



HAL
open science

Intersection Types and Resource Calculi in the Denotational Semantics of Lambda-Calculus

Federico Olimpieri

► **To cite this version:**

Federico Olimpieri. Intersection Types and Resource Calculi in the Denotational Semantics of Lambda-Calculus. Mathematics [math]. Aix-Marseille Universite, 2020. English. NNT : . tel-03123485

HAL Id: tel-03123485

<https://theses.hal.science/tel-03123485v1>

Submitted on 27 Jan 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Aix-Marseille Université
Università degli Studi Roma Tre
École Doctorale en Mathématiques et Informatique

Dipartimento di Matematica e Fisica

Institut de Mathématiques de Marseille

Thèse présentée pour obtenir le grade universitaire de docteur

Discipline : Mathématiques

Federico OLIMPIERI

Intersection Types and Resource Calculi in the Denotational
Semantics of λ -calculus

Dirigée par Laurent Regnier, Lorenzo Tortora de Falco et Lionel Vaux.

Rapportée par Marcelo Fiore et Paul-André Melliès.

Soutenue le 27/11/2020 devant le jury composé de :

Marcelo FIORE	University of Cambridge	Rapporteur
Ugo DAL LAGO	Università di Bologna	Examineur
Giuseppe ROSOLINI	Università di Genova	Examineur
Christine TASSON	Université Paris 6	Examineur
Laurent REGNIER	Aix-Marseille Université	Directeur de thèse
Lionel VAUX	Aix-Marseille Université	Co-directeur de thèse
Lorenzo TORTORA DE FALCO	Università degli Studi Roma Tre	Directeur de thèse

Numéro national de thèse/suffixe local : 2020AIXM0380/018ED184



Cette œuvre est mise à disposition selon les termes de la [Licence Creative Commons Attribution - Pas d'Utilisation Commerciale - Pas de Modification 4.0 International](https://creativecommons.org/licenses/by-nc-nd/4.0/).

Résumé

Cette thèse étudie la notion d'approximation dans le lambda-calcul selon différentes perspectives.

Ehrhard et Regnier ont introduit le développement de Taylor des lambda-termes : on peut voir un lambda-terme comme une série infinie des ses approximations linéaires. Une autre notion d'approximation dans le lambda-calcul est donné par les types intersections, introduits par Coppo et Dezani dans les années 80.

Dans une première partie, nous étendons la définition standard du développement de Taylor à un lambda-calcul non-déterministe. On introduit un calcul avec ressources rigide et on établit une relation combinatoire entre les termes de ce calcul et les éléments du développement. On démontre un théorème de commutation entre développement de Taylor et arbres de Böhm dans ce contexte non-déterministe.

Dans une deuxième partie de la thèse, on introduit le cadre bicatégorique des distributeurs. On présente une collection de 2-monades, les monades de ressources, et on les transpose dans la bicatégorie des distributeurs, en utilisant une méthode introduite par Fiore, Gambino, Hyland et Winskel. On considère les bicatégories de Kleisli pour ces pseudomonades et on donne une condition suffisante pour qu'une telle bicatégorie soit cartésienne fermée, donc un modèle du lambda-calcul simplement typé.

Dans une troisième et dernière partie, on introduit les distributeurs de types intersections et, inspiré par le travail de Tsukada, Asada et Ong, le développement rigide des lambda-termes. Ces deux notions d'approximation sont une présentation syntaxique de la sémantique bicatégorique induite par les bicatégories de Kleisli étudiées dans la deuxième partie. La notion de distributeur de types intersections nous permet de considérer des systèmes de types intersections avec sous-typage. Ces modèles donnent une sémantique dénotationnelle sensible aux preuves, au sens où la sémantique d'un terme lui associe l'ensemble des ses dérivations de typage dans ces systèmes. Le sous-typage est induit par la structure particulière d'une catégorie de types. Notre construction est paramétrique sur les monades de ressources et produit quatre systèmes de types intersections. On montre que les distributeurs de types intersections sont naturellement isomorphes au développement rigide. On étudie ces structures sous réduction.

Mots clés : sémantique dénotationnelle, lambda-calcul, logique linéaire, développement de Taylor, types intersections, distributeurs.

Riassunto

Questa tesi studia il concetto di approssimazione nel lambda-calcolo da diverse prospettive.

Ehrhard e Regnier hanno introdotto lo sviluppo di Taylor dei lambda-termini: si può vedere un lambda-termini come somma formale infinita delle sue approssimazioni lineari. Un'altra nozione di approssimazione nel lambda-calcolo è data dai tipi intersezione, introdotti da Coppo e Dezani negli anni Ottanta.

Nella prima parte della tesi, estendiamo la definizione standard dello sviluppo di Taylor a un lambda-calcolo non deterministico. Introduciamo un calcolo con risorse rigido e stabiliamo una relazione combinatoria tra termini con risorse rigidi e gli elementi dello sviluppo di Taylor. Proviamo un teorema di commutazione tra lo sviluppo di Taylor e gli alberi di Böhm in questo contesto non deterministico.

Nella seconda parte della tesi viene introdotto il quadro bicategorico dei distributori. Presentiamo una raccolta di dottrine, le monadi con risorse, e le estendiamo alla bicategoria dei distributori, utilizzando un metodo introdotto da Fiore, Gambino, Hyland e Winskel. Consideriamo le bicategorie di Kleisli di queste pseudomonadi e diamo una condizione sufficiente che rende queste bicategorie cartesiane chiuse, quindi modelli del lambda-calcolo semplicemente tipato.

Nella terza e ultima parte, introduciamo i distributori di tipi intersezione e, ispirandoci al lavoro di Tsukada, Asada e Ong, lo sviluppo rigido dei lambda-termini. Queste due nozioni di approssimazione sono una presentazione sintattica della semantica bicategorica indotta dalle costruzioni di Kleisli studiate nella seconda parte. I distributori di tipi intersezione determinano dei sistemi di tipi intersezione con sottotipaggio. Questo modello fornisce una semantica denotazionale sensibile alle derivazioni, nel senso che la semantica di un termine associa ad esso l'insieme delle sue derivazioni di tipaggio in sistemi di tipi appropriati. Il sottotipaggio è indotto dalla particolare struttura di una categoria di tipi. La nostra costruzione è parametrica sulle monadi con risorse e produce quattro sistemi di tipi intersezione. Mostriamo che i distributori di tipi intersezione sono naturalmente isomorfi allo sviluppo rigido. Studiamo queste strutture sotto riduzione.

Parole chiave: semantica denotazionale, lambda calcolo, logica lineare, sviluppo di Taylor, tipi intersezione, distributori.

Abstract

This thesis studies the notion of approximation in lambda-calculus from different perspectives.

Ehrhard and Regnier introduced the Taylor expansion of lambda-terms: one can see a lambda-term as an infinite formal sum of its linear approximants. Another notion of approximation in lambda-calculus is given by intersection types, introduced by Coppo and Dezani in the 80s.

In a first part, we extend the standard definition of Taylor expansion to a non-deterministic lambda-calculus. We introduce a rigid resource calculus and we establish a combinatorial relation between rigid resource terms and the elements of the Taylor expansion. We prove a commutation theorem between Taylor expansion and Böhm trees in this non-deterministic context.

In a second part of the thesis, the bicategorical framework of distributors is introduced. We present a collection of doctrines, the resource monads, and we lift them to the bicategory of distributors, using a method introduced by Fiore, Gambino, Hyland and Winskel. We consider the Kleisli bicategories of these pseudomonads and we give a sufficient condition that makes these bicategories cartesian closed, thus models of simply typed lambda-calculus.

In a third and last part, we introduce intersection type distributors and, inspired by the work of Tsukada, Asada and Ong, the rigid expansion of lambda-terms. These two notions of approximation are a syntactic presentation of the bicategorical semantics induced by the Kleisli constructions studied in the second part. Intersection type distributors determine intersection type systems with subtyping. This model gives a proof-relevant denotational semantics, in the sense that the semantics of a term associates to it the set of its typing derivations in appropriate systems. Subtyping is induced from the particular structure of a category of types. Our construction is parametric on resource monads and produces four intersection type systems. We show that intersection type distributors are naturally isomorphic to the rigid expansion. We study these structures under reduction.

Keywords: denotational semantics, lambda-calculus, linear logic, Taylor expansion, intersection types, distributors.

Remerciements

Je commence par remercier Lionel pour avoir d'abord encadré mon stage de recherche et après avoir accepté de diriger ma thèse. Ce travail dépend énormément des nos fréquent discussions et de ses observations, toujours éclairantes.

Je remercie aussi Laurent et Lorenzo, mes deux autres directeurs, toujours très disponibles et prêts à donner des conseils valables. Lorenzo, en particulier, c'est le principal responsable de ma migration à Marseille. Le double diplôme entre Rome et Marseille, qu'il gère du côté romain, c'est une opportunité remarquable et c'était une des conditions de possibilité principales de mon travail de recherche.

Merci à Marcelo Fiore et Paul-André Melliès pour avoir accepté d'être les rapporteurs de cette thèse. Merci également à Ugo dal Lago, Christine Tasson et Giuseppe Rosolini pour leur participation au jury de ma soutenance.

Merci à toute l'équipe LDP, j'ai toujours profité des échanges avec vous et grâce aux Groupes de Travail j'ai appris beaucoup de jolies mathématiques.

Merci aux membres de l'IRN Linear Logic et des Groupes de Travail LHC (Logique, Homotopie, Catégories) et Scalp (Structures formelles pour le CALCUL et les Preuves) pour avoir créé une ambiance favorable à la recherche et au partage des idées. En particulier je remercie Giulio Guerrieri, Zeinab Galal, Thomas Ehrhard, Damiano Mazza, Luc Pellissier, Flavien Breuvert, Marie Kerjean, Kenji Maillard, Giulio Manzonetto, Stefano Guerrini.

Merci à tou(te)s les copain(e)s que j'ai rencontré pendant mon séjour en France et avec lequel(le)s j'ai pas mal rigolé: Eugène, Davide B., Davide C., Andrea, Guillaume, Elena, Matteo A., Jean-Michel, Mamah, Bastien, Leonardo C., Jules, Alejandro, Alberto, Diogo, Marta, Santiago, Claudio, Marianna, Davo, Sofia, Serena, Mélodie, Nicolas, Axel, Gaia, Letizia, Jacopo, Paolo P., Paolo M., Riccardo, Léa, Pavlo, Lorenzo, Luigi, Edoardo, Francesca, Giulia, Jean-Baptiste, Rasa, Zac, tou(te)s les participants du Café de langues, les camarades du 9-10 et tou(te)s les autres que je n'ai pas mentionné mais avec lequel(le)s j'ai fait des choses sympathiques.

Merci à mes copains italiens de longue date: les romains, Matteo C., Vittorio, Valerio, Leonardo L., la gente non alla moda della Tana; les plus au moins "étrusques", Francesco, Felice, Jacopo, Giacomo, Guido, Emanuele. En particulier, merci aux copain(e)s du PB, un vrai refuge dans ces temps bizarres.

Merci à Marseille, une ville avec des énormes problèmes mais aussi avec grande beauté humaine et naturelle. C'est difficile de partir.

Je termine en remerciant ma famille et, en particulier, mes parents, Francesca et Angelo, pour leur soutien inconditionnel. Cette thèse est dédiée à eux.

Contents

Contents	7
List of Figures	10
Introduction	17
1 Taylor Expansion for λ-Terms with Choice Operator	18
1.1 Introduction	18
1.2 Some Basic Facts on Groups and Group Actions	21
1.3 A Generic Non-Deterministic λ -Calculus	22
1.3.1 λ_{\oplus} -Terms	22
1.3.2 Böhm Trees	24
1.4 Taylor Expansion in a Uniform Non-Deterministic Setting	25
1.4.1 Resource terms	25
1.4.2 Taylor Expansion of λ_{\oplus} -Terms	27
1.4.3 Multiplicity Coefficients	29
1.4.4 Taylor Expansion of Böhm Trees	30
1.5 The Groupoid of Permutations of Rigid Resource Terms	32
1.5.1 Rigid Resource Terms and Permutation Terms	32
1.5.2 Rigid Substitution	34
1.5.3 Substitution for Permutation Expressions	36
1.5.4 The Combinatorics of Permutation Expressions under Coherent Substitution	40
1.6 Normalizing the Taylor Expansion	43
1.6.1 Normalizing Resource Expressions in a Uniform Setting	43
1.6.2 Commutation	45
2 Categorical Interlude	47
2.1 Monoidal Categories	48
2.1.1 Unbiased Monoidal Categories	52
2.2 Bicategories	54
2.2.1 Cartesian Closed Structure	59
2.3 Two-Dimensional Monad Theory	60
2.3.1 Pseudomonads and Lax Algebras	61
2.3.2 Relative Pseudomonads	63
2.3.3 (Pseudo) Algebras Lifting of Relative Pseudomonads	65

2.4	Kan Extensions and Coends	68
2.4.1	Coends	68
2.4.2	Kan Extensions	71
2.5	The Category of Presheaves	72
2.5.1	Free Cocompletion of (Product) Categories	72
2.5.2	Day Convolution	74
2.6	Monads and Resources	77
2.6.1	Boom Hierarchy of Data Types	78
2.6.2	Integers and Lists	79
2.6.3	Resource Monads	82
2.7	Relations, Preorders, Distributors	85
2.7.1	The Category of Sets and Relations	85
2.7.2	The Category of Preorders and Monotonic Relations	86
2.7.3	The Bicategory of Distributors	88
2.8	The Bicategories $\mathcal{S}\text{-Dist}$ and $\mathcal{S}\text{-CatSym}$	89
2.8.1	The Bicategory $\mathcal{S}\text{-Dist}$	90
2.8.2	The Bicategory $\mathcal{S}\text{-CatSym}$	91
3	Intersection Type Distributors	95
3.1	Introduction	95
3.2	Models for the Simply Typed λ -calculus	97
3.3	Models for pure λ -calculus	101
3.4	Intersection Types as Distributors	103
3.4.1	The Denotation is Isomorphic to the Semantics	106
3.4.2	The Denotation as an Intersection Type System	107
3.4.3	Type Derivations under Reduction	118
3.5	Normalization Theorems	121
3.6	Some Worked Out Examples	127
3.6.1	Example 1: Linear Resources	127
3.6.2	Example 2: Cartesian Resources	132
4	Subtyping-Aware Polyadic Calculus and Rigid Expansion of Ordinary λ-Terms	134
4.1	Introduction	134
4.2	The Calculus of Subtyping-Aware Polyadic Terms	137
4.2.1	Actions on Polyadic Type Derivations	142
4.2.2	Congruence on Polyadic Type Derivations	147
4.2.3	Substitution Operation on Type Derivations	149
4.2.4	Points of Type Derivations	154
4.2.5	Congruence on Points	160
4.3	Rigid Expansion	161
4.3.1	Reduction and Congruence	164
4.4	An Example: the Linear Case	167

4.4.1	Points of Linear Type Derivations	169
Conclusion		173
Bibliography		174
Appendix		182
A	Technical Proofs of Chapter 3	182
B	Technical Proofs of Chapter 4	187
B.1	Actions and Congruence under Substitution of Type Derivations	187
B.2	Canonical Forms of Type Derivations	192
B.3	Actions and Congruence under Reduction of Points	198
B.4	Linear Case	205

List of Figures

1.1	Reduction rules of the λ_{\oplus} -calculus	23
1.2	The approximation order on Λ_{\perp}	24
1.3	Reduction rules of the resource calculus with sums	26
1.4	Rules for the coherence relation on $\Delta_{\oplus}^{(!)}$	29
1.5	Rules for the rigid representation relation	32
1.6	Action of permutation expressions on rigid expressions	33
1.7	Graphical representation of $\sigma \cdot (\tau_1, \dots, \tau_n)$	38
1.8	Rules for the coherence relation on $D^{(!)}$	40
1.9	Reduction rules of the rigid resource calculus	43
3.1	Simply typed λ -calculus	98
3.2	Denotation of simply typed λ -terms.	99
3.3	Category of Types D_A and parametric Intersection Type System E_A^S	102
3.4	Denotation of λ -terms.	103
3.5	Left action on derivations.	107
3.6	Right action on derivations.	110
3.7	Intersection type systems R_A and C_A	130
4.1	Right action on polyadic type derivations.	143
4.2	Left action on polyadic type derivations.	144
4.3	Left action on points.	157
4.4	Polyadic approximation of λ -terms.	161
4.5	Approximation of β -reduction steps.	165
4.6	Polyadic approximation of resource terms.	170
.1	Canonical forms.	193

Introduction

Curry-Howard-Lambek Correspondence Denotational semantics was introduced in the 70s, mostly by the work of Strachey and Scott [Sco70; SS71], and has since then represented an elegant tool to obtain a mathematical understanding of programs. Roughly speaking, a denotational semantics for a class of programs consists of finding appropriate invariants for computations in some kind of mathematical structure. This approach to the semantics of programming languages was fruitfully recast into the *Curry-Howard-Lambek correspondence* [LS86]. The slogans

Formulae as Types as Objects

Proofs as Programs as Morphisms

in their simplicity and elegance offer a direct way to give a mathematical interpretation to programs as morphisms in an appropriate category. Minimal implicative logic, simply typed λ -calculus and cartesian closed categories are the three tiers of the seminal version of this correspondence. Scott's work fits completely in this framework: the first denotational semantics of λ -calculus was given in the category of domains, a special kind of lattices, and continuous functions.

