check that:

$$\mathfrak{F}_{(\mathcal{B}_*, \overrightarrow{\mathcal{B}}), (f_*, \overrightarrow{f})} = \left\{a \subseteq \bigcup \overrightarrow{|\mathcal{A}|};\ \exists p \in \mathbf{N},\ a \subseteq |\mathcal{A}_p| \wedge \forall n \in \mathbf{N},\ a \cap |\mathcal{A}_n| \in \mathfrak{F}(\mathcal{A}_n)\right\}.$$

We conclude that $\bigcup \overrightarrow{\mathfrak{F}(\mathcal{A})} = \mathfrak{F}_{(\mathcal{B}_*, \overrightarrow{\mathcal{B}}), (f_*, \overrightarrow{f})}$, hence $\bigcup \overrightarrow{\mathfrak{F}(\mathcal{A})}$ is a finiteness structure. □

[3] This preorder is considered by Ehrhard (2005, unpublished preliminary version) in order to describe the interpretation of second order quantification of linear logic.

Notice that this relies heavily on both the linear ordering of the family and the extension order as is shown by the following counter-examples.

**Counter-example 4.5 (A directed family for finiteness extension)** *Notice that the family of finiteness spaces in Counter-example 4.2, whose supremum is not exact, is however directed for $\preceq$.*

**Counter-example 4.6 (An increasing sequence for finiteness inclusion)** *Recall that the sequence of finiteness structures $(\mathfrak{C}_n)_{n\in\mathbf{N}}$ of Counter-example 3.3 is increasing for inclusion. We then form the sequence $(\mathcal{C}_n)_{n\in\mathbf{N}}$ where $\mathcal{C}_n = (\mathbf{N}\times\mathbf{N}, \mathfrak{C}_n)$, which is increasing for $\sqsubseteq$. Then $\bigsqcup \overrightarrow{\mathcal{C}}$ is not exact, because $\bigcup \overrightarrow{\mathfrak{C}}$ is a fake finiteness structure.*

Lemma 4.4 emphasizes the fact that we should not focus on finiteness inclusion or finiteness extension separately, but rather investigate how they can interact. Notice for instance that, as a corollary of Lemma 4.4, for all $\preceq$-increasing functor $\mathcal{T}$ from $\underline{\mathrm{Fin}}$ to $\underline{\mathrm{Fin}}$, $\mu\mathcal{T} = \bigsqcup_{n\in\mathbf{N}} \mathcal{T}^n\mathbf{0}$ is exact. In the following we show that this defines the least fixpoint of $\mathcal{T}$ up to some hypotheses on $\mathcal{T}$ w.r.t. both finiteness inclusion and finiteness extension.

## 4.2 Exact continuity and direct continuity

A *directed supremum* is the $\sqsubseteq$-supremum of a $\sqsubseteq$-directed family. We say a $\sqsubseteq$-monotonic functor $\mathcal{T}$ in $\underline{\mathrm{Fin}}$ is:

- *weakly continuous* if $\mathcal{T}$ commutes to directed suprema when they are exact;
- *exactly continuous* if $\mathcal{T}$ commutes to exact directed suprema;
- *directly continuous* if $\mathcal{T}$ commutes to all directed suprema.

Let us precise the second case: $\mathcal{T}$ is exactly continuous iff it is weakly continuous and $\mathcal{T}\bigsqcup\overrightarrow{\mathcal{A}} = \bigsqcup\overrightarrow{\mathcal{T}\mathcal{A}}$ is exact for all exact directed supremum $\bigsqcup\overrightarrow{\mathcal{A}}$. In particular, both direct continuity and exact continuity imply weak continuity, but there is no *a priori* implication between direct continuity and exact continuity: a directly continuous functor may not preserve exactness; an exactly continuous functor may not preserve non-exact suprema. Moreover, notice that exactly continuous (resp. directly continuous) functors compose, but weakly continuous ones may not: if $\mathcal{T}$ and $\mathcal{U}$ are weakly continous functors and $\bigsqcup\overrightarrow{\mathcal{A}}$ is an exact directed supremum, we do not know whether $\bigsqcup\overrightarrow{\mathcal{T}\mathcal{A}}$ is exact, hence we can not deduce that $\mathcal{U}$ commutes to this supremum.

The main property we shall use about weakly continuous functors (and *a fortiori* exactly continuous or directly continous ones) is that they admit least fixpoints, as soon as they preserve finiteness extensions.

**Lemma 4.7** *If $\mathcal{T}$ is a weakly continuous functor, which is moreover $\preceq$-increasing, then $\mu\mathcal{T} = \bigsqcup_{n\in\mathbf{N}}\mathcal{T}^n\mathbf{0}$ is the ($\sqsubseteq$-)least fixpoint of $\mathcal{T}$.*

**Proof** We have already remarked in section 4.1 that $\mu\mathcal{T}$ is an exact directed supremum: hence $\mathcal{T}\mu\mathcal{T} = \bigsqcup_{n\in\mathbf{N}}\mathcal{T}^{n+1}\mathbf{0} = \mu\mathcal{T}$ because $\mathbf{0}$ is minimum. Now let $\mathcal{Y}$ be any fixpoint of $\mathcal{T}$: by iterating the application of $\mathcal{T}$ to the inequation $\mathbf{0}\preceq\mathcal{Y}$, we obtain $\mathcal{T}^n\mathbf{0}\preceq\mathcal{T}^n\mathcal{Y} = \mathcal{Y}$, hence $\mathcal{T}^n\mathbf{0}\sqsubseteq\mathcal{Y}$ for all $n\in\mathbf{N}$, and finally $\mu\mathcal{T}\sqsubseteq\mathcal{Y}$. □

In order to generalize the definitions of continuity to $I$-ary functors, we adapt the same conventions as in the relational setting. By $\overleftrightarrow{\mathcal{A}}$, we denote an $I$-indexed family $\left(\overleftarrow{\mathcal{A}_i}\right)_{i\in I}$ of families of finiteness spaces, where each $\overleftarrow{\mathcal{A}_i} = (\mathcal{A}_{i,j})_{j\in J_i}$ takes indices in some variable set $J_i$. We say $\overleftrightarrow{\mathcal{A}}$ is directed if each $\overleftarrow{\mathcal{A}_i}$ is directed. We write $\overrightarrow{\bigsqcup\overleftarrow{\mathcal{A}}}$ for $\left(\bigsqcup\overleftarrow{\mathcal{A}_i}\right)_{i\in I} = \left(\bigsqcup_{j\in J_i}\mathcal{A}_{i,j}\right)_{i\in I}$ and call this family the supremum of $\overleftrightarrow{\mathcal{A}}$: we say this supremum is exact if each $\bigsqcup\overleftarrow{\mathcal{A}_i}$ is. Finally, if $\mathcal{T}$ is a functor from $\underline{\mathrm{Fin}}^I$ to $\underline{\mathrm{Fin}}$, we write $\overleftarrow{\mathcal{T}\overrightarrow{\mathcal{A}}}$ for $\left(\mathcal{T}\left(\mathcal{A}_{i,j_i}\right)_{i\in I}\right)_{\overrightarrow{j\in J}}$.

**Definition 4.8** *Let $\mathcal{T}$ be a $\sqsubseteq$-monotonic functor from $\underline{\mathrm{Fin}}^I$ to $\underline{\mathrm{Fin}}$. We say $\mathcal{T}$ is:*

- exactly continuous *if it commutes to all exact directed suprema, i.e. $\bigsqcup \overleftarrow{\mathcal{T}\overrightarrow{\mathcal{A}}}$ is exact and $\mathcal{T}\left(\overrightarrow{\bigsqcup \overleftarrow{\mathcal{A}}}\right) = \bigsqcup \overleftarrow{\mathcal{T}\overrightarrow{\mathcal{A}}}$ and as soon as $\overrightarrow{\overleftarrow{\mathcal{A}}}$ is directed and $\overrightarrow{\bigsqcup \overleftarrow{\mathcal{A}}}$ is exact;*
- directly continuous *if it commutes to all directed suprema, i.e. $\mathcal{T}\left(\overrightarrow{\bigsqcup \overleftarrow{\mathcal{A}}}\right) = \bigsqcup \overleftarrow{\mathcal{T}\overrightarrow{\mathcal{A}}}$ as soon as $\overrightarrow{\overleftarrow{\mathcal{A}}}$ is directed.*

The following result follows from the associativity of suprema:

**Lemma 4.9** *Directly continuous (resp. exactly continuous) functors compose: if $\mathcal{T}$ is a directly continuous (resp. exactly continuous) functor from $\underline{\mathrm{Fin}}^I$ to $\underline{\mathrm{Fin}}$ and, for all $i \in I$, $U_i$ is a directly continuous (resp. exactly continuous) functor from $\underline{\mathrm{Fin}}^{J_i}$ to $\underline{\mathrm{Fin}}$ then $\mathcal{T} \circ \overrightarrow{U}$ is a directly continuous (resp. exactly continuous) functor from $\underline{\mathrm{Fin}}^{\Sigma \overrightarrow{J}}$ to $\underline{\mathrm{Fin}}$.*

We do not detail the proof as it amounts to a futile exercise in formality: we have to consider families of families of families of finiteness spaces, then simply check that the above definitions apply, up to some juggling with indices.

## 4.3 Continuity of transport functors

In this section, we establish the properties of type blind transport functors w.r.t. the order relations on finiteness spaces. Notice that the same results would actually hold for arbitrary transport functors, provided the properties established in Lemma 2.7 hold.

**Lemma 4.10** *All type blind transport functors are monotonic for both $\preceq$ and $\sqsubseteq$.*

**Proof** Let $\mathcal{T}$ be a type blind transport functor with ownership relation $\overrightarrow{\mathsf{own}}$ and assume that $\overrightarrow{\mathcal{A} \sqsubseteq \mathcal{B}}$. Since $|\mathcal{T}|$ is type blind, Lemma 2.3 entails that $\left|\mathcal{T}\overrightarrow{\mathcal{A}}\right| \subseteq \left|\mathcal{T}\overrightarrow{\mathcal{B}}\right|$. Then let $\widetilde{a} \subseteq \left|\mathcal{T}\overrightarrow{\mathcal{A}}\right|$. We have $\widetilde{a} \in \mathfrak{F}\left(\mathcal{T}\overrightarrow{\mathcal{A}}\right)$ iff for all $i \in I$, $\mathsf{own}_i^{\overrightarrow{|\mathcal{A}|}} \cdot \widetilde{a} \in \mathfrak{F}(\mathcal{A}_i)$. Since $\overrightarrow{|\mathcal{A}| \subseteq |\mathcal{B}|}$ and $\mathsf{own}_i$ is a lax natural transformation, Lemma 2.7 implies $\mathsf{own}_i^{\overrightarrow{|\mathcal{A}|}} \cdot \widetilde{a} = \mathsf{own}_i^{\overrightarrow{|\mathcal{B}|}} \cdot \widetilde{a}$. Moreover, $\mathfrak{F}(\mathcal{A}_i) \subseteq \mathfrak{F}(\mathcal{B}_i)$, hence $\mathsf{own}_i^{\overrightarrow{|\mathcal{B}|}} \cdot \widetilde{a} \in \mathfrak{F}(\mathcal{B}_i)$. We thus obtain $\widetilde{a} \in \mathfrak{F}\left(\mathcal{T}\overrightarrow{\mathcal{B}}\right)$. We conclude that $\mathcal{T}\overrightarrow{\mathcal{A}} \sqsubseteq \mathcal{T}\overrightarrow{\mathcal{B}}$.

