Under consideration for publication in Math. Struct. in Comp. Science

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.*

Received April 13, 2010

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 (Ehr05), 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 the finitary relational model of linear logic, the usual co-Kleisli construction provides a cartesian closed category, hence a model of the simply typed λ-calculus. The defining property of finiteness spaces is that the intersection of two finitary subsets of dual types is always finite. This feature allows to reformulate Girard's quantitative semantics (Gir88) in a standard algebraic setting, where morphisms interpreting typed λ-terms are analytic functions between the topological vector spaces generated by vectors with finitary supports. This provided the semantic foundations of Ehrhard-Regnier's differential λ-calculus (ER03) and motivated the general study of a differential extension of linear logic (ER05; ER06; EL07; Tra08; Vau09b; Tas09; PT09, *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 mentionned, the relational semantics of a proof is always finitary. The present paper studies the connexion between the category $\underline{\mathrm{Rel}}$ of sets and relations and the category $\underline{\mathrm{Fin}}$ of finiteness spaces and finitary

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

relations, while maintaining a similar standpoint: we investigate whether and how some of the most distinctive features of $\underline{\mathrm{Rel}}$ can be given counterparts in $\underline{\mathrm{Fin}}$.

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 $\underline{\mathrm{Rel}}$ to give rise to a functor in $\underline{\mathrm{Fin}}$ *via* the transport lemma: again, this generalizes the functoriality of the positive connectives of linear logic.

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 (CKS84; CKW91) and the properties of monotonic functors allowed for an abstract description of datatypes (BH03; BdBH+91). In particular, this provided the basis of a categorical account of container types (HDM00). 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. This prompted us to consider two orders on finiteness spaces derived from set inclusion: the most restrictive one, *finiteness extension*, was used by Ehrhard to provide an interpretation of second order linear logic (Ehr05, unpublished preliminary version), 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 mentionned co-Keisli $\underline{\mathrm{Fin}}^!$ of $\underline{\mathrm{Fin}}$ provides a model of the λ-calculus can again be understood as a property of the interpretation in $\underline{\mathrm{Rel}}^!$: the relational semantics of a *simply typed* λ-term is always finitary. It is however worth noticing that, whereas the relational model can accomodate untyped λ-calculi (dC08; BEM07), 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 on objects, the category $\underline{\mathrm{Fin}}^!$ (and thus $\underline{\mathrm{Fin}}$) is not enriched on complete partial orders.

Despite this restrictive design, Ehrhard was able to define a finitary interpretation of tail-recursive iteration (Ehr05, Section 3): 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 (Thi82; LS88) in the associated model of the λ-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 (GTL89): 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 (Vau09c) 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.

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 they provide a relational semantics of the typed λ-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 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 λ-calculus in vectorial finiteness spaces to functional programming with base types. 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 first author's extension of the algebraic λ-calculus (Vau09a) with a type of booleans, together with a semantic characterization of total terms which is proved to be complete on boolean functions (Tas09). In previous unpublished work, we also proposed a quantitative semantics of tail recursive iteration. As we mentionned 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, 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 (Gim09), which allows to interpret, e.g., PCF. As such this system can be seen as a graphical syntax for general recursion. Along similar lines, Mackie *et al.* have proposed a system of interaction nets which models iteration on recursive datatypes (FMSW09). In both cases, no particular denotational semantics is considered. Let us also mention Baelde–Miller's μMALL (BM07) 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 μLJ of intuitionistic logic with fixpoints, for which Clairambault later proposed a cut elimination procedure allowing to encode system T, together with a game semantics accounting for typed recursion (Cla10).

The notion of transport functor we use to describe how functors in Fin can be derived from functors in Rel is strikingly similar to the categorical characterization of container types by Hoogendijk and De Moor we already mentionned (HDM00). These authors characterize containers as relators with membership: relators are functors in Rel which are monotonic for inclusion; membership relations are particular lax natural transformations associated with these functors. The hypotheses we consider on tranport functors are weaker than those on relators with membership. On the other hand, in order to ensure functoriality in Fin, we are led to refer to a *shape* relation: it might be the case that restricting transport functors to relators with finite membership will allow us to drop this somewhat inelegant side condition.

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 λ-calculi: we first recall the semantics of the simply typed λ-calculus in section 5, and 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 μ is such a multiset, we write $\mathit{supp}(\mu)$ 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 $(\alpha_1, \dots, \alpha_n) \in A^n$, we write $[\alpha_1, \dots, \alpha_n]$ for the corresponding finite multiset: $\alpha \in A \mapsto \#\{\alpha_i;\ \alpha_i = \alpha\}$. We also write $\#[\alpha_1, \dots, \alpha_n] = n$ 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. $\overline{\alpha} = [\alpha_1, \dots, \alpha_n] \in \mathfrak{M}_{\mathrm{f}}(A)$).

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 obvious 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 reverse 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 α. 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 $(\beta, \gamma) \in g$. The identity relation on A is just the diagonal: $\mathsf{id}^A = \{(\alpha, \alpha);\ \alpha \in A\} \subseteq A \times A$. Equipped with this notion of composition, relations form a category $\underline{\mathrm{Rel}}$ whose objects are sets: $f \in \underline{\mathrm{Rel}}(A, B)$ iff $f \subseteq A \times B$. Notice that our presentation goes against standard definitions of categories such as Mac Lane's (Lan98), where homsets are supposed to be pairwise disjoint or, equivalently, every morphism has exactly one source and one target. This has non-trivial consequences since it makes the following definition of functors stronger than usual.[†]

Definition 2.1. A *functor* T in $\underline{\mathrm{Rel}}$ is the data of a set TA for all set A and a relation Tf for all relation f, such that $Tf \subseteq TA \times TB$ as soon as $f \subseteq A \times B$, preserving identities and composition: $T\mathsf{id}_A = \mathsf{id}_{TA}$ for all set A, and $T(f \circ g) = Tf \circ Tg$ for all relations $f \subseteq A \times B$ and $g \subseteq B \times C$.

Notice that a functor in that definition is a functor in the usual sense, subject to the

[†] As an exercise, the playful reader may try and exhibit a functor in the usual sense which does not fit our definition: counter-counter-examples are easy to come with...

additional property that the image of a morphism does not depend on its type. The reason why we depart from standard terminology is that this condition is verified by all the constructions we use, and it has many useful implications that we would otherwise need to add as separate hypotheses in our results. For instance, functors are always monotonic for set inclusion: if $A \subseteq B$, then $\mathsf{id}_A \subseteq B \times B$ and $T\mathsf{id}_A = \mathsf{id}_{TA} \subseteq TB \times TB$, hence $TA \subseteq TB$.

A functor T is said to be *symmetric* if $T\,{}^t f = {}^t T f$ for all f. A functor which is monotonic for relation inclusion is called a *relator*. Relators were the object of several works on the theory of datatypes (BdBH+91; HDM00; BH03). Notice that they are automatically symmetric. A relator is said to be *continuous* if it preserves directed unions of sets and relations: $T \bigcup \overrightarrow{A} = \bigcup \overrightarrow{TA}$ (resp. $T \bigcup \overrightarrow{f} = \bigcup \overrightarrow{Tf}$) for all family of sets $\overrightarrow{A}$ (resp. of relations $\overrightarrow{f}$) which is directed for inclusion. Of course, the identity functor is a continuous relator, as is the (contravariant) reverse functor, which is the identity on sets and sends every relation f to its reverse ${}^t f$. Another standard example is the multiset functor given by $!A = \mathfrak{M}_{\mathrm{f}}(A)$ and, for all relation f, $!f = \{([\alpha_1, \dots, \alpha_n], [\beta_1, \dots, \beta_n]);\ n \in \mathbf{N} \wedge \forall k,\ (\alpha_k, \beta_k) \in f\}$. When $a \subseteq A$, we write $a^! = \mathfrak{M}_{\mathrm{f}}(a) \in\, !A$ (rather that $!a \in\, !A$) 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* when 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 id on sets of the form TA constitute a natural transformation from T to itself. For all set A, consider the only relation supp from $!A$ to A such that $\mathsf{supp} \cdot \overline{\alpha} = \mathit{supp}\,(\overline{\alpha})$ for all $\overline{\alpha} \in\, !A$. This defines a 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. Sometimes, it is useful to relate f^A with f^B where $A \subseteq B$.

Lemma 2.1. Let f be a lax natural transformation from T to U. Then, if $A \subseteq B$:
— for all $\widetilde{a} \subseteq TA$, $f^A \cdot \widetilde{a} = f^B \cdot \widetilde{a}$;
— for all $\widehat{b} \subseteq UB$, $f^A \setminus \widehat{b} = TA \cap f^B \setminus \widehat{b}$.

Proof. By applying the naturality condition to the identity 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 properties follow. □

We now generalize these notions to families of relations: 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: for all $i \in I$, $f_i \subseteq A_i \times B_i$. We denote by $\underline{\mathrm{Rel}}^I$ the category of families of sets and families of relations, with componentwise identities and composition. Again, morphisms are not typed *a priori*, and componentwise identities are preserved so that all functors are increasing for set inclusion: if $\overrightarrow{A \subset B}$ then $T\overrightarrow{A} \subset T\overrightarrow{B}$. We denote by Π_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 $\overrightarrow{B}$,

$\Pi_i \overrightarrow{f} = f_i$. We say a functor T from $\underline{\mathrm{Rel}}^I$ to $\underline{\mathrm{Rel}}$ is *symmetric* if $T^t\overrightarrow{f} = {}^t\left(T\overrightarrow{f}\right)$ for all $\overrightarrow{f}$; we say it is a *relator* if $\overrightarrow{f \subseteq g}$ implies $T\overrightarrow{f} \subseteq T\overrightarrow{g}$.