Linear Logic A direct descendent of this approach to programming language semantics is *linear logic*. The introduction of *coherence spaces* led Girard [Gir87] to the discovery that the intuitionistic arrow $A \Rightarrow B$ can be decomposed into two more fundamental connectives:

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

From a Curry-Howard-Lambek perspective the former formula determines the decomposition of a model of computation, *i.e.* a cartesian closed category, into more essential structures¹.

It turns out that the new implication connective, $A \multimap B$, called *linear implication*, can be interpreted in a resource-sensitive way: a program typed with $A \multimap B$ is a program which uses its input exactly once during computation. The modality $!$ then breaks this kind of linearity: $!A \multimap B$ means that we are allowed to use the input as much as we want, allowing duplication and erasing of resources. Linear logic is then an appropriate framework for the formalization of computational resource

1. We refer to [Mel09] for what concerns the categorical semantics of linear logic.

structures. In particular, this *quantitative* aspect of linear logic is correlated to a *quantitative approach* to denotational semantics.

Quantitative Semantics The field of quantitative semantics, in the sense originally introduced by Girard [Gir88], is rooted on the basic idea to interpret λ -terms as generalized power series, hence associated with analytic maps — instead of continuous maps, *à la* Scott. It was later revisited, e.g. by Lamarche [Lam92] and Hasegawa [Has02], to provide a denotational interpretation of linear logic proofs as matrices. More recently Ehrhard [Ehr05] introduced models of linear logic based on a particular class of topological vector spaces, and thus accommodating *differentiation*.

In that setting, the analytic maps associated with λ -terms are also smooth maps, *i.e.* they are infinitely differentiable. This led to the *differential extensions* of λ -calculus [ER03] and linear logic [ER06b] by Ehrhard and Regnier. The keystone of this line of work is an analogue of the Taylor expansion formula, which allows to translate terms (or proofs) into infinite linear combinations of finite approximants [ER08]: in the case of λ -calculus, those approximants are the terms of a resource calculus, in which the copies of arguments of a function must be provided explicitly, and then consumed linearly, instead of duplicated or discarded during reduction.

Indeed, by contrast with denotational semantics, resource approximants retain a dynamics, albeit very simple and finitary: the size of terms is strictly decreasing under reduction. The seminal result relating the reduction of λ -terms with that of their approximants is the commutation between Taylor expansion and normalization: Ehrhard and Regnier have shown that the Taylor expansion M^* of a λ -term M can always be normalized, and that its normal form is nothing but the Taylor expansion of the Böhm tree $BT(M)$ of M [ER08; ER06a]. In particular, the *normal form* of Taylor expansion defines a proper denotational semantics.

Denotation via Types A few years before Girard’s introduction of linear logic, Coppo and Dezani [CD78] proposed *intersection types*, a type-theoretic framework sensitive to the fact that a λ -term can be typed in several ways. In order to define an intersection type system, they add another constructor to the syntax: $a \cap b$. The basic intuition for *intersection types* derives from realizability semantics: programs that realize $a \cap b$ are exactly the programs that realizes both a and b , *i.e.* $\llbracket a \cap b \rrbracket = \llbracket a \rrbracket \cap \llbracket b \rrbracket$. Typability with an intersection type is then equivalent to being typable with *both* types a and b .

This kind of type disciplines are very useful to characterize fundamental notions of normalization in λ -calculus (*e.g.*, head-normalization, β -normalization, strong normalization) [Kri93; BL13; BKV17]. Moreover, if we consider a *non-idempotent* version of the intersection operator [Gar94; Car07], *i.e.*, $a \cap a \neq a$, the considered type system is *resource sensitive*. In that case, the arrow type

$$a_1 \cap \cdots \cap a_k \Rightarrow a$$

encodes the *exact* number of times that the program needs its input during computation. The resource awareness of non-idempotent intersection has been used to prove normalization and standardization results by *combinatorial* means [BKV17] and to express the execution time of programs and proof-nets [Car07; CPF11; CF16]. Another important feature of intersection type systems is that they determine a class of *filter models* for pure λ -calculus [Cop+84]. The correspondence between intersection types and Engeler-like models is also well-known [Hyl+06]. Hence intersection types are both *syntactic* and *semantic* objects.

A Categorical Approach to Intersection Types The semantic side of intersection types is connected also to linear logic-inspired categorical semantics. A simple and informative categorical model for λ -calculus is the *relational model* (MRel), that is roughly what is left of coherence spaces when we forget about coherence². Objects of MRel are sets, while morphisms are *multirelations* $f \subseteq M_f(A) \times B$, where $M_f(A)$ is the free commutative monoid over A . This model arises from the linear logic decomposition. The relational model can be considered as the most canonical example of quantitative semantics. Indeed, it is well-known that the semantics induced by the category MRel corresponds to the *non-idempotent* intersection type system \mathcal{R} [Car07]. This correspondence says that the abstract categorical interpretation of a λ -term can be presented in a concrete way, as a form of type assignment. The intersection type constructor \cap corresponds to the product in the free commutative monoid over types. This fact suggests the possibility to model, in all generality, the intersection type construction *via monads*. With some relevant modifications, one can also achieve in this way an idempotent intersection [Ehr12b; Ehr16]. Moreover, the non-idempotent intersection type system \mathcal{R} is also tightly connected to the Taylor expansion of λ -terms [ER08; Car07]. In particular, the interpretation of a term M in the relational model can be recovered straightforwardly from the normal form of its expansion [Car07].

Lifting to the Second Dimension Intersection types and λ -terms approximants are also related to more complex categorical structures. Melliès and Zeilberger [MZ15] gave a *categorification* of type systems: a type system is a functor between a *category of type derivations* and a *category of terms*:

$$U : D \rightarrow \Lambda$$

where U can be seen as a sort of forgetful functor³. Building on that work and on the 2-categorical and operadic approaches to λ -calculus studied respectively in [See87; Hir13] and [Hyl17], Mazza, Pellissier and Vial [MPV18] presented a novel perspective on intersection types and linear approximation, rooted in the framework of *multicategories* and relational (discrete) *distributors*. Intersection type

2. For a general survey about the main results on the relational model we refer to [Ong17].

3. In all generality, *every* functor can be interpreted in this type-theoretic way.

systems are seen as as special kind of *fibrations*, that gives rise, *via* a Grothendieck construction, to an *approximation presheaf* which depends on the correspondence between intersection types and a special kind of approximants, the *polyadic terms* [Maz17]. This construction can be seen as a "syntactic categorification" of intersection types. Indeed, while the construction of Mazza *et al.* is an elegant and very general approach to intersection type disciplines, that also allows to prove normalization theorems in a modular way, it does not provide a *type-theoretic denotational semantics*⁴.

Contributions The main conceptual objective of the thesis is to sublimate the opposition between a *static* denotational semantics and a *dynamic* operational one. Bicategories are a natural appropriate setting for this: the interpretation of λ -terms is preserved under reduction only up to isomorphism. More precisely, if $M \rightarrow_{\beta} N$, then

$$\beta : \llbracket M \rrbracket \cong \llbracket N \rrbracket$$

hence, the strictness of invariance imposed by standard denotational semantics is gone. Moreover, Tsukada, Asada and Ong [TAO17] showed how the former isomorphism can be interpreted *via a reduction relation* over a special kind of approximants of λ -terms. Hence, the isomorphism is given by an appropriate *operational semantics* for a resource calculus.

We consider this to be a major technical and theoretical improvement, and we take it as the starting point of our semantic investigations. We advocate a novel approach to denotational semantics, which builds on [Fio+08; Fio+17; GJ17; TAO17; MZ18]. The denotational semantics we are looking for has to satisfy the following informal conditions.

1. We want to be able to interpret simply typed and untyped λ -calculi, in all their generality. In particular, we want that the following diagram "commutes" up to "invertible 2-cell"

$$\begin{array}{ccc} ST(\Lambda) & \xrightarrow{U} & \Lambda \\ & \searrow & \downarrow \llbracket - \rrbracket_{\Lambda} \\ & \llbracket - \rrbracket_{St} & \mathcal{C} \end{array}$$

where $ST(\Lambda)$, Λ denote respectively simply typed λ -calculus and the untyped one and U is a "forgetful functor". \mathcal{C} is our chosen bicategory and $\llbracket - \rrbracket_{St}$, $\llbracket - \rrbracket_{\Lambda}$ are respectively our typed and untyped interpretation. We believe that the former diagram can be formalized in the 2-operadic framework, following [MZ15; MPV18; Maz17], giving rise to an appropriate *interpretation pseud-*

4. Given $M \rightarrow_{\beta} N$, the value of the approximation presheaf on M is not in general isomorphic to its value on N [Maz17][pp. 65-66]. Hence, the approximation presheaf is not a pseudofunctor but only a lax morphism.

ofunctor⁵. However, we leave this formalization to future work. We then stick to a classic notion of interpretation of λ -calculi [AC98], adding to it the 2-dimensional ingredient: an interpretation of β -reduction steps.

2. We want to lift the correspondence

Intersection Types - Relational Semantics - (NF of) Taylor Expansion

to our considered bicategorical framework. The natural choice is then the bicategory of *distributors* [Bén00]. Distributors can be seen as a *categorification* of relations between sets. A relation $f \subseteq A \times B$ is equivalent to its characteristic function

$$\chi_f : A \times B \rightarrow \{0, 1\}.$$

In particular, the former function naturally induces a functors form $A \times B$, taken as discrete category, to the 2 elements category. It is then natural to relax the hypothesis and consider functors of the shape

$$F : B^o \times A \rightarrow \text{Set}$$

where A and B are arbitrary small categories. These functors are called distributors⁶. We want to obtain a corresponding generalization for what concerns intersection types and Taylor expansion. In particular, our construction will be *proof-relevant*, in a sense that will be clarified in Chapter 3.

Distributors-induced semantics of λ -calculus has already an history of its own. Cattani and Winskel [CW05] proposed a bicategorical model of linear logic in the category of distributors, generalizing Scott's domains. In a subsequent paper, Fiore, Gambino, Hyland and Winskel [Fio+08] introduced the bicategory of *generalized species of structures* (Esp), a rich framework encompassing both multirelations and Joyal's combinatorial species [Joy86]. They also proved that Esp is cartesian closed and, hence, a bicategorical model for λ -calculus. It is in this setting that Tsukada, Asada and Ong introduced a *rigid Taylor expansion* for a non deterministic simply typed λ -calculus with fixpoints. The rigid Taylor expansion consists roughly of a distributor which take as input a type context Δ and a type a , that lives in an appropriate *groupoid of types*, and returns a set

$$\mathcal{T}_{\text{rig}}(M)(\Delta, a) = \{\tilde{p} \mid p \text{ is an approximant of } M \text{ s.t. } \Delta \vdash p : a\}.$$

Approximants are a special kind of Mazza's *polyadic terms* [Maz12]. The equivalence relation is induced by the structure of the model and it is central in establishing the denotational semantics⁷. Hence, this construction overcomes the semantic

5. In the context of [Maz17] this would imply that the approximation presheaf is a pseudo-functor.

6. Another popular name for this kind of structures is *profunctor*.

7. Without it there would not be an isomorphism $\mathcal{T}_{\text{rig}}(M) \cong \mathcal{T}_{\text{rig}}(N)$ whenever $M \rightarrow_{\beta} N$.

”limitation” of the standard Taylor expansion, which, as we saw, does not directly induce a denotational semantics. The rigid Taylor expansion is even isomorphic to the categorical semantics induced by the generalized species of structures. Hence, it can be seen as the *syntactic presentation* of this semantics. Moreover, the cardinality of $\mathcal{T}_{\text{rig}}(M)(\Delta, a)$, under some specific conditions, is equal to the number of reduction paths from M to its normal form. Thus, we obtain a new quantitative information that was not directly deducible from the standard relational semantics.

However, their work is restricted to the η -long fragment of their calculus and, as we shall see, it does not directly lift to the full simply typed λ -calculus and to the untyped one (Chapter 4). Hence, our main goal will be to find a suitable workable generalization of their approach, which also encompasses intersection types.

Outline of the Thesis Each chapter begins with a small introduction, which will give the technical context of its particular content, together with a concise discussion of related work.

- In Chapter 1, we extend the definition of standard Taylor expansion to a non-deterministic setting, preserving the uniformity condition on resource terms [ER08]. Moreover, as a first attempt towards a dynamic denotational semantics, we introduce a *rigid resource calculus*. We show that the *groupoid structure* of these rigid terms determines a fine-grained combinatorial characterization of the quantitative information given by coefficients in the Taylor expansion. We also generalize the commutation theorem between Taylor expansion and Böhm trees to this non-deterministic setting.
- In Chapter 2 we introduce some categorical background that we shall need in the rest of the thesis. The main objective of the chapter is to define a family of Kleisli bicategories of distributors, associated with the lifting of suitable collection of doctrines, the *resource monads*. In order to do so, we detail the method introduced in [Fio+17], that allows to lift 2-monads over categories to pseudomonads over distributors, *via* the introduction of *relative pseudomonads*.
- In Chapter 3 we define *intersection type distributors*, which will constitute the main ingredient of our denotational semantics. We prove that intersection type distributors give a syntactic presentation, up to isomorphism, of the λ -calculus bicategorical semantics induced by the family of bicategories built in Chapter 2. We give a refined statement of subject reduction and expansion for intersection type distributors, making explicit the natural isomorphism

$$\beta : \llbracket M \rrbracket \cong \llbracket N \rrbracket$$

associated with a β -reduction step $M \rightarrow_{\beta} N$. We prove normalization theorems for intersection type distributors, adapting classic realizability arguments [Kri93]. We conclude the chapter by studying two concrete examples of our construction.

- In Chapter 4 we consider the question of giving a denotational semantics of approximants. In order to do so, we define a *subtyping-aware* extension of polyadic calculus. We show that the dynamical information of a type derivation in our system is encoded not only in its associated polyadic term but also in a morphism. For this reason, we shall introduce the *points of type derivations*, that consist in couples of terms and morphisms. We then define the *rigid expansion* of ordinary λ -terms and we study this structure under reduction, proving that it is isomorphic to intersection type distributors.

Notations

Most of the notations that we shall use are introduced in the thesis, we recall here some general notational choices for well-known structures.

- We use \mathbb{N} for the set of natural numbers and, given $n \in \mathbb{N}$, we denote as \mathfrak{S}_n the symmetric group over n .
- Given a binary relation on a set $\rightarrow \subseteq X^2$ we write $\rightarrow^=$ for its reflexive closure.
- Given a category A , we denote as A^o its opposite category. Given (bi) categories A_1, \dots, A_n we denote as $\prod_{i=1}^n A_i$ their product. Given categories A_1, \dots, A_n we denote as either $\sqcup_{i=1}^n A_i$ or $\sum_{i=1}^n A_i$ their coproduct. Given categories A, B , we use either $[A, B]$ or $\text{Cat}(A, B)$ to denote functor categories. We denote the initial category as \emptyset and the terminal one as $\mathbf{1}$. We use linear logic notations for the general notions of product, linear exponential, etc.

1. Taylor Expansion for λ -Terms with Choice Operator

1.1. Introduction

This chapter integrates, almost *verbatim*, the paper [OA20]. For the scope of this thesis, the content of the chapter can be seen as a first attempt in defining an appropriate resource calculus that could eventually determine a syntactic presentation of a distributors-induced (bicategorical) denotational semantics in an untyped λ -calculus (non-deterministic) setting. We define a *rigid resource calculus*, where instead of multisets of terms we have *lists*. This refinement determines a natural groupoid structure for our term language, where morphisms are given by inductive permutations. However, while our approach leads to finer understanding of the combinatorics of Ehrhard's and Regnier's Taylor expansion [ER08] and to an uniform approach to non-determinism, the rigid resource calculus presented here is not suitable for the semantic constructions of Chapter 3 and 4. As we shall see, we need to make an ulterior step in rigidification and consider *polyadic terms* [Maz12; MPV18].

Contributions Ehrhard and Regnier's proof of the identity $BT(M)^* = \text{nf}(M^*)$ can be summed up as follows:

- Step 1: The non-zero coefficients of resource terms in M^* do not depend on M . More precisely, we can write $M^* = \sum_{s \in \mathcal{T}(M)} \frac{1}{m(s)} s$, where $\mathcal{T}(M)$ is the support set of Taylor expansion and $m(s)$ is an integer coefficient depending only on the resource term s .
- Step 2: The set $\mathcal{T}(M)$ is a clique for the coherence relation obtained by setting $s \subset s'$ iff s and s' differ only by the multiplicity of arguments in applications.
- Step 3: If s is uniform, *i.e.* $s \subset s$, and t is in the support of $NF(s)$ (the normal form of s , which is a finite sum of resource terms) then $m(t)$ divides $m(s)$ and the coefficient of t in $NF(s)$ is $\frac{m(s)}{m(t)}$.
- Step 4: The respective supports of $NF(s)$ and $NF(s')$ are disjoint cliques whenever $s \subset s'$ and $s \neq s'$. Then one can set $NF(M^*) = \sum_{s \in \mathcal{T}(M)} \frac{1}{m(s)} NF(s)$, the summands being pairwise disjoint.
- Step 5: By Step 1, $BT(M)^* = \sum_{t \in \mathcal{T}(BT(M))} \frac{1}{m(t)} t$. To deduce the identity $BT(M)^* = \text{nf}(M^*)$ from the previous results, it is then sufficient to prove that $t \in$

1. Taylor Expansion for λ -Terms with Choice Operator – 1.1. Introduction

$\mathcal{T}(BT(M))$ iff there exists $s \in \mathcal{T}(M)$ such that t is in the support of $NF(s)$.