If we moreover assume that $\overrightarrow{\mathcal{A} \preceq \mathcal{B}}$ then $\mathsf{own}_i^{\overrightarrow{|\mathcal{A}|}} \cdot \widetilde{a} \in \mathfrak{F}(\mathcal{A}_i)$ iff $\mathsf{own}_i^{\overrightarrow{|\mathcal{A}|}} \cdot \widetilde{a} \in \mathfrak{F}(\mathcal{B}_i)$ and we obtain $\widetilde{a} \in \mathfrak{F}\left(\mathcal{T}\overrightarrow{\mathcal{A}}\right)$ iff $\widetilde{a} \in \mathfrak{F}\left(\mathcal{T}\overrightarrow{\mathcal{B}}\right)$. We conclude that $\mathcal{T}\overrightarrow{\mathcal{A}} \preceq \mathcal{T}\overrightarrow{\mathcal{B}}$. □

**Lemma 4.11** *A type blind transport functor is exactly continuous as soon as its underlying web functor is continuous on sets.*

**Proof** Let $\mathcal{T}$ be a type blind transport functor with ownership $\overrightarrow{\mathsf{own}}$ and assume its web functor $T$ is continous on sets. Let $\overrightarrow{\overleftarrow{\mathcal{A}}}$ be directed and such that each $\bigsqcup \overleftarrow{\mathcal{A}}_i$ is exact. We prove that $\mathcal{T}\overrightarrow{\bigsqcup \overleftarrow{\mathcal{A}}} = \bigsqcup \overleftarrow{\mathcal{T}\overrightarrow{\mathcal{A}}}$. First notice that the webs $\left|\mathcal{T}\overrightarrow{\bigsqcup \overleftarrow{\mathcal{A}}}\right| = T\bigcup \overrightarrow{|\mathcal{A}|}$ and $\left|\bigsqcup \overleftarrow{\mathcal{T}\overrightarrow{\mathcal{A}}}\right| = \bigcup T\overrightarrow{|\mathcal{A}|}$ are equal because $T$ is continuous on sets. We are left to prove the equality of finiteness structures, that is $\mathfrak{F}\left(\mathcal{T}\overrightarrow{\bigsqcup \overleftarrow{\mathcal{A}}}\right) = \mathfrak{F}\left(\bigsqcup \overleftarrow{\mathcal{T}\overrightarrow{\mathcal{A}}}\right)$ or equivalently $\mathfrak{F}\left(\mathcal{T}\overrightarrow{\bigsqcup \overleftarrow{\mathcal{A}}}\right)^{\perp} = \mathfrak{F}\left(\bigsqcup \overleftarrow{\mathcal{T}\overrightarrow{\mathcal{A}}}\right)^{\perp}$. Let's make explicit that by Corollary 3.6 and the definition of $\bigsqcup$:

(a) $\widetilde{a}' \in \mathfrak{F}\left(\mathcal{T}\overrightarrow{\bigsqcup \overleftarrow{\mathcal{A}}}\right)^{\perp}$ iff for all $\overrightarrow{a}$ with $a_i \in \mathfrak{F}\left(\bigsqcup \overleftarrow{\mathcal{A}}_i\right)$ for all $i \in I$, $\widetilde{a}' \perp_{\mathrm{f}} \overrightarrow{\mathsf{own}}^{\overrightarrow{\bigcup \overleftarrow{|\mathcal{A}|}}} \setminus \overrightarrow{a}$;

(b) $\widetilde{a}' \in \mathfrak{F}\left(\bigsqcup \overleftarrow{\mathcal{T}\overrightarrow{\mathcal{A}}}\right)^{\perp}$ iff for all $\overrightarrow{j} \in \prod \overrightarrow{J}$ and all $\widetilde{a} \in \mathfrak{F}\left(\mathcal{T}\overrightarrow{\mathcal{A}}_{\overrightarrow{j}}\right)$, $\widetilde{a}' \perp_{\mathrm{f}} \widetilde{a}$.

We prove both characterizations are equivalent.

Assume the condition in (a) holds and let $\overrightarrow{j} \in \prod \overrightarrow{J}$ and $\widetilde{a} \in \mathfrak{F}\left(\mathcal{T}\overrightarrow{\mathcal{A}}_{\overrightarrow{j}}\right)$. For all $i \in I$, let $a_i = \mathsf{own}_i^{\overrightarrow{|\mathcal{A}|_{\overrightarrow{j}}}} \cdot \widetilde{a}$: $a_i \in \mathfrak{F}(\mathcal{A}_{i,j_i}) \subseteq \bigcup \overleftarrow{\mathfrak{F}(\mathcal{A}_i)} \subseteq \mathfrak{F}\left(\bigsqcup \overleftarrow{\mathcal{A}}_i\right)$. Then $\widetilde{a} \subseteq \overrightarrow{\mathsf{own}}^{\overrightarrow{|\mathcal{A}|_{\overrightarrow{j}}}} \setminus \overrightarrow{a} = \overrightarrow{\mathsf{own}}^{\overrightarrow{\bigcup \overleftarrow{|\mathcal{A}|}}} \setminus \overrightarrow{a}$ by Lemma 2.7; by condition (a), we deduce that $\widetilde{a}' \cap \widetilde{a}$ is finite.

Now assume the condition in (b) holds and let $\overrightarrow{a}$ be such that $a_i \in \mathfrak{F}\left(\bigsqcup \overleftarrow{\mathcal{A}}_i\right)$ for all $i \in I$. Since each of these suprema is exact, i.e. $\mathfrak{F}\left(\bigsqcup \overleftarrow{\mathcal{A}}_i\right) = \bigcup \overleftarrow{\mathfrak{F}(\mathcal{A}_i)}$, there exists $\overrightarrow{j \in J}$ such that $a_i \in \mathfrak{F}(\mathcal{A}_{i,j_i})$ for all $i \in I$. Hence $\overrightarrow{\mathsf{own}}^{\overrightarrow{|\mathcal{A}|_{\overrightarrow{j}}}} \setminus \overrightarrow{a} \in \mathfrak{F}\left(\mathcal{T}\overrightarrow{\mathcal{A}}_{\overrightarrow{j}}\right)$ and we conclude by Lemma 2.7.

It remains only to prove that $\bigsqcup \overleftarrow{\mathcal{T}\overrightarrow{\mathcal{A}}}$ is exact. Let $\widetilde{a} \subseteq T\overrightarrow{\bigcup \overleftarrow{|\mathcal{A}|}}$. We have just proved that $\widetilde{a} \in \mathfrak{F}\left(\bigsqcup \overleftarrow{\mathcal{T}\overrightarrow{\mathcal{A}}}\right)$ iff $\widetilde{a} \in \mathfrak{F}\left(\mathcal{T}\overrightarrow{\bigsqcup \overleftarrow{\mathcal{A}}}\right)$ iff for all $i \in I$, $\mathsf{own}_i^{\overrightarrow{\bigcup \overleftarrow{|\mathcal{A}|}}} \cdot \widetilde{a} \in \mathfrak{F}\left(\bigsqcup \overleftarrow{\mathcal{A}}_i\right)$. Now, because $\bigsqcup \overleftarrow{\mathcal{A}}_i$ is exact, $\mathfrak{F}\left(\bigsqcup \overleftarrow{\mathcal{A}}_i\right) = \bigcup \overleftarrow{\mathfrak{F}(\mathcal{A}_i)}$. Thus $\widetilde{a} \in \mathfrak{F}\left(\bigsqcup \overleftarrow{\mathcal{T}\overrightarrow{\mathcal{A}}}\right)$ iff for all $i \in I$, there exists $j_i \in J_i$ such that $\mathsf{own}_i^{\overrightarrow{\bigcup \overleftarrow{|\mathcal{A}|}}} \cdot \widetilde{a} \in \mathfrak{F}(\mathcal{A}_{i,j_i})$. Then $\widetilde{a} \in \mathfrak{F}\left(\mathcal{T}\overrightarrow{\mathcal{A}}_{\overrightarrow{j}}\right)$, since $\mathsf{own}_i^{\overrightarrow{|\mathcal{A}_{\overrightarrow{j}}|}} \cdot \widetilde{a} \subseteq \mathsf{own}_i^{\overrightarrow{\bigcup \overleftarrow{|\mathcal{A}|}}} \cdot \widetilde{a}$ for all $i \in I$ (again by Lemma 2.7). □

**Example 4.12** *The web functors of sums, finite multisets and products are all continuous, hence* $\mathbin{⅋}$, $\bigoplus$, ! *and binary* $\otimes$ *are exactly continuous.*

We say the ownership relation $\overrightarrow{\mathsf{own}}$ is *local* if, for all family $\overrightarrow{A}$ and all $i \in I$:

- $\mathsf{own}_i \cdot (\mathsf{own}_i \setminus a_i) = a_i$ for all $a_i \subseteq A_i$;
- $\mathsf{own}_j \cdot (\mathsf{own}_i \setminus a_i) = A_j$ for all $a_i \subseteq A_i$ and all $j \neq i$;
- $\mathsf{own}_i$ preserves intersections, i.e. $\mathsf{own}_i \cdot \bigcap \overleftarrow{\widetilde{a}} = \bigcap \overleftarrow{\mathsf{own}_i \cdot \widetilde{a}}$ for all $\overleftarrow{\widetilde{a} \in T\overrightarrow{A}}$.

Intuitively, an ownership relation is local if its components do not interact with each other. In particular, if $\overrightarrow{\mathsf{own}}$ is local then $\mathsf{own}_i \cdot \left(\overrightarrow{\mathsf{own}} \setminus \overrightarrow{a}\right) = a_i$ for all $i \in I$.

**Lemma 4.13** *A type blind transport functor is directly continuous as soon as its underlying web functor is continuous and its ownership relation is local.*

**Proof** The proof differs from the previous one only in the direction (b) to (a), where we used the exactness condition, which is no longer available. So, assuming (b) and in order to establish (a), we first prove the following intermediate result: $\mathsf{own}_i^{\overrightarrow{\bigcup \overleftarrow{|\mathcal{A}|}}} \cdot \widetilde{a}' \in \mathfrak{F}\left(\bigsqcup \overleftarrow{\mathcal{A}}_i\right)^{\perp} = \left(\bigcup \overleftarrow{\mathfrak{F}(\mathcal{A}_i)}\right)^{\perp}$ for all $i \in I$. Indeed, let $\overrightarrow{j} \in \prod \overrightarrow{J}$ and $\overrightarrow{a \in \mathfrak{F}\left(\mathcal{A}_{\overrightarrow{j}}\right)}$ (in particular we chose $j_i$ to be any index in $J_i$ and $a_i$ to be any finitary subset of $\mathcal{A}_{i,j_i}$) and write $\widetilde{a}'' = \overrightarrow{\mathsf{own}}^{\overrightarrow{\bigcup \overleftarrow{|\mathcal{A}|}}} \setminus \overrightarrow{a}$: by Lemma 2.7, $\widetilde{a}'' = \overrightarrow{\mathsf{own}}^{\overrightarrow{|\mathcal{A}_{\overrightarrow{j}}|}} \setminus \overrightarrow{a} \in \mathfrak{F}\left(\mathcal{T}\overrightarrow{\mathcal{A}}_{\overrightarrow{j}}\right)$ and thus $\widetilde{a} = \widetilde{a}' \cap \widetilde{a}''$ is finite. Moreover, for all $i \in I$, $\mathsf{own}_i^{\overrightarrow{\bigcup \overleftarrow{|\mathcal{A}|}}} \cdot \widetilde{a}'' = a_i$, because $\overrightarrow{\mathsf{own}}$ is local. Hence $\left(\mathsf{own}_i^{\overrightarrow{\bigcup \overleftarrow{|\mathcal{A}|}}} \cdot \widetilde{a}'\right) \cap a_i = \left(\mathsf{own}_i^{\overrightarrow{\bigcup \overleftarrow{|\mathcal{A}|}}} \cdot \widetilde{a}'\right) \cap \left(\mathsf{own}_i^{\overrightarrow{\bigcup \overleftarrow{|\mathcal{A}|}}} \cdot \widetilde{a}''\right) = \mathsf{own}_i^{\overrightarrow{\bigcup \overleftarrow{|\mathcal{A}|}}} \cdot \widetilde{a}$ because $\mathsf{own}_i$ preserves intersections. Since $\widetilde{a}$ is finite and $\mathsf{own}_i$ is quasi-functional, we conclude that $\mathsf{own}_i^{\overrightarrow{\bigcup \overleftarrow{|\mathcal{A}|}}} \cdot \widetilde{a}' \perp_{\mathrm{f}} a_i$. Since this holds for all $a_i \in \bigcup \overleftarrow{\mathfrak{F}(\mathcal{A}_i)}$, we obtain $\mathsf{own}_i^{\overrightarrow{\bigcup \overleftarrow{|\mathcal{A}|}}} \cdot \widetilde{a}' \in \mathfrak{F}\left(\bigsqcup \overleftarrow{\mathcal{A}}_i\right)^{\perp}$.