In order to define the continuity of I-ary relators, we introduce the following useful conventions. By $\overrightarrow{\overrightarrow{A}}$, we denote an I-indexed family $\left(\overrightarrow{A}_i\right)_{i\in I}$ of families of sets, where each $\overrightarrow{A}_i = (A_{i,j})_{j\in J_i}$ 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 say $\overrightarrow{\overrightarrow{A}}$ is componentwise directed if each $\overrightarrow{A}_i$ is directed for inclusion. We write $\overrightarrow{\overrightarrow{\bigcup A}}$ for $\left(\bigcup \overrightarrow{A}_i\right)_{i\in I}$ and call this family of unions the componentwise union of $\overrightarrow{\overrightarrow{A}}$. Finally, if T is a functor from $\underline{\mathrm{Rel}}^I$ to $\underline{\mathrm{Rel}}$, we write $\overrightarrow{\overrightarrow{TA}}$ for the $\prod \overrightarrow{J}$-indexed family $\left(T\overrightarrow{A}_{\overrightarrow{j}}\right)_{\overrightarrow{j\in J}}$. Then we say T is *continuous* if it commutes to componentwise directed unions: $T\overrightarrow{\overrightarrow{\bigcup A}} = \bigcup \overrightarrow{\overrightarrow{TA}}$ as soon as $\overrightarrow{\overrightarrow{A}}$ is componentwise directed. For instance, projection functors are continuous relators. Other standard examples include: the *cartesian product* functor, given by $\bigotimes \overrightarrow{A} = \prod \overrightarrow{A}$ and $\bigotimes \overrightarrow{f} = \left\{\left(\overrightarrow{\alpha}, \overrightarrow{\beta}\right);\ \forall i \in I,\ (\alpha_i, \beta_i) \in f_i\right)\right\}$; 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 id on sets of the form $T\overrightarrow{A}$ define a natural transformation from T to itself. For all $i \in I$, we denote by proj_i the i-th projection relation from $\bigotimes \overrightarrow{A}$ to A_i: $\mathsf{proj}_i = \left\{(\overrightarrow{\alpha}, \alpha_i);\ \overrightarrow{\alpha} \in \bigotimes \overrightarrow{A}\right\}$. Also, we denote by rest_i the restriction relation from $\bigoplus \overrightarrow{A}$ to A_i: $\mathsf{rest}_i = \{((i,\alpha),\alpha);\ \alpha \in A_i\}$, and by indx the index relation from $\bigoplus \overrightarrow{A}$ to I: $\{((i,\alpha),i);\ i \in I \wedge \alpha \in A_i\}$. Each proj_i is a natural transformation from $\bigotimes$ to Π_i, and each rest_i is a natural transformation from $\bigoplus$ to Π_i; moreover indx is a natural transformation from $\bigoplus$ to E_I, which is the constant functor $E_I\overrightarrow{A} = I$ and $E_I\overrightarrow{f} = \mathsf{id}^I$.

Finally, notice that, by the same argument as in Lemma 2.1, if f is a natural transformation from T to U, and $\overrightarrow{A \subseteq B}$ then:

— for all $\widetilde{a} \subseteq T\overrightarrow{A}$, $f^{\overrightarrow{A}} \cdot \widetilde{a} = f^{\overrightarrow{B}} \cdot \widetilde{a}$;

— for all $\widehat{b} \subseteq U\overrightarrow{B}$, $f^{\overrightarrow{A}} \setminus \widehat{b} = T\overrightarrow{A} \cap f^{\overrightarrow{B}} \setminus \widehat{b}$.

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 a standard argument, we have the following properties: $\mathfrak{P}_{\mathrm{f}}(A) \subseteq \mathfrak{A}^{\perp_A}$; $\mathfrak{A} \subseteq \mathfrak{A}^{\perp\perp_A}$; if $\mathfrak{A}' \subseteq \mathfrak{A}$, then $\mathfrak{A}^{\perp_A} \subseteq \mathfrak{A}'^{\perp_A}$. By the last two, we get $\mathfrak{A}^{\perp_A} = \mathfrak{A}^{\perp\perp\perp_A}$. A *finiteness structure* on A is a set $\mathfrak{A}$ of subsets of A such that $\mathfrak{A}^{\perp\perp_A} = \mathfrak{A}$. Then a *finiteness space* is a dependant 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: $|\mathcal{A}^\perp| = |\mathcal{A}|$ and $\mathfrak{F}(\mathcal{A}^\perp) = \mathfrak{F}(\mathcal{A})^{\perp_{|\mathcal{A}|}}$. The elements of $\mathfrak{F}(\mathcal{A})$ are called the *finitary subsets* of $\mathcal{A}$.

For all 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 $\emptyset$, the singleton finiteness space $\mathbf{1}$ with web $\{\emptyset\}$ and the space of *flat natural numbers* $\mathcal{N} = (\mathbf{N}, \mathfrak{P}_{\mathrm{f}}(\mathbf{N}))$.

The following property is easily established.

Lemma 3.1. For all finiteness structure $\mathfrak{A}$ on A, $\mathfrak{P}_{\mathrm{f}}(A) \subseteq \mathfrak{A}$, and $\mathfrak{A}$ is downwards closed for inclusion and closed under finite unions.

It is however not sufficient to characterize finiteness structures, as shown by the following counter-example, communicated to us by Laurent Regnier.

Counter-example 3.1. We say $t \subseteq \mathbf{N}$ is *thin* if the sequence $\left(\frac{\# t\cap\{0,\dots,n-1\}}{n}\right)_{n\in\mathbf{N}}$ 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 $\{n^n;\ n \in \mathbf{N}\}$. Notice that $\mathfrak{P}_{\mathrm{f}}(\mathbf{N}) \subseteq \mathfrak{T}$, and that $\mathfrak{T}$ is downwards closed for inclusion and closed under finite unions. Moreover, 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, $\{\alpha_{n^2};\ n \in \mathbf{N}\} \in \mathfrak{T}$. We obtain that $\mathfrak{T}^{\perp\perp_{\mathbf{N}}} = \mathfrak{P}(\mathbf{N})$.

All along the text, we tried to 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, where $\mathfrak{A} \subseteq \mathfrak{P}(A)$ is downwards closed for inclusion, closed under finite unions, and contains $\mathfrak{P}_{\mathrm{f}}(A)$, but $\mathfrak{A} \subsetneq \mathfrak{A}^{\perp\perp_A}$. In that case, we say $\mathfrak{A}$ is a *fake finiteness structure* on A. Below we present another fake finiteness structure, the properties of which will be useful in some of our arguments.

Counter-example 3.2. 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 one can check that $\mathfrak{C}^{\perp_{\mathbf{N}\times\mathbf{N}}} = \mathfrak{P}_{\mathrm{f}}(\mathbf{N}\times\mathbf{N})$ 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 3.2 (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 $\alpha \in A$. Then $\mathfrak{F}_{\mathcal{B},f} = \{a \subseteq A;\ f \cdot a \in \mathfrak{F}(\mathcal{B})\}$ is a finiteness structure on A and $\mathfrak{F}_{\mathcal{B},f} = \{f \setminus b;\ b \in \mathfrak{F}(\mathcal{B})\}^{\perp\perp_A}$.

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 $b' \in \mathfrak{F}\left(\mathcal{B}^{\perp}\right)$, $b'' = (f \cdot a) \cap b'$ 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 $b'' \subseteq b' \in \mathfrak{F}\left(\mathcal{B}^{\perp}\right)$, 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 ϕ is a function and $b'' \cap b \subseteq b' \cap b$ is finite as $b' \in \mathfrak{F}\left(\mathcal{B}^{\perp}\right)$ and $b \in \mathfrak{F}(\mathcal{B})$. □

The following example shows how to construct the exponential of a finiteness space thanks to the transport lemma applied to the support relation.

Example 3.3. Let $\mathcal{A} = (A, \mathfrak{A})$ be a finiteness space, and recall that supp_A is the only relation from $!A$ to A such that $\mathsf{supp}_A \cdot \overline{\alpha} = \mathit{supp}(\overline{\alpha})$ for all $\overline{\alpha} \in\ !A$. Notice in particular that $\mathit{supp}(\overline{\alpha}) \in \mathfrak{P}_{\mathrm{f}}(A) \subseteq \mathfrak{A}$. 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 $\mathsf{supp}_A \setminus a = \mathfrak{M}_{\mathrm{f}}(a) = a^!$: we obtain $\mathfrak{F}(!\mathcal{A}) = \left\{a^!;\ a \in \mathfrak{F}(\mathcal{A})\right\}^{\perp\perp_A}$.

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 3.4. 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 3.2, each $\mathfrak{F}_{\mathcal{B}_i,f_i}$ is a finiteness structure on A. As bidual closure commutes to intersections, $\mathfrak{F}_{\overrightarrow{\mathcal{B}},\overrightarrow{f}} = \bigcap_{i \in I} \mathfrak{F}_{\mathcal{B}_i,f_i}$ is a finiteness structure and $\mathfrak{F}_{\overrightarrow{\mathcal{B}},\overrightarrow{f}} = \left(\bigcap_{i \in I}\{f_i \setminus b_i;\ b_i \in \mathfrak{F}(\mathcal{B}_i)\}\right)^{\perp\perp_A}$. 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 $\left\{\bigcap_{i\in I}(f_i \setminus b_i)\,;\ \forall i \in I,\ b_i \in \mathfrak{F}(\mathcal{B}_i)\right\} \subseteq \mathfrak{F}_{\overrightarrow{\mathcal{B}},\overrightarrow{f}}$, and we conclude since bidual closure is monotonic and idempotent. □

Example 3.5. 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 $\&\,\overrightarrow{\mathcal{A}}$ be the finiteness space $\left(\sum \overrightarrow{|\mathcal{A}|}, \mathfrak{F}_{\overrightarrow{\mathcal{A}},\overrightarrow{\mathsf{rest}}}\right)$: for all $\widetilde{a} \subseteq \sum \overrightarrow{|\mathcal{A}|}$, $\widetilde{a} \in \mathfrak{F}\left(\&\,\overrightarrow{\mathcal{A}}\right)$ iff $\mathsf{rest}_i \cdot \widetilde{a} \in \mathfrak{F}(\mathcal{A}_i)$ for all $i \in I$. Notice that this implies $\mathfrak{F}\left(\&\,\overrightarrow{\mathcal{A}}\right) = \left\{\sum \overrightarrow{a};\ \overrightarrow{a \in \mathfrak{F}(\mathcal{A})}\right\}$, hence the bidual closure is optional in that case.