The first two steps are easy consequences of the definitions. For Step 4, it is sufficient to observe that elementary resource reduction steps preserve coherence. Step 3 relies on a careful investigation of the combinatorics of substitution in the resource calculus: this involves an elaborate argument about the structure of particular subgroups of the group of permutations of variable occurrences [ER08, Section 4]. Finally, Ehrhard and Regnier establish Step 5 by relating Taylor expansion with execution in an abstract machine [ER06a].

In the present work, we propose to revisit this seminal result, along three directions.

- (i) We largely simplify Step 5, relying on a technique introduced in [Vau17]. We consider the hereditary head reduction strategy (a slight variant of leftmost reduction, underlying the construction of Böhm trees) and show that it can be simulated directly in the resource calculus, through Taylor expansion. We thus avoid the intricacies of an abstract machine with resource state.
- (ii) We extend all the results to a model of non-determinism, introduced as a formal binary choice operator in the calculus. By contrast with previous proposals from Ehrhard [Ehr10], or Pagani, Tasson and Vaux Auclair [PTV16; Vau17], we show that uniformity can still be relied upon, provided one keeps track of choices in the resource calculus: the coherence associated with non-deterministic choice is then that of the *with* connective ($\&$) of linear logic.
- (iii) We analyse coefficients in the Taylor expansion by introducing a groupoid of permutation terms acting on a rigid variant of resource terms, where multisets of arguments are replaced with lists. This is more in accordance with the intuition that $m(s)$ is the number of permutations of arguments that leave s (or rather, any rigid representation of s) invariant: Ehrhard and Regnier rather worked on permutations of variable occurrences, which allowed them to consider groups rather than a groupoid.

Those three contributions are completely independent from each other.

Discussion of Related Work Our contribution (i) establishes that, although it is interesting in itself, Ehrhard and Regnier’s study of the relationship between elements in the Taylor expansion of a term and its execution in an abstract machine is essentially superfluous for proving the commutation theorem.

Barbarossa and Manzonetto have independently proposed another technique which amounts to show that any reduction from an element of $\mathcal{T}(M)$ can be completed into a sequence of reductions simulating a β -reduction step [BM20, Section 4.1]. The strength of our own proposal is that it is a commutation result itself: hereditary head reduction commutes with Taylor expansion, even taking coefficients into account [Vau17]. Moreover, the Böhm tree of a λ -term is the limit of its hereditary head reducts and, if the convergence of the sum defining the normal form of Taylor expansion is already known, the main commutation theorem

1. Taylor Expansion for λ -Terms with Choice Operator – 1.1. Introduction

ensues directly. This is the path followed in [Vau17] for the algebraic λ -calculus [Vau09], and by Dal Lago and Leventis [LL19] for the probabilistic case.

As stated before, our proposal (ii) to restore uniformity in a non deterministic setting is only valid because the resource calculus keeps a syntactic track of choices. The corresponding constructors are exactly those used by Tsukada, Asada and Ong [TAO17] who were interested in identifying equivalent execution paths of non deterministic programs, but those authors do not mention, nor rely upon any coherence property: this forbids Steps 1 to 4 and, instead, they depend on infinite sums of arbitrary coefficients to be well defined. By contrast, Dal Lago and Leventis have independently proposed nearly the same solution as ours [LL19, Section 2.2], with only a minor technical difference in the case of sums.

The previous two proposals (i) and (ii) may be considered as purely technical improvements of the state of the art in the study of Taylor expansion. What we deem to be the most meaningful contribution of the present work is our study of the groupoid of rigid resource terms. This provides us with a new understanding of the coefficients in the Taylor expansion of a term, in which we can recast the proof of the commutation theorem, especially Step 3: apart from this change of focus, the general architecture of our approach does not depart much from that of Ehrhard and Regnier, but we believe the obtained combinatorial results are closer to the original intuition behind the definition of m . In fact, a notable intermediate result is that the function that maps each permutation term to the permutation it induces on the occurrences of a fixed variable is functorial: one might understand Ehrhard and Regnier’s proof of Step 3 as the image of ours through that functor. Moreover, our study suggests interesting connexions with otherwise independent approaches to denotational semantics based on generalized species of structures [Fio+08; TAO17] and rigid intersection type systems [MPV18].

It is indeed most natural to compare our proposals to the line of work of Tsukada, Asada and Ong [TAO17; TAO18]. On the one hand, Tsukada *et al.* thrive to develop an abstract understanding of reduction paths in a non-deterministic λ -calculus. They are led to consider a polyadic calculus *à la* Mazza [Maz12; MPV18] with syntactic markers for non-deterministic choice, moreover obeying linearity, typing and η -expansion constraints. Then they can define a groupoid of types, whose isomorphisms act on typed terms by permuting variables bound in abstractions and lists of arguments in applications. They show that the obtained groupoid is a bicategorical model of the simply typed $\lambda\mathbf{Y}$ -calculus, the interpretation being given by a variant of Taylor expansion. This interpretation is moreover isomorphic to the one obtained in generalized species of structures [Fio+08].

On the other hand, our results show that Ehrhard and Regnier’s technique can already be adapted to same kind of non determinism without introducing any new concept, and we only introduce the groupoid of rigid resource terms and permutation terms to unveil the combinatorial structure of ordinary resource terms under reduction and normalization. Besides having markers for non-deterministic choice, the only difference between our rigid terms and the ordinary resource terms

is that arguments are linearly ordered. Nevertheless, the action of our permutation terms on rigid terms is very similar to that of the typed isomorphisms considered by Tsukada *et al.*, and this suggests directions for further investigations.

Structure of the Chapter In the very brief Section 1.2, we review some results from group theory that will be useful later.

In Section 1.3 we extend the ordinary untyped λ -calculus with a generic non deterministic choice operator, and present its operational semantics, inspired from that of the algebraic λ -calculus, as well as the corresponding notion of (non extensional) Böhm trees.

Section 1.4 recalls and adapts the definitions of the resource calculus and Taylor expansion. We obtain Step 2 as a straightforward consequence of the definitions and Step 5 by showing that the support of Taylor expansion is compatible with hereditary head reduction. We moreover complete Step 1, making prominent the rôle played by permutations acting on lists of resource terms.

Section 1.5 is the core of the chapter, in which we introduce both the rigid version of resource terms and the permutation terms acting on them, and explore the relationship between the groupoid thus formed and the combinatorics of Taylor expansion. We first show that the coefficient $m(s)$ is nothing but the cardinality of the group of endomorphisms of any rigid version of s . Then we study the structure of permutation terms acting on a substitution, and leverage the obtained results to determine the coefficient of any resource term in the symmetric multilinear substitution associated with a reduction step in the resource calculus.

The final Section 1.6 builds on the study of rigid resource terms and permutation terms to achieve Steps 3 and 4. We conclude the chapter with the commutation theorem.

1.2. Some Basic Facts on Groups and Group Actions

Let \mathcal{G} be a group, X be a set, and write $(g, a) \in \mathcal{G} \times X \mapsto [g]a \in X$ for a left action of \mathcal{G} on X . If $a \in X$, then the *stabilizer* of a under this action is $St(a) := \{g \in \mathcal{G} \mid [g]a = a\}$, which is a subgroup of \mathcal{G} (also called the isotropy group of a); and the *orbit* of a is the set $[\mathcal{G}]a := \{[g]a \mid g \in \mathcal{G}\} \subseteq X$. If $\mathcal{H}, \mathcal{K} \subseteq \mathcal{G}$, we write $\mathcal{H}\mathcal{K} := \{hk \mid h \in \mathcal{H}, k \in \mathcal{K}\}$. If $f : X \rightarrow Y$, $X' \subseteq X$ and $Y' \subseteq Y$ we write $f(X') := \{f(x) \mid x \in X'\}$ and $f^{-1}(Y') := \{x \mid f(x) \in Y'\}$.

Assuming that \mathcal{G} is finite, the following three facts are standard results of group theory.

Fact 1.2.1. *For any $a \in X$,*

$$Card([\mathcal{G}]a) = \frac{Card(\mathcal{G})}{Card(St(a))} .$$

1. Taylor Expansion for λ -Terms with Choice Operator – 1.3. A Generic Non-Deterministic λ -Calculus

Proof. [Lan02, Proposition 5.1]. □

Fact 1.2.2. Let \mathcal{H} and \mathcal{K} be any subgroups of \mathcal{G} . Then

$$\text{Card}(\mathcal{H}\mathcal{K}) = \frac{\text{Card}(\mathcal{H})\text{Card}(\mathcal{K})}{\text{Card}(\mathcal{H} \cap \mathcal{K})} .$$

Proof. [Suz82, §(3.11)]. □

Fact 1.2.3. Let $f : \mathcal{G} \rightarrow \mathcal{H}$ be a group homomorphism and \mathcal{K} be a subgroup of \mathcal{H} . Then

$$\frac{\text{Card}(\mathcal{G})}{\text{Card}(f^{-1}(\mathcal{K}))} = \frac{\text{Card}(f(\mathcal{G}))}{\text{Card}(f(\mathcal{G}) \cap \mathcal{K})} .$$

Proof. Since $f^{-1}(\mathcal{K})$ is a subgroup of \mathcal{G} that contains the kernel of f , we can apply the theorem of correspondence under homomorphisms [Suz82, Theorem 5.5 (1)], which gives: $|\mathcal{G} : f^{-1}(\mathcal{K})| = |f(\mathcal{G}) : f(f^{-1}(\mathcal{K}))|$. We conclude observing that $f(f^{-1}(\mathcal{K})) = f(\mathcal{G}) \cap \mathcal{K}$. □

1.3. A Generic Non-Deterministic λ -Calculus

1.3.1. λ_{\oplus} -Terms

We consider a non-deterministic version of λ -calculus in a pure, untyped setting. The terms are those of the pure λ -calculus, augmented with a binary operator \oplus denoting a form of non-deterministic superposition:¹

$$\Lambda_{\oplus} \ni M, N, P, Q ::= x \mid \lambda x.M \mid MN \mid M \oplus N.$$

As usual λ_{\oplus} -terms are considered up to renaming bound variables, and we write $M[N/x]$ for the capture avoiding substitution of N for x in M . We give precedence to application over abstraction, and to abstraction over \oplus , and moreover associate applications on the left, so that we may write $\lambda x.MNP \oplus Q$ for $(\lambda x.((MN)P)) \oplus Q$. We write $\lambda \vec{x}.M$ for a term of the form $\lambda x_1. \dots \lambda x_n.M$.

Rather than specifying the computational effect of \oplus explicitly, we consider two reductions rules

$$(M \oplus N)P \rightarrow MP \oplus NP \quad \text{and} \quad \lambda x.(M \oplus N) \rightarrow \lambda x.M \oplus \lambda x.N$$

in addition to the β -reduction rule.² Formally, \rightarrow is defined inductively by the inference rules of Figure 1.1: we simply extend the three base cases contextually.

1. Throughout the thesis, we use a self explanatory if not standard variant of BNF notation for introducing syntactic objects: here we define the set Λ_{\oplus} as that inductively generated by variables, λ -abstraction, application and sum, and we will denote terms using letters among

1. Taylor Expansion for λ -Terms with Choice Operator – 1.3. A Generic Non-Deterministic λ -Calculus

$$\begin{array}{c}
\overline{(\lambda x.M)N \rightarrow M[N/x]} \quad \overline{(M \oplus N)P \rightarrow MP \oplus NP} \quad \overline{\lambda x.(M \oplus N) \rightarrow \lambda x.M \oplus \lambda x.N} \\
\hline
\frac{M \rightarrow M'}{\lambda x.M \rightarrow \lambda x.M'} \quad \frac{M \rightarrow M'}{MN \rightarrow M'N} \quad \frac{M \rightarrow M'}{NM \rightarrow NM'} \quad \frac{M \rightarrow M'}{M \oplus N \rightarrow M' \oplus N} \quad \frac{M \rightarrow M'}{N \oplus M \rightarrow N \oplus M'}
\end{array}$$

Figure 1.1. – Reduction rules of the λ_{\oplus} -calculus

In fact we will not really consider this reduction relation in the present work, and rather focus on the *hereditary head reduction strategy* obtained by defining the function $L : \Lambda_{\oplus} \rightarrow \Lambda_{\oplus}$ inductively as follows:

$$\begin{aligned}
L(M \oplus N) &:= L(M) \oplus L(N) \\
L(\lambda \vec{x}.\lambda y.(M \oplus N)) &:= \lambda \vec{x} . (\lambda y.M \oplus \lambda y.N) \\
L(\lambda \vec{x}.(M \oplus N)PQ_1 \cdots Q_k) &:= \lambda \vec{x} . (MP \oplus NP)Q_1 \cdots Q_k \\
L(\lambda \vec{x}.yQ_1 \cdots Q_k) &:= \lambda \vec{x} . yL(Q_1) \cdots L(Q_k) \\
L(\lambda \vec{x} . (\lambda y.M)NQ_1 \cdots Q_k) &:= \lambda \vec{x} . M[N/y]Q_1 \cdots Q_k \quad .
\end{aligned}$$

Observe that this definition is exhaustive because any term in Λ_{\oplus} is either of the form $M \oplus N$ or of the form $\lambda \vec{x}.\lambda y.(M \oplus N)$ or of the form $\lambda \vec{x}.RQ_1 \cdots Q_k$ with $R = (\lambda y.M)N$ or $R = (M \oplus N)P$ or $R = y$.

It should be clear that $M \rightarrow^* L(M)$ and that $L(M) = M$ whenever M is normal³ but the converse does not necessarily hold. It can moreover be shown that any normalizable term M reaches its normal form by repeatedly applying the function L , for instance by adapting the standardization techniques of Leventis [Lev16; Lev19], but this is not the focus of the present work. Indeed, we are only interested in the construction of Böhm trees: the Böhm tree of a term M can be understood as the limit of the sequence $(L^n(M))_{n \in \mathbf{N}}$.

M, N, P, Q , possibly with sub- and superscripts.

2. This is in accordance with most of the literature associated with the Taylor expansion of λ -terms [ER03; Ehr10; PTV16; Vau17] and quantitative denotational semantics [Ehr05], where non-deterministic choice is modelled by the sum of denotations: λ -abstraction is linear and term application is left-linear. In fact, only the rule $(M \oplus N)P \rightarrow MP \oplus NP$ is really necessary in order to enable the potential redexes that can occur if M or N is an abstraction. The other reduction rule can be derived in case one admits extensionality in the models or the η -rule in the calculus (here we don't, though); and the results of this chapter could be developed similarly without it. We chose to keep it nonetheless, because it simplifies the underlying theory of Böhm trees and allows us to obtain Ehrhard and Regnier's results [ER08; ER06a] as a particular case of our own.

3. If one considers \oplus as a non-deterministic choice operator, normalizability is meant in its *must* flavour here. Indeed, we do not perform the choice within the reduction relation itself, so $M \oplus N$ is normal iff M and N both are.

$$\begin{array}{c}
\frac{}{\perp \leq M} \quad \frac{}{M \leq M} \quad \frac{M \leq N \quad N \leq P}{M \leq P} \\
\frac{M \leq M'}{\lambda x.M \leq \lambda x.M'} \quad \frac{M \leq M' \quad N \leq N'}{MN \leq M'N'} \quad \frac{M \leq M' \quad N \leq N'}{M \oplus N \leq M' \oplus N'}
\end{array}$$

Figure 1.2. – The approximation order on Λ_{\perp} .

1.3.2. Böhm Trees

We first define the set Λ_{\perp} of *term approximants* as follows:

$$\Lambda_{\perp} \ni M, N, P, Q ::= \perp \mid x \mid \lambda x.M \mid MN \mid M \oplus N$$

then we consider the least partial order $\leq \subseteq \Lambda_{\perp} \times \Lambda_{\perp}$ that is compatible with syntactic constructs and such that $\perp \leq M$ for each $M \in \Lambda_{\perp}$. Formally, \leq is defined inductively by the rules of Figure 1.2.

The set $\mathcal{N} \subset \Lambda_{\perp}$ of *elementary Böhm trees* is the least set of approximants such that:

- $\perp \in \mathcal{N}$;
- $\lambda \vec{x}.xN_1 \cdots N_n \in \mathcal{N}$ as soon as $N_1, \dots, N_n \in \mathcal{N}$; and
- $N_1 \oplus N_2 \in \mathcal{N}$ as soon as $N_1, N_2 \in \mathcal{N}$.

The partial order \leq on \mathcal{N} is inherited from that on Λ_{\perp} . For each λ_{\oplus} -term M , we construct an elementary Böhm tree $\mathcal{N}(M)$ as follows:

$$\begin{aligned}
\mathcal{N}(M \oplus N) &:= \mathcal{N}(M) \oplus \mathcal{N}(N) \\
\mathcal{N}(\lambda \vec{x}.xQ_1 \cdots Q_k) &:= \lambda \vec{x}.x\mathcal{N}(Q_1) \cdots \mathcal{N}(Q_k) \\
\mathcal{N}(M) &:= \perp \quad \text{in all other cases.}
\end{aligned}$$

Lemma 1.3.1. *For any $M \in \Lambda_{\oplus}$, $\mathcal{N}(M) \leq \mathcal{N}(L(M))$.*

Proof. By induction on M . If $M = M_1 \oplus M_2$ then $\mathcal{N}(M) = \mathcal{N}(M_1) \oplus \mathcal{N}(M_2)$ and $L(M) = L(M_1) \oplus L(M_2)$, hence $\mathcal{N}(L(M)) = \mathcal{N}(L(M_1)) \oplus \mathcal{N}(L(M_2))$ and we conclude by induction hypothesis. The case $M = \lambda \vec{x}.xQ_1 \cdots Q_k$ is similar. Otherwise, $\mathcal{N}(M) = \perp \leq \mathcal{N}(L(M))$. \square

Hence for a fixed λ_{\oplus} -term M , the sequence $(\mathcal{N}(L^n(M)))_{n \in \mathbb{N}}$ is increasing, and we call its downwards closure *the Böhm tree of M* , that we denote by $BT(M)$. It could be shown that Böhm trees define a denotational semantics: if $M \rightarrow M'$ then $BT(M) = BT(M')$.⁴ Here we only use the fact that Böhm trees are invariant under hereditary head reduction, which follows directly from the definition:

4. Again, this would require the adaptation of standardization techniques to λ_{\oplus} , similar to those developed by Leventis for the probabilistic λ -calculus [Lev19].