Then let $\overrightarrow{a}$ be such that $a_i \in \mathfrak{F}\left(\bigsqcup \overleftarrow{\mathcal{A}_i}\right)$ for all $i \in I$ and write $\widetilde{a}'' = \overrightarrow{\mathsf{own}}^{\overrightarrow{\bigcup^{\overleftarrow{|\mathcal{A}|}}}} \setminus \overrightarrow{a}$: we must show that $\widetilde{a}''' = \widetilde{a}' \cap \widetilde{a}''$ is finite. For all $i \in I$, $a_i''' = \mathsf{own}_i^{\overrightarrow{\bigcup^{\overleftarrow{|\mathcal{A}|}}}} \cdot \widetilde{a}''' \subseteq \left(\mathsf{own}_i^{\overrightarrow{\bigcup^{\overleftarrow{|\mathcal{A}|}}}} \cdot \widetilde{a}'\right) \cap a_i$ is finite: hence $a_i''' \in \mathfrak{F}(\mathcal{A}_{i,j})$ for any $j \in J_i$ such that $a_i''' \subseteq |\mathcal{A}_{i,j}|$. Fix $j_i$ to be one such $j$ for all $i \in I$. We obtain $\widetilde{a}''' \in \mathfrak{F}\left(\mathcal{T}\overrightarrow{\mathcal{A}_{\overrightarrow{j}}}\right)$. We conclude since $\widetilde{a}' \perp_{\mathrm{f}} \widetilde{a}'''$ and thus $\widetilde{a}' \cap \widetilde{a}''' = \widetilde{a}'''$ is finite. □

**Example 4.14** *Since* $\overrightarrow{\mathsf{rest}}$, $\mathsf{indx}$ *and* $\mathsf{supp}$ *are local,* $\parr$, $\oplus$ *and* ! *are directly continuous.*

The conditions under which we proved direct continuity of transport functors are not minimal. For instance $\overrightarrow{\mathsf{proj}}$ is not local even for $I = \{1, 2\}$: since $A \times \emptyset = \emptyset$, $(\mathsf{proj}_1^{A,\emptyset}, \mathsf{proj}_2^{A,\emptyset}) \setminus (a, \emptyset) = \emptyset$ for all $a \subseteq A$ and then $\mathsf{proj}_1^{A,\emptyset} \cdot \emptyset = \emptyset \neq a$ in general. However:

**Lemma 4.15** *Finite tensor products are directly continuous.*

**Proof** It is sufficient to consider binary tensor products and prove continuity w.r.t. one of the parameters. Let $\overleftarrow{\mathcal{B}} = (\mathcal{B}_j)_{j \in J}$ be a directed supremum of finiteness spaces: we prove $\mathcal{A} \otimes \bigsqcup \overleftarrow{\mathcal{B}} = \bigsqcup (\mathcal{A} \otimes \mathcal{B}_j)_{j \in J}$ or, equivalently, $\left(\mathcal{A} \otimes \bigsqcup \overleftarrow{\mathcal{B}}\right)^{\perp} = \left(\bigsqcup (\mathcal{A} \otimes \mathcal{B}_j)_{j \in J}\right)^{\perp}$. That $\mathfrak{F}\left(\mathcal{A} \otimes \bigsqcup \overleftarrow{\mathcal{B}}\right)^{\perp} \subseteq \mathfrak{F}\left(\bigsqcup (\mathcal{A} \otimes \mathcal{B}_j)_{j \in J}\right)^{\perp}$ goes by the same argument as in Lemma 4.11.

Assume that $c' \in \mathfrak{F}\left(\bigsqcup (\mathcal{A} \otimes \mathcal{B}_j)_{j \in J}\right)^{\perp}$: we prove $c' \in \mathfrak{F}\left(\mathcal{A} \otimes \bigsqcup \overleftarrow{\mathcal{B}}\right)^{\perp} = \mathfrak{F}\left(\mathcal{A} \multimap \left(\bigsqcup \overleftarrow{\mathcal{B}}\right)^{\perp}\right)$. If $a \in \mathfrak{F}(\mathcal{A})$ then $c' \cdot a \in \mathfrak{F}\left(\bigsqcup \overleftarrow{\mathcal{B}}\right)^{\perp}$. Indeed, for all $j \in J$ and $b \in \mathfrak{F}(\mathcal{B}_j)$, we have $a \times b \in \mathfrak{F}(\mathcal{A} \otimes \mathcal{B}_j) \subseteq \mathfrak{F}\left(\bigsqcup (\mathcal{A} \otimes \mathcal{B}_j)_{j \in J}\right)$, hence $c' \perp_{\mathrm{f}} a \times b$: then $c' \cdot a \perp_{\mathrm{f}} b$. In the other direction, let $\beta \in \left|\bigsqcup \overleftarrow{\mathcal{B}}\right| = \bigcup \overleftarrow{|\mathcal{B}|}$: let $j \in J$ such that $\beta \in |\mathcal{B}_j|$. Then, for all $a \in \mathfrak{F}(\mathcal{A})$, $a \times \{\beta\} \in \mathfrak{F}(\mathcal{A} \otimes \mathcal{B}_j)$, hence $c' \perp_{\mathrm{f}} a \times \{\beta\}$ and we obtain ${}^t c' \cdot \beta \perp_{\mathrm{f}} a$. We have thus proved that ${}^t c' \cdot \beta \in \mathfrak{F}\left(\mathcal{A}^{\perp}\right)$, which concludes the proof. □

It is still unclear to us if this argument can be adapted to lift the condition on the locality of $\overrightarrow{\mathsf{own}}$ in Lemma 4.13, and thus generalize direct continuity to all type blind transport functors with continuous web functors.

To sum up, remind from Section 4.2 that direct continuity and exact continuity imply weak continuity, from Lemma 4.7 that weakly continuous functors admit fixpoints and finally from Lemma 4.9 that direct continuity and exact continuity are stable under composition. Then, we can infer that the functors (in one variable) resulting of the composition of $\parr$, $\oplus$, ! and finite $\otimes$ admit fixpoints.

The status of the linear arrow functor $\multimap$ is still unclear in that regard, if only because its properties with relation to the orders on finiteness spaces are not straightforward. First notice that $\mathcal{A} \multimap \mathcal{B} = \left(\mathcal{A} \otimes \mathcal{B}^{\perp}\right)^{\perp}$ is functorial in $\mathcal{B}$ but cofunctorial in $\mathcal{A}$. By contrast, $\mathcal{A} \multimap \mathcal{B}$ is $\preceq$-increasing in both $\mathcal{A}$ and $\mathcal{B}$, whereas it is only $\sqsubseteq$-increasing in $\mathcal{B}$ and is not monotonic in $\mathcal{A}$.

# 5 The finitary relational model of the $\lambda$-calculus

It is a well known fact that Rel is a model of classical linear logic (Ehrhard; 2005, Appendix), and even of differential linear logic where:

- linear negation is the transpose cofunctor;
- multiplicatives are interpreted by cartesian products;
- additives are interpreted by disjoint unions;
- exponentials are interpreted by finite multisets.

The category of finiteness spaces and finitary relations $\underline{\mathrm{Fin}}$ is also a model of classical linear logic, which is the subject of the first part of Ehrhard's paper (2005). This result could actually be stated as follows: for any finiteness structure we impose on the relational interpretation of atomic formulas, the relational semantics of a proof is always finitary in the finiteness space denoted by its conclusion. In other words, that $\underline{\mathrm{Fin}}$ is a model of linear logic can be stated as a *property* of the interpretation of linear logic in $\underline{\mathrm{Rel}}$. This viewpoint fits very well with the previous developments of our paper, in which we explore how distinctive constructions and properties of $\underline{\mathrm{Rel}}$ can be transported to $\underline{\mathrm{Fin}}$. In the present section, we extend this stand to the study of the $\lambda$-calculus, which will allow us to discuss datatypes in the next section.

From the relational model of linear logic, we can derive an extensional model of the simply typed $\lambda$-calculus by the co-Kleisli construction: this gives rise to a cartesian closed category $\underline{\mathrm{Rel}}^!$. Objects in $\underline{\mathrm{Rel}}^!$ are sets and morphisms from $A$ to $B$ are *multirelations*, that is subsets of $A \Rightarrow B = \mathfrak{M}_{\mathrm{f}}(A) \times B$. Notice that this definition is an instance of Girard's translation of the intuitionistic arrow: $A \Rightarrow B = !A \multimap B$. Composition of multirelations is given by

$$g \circ^! f = \left\{ \left( \sum_{i=1}^{n} \overline{\alpha}_i, \gamma \right) ;\ n \in \mathbf{N} \wedge \exists \overline{\beta} = [\beta_1, \dots, \beta_n] \in\ !B,\ (\overline{\beta}, \gamma) \in g \wedge \forall i\, (\overline{\alpha}_i, \beta_i) \in f \right\}$$

as soon as $f \in \underline{\mathrm{Rel}}^!(A, B)$ and $g \in \underline{\mathrm{Rel}}^!(B, C)$. The identity multirelation on $A$ is the *dereliction*: $\mathsf{der}^A = \{([\alpha], \alpha);\ \alpha \in A\}$. The cartesian product is given by the disjoint union of sets $\mathop{\&}\limits_{\mathcal{L}} \overrightarrow{A}$, with projections $\overrightarrow{\pi} = \overrightarrow{\mathsf{rest} \circ \mathsf{der}}$. If, for all $i \in I$, $f_i \in \underline{\mathrm{Rel}}^!(A, B_i)$, then the unique morphism $\left\langle \overrightarrow{f} \right\rangle$ from $A$ to $\mathop{\&}\limits_{\mathcal{L}} \overrightarrow{B}$ such that $\mathsf{proj}_i \circ^! \left\langle \overrightarrow{f} \right\rangle = f_i$ for all $i$ is $\{(\overline{\alpha}, (i, \beta));\ (\overline{\alpha}, \beta) \in f_i,\ i \in I\}$. The terminal object denoted $\top$ is the empty set $\emptyset$, the unique multirelation from $A$ to $\emptyset$ being empty. The adjunction for closedness is $\underline{\mathrm{Rel}}^!(A \mathbin{\&} B, C) \cong \underline{\mathrm{Rel}}^!(A, !B \multimap C)$ which boils down to the natural bijection $!(A \mathbin{\&} B) \cong !A \otimes !B$.

## 5.1 Relational interpretation and finiteness property

In this section, we give an explicit description of the interpretation in $\underline{\mathrm{Rel}}^!$ of the basic constructions of simply typed $\lambda$-calculi with products. Type and term expressions are given by:

$$A, B = X \mid A \Rightarrow B \mid A \mathbin{\&} B \mid \top \qquad \text{and} \qquad s, t = x \mid a \mid \lambda x\, s \mid s\, t \mid \langle s, t \rangle \mid \pi_1 s \mid \pi_2 s \mid \langle\rangle$$

where $X$ ranges over a fixed set $\mathfrak{A}$ of atomic types, $x$ ranges over term variables and $a$ ranges over term constants. Of course, the variable $x$ is bound by the abstraction in $\lambda x\, s$, we consider terms up to $\alpha$-equivalence, and we denote by $s\,[x := t]$ the capture-avoiding substitution of term $t$ for variable $x$ in $s$.

To each variable or constant, we associate a type, so that each type admits infinitely many variables.[4] We write $\mathfrak{C}_A$ for the collection of constants of type $A$. A typing judgement is an expression $\Gamma \vdash s : A$ derived from the rules in Figure 1 where contexts $\Gamma$ and $\Delta$ range over finite lists $(x_1 : A_1, \dots, x_n : A_n)$ of typed variables. Since we do not impose Barendregt's convention on $\lambda$-terms, variables occuring in a context need not be pairwise distinct, hence the shape of rule (Var). If a term $s$ is typable, then its type is uniquely determined, say $A$, and then $\Gamma \vdash s : A$ iff $\Gamma$ contains the free variables of $s$. The operational semantics of a typed $\lambda$-calculus is given by a contextual equivalence relation $\simeq$ on typed terms: if $s \simeq t$, then $s$ and $t$ have the same type, say $A$; we then write $\Gamma \vdash s \simeq t : A$ for any suitable $\Gamma$. We write $\simeq_0$ for the least one such that $\pi_1\langle s, t\rangle \simeq_0 s$, $\pi_2\langle s, t\rangle \simeq_0 t$ and $(\lambda x\, s)\, t \simeq_0 s\,[x := t]$ (provided $t$ and $x$ have the same type).

Assume a set $[\![X]\!]$ is given for each atomic type $X$; then we interpret type constructions by $[\![A \Rightarrow B]\!] = [\![A]\!] \Rightarrow [\![B]\!]$, $[\![A \mathbin{\&} B]\!] = [\![A]\!] \mathbin{\&} [\![B]\!]$ and $[\![\top]\!] = \emptyset$. Further assume that with every constant $a \in \mathfrak{C}_A$ is associated a subset $[\![a]\!] \subseteq [\![A]\!]$. The relational semantics of a derivable

[4] The type system we present is thus in the style of Church rather than in the style of Curry: typing is syntax directed. This is only a technical convenience and the remaining of the paper could very well be recast in a Curry-style setting.

$$\frac{x \notin \Delta}{\Gamma, x : A, \Delta \vdash x : A}\ (\text{Var}) \qquad \frac{}{\Gamma \vdash \langle\rangle : \top}\ (\text{Unit}) \qquad \frac{a \in \mathfrak{C}_A}{\Gamma \vdash a : A}\ (\text{Const})$$

$$\frac{\Gamma, x : A \vdash s : B}{\Gamma \vdash \lambda x\, s : A \Rightarrow B}\ (\text{Abs}) \qquad \frac{\Gamma \vdash s : A \Rightarrow B \qquad \Gamma \vdash t : A}{\Gamma \vdash s\, t : B}\ (\text{App})$$

$$\frac{\Gamma \vdash s : A \qquad \Gamma \vdash t : B}{\Gamma \vdash \langle s, t\rangle : A \,\&\, B}\ (\text{Pair}) \qquad \frac{\Gamma \vdash s : A \,\&\, B}{\Gamma \vdash \pi_1 s : A}\ (\text{Left}) \qquad \frac{\Gamma \vdash s : A \,\&\, B}{\Gamma \vdash \pi_2 s : B}\ (\text{Right})$$