If we write $\mathcal{I} = (I, \mathfrak{P}_{\mathrm{f}}(I))$, we can moreover define the finiteness space $\bigoplus \overrightarrow{\mathcal{A}}$ by $\left(\sum \overrightarrow{|\mathcal{A}|}, \mathfrak{F}_{(\overrightarrow{\mathcal{A}},\mathcal{I}),(\overrightarrow{\mathsf{rest}},\mathsf{indx})}\right)$: $\widetilde{a} \in \mathfrak{F}\left(\bigoplus \overrightarrow{\mathcal{A}}\right)$ 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 $\left(\bigoplus \overrightarrow{\mathcal{A}}\right)^{\perp} = \&\,\overrightarrow{\mathcal{A}^{\perp}}$, and moreover $\bigoplus \overrightarrow{\mathcal{A}} = \&\,\overrightarrow{\mathcal{A}}$ 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 $\bigparr \overrightarrow{\mathcal{A}} = \left(\bigotimes \overrightarrow{\mathcal{A}^{\perp}}\right)^{\perp}$. 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 easily established (Ehr05, Section 1.1).

Lemma 3.3. 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 $\beta \in |\mathcal{B}|$, ${}^t f \cdot \beta \in \mathfrak{F}\left(\mathcal{A}^{\perp}\right)$;
(d) $f \in \mathfrak{F}(\mathcal{A} \multimap \mathcal{B})$.

Notice that, whenever $A \subset |\mathcal{B}|$, the identity relation id_A is finitary from $\mathcal{B}$ to itself, and finitary relations compose: we thus introduce the category $\underline{\mathrm{Fin}}$ whose objects are finiteness spaces and morphisms are relations, so that $f \in \underline{\mathrm{Fin}}(\mathcal{A}, \mathcal{B})$ iff f is finitary from $\mathcal{A}$ to $\mathcal{B}$, i.e. $\underline{\mathrm{Fin}}(\mathcal{A}, \mathcal{B}) = \mathfrak{F}(\mathcal{A} \multimap \mathcal{B})$. Similarly to $\underline{\mathrm{Rel}}$, morphisms are *a priori* untyped and homsets need not be disjoint: notice in particular that a relation from A to B is always finitary from $(A, \mathfrak{P}_{\mathrm{f}}(A))$ to $(B, \mathfrak{P}(B))$. We moreover require functors in $\underline{\mathrm{Fin}}$ to be defined from functors in $\underline{\mathrm{Rel}}$. More precisely:

Definition 3.6. A functor $\mathcal{T}$ in $\underline{\mathrm{Fin}}$ is the data of a functor $T = |\mathcal{T}|$ in $\underline{\mathrm{Rel}}$ (the *web* of $\mathcal{T}$) and of a finiteness structure $\mathfrak{F}_{\mathcal{T}}(\mathcal{A})$ on $T\,|\mathcal{A}|$ for all finiteness space $\mathcal{A}$, so that

$\mathcal{T}\mathcal{A} = (T\,|\mathcal{A}|\,, \mathfrak{F}_{\mathcal{T}}\,(\mathcal{A}))$ is a finiteness space, moreover subject to the condition that $\mathcal{T}f = Tf \in \underline{\mathrm{Fin}}(\mathcal{T}\mathcal{A}, \mathcal{T}\mathcal{B})$ as soon as $f \in \underline{\mathrm{Fin}}(\mathcal{A}, \mathcal{B})$.

Again, this is a stronger requirement than usual: when $f \subset A \times B$, Tf must be finitary from $\mathcal{T}(A, \mathfrak{A})$ to $\mathcal{T}(B, \mathfrak{B})$ whatever finiteness structure $\mathfrak{A}$ and $\mathfrak{B}$ we impose on A and B. If $\overrightarrow{\mathcal{A}}$ is a family of finiteness spaces, we say relation $\overrightarrow{f}$ from $|\overrightarrow{\mathcal{A}}|$ to $|\overrightarrow{\mathcal{B}}|$ is finitary from $\overrightarrow{\mathcal{A}}$ to $\overrightarrow{\mathcal{B}}$ if f_i is finitary from $\mathcal{A}_i$ to $\mathcal{B}_i$ for all $i \in I$. The category $\underline{\mathrm{Fin}}^I$ of families of finiteness spaces and families of relations is then derived from $\underline{\mathrm{Fin}}$ in the same way as $\underline{\mathrm{Rel}}^I$ is derived from $\underline{\mathrm{Rel}}$: notice that $\underline{\mathrm{Fin}}^I\left(\overrightarrow{\mathcal{A}}, \overrightarrow{\mathcal{B}}\right) = \prod_{i\in I} \underline{\mathrm{Fin}}(\mathcal{A}_i, \mathcal{B}_i) \cong \mathfrak{F}\left(\& \overrightarrow{\mathcal{A} \multimap \mathcal{B}}\right)$. An I-ary functor in $\underline{\mathrm{Fin}}$ is a functor from $\underline{\mathrm{Fin}}^I$ to $\underline{\mathrm{Fin}}$, in accordance with the obvious generalisation of the above definition. We then import the vocabulary from $\underline{\mathrm{Rel}}$ to $\underline{\mathrm{Fin}}$: a functor $\mathcal{T}$ is said to be symmetric if $|\mathcal{T}|$ is; it is a *finitary relator* if $|\mathcal{T}|$ is a relator. We will discuss the continuity of finiteness relators in section 4. The following section explains how functors in $\underline{\mathrm{Fin}}$ can be derived from functors in $\underline{\mathrm{Rel}}$ *via* the transport lemma.

On a side note, remark that the construction of a finiteness spaces 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.

Counter-example 3.7. The relation supp_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 $\mathsf{supp}_A \setminus \{\alpha\} = \{\alpha\}^! \in \mathfrak{F}\,(!\mathcal{A})$ is infinite and cannot be in $\mathfrak{F}\,(!\mathcal{A})^{\perp_{|\mathcal{A}|}}$; we conclude by Lemma 3.3.

3.4. *Transport functors*

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 own_i from T to the projection functor Π_i, for all $i \in I$. Notice that any ownership relation on T satisfies the hypotheses of Corollary 3.4. Indeed, since own_i is quasi-functional, $\mathsf{own}_i \cdot \widetilde{\alpha}$ is finite for all $\widetilde{\alpha} \in T|\overrightarrow{\mathcal{A}}|$, hence it is finitary in $\mathcal{A}_i$ whatever $\mathfrak{F}\,(\mathcal{A}_i)$. 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 funtor 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 $T_{\overrightarrow{\mathsf{own}}}\mathcal{A}$ 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 this 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 shp from T to the constant functor E_S which sends every set to S and every relation to 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 shp is quasi-functionnal). 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

shp is a lax natural transformation). Notice that when T is symmetric, ${}^t\overrightarrow{T f} = \overrightarrow{T^t f}$, 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 components is itself finite.

Lemma 3.4. 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 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 own_i is a lax natural transformation from T to Π_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 ${}^t\widetilde{f} = T\overrightarrow{{}^t f}$. Then, since 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 shp is quasi-functional. Similarly, for all $i \in I$, own_i is a lax natural transformation from T to Π_i, hence $\mathsf{own}_i \circ {}^t\widetilde{f} \subseteq {}^t f_i \circ \mathsf{own}_i$: we obtain $\mathsf{own}_i \cdot \widetilde{a}' \subseteq {}^t f_i \cdot \mathsf{own}_i \cdot \widetilde{\beta}$. Since 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 ${}^t f_i \cdot \mathsf{own}_i \cdot \widetilde{\beta} \in \mathfrak{F}\left(\mathcal{A}_i^{\perp}\right)$, 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. $\square$

Notice that the use of the shape relation is crucial, since some transport situations with symmetric functor do not preserve finitary relations:

Counter-example 3.8. 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 relation $s = \{(\overrightarrow{\alpha}, \alpha);\ \exists n \in \mathbf{N},\ \alpha_n = \alpha\}$ is an ownership relation on S. Now consider the unique finiteness space 2 with web $\{0,1\}$. Then $S\,|2| = \{0,1\}^{\mathbf{N}}$ and $\mathfrak{F}_{2,s} = \{\overrightarrow{\alpha};\ s \cdot \overrightarrow{\alpha} \in \mathfrak{F}(2)\} = \mathfrak{P}(S\,|2|)$; in particular $\mathfrak{F}_{2,s}^{\perp} = \mathfrak{P}_{\mathrm{f}}(S\,|2|)$. 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 3.9. The transport functor ! in <u>Fin</u> 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 <u>Fin</u> is derived from the transport situation $\left(\bigoplus, \overrightarrow{\mathsf{rest}}\right)$ (resp. $\left(\bigoplus, \overrightarrow{\mathsf{rest}}, \mathsf{indx}\right)$) with shape relation indx (resp. $\emptyset$). Finally, we only consider finite tensor products: the binary functor $\otimes$ in $\underline{\mathrm{Fin}}$ 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

It is well known that $\underline{\mathrm{Rel}}$ endowed with the inclusion order is a complete lattice. Besides the fixpoint of any $n+1$-ary functor exists and is an n-ary functor in $\underline{\mathrm{Rel}}$. The situation in $\underline{\mathrm{Fin}}$ is more complex as the transport of finiteness structures and finitary relations is far from automatic. First, we describe the different orders that can endow $\underline{\mathrm{Fin}}$ and the continuity notions that come with. Then, we apply the transport lemma to address the problem of fixpoints of continuous functor in $\underline{\mathrm{Fin}}$. Actually, we prove that any transport functor admits a fixpoint which is a finiteness space. However we do not know at the time of the redaction if the fixpoint of any $n+1$-ary (transport) functor is an n-ary (transport) functor in $\underline{\mathrm{Fin}}$.