1. Taylor Expansion for λ -Terms with Choice Operator – 1.4. Taylor Expansion in a Uniform Non-Deterministic Setting

Lemma 1.3.2. *Let $M \in \Lambda_{\oplus}$. Then $BT(M) = BT(L(M))$.*

This result will allow us to establish Step 5, *i.e.* the qualitative version of the commutation between normalization and the Taylor expansion of λ_{\oplus} -terms, to be defined in the next section.

1.4. Taylor Expansion in a Uniform Non-Deterministic Setting

In order to define Taylor expansion, we need to introduce an auxiliary language: the resource calculus.

1.4.1. Resource terms

We call *resource expressions* the elements of $\Delta_{\oplus}^{(!)} = \Delta_{\oplus} \cup \Delta_{\oplus}^!$, where the set of *resource terms* Δ_{\oplus} and the set of *resource monomials* $\Delta_{\oplus}^!$ are defined by mutual induction as follows:⁵

$$\Delta_{\oplus} \ni s, t, u, v ::= x \mid \lambda x.s \mid \langle s \rangle \bar{t} \mid s \oplus \bullet \mid \bullet \oplus s \qquad \Delta_{\oplus}^! \ni \bar{s}, \bar{t}, \bar{u}, \bar{v} ::= [s_1, \dots, s_n]$$

and, in addition to α -equivalence, we consider resource expressions up to permutations of terms in monomials, so that $[s_1, \dots, s_n]$ denotes a multiset of terms. We give precedence to application and abstraction over $- \oplus \bullet$ and $\bullet \oplus -$, and we write $\langle s \rangle \bar{t}_1 \cdots \bar{t}_n$ for $\langle \cdots \langle s \rangle \bar{t}_1 \cdots \rangle \bar{t}_n$, so that we may write $\lambda x. \langle s \rangle \bar{t} \bar{u} \oplus \bullet$ for $(\lambda x. (\langle \langle s \rangle \bar{t} \bar{u} \rangle) \oplus \bullet)$. We write $\lambda \vec{x}. s$ for a term of the form $\lambda x_1. \cdots \lambda x_n. s$. We moreover write $\bar{s} \cdot \bar{t}$ for the multiset union of \bar{s} and \bar{t} , and if $\bar{s} = [s_1, \dots, s_n]$ then $\text{len}(\bar{s}) := n$ for the size of \bar{s} .

If X is a set, we write $\mathbb{N}[X]$ for the set of finite formal sums of elements of X , or equivalently the set of finite linear combinations of elements of X with coefficients in \mathbb{N} . We extend the syntactical constructs of the resource calculus to finite sums of resource expressions by linearity, so that:

- if $S = \sum_{i=1}^n s_i$ then $\lambda x.S = \sum_{i=1}^n \lambda x.s_i$, $\bullet \oplus S = \sum_{i=1}^n \bullet \oplus s_i$ and $S \oplus \bullet = \sum_{i=1}^n s_i \oplus \bullet$;
- if moreover $\bar{T} = \sum_{j=1}^m \bar{t}_j$ then $\langle S \rangle \bar{T} = \sum_{i=1}^n \sum_{j=1}^m \langle s_i \rangle \bar{t}_j$ and $[S] \cdot \bar{T} = \sum_{i=1}^n \sum_{j=1}^m [s_i] \cdot \bar{t}_j$.

For any resource expression e , we write $n_x(e)$ for the number of occurrences of variable x in e . If $e \in \Delta_{\oplus}^{(!)}$, $\bar{u} = [u_1, \dots, u_n] \in \Delta_{\oplus}^!$ and $x \in \mathcal{V}$, we introduce the

5. Recall that the cartesian product of vector spaces is given by the disjoint union of bases: this is the intuition behind the operators $- \oplus \bullet$ and $\bullet \oplus -$, which will serve in the Taylor expansion of the operator \oplus of Λ_{\oplus} . Indeed, we leave the exact computational behavior of \oplus unspecified, and we treat it generically as a pairing operator (without projections): in this we follow Tsukada, Asada and Ong [TAO17].

1. Taylor Expansion for λ -Terms with Choice Operator – 1.4. Taylor Expansion in a Uniform Non-Deterministic Setting

$$\begin{array}{c}
\overline{\langle \lambda x.s \rangle \bar{t}} \rightarrow_{\partial} \overline{\partial_x s \cdot \bar{t}} \quad \overline{\langle s \oplus \bullet \rangle \bar{t}} \rightarrow_{\partial} \overline{\langle s \rangle \bar{t} \oplus \bullet} \quad \overline{\langle \bullet \oplus s \rangle \bar{t}} \rightarrow_{\partial} \overline{\bullet \oplus \langle s \rangle \bar{t}} \\
\\
\overline{\lambda x.(s \oplus \bullet)} \rightarrow_{\partial} \overline{\lambda x.s \oplus \bullet} \quad \overline{\lambda x.(\bullet \oplus s)} \rightarrow_{\partial} \overline{\bullet \oplus \lambda x.s} \\
\\
\frac{s \rightarrow_{\partial} S'}{\lambda x.s \rightarrow_{\partial} \lambda x.S'} \quad \frac{s \rightarrow_{\partial} S'}{\langle s \rangle \bar{t} \rightarrow_{\partial} \langle S' \rangle \bar{t}} \quad \frac{\bar{s} \rightarrow_{\partial} \bar{S}'}{\langle t \rangle \bar{s} \rightarrow_{\partial} \langle t \rangle \bar{S}'} \\
\\
\frac{s \rightarrow_{\partial} S'}{s \oplus \bullet \rightarrow_{\partial} S' \oplus \bullet} \quad \frac{s \rightarrow_{\partial} S'}{\bullet \oplus s \rightarrow_{\partial} \bullet \oplus S'} \quad \frac{s \rightarrow_{\partial} S'}{[s] \cdot \bar{t} \rightarrow_{\partial} [S'] \cdot \bar{t}}
\end{array}$$

Figure 1.3. – Reduction rules of the resource calculus with sums

symmetric n -linear substitution $\partial_x e \cdot \bar{u} \in \mathbb{N}[\Delta_{\oplus}^{(l)}]$ of \bar{u} for x in e , which is informally defined as follows:

$$\partial_x e \cdot \bar{u} := \begin{cases} \sum_{\sigma \in \mathfrak{S}_n} e[u_{\sigma(1)}/x_1, \dots, u_{\sigma(n)}/x_n] & \text{if } n_x(e) = n \\ 0 & \text{otherwise} \end{cases}$$

where $x_1, \dots, x_{n_x(e)}$ enumerate the occurrences of x in e .⁶

Formally, $\partial_x e \cdot \bar{u}$ is defined by induction on e , setting:

$$\begin{aligned}
\partial_x y \cdot \bar{u} &:= \begin{cases} y & \text{if } y \neq x \text{ and } n = 0 \\ u_1 & \text{if } y = x \text{ and } n = 1 \\ 0 & \text{otherwise} \end{cases} \\
\partial_x \lambda y.s \cdot \bar{u} &:= \lambda y.(\partial_x s \cdot \bar{u}) \\
\partial_x (s \oplus \bullet) \cdot \bar{u} &:= \partial_x s \cdot \bar{u} \oplus \bullet \\
\partial_x (\bullet \oplus s) \cdot \bar{u} &:= \bullet \oplus \partial_x s \cdot \bar{u} \\
\partial_x \langle s \rangle \bar{t} \cdot \bar{u} &:= \sum_{(I_0, I_1) \text{ partition of } \{1, \dots, n\}} \langle \partial_x s \cdot \bar{u}_{I_0} \rangle \partial_x \bar{t} \cdot \bar{u}_{I_1} \\
\partial_x [t_1, \dots, t_k] \cdot \bar{u} &:= \sum_{(I_1, \dots, I_k) \text{ partition of } \{1, \dots, n\}} [\partial_x t_1 \cdot \bar{u}_{I_1}, \dots, \partial_x t_k \cdot \bar{u}_{I_k}]
\end{aligned}$$

where we write $\bar{u}_{\{i_1, \dots, i_j\}} := [u_{i_1}, \dots, u_{i_j}]$ whenever $1 \leq i_1 < \dots < i_j \leq n$.⁷

The reduction of the resource calculus is the relation from resource expressions

6. Enumerating the occurrences of x in e only makes sense if we fix an ordering of each monomial in e : the rigid resource calculus to be introduced later in this chapter will allow us to give a more formal account of this intuitive presentation. For now we stick to the alternative definition given in the next paragraph.

7. To be precise, we say (I_1, \dots, I_k) is a partition of a set X if the I_j 's are (possibly empty) pairwise disjoint subsets of X and $X = \bigcup_j I_j$. This data is equivalent to a function $X \rightarrow \{1, \dots, k\}$.

1. Taylor Expansion for λ -Terms with Choice Operator – 1.4. Taylor Expansion in a Uniform Non-Deterministic Setting

to finite formal sums of resource expressions induced by the rules of Figure 1.3: the first rule is the counterpart of β -reduction in the resource calculus; the next four rules implement the commutation of \oplus with abstraction and application to a monomial; the final six rules ensure the contextuality of the resulting relation.

It is extended to a binary relation on $\Delta_{\oplus}^{(!)}$ by setting $e + F \rightarrow_{\partial} E' + F$ whenever $e \rightarrow_{\partial} E'$. As for the original resource calculus [ER08], the reduction relation \rightarrow_{∂} is confluent and strongly normalizing. We write $\text{nf}(E)$ for the unique normal form of $E \in \mathbb{N}[\Delta_{\oplus}^{(!)}]$, which is a linear operator: $\text{nf}(\sum_{i=1}^k e_i) = \sum_{i=1}^k \text{nf}(e_i)$.

Again, we do not focus on the reduction relation itself, and we rather consider the *hereditary head reduction* strategy obtained by defining the function $L_{\partial} : \Delta_{\oplus}^{(!)} \rightarrow \mathbb{N}[\Delta_{\oplus}^{(!)}]$ inductively as follows:

$$\begin{aligned}
L_{\partial}(s \oplus \bullet) &:= L_{\partial}(s) \oplus \bullet & L_{\partial}(\bullet \oplus s) &:= \bullet \oplus L_{\partial}(s) \\
L_{\partial}(\lambda \vec{x}. \lambda y. (s \oplus \bullet)) &:= \lambda \vec{x}. (\lambda y. s \oplus \bullet) & L_{\partial}(\lambda \vec{x}. \lambda y. (\bullet \oplus s)) &:= \lambda \vec{x}. (\bullet \oplus \lambda y. s) \\
L_{\partial}(\lambda \vec{x}. \langle \langle s \oplus \bullet \rangle \bar{t} \rangle \bar{u}_1 \cdots \bar{u}_k) &:= \lambda \vec{x}. \langle \langle s \rangle \bar{t} \oplus \bullet \rangle \bar{u}_1 \cdots \bar{u}_k \\
L_{\partial}(\lambda \vec{x}. \langle \langle \bullet \oplus s \rangle \bar{t} \rangle \bar{u}_1 \cdots \bar{u}_k) &:= \lambda \vec{x}. \langle \bullet \oplus \langle s \rangle \bar{t} \rangle \bar{u}_1 \cdots \bar{u}_k \\
L_{\partial}(\lambda \vec{x}. \langle y \rangle \bar{s}_1 \cdots \bar{s}_k) &:= \lambda \vec{x}. \langle y \rangle L_{\partial}(\bar{s}_1) \cdots L_{\partial}(\bar{s}_k) \\
L_{\partial}([s_1, \dots, s_k]) &:= [L_{\partial}(s_1), \dots, L_{\partial}(s_k)] \\
L_{\partial}(\lambda \vec{x}. \langle \lambda y. s \rangle \bar{t} \bar{u}_1 \cdots \bar{u}_k) &:= \lambda \vec{x}. \langle \partial_y s \cdot \bar{t} \rangle \bar{u}_1 \cdots \bar{u}_k
\end{aligned}$$

extended to sums of resource expressions by linearity, setting $L_{\partial}(\sum_{i=1}^k e_i) := \sum_{i=1}^k L_{\partial}(e_i)$.

It should be clear that $E \rightarrow_{\partial}^* L_{\partial}(E)$, and $E = L_{\partial}(E)$ iff E is normal: here we obtain an equivalence because, if $e \rightarrow_{\partial} E'$ then each $e' \in \text{supp}(E')$ is strictly smaller than e (in the sense of the number of syntactic constructs). It moreover follows that L_{∂} is normalizing: for all $s \in \Delta_{\oplus}$, there is n such that $L_{\partial}^n(s) = \text{nf}(s)$.

1.4.2. Taylor Expansion of λ_{\oplus} -Terms

The Taylor expansion of a λ_{\oplus} -term will be an infinite linear combination of resource terms: to introduce it, we first need some preliminary notations and results.

If X is a set, we write $\mathbb{Q}^+\langle X \rangle$ for the set of possibly infinite linear combinations of elements of X with non negative rational coefficients (in fact we could use any commutative semifield): equivalently, $\mathbb{Q}^+\langle X \rangle$ is the set of functions from X to the set of non negative rational numbers. We write $A = \sum_{a \in X} A_a \cdot a \in \mathbb{Q}^+\langle X \rangle$ and then the *support set* of A is $\text{supp}(A) = \{a \in X \mid A_a \neq 0\}$. If $\vec{a} = (a_1, \dots, a_n) \in A^n$, we write $A^{\vec{a}} = \prod_{i=1}^n A_{a_i}$: observe that this does not depend on the ordering of the a_i 's, so if $\bar{a} = [a_1, \dots, a_n]$ is a finite multiset of elements of A , we may as well write $A^{\bar{a}} = A^{(a_1, \dots, a_n)}$.

All the syntactic constructs we have introduced on resource expressions can be extended by linear-continuity: e.g, if $S \in \mathbb{Q}^+\langle \Delta_{\oplus} \rangle$ then $\lambda x. S = \sum_{s \in \Delta_{\oplus}} \lambda x. S_s \cdot s$. They

1. Taylor Expansion for λ -Terms with Choice Operator – 1.4. Taylor Expansion in a Uniform Non-Deterministic Setting

also extend to sets of expressions, *via* the support function: e.g, if $E \subseteq \Delta_{\oplus}^{(!)}$ and $\bar{T} \subseteq \Delta_{\oplus}^!$ then $\partial_x E \cdot \bar{T} = \bigcup_{e \in E} \bigcup_{\bar{t} \in \bar{T}} \text{supp}(\partial_x e \cdot \bar{t})$.⁸

Let $S \in \mathbb{Q}^+ \langle \Delta_{\oplus} \rangle$. We define $S^n \in \mathbb{Q}^+ \langle \Delta_{\oplus}^! \rangle$ by induction on n : $S^0 = []$ and $S^{n+1} = [S] \cdot S^n$. Then we define the *promotion* of S as the series $S^! = \sum_{n=0}^{\infty} \frac{1}{n!} S^n$: because the supports of S^n and S^p are disjoint when $n \neq p$, this sum is pointwise finite. If $S \subseteq \Delta_{\oplus}$ is a set of terms, we may also write $S^! = \{[s_1, \dots, s_n] \mid s_1, \dots, s_n \in S\}$ for the set of monomials of terms in S , so that $\text{supp}(S^!) = \text{supp}(S)^!$ for any $S \in \mathbb{Q}^+ \langle \Delta_{\oplus} \rangle$.

We define the *Taylor expansion* $M^* \in \mathbb{Q}^+ \langle \Delta_{\oplus} \rangle$ of $M \in \Lambda_{\oplus}$ inductively as follows:

$$\begin{aligned} x^* &:= x \\ (\lambda x.N)^* &:= \lambda x.N^* \\ (PQ)^* &:= \langle P^* \rangle (Q^*)^! \\ (P \oplus Q)^* &:= (P^* \oplus \bullet) + (\bullet \oplus Q^*) \quad . \end{aligned}$$

Note that this definition follows the one for the ordinary λ -calculus given by Ehrhard and Regnier [ER08], in the form described in their Lemma 18. We extend it to \oplus by encoding the pair of vectors (P^*, Q^*) as the sum vector $(P^* \oplus \bullet) + (\bullet \oplus Q^*)$.