Figure 1: Rules of typed $\lambda$-calculi with products

$$\frac{x \notin \Delta}{\Gamma^{[]}, x^{[\alpha]} : A, \Delta^{[]} \vdash x^{\alpha} : A}\ [\![\text{Var}]\!] \qquad \frac{a \in \mathfrak{C}_A \quad \alpha \in [\![a]\!]}{\Gamma^{[]} \vdash a^{\alpha} : A}\ [\![\text{Const}]\!]$$

$$\frac{\Gamma, x^{\overline{\alpha}} : A \vdash s^{\beta} : B}{\Gamma \vdash \lambda x\, s^{(\overline{\alpha},\beta)} : A \Rightarrow B}\ [\![\text{Abs}]\!]$$

$$\frac{\Gamma_0 \vdash s^{([\alpha_1,\ldots,\alpha_k],\beta)} : A \Rightarrow B \qquad \Gamma_1 \vdash t^{\alpha_1} : A \qquad \cdots \qquad \Gamma_k \vdash t^{\alpha_k} : A}{\sum_{j=0}^{k} \Gamma_j \vdash s\, t^{\beta} : B}\ [\![\text{App}]\!]$$

$$\frac{\Gamma \vdash s_i^{\alpha} : A_i}{\Gamma \vdash \langle s_1, s_2\rangle^{(i,\alpha)} : A_1 \,\&\, A_2}\ [\![\text{Pair}_i]\!] \qquad \frac{\Gamma \vdash s^{(1,\alpha)} : A \,\&\, B}{\Gamma \vdash \pi_1 s^{\alpha} : A}\ [\![\text{Left}]\!]$$

$$\frac{\Gamma \vdash s^{(2,\beta)} : A \,\&\, B}{\Gamma \vdash \pi_2 s^{\beta} : B}\ [\![\text{Right}]\!]$$

Figure 2: Computing points in the relational semantics

typing judgement $x_1 : A_1, \ldots, x_n : A_n \vdash s : A$ will be an $n$-ary multirelation $[\![s]\!]_{x_1:A_1,\ldots,x_n:A_n} \subseteq [\![A_1 \Rightarrow \cdots \Rightarrow A_n \Rightarrow A]\!]$. We first introduce the deductive system of Figure 2, which is a straightforward adaptation of de Carvalho's system $R$ (2008) to the simply typed case. In this system, derivable judgements are semantic annotations of typing judgements: $x_1^{\overline{\alpha}_1} : A_1, \ldots, x_n^{\overline{\alpha}_n} : A_n \vdash s^{\alpha} : A$ stands for $(\overline{\alpha}_1, \ldots, \overline{\alpha}_n, \alpha) \in [\![s]\!]_{x_1:A_1,\ldots,x_n:A_n}$ where each $\overline{\alpha}_i \in \mathfrak{M}_{\mathrm{f}}([\![A_i]\!])$ and $\alpha \in [\![A]\!]$. In rules $[\![\text{Var}]\!]$ and $[\![\text{Const}]\!]$, $\Gamma^{[]}$ denotes an annotated context of the form $x_1^{[]} : A_1, \ldots, x_n^{[]} : A_n$. In rule $[\![\text{App}]\!]$, the sum of annotated contexts is defined pointwise: $\left(x_1^{\overline{\alpha}_1} : A_1, \ldots, x_n^{\overline{\alpha}_n} : A_n\right) + \left(x_1^{\overline{\alpha}'_1} : A_1, \ldots, x_n^{\overline{\alpha}'_n} : A_n\right) = \left(x_1^{\overline{\alpha}_1 + \overline{\alpha}'_1} : A_1, \ldots, x_n^{\overline{\alpha}_n + \overline{\alpha}'_n} : A_n\right)$. The semantics of a term is given by: $[\![s]\!]_{x_1:A_1,\ldots,x_n:A_n} = \left\{(\overline{\alpha}_1, \ldots, \overline{\alpha}_n, \alpha);\ x_1^{\overline{\alpha}_1} : A_1, \ldots, x_n^{\overline{\alpha}_n} : A_n \vdash s^{\alpha} : A\right\}$. Notice that there is no rule for $\langle\rangle$ in Figure 2, because $[\![\langle\rangle]\!]_{\Gamma} = \emptyset$. Since we follow the standard interpretation of typed $\lambda$-calculi in cartesian closed categories (see Lambek and Scott (1986)), in the particular case of $\underline{\mathrm{Rel}}^{!}$, we obtain:

**Lemma 5.1 (Invariance)** *If $\Gamma \vdash s \simeq_0 t : A$ then $[\![s]\!]_{\Gamma} = [\![t]\!]_{\Gamma}$.*

For all finiteness spaces $\mathcal{A}$ and $\mathcal{B}$, write $\mathcal{A} \Rightarrow \mathcal{B} = !\mathcal{A} \multimap \mathcal{B}$.

**Lemma 5.2** *Let $f$ be a multirelation from $|\mathcal{A}|$ to $|\mathcal{B}|$. Then $f \in \mathfrak{F}(\mathcal{A} \Rightarrow \mathcal{B})$ iff, for all $a \in \mathfrak{F}(\mathcal{A})$, $f \cdot a^{!} \in \mathfrak{F}(\mathcal{B})$ and for all $\beta \in |\mathcal{B}|$, ${}^{t}f \cdot \beta \perp_{\mathrm{f}} a^{!}$.*

**Proof** By Lemma 3.8, $f \in \mathfrak{F}(\mathcal{A} \Rightarrow \mathcal{B})$ iff for all $\overline{a} \in \mathfrak{F}(!\mathcal{A})$, $f \cdot \overline{a} \in \mathfrak{F}(\mathcal{B})$ and for all $\beta \in |\mathcal{B}|$, ${}^{t}f \cdot \beta \perp_{\mathrm{f}} \overline{a}$. By the characterization of $!\mathcal{A}$ given in Example 3.5:

$$\mathfrak{F}(!\mathcal{A}) = \left\{\overline{a} \subseteq !|\mathcal{A}|;\ \mathsf{supp}^{|\mathcal{A}|} \cdot \overline{a} \in \mathfrak{F}(\mathcal{A})\right\} = \left\{a^{!};\ a \in \mathfrak{F}(\mathcal{A})\right\}^{\perp\perp_{|\mathcal{A}|}}.$$

Then the result follows from the inclusions $\left\{a^{!};\ a \in \mathfrak{F}(\mathcal{A})\right\} \subseteq \mathfrak{F}(!\mathcal{A})$ and $a \subseteq \mathit{supp}\left(a^{!}\right)$ for all $a \subseteq |\mathcal{A}|$. □

We call *finitary multirelations* from $\mathcal{A}$ to $\mathcal{B}$ the elements of $\mathfrak{F}(\mathcal{A} \Rightarrow \mathcal{B})$. Then the category $\underline{\mathrm{Fin}}^!$ of finiteness spaces and finitary multirelations is no other than the co-Kleisli category derived from $\underline{\mathrm{Fin}}$. The relational interpretation $[\![\cdot]\!]$ of simply typed $\lambda$-calculi thus defines a semantics in $\underline{\mathrm{Fin}}^!$ as follows. Assume a finiteness structure $\mathfrak{F}(X)$ on $[\![X]\!]$ is given for all atomic type $X$, and write $X^*$ for the finiteness space $([\![X]\!], \mathfrak{F}(X))$. We set $(A \Rightarrow B)^* = A^* \Rightarrow B^*$, $(A \,\&\, B)^* = A^* \,\&\, B^*$ and $\top^* = \top$. Then, further assuming that, for all $a \in \mathfrak{C}_A$, $[\![a]\!] \in \mathfrak{F}(A^*)$, we obtain:

**Lemma 5.3 (Finiteness)** *If* $x_1 : A_1, \ldots, x_n : A_n \vdash s : A$ *then*

$$[\![s]\!]_{x_1:A_1,\ldots,x_n:A_n} \in \mathfrak{F}(A_1^* \Rightarrow \cdots \Rightarrow A_n^* \Rightarrow A^*)\,.$$

## 5.2 On the relations denoted by $\lambda$-terms

Pure typed $\lambda$-calculi are those with no additional constant or conversion rule: fix a set $\mathfrak{A}$ of atomic types, and write $\Lambda_0^{\mathfrak{A}}$ for the calculus where $\mathfrak{C}_A = \emptyset$ for every type $A$, and $s \simeq t$ iff $s \simeq_0 t$. This is the most basic case and we have just shown that $\underline{\mathrm{Rel}}^!$ and $\underline{\mathrm{Fin}}^!$ model $\simeq_0$. Be aware that if we introduce no atomic type, then the semantics is actually trivial: in $\Lambda_0^{\emptyset}$, all types and terms are interpreted by $\emptyset$.

By contrast, we can consider the internal language $\Lambda_{\underline{\mathrm{Rel}}}$ of $\underline{\mathrm{Rel}}^!$ in which all relations can be described as terms: fix the atomic types $\mathfrak{A}$ as the collection of all sets and the constants $\mathfrak{C}_A = \mathfrak{P}([\![A]\!])$. Then set $s \simeq_{\underline{\mathrm{Rel}}} t$ iff $[\![s]\!]_\Gamma = [\![t]\!]_\Gamma$, for any suitable $\Gamma$. The point in defining such a monstrous language is to enable very natural notations for relations: in general, we will identify closed terms in $\Lambda_{\underline{\mathrm{Rel}}}$ with the relations they denote in the empty context. For instance, we write $\mathsf{der}^A = \lambda x\, x$ with $x$ of type $A$; and if $f \in \underline{\mathrm{Rel}}^!(A, B)$ and $g \in \underline{\mathrm{Rel}}^!(B, C)$, we have $g \circ^! f = \lambda x\,(g\,(f\,x))$. More generally, if $s$ and $t$ are terms in $\Lambda_{\underline{\mathrm{Rel}}}$ of type $A$ in context $\Gamma$, we may simply write $\Gamma \vdash s = t : A$ for $[\![s]\!]_\Gamma = [\![t]\!]_\Gamma \in [\![A]\!]$. Similarly, the internal language $\Lambda_{\underline{\mathrm{Fin}}}$ of $\underline{\mathrm{Fin}}^!$, where $\mathfrak{A}$ is the collection of all finiteness spaces and $\mathfrak{C}_A = \mathfrak{F}(A^*)$, allows to denote conveniently all finitary relations and equations between them.

Before we address the problem of algebraic types, we review some basic properties of the semantics. First, $\underline{\mathrm{Rel}}^!$ and $\underline{\mathrm{Fin}}^!$ being cartesian closed categories, they actually model typed $\lambda$-calculi with extensionality: $s : A \Rightarrow A \vdash \lambda x\,(s x) = s : A \Rightarrow A$ as soon as $x$ is not free in $s$. Moreover, they admit all products, and they are models of $\lambda$-calculi with surjective tuples of arbitrary arity, that is $t : \mathop{\&}\overrightarrow{A} \vdash \langle \pi_i t \rangle_{i \in I} = t$. In accordance with this last remark, we may identify any variable of type $\mathop{\&}\overrightarrow{A}$ with a tuple and write, e.g., $\pi_i = \lambda \overrightarrow{x}\, x_i$.

Being cpo-enriched, $\underline{\mathrm{Rel}}^!$ admits fixpoints at all types and the least fix point operator on $A$ is the least multirelation $\mathsf{fix} \subseteq (A \Rightarrow A) \Rightarrow A$ such that $\mathsf{fix} = \lambda f\,(f\,(\mathsf{fix}\, f))$, i.e. $\mathsf{fix} = \bigcup_{n \in \mathbf{N}} \mathsf{fix}_n$ where $\mathsf{fix}_0 = \emptyset$ and $\mathsf{fix}_{n+1} = \lambda f\,(f\,(\mathsf{fix}_n\, f))$ (see for instance Amadio and Curien (1998, Chapter 6)). More explicitly, we get:

$$\mathsf{fix}_{n+1} = \left\{ \left( [([\alpha_1, \ldots, \alpha_p], \alpha)] + \sum_{k=1}^{p} \overline{\phi}_k, \alpha \right);\ p \in \mathbf{N} \wedge \forall k \in \{1, \ldots, p\},\ (\overline{\phi}_k, \alpha_k) \in \mathsf{fix}_n \right\}.$$

Notice that, for all $n \in \mathbf{N}$ and all finiteness space $\mathcal{A}$, $\mathsf{fix}_n^{|\mathcal{A}|} \in \mathfrak{F}((\mathcal{A} \Rightarrow \mathcal{A}) \Rightarrow \mathcal{A})$. But in general, $\mathsf{fix}$ is not finitary: Ehrhard (2005, Section 3) details a counter-example, but we can actually show that the least fixpoint operator is never finitary on non-empty webs (and thus no fixpoint operator is, since finiteness structures are downward closed for inclusion).

**Lemma 5.4** *If* $|\mathcal{A}| \neq \emptyset$, *then* $\mathsf{fix}^{|\mathcal{A}|} \notin \mathfrak{F}((\mathcal{A} \Rightarrow \mathcal{A}) \Rightarrow \mathcal{A})$.