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, $\mathcal{A} \sqsubseteq\!\!\!\sqsubseteq \mathcal{B}$ 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 $\mathcal{A} \sqsubseteq\!\!\!\sqsubseteq \mathcal{B}$ simultaneously. Moreover $\top$ is the minimum of each of these orders. From now on, we consider only $\sqsubseteq$ and $\preceq$: the properties of $\sqsubseteq\!\!\!\sqsubseteq$ 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 upper 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 $\mathfrak{F}\left(\bigsqcap \overrightarrow{\mathcal{A}}\right) = \bigcap \overrightarrow{\mathfrak{F}(\mathcal{A})}$. Hence finiteness spaces form a complete lattice w.r.t. $\sqsubseteq$.

Proof. This is a general fact for fixpoints of 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 $|\bigsqcup \mathcal{A}|$ by itself, hence the bidual closure in $\mathfrak{F}\left(\bigsqcup \overrightarrow{\mathcal{A}}\right)$:

Counter-example 4.1. Let $\mathfrak{F}$ be any fake finiteness structure on some set A, that is such that $\mathfrak{F} \subsetneq \mathfrak{F}^{\perp\perp_A}$. For all $f \in \mathfrak{F}$, let $\mathcal{A}_f = (f, \mathfrak{P}(f))$. Then $\bigcup_{f\in\mathfrak{F}} \mathfrak{F}(\mathcal{A}_f) = \mathfrak{F}$, but $\mathfrak{F}\left(\bigsqcup_{t\in\mathfrak{F}} \mathcal{A}_t\right) = \mathfrak{F}^{\perp\perp_A}$.

When however $\bigcup \overrightarrow{\mathfrak{F}(\mathcal{A})}$ is a finiteness strucutre, we have that $\mathfrak{F}(\bigsqcup \mathcal{A}) = \bigcup \overrightarrow{\mathfrak{F}(\mathcal{A})}$ 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 $\mathcal{A}' \preceq \mathcal{B}$.[‡]

Counter-example 4.2. Let $\overrightarrow{\mathcal{F}} = (\mathcal{F}_n)_{n\in\mathbf{N}}$ be the unique sequence of finiteness spaces such that, for all $n \in \mathbf{N}$, $|\mathcal{F}_n| = \{0, \ldots, n-1\}$: then any finiteness space of web $\mathbf{N}$ is a $\preceq$-upper bound of all the $\mathcal{F}_n$'s, hence a $\precsim$-upper bound; but, e.g., $\mathcal{N}$ and $\mathcal{N}^\perp$ have no common $\precsim$-lower bound.

Notice however that in that case $\bigsqcup \overrightarrow{\mathcal{F}} = \mathcal{N}$ is an exact supremum. This remark is actually an instance of a more general fact. We call *extension sequence* any $\preceq$-increasing sequence of finiteness spaces. Then:

Lemma 4.2. If $\overrightarrow{A}$ is an extension sequence then $\bigsqcup \overrightarrow{A}$ is exact.

Proof. Apply the transport lemma to the following $(\{*\} \cup \mathbf{N})$-indexed family $\overrightarrow{f}$ of relations: $f_* = \{(\alpha, n);\ \alpha \notin |\mathcal{A}_n|\}$ and for all $n \in \mathbf{N}$, $f_n = \mathsf{id}_{|\mathcal{A}_n|}$. Then the reader can easily check that $\bigcup \overrightarrow{\mathfrak{F}(\mathcal{A})} = \mathfrak{F}_{\overrightarrow{f}}$. □

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 4.3 (A directed family for finiteness extension). Notice that the family of finiteness spaces in Counter-example 4.1 is directed for $\preceq$.

Counter-example 4.4 (An increasing sequence for finiteness inclusion). Consider the sequence of finiteness structures $(\mathfrak{C}_n)_{n\in\mathbf{N}}$ of Counter-example 3.2: it 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 $\sqsubseteq$.

Lemma 4.2, 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 4.2, for all $\preceq$-increasing functor T from <u>Fin</u> to <u>Fin</u>, $\mu T = \bigsqcup_{n\in\mathbf{N}} T^n \top$ is exact. In the following we show that this defines the least fixpoint of T up to some hypotheses on T w.r.t. both finiteness inclusion and finiteness extension.

‡ This preorder is considered in the unpublished preliminary version of (Ehr05), where it is used to describe the interpretation of second order quantification of linear logic. The counter example is given there.

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 T in $\underline{\mathrm{Fin}}$ is:

— *weakly continuous* if T commutes to directed suprema when they are exact;
— *exactly continuous* if T commutes to exact directed suprema;
— *directly continuous* if T commutes to all directed suprema.

Let us precise the second case: T is exactly continuous iff it is weakly continuous and, $T \bigsqcup \overrightarrow{\mathcal{A}} = \bigsqcup \overrightarrow{T\mathcal{A}}$ 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 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 T and U are weakly continous functors and $\bigsqcup \overrightarrow{\mathcal{A}}$ is an exact directed supremum, we do not know whether $\bigsqcup \overrightarrow{T\mathcal{A}}$ is exact, hence we can not deduce that 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 4.3. If T is a weakly continuous functor, which is moreover $\preceq$-increasing, then $\mu T = \bigsqcup_{n \in \mathbf{N}} T^n \top$ is the ($\sqsubseteq$-)least fixpoint of T.

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

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

$$T\left(\overrightarrow{\bigsqcup \overrightarrow{\mathcal{A}}}\right) = \bigsqcup \overrightarrow{T \overrightarrow{\mathcal{A}}}. \tag{1}$$

Definition 4.5. Let T be a $\sqsubseteq$-monotonic functor from $\underline{\mathrm{Fin}}^I$ to $\underline{\mathrm{Fin}}$. We say T is:

— *exactly continuous* if it commutes to all exact componentwise directed suprema, i.e.

$\bigsqcup \overrightarrow{\mathcal{T}\overrightarrow{\mathcal{A}}}$ is exact and Equation 1 holds and as soon as $\overrightarrow{\overrightarrow{\mathcal{A}}}$ is componentwise directed and $\overrightarrow{\bigsqcup \overrightarrow{\mathcal{A}}}$ is exact;

— *directly continuous* if it commutes to all componentwise directed suprema, i.e. Equation 1 holds as soon as $\overrightarrow{\overrightarrow{\mathcal{A}}}$ is componentwise directed.

The following result follows from the associativity of suprema:

Lemma 4.4. Directly continuous (resp. exactly continuous) functors compose: if T is a directly continuous (resp. exactly continuous) functor from $\underline{\mathrm{Fin}}^I$ to $\underline{\mathrm{Fin}}$ and, for all $i \in I$, each U_i is a directly continuous (resp. exactly continuous) functor from $\underline{\mathrm{Fin}}^{J_i}$ to $\underline{\mathrm{Fin}}$ then $T \circ \overrightarrow{U}$ is a directly continuous (resp. exactly continuous) functor from $\underline{\mathrm{Fin}}^{\sum \overrightarrow{J}}$ to $\underline{\mathrm{Fin}}$.

We do not detail the proof as it amounts to a futile exercise in formality: we have 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*

Lemma 4.5. Transport functors are monotonic for both $\preceq$ and $\sqsubseteq$.

Proof. Let $\mathcal{T}$ be a transport functor with ownership relation $\overrightarrow{\mathsf{own}}$ and assume $\overrightarrow{\mathcal{A} \sqsubseteq \mathcal{B}}$. First recall that $\left|\mathcal{T}\overrightarrow{\mathcal{A}}\right| \subseteq \left|\mathcal{T}\overrightarrow{\mathcal{B}}\right|$ as a general fact (see Section 2.2). Then let $\widetilde{a} \in \left|\mathcal{T}\overrightarrow{\mathcal{A}}\right|$. We have $\widetilde{a} \in \mathfrak{F}\left(\mathcal{T}\overrightarrow{\mathcal{A}}\right)$ iff for all $i \in I$, $\mathsf{own}_i^{\overrightarrow{|\mathcal{A}|}} \cdot \widetilde{a} \in \mathfrak{F}\left(\mathcal{A}_i\right)$. Since $\overrightarrow{|\mathcal{A}| \subseteq |\mathcal{B}|}$ and own_i is a lax natural transformation, $\mathsf{own}_i^{\overrightarrow{|\mathcal{A}|}} \cdot \widetilde{a} = \mathsf{own}_i^{\overrightarrow{|\mathcal{B}|}} \cdot \widetilde{a}$. Moreover, $\mathfrak{F}\left(\mathcal{A}_i\right) \subseteq \mathfrak{F}\left(\mathcal{B}_i\right)$, hence $\mathsf{own}_i^{\overrightarrow{|\mathcal{B}|}} \cdot \widetilde{a} \in \mathfrak{F}\left(\mathcal{B}_i\right)$. We thus obtain $\widetilde{a} \in \mathfrak{F}\left(\mathcal{T}\overrightarrow{\mathcal{B}}\right)$.

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

Lemma 4.6. A transport functor is exactly continuous as soon as its underlying web functor is continuous.