Writing $\mathcal{T}(M) := \text{supp}(M^*)$ for the support of Taylor expansion, we obtain:

$$\begin{aligned} \mathcal{T}(x) &= \{x\} \\ \mathcal{T}(\lambda x.N) &= \lambda x.\mathcal{T}(N) = \{\lambda x.t \mid t \in \mathcal{T}(N)\} \\ \mathcal{T}(PQ) &= \langle \mathcal{T}(P) \rangle \mathcal{T}(Q)^! = \{[s][t_1, \dots, t_n] \mid s \in \mathcal{T}(P) \text{ and } t_1, \dots, t_n \in \mathcal{T}(Q)\} \\ \mathcal{T}(P \oplus Q) &= (\mathcal{T}(P) \oplus \bullet) \cup (\bullet \oplus \mathcal{T}(Q)) = \{s \oplus \bullet \mid s \in \mathcal{T}(P)\} \cup \{\bullet \oplus t \mid t \in \mathcal{T}(Q)\} \end{aligned}$$

so that $M^* = \sum_{s \in \mathcal{T}(M)} M_s^* s$.

We can immediately check that Step 2 still holds for our extension of Taylor expansion to λ_{\oplus} -terms: we prove that $\mathcal{T}(M)$ is always a clique for the coherence relation $\supset \subseteq \Delta_{\oplus}^{(!)} \times \Delta_{\oplus}^{(!)}$ inductively defined by the rules of Figure 1.4. The first four rules are exactly those for the ordinary resource calculus [ER08, Section 3], while the last three rules are reminiscent of the definition of the cartesian product of coherence spaces [Gir87, Definition 5]. Again, this is consistent with the fact that we treat \oplus as a pairing construct, denoting an unspecified superposition operation.

Observe that the relation \supset is automatically symmetric, but not reflexive: e.g., $[s, t] \not\supset [s, t]$ when $s \not\supset t$. We say a resource expression e is *uniform* if $e \supset e$, so that uniform expressions form a coherence space in the usual sense.

We obtain the expected result by a straightforward induction on λ_{\oplus} -terms:

8. Alternatively, we could consider subsets of $\Delta_{\oplus}^{(!)}$ as infinite linear combinations of resource expressions with boolean coefficients, and apply linear-continuity in this context: the two approaches coincide here because syntactic constructs extended by linear-continuity commute with the support function, e.g., $\lambda x.\text{supp}(S) = \text{supp}(\lambda x.S)$.

1. Taylor Expansion for λ -Terms with Choice Operator – 1.4. Taylor Expansion in a Uniform Non-Deterministic Setting

$$\begin{array}{c}
\frac{x \circ x}{x \circ x} \quad \frac{s \circ s'}{\lambda x.s \circ \lambda x.s'} \quad \frac{s \circ s' \quad \bar{t} \circ \bar{t}'}{\langle s \rangle \bar{t} \circ \langle s' \rangle \bar{t}'} \quad \frac{t_i \circ t_j \text{ for } 1 \leq i, j \leq n+m}{[t_1, \dots, t_n] \circ [t_{n+1}, \dots, t_{n+m}]} \\
\frac{s \circ s'}{s \oplus \bullet \circ s' \oplus \bullet} \quad \frac{s \circ s'}{\bullet \oplus s \circ \bullet \oplus s'} \quad \frac{}{s \oplus \bullet \circ \bullet \oplus s'}
\end{array}$$

Figure 1.4. – Rules for the coherence relation on $\Delta_{\oplus}^{(!)}$.

Theorem 1.4.1 (Step 2). *The Taylor support $\mathcal{T}(M)$ is a clique: $s \circ s'$ for all $s, s' \in \mathcal{T}(M)$.*

1.4.3. Multiplicity Coefficients

We now generalize Step 1 in our generic non-deterministic setting: we can define a multiplicity coefficient $m(s)$ for each $s \in \Delta_{\oplus}$ so that $M_s^* = \frac{1}{m(s)}$ whenever $s \in \mathcal{T}(M)$.

Given any set X and $n \in \mathbb{N}$, we consider the left action of the group \mathfrak{S}_n of all permutations of $\{1, \dots, n\}$ on the set X^n of n -tuples, defined as follows: if $\vec{a} = (a_1, \dots, a_n)$ and $\sigma \in \mathfrak{S}_n$ then $[\sigma]\vec{a} = (a_{\sigma^{-1}(1)}, \dots, a_{\sigma^{-1}(n)})$. Writing $[\sigma]\vec{a} = (a'_1, \dots, a'_n)$, we obtain $a'_{\sigma(i)} = a_i$. Let us recall that if $\vec{a} \in X^n$, then the stabilizer of \vec{a} is $St(\vec{a}) = \{\sigma \in \mathfrak{S}_n \mid [\sigma]\vec{a} = \vec{a}\}$.

Lemma 1.4.2. *Let $S \in \mathbb{Q}^+\langle \Delta_{\oplus} \rangle$ and $\bar{s} \in \text{supp}(S^!)$. If $\vec{s} = (s_1, \dots, s_n)$ is an enumeration of \bar{s} , i.e. $[s_1, \dots, s_n] = \bar{s}$, then $(S^!)_{\bar{s}} = \frac{S^{\bar{s}}}{\text{Card}(St(\vec{s}))}$.*

Proof. By definition and by linearity we have

$$S^! = \sum_{n=0}^{\infty} \sum_{(s_1, \dots, s_n) \in \Delta^n} \frac{S_{s_1} \cdots S_{s_n}}{n!} [s_1, \dots, s_n] \quad .$$

If $\text{len}(\bar{s}) = n$, we thus obtain:

$$(S^!)_{\bar{s}} = \frac{\text{Card}(\{(s_1, \dots, s_n) \mid [s_1, \dots, s_n] = \bar{s}\})}{n!} S^{\bar{s}} \quad .$$

Observing that $\{(s_1, \dots, s_n) \mid [s_1, \dots, s_n] = \bar{s}\}$ is the orbit of any enumeration of \bar{s} under the action of \mathfrak{S}_n , and that $\text{Card}(\mathfrak{S}_n) = n!$, we conclude by Fact 1.2.1. \square

Let $s \in \Delta_{\oplus}$. We inductively define $m(s)$, the *multiplicity coefficient* of s , as

1. Taylor Expansion for λ -Terms with Choice Operator – 1.4. Taylor Expansion in a Uniform Non-Deterministic Setting

follows:

$$\begin{aligned}
m(x) &:= 1 \\
\left. \begin{array}{l} m(\lambda x.s) \\ m(s \oplus \bullet) \\ m(\bullet \oplus s) \end{array} \right\} &:= m(s) \\
m(\langle s \rangle \bar{t}) &:= m(s)m(\bar{t}) \\
m([t_1^{n_1}, \dots, t_n^{n_n}]) &:= \prod_{i=1}^n n_i! m(t_i)^{n_i}
\end{aligned}$$

assuming the t_i 's are pairwise distinct in the case of a monomial. Again, this definition extends straightforwardly the one given by Ehrhard and Regnier for their resource calculus [ER08, Section 2.2.1], given that $-\oplus\bullet$ and $\bullet\oplus-$ are both linear. Observe that, considering the function m as a vector $m \in \mathbf{N}\langle\Delta_{\oplus}^{(!)}\rangle \subseteq \mathbb{Q}^+\langle\Delta_{\oplus}^{(!)}\rangle$, if \bar{s} is an enumeration of \bar{s} then $m(\bar{s}) = m^{\bar{s}} \text{Card}(St(\bar{s}))$.

Theorem 1.4.3 (Step 1). *Let $s \in \mathcal{T}(M)$. Then $M_s^* = \frac{1}{m(s)}$.*

Proof. The only interesting case is that of an application: $M = PQ$. Assume $s \in \mathcal{T}(M)$: then $s = \langle u \rangle \bar{v}$ with $u \in \mathcal{T}(P)$ and $\bar{v} = [v_1, \dots, v_n] \in \mathcal{T}(Q)!$. By definition, $M_s^* = (\langle P^* \rangle (Q^*)!)_{\langle u \rangle \bar{v}} = P_u^* (Q^*)!_{\bar{v}}$. Setting $\vec{v} = (v_1, \dots, v_n)$, we obtain $M_s^* = P_u^* (Q^*)^{\bar{v}} / \text{Card}(St(\vec{v}))$ by Lemma 1.4.2. By the induction hypothesis applied to P and Q , we obtain $1/P_u^* = m(u)$ and $1/Q_{v_i}^* = m(v_i)$ hence $1/M_s^* = m(u)m^{\bar{v}} \text{Card}(St(\vec{v})) = m(u)m(\bar{v}) = m(s)$. \square

We could as well obtain Step 3 following Ehrhard and Regnier's study of permutations of variables occurrences. At this point, however, we hope the reader will share our opinion that the combinatorics of Taylor expansion is more intimately connected with the action of permutations on the enumerations of monomials occurring in resource expressions.

In the upcoming Section 1.5, we propose to flesh out this viewpoint, and to recast resource expressions as equivalence classes of their rigid (*i.e.* non-commutative) representatives, up to the action of a groupoid of permutation terms inductively defined on the syntactic structure.

The other remaining Steps 4 and 5 are purely qualitative properties of the Taylor support. We chose to also treat Step 4 in the rigid setting, to be introduced later, because it is essentially a property of rigid reduction. On the other hand, the commutation of Step 5 can be established directly.

1.4.4. Taylor Expansion of Böhm Trees

The Taylor expansion of a Böhm tree is obtained as follows. First we extend the definition of Taylor expansion from Λ_{\oplus} to Λ_{\perp} by adding the inductive case $\perp^* := 0$, hence $\mathcal{T}(\perp) = \emptyset$. Then we set $\mathcal{T}(BT(M)) := \bigcup_{B \in BT(M)} \mathcal{T}(B)$.

1. Taylor Expansion for λ -Terms with Choice Operator – 1.4. Taylor Expansion in a Uniform Non-Deterministic Setting

We can already observe that if $s \in \mathcal{T}(BT(M))$ then s is normal. Moreover, it is not difficult to extend Theorem 1.4.3 to elementary Böhm trees, hence $B_s = \frac{1}{m(s)}$ whenever $s \in \mathcal{T}(B)$. Thus, it only makes sense to define the Taylor expansion of a Böhm tree as: $BT(M)^* := \sum_{s \in \mathcal{T}(BT(M))} \frac{1}{m(s)}$.

We shall achieve Step 5 by showing that the parallel left strategy in Λ_{\oplus} can be simulated in the support of Taylor expansion, and that $\mathcal{T}(BT(M))$ is formed by accumulating the normal forms reached from $\mathcal{T}(M)$ by this strategy.

Lemma 1.4.4. *Let M be a λ_{\oplus} -term. Then $L_{\partial}(\mathcal{T}(M)) = \mathcal{T}(L(M))$.*

Proof. The proof is the same as for λ -terms [Vau17], the case of \oplus being direct. The base case requires to prove that $\mathcal{T}(M[N/x]) = \partial_x \mathcal{T}(M) \cdot \mathcal{T}(N)^\dagger$, which is done by a straightforward induction on M . \square

Lemma 1.4.5. *Let $A, B \in \Lambda_{\perp}$. If $A \leq B$ then $T(A) \subseteq T(B)$.*

Proof. By induction on A . If $A = \perp$ the proof is trivial. If $A = \lambda \vec{x}.x.A_1 \cdots A_n$ then by definition of the partial order $B = \lambda \vec{x}.xB_1 \cdots B_n$ with $A_i \leq B_i$ for $i \in \{1, \dots, n\}$. By induction hypothesis we have that $T(A_i) \subseteq T(B_i)$. Therefore by definition $T(A) \subseteq T(B)$. If $A = A_1 \oplus A_2$, we reason similarly, as we necessarily have $B = B_1 \oplus B_2$. \square

Lemma 1.4.6. *For any $M \in \Lambda_{\oplus}$, $\mathcal{T}(\mathcal{N}(M)) = \{s \in \mathcal{T}(M) \mid s \text{ is normal}\}$.*

Proof. The inclusion \subseteq follows from Lemma 1.4.5 and the obvious fact that $\mathcal{N}(M) \leq M$. Conversely, if $s \in \mathcal{T}(M)$ and s is normal, then either $M = N \oplus P$ and $s = t \oplus \bullet$ or $s = \bullet \oplus u$ with $t \in \mathcal{T}(N)$ or $u \in \mathcal{T}(P)$; or $M = \lambda \vec{x}.x.Q_1 \cdots Q_k$ and $s = \lambda \vec{x}.\langle x \rangle \bar{q}_1 \cdots \bar{q}_k$ with $\bar{q}_i \in \mathcal{T}(Q_i)^\dagger$ for $1 \leq i \leq k$. We obtain inductively $t \in \mathcal{T}(\mathcal{N}(N))$ or $u \in \mathcal{T}(\mathcal{N}(P))$ or $\bar{q}_i \in \mathcal{T}(\mathcal{N}(Q_i))^\dagger$ for $1 \leq i \leq k$, and then $s \in \mathcal{T}(\mathcal{N}(M))$. \square

Step 5 then follows, using the fact that $BT(M)$ is the downwards closure of $\{\mathcal{N}(L^n(M)) \mid n \in \mathbb{N}\}$:

Theorem 1.4.7 (Step 5). *Let $M \in \Lambda_{\oplus}$. Then $\mathcal{T}(BT(M)) = \text{nf}(\mathcal{T}(M))$.*

Proof. Observe that $\text{nf}(\mathcal{T}(M)) = \bigcup_{s \in \mathcal{T}(M)} \text{supp}(\text{nf}(s))$. The proof is by double inclusion.

(\subseteq) Let $t \in \mathcal{T}(BT(M))$, i.e. $t \in \mathcal{T}(B)$ for some $B \in BT(M)$. By the definition of $BT(M)$, there exists $n \in \mathbb{N}$ such that $B \leq \mathcal{N}(L^n(M))$, and then by Lemma 1.4.5 $t \in \mathcal{T}(\mathcal{N}(L^n(M)))$. By Lemma 1.4.6, t is normal and $t \in \mathcal{T}(L^n(M))$. By Lemma 1.4.4, $t \in L_{\partial}^n(\mathcal{T}(M))$, hence there exists $s \in \mathcal{T}(M)$ such that $t \in \text{supp}(L_{\partial}^n(s))$. Since t is normal, $t \in \text{supp}(\text{nf}(s))$.

(\supseteq) If $t \in \text{nf}(\mathcal{T}(M))$ we can fix $s \in \mathcal{T}(M)$ such that $t \in \text{supp}(\text{nf}(s))$. Then there exists $n \in \mathbb{N}$ such that $\text{nf}(s) = L_{\partial}^n(s)$. Hence t is normal and $t \in L_{\partial}^n(\mathcal{T}(M))$. By Lemma 1.4.4, $t \in \mathcal{T}(L^n(M))$ and since t is normal, Lemma 1.4.6 entails that $t \in \mathcal{T}(\mathcal{N}(L^n(M)))$, hence $t \in \mathcal{T}(BT(L^n(M)))$ and we conclude by Lemma 1.3.2. \square

1. Taylor Expansion for λ -Terms with Choice Operator – 1.5. The Groupoid of Permutations of Rigid Resource Terms

$$\begin{array}{c}
\frac{}{x \triangleleft x} \qquad \frac{a \triangleleft s}{\lambda x.a \triangleleft \lambda x.s} \qquad \frac{a \triangleleft s}{a \oplus \bullet \triangleleft s \oplus \bullet} \qquad \frac{a \triangleleft s}{\bullet \oplus a \triangleleft \bullet \oplus s} \\
\frac{c \triangleleft s \quad \vec{d} \triangleleft \vec{t}}{\langle c \rangle \vec{d} \triangleleft \langle s \rangle \vec{t}} \qquad \frac{a_1 \triangleleft t_1 \quad \cdots \quad a_n \triangleleft t_n}{(a_1, \dots, a_n) \triangleleft [t_1, \dots, t_n]}
\end{array}$$

Figure 1.5. – Rules for the rigid representation relation

1.5. The Groupoid of Permutations of Rigid Resource Terms

1.5.1. Rigid Resource Terms and Permutation Terms

We introduce the set of rigid resource terms D and the set of rigid monomials $D^!$ by mutual induction as follows:

$$D \ni a, b, c, d ::= x \mid \lambda x.a \mid \langle a \rangle \vec{b} \mid \bullet \oplus a \mid a \oplus \bullet \qquad D^! \ni \vec{a}, \vec{b}, \vec{c}, \vec{d} ::= (a_1, \dots, a_n) \quad .$$

Rigid resource terms are considered up to renaming of bound variables: the only difference with resource terms is that rigid monomials are ordered lists rather than finite multisets. We write $\text{len}((a_1, \dots, a_n)) := n$, and $(a_1, \dots, a_n) ::= (a_{n+1}, \dots, a_{n+m}) := (a_1, \dots, a_{n+m})$. We call *rigid resource expressions* the elements of $D^{(!)} = D \cup D^!$. Again, for any $r \in D^{(!)}$, we write $n_x(r)$ for the number of free occurrences of the variable x in r , and we use notations and priority conventions similar to those for non rigid expressions: e.g., we may write $\lambda \vec{x}. \langle a \rangle \vec{b} \vec{c} \oplus \bullet$ for $(\lambda x_1. \dots \lambda x_n. (\langle a \rangle \vec{b} \vec{c})) \oplus \bullet$.

As we have already stated, rigid resource expressions are nothing but resource expressions for which the order of terms in monomials matter. To make this connexion formal, consider the *representation relation* $\triangleleft \subseteq D^{(!)} \times \Delta_{\oplus}^{(!)}$ defined by the rules of Figure 1.5. Observe that the relation \triangleleft is the graph of a function: if $r \in D^{(!)}$, there exists a unique $e \in \Delta_{\oplus}^{(!)}$ such that $r \triangleleft e$, and then we write $e = \|r\|$. Moreover observe that, if $\vec{a} \triangleleft \vec{t}$ and $\text{len}(\vec{a}) = n$ then for any $\sigma \in \mathfrak{S}_n$, $[\sigma]\vec{a} \triangleleft \vec{t}$, i.e. $\|[\sigma]\vec{a}\| = \|\vec{a}\|$.