**Proof** Let $\alpha \in |\mathcal{A}|$ and $f = \{([\,], \alpha), ([\alpha], \alpha)\} \in \mathfrak{P}_{\mathrm{f}}(\mathcal{A} \Rightarrow \mathcal{A}) \subseteq \mathfrak{F}(\mathcal{A} \Rightarrow \mathcal{A})$. Observe that $([([\,], \alpha)], \alpha) \in \mathsf{fix}_1$, $([([\,], \alpha), ([\alpha], \alpha)], \alpha) \in \mathsf{fix}_2$, and more generally $([([\,], \alpha)] + n\,[([\alpha], \alpha)], \alpha) \in \mathsf{fix}_{n+1}$. Hence $f^! \not\perp_{\mathrm{f}} {}^t\mathsf{fix} \cdot \alpha$ and we conclude by Lemma 5.2. □

This result indicates that the finitary semantics refuses infinite computations and will not accomodate general recursion, e.g. in the sense of PCF. It is thus very natural to investigate the nature of the algorithms that can be studied in a finitary setting. It was already known from Ehrhard's original paper (2005) that one can model a restricted form of tail-recursive iteration. In recent work (Vaux; 2009c), the second author showed that the finitary relational model of the $\lambda$-calculus can actually be extended to Gödel's system $T$, i.e. typed recursion on integers. The remaining of the paper provides a generalization of this result to recursive algebraic datatypes.

# 6 Lazy recursive algebraic datatypes

An algebraic datatype is a composite of products, sums and base types: products are equipped with projections and a tupling operation (i.e. pairing, in the binary case), while sums are equipped with injections and a case definition operator (which is essentially a pattern matching operator). Of course, datatype constructors are meant to be polymorphic: in other words they are particular functors. In a cartesian closed category, it is only natural to interpret products as categorical products. On the other hand, coproducts are not always available, hence the interpretation of sums might not be as canonical.

In this concluding section of our paper, we first discuss the status of sums in $\underline{\mathrm{Rel}}^!$ and $\underline{\mathrm{Fin}}^!$. We are then led to investigate the semantics of recursive algebraic datatypes we obtain by taking the fixpoints of algebraic functors. In particular, we remark that the relational interpretation gives rise to a *lazy* semantics. For instance the web of the datatype of trees is not a set of trees but a set of paths in trees: this generalizes a similar feature of the coherence semantics of system $T$ (Girard et al.; 1989) and its relational variant (Vaux; 2009c). We finish the paper by providing an explicit description of the relational interpretation of the constructors and destructors of recursive algebraic datatypes, which enables us to prove them finitary.

## 6.1 Sums

By contrast with the cartesian structure, the cocartesian structure is ruled out by the co-Kleisli construction from $\underline{\mathrm{Rel}}$ to $\underline{\mathrm{Rel}}^!$ (as by the one from $\underline{\mathrm{Fin}}$ to $\underline{\mathrm{Fin}}^!$): $\underline{\mathrm{Rel}}^!$ does not have coproducts.

**Counter-example 6.1** *There is no coproduct for the pair of sets $(\emptyset,\emptyset)$ in $\underline{\mathrm{Rel}}^!$. Indeed, assume that there exists a set $A$ and multirelations $i_0$ and $i_1$ from $\emptyset$ to $A$, such that for all set $B$ and all multirelations $f_0$ and $f_1$ from $\emptyset$ to $B$ there exists a unique $h \in \underline{\mathrm{Rel}}^!(A,B)$ such that $h \circ^! i_k = f_k$ for $k = 0,1$. Necessarily, there exist $\alpha_0$ and $\alpha_1$, such that $([]\,,\alpha_k) \in i_k$ but $([]\,,\alpha_k) \notin i_{1-k}$ for $k = 0,1$: otherwise, e.g., $h \circ^! i_0 \subseteq h \circ^! i_1$ for all $h$. Now consider relations from $A$ to $\{0\}$, $h' = \{([\alpha_0]\,,0),([\alpha_1]\,,0)\}$ and $h'' = \{([]\,,0)\}$: we have $h' \circ^! i_k = h'' \circ^! i_k$ for $k = 0,1$ but $h' \neq h''$, which contradicts the unicity property of the coproduct.*

We can however provide an adequate interpretation of sum types, adapting Girard's interpretation of intuitionistic logic in coherence spaces (Girard et al.; 1989). We write $A \overset{\bullet}{\oplus} B$ for the lifted sum $\{1,2\} \cup A \oplus B$ of $A$ and $B$, and more generally: $\overset{\bullet}{\bigoplus} \overrightarrow{A} = I \cup \bigoplus \overrightarrow{A}$.[5] The idea is that indices stand for tokens without associated value: where $(i,\alpha)$ can be read as "the element $\alpha$ in $A_i$", $i$ represents some undetermined element of which we only know it is in $A_i$. Then, for all $i \in I$, we set $\mathsf{inj}_i^{\overrightarrow{A}} = \{([]\,,i)\} \cup \{([\alpha]\,,(i,\alpha));\ \alpha \in A_i\}$. Moreover, if $\overrightarrow{f}$ is a relation from $\overrightarrow{A}$ to $\overrightarrow{B}$, we set $\overset{\bullet}{\bigoplus} \overrightarrow{f} = \mathsf{id}^I \cup \bigoplus \overrightarrow{f}$ so that $\overset{\bullet}{\bigoplus}$ is a continuous $I$-ary functor from $\underline{\mathrm{Rel}}^I$ to $\underline{\mathrm{Rel}}$. For all $i \in I$ and $\overline{\alpha} = [\alpha_1,\dots,\alpha_n] \in\, !A_i$, we write $i\overline{\alpha}$ for $[(i,\alpha_1),\dots,(i,\alpha_n)] \in\, !\overset{\bullet}{\bigoplus} \overrightarrow{A}$. Then, for all family $\overrightarrow{f}$ of multirelations such that $f_i \in \underline{\mathrm{Rel}}^!(A_i,B)$ for all $i \in I$, we define $\left\{\overrightarrow{f}\right\} = \{([i] + i\overline{\alpha},\beta);\ i \in I \wedge (\overline{\alpha},\beta) \in f_i\}$ and obtain $\left\{\overrightarrow{f}\right\} \circ^! \mathsf{inj}_i = f_i$ for all $i \in I$.

[5] Another possibility for interpreting sums is to consider $A \oplus^! B = !A \oplus !B$ which is preferred by Girard to interpret intuitionistic disjunction because it enjoys an extensionality property. There is no doubt we could adapt the following sections of our paper to this notion of sum.

Notice however that $\left\{ \overrightarrow{f} \right\}$ is not characterized by this property, since we have already remarked that $\overset{\bullet}{\bigoplus}$ is not a coproduct in $\underline{\mathrm{Rel}}^!$. For instance, $\{([i,i]+i\overline{\alpha},\beta);\ (\overline{\alpha},\beta)\in f_i\}$ behaves similarly (we just added a copy of the token $i\in I$). This *case definition* construction can be internalized as a multirelation, by setting $\mathsf{case}^{\overrightarrow{A},B}=\lambda\overrightarrow{f}\left\{\overrightarrow{f}\right\}\subseteq \&_{i\in I}\,(A_i\Rightarrow B)\Rightarrow\overset{\bullet}{\bigoplus}\overrightarrow{A}\Rightarrow B$. More explicitly:

$$\mathsf{case}^{\overrightarrow{A},B}=\{([(i,(\overline{\alpha},\beta))]\,,[i]+i\overline{\alpha},\beta)\,;\ i\in I\wedge\overline{\alpha}\in !A_i\wedge\beta\in B\}$$

and we obtain:

**Lemma 6.2** *For all $j\in I$, $\overrightarrow{f}:\&_{i\in I}\,(A_i\Rightarrow B)\,,s:A_j\vdash\mathsf{case}\,\overrightarrow{f}\,\left(\mathsf{inj}_j\,s\right)=f_j\,s:B$.*

This provides a lazy implementation of sum types. For instance, we have

$$\mathsf{case}^{\overrightarrow{A},B}\,\langle\lambda x\,b_i\rangle_{i\in I}\,\left(\mathsf{inj}_j\,a\right)=b_j$$

for all $\overrightarrow{b}\in\mathfrak{P}\,(B)^I$, $j\in I$ and $a\subseteq A_j$, even when $a$ is undefined, i.e. $a=\emptyset$.

For all $i\in I$, the restriction $\mathsf{rest}_i^{\overrightarrow{A}}$ is a quasi-functional lax natural transformation from $\overset{\bullet}{\bigoplus}$ to $\Pi_i$. The same holds for the index relation $\mathsf{indx}\cup\mathsf{id}^I$ from $\overset{\bullet}{\bigoplus}$ to $E_I$. We thus have a transport situation, which moreover defines a functor $\overset{\bullet}{\bigoplus}$ from $\underline{\mathrm{Fin}}^I$ to $\underline{\mathrm{Fin}}$, because it admits a shape relation: $\mathsf{indx}$ itself (see Lemma 3.11). We obtain $\left|\overset{\bullet}{\bigoplus}\overrightarrow{\mathcal{A}}\right|=\overset{\bullet}{\bigoplus}\overrightarrow{|\mathcal{A}|}$ and $\mathfrak{F}\left(\overset{\bullet}{\bigoplus}\overrightarrow{\mathcal{A}}\right)=\left\{J\cup\sum_{i\in K}a_i;\ J\cup K\subseteq_{\mathrm{f}} I\wedge\forall i\in K,\ a_i\in\mathfrak{F}\,(\mathcal{A}_i)\right\}$. This defines a functor suitable to interpret sum types in $\underline{\mathrm{Fin}}^!$ (although not a coproduct) because injections and the case definition operator are finitary:

**Lemma 6.3** *For all finiteness spaces $\overrightarrow{\mathcal{A}}$ and $\mathcal{B}$, $\mathsf{inj}_i\in\mathfrak{F}\left(\mathcal{A}_i\Rightarrow\overset{\bullet}{\bigoplus}\overrightarrow{\mathcal{A}}\right)$ and*

$$\mathsf{case}\in\mathfrak{F}\left(\overset{\bullet}{\bigoplus}\overrightarrow{\mathcal{A}}\Rightarrow\mathop{\&}_{i\in I}\,(\mathcal{A}_i\Rightarrow\mathcal{B})\Rightarrow\mathcal{B}\right).$$

**Proof** This is a direct application of the definitions. □

We call *algebraic datatype* any functor built from projection functors, $\top$, $\&$ and $\overset{\bullet}{\bigoplus}$. The most basic example of composite datatype is that of booleans, $\mathcal{B}ool=\top\overset{\bullet}{\oplus}\top$: assuming this lifted sum is indexed by the two point set $\{tt,ff\}$, $\mathcal{B}ool$ is the only finiteness space with $|\mathcal{B}ool|=\{tt,ff\}$. The injections $\mathsf{inj}_{tt}^{(\emptyset,\emptyset)}=\{([]\,,tt)\}$ and $\mathsf{inj}_{ff}^{(\emptyset,\emptyset)}=\{([]\,,ff)\}$ are constant multirelations: up to the isomorphism $\emptyset\Rightarrow|\mathcal{B}ool|\cong|\mathcal{B}ool|$, we thus consider their respective images $\mathsf{true}=\{tt\}\in\mathfrak{F}\,(\mathcal{B}ool)$ and $\mathsf{false}=\{ff\}\in\mathfrak{F}\,(\mathcal{B}ool)$ as the constructors of $\mathcal{B}ool$. Similarly, the case definition $\mathsf{case}^{(\emptyset,\emptyset),A}=\{([x]\,,[(x,([]\,,\alpha))]\,,\alpha)\,;\ x\in\{tt,ff\}\wedge\alpha\in A\}$ corresponds with the conditional $\mathsf{if}^A=\{([tt]\,,[\alpha]\,,[]\,,\alpha);\ \alpha\in A\}\cup\{([ff]\,,[]\,,[\alpha]\,,\alpha);\ \alpha\in A\}$ up to the isomorphisms $\emptyset\Rightarrow A\cong A$ and $A\,\&\,A\Rightarrow A\cong A\Rightarrow A\Rightarrow A$, so that $\mathsf{if}^{|\mathcal{A}|}\in\mathfrak{F}\,(\mathcal{B}ool\Rightarrow\mathcal{A}\Rightarrow\mathcal{A}\Rightarrow\mathcal{A})$ for all finiteness space $\mathcal{A}$. Of course, we obtain $\mathsf{iftrue}=\lambda x\,\lambda y\,x$ and $\mathsf{iffalse}=\lambda x\,\lambda y\,y$.