Proof. Let $\mathcal{T}$ be a transport functor from $\underline{\mathrm{Fin}}^I$ to $\underline{\mathrm{Fin}}$, with ownership $\overrightarrow{\mathsf{own}}$ and underlying continuous web functor T. Let $\overrightarrow{\overrightarrow{\mathcal{A}}}$ be directed and such that each $\bigsqcup \overrightarrow{\mathcal{A}}_i$ is exact. We prove that $\mathcal{T}\overrightarrow{\bigsqcup \overrightarrow{\mathcal{A}}} = \bigsqcup \overrightarrow{\mathcal{T}\overrightarrow{\mathcal{A}}}$. First notice that the webs $\left|\mathcal{T}\overrightarrow{\bigsqcup \overrightarrow{\mathcal{A}}}\right| = T\overrightarrow{\bigcup \overrightarrow{|\mathcal{A}|}}$ and $\left|\bigsqcup \overrightarrow{\mathcal{T}\overrightarrow{\mathcal{A}}}\right| = \bigcup \overrightarrow{T\overrightarrow{|\mathcal{A}|}}$ are equal because T is continuous. We are left to prove the equality of finiteness structure, that is $\mathfrak{F}\left(\mathcal{T}\overrightarrow{\bigsqcup \overrightarrow{\mathcal{A}}}\right) = \mathfrak{F}\left(\bigsqcup \overrightarrow{\mathcal{T}\overrightarrow{\mathcal{A}}}\right)$ or equivalently $\mathfrak{F}\left(\mathcal{T}\overrightarrow{\bigsqcup \overrightarrow{\mathcal{A}}}\right)^{\perp} = \mathfrak{F}\left(\bigsqcup \overrightarrow{\mathcal{T}\overrightarrow{\mathcal{A}}}\right)^{\perp}$. Let's make explicit that by definition:

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

(b) $\widetilde{a}' \in \mathfrak{F}\left(\bigsqcup \overrightarrow{\mathcal{T}\mathcal{A}}\right)^{\perp}$ iff for all $\overrightarrow{j \in J}$ 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 $\overrightarrow{j \in J}$ and $\widetilde{a} \in \mathfrak{F}\left(\mathcal{T}\overrightarrow{\mathcal{A}}_{\overrightarrow{j}}\right)$. For all $i \in I$, let $a_i = \mathsf{own}_i^{\left|\overrightarrow{\mathcal{A}}_{\overrightarrow{j}}\right|} \cdot \widetilde{a}$: $a_i \in \mathfrak{F}(\mathcal{A}_{i,j_i}) \subseteq \bigcup \overrightarrow{\mathfrak{F}(\mathcal{A}_i)} \subseteq \mathfrak{F}\left(\bigsqcup \overrightarrow{\mathcal{A}_i}\right)$. Since $\widetilde{a} \subseteq \overrightarrow{\mathsf{own}} \setminus \overrightarrow{a}$ and by condition (a), we deduce that $\widetilde{a}' \cap \widetilde{a} \subseteq \widetilde{a}' \cap \left(\overrightarrow{\mathsf{own}} \setminus \overrightarrow{a}\right)$ is finite.

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

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

Example 4.6. The web functors of products, sums and finite multisets are all continuous, hence $\&$, $\bigoplus$, ! 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 $j \neq i$;
— own_i preserves intersections, i.e. $\mathsf{own}_i \cdot \bigcap \overrightarrow{\widetilde{a}} = \bigcap \overrightarrow{\mathsf{own}_i \cdot \widetilde{a}}$ for all $\overrightarrow{\widetilde{a} \in TA}$.

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 4.7. A 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: $\mathsf{own}_i \cdot \widetilde{a}' \in \mathfrak{F}\left(\bigsqcup \overrightarrow{\mathcal{A}_i}\right)^{\perp} = \left(\bigcup \overrightarrow{\mathfrak{F}(\mathcal{A}_i)}\right)^{\perp}$ for all $i \in I$. Indeed, let $\overrightarrow{j \in J}$ 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 $\mathcal{A}_{i,j_i}$): then $\overrightarrow{\mathsf{own}} \setminus \overrightarrow{a} \in \mathfrak{F}\left(\mathcal{T}\overrightarrow{\mathcal{A}}_{\overrightarrow{j}}\right)$ and thus $\widetilde{a} = \widetilde{a}' \cap \left(\overrightarrow{\mathsf{own}} \setminus \overrightarrow{a}\right)$ is finite. Moreover, for all $i \in I$, $\mathsf{own}_i \cdot \overrightarrow{\mathsf{own}} \setminus \overrightarrow{a} = a_i$, because $\overrightarrow{\mathsf{own}}$ is local. Hence $(\mathsf{own}_i \cdot \widetilde{a}') \cap a_i = (\mathsf{own}_i \cdot \widetilde{a}') \cap \left(\mathsf{own}_i \cdot \overrightarrow{\mathsf{own}} \setminus \overrightarrow{a}\right) = \mathsf{own}_i \cdot \widetilde{a}$ because own_i preserves intersections. Since $\widetilde{a}$ is finite and own_i is quasi-functional, we conclude that $\mathsf{own}_i \cdot \widetilde{a}' \perp_{\mathrm{f}} a_i$. Since this holds for all $a_i \in \bigcup \overrightarrow{\mathfrak{F}(\mathcal{A}_i)}$, we obtain $\mathsf{own}_i \cdot \widetilde{a}' \in \mathfrak{F}\left(\bigsqcup \overrightarrow{\mathcal{A}_i}\right)^{\perp}$.

Then let $\overrightarrow{a}$ be such that $a_i \in \mathfrak{F}\left(\bigsqcup \overrightarrow{\mathcal{A}}_i\right)$ for all $i \in I$. We must show that $\widetilde{a}'' = \widetilde{a}' \cap \left(\overrightarrow{\mathsf{own}} \setminus \overrightarrow{a}\right)$ is finite. For all $i \in I$, $a''_i = \mathsf{own}_i \cdot \widetilde{a}'' \subseteq (\mathsf{own}_i \cdot \widetilde{a}') \cap a_i$ 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. $\square$

Example 4.7. Since $\overrightarrow{\mathsf{rest}}$, indx and supp are local, $\&$, $\bigoplus$ and $!$ are directly continuous.

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

Lemma 4.8. 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 $\overrightarrow{\mathcal{B}} = (\mathcal{B}_j)_{j\in J}$ be a directed supremum of finiteness spaces: we prove $\mathcal{A} \otimes \bigsqcup \overrightarrow{\mathcal{B}} = \bigsqcup (\mathcal{A} \otimes \mathcal{B}_j)_{j\in J}$ or, equivalently, $\mathcal{A} \otimes \bigsqcup \overrightarrow{\mathcal{B}}^{\perp} = \bigsqcup (\mathcal{A} \otimes \mathcal{B}_j)^{\perp}_{j\in J}$. Let $c' \subseteq |\mathcal{A}| \times \bigcup |\overrightarrow{\mathcal{B}}|$. That $c' \in \mathfrak{F}\left(\mathcal{A} \otimes \bigsqcup \overrightarrow{\mathcal{B}}\right)^{\perp}$ implies $c' \in \mathfrak{F}\left(\bigsqcup (\mathcal{A} \otimes \mathcal{B}_j)_{j\in J}\right)^{\perp}$ goes by the same argument as in Lemma 4.6. Assume that $c' \in \mathfrak{F}\left(\bigsqcup (\mathcal{A} \otimes \mathcal{B}_j)_{j\in J}\right)^{\perp}$: we prove that $c' \in \mathfrak{F}\left(\mathcal{A} \otimes \bigsqcup \overrightarrow{\mathcal{B}}\right)^{\perp} = \mathfrak{F}\left(\mathcal{A} \multimap \bigsqcup \overrightarrow{\mathcal{B}}^{\perp}\right)$. If $a \in \mathfrak{F}(\mathcal{A})$ then $c' \cdot a \in \mathfrak{F}\left(\bigsqcup \overrightarrow{\mathcal{B}}^{\perp}\right)$. 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 $\beta \in \left|\bigsqcup \overrightarrow{\mathcal{B}}\right| = \bigcup |\overrightarrow{\mathcal{B}}|$: 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 ${}^t c' \cdot \beta \in \mathfrak{F}(\mathcal{A}^{\perp})$, which concludes the proof. $\square$

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 4.7, and thus generalize direct continuity to all transport functors with continuous web functors.

5. The finitary relational model of the λ-calculus

It is a well known fact that Rel is a model of classical linear logic, and even of differential linear logic where:

- linear negation is the contravariant functor which is the identity on sets and sends every relation f to ${}^t f$;
- multiplicatives are interpreted by cartesian products;
- additives are interpreted by disjoint unions;
- exponentials are interpreted by finite multisets.[§]

[§] The reader not familiar with this folklore construction may refer to the appendix of (Ehr05) where it is explicitly described.

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 seminal paper (Ehr05, Section 1). This result could actually be stated as follows: for all 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 λ-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 λ-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 subsets of $A \Rightarrow B = \mathfrak{M}_{\mathrm{f}}(A) \times B$, which we call *multirelations*. Composition of multirelations is given by

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

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 $\mathop{\&} \overrightarrow{A}$, 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 $\mathop{\&} \overrightarrow{B}$ 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 is the empty set $\emptyset$, the unique multirelation from A to $\emptyset$ being empty. The adjunction for closedness is $\underline{\mathrm{Rel}}^!(A+B, C) \cong \underline{\mathrm{Rel}}^!(A, \mathfrak{M}_{\mathrm{f}}(B) \times C)$ which boils down to the 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 typed λ-calculi with products. Type and term expressions are given by:

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

where X ranges over a fixed set $\mathfrak{A}$ of atomic types, x ranges over term variables and a ranges over term constants. To each variable or constant, we associate a type, and 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 Γ and Δ range over lists $(x_1 : A_1, \ldots, x_n : A_n)$ of typed variables. The operational semantics of a typed λ-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 Γ. 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 $(\lambda x\, s)\, t \simeq_0 s\, [x := t]$ (with the obvious assumptions ensuring typability).