We now introduce a syntax for the trees of permutations that can act on monomials at any depth in a rigid expression. The language of such *permutation expressions* is given as follows:

$$\mathcal{D} \ni \alpha, \beta, \gamma, \delta ::= id_x \mid \lambda x.\alpha \mid \langle \alpha \rangle \tilde{\beta} \mid \alpha \oplus \bullet \mid \bullet \oplus \alpha \qquad \mathcal{D}^! \ni \tilde{\alpha}, \tilde{\beta}, \tilde{\gamma}, \tilde{\delta} ::= (\sigma, (\alpha_1, \dots, \alpha_n))$$

where x ranges over variables and σ ranges over \mathfrak{S}_n in the pair $(\sigma, (\alpha_1, \dots, \alpha_n))$. In other words, a *permutation term* (resp. *permutation monomial*) is nothing but a rigid term (resp. rigid monomial), with a permutation attached with each list

1. Taylor Expansion for λ -Terms with Choice Operator – 1.5. The Groupoid of Permutations of Rigid Resource Terms

$$\begin{array}{c}
\frac{}{id_x : x \cong x} \quad \frac{\alpha : a \cong a'}{\lambda x. \alpha : \lambda x. a \cong \lambda x. a'} \quad \frac{\gamma : c \cong c' \quad \delta : \vec{d} \cong \vec{d}'}{\langle \gamma \rangle \delta : \langle c \rangle \vec{d} \cong \langle c' \rangle \vec{d}'} \\
\frac{\alpha : a \cong a'}{\alpha \oplus \bullet : a \oplus \bullet \cong a' \oplus \bullet} \quad \frac{\alpha : a \cong a'}{\bullet \oplus \alpha : \bullet \oplus a \cong \bullet \oplus a'} \\
\frac{\sigma \in \mathfrak{S}_n \quad \alpha_1 : a_1 \cong a'_{\sigma(1)} \quad \cdots \quad \alpha_n : a_n \cong a'_{\sigma(n)}}{(\sigma, \alpha_1, \dots, \alpha_n) : (a_1, \dots, a_n) \cong (a'_1, \dots, a'_n)}
\end{array}$$

Figure 1.6. – Action of permutation expressions on rigid expressions

of arguments. In general, we will simply write $(\sigma, \alpha_1, \dots, \alpha_n)$ for the permutation monomial $(\sigma, (\alpha_1, \dots, \alpha_n))$.

We say $\epsilon \in \mathcal{D}^{(l)}$ maps $r \in D^{(l)}$ to $r' \in D^{(l)}$ if the statement $\epsilon : r \cong r'$ is derivable from the rules of Figure 1.6. Observe that, given $r \in D^{(l)}$ and $\epsilon \in \mathcal{D}^{(l)}$ there is at most one $r' \in D^{(l)}$ such that $\epsilon : r \cong r'$, in which case we write $[\epsilon]r := r'$. We then write $\mathcal{D}^{(l)}(r, r')$ for the set of permutation expressions that map r to r' , *i.e.* $\mathcal{D}^{(l)}(r, r') = \{\epsilon \in \mathcal{D}^{(l)} \mid \epsilon : r \cong r'\}$. We moreover write $r \cong r'$ if there exists some $\epsilon \in \mathcal{D}^{(l)}$ such that $\epsilon : r \cong r'$.

As a direct consequence of the definitions, we obtain that \cong is nothing but the equivalence kernel of the function $r \in D^{(l)} \mapsto \|r\| \in \Delta_{\oplus}^{(l)}$:

Lemma 1.5.1. *For all $r, r' \in D^{(l)}$, $r \cong r'$ iff $\|r\| = \|r'\|$.*

The equivalence classes for \cong are thus exactly the sets of rigid representations of each resource expression. We can organize the permutation expressions witnessing this equivalence relation into a groupoid \mathbb{G} : take $D^{(l)}$ as the collection of objects and for $r, r' \in D^{(l)}$, $\mathbb{G}(r, r') = \{(r, r', \epsilon) \mid \epsilon : r \cong r'\}$. We will in general abuse the definition and identify $(r, r', \epsilon) \in \mathbb{G}(r, r')$ with $\epsilon \in \mathcal{D}^{(l)}(r, r')$. The composition $\epsilon' \epsilon \in \mathbb{G}(r, r'')$ of $\epsilon \in \mathbb{G}(r, r')$ and $\epsilon' \in \mathbb{G}(r', r'')$ is defined by induction on the syntax of rigid resource expressions in the obvious way: the only interesting case is that of permutation monomials, for which we set $(\sigma', \alpha'_1, \dots, \alpha'_n)(\sigma, \alpha_1, \dots, \alpha_n) := (\sigma' \sigma, \alpha'_{\sigma(1)} \alpha_1, \dots, \alpha'_{\sigma(n)} \alpha_n)$. And the identity 1_r on r is the same as r , with each variable occurrence x replaced with 1_x . Inverses are also defined inductively, the key case of monomials being: $(\sigma, \alpha_1, \dots, \alpha_n)^{-1} := (\sigma^{-1}, \alpha_{\sigma^{-1}(1)}^{-1}, \dots, \alpha_{\sigma^{-1}(n)}^{-1})$.

We obtain a left action of the groupoid \mathbb{G} on $D^{(l)}$: $[1_r]r = r$ and $[\epsilon' \epsilon]r = [\epsilon'][\epsilon]r$ whenever either side of the identity is defined. It is also worth noting that $\mathbb{G} = \mathbb{D} \oplus \mathbb{D}^!$ where \mathbb{D} is the full subcategory of \mathbb{G} defined by rigid terms, while $\mathbb{D}^!$ is the full subcategory of \mathbb{G} defined by rigid monomials. Moreover observe that $\mathbb{D}^!$ is the free symmetric strict monoidal category over \mathbb{D} [Fio+08]. Then if $[\alpha_i]a_i$ is defined for $1 \leq i \leq n$ then $[(\sigma, \alpha_1, \dots, \alpha_n)](a_1, \dots, a_n) = [\sigma]([\alpha_1]a_1, \dots, [\alpha_n]a_n)$.

1. Taylor Expansion for λ -Terms with Choice Operator – 1.5. The Groupoid of Permutations of Rigid Resource Terms

If $\vec{a} = (a_1, \dots, a_n)$ and $\vec{a}' = (a'_1, \dots, a'_n)$, we set $\vec{\mathbb{D}}(\vec{a}, \vec{a}') := \prod_{i=1}^n \mathbb{G}(a_i, a'_i)$. Observe that $\mathbb{G}(\vec{a}, \vec{a}') = \sum_{\sigma \in \mathfrak{S}_n} \vec{\mathbb{D}}(\vec{a}, [\sigma^{-1}] \vec{a}')$. We call *quasi-stabilizer* of \vec{a} the subgroup of \mathfrak{S}_n defined by $S(\vec{a}) := \{\sigma \in \mathfrak{S}_n \mid \text{for } 1 \leq i \leq n, a_i \cong a_{\sigma(i)}\}$. Observe that $S(\vec{a}) = St((\|a_1\|, \dots, \|a_n\|))$ and $\sigma \in S(\vec{a})$ iff $\vec{\mathbb{D}}(\vec{a}, [\sigma^{-1}] \vec{a}) \neq \emptyset$.

Let us write $\mathbb{G}(r)$ for the group of automorphisms of r : $\mathbb{G}(r) := \mathbb{G}(r, r)$. Similarly, we will write $\vec{\mathbb{D}}(\vec{a}) := \vec{\mathbb{D}}(\vec{a}, \vec{a})$.

Lemma 1.5.2. *For any $\vec{a} = (a_1, \dots, a_n) \in D^!$, $Card(\mathbb{G}(\vec{a})) = Card(S(\vec{a}) \times \vec{\mathbb{D}}(\vec{a}))$.*

Proof. Since \mathbb{G} is a groupoid, for any morphism $\epsilon : r \cong r'$, postcomposition by ϵ defines a bijection from $\mathbb{G}(r)$ to $\mathbb{G}(r, r')$. It follows that $\mathbb{G}(\vec{a}) = \sum_{\sigma \in \mathfrak{S}_n} \vec{\mathbb{D}}(\vec{a}, [\sigma^{-1}] \vec{a}) = \sum_{\sigma \in S(\vec{a})} \prod_{i=1}^n \mathbb{G}(a_i, a_{\sigma(i)})$ is in bijection with $\sum_{\sigma \in S(\vec{a})} \prod_{i=1}^n \mathbb{G}(a_i) = S(\vec{a}) \times \vec{\mathbb{D}}(\vec{a})$. \square

We are then able to formalize the interpretation of the multiplicity of a resource term s as the number of permutations of monomials in s leaving any of its writings $a \triangleleft s$ unchanged:

Lemma 1.5.3. *Let $e \in \Delta_{\oplus}^{(1)}$ and let $r \triangleleft e$. Then $m(e) = Card(\mathbb{G}(r))$.*

Proof. By induction on the structure of e . We prove the multiset case. Assume $e = \bar{s}$ and $\vec{a} = (a_1, \dots, a_n) \triangleleft \bar{s}$. Then we can write $\bar{s} = [s_1, \dots, s_n]$ so that $a_i \triangleleft s_i$ and the induction hypothesis gives $m(s_i) = Card(\mathbb{G}(a_i))$ for $1 \leq i \leq n$. Then $m(e) = Card(St((s_1, \dots, s_n))) \prod_{i=1}^n Card(\mathbb{G}(a_i)) = Card(S(\vec{a})) \times Card(\vec{\mathbb{D}}(\vec{a}))$, and we conclude by Lemma 1.5.2. \square

1.5.2. Rigid Substitution

We are now able to formalize the intuitive definition of the symmetric multilinear substitution.

For any $r \in D^{(1)}$ and $\vec{b} \in D^!$ such that $\text{len}(\vec{b}) = n_x(r) = n$, we define the *n-linear substitution* $r[\vec{b}/x]$ of \vec{b} for x in r inductively as follows:

$$\begin{aligned} x[(b)/x] &:= b \\ y[()/x] &:= y \\ (a \oplus \bullet)[\vec{b}/x] &:= a[\vec{b}/x] \oplus \bullet \\ (\bullet \oplus a)[\vec{b}/x] &:= \bullet \oplus a[\vec{b}/x] \\ (\lambda z.a)[\vec{b}/x] &:= \lambda z.a[\vec{b}/x] \\ \langle c \rangle \vec{d}[\vec{b}_0 :: \vec{b}_1/x] &:= \langle c[\vec{b}_0/x] \rangle \vec{d}[\vec{b}_1/x] \\ (a_1, \dots, a_n)[\vec{b}_1 :: \dots :: \vec{b}_n/x] &:= (a_1[\vec{b}_1/x], \dots, a_n[\vec{b}_n/x]) \end{aligned}$$

where we assume that $y \neq x$, $z \notin \{x\} \cup FV(\vec{b})$, $\text{len}(\vec{b}) = n_x(a)$, $\text{len}(\vec{b}_0) = n_x(c)$, $\text{len}(\vec{b}_1) = n_x(\vec{d})$, and $\text{len}(\vec{b}_i) = n_x(a_i)$ for $1 \leq i \leq n$.

1. Taylor Expansion for λ -Terms with Choice Operator – 1.5. The Groupoid of Permutations of Rigid Resource Terms

Observe that this substitution is only partially defined. In order to deal with the general case, we will use the nullary sum of rigid expressions $0 \in \mathbb{N}[D^{(l)}]$: again, we consider all the syntactic constructs to be linear so that we may write, e.g., $\lambda x.a$ for $a \in D \cup \{0\}$ with $\lambda x.0 = 0$. We call *partial rigid expressions* the elements of $D^{(l)} \cup \{0\}$: we generally use the same typographic conventions for partial expressions as for regular ones.

Whenever $r \in D^{(l)} \cup \{0\}$, $x \in \mathcal{V}$ and $\vec{b} \in D^l \cup \{0\}$, we define the *rigid substitution* $r[\vec{b}/x]$ of \vec{b} for x in r as above if $r \in D^{(l)}$, $\vec{b} \in D^l$ and $n_x(r) = \text{len}(\vec{b})$, and set $r[\vec{b}/x] := 0$ otherwise.

Lemma 1.5.4. *If $r \triangleleft e$ and $\vec{b} \triangleleft \vec{t}$ then $n_x(r) = n_x(e)$ and $\text{len}(\vec{b}) = \text{len}(\vec{t})$. Moreover $\partial_x e \cdot \vec{t} = \sum_{\sigma \in \mathfrak{S}_{\text{len}(\vec{b})}} \|r[[\sigma]\vec{b}/x]\|$.*

Proof. The first two identities follow directly from the definitions. If $n_x(r) \neq \text{len}(\vec{b})$ then both sides of the third identity are 0. Otherwise, it is proved by induction on r .

Let us treat the case of a monomial: write $r = (a_1, \dots, a_n)$ and $e = [s_1, \dots, s_n]$ with $a_i \triangleleft s_i$ for $1 \leq i \leq n$. Then

$$\begin{aligned} \partial_x e \cdot \vec{t} &= \sum_{(I_1, \dots, I_n) \text{ partition of } \{1, \dots, \text{len}(\vec{b})\}} [\partial_x s_1 \cdot \vec{t}_{I_1}, \dots, \partial_x s_n \cdot \vec{t}_{I_n}] \\ &= \sum_{(I_1, \dots, I_n) \in \mathcal{P}_{k_1, \dots, k_n}(\text{len}(\vec{b}))} [\partial_x s_1 \cdot \vec{t}_{I_1}, \dots, \partial_x s_n \cdot \vec{t}_{I_n}] \end{aligned}$$

where we write $k_i = n_x(s_i)$ for $1 \leq i \leq n$ and $\mathcal{P}_{k_1, \dots, k_n}(k)$ is the set of partitions (I_1, \dots, I_n) of $\{1, \dots, k\}$ such that $\text{Card}(I_i) = k_i$ for $1 \leq i \leq n$.

If $I \subseteq \{1, \dots, \text{len}(\vec{b})\}$ then we write $\vec{b}_I = (b_{i_1}, \dots, b_{i_k})$ where $i_1 < \dots < i_k$ enumerate I . By induction hypothesis we obtain

$$\begin{aligned} \partial_x e \cdot \vec{t} &= \sum_{(I_1, \dots, I_n) \in \mathcal{P}_{k_1, \dots, k_n}(\text{len}(\vec{b}))} \left[\sum_{\sigma_1 \in \mathfrak{S}_{k_1}} \|a_1[[\sigma_1]\vec{b}_{I_1}/x]\|, \dots, \sum_{\sigma_n \in \mathfrak{S}_{k_n}} \|a_n[[\sigma_n]\vec{b}_{I_n}/x]\| \right] \\ &= \sum_{(I_1, \dots, I_n) \in \mathcal{P}_{k_1, \dots, k_n}(\text{len}(\vec{b}))} \sum_{\sigma_1 \in \mathfrak{S}_{k_1}} \cdots \sum_{\sigma_n \in \mathfrak{S}_{k_n}} \|r[[\sigma_1]\vec{b}_{I_1} :: \cdots :: [\sigma_n]\vec{b}_{I_n}/x]\| \end{aligned}$$

and we conclude, observing that the families

$$\left([\sigma_1]\vec{b}_{I_1} :: \cdots :: [\sigma_n]\vec{b}_{I_n} \right)_{(I_1, \dots, I_n) \in \mathcal{P}_{k_1, \dots, k_n}(\text{len}(\vec{b})), (\sigma_1, \dots, \sigma_n) \in \mathfrak{S}_{k_1} \times \cdots \times \mathfrak{S}_{k_n}}$$

and $([\sigma]\vec{b})_{\sigma \in \mathfrak{S}_{\text{len}(\vec{b})}}$ coincide up to reindexing. \square

Informally, everything works out as if $[s_1, \dots, s_n] = \sum_{\sigma \in \mathfrak{S}_n} (s_1, \dots, s_n)$, which is to be related with the $\frac{1}{n!}$ coefficient in the Taylor expansion, cancelling out the cardinality of \mathfrak{S}_n .

1. Taylor Expansion for λ -Terms with Choice Operator – 1.5. The Groupoid of Permutations of Rigid Resource Terms

Forgetting about coefficients, we obtain:

Lemma 1.5.5. *If $n_x(e) = \text{len}(\bar{t})$ then, for any $r \triangleleft e$ and $\vec{b} \triangleleft \bar{t}$, $\text{supp}(\partial_x e \cdot \bar{t}) = \{\|r[[\sigma]\vec{b}/x]\| \mid \sigma \in \mathfrak{S}_{\text{len}(\vec{b})}\}$. Conversely, if $r' \triangleleft e' \in \text{supp}(\partial_x e \cdot \bar{t})$ then $n_x(e) = \text{len}(\bar{t})$ and there exist $r \triangleleft e$ and $\vec{b} \triangleleft \bar{t}$ such that $r' = r[\vec{b}/x]$.*

Proof. The first part follows directly from Lemma 1.5.4. We prove the second part by induction on the structure of e .

If $e = x$ then $\bar{t} = [t]$ for some $t \in \Delta_{\oplus}$ and $e' = t$. If $r' \triangleleft e' = t$ then we can set $r = x$ and $\vec{b} = (r')$. If $e = y \neq x$ then $\bar{t} = []$ and we can set $r = y$ and $\vec{b} = ()$. The abstraction and sum cases follow immediately from the induction hypothesis.