## 6.2 Tree types as fixpoints

Algebraic datatypes are $\preceq$-increasing functors, and both exactly continuous and directly continuous. Hence they admit least fixpoints, which are obtained as exact suprema by Lemma 4.7. We investigate how this construction could provide an interpretation of recursive algebraic datatypes. We may first consider a finiteness space of trees:

**Counter-example 6.4** *Let $\mathcal{A}$ and $\mathcal{B}$ be finiteness spaces and $\mathcal{T}:\mathcal{X}\mapsto\mathcal{A}\oplus(\mathcal{X}\otimes\mathcal{B}\otimes\mathcal{X})$. We can describe the least fixpoint $\mu\,\mathcal{T}$ as follows:*

- *$|\mu\mathcal{T}|$ is the set of all finite binary trees, with leaves labelled by elements of $|\mathcal{A}|$ and nodes labelled by elements of $|\mathcal{B}|$;*
- *a set $t$ of trees is finitary in $\mu\mathcal{T}$ when the set of all the labels of nodes (resp. leaves) of trees in $t$ is finitary in $\mathcal{B}$ (resp. $\mathcal{A}$) and moreover the height of trees in $t$ is bounded.*

*Moreover, $\mu\mathcal{T}$ is functorial in variables $\mathcal{A}$ and $\mathcal{B}$ because, by the above description, it can be defined directly as a transport functor. It should not however be considered as the datatype of binary trees with nodes of type $\mathcal{B}$ and leaves of type $\mathcal{A}$. Indeed, since this type relies on $\oplus$ which does not define a sum, we would also fail to define a suitable relational interpretation of pattern matching for this type of trees. Notice that this is not related with a finiteness argument: the same would hold for the relational model (or the coherence model for that matter).*

In light of this example, of the discussion on sums and of previous work on the semantics of system $T$ (Vaux; 2009c), we are led to study the finiteness properties of the datatypes of trees obtained as fixpoints of power series functors:

**Definition 6.5** *Let $I$ be a set of indices, $\overrightarrow{\mathcal{A}} = (\mathcal{A}_i)_{i\in I}$ a family of finiteness spaces and $\overrightarrow{J} = (J_i)_{i\in I}$ a family of sets of indices. We write $\mathcal{L}^{I,\overrightarrow{J}}\overrightarrow{\mathcal{A}}$ for the least fixpoint of the algebraic functor $\mathcal{X} \mapsto \bigoplus^{\bullet}_{i\in I} \mathcal{A}_i \,\&\, \mathcal{X}^{\&J_i}$, where $\mathcal{X}^{\&J}$ denotes $\&_{j\in J}\,\mathcal{X}$.*

We will in general simply write $\mathcal{L}\overrightarrow{\mathcal{A}}$ for $\mathcal{L}^{I,\overrightarrow{J}}\overrightarrow{\mathcal{A}}$. Intuitively $\mathcal{L}\overrightarrow{\mathcal{A}}$ is the datatype of trees, in which nodes of sort $i \in I$ are of arity $J_i$ and bear labels in $\mathcal{A}_i$. More precisely, we will show that $\left|\mathcal{L}\overrightarrow{\mathcal{A}}\right|$ is the set of paths starting from the root in such trees. Recall that terms are in general interpreted by subsets of the web of their type: the subsets interpreting terms of type $\mathcal{L}\overrightarrow{\mathcal{A}}$ will be the sets of paths in the corresponding trees. We will moreover obtain that these interpretations are all finitary.

In order to describe $\left|\mathcal{L}^{I,\overrightarrow{J}}\overrightarrow{\mathcal{A}}\right|$, we introduce the associated construction $L^{I,\overrightarrow{J}}$ in $\underline{\mathrm{Rel}}$: $L^{I,\overrightarrow{J}}\overrightarrow{A}$ is defined as the least fixpoint of the continuous functor $T : X \mapsto \bigoplus^{\bullet}_{i\in I} A_i \,\&\, X^{\&J_i}$, i.e. $L_{I,\overrightarrow{J}}\overrightarrow{A} = \bigcup_{n\in\mathbf{N}} T^n\emptyset$, which we simply write $L\overrightarrow{A}$ in general. Since $T$ is actually an $(I+1)$-ary continuous functor, $L$ is itself an $I$-ary continuous functor in $\underline{\mathrm{Rel}}$. Before we inspect the general form of $L\overrightarrow{A}$, let us first give an intuitive account of the binary case:

**Example 6.6** *Consider the finiteness space $\mathcal{BT} = \mathcal{L}^{I,\overrightarrow{J}}$ obtained by Definition 6.5, where we set $I = \{\mathsf{F},\mathsf{N}\}$, $J_{\mathsf{F}} = \emptyset$ and $J_{\mathsf{N}} = \{\mathsf{G},\mathsf{D}\}$: this is the least fixpoint of the functor $\mathcal{X} \mapsto \mathcal{A} \overset{\bullet}{\oplus} (\mathcal{B} \,\&\, (\mathcal{X} \,\&\, \mathcal{X}))$. This is meant to represent the datatype of binary trees with leaves of type $\mathcal{A}$ and nodes of type $\mathcal{B}$. The various indices can be interpreted as follows: $\mathsf{F}$ denotes a leaf whereas $\mathsf{N}$ denotes an internal node; $\mathsf{G}$ denotes the left child of a node, whereas $\mathsf{D}$ denotes its right child. Then the elements of $|\mathcal{BT}|$ are sequences of the following four shapes:*

- $(\mathsf{N},(2,(j_1,(\mathsf{N},(2,(j_2,\ldots,(\mathsf{N},(2,(j_n,\mathsf{N})))\cdots)))))))$ *where $j_k \in \{\mathsf{G},\mathsf{D}\}$ for all $k$, which denotes a path to an internal node;*
- $(\mathsf{N},(2,(j_1,(\mathsf{N},(2,(j_2,\ldots,(\mathsf{N},(2,(j_n,(\mathsf{N},(1,\beta)))))\cdots)))))))$ *where $\beta \in |\mathcal{B}|$, which denotes a path to an internal node, with a value in the interpretation of this node;*
- $(\mathsf{N},(2,(j_1,(\mathsf{N},(2,(j_2,\ldots,(\mathsf{N},(2,(j_n,\mathsf{F})))\cdots)))))))$, *which denotes a path to a leaf;*
- $(\mathsf{N},(2,(j_1,(\mathsf{N},(2,(j_2,\ldots,(\mathsf{N},(2,(j_n,(\mathsf{F},\alpha))))\cdots)))))))$ *where $\alpha \in |\mathcal{A}|$, which denotes a path to a leaf, with a value in the interpretation of the label of the leaf.*

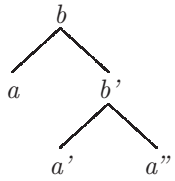
*It makes only sense to adopt a more compact notation and write, e.g., $\mathsf{N}j_1\mathsf{N}j_2\cdots\mathsf{N}j_n\mathsf{N}$ for*

$$(\mathsf{N},(2,(j_1,(\mathsf{N},(2,(j_2,\ldots,(\mathsf{N},(2,(j_n,\mathsf{N})))\cdots)))))),$$

*and similarly* $\mathsf{N}j_1\mathsf{N}j_2\cdots\mathsf{N}j_n\mathsf{N}\beta$ *for*

$$(\mathsf{N},(2,(j_1,(\mathsf{N},(2,(j_2,\ldots,(\mathsf{N},(2,(j_n,(\mathsf{N},(1,\beta)))))\cdots)))))).$$

*Then let* $a,a',a''$ *be constants of type* $\mathcal{A}$ *and* $b,b'$ *be constants of type* $\mathcal{B}$*, i.e.* $a,a',a''\in\mathfrak{F}(\mathcal{A})$ *and* $b,b'\in\mathfrak{F}(\mathcal{B})$*. The tree:*




*will be interpreted by the subset:*

$$\{\mathsf{N}\}\cup\{\mathsf{N}\beta;\ \beta\in b\}\cup\{\mathsf{NGF}\}\cup\{\mathsf{NGF}\alpha;\ \alpha\in a\}\cup\{\mathsf{NDN}\}\cup\{\mathsf{NDN}\beta;\ \beta\in b'\}$$
$$\cup\{\mathsf{NDNGF}\}\cup\{\mathsf{NDNGF}\alpha;\ \alpha\in a'\}\cup\{\mathsf{NDNDF}\}\cup\{\mathsf{NDNDF}\alpha;\ \alpha\in a''\}.$$

Let us turn to the general case. By its definition, $L\overrightarrow{A}$ is the least set such that: $I\subseteq L\overrightarrow{A}$; $(i,(1,\alpha))\in L\overrightarrow{A}$ for all $i\in I$ and $\alpha\in A_i$; and $(i,(2,(j,\tau)))\in L\overrightarrow{A}$ for all $i\in I$, $j\in J_i$ and $\tau$ in $L\overrightarrow{A}$. Hence the general form of an element $\tau\in L\overrightarrow{A}$ is: $\tau=(i_1,(2,(j_1,\ldots(j_n,i_{n+1})\cdots)))))$ or $\tau=(i_1,(2,(j_1,\ldots(j_n,(i_{n+1},(1,\alpha)))\cdots)))))$ where $j_k\in J_{i_k}$ for all $k\leq n$ and, in the second case, $\alpha\in A_{i_{n+1}}$. As in the above example, we introduce the following conventions for the sole purpose of making this description of the elements $L\overrightarrow{A}$ more reasonable. We call *addresses* all finite sequences $i_1j_1i_2j_2\cdots i_nj_n$ such that $j_k\in J_{i_k}$ for all $k\leq n$ and write $\mathbf{A}$ for the set of all addresses. We call *value* any element $\nu$ of $\overset{\bullet}{\bigoplus}\overrightarrow{A}$. We say $\nu$ is of type $i\in I$ if $\nu=i$ or $\nu=(i,\alpha)$ with $\alpha\in A_i$: we then write $type\,(\nu)=i$. A *path* is the data $\pi\nu$ of an address and a value. We may factor prefixes out of multisets of paths or addresses: for instance, if $\overline{\tau}=[\tau_1,\ldots,\tau_n]$ is a multiset of paths, we may write $ij\overline{\tau}=[ij\tau_1,\ldots,ij\tau_n]$. Then $L\overrightarrow{A}$ is in bijection with the set of all paths: from now on we consider $L\overrightarrow{A}$, and thus $\mathcal{L}\overrightarrow{\mathcal{A}}$, up to this bijection.

Notice that the relation $\mathsf{val}_i^{\overrightarrow{A}}=\{(\pi i\alpha,\alpha);\ \pi\in\mathbf{A}\wedge\alpha\in A_i\}$ is a quasi-functional lax natural transformation from $L$ to the projection functor $\Pi_i$, for all $i\in I$. Moreover, the relation $\mathsf{len}^{\overrightarrow{A}}=\left\{(i_1j_1\cdots i_nj_n\nu,n);\ n\in\mathbf{N}\wedge i_1j_1\cdots i_nj_n\nu\in L\overrightarrow{A}\right\}$ is a quasi-functional lax natural transformation from $L$ to $E_{\mathbf{N}}$ where $\mathbf{N}$ is the functor of shapes defined by natural numbers (see Section 3.4).

**Example 6.7** *In the setting of Example 6.6, we obtain:* $\mathsf{val}_{\mathsf{N}}\cdot t=b\cup b'$, $\mathsf{val}_{\mathsf{F}}\cdot t=a\cup a'\cup a''$ *and* $\mathsf{len}\cdot t=\{0,1,2\}$.

We have thus given a precise account of the web $\left|\mathcal{L}\overrightarrow{\mathcal{A}}\right|=L|\overrightarrow{\mathcal{A}}|$. Moreover, since $\mathcal{L}\overrightarrow{\mathcal{A}}$ is defined as the least fixpoint of the algebraic functor $\mathcal{T}$ given in Definition 6.5, and this fixpoint is an exact supremum, we obtain: $\mathfrak{F}\left(\mathcal{L}\overrightarrow{\mathcal{A}}\right)=\bigcup_{n\in\mathbf{N}}\mathfrak{F}(\mathcal{T}^n\mathbf{0})$. We can thus characterize this finiteness structure as follows:

**Lemma 6.8** *Let* $t$ *be a set of paths. Then* $t\in\mathfrak{F}\left(\mathcal{L}\overrightarrow{\mathcal{A}}\right)$ *iff* $\mathsf{len}\cdot t\in\mathfrak{P}_{\mathrm{f}}(\mathbf{N})$ *and* $\mathsf{val}_i\cdot t\in\mathfrak{F}(A_i)$ *for all* $i\in I$*. Moreover, if* $\mathsf{len}\cdot t$ *is finite and, for all* $i\in I$, $\mathsf{val}_i\cdot$ *is finite, then* $t$ *is itself finite.*

We could thus have presented $\mathcal{L}$ equivalently as the functor of paths, with web functor $L$, finiteness structure being transported by $\overrightarrow{\mathsf{val}}$ and $\mathsf{len}$ (see Lemma 3.11). It is important to notice that only the above careful explicitation of the structure of $\mathcal{L}$ allowed us to deduce this functoriality. At the time of writing, it is unclear to us whether this technique generalizes to a larger class of transport functors.