Assume a set $[\![X]\!]$ is given for each base 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 rela-

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

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

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

Figure 1. Rules of typed λ-calculi with products

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

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

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

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

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

Figure 2. Computing points in the relational semantics

tional semantics of a derivable typing judgement $x_1 : A_1, \ldots, x_n : A_n \vdash s : A$ will be a relation $[\![s]\!]_{x_1:A_1,\ldots,x_n:A_n} \subseteq [\![A_1 \Rightarrow \cdots \Rightarrow A_n \Rightarrow A]\!]$. We first introduce the deductive system of Figure 2, which is a straightforward adaptation of de Carvalho's system R to the simply typed case (dC08). 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 $\overline{\alpha}_i \in [\![A_i]\!]^!$ and $\alpha \in [\![A]\!]$. In rules $[\![\mathrm{Var}]\!]$ and $[\![\mathrm{Const}]\!]$, $\Gamma^{[]}$ denotes an annotated context of the form $x_1^{[]} : A_1, \ldots, x_n^{[]} : A_n$. In rule $[\![\mathrm{App}]\!]$, 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 there is no rule for $\langle\rangle$ in Figure 2, because $[\![\langle\rangle]\!]_{\Gamma} = \emptyset$.

Since we followed the standard interpretation of typed λ-calculi in cartesian closed categories, 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 $\mathcal{A} \Rightarrow \mathcal{B} = !\mathcal{A} \multimap \mathcal{B}$. As a straightforward consequence of Lemma 3.3 and the characterization of $!\mathcal{A}$, we obtain:

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 $\beta \in |\mathcal{B}|$, ${}^t f \cdot \beta \perp_{\mathrm{f}} a^!$.

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 none by the co-Kleisli category derived from $\underline{\mathrm{Fin}}$. The relational interpretation thus defines a semantics in $\underline{\mathrm{Fin}}^!$ as follows. Assume a finiteness structure $\mathfrak{F}(X)$ is given for all atomic type X, so that $X^* = (\llbracket X \rrbracket, \mathfrak{F}(X))$ is a finiteness space, and set $(A \Rightarrow B)^* = A^* \Rightarrow B^*$, $(A \mathbin{\&} B)^* = A^* \mathbin{\&} B^*$ and $\top^* = \top$. Then, further assuming that, for all $a \in \mathfrak{C}_A$, $\llbracket a \rrbracket \in \mathfrak{F}(A^*)$, we obtain:

Lemma 5.3 (Finiteness). If $x_1 : A_1, \ldots, x_n : A_n \vdash s : A$ then $\llbracket s \rrbracket_{x_1:A_1,\ldots,x_n:A_n} \in \mathfrak{F}(A_1^* \Rightarrow \cdots \Rightarrow A_n^* \Rightarrow A^*)$.

5.2. *On the relations denoted by λ-terms*

Pure typed λ-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 all 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 (or a fixed set of sets) and the constants $\mathfrak{C}_A = \mathfrak{P}(\llbracket A \rrbracket)$. Then set $s \simeq_{\underline{\mathrm{Rel}}} t$ iff $\llbracket s \rrbracket_\Gamma = \llbracket t \rrbracket_\Gamma$, for any suitable Γ. 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 $id_A = \lambda x\, x$ 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 Γ, we may simply write $\Gamma \vdash s = t : A$ for $\llbracket s \rrbracket_\Gamma = \llbracket t \rrbracket_\Gamma \in \llbracket A \rrbracket$. 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 λ-calculi with extensionality: $s : A \Rightarrow A \vdash \lambda x\, (s x) = s$ as soon as x is not free in s. Moreover, they admit all products, and they are models of λ-calculi with surjective tuples of arbitrary arity: $t : \mathop{\&} \overrightarrow{A} \vdash \langle \pi_i t \rangle_{i \in I} = t$. In accordance with this last remark, we may identify any variable of type $\mathop{\&} \overrightarrow{A}$ with the corresponding 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 given by $\mathsf{fix} = \bigcup_{n \in \mathbf{N}} \mathsf{fix}_n \subseteq (A \Rightarrow A) \Rightarrow A$ where $\mathsf{fix}_0 = \emptyset$ and

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

Indeed, $\mathsf{fix}_{n+1} = \lambda f\, (f\, (\mathsf{fix}_n\, f))$ so fix is the least multirelation such that $\mathsf{fix} = \lambda f\, (f\, (\mathsf{fix}_n\, f))$. 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, fix is not finitary: Ehrhard details a counter-example (Ehr05, Section 3), but we can actually show that the fixpoint operator is never finitary on non-empty webs.

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 $f = \{([], \alpha)\} \cup \{([\alpha], \alpha)\} \in \mathfrak{P}_{\mathrm{f}}(\mathcal{A} \Rightarrow \mathcal{A}) \subseteq \mathfrak{F}(\mathcal{A} \Rightarrow \mathcal{A})$. 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 $f^! \cap \left({}^t\mathsf{fix} \cdot \alpha\right)$ is infinite although $f \in \mathfrak{F}\,(\mathcal{A} \Rightarrow \mathcal{A})$. $\square$

More informally, this result indicates that the finitary semantics strongly refuses infinite computations and will not accomodate general recursion. 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 (Ehr05) that one can model a restricted form of tail-recursive iteration. In recent work (Vau09c), the second author showed that the finitary relational model of the λ-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

Intuitively, an algebraic datatype is a composite of products and sums of 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 conditional, or more generally 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 algebraic datatypes we obtain: 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 coherences semantics of system T (GTL89) and its relational variant (Vau09c). 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 in 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 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 α_0 and α_1, such that $([\,]\,, \alpha_k) \in i_k$ but $([\,]\,, \alpha_k) \notin i_{1-k}$ for $k = 0, 1$. Now consider $h' = \{([\alpha_0]\,, 0), ([\alpha_1]\,, 0)\}$ 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 intuitionnistic logic in coherence spaces (GTL89). We write $A \overset{\bullet}{\oplus} B$ for the lifted sum $\{1,2\} \cup A \oplus B$ of A and B, and more generally: $\overset{\bullet}{\bigoplus} \overrightarrow{A} = I \cup \bigoplus \overrightarrow{A}$.[¶] The idea is that indices stand for tokens without associated value: where (i, α) can be read as "the element α 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 $\overset{\bullet}{\bigoplus} \overrightarrow{f} = \bigoplus \overrightarrow{f} \cup \mathsf{id}^I$ so that $\overset{\bullet}{\bigoplus}$ is a continuous I-ary functor from $\underline{\mathrm{Rel}}$ to $\underline{\mathrm{Rel}}$. Now let $\overrightarrow{f}$ be a family of multirelations such that $f_i \in \underline{\mathrm{Rel}}^!(A_i, B)$ for all $i \in I$: we define

$$\left\{ \overrightarrow{f} \right\} = \{([i, (i, \alpha_1), \ldots, (i, \alpha_n)], \beta);\ ([\alpha_1, \ldots, \alpha_n], \beta) \in f_i\}$$

and obtain, for all $i \in I$, $\left\{ \overrightarrow{f} \right\} \circ^! \mathsf{inj} i = f_i$. 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, $\{([i, i, (i, \alpha_1), \ldots, (i, \alpha_n)], \beta);\ ([\alpha_1, \ldots, \alpha_n], \beta) \in f_i\}$ behaves similarly (we just added a copy of the token $i \in I$). This "case definition" construction can be internalized as a multirelation:

$$\begin{aligned} \mathsf{case}_{\overrightarrow{A}, B} &= \left[\!\left[\lambda \overrightarrow{f} \left\{ \overrightarrow{f} \right\} \right]\!\right] \\ &= \{([i] + i\overline{\alpha}, [(i, (\overline{\alpha}, \beta))], \beta);\ i \in I \wedge \overline{\alpha} \in {!A_i} \wedge \beta \in B\} \\ &\subseteq \overset{\bullet}{\bigoplus} \overrightarrow{A} \Rightarrow \mathop{\&}_{i \in I} (A_i \Rightarrow B) \Rightarrow B \end{aligned}$$

For all $i \in I$, the restriction $\mathsf{rest}_i^{\overrightarrow{A}}$ is a quasi-functional lax natural transformation from $\overset{\bullet}{\bigoplus}$ to Π_i. The same holds for the index relation $\mathsf{indx} \cup \mathsf{id}^I$. 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: indx itself (see Lemma 3.4). 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 6.1. For all finiteness spaces $\overrightarrow{\mathcal{A}}$ and $\mathcal{B}$:

$$\overrightarrow{f} : \mathop{\&}_{i \in I} (\mathcal{A}_i \Rightarrow \mathcal{B})], a : \mathcal{A}_i \vdash \mathsf{case}\, \overrightarrow{f}\, (\mathsf{inj}_i a) = f_i a$$

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

[¶] 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.

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

We call *algebraic datatype*, any functor build from projections, $\top$, $\&$ and $\dot{\bigoplus}$. The most basic example of composite datatype is that of booleans, $\mathcal{B}ool = \top \dot{\oplus} \top$: 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 $\mathsf{if}^A = \{([tt], [\alpha], [\,], \alpha);\; \alpha \in A\} \cup \{([tt], [\,], [\alpha], \alpha);\; \alpha \in A\}$ 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. *Fixpoints of power series functors*

We now explore the interpretation of recursive algebraic datatypes that we obtain, by a careful inspection of the structure of tree types. First consider the following construction of a finiteness space of trees:

Counter-example 6.2. Let $\mathcal{A}$ and $\mathcal{B}$ be finiteness spaces. The functor $\mathcal{T} : \mathcal{X} \mapsto \mathcal{A} \oplus (\mathcal{X} \otimes \mathcal{B} \otimes \mathcal{X})$ in $\underline{\mathrm{Fin}}$ is clearly a transport functor. By Lemma 4.3, its fixpoint $\mu\,\mathcal{T}$ is a finiteness space, which we describe as follows:

- $|\mu\,\mathcal{T}|$ is the set of all finite binary trees, with nodes labelled by elements of $|\mathcal{A}|$ and leaves 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 $\mathcal{A}$ (resp. $\mathcal{B}$) 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 $\mathcal{A}$ and leaves of type $\mathcal{B}$. Indeed, due to the fact that $\oplus$ does not define a sum, we would fail to define a suitable relational interpretation of pattern matching for this type of trees. Notice 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, the discussion on sums and previous work on the semantics of system T, we are led to study the finiteness properties of the datatype of trees, obtained as fixpoints of power series functors. Let I be a set of indices, $\overrightarrow{\mathcal{A}}$ a family of finiteness spaces and $\overrightarrow{J}$ a family of sets of indices.