If $e = \langle s \rangle \bar{v}$, we write $\bar{t} = [t_1, \dots, t_n]$ and obtain

$$\partial_x e \cdot \bar{t} = \sum_{(I_0, I_1) \text{ partition of } \{1, \dots, n\}} \langle \partial_x s \cdot \bar{t}_{I_0} \rangle \partial_x \bar{v} \cdot \bar{t}_{I_1} \quad .$$

Then $e' = \langle s' \rangle \bar{v}'$ with $s' \in \text{supp}(\partial_x s \cdot \bar{t}_{I_0})$ and $\bar{v}' \in \text{supp}(\partial_x \bar{v} \cdot \bar{t}_{I_1})$ for some partition (I_0, I_1) of $\{1, \dots, n\}$. It follows that $r' = \langle a \rangle \vec{d}$ with $a \triangleleft s'$ and $\vec{d} \triangleleft \bar{v}'$. By induction hypothesis, we obtain $c_0 \triangleleft s$, $\vec{b}_0 \triangleleft \bar{t}_{I_0}$, $\vec{c}_1 \triangleleft \bar{v}$ and $\vec{b}_1 \triangleleft \bar{t}_{I_1}$ such that $a = c_0[\vec{b}_0/x]$ and $\vec{d} = \vec{c}_1[\vec{b}_1/x]$. Then we conclude by setting $r = \langle c_0 \rangle \vec{c}_1 \triangleleft \langle s \rangle \bar{v} = e$ and $\vec{b} = \vec{b}_0 :: \vec{b}_1 \triangleleft \bar{t}_{I_0} \cdot \bar{t}_{I_1} = \bar{t}$.

The case of monomials is similar. □

1.5.3. Substitution for Permutation Expressions

The key intermediate result for Step 3 is the fact that if $e \supset e'$ and $e' \in \text{supp}(\partial_x e \cdot \bar{t})$ then $(\partial_x e \cdot \bar{t})_{e'} = \frac{m(e)m(\bar{t})}{m(e')}$: this will be established in Lemma 1.5.18, which concludes the present section. With that goal in mind, and having characterized $m(e)$ as the cardinality of the group $\mathbb{G}(r)$ for any $r \triangleleft e$, it becomes essential to study how the permutation expressions acting on $r' \triangleleft e' \in \text{supp}(\partial_x e \cdot \bar{t})$ are related with those acting on some $r \triangleleft e$ and $\vec{b} \triangleleft \bar{t}$: by Lemma 1.5.5, we can choose r and \vec{b} such that $r' = r[\vec{b}/x]$. Then it seems natural to consider some form of substitution for permutation expressions, following the structure of rigid substitution.

We define the *substitution of permutation terms for a variable* as follows. Given $\epsilon \in \mathbb{G}(r, r')$ and $\vec{\beta} \in \mathbb{D}(\vec{b}, \vec{b}')$ with $\text{len}(\vec{b}) = n_x(r)$, we construct $\epsilon[\vec{\beta}/x]$ by induction

1. Taylor Expansion for λ -Terms with Choice Operator – 1.5. The Groupoid of Permutations of Rigid Resource Terms

on ϵ :

$$\begin{aligned}
(id_x)[(\beta)/x] &:= \beta \\
(id_y)[()/x] &:= id_y \\
(\lambda y.\alpha)[\vec{\beta}/x] &:= \lambda y.\alpha[\vec{\beta}/x] \\
(\alpha \oplus \bullet)[\vec{\beta}/x] &:= \alpha[\vec{\beta}/x] \oplus \bullet \\
(\bullet \oplus \alpha)[\vec{\beta}/x] &:= \bullet \oplus \alpha[\vec{\beta}/x] \\
(\langle \gamma \rangle \tilde{\delta})[\vec{\beta}_0 :: \vec{\beta}_1/x] &:= \langle \gamma[\vec{\beta}_0/x] \rangle \tilde{\delta}[\vec{\beta}_1/x] \\
(\sigma, (\alpha_1, \dots, \alpha_n))[\vec{\beta}_1 :: \dots :: \vec{\beta}_n/x] &:= (\sigma, (\alpha_1[\vec{\beta}_1/x], \dots, \alpha_n[\vec{\beta}_n/x]))
\end{aligned}$$

where we assume that $y \neq x$, $z \notin \{x\} \cup FV(\vec{\beta})$, $\text{len}(\vec{\beta}_0) = n_x(\gamma)$, $\text{len}(\vec{\beta}_1) = n_x(\delta)$, and $\text{len}(\vec{\beta}_i) = n_x(\alpha_i)$ for $1 \leq i \leq n$.

The action of $\epsilon[\vec{\beta}/x]$ on $r[\vec{b}/x]$ is quite intricate: in general, $\epsilon[\vec{\beta}/x] \notin \mathbb{G}(r[\vec{b}/x], r'[\vec{b}'/x])$.

Example 1.5.6. Consider the rigid monomials $\vec{a} = (x, x)$ and $\vec{b} = (\langle z \rangle(), \langle z \rangle(z))$. Writing τ for the unique transposition of \mathfrak{S}_2 , we obtain $\alpha = (\tau, id_x, id_x) \in \mathbb{G}(\vec{a})$. Let $\vec{\beta} = (id_{\langle z \rangle()}, id_{\langle z \rangle(z)}) \in \mathbb{D}(\vec{b})$. Then $\alpha[\vec{\beta}/x] = (\tau, id_{\langle z \rangle()}, id_{\langle z \rangle(z)})$, hence $\alpha[\vec{\beta}/x] : a[\vec{b}/x] \cong (\langle z \rangle(z), \langle z \rangle()) \neq a[\vec{b}/x]$.

To describe the image of $r[\vec{b}/x]$ through $\epsilon[\vec{\beta}/x]$, we first introduce two operations on permutations. If $\sigma \in \mathfrak{S}_n$, $\tau \in \mathfrak{S}_p$ and $\tau_1 \in \mathfrak{S}_{k_1}, \dots, \tau_n \in \mathfrak{S}_{k_n}$, we define the concatenation $\sigma \otimes \tau \in \mathfrak{S}_{n+p}$ and the multiplexing $\sigma \cdot (\tau_1, \dots, \tau_n) \in \mathfrak{S}_{k_1+\dots+k_n}$ by:

$$\begin{aligned}
(\sigma \otimes \tau)(i) &:= \sigma(i) & (\sigma \otimes \tau)(n+j) &:= n + \tau(j) \\
(\sigma \cdot (\tau_1, \dots, \tau_n)) \left(\sum_{j=1}^{i-1} k_j + l \right) &:= \sum_{j=1}^{\sigma(i)-1} k_{\sigma^{-1}(j)} + \tau_i(l)
\end{aligned}$$

for $1 \leq i \leq n$, $1 \leq j \leq p$ and $1 \leq l \leq k_i$.

The tensor product nomenclature is justified since, in the category \mathbb{P} of natural numbers and permutations, the concatenation of permutations defines a tensor product (which is the sum of natural numbers on objects). Multiplexing may be described in this category as follows: $\sigma \cdot (\tau_1, \dots, \tau_n) = \sigma_{k_1, \dots, k_n} \circ (\tau_1 \otimes \dots \otimes \tau_n)$ where σ_{k_1, \dots, k_n} is the canonical symmetry map $k_1 + \dots + k_n \rightarrow k_{\sigma^{-1}(1)} + \dots + k_{\sigma^{-1}(n)} = [\sigma](k_1 + \dots + k_n)$ (see Figure 1.7).

Multiplexed permutations compose as follows:

Lemma 1.5.7. If $\sigma, \sigma' \in \mathfrak{S}_n$, $\tau_i \in \mathfrak{S}_{k_i}$ and $\tau'_i \in \mathfrak{S}_{k_{\sigma^{-1}(i)}}$ for $1 \leq i \leq n$, then

$$(\sigma' \cdot (\tau'_1, \dots, \tau'_n)) (\sigma \cdot (\tau_1, \dots, \tau_n)) = (\sigma' \sigma) \cdot (\tau'_{\sigma(1)} \tau_1, \dots, \tau'_{\sigma(n)} \tau_n)$$

and

$$(\sigma \cdot (\tau_1, \dots, \tau_n))^{-1} = \sigma^{-1} \cdot (\tau_{\sigma^{-1}(1)}^{-1}, \dots, \tau_{\sigma^{-1}(n)}^{-1}) \quad .$$

1. Taylor Expansion for λ -Terms with Choice Operator – 1.5. The Groupoid of Permutations of Rigid Resource Terms

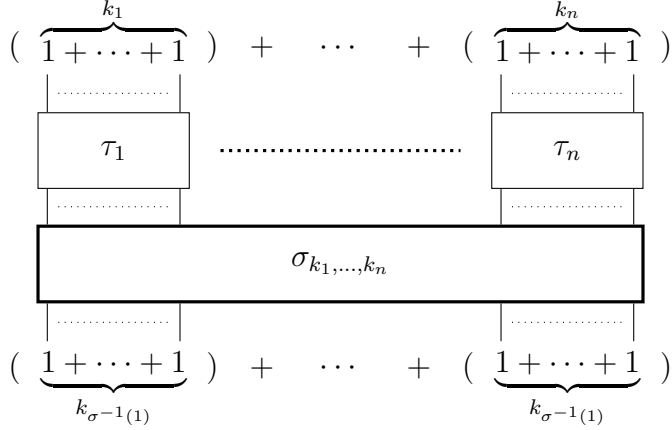


Figure 1.7. – Graphical representation of $\sigma \cdot (\tau_1, \dots, \tau_n)$

Proof. We detail the proof only in case the result is not obvious to the reader from the above categorical presentation of multiplexing. Let $\alpha = \sigma \cdot (\tau_1, \dots, \tau_n)$ and $\alpha' = \sigma' \cdot (\tau'_1, \dots, \tau'_n)$. For $1 \leq i \leq n$ and $1 \leq l \leq k_i$:

$$\begin{aligned}
 \alpha' \left(\alpha \left(\sum_{j=1}^{i-1} k_j + l \right) \right) &= \alpha' \left(\sum_{j=1}^{\sigma(i)-1} k_{\sigma^{-1}(j)} + \tau_i(l) \right) \\
 &= \sum_{j=1}^{\sigma'(\sigma(i))-1} k'_{\sigma'^{-1}(j)} + \tau'_{\sigma'(i)}(\tau_i(l)) \quad (\text{writing } k'_i = k_{\sigma^{-1}(i)}) \\
 &= \sum_{j=1}^{(\sigma'\sigma)(i)-1} k_{(\sigma'\sigma)^{-1}(j)} + (\tau'_{\sigma'(i)}\tau_i)(l)
 \end{aligned}$$

which establishes the first identity. The second identity follows directly. \square

The action of multiplexed permutations on sequences is as follows:

Lemma 1.5.8. *Let $\vec{b}, \vec{b}_1, \dots, \vec{b}_n \in D^l$, $\sigma \in \mathfrak{S}_n$ and $\tau_i \in \mathfrak{S}_{\text{len}(\vec{b}_i)}$ for all $i \in \{1, \dots, n\}$. If $\vec{b} = \vec{b}_1 \ :: \ \dots \ :: \ \vec{b}_n$ then $[\sigma \cdot (\tau_1, \dots, \tau_n)]\vec{b} = [\tau_{\sigma^{-1}(1)}]\vec{b}_{\sigma^{-1}(1)} \ :: \ \dots \ :: \ [\tau_{\sigma^{-1}(n)}]\vec{b}_{\sigma^{-1}(n)}$.*

Proof. Again, we detail the proof only in case the result is not obvious from the categorical presentation. Set $\text{len}(\vec{b}_i) = k_i$, so that $\text{len}(\vec{b}) = \sum_{i=1}^n k_i$. Write $\vec{b}' = [\sigma \cdot (\tau_1, \dots, \tau_n)]\vec{b}$. For $1 \leq p \leq \text{len}(\vec{b}') = \text{len}(\vec{b}) = \sum_{j=1}^n k_{\sigma^{-1}(j)}$, we can write $p = \sum_{j=1}^{i-1} k_{\sigma^{-1}(j)} + l$ with $i \in \{1, \dots, n\}$ and $l \in \{1, \dots, k_{\sigma^{-1}(i)}\}$. Then, by Lemma 1.5.7, $(\sigma \cdot (\tau_1, \dots, \tau_n))^{-1}(p) = \sum_{j=1}^{\sigma^{-1}(i)-1} k_j + \tau_{\sigma^{-1}(i)}^{-1}(l)$ and $b'_p = b_{(\sigma \cdot (\tau_1, \dots, \tau_n))^{-1}(p)} = (\vec{b}_{\sigma^{-1}(i)})_{\tau_{\sigma^{-1}(i)}^{-1}(l)} = ([\tau_{\sigma^{-1}(i)}]\vec{b}_{\sigma^{-1}(i)})_l$. \square

1. Taylor Expansion for λ -Terms with Choice Operator – 1.5. The Groupoid of Permutations of Rigid Resource Terms

We can now define the *restriction* $\epsilon_{|x} \in \mathfrak{S}_{n_x(r)}$ of $\epsilon \in \mathbb{G}(r, r')$ to the occurrences of x in r , by induction on ϵ :

$$\begin{aligned} (id_x)_{|x} &:= id_{\{1\}} \\ (id_y)_{|x} &:= id_{\emptyset} \\ \left. \begin{aligned} (\lambda y.\alpha)_{|x} \\ (\alpha \oplus \bullet)_{|x} \\ (\bullet \oplus \alpha)_{|x} \end{aligned} \right\} &:= \alpha_{|x} \\ ((\gamma)\tilde{\delta})_{|x} &:= \gamma_{|x} \otimes \tilde{\delta}_{|x} \\ (\sigma, \alpha_1, \dots, \alpha_n)_{|x} &:= \sigma \cdot (\alpha_{1|x}, \dots, \alpha_{n|x}) \end{aligned}$$

where we assume $x \neq y$. Intuitively $\epsilon_{|x}$ is the permutation induced by ϵ on the occurrences $x_1, \dots, x_{n_x(r)}$ of x in r , taken from left to right.

We recall that \mathbb{P} denotes the category of finite cardinals and permutations. For any $x \in \mathcal{V}$, we define an application F_x from \mathbb{G} to \mathbb{P} as follows: $F_x(r) := n_x(r)$ and $F_x(\alpha) := \alpha_{|x}$.

Lemma 1.5.9. F_x is a functor from \mathbb{G} to \mathbb{P} .

Proof. By induction on permutation expressions. We focus on the composition condition for the list case. Let $\tilde{\alpha} : \vec{a} = (a_1, \dots, a_n) \cong \vec{b} = (b_1, \dots, b_n)$ and $\tilde{\beta} : \vec{b} \cong \vec{c} = (c_1, \dots, c_n)$. By definition $\tilde{\alpha} = (\sigma, \alpha_1, \dots, \alpha_n)$ and $\tilde{\beta} = (\tau, \beta_1, \dots, \beta_n)$, for some σ, τ in \mathfrak{S}_n and with $\alpha_i : a_i \cong b_{\sigma(i)}$ and $\beta_i : b_i \cong c_{\tau(i)}$. The composition $\tilde{\beta}\tilde{\alpha}$ is then defined as the isomorphism $(\tau\sigma, \beta_{\sigma(1)}\alpha_1, \dots, \beta_{\sigma(n)}\alpha_n)$.

We have to prove that $(\tilde{\beta}\tilde{\alpha})_{|x} = \tilde{\beta}_{|x}\tilde{\alpha}_{|x}$, that is

$$(\tau\sigma) \cdot ((\beta_{\sigma(1)}\alpha_1)_{|x}, \dots, (\beta_{\sigma(n)}\alpha_n)_{|x}) = (\tau \cdot (\beta_{1|x}, \dots, \beta_{n|x}))(\sigma \cdot (\alpha_{1|x}, \dots, \alpha_{n|x}))$$

which is a direct consequence of the inductive hypothesis, $(\beta_{\sigma(i)}\alpha_i)_{|x} = \beta_{\sigma(i)|x}\alpha_{i|x}$ for $1 \leq i \leq n$, via Lemma 1.5.7. \square

Observe in particular that $(\epsilon_{|x})^{-1} = (\epsilon^{-1})_{|x}$, so that we may simply write $\epsilon_{|x}^{-1}$ in the following.

We can now describe the action of $\epsilon[\vec{\beta}/x]$ on $r[\vec{b}/x]$ as follows:

Lemma 1.5.10. If $\epsilon : r \cong r'$ and $\vec{\beta} \in \vec{\mathbb{D}}(\vec{b}, \vec{b}')$ with $\text{len}(\vec{\beta}) = n_x(r)$ then $\epsilon[\vec{\beta}/x] : r[\vec{b}/x] \cong r'[[\epsilon_{|x}]\vec{b}'/x]$.

Proof. By induction on the structure of r . The interesting case is the list case. Assume $r = (a_1, \dots, a_n)$, $r' = (a'_1, \dots, a'_n)$, $\epsilon = (\sigma, \alpha_1, \dots, \alpha_n)$ and $\vec{\beta} = \vec{\beta}_1 :: \dots :: \vec{\beta}_n$, with $\alpha_i : a_i \cong a'_{\sigma(i)}$, $\vec{b} = \vec{b}_1 :: \dots :: \vec{b}_n$, $\vec{b}' = \vec{b}'_1 :: \dots :: \vec{b}'_n$, $\text{len}(\vec{\beta}_i) = n_x(a_i)$ and $\vec{\beta}_i \in \vec{\mathbb{D}}(\vec{b}_i, \vec{b}'_i)$. By definition, we have $\alpha[\vec{\beta}/x] = (\sigma, \alpha_1[\vec{\beta}_1/x], \dots, \alpha_n[\vec{\beta}_n/x])$. Since $\alpha_i : a_i \cong a'_{\sigma(i)}$, we obtain $\alpha_i[\vec{\beta}_i/x] : a_i[\vec{b}_i/x] \cong a'_{\sigma(i)}[[\alpha_{i|x}]\vec{b}'_i/x]$ by induction hypothesis.