**Example 6.9** *Lemma 6.8 implies that the interpretation of the binary tree of Example 6.6 is finitary in* $\mathcal{BT}$*.*

## 6.3 The finitary datatype of trees

We are now ready to describe the interpretation of the datatype of trees:

- $\mathcal{L}$ provides a *lazy* implementation of the datatype of trees where nodes of type $i$ bear labels in $A_i$ and have arity $J_i$;
- this implementation is *finitary* in the sense that constructors, destructors and iterators on trees are finitary relations.

The *lazy tree constructor* $\mathsf{node}_i \subseteq A_i \Rightarrow \left(L\overrightarrow{A}\right)^{\&J_i} \Rightarrow L\overrightarrow{A}$ is given by:

$$\mathsf{node}_i = \{([]\,,[]\,,i)\} \cup \{([\alpha]\,,[]\,,i\alpha);\ \alpha \in |\mathcal{A}_i|\} \cup \left\{([]\,,[(j,\tau)]\,,ij\tau);\ j \in J_i \wedge \tau \in L\overrightarrow{A}\right\}$$

which is actually an instance of

$$\mathsf{inj}_i \subseteq \left(A_i \,\&\, \left(L\overrightarrow{A}\right)^{\&J_i}\right) \Rightarrow L\overrightarrow{A}$$

up to our notations of addresses and the cartesian adjunction in $\underline{\mathrm{Rel}}^!$. Since $\mathsf{inj}_i$ is finitary, we moreover obtain:

$$\mathsf{node}_i \in \mathfrak{F}\left(\mathcal{A}_i \Rightarrow \left(\mathcal{L}\overrightarrow{\mathcal{A}}\right)^{\&J_i} \Rightarrow \mathcal{L}\overrightarrow{\mathcal{A}}\right)$$

for all family $\overrightarrow{\mathcal{A}}$ of finiteness spaces.

**Example 6.10** *Recall the finiteness space $\mathcal{BT}$ of Example 6.6. Notice that $\mathsf{node}_\mathsf{F} \subseteq \mathcal{A} \to \emptyset \to \mathcal{BT}$ and $\mathsf{node}_\mathsf{N} \subseteq \mathcal{B} \to (\mathcal{BT} \,\&\, \mathcal{BT}) \to \mathcal{BT}$: up to standard isomorphisms, we consider the binary tree constructors $\mathsf{leaf} = \lambda x^{\mathcal{A}}\,(\mathsf{node}_\mathsf{F} x\,\langle\rangle) \subseteq \mathcal{A} \to \mathcal{BT}$ and $\mathsf{node} = \lambda y^{\mathcal{B}}\,\lambda t^{\mathcal{BT}}\,\lambda u^{\mathcal{BT}}\,(\mathsf{node}_\mathsf{N} y\,\langle t,u\rangle) \subseteq \mathcal{B} \to \mathcal{BT} \to \mathcal{BT} \to \mathcal{BT}$. Then the tree $t$ of Example 6.6 is obtained as*

$$t = \mathsf{node}\,b\,(\mathsf{leaf}\,a)\,(\mathsf{node}\,b'\,(\mathsf{leaf}\,a')\,(\mathsf{leaf}\,a''))$$

*and we can check that the interpretation given there agrees with this identity.*

Similarly, the *pattern matching operator* is given by:

$$\begin{aligned}\mathsf{match} \quad = \quad & \left\{\left([(\overline{\alpha},[(j_1,\tau_1),\ldots,(j_n,\tau_n)]\,,\beta)]\,,[i]+i\overline{\alpha}+\sum_{k=1}^{n}[ij_k\tau_k]\,,\beta\right);\right.\\ & \qquad\qquad \left. i \in I \wedge \beta \in B \wedge \overline{\alpha} \in\, !A_i \wedge \forall k,\ j_k \in J_i \wedge \tau_k \in L\overrightarrow{A}\right\}\\ \subseteq \quad & \mathop{\&}_{i\in I}\left(A_i \Rightarrow \left(L\overrightarrow{A}\right)^{\&J_i} \Rightarrow B\right) \Rightarrow L\overrightarrow{A} \Rightarrow B\end{aligned}$$

which is an instance of $\mathsf{case} \subseteq \&_{i\in I}\left(\left(A_i \,\&\, \left(L\overrightarrow{A}\right)^{\&J_i}\right) \Rightarrow B\right) \Rightarrow L\overrightarrow{A} \Rightarrow B$ up to our notations of addresses and the cartesian adjunction in $\underline{\mathrm{Rel}}^!$. As such, it is finitary: for all finiteness spaces $\overrightarrow{\mathcal{A}}$ and $\mathcal{B}$, $\mathsf{match} \in \mathfrak{F}\left(\mathcal{L}\overrightarrow{\mathcal{A}} \Rightarrow \&_{i\in I}\left(\mathcal{A}_i \Rightarrow \left(\mathcal{L}\overrightarrow{\mathcal{A}}\right)^{\&J_i} \Rightarrow \mathcal{B}\right) \Rightarrow \mathcal{B}\right)$.

As an application of Lemma 6.2, we moreover obtain that pattern matching is correct:

$$\overrightarrow{f} : \mathop{\&}_{i\in I}\left(A_i \Rightarrow \left(L\overrightarrow{A}\right)^{\&J_i} \Rightarrow B\right), a : A_i,\ \overrightarrow{t} : \left(L\overrightarrow{A}\right)^{\&J_i} \vdash \mathsf{match}\,\overrightarrow{f}\,\left(\mathsf{node}_i a\,\overrightarrow{t}\right) = f_i a\,\overrightarrow{t} : B$$

for all $i \in I$. Similarly to that of sums, this encoding of trees is lazy in the sense that, for all $\overrightarrow{b} \in \mathfrak{P}(B)^I$ and $i \in I$, $\mathsf{match}\langle\lambda x\,\lambda y\,b_i\rangle_{i\in I}\left(\mathsf{node}_i\emptyset\,\langle\emptyset\rangle_{i\in I}\right) = b_i$.

We can then construct the *iterator on trees*:

$$\begin{array}{rcl}\mathsf{iter} & = & \mathsf{fix}\ \lambda F\,\lambda t\,\lambda \overrightarrow{f}\left(\mathsf{match}\left\langle \lambda a\,\lambda \overrightarrow{t}\left(f_i\,a\left\langle F\,t_j\,\overrightarrow{f}\right\rangle_{j\in J_i}\right)\right\rangle_{i\in I} t\right)\\ & \subseteq & L\overrightarrow{A}\Rightarrow \&_{i\in I}\left(A_i\Rightarrow B^{\&J_i}\Rightarrow B\right)\Rightarrow B\end{array}$$

which automatically satisfies

$$\overrightarrow{f}:\mathop{\&}_{i\in I}\left(A_i\Rightarrow B^{\&J_i}\Rightarrow B\right), a:A_i, \overrightarrow{t}:\left(L\overrightarrow{A}\right)^{\&J_i}\vdash \mathsf{iter}\left(\mathsf{node}_i a\,\overrightarrow{t}\right)\overrightarrow{f}=f_i a\left\langle \mathsf{iter}\,t_j\,\overrightarrow{f}\right\rangle_{j\in J_i}.$$

The following lemma makes the structure of iter explicit:

**Lemma 6.11** *Let* $\mathsf{iter}_0=\emptyset$ *and, for all* $n\in\mathbf{N}$, *let*

$$\begin{array}{rcl}\mathsf{iter}_{n+1} & = & \left\{\left([i]+i\overline{\alpha}+\sum_{k=1}^p ij_k\overline{\tau}_k, [(i,\overline{\alpha},\sum_{k=1}^p[(j_k,\beta_k)],\beta)]+\sum_{k=1}^p\overline{\phi}_k,\beta\right);\right.\\ & & \left. i\in I\wedge p\in\mathbf{N}\wedge\forall k,\ j_k\in J_k\wedge(\overline{\phi}_k,\overline{\tau}_k,\beta_k)\in\mathsf{iter}_n\right\}.\end{array}$$

*Then* $(\mathsf{iter}_n)_{n\in\mathbf{N}}$ *is increasing for inclusion and* $\mathsf{iter}=\bigcup_{n\in\mathbf{N}}\mathsf{iter}_n$. *Moreover, if* $\iota=(\overline{\phi},\overline{\tau},\beta)\in\mathsf{iter}$, *then* $\iota\in\mathsf{iter}_{\max(\mathsf{len}\cdot supp(\overline{\tau}))+1}$.

**Proof** The equation $\mathsf{iter}=\bigcup_{n\in\mathbf{N}}\mathsf{iter}_n$ is just an unfolding of the definitions: if we write $f=\lambda F\,\lambda t\,\lambda\overrightarrow{f}\left(\mathsf{match}\left\langle\lambda a\,\lambda\overrightarrow{t}\left(f_i\,a\left\langle F\,t_j\,\overrightarrow{f}\right\rangle_{j\in J_i}\right)\right\rangle_{i\in I}t\right)$ then $\mathsf{iter}=\mathsf{fix}f=\bigcup_{n\in\mathbf{N}}f^n\emptyset$ and we just have to check that $f^n\emptyset=\mathsf{iter}_n$ by induction on $n$. The additional result is straightforwardly deduced from this explicitation. □

We now relate precisely the indices and values in the input paths of iter with those used in the associated instance of iterated functions. First, if $\tau=i_1j_1\cdots i_nj_n\nu\in L\overrightarrow{A}$, we write $ind(\tau)=\{i_1,j_1,\ldots,i_n,j_n,type(\nu)\}$ and $val(\tau)=\bigcup_{i\in I}\mathsf{val}_i\cdot\tau$ which are both finite. Moreover, if $\phi=(i,\overline{\alpha},\sum_{k=1}^n[(j_k,\beta_k)],\beta)\in\&_{i\in I}\left(A_i\Rightarrow B^{\&J_i}\Rightarrow B\right)$, we set $ind(\phi)=\{i\}\cup\{j_k;\ 1\le k\le n\}$ and $val(\phi)=supp(\overline{\alpha})$. We extend these to multisets by taking the union of images as in $ind(\overline{\tau})=\bigcup_{\tau\in supp(\overline{\tau})}ind(\tau)$. Recall that if $\overline{\alpha}\in!A$, $\#\overline{\alpha}$ denotes the multiset cardinality of $\overline{\alpha}$. When $\overline{\phi}=\sum_{k=1}^n[(i_k,\overline{\alpha}_k,\overline{\tau}_k,\beta_k)]\in!\&_{i\in I}\left(A_i\Rightarrow B^{\&J_i}\Rightarrow B\right)$, we write $\#\#(\overline{\phi})=\sum_{k=1}^n\#\overline{\alpha}_k$.

**Lemma 6.12** *For all* $\iota=(\overline{\tau},\overline{\phi},\beta)\in\mathsf{iter}$, *we have:*

- $ind(\overline{\tau})=ind(\overline{\phi})$;
- $val(\overline{\tau})=val(\overline{\phi})$;
- $\#\overline{\tau}=\#\overline{\phi}+\#\#\overline{\phi}$.

**Proof** The result is easily established for all $\iota\in\mathsf{iter}_n$, by induction on $n$. □

**Lemma 6.13** *Iteration is finitary:* $\mathsf{iter}\in\mathfrak{F}\left(\mathcal{L}\overrightarrow{\mathcal{A}}\Rightarrow\&_{i\in I}\left(\mathcal{A}_i\Rightarrow\mathcal{B}^{\&J_i}\Rightarrow\mathcal{B}\right)\Rightarrow\mathcal{B}\right)$.