Proposition 6.3. The functor $\mathcal{T} : \mathcal{X} \mapsto \dot{\bigoplus}_{i \in I} \mathcal{A}_i \,\&\, \mathcal{X}^{\& j_i}$ is exactly continuous, $\preceq$-increasing and has a least fixpoint $\mu\,\mathcal{T}$ which is a finiteness space.

Proof. As the functor $\mathcal{T}$ is the composite of exactly continuous and $\preceq$-increasing functors, it is also exactly continuous (see Lemma 4.4) and $\preceq$-increasing. The existence of the least fixpoint in $\underline{\mathrm{Fin}}$ is then a direct consequence of Lemma 4.3. $\square$

Intuitively $\mu\mathcal{T}$ is the datatype of trees, in which nodes of sort $i \in I$ are of arity J_i and bear labels in $\mathcal{A}_i$: this will be made formal in the next section. We denote $\mu\mathcal{T}$ by $\mathcal{L}_{I,\overrightarrow{J}}\overrightarrow{\mathcal{A}}$ or, shortly $\mathcal{L}\overrightarrow{\mathcal{A}}$.

Let us first describe the associated web functor $L_{I,\overrightarrow{J}} = \left|\mathcal{L}_{I,\overrightarrow{J}}\right|$: $L_{I,\overrightarrow{J}}\overrightarrow{A} = \bigcup_{n\in\mathbf{N}} T^n\emptyset$ where $T : X \mapsto \dot{\bigoplus}_{i\in I} A_i \,\&\, X^{\& J_i}$. Again, we will simply write $L\overrightarrow{A}$ in general.

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 τ in $L\overrightarrow{A}$. Hence the general form of an element $\tau \in L\overrightarrow{A}$ is: $\tau = (i_0,(2,(j_1,(i_1,(2,\dots(j_n,i_n)\cdots)))))$ or $\tau = (i_0,(2,(j_1,(i_1,(2,\dots(j_n,(i_n,(1,\alpha)))\cdots)))))$ where $j_{k+1} \in J_{i_k}$ for all $k < n$ and, in the second case, $\alpha \in A_{i_n}$. We introduce the following conventions for the sole purpose of making this description of the elements $L\overrightarrow{A}$ more reasonable. We denote finite sequences of indices by plain concatenation: $\overrightarrow{i} = i_1\cdots i_n$. If $i_0 \in I$, we call ***addresses of type*** i_0 all finite sequences $\overrightarrow{ji} = j_1i_1j_2\cdots j_ni_n$ such that $j_{k+1} \in J_{i_k}$ for all $k < n$. We call ***typed address*** the data π of an index i and an address $\overrightarrow{ji}$ of type i, which we write $i\overrightarrow{ji}$, or simply i if $\overrightarrow{ji}$ is the empty sequence. If $\pi = i_0j_1i_1\cdots j_ni_n$ we write $\omega\pi = i_n$; in particular $\omega i = i$. We call ***value of type*** i any element of $A_i \cup \{*\}$ (we suppose this union is always disjoint): we use starred letters such as α^* for values. A ***path*** is the data $\pi\alpha^*$ of a typed address π and a value of type $\omega\pi$. We write $\mathbf{A}_i$ for the set of addresses of type i, and $\mathbf{A}$ for the set of all typed addresses. We may factor prefixes out of multisets of paths or addresses: for instance, if $\overline{\tau} = [\tau_1,\dots,\tau_n] \in !\mathbf{A}_i$ is a multiset of paths, we may write $ij\overline{\tau} = [ij\tau_1,\dots,ij\tau_n]$. Notice that $L\overrightarrow{A}$ is in bijection with the set of all paths $\{\pi\alpha^*;\ \pi \in \mathbf{A} \wedge \alpha^* \in A_{\omega\pi} \cup \{*\}\}$. From now on we consider $L\overrightarrow{A}$, and thus $\mathcal{L}\overrightarrow{\mathcal{A}}$, up to this bijection.

Notice that the relation between paths and values of type i given by

$$\mathsf{val}_i^{\overrightarrow{A}} = \{(\pi\alpha,\alpha);\ \pi \in \mathbf{A} \wedge \omega\pi = i \wedge \alpha \in A_i\}$$

is a quasi-functional lax natural transformation from L to the projection functor Π_i for all $i \in I$. Moreover, the length relation

$$\mathsf{len}^{\overrightarrow{A}} = \left\{(i_0j_1i_1\cdots j_ni_n\alpha^*,n);\ i_0\overrightarrow{ji}\alpha^* \in L\overrightarrow{A}\right\}$$

is a quasi-functional lax natural transformation from L to $E_{\mathbf{N}}$ where $\mathbf{N}$ is the set of shapes made of natural numbers (see Section 3.4). Now, because T is exact and $\mathcal{L}\overrightarrow{\mathcal{A}}$ is defined as the least fixpoint of T, we can characterize its action on finiteness structures:

Lemma 6.2. $t \in \mathfrak{F}\left(\mathcal{L}\overrightarrow{\mathcal{A}}\right)$ iff $\mathsf{val}_i \cdot t \in \mathfrak{F}(A_i)$ and $\mathsf{len} \cdot t \in \mathfrak{P}_{\mathrm{f}}(\mathbf{N})$.

We could thus have presented $\mathcal{L}$ equivalently as the functor[‖] of paths, with web functor L, finiteness transported by $\overrightarrow{\mathsf{val}}$ and len, and equipped with the shape relation len (see Lemma 3.4).

6.3. *The finitary datatype of trees*

We are now ready to describe the interpretation of the datatype of tree. In particular, in this section we show that:

— $\mathcal{L}$ 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.

Remark that a similar description would be feasible for fixpoints of algebraic functors other that power series, although we cannot simply rely on some simple distributivity argument: in general $(A \overset{\bullet}{\oplus} B) \,\&\, C \not\cong (A \,\&\, C) \overset{\bullet}{\oplus} (B \,\&\, C)$. But even when we consider arbitrary algebraic functors, we always end up with sets of addresses, where finitary subsets are those with bounded length and finitary support: the only difference is the form of the addresses. A general presentation would thus be overly technical, without any new idea involved.

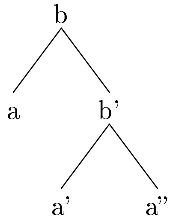
The *lazy tree constructor* $\mathsf{node}_i \subseteq A_i \Rightarrow \left(L\overrightarrow{A}\right)^{\& J_i} \Rightarrow L\overrightarrow{A}$ is given by:

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

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 $\underline{\mathrm{Rel}}^!$.

Example 6.4. We consider the functor $\mathcal{T} : \mathcal{X} \mapsto \mathcal{A} \overset{\bullet}{\oplus} (\mathcal{B} \,\&\, (\mathcal{X} \,\&\, \mathcal{X}))$ defining the data structure of binary trees with leaves labelled by $\mathcal{A}$ and nodes labelled by $\mathcal{B}$: write $\mathcal{BT} = \mathcal{L}(\mathcal{A}, \mathcal{B})$. We set $I = \{\mathsf{F}, \mathsf{N}\}$, $J_\mathsf{F} = \emptyset$ and $J_\mathsf{N} = \{\mathsf{G}, \mathsf{D}\}$. Since no confusion is possible, we simply write G (resp. D) for the ordered pair NG (resp. ND). 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 $\mathsf{leaf} = \lambda x^{\mathcal{A}}\, \lambda\_\ (\mathsf{node}_\mathsf{F} x) \subseteq \mathcal{A} \to \mathcal{BT}$ and $\mathsf{node} = \lambda y^{\mathcal{B}}\, \lambda t^{\mathcal{BT}}\, \lambda u^{\mathcal{BT}}\, (\mathsf{node}_\mathsf{N} y\, \langle t, u\rangle) \subseteq \mathcal{B} \to \mathcal{BT} \to \mathcal{BT} \to \mathcal{BT}$. Now, let $a, a', a'' \subseteq A$ and $b, b' \subseteq B$ the tree:

[‖] Notice that we have proved with bare hand that the fixpoint $\mathcal{L}$ of the transport functor T is also a transport functor. It will be interesting to build a transport situation and a shape relation for every fixpoint of transport functor. We could then deduce that the fixpoint is a transport functor.



corresponds to

$$
\begin{array}{rcccc}
\mathsf{node}\, b\, (\mathsf{leaf}\, a)\, (\mathsf{node}\, b'\, (\mathsf{leaf}\, a')\, (\mathsf{leaf}\, a'')) = & & \{\mathsf{N}*\} & \cup & \{\mathsf{N}\beta;\ \alpha \in b\} \\
& \cup & \{\mathsf{GF}*\} & \cup & \{\mathsf{GF}\alpha;\ \alpha \in a\} \\
& \cup & \{\mathsf{DN}*\} & \cup & \{\mathsf{DN}\beta;\ \beta \in b'\} \\
& \cup & \{\mathsf{DGF}*\} & \cup & \{\mathsf{DGF}\alpha;\ \alpha \in a'\} \\
& \cup & \{\mathsf{DDF}*\} & \cup & \{\mathsf{DDF}\alpha;\ \alpha \in a''\}\,.
\end{array}
$$

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

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

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

$$
\begin{aligned}
\mathsf{node}_i \;&\in\; \mathfrak{F}\left(\mathcal{A}_i \Rightarrow \left(\mathcal{L}\overrightarrow{\mathcal{A}}\right)^{\& J_i} \Rightarrow \mathcal{L}\overrightarrow{\mathcal{A}}\right)\\
\mathsf{match} \;&\in\; \mathfrak{F}\left(\mathcal{L}\overrightarrow{\mathcal{A}} \Rightarrow \mathop{\&}_{i\in I}\left(\mathcal{A}_i \Rightarrow \left(\mathcal{L}\overrightarrow{\mathcal{A}}\right)^{\& J_i} \Rightarrow \mathcal{B}\right) \Rightarrow \mathcal{B}\right).
\end{aligned}
$$

As an application of Lemma 6.1, we thus obtain $\left[\!\left[\mathsf{match}\,\overrightarrow{f}\,\left(\mathsf{node}_i a\,\overrightarrow{t}\right)\right]\!\right] = \left[\!\left[f_i a\,\overrightarrow{t}\right]\!\right]$.