1. Taylor Expansion for λ -Terms with Choice Operator – 1.5. The Groupoid of Permutations of Rigid Resource Terms

We obtain

$$\begin{aligned}
\alpha[\vec{\beta}/x] : r[\vec{b}/x] &\cong [\sigma] \left([\alpha_1[\vec{\beta}_1/x]](a_1[\vec{b}_1/x]), \dots, [\alpha_n[\vec{\beta}_n/x]](a_n[\vec{b}_n/x]) \right) \\
&= [\sigma] \left(a'_{\sigma(1)}[\alpha_{1|x}[\vec{b}'_1/x]], \dots, a'_{\sigma(n)}[\alpha_{n|x}[\vec{b}'_n/x]] \right) \\
&= \left(a'_1[\alpha_{\sigma^{-1}(1)|x}[\vec{b}'_{\sigma^{-1}(1)}/x]], \dots, a'_n[\alpha_{\sigma^{-1}(n)|x}[\vec{b}'_{\sigma^{-1}(n)}/x]] \right) \\
&= r' \left[[\alpha_{\sigma^{-1}(1)|x}[\vec{b}'_{\sigma^{-1}(1)}] \vdash \dots \vdash [\alpha_{\sigma^{-1}(n)|x}[\vec{b}'_{\sigma^{-1}(n)}]/x] \right]
\end{aligned}$$

and we conclude by Lemma 1.5.8. \square

1.5.4. The Combinatorics of Permutation Expressions under Coherent Substitution

Substitution is injective on parallel permutation expressions, in the following sense:

Lemma 1.5.11. *Let $r, r' \in D^{(l)}$ and $\vec{b}, \vec{b}' \in D^l$ with $\text{len}(\vec{b}) = n_x(r)$ and $\text{len}(\vec{b}') = n_x(r')$, and let $\epsilon, \epsilon' \in \mathbb{G}(r, r')$ and $\vec{\beta}, \vec{\beta}' \in \mathbb{D}(\vec{b}, \vec{b}')$. If $\epsilon[\vec{\beta}/x] = \epsilon'[\vec{\beta}'/x]$ then $\epsilon = \epsilon'$ and $\vec{\beta} = \vec{\beta}'$.*

Proof. By a straightforward induction on the structure of r . \square

On the other hand, surjectivity does not hold in general, because the substitution might enable new morphisms $r[\vec{b}/x] \cong r'[\vec{b}'/x]$, not induced by morphisms in $\mathbb{G}(r, r')$ and $\mathbb{D}(\vec{b}, \vec{b}')$:

Example 1.5.12. *Let $a = \langle\langle y \rangle(x) \rangle\langle z \rangle(x)$, $a' = \langle\langle x \rangle(y) \rangle\langle z \rangle(x)$ and $\vec{b} = (y, z)$. Then $a[\vec{b}/x] = a'[\vec{b}/x]$ but $a \not\cong a'$.*

Observe that, in the above example, $\|a\| \not\cong \|a'\|$. Indeed, in the following, we will establish that coherence allows to restore a precise correspondence between the permutation expressions acting on a substitution $r[(b_1, \dots, b_n)/x]$ and the $(1+n)$ -tuples of permutation expressions acting respectively on r and each of the b_i 's. It will be useful to consider the coherence relation defined on rigid expressions by the rules of Figure 1.8, so that $r \circ r'$ iff $\|r\| \circ \|r'\|$.

$$\begin{array}{c}
\frac{}{x \circ x} \quad \frac{a \circ a'}{\lambda x. a \circ \lambda x. a'} \quad \frac{c \circ c' \quad \vec{d} \circ \vec{d}'}{\langle c \rangle \vec{d} \circ \langle c' \rangle \vec{d}'} \quad \frac{b_i \circ b_j \text{ for } 1 \leq i, j \leq n+m}{(b_1, \dots, b_n) \circ (b_{n+1}, \dots, b_{n+m})} \\
\frac{a \circ a'}{a \oplus \bullet \circ a' \oplus \bullet} \quad \frac{a \circ a'}{\bullet \oplus s \circ \bullet \oplus s'} \quad \frac{}{a \oplus \bullet \circ \bullet \oplus a'} \quad \cdot
\end{array}$$

Figure 1.8. – Rules for the coherence relation on $D^{(l)}$.

1. Taylor Expansion for λ -Terms with Choice Operator – 1.5. The Groupoid of Permutations of Rigid Resource Terms

Then we obtain:

Lemma 1.5.13. *Let $r, r' \in D^{(l)}$ and $\vec{b}, \vec{b}' \in D^l$ with $\text{len}(\vec{b}) = n_x(r)$ and $\text{len}(\vec{b}') = n_x(r')$. If $r \supset r'$ then for all $\phi \in \mathbb{G}(r[\vec{b}/x], r'[\vec{b}'/x])$ there exist $\epsilon \in \mathbb{G}(r, r')$ and $\vec{\beta} \in \mathbb{D}(\vec{b}, [\epsilon_{|x}^{-1} \vec{b}'])$ such that $\phi = \epsilon[\vec{\beta}/x]$.*

Proof. By induction on the structure of r : the coherence hypothesis $r \supset r'$ induces that r and r' are of the same syntactic nature.

If $r = x$ then $r' = x$ and we can write $\vec{b} = (b)$, $\vec{b}' = (b')$ with $\phi : b \cong b'$. Then we set $\epsilon = 1_x$ and $\vec{\beta} = (\beta)$. If $r = y \neq x$ then $r' = y$ and $\phi = 1_y$, and we set $\epsilon = 1_y$ and $\vec{\beta} = ()$. The abstraction and application cases follow straightforwardly from the induction hypotheses. We detail the list case.

We have $r = (a_1, \dots, a_n)$ and $r' = (a'_1, \dots, a'_m)$. Since $\phi : r[\vec{b}/x] \cong r'[\vec{b}'/x]$ we must have $m = n$, $\vec{b} = \vec{b}_1 :: \dots :: \vec{b}_n$, $\vec{b}' = \vec{b}'_1 :: \dots :: \vec{b}'_n$ and $\phi = (\sigma, \gamma_1, \dots, \gamma_n)$ with $\gamma_i \in \mathbb{G}(a_i[\vec{b}_i/x], a'_{\sigma(i)}[\vec{b}'_{\sigma(i)}/x])$. Since $r \supset r'$ we have in particular $a_i \supset a'_{\sigma(i)}$ for $1 \leq i \leq n$.

By the induction hypothesis, we obtain $\gamma_i = \alpha_i[\vec{\beta}_i/x]$ with $\alpha_i \in \mathbb{G}(a_i, a'_{\sigma(i)})$ and $\vec{\beta}_i \in \mathbb{D}(\vec{b}_i, [\alpha_{i|_x}^{-1} \vec{b}'_{\sigma(i)}])$. Then by definition $\epsilon := (\sigma, \alpha_1, \dots, \alpha_n) : r \cong r'$ and

$$\begin{aligned} \vec{\beta} &:= \vec{\beta}_1 :: \dots :: \vec{\beta}_n : \vec{b} \cong [\alpha_{1|x}^{-1} \vec{b}'_{\sigma(1)}] :: \dots :: [\alpha_{n|x}^{-1} \vec{b}'_{\sigma(n)}] \\ &= [\sigma^{-1} \cdot (\alpha_{\sigma^{-1}(1)|_x}^{-1}, \dots, \alpha_{\sigma^{-1}(n)|_x}^{-1})] \vec{b}' \quad (\text{by Lemma 1.5.8}) \end{aligned}$$

and it remains only to prove that $\sigma^{-1} \cdot (\alpha_{\sigma^{-1}(1)|_x}^{-1}, \dots, \alpha_{\sigma^{-1}(n)|_x}^{-1}) = \epsilon_{|x}^{-1}$, which follows from Lemma 1.5.7. \square

In particular, we obtain $(\epsilon_{|x}, \vec{\beta}) \in \mathbb{G}(\vec{b}, \vec{b}')$, hence:

Corollary 1.5.14. *If $r \supset r'$ and $r[\vec{b}/x] \cong r'[\vec{b}'/x]$ then $r \cong r'$ and $\vec{b} \cong \vec{b}'$.*

Given $r \triangleleft e$, $\vec{b} \triangleleft \vec{t}$ and $e' \in \text{supp}(\partial_x e \cdot \vec{t})$ such that $r[\vec{b}/x] \triangleleft e'$, we are about to determine the coefficient of e' in $\partial_x e \cdot \vec{t}$ by enumerating the permutations σ such that $r[[\sigma]\vec{b}/x] \triangleleft e'$, i.e. $r[[\sigma]\vec{b}/x] \cong r[\vec{b}/x]$. We thus define $\mathcal{S}_x(r, \vec{b}) := \{\sigma \in \mathfrak{S}_{n_x(r)} \mid r[\vec{b}/x] \cong r[[\sigma]\vec{b}/x]\}$ whenever $\text{len}(\vec{b}) = n_x(r)$.

Lemma 1.5.15. *Let $r \in D^{(l)}$ and $\vec{b} \in D^l$ with $\text{len}(\vec{b}) = n_x(r)$. If $r \supset r$ then $\mathcal{S}_x(r, \vec{b}) = \mathbb{G}(r)_{|x} S(\vec{b})$.*

Proof. Let $\tau \in S(\vec{b})$: by definition, we obtain $\vec{\beta} \in \mathbb{D}(\vec{b}, [\tau]\vec{b})$. If moreover $\epsilon \in \mathbb{G}(r)$ then, by Lemma 1.5.10, $\epsilon[\vec{\beta}/x] \in \mathbb{G}(r[\vec{b}/x], r[[\epsilon_{|x}\tau]\vec{b}/x])$ hence $\epsilon_{|x}\tau \in \mathcal{S}_x(r, \vec{b})$. It remains only to show that the function $(\epsilon, \tau) \in \mathbb{G}(r) \times S(\vec{b}) \mapsto \epsilon_{|x}\tau \in \mathcal{S}_x(r, \vec{b})$ is surjective.

If $\sigma \in \mathcal{S}_x(r, \vec{b})$, there exists $\phi \in \mathbb{G}(r[\vec{b}/x], r[[\sigma]\vec{b}/x])$. Since $r \supset r$, we can apply Lemma 1.5.13 and obtain $\epsilon \in \mathbb{G}(r)$ and $\vec{\beta} \in \mathbb{D}(\vec{b}, [\epsilon_{|x}^{-1}\sigma]\vec{b})$: in particular, $\epsilon_{|x}^{-1}\sigma \in S(\vec{b})$, and we conclude since $\sigma = \epsilon_{|x}(\epsilon_{|x}^{-1}\sigma)$ by Lemma 1.5.9. \square

1. Taylor Expansion for λ -Terms with Choice Operator – 1.5. The Groupoid of Permutations of Rigid Resource Terms

Our argument will moreover rely on the following construction: if $\text{len}(\vec{b}) = n_x(r)$, we set $\mathbb{I}_x(r, \vec{b}) := \{\epsilon \in \mathbb{G}(r) \mid \epsilon|_x \in S(\vec{b})\} = F_x^{-1}(S(\vec{b}))$, which is a subgroup of $\mathbb{G}(r)$ because F_x is a group homomorphism from $\mathbb{G}(r)$ to $\mathfrak{S}_{n_x(r)}$ by Lemma 1.5.9.

Lemma 1.5.16. *Let $r \in D^{(!)}$ and $\vec{b} \in D^!$ with $\text{len}(\vec{b}) = n_x(r)$. If $r \succ r$ then $\text{Card}(\mathbb{G}(r[\vec{b}/x])) = \text{Card}(\mathbb{I}_x(r, \vec{b}))\text{Card}(\vec{\mathbb{D}}(\vec{b}))$.*

Proof. By Lemma 1.5.10, if $\epsilon \in \mathbb{G}(r)$ and $\vec{\beta} \in \vec{\mathbb{D}}(\vec{b}, [\epsilon|_x^{-1}\vec{b}])$ then $\epsilon[\vec{\beta}/x] \in \mathbb{G}(r[\vec{b}/x])$. If moreover $\epsilon \in \mathbb{I}_x(r, \vec{b})$ then $\epsilon|_x^{-1} \in S(\vec{b})$: as already remarked in the proof of Lemma 1.5.2, this entails that $\text{Card}(\vec{\mathbb{D}}(\vec{b}, [\epsilon|_x^{-1}\vec{b}])) = \text{Card}(\vec{\mathbb{D}}(\vec{b}))$. It is thus sufficient to establish that the substitution operation $(\epsilon, \vec{\beta}) \mapsto \epsilon[\vec{\beta}/x]$ defines a bijection from $\sum_{\epsilon \in \mathbb{I}_x(r, \vec{b})} \vec{\mathbb{D}}(\vec{b}, [\epsilon|_x^{-1}\vec{b}])$ to $\mathbb{G}(r[\vec{b}/x])$. This fact derives immediately from Lemma 1.5.11 (injectivity) and Lemma 1.5.13 (surjectivity). \square

Lemma 1.5.17. *Let $r \in D^{(!)}$ and $\vec{b} \in D^!$ with $r \succ r$ and $\text{len}(\vec{b}) = n_x(r)$. Then*

$$\text{Card}(\mathcal{S}_x(r, \vec{b})) = \frac{\text{Card}(\mathbb{G}(r))\text{Card}(\mathbb{G}(\vec{b}))}{\text{Card}(\mathbb{G}(r[\vec{b}/x]))}.$$

Proof. Write $k = n_x(r)$. We know that $S(\vec{b})$ and $\mathbb{G}(r)|_x$ are subgroups of \mathfrak{S}_k . Lemma 1.5.15 and Fact 1.2.2 entail that

$$\text{Card}(\mathcal{S}_x(r, \vec{b})) = \frac{\text{Card}(\mathbb{G}(r)|_x)\text{Card}(S(\vec{b}))}{\text{Card}(\mathbb{G}(r)|_x \cap S(\vec{b}))}.$$

Using Lemma 1.5.16, it will thus be sufficient to prove:

$$\frac{\text{Card}(\mathbb{G}(r))\text{Card}(\mathbb{G}(\vec{b}))}{\text{Card}(\mathbb{I}_x(r, \vec{b}))\text{Card}(\vec{\mathbb{D}}(\vec{b}))} = \frac{\text{Card}(\mathbb{G}(r)|_x)\text{Card}(S(\vec{b}))}{\text{Card}(\mathbb{G}(r)|_x \cap S(\vec{b}))}$$

which simplifies to

$$\frac{\text{Card}(\mathbb{G}(r))}{\text{Card}(\mathbb{I}_x(r, \vec{b}))} = \frac{\text{Card}(\mathbb{G}(r)|_x)}{\text{Card}(\mathbb{G}(r)|_x \cap S(\vec{b}))}$$

by Lemma 1.5.2. We conclude by Fact 1.2.3. \square

Lemma 1.5.18. *Let $e \in \Delta_{\oplus}^{(!)}$ be such that $e \succ e$ and let $\bar{t} \in \Delta_{\oplus}^!$. If $e' \in \text{supp}(\partial_x e \cdot \bar{t})$ then $(\partial_x e \cdot \bar{t})_{e'} = \frac{m(e)m(\bar{t})}{m(e')}$.*

Proof. Let $r' \triangleleft e'$ and $k = n_x(r)$. By Lemma 1.5.5 there exists $r \triangleleft e$ and $\vec{b} \triangleleft \bar{t}$ such that $r' = r[\vec{b}/x]$. Then, by Lemma 1.5.4, $(\partial_x e \cdot \bar{t})_{e'} = \text{Card}(\{\sigma \in \mathfrak{S}_k \mid r[[\sigma]\vec{b}/x] \triangleleft e'\})$. By Lemma 1.5.1, we have $r[[\sigma]\vec{b}/x] \triangleleft e'$ iff $r[\vec{b}/x] \cong r[[\sigma]\vec{b}/x]$, hence $(\partial_x e \cdot \bar{t})_{e'} = \text{Card}(\mathcal{S}_x(r, \vec{b}))$. Then we conclude by Lemmas 1.5.17 and 1.5.3. \square

1. Taylor Expansion for λ -Terms with Choice Operator – 1.6. Normalizing the Taylor Expansion

$$\begin{array}{c}
\frac{}{\langle \lambda x.a \rangle \vec{b} \rightarrow_r a[\vec{b}/x]} \quad \frac{}{\langle a \oplus \bullet \rangle \vec{b} \rightarrow_r \langle a \rangle \vec{b} \oplus \bullet} \quad \frac{}{\langle \bullet \oplus a \rangle \vec{b} \rightarrow_r \bullet \oplus \langle a \rangle \vec{b}} \\
\frac{}{\lambda x.(a \oplus \bullet) \rightarrow_r \lambda x.a \oplus \bullet} \quad \frac{}{\lambda x.(\bullet \oplus a) \rightarrow_r \bullet \oplus \lambda x.a} \\
\frac{a \rightarrow_r a'}{\lambda x.a \rightarrow_r \lambda x.a'} \quad \frac{a \rightarrow_r a'}{\langle a \rangle