**Proof** If $t\in\mathfrak{F}\left(\mathcal{L}\overrightarrow{\mathcal{A}}\right)$, then $\mathsf{len}\cdot t$ is finite: we write $n=\max(\mathsf{len}\cdot t)$. By Lemma 6.11, $\mathsf{iter}\cdot t^!=\mathsf{iter}_{n+1}\cdot t^!\in\mathfrak{F}\left(\&_{i\in I}\left(\mathcal{A}_i\Rightarrow\mathcal{B}^{\&J_i}\Rightarrow\mathcal{B}\right)\Rightarrow\mathcal{B}\right)$ because $\mathsf{iter}_{n+1}$ is finitary. Now fix $(\overline{\phi},\beta)\in\left|\&_{i\in I}\left(\mathcal{A}_i\Rightarrow\mathcal{B}^{\&J_i}\Rightarrow\mathcal{B}\right)\Rightarrow\mathcal{B}\right|$ and $\overline{t}'={}^t\mathsf{iter}\cdot(\overline{\phi},\beta)$: we prove that $\overline{t}'\perp_{\mathrm{f}}t^!$. By the previous lemma, for all $\overline{\tau}\in\overline{t}'$, $val(\overline{\tau})=val(\overline{\phi})$, $ind(\overline{\tau})=ind(\overline{\phi})$, $\#\overline{\tau}=\#\overline{\phi}+\#\#\overline{\phi}$. Paths in $supp\left(\overline{t}'\right)\cap t$ have addresses of length at most $n$ with indices taken in a fixed finite set; moreover they hold values taken in a fixed finite set. We deduce $supp(t')\cap t$ is finite. Moreover, multisets in $\overline{t}'\cap t^!$ are of fixed size: hence $\overline{t}'\cap t^!$ is finite. □

Summing up the results in section 6.2 and the current section, we obtain:

**Theorem 6.14** *For all choice of sets of indices $I$ and $\overrightarrow{J}$, $\mathcal{L}\overrightarrow{\mathcal{A}}$ is the finiteness space of paths whose finiteness structure is transported by $\overrightarrow{\mathsf{val}}$ and* $\mathsf{len}$. *Moreover, there are multirelations* $\mathsf{node}_i$ *and* $\mathsf{iter}$ *such that:*

- $\mathsf{node}_i \in \mathfrak{F}\left(\mathcal{A}_i \Rightarrow \left(\mathcal{L}\overrightarrow{\mathcal{A}}\right)^{\&J_i} \Rightarrow \mathcal{L}\overrightarrow{\mathcal{A}}\right)$;
- $\mathsf{iter} \in \mathfrak{F}\left(\mathcal{L}\overrightarrow{\mathcal{A}} \Rightarrow \&_{i\in I}\left(\mathcal{A}_i \Rightarrow \mathcal{B}^{\&J_i} \Rightarrow \mathcal{B}\right) \Rightarrow \mathcal{B}\right)$;
- $\lambda a\, \lambda \overrightarrow{t}\, \lambda \overrightarrow{f}\, \left(\mathsf{iter}\left(\mathsf{node}_i\, a\, \overrightarrow{t}\right)\, \overrightarrow{f}\right) = \lambda a\, \lambda \overrightarrow{t}\, \lambda \overrightarrow{f}\, \left(f_i\, a \left\langle \mathsf{iter}\, t_j\, \overrightarrow{f}\right\rangle_{j\in J_i}\right)$.

*Hence $\mathcal{L}\overrightarrow{\mathcal{A}}$ is the datatype of trees whose nodes of sort $i \in I$ are labelled with values in $\mathcal{A}_i$ and of arity $J_i$.*

As an example of application of this theorem, consider the functor $\mathcal{T} : \mathcal{X} \mapsto \top \overset{\bullet}{\oplus} \mathcal{X}$, that is obtained for $I = \{\mathsf{O}, \mathsf{S}\}$, $J_{\mathsf{O}} = \emptyset$, $J_{\mathsf{S}}$ any singleton set, and $\mathcal{A}_{\mathsf{O}} = \mathcal{A}_{\mathsf{S}} = \top$. Then $\mu\mathcal{T} \cong \mathcal{N}_l$ where $|\mathcal{N}_l| = \mathbf{N} \cup \mathbf{N}^{>}$, $\mathfrak{F}(\mathcal{N}_l) = \mathfrak{P}_{\mathrm{f}}(|\mathcal{N}_l|)$ and $\mathbf{N}^{>} = \{n^{>};\ n \in \mathbf{N}\}$ is just a disjoint copy of $\mathbf{N}$: $n \in \mathbf{N}$ (resp. $n^{>} \in \mathbf{N}^{>}$) corresponds with the only path $\tau = \pi\nu$ such that $\mathsf{len} \cdot \tau = \{n\}$ and $\mathit{type}(\nu) = \mathsf{O}$ (resp. $\mathit{type}(\nu) = \mathsf{S}$). The finiteness space $\mathcal{N}_l$ is intuitively that of *lazy natural numbers*: $n$ stands for "exactly $n$" whereas $n^{>}$ stands for "strictly more that $n$". From $\mathsf{inj}_0$ and $\mathsf{inj}_1$, we derive $\mathsf{zero} = \{0\} \in \mathfrak{F}(\mathcal{N}_l)$ and $\mathsf{succ} = \{([\,], 0^{>})\} \cup \{([\nu], \nu^{+});\ \nu \in |\mathcal{N}_l|\} \in \mathfrak{F}(\mathcal{N}_l \Rightarrow \mathcal{N}_l)$ where $n^{+} = n + 1$ and $n^{>+} = (n+1)^{>}$. Up to some standard isomorphisms, we derive a variant $\mathsf{natiter}$ of $\mathsf{iter}$ such that:

- $\mathsf{natiter} \in \mathfrak{F}(\mathcal{N}_l \Rightarrow (\mathcal{A} \Rightarrow \mathcal{A}) \Rightarrow \mathcal{A} \Rightarrow \mathcal{A})$;
- $\mathsf{natiter}\,\mathsf{zero} = \lambda f\, \lambda x\, x$;
- $\lambda n\, (\mathsf{natiter}\, (\mathsf{succ}\, n)) = \lambda n\, \lambda f\, \lambda x\, (f\, (\mathsf{natiter}\, n f x))$.

This provides a finitary relational semantics of Gödel's system $T$, which shows that $\underline{\mathrm{Fin}}^{!}$ can accomodate the standard notion of computational iteration. This was the subject of a previous article by the second author (Vaux; 2009c) which moreover shows that the same can be done for the recursor variant of system $T$.

The same applies here, actually: we could very well reproduce the results of this section, replacing $\mathsf{iter}$ with

$$\mathsf{rec} = \mathsf{fix}\left(\lambda F\, \lambda t\, \lambda \overrightarrow{f}\, \left(\mathsf{match}\left\langle \lambda a\, \lambda \overrightarrow{t}\, \left(f_i\, a\, \overrightarrow{t}\, \left\langle F\, t_j\, \overrightarrow{f}\right\rangle_{j\in J_i}\right)_{i\in I}\, t\right\rangle\right)\right)$$

which automatically satisfies

$$\lambda a\, \lambda \overrightarrow{t}\, \lambda \overrightarrow{f}\, \left(\mathsf{rec}\left(\mathsf{node}_i\, a\, \overrightarrow{t}\right)\, \overrightarrow{f}\right) = \lambda a\, \lambda \overrightarrow{t}\, \lambda \overrightarrow{f}\, \left(f_i\, a\, \overrightarrow{t}\, \left\langle \mathsf{rec}\, t_j\, \overrightarrow{f}\right\rangle_{j\in J_i}\right).$$

We would then verify that $\mathsf{rec} \in \mathfrak{F}\left(\mathcal{L}\overrightarrow{\mathcal{A}} \Rightarrow \&_{i\in I}\left(\mathcal{A}_i \Rightarrow \mathcal{L}\overrightarrow{\mathcal{A}}^{\&J_i} \Rightarrow \mathcal{B}^{\&J_i} \Rightarrow \mathcal{B}\right) \Rightarrow \mathcal{B}\right)$ for all finiteness spaces $\overrightarrow{\mathcal{A}}$ and $\mathcal{B}$.

# References


Amadio, R. and Curien, P.-L. (1998). *Domains and lambda-calculi, volume 46 of Cambridge Tracts in Theoretical Computer Science*, Vol. 2.

Backhouse, R. C., de Bruin, P. J., Hoogendijk, P. F., Malcolm, G., Voermans, E. and van der Woude, J. (1991). Polynomial relators, *in* M. Nivat, C. Rattray, T. Rus and G. Scollo (eds), *AMAST*, Workshops in Computing, Springer, pp. 303–326.

Backhouse, R. C. and Hoogendijk, P. F. (2003). Generic properties of datatypes, *in* R. C. Backhouse and J. Gibbons (eds), *Generic Programming*, Vol. 2793 of *Lecture Notes in Computer Science*, Springer, pp. 97–132.

Baelde, D. and Miller, D. (2007). Least and greatest fixed points in linear logic, *in* N. Dershowitz and A. Voronkov (eds), *LPAR*, Vol. 4790 of *Lecture Notes in Computer Science*, Springer, pp. 92–106.

Bierman, G. M. (1995). What is a categorical model of intuitionistic linear logic?, *in* M. Dezani (ed.), *Proceedings of Conference on Typed lambda calculus and Applications*, Springer-Verlag LNCS 902.

Bird, R. and de Moor, O. (1997). *Algebra of programming*, Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Bucciarelli, A., Ehrhard, T. and Manzonetto, G. (2007). Not enough points is enough, *Computer Science Logic*, Vol. 4646 of *Lecture Notes in Computer Science*, Springer Berlin, pp. 298–312.

Carboni, A., Kasangian, S. and Street, R. (1984). Bicategories of spans and relations, *Journal of Pure and Applied Algebra* **33**(3): 259 – 267.

Carboni, A., Kelly, G. M. and Wood, R. J. (1991). A 2-categorical approach to change of base and geometric morphisms. I, *Cahiers Topologie Géom. Différentielle Catég.* **32**(1): 47–95. International Category Theory Meeting (Bangor, 1989 and Cambridge, 1990).

Clairambault, P. (2010). *Logique et Interaction : une Étude Sémantique de la Totalité*, Thèse d'université, Université Paris 7.

Curien, P.-L. (ed.) (2009). *Typed Lambda Calculi and Applications, 9th International Conference, TLCA 2009, Brasilia, Brazil, July 1-3, 2009. Proceedings*, Vol. 5608 of *Lecture Notes in Computer Science*, Springer.

de Carvalho, D. (2008). Execution time of lambda-terms via denotational semantics and intersection types, *Technical report*. Rapport de recherche INRIA n° 6638.

Ehrhard, T. (1993). Hypercoherences: A strongly stable model of linear logic, *Mathematical Structures in Computer Science* **3**(4): 365–385.

Ehrhard, T. (2005). Finiteness spaces, *Mathematical Structures in Computer Science* **15**(4): 615–646.

Ehrhard, T. and Laurent, O. (2007). Interpreting a finitary pi-calculus in differential interaction nets, *in* L. Caires and V. T. Vasconcelos (eds), *Concurrency Theory (CONCUR '07)*, Vol. 4703 of *Lecture Notes in Computer Science*, Springer, pp. 333–348.

Ehrhard, T. and Regnier, L. (2003). The differential lambda-calculus, *Theoretical Computer Science* **309**: 1–41.

Ehrhard, T. and Regnier, L. (2005). Differential interaction nets., *Electr. Notes Theor. Comput. Sci.* **123**: 35–74.

Ehrhard, T. and Regnier, L. (2006). Böhm trees, Krivine's machine and the Taylor expansion of $\lambda$-terms, *in* A. Beckmann, U. Berger, B. Löwe and J. V. Tucker (eds), *CiE*, Vol. 3988 of *Lecture Notes in Computer Science*, Springer, pp. 186–197.

Fernández, M., Mackie, I., Sato, S. and Walker, M. (2009). Recursive functions with pattern matching in interaction nets, *Electr. Notes Theor. Comput. Sci.* **253**(4): 55–71.

Gimenez, S. (2009). *Programmer, Calculer et Raisonner avec les Réseaux de la Logique Linéaire*, Thèse d'université, Université Paris 7.

Girard, J.-Y. (1988). Normal functors, power series and lambda-calculus, *Annals of Pure and Applied Logic* **37**(2): 129–177.

Girard, J.-Y., Taylor, P. and Lafont, Y. (1989). *Proofs and types*, CUP, Cambridge.

Hoogendijk, P. and De Moor, O. (2000). Container types categorically, *J. Funct. Program.* **10**: 191–225.

Hyland, M. and Schalk, A. (2003). Glueing and orthogonality for models of linear logic, *Theor. Comput. Sci.* **294**(1/2): 183–231.

Lambek, J. and Scott, P. (1986). *Introduction to Higher Order Categorical Logic*, number 7 in *Cambridge Studies in Advanced Mathematics*, Cambridge University Press.

Lambek, J. and Scott, P. J. (1988). *Introduction to higher order categorical logic*, Cambridge University Press, New York, NY, USA.

Loader, R. (1994). Linear logic, totality and full completeness, *LICS*, IEEE Computer Society, pp. 292–298.

Mac Lane, S. (1998). *Categories for the Working Mathematician*, Springer.

Pagani, M. and Tasson, C. (2009). The inverse Taylor expansion problem in Linear Logic, *in* A. M. Pitts (ed.), *Proceedings of the 24th Annual IEEE Symposium on Logic in Computer Science, LICS 2009*, IEEE Computer Society, pp. 222–231.

Smyth, M. B. and Plotkin, G. D. (1982). The category-theoretic solution of recursive domain equations, *SIAM J. Comput.* **11**(4): 761–783.

Tasson, C. (2009). Algebraic totality, towards completeness, *in* Curien (2009), pp. 325–340.

Thibault, M.-F. (1982). Pre-recursive categories, *Journal of Pure and Applied Algebra* **24**: 79–93.

Tranquilli, P. (2008). Intuitionistic differential nets and lambda-calculus. To appear in Theoretical Computer Science.

Vaux, L. (2009a). The algebraic lambda calculus, *Mathematical Structures in Computer Science* **19**(5): 1029–1059.

Vaux, L. (2009b). Differential linear logic and polarization, *in* Curien (2009), pp. 371–385.

Vaux, L. (2009c). A non-uniform finitary relational semantics of system T, *in* R. Matthes and T. Uustalu (eds), *Proceedings of the 6th Workshop on Fixed Points in Computer Science*, Institute of Cybernetics at Tallinn University of Technology.