We can then construct the *iterator on trees*:

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

which automatically satisfies $\left[\!\left[\mathsf{iter}\left(\mathsf{node}_i a\,\overrightarrow{t}\right)\overrightarrow{f}\right]\!\right] = \left[\!\left[f_i a\left\langle \mathsf{iter}\,\overrightarrow{f}\,t_j\right\rangle_{j\in J_i}\right]\!\right]$. The following lemma makes the structure of iter explicit:

Lemma 6.3. Let $\mathsf{iter}_0 = \emptyset$ and, for all $n \in \mathbf{N}$, let

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

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. That the equation $\mathsf{iter} = \bigcup_{n\in\mathbf{N}} \mathsf{iter}_n$ is just an unfolding of the definitions: if we write $f = \left[\!\!\left[\left(\lambda F\,\lambda t\,\lambda\overrightarrow{f}\ \mathsf{match}\,t\left\langle\lambda a\,\lambda\overrightarrow{t}\ f_i a\left\langle F\overrightarrow{f}\,t_j\right\rangle_{j\in J_i}\right\rangle_{i\in I}\right)\right]\!\!\right]$ 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 $\tau = i_0 j_1 i_1 \cdots j_n i_n \alpha^* \in L\overrightarrow{A}$, we write $ind(\tau) = \{i_0, j_1, i_1, \ldots j_n i_n\}$ and $val(\tau) = \bigcup_{i\in I} \mathsf{val}_i \cdot \tau$ which is 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 $\overline{\alpha} \in \mathfrak{M}_{\mathrm{f}}\,(A)$, $\#\,\overline{\alpha}$ denotes the multiset cardinality of $\overline{\alpha}$. When $\overline{\phi} = \sum_{k=1}^n [(i_k, \overline{\alpha}_k, \overline{\tau}_k, \beta_k)] \in \mathfrak{M}_{\mathrm{f}}\left(\sum_{i\in I}\left(A_i \Rightarrow B^{\& J_i} \Rightarrow B\right)\right)$, we write $\#\#(\overline{\phi}) = \sum_{k=1}^n \#\,\overline{\alpha}_k$.

Lemma 6.4. For all $\iota = (\overline{\tau}, \overline{\phi}, \beta) \in \mathsf{iter}$, we have:

— $\iota \in \mathsf{iter}_{\max(\mathsf{len}\cdot supp(\overline{\tau}))+1}$;
— $ind(\overline{\tau}) = ind(\overline{\phi})$;
— $val(\overline{\tau}) = val(\overline{\phi})$;
— $\#\,\overline{\tau} = \#\,\overline{\phi} + \#\#\overline{\phi}$.

Lemma 6.5. 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 $n = \max(\mathsf{len}\cdot t)$. As a consequence $\mathsf{iter}\cdot\mathfrak{M}_{\mathrm{f}}\,(t) = \mathsf{iter}_{n+1}\cdot\mathfrak{M}_{\mathrm{f}}\,(t) \in \mathfrak{F}\left(\&_{i\in I}\left(\mathcal{A}_i \Rightarrow \mathcal{B}^{\& J_i} \Rightarrow \mathcal{B}\right) \Rightarrow \mathcal{B}\right)$ because 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 let $\overline{t}' = {}^t\mathsf{iter}\cdot(\overline{\phi}, \beta)$: we prove $\overline{t}' \perp_{\mathrm{f}} \mathfrak{M}_{\mathrm{f}}\,(t)$ is finite. By the previous lemma, for all $\overline{\tau} \in t'$, $val(\overline{\tau}) = val(\overline{\phi})$, $ind(\overline{\tau}) = ind(\overline{\phi})$, $\#\,\overline{\tau} = \#\,\overline{\phi} + \#\#\overline{\phi}$. 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 $\overline{t} \cap \mathfrak{M}_{\mathrm{f}}\,(t')$ are of fixed size: hence $\overline{t} \cap \mathfrak{M}_{\mathrm{f}}\,(t')$. □

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

Theorem 6.5. For all choice of sets of indices I and $\overrightarrow{J}$, $\mathcal{L}\overrightarrow{\mathcal{A}}$ is the finiteness spaces of paths whose finiteness is transported by $\overrightarrow{\mathsf{val}}$ and len. Moreover, there are multirelations node_i and iter such that:

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

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 case where $I = \{0, 1\}$, $\# J_0 = 0$, $\# J_1 = 1$ and $\mathcal{A}_0 = \mathcal{A}_1 = \top$. Then $\mathcal{L} \cong \mathcal{N}_l$ 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 address π such that $\mathsf{len} \cdot \pi = n$ and $\omega\pi = 0$ (resp. $\omega\pi = 1$). 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 inj_0 and 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 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 (Vau09c) 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 iter with

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

which automatically satisfies

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

We would then obtain 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

Roland Carl Backhouse, Peter J. de Bruin, Paul F. Hoogendijk, Grant Malcolm, Ed Voermans, and Jaap van der Woude. Polynomial relators. In Maurice Nivat, Charles Rattray, Teodor Rus, and Giuseppe Scollo, editors, *AMAST*, Workshops in Computing, pages 303–326. Springer, 1991.

Antonio Bucciarelli, Thomas Ehrhard, and Giulio Manzonetto. Not enough points is enough. In *Computer Science Logic*, volume 4646 of *Lecture Notes in Computer Science*, pages 298–312. Springer Berlin, 2007.

Roland Carl Backhouse and Paul F. Hoogendijk. Generic properties of datatypes. In Roland Carl Backhouse and Jeremy Gibbons, editors, *Generic Programming*, volume 2793 of *Lecture Notes in Computer Science*, pages 97–132. Springer, 2003.

David Baelde and Dale Miller. Least and greatest fixed points in linear logic. In Nachum Dershowitz and Andrei Voronkov, editors, *LPAR*, volume 4790 of *Lecture Notes in Computer Science*, pages 92–106. Springer, 2007.

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

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

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

Pierre-Louis Curien, editor. *Typed Lambda Calculi and Applications, 9th International Conference, TLCA 2009, Brasilia, Brazil, July 1-3, 2009. Proceedings*, volume 5608 of *Lecture Notes in Computer Science*. Springer, 2009.

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

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

Thomas Ehrhard and Olivier Laurent. Interpreting a finitary pi-calculus in differential interaction nets. In Luis Caires and Vasco T. Vasconcelos, editors, *Concurrency Theory (CONCUR '07)*, volume 4703 of *Lecture Notes in Computer Science*, pages 333–348. Springer, September 2007.

Thomas Ehrhard and Laurent Regnier. The differential lambda-calculus. *Theoretical Computer Science*, 309:1–41, 2003.

Thomas Ehrhard and Laurent Regnier. Differential interaction nets. *Electr. Notes Theor. Comput. Sci.*, 123:35–74, 2005.

Thomas Ehrhard and Laurent Regnier. Böhm trees, Krivine's machine and the Taylor expansion of λ-terms. In Arnold Beckmann, Ulrich Berger, Benedikt Löwe, and John V. Tucker, editors, *CiE*, volume 3988 of *Lecture Notes in Computer Science*, pages 186–197. Springer, 2006.

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

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

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

Jean-Yves Girard, Paul Taylor, and Yves Lafont. *Proofs and types.* CUP, Cambridge, 1989.

Paul Hoogendijk and Oege De Moor. Container types categorically. *J. Funct. Program.*, 10:191–225, March 2000.

Saunders Mac Lane. *Categories for the Working Mathematician.* Springer, 1998.

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

Michele Pagani and Christine Tasson. The inverse Taylor expansion problem in Linear Logic. In Andrew M. Pitts, editor, *Proceedings of the 24th Annual IEEE Symposium on Logic in Computer Science, LICS 2009*, pages 222–231. IEEE Computer Society, 2009.

Christine Tasson. Algebraic totality, towards completeness. In Curien (Cur09), pages 325–340.

Marie-France Thibault. Pre-recursive categories. *Journal of Pure and Applied Algebra*, 24:79–93, 1982.

Paolo Tranquilli. Intuitionistic differential nets and lambda-calculus. To appear in Theor. Comput. Sci, 2008.

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

Lionel Vaux. Differential linear logic and polarization. In Curien (Cur09), pages 371–385.

Lionel Vaux. A non-uniform finitary relational semantics of system T. In Ralph Matthes and Tarmo Uustalu, editors, *Proceedings of the 6th Workshop on Fixed Points in Computer Science.* Institute of Cybernetics at Tallinn University of Technology, 2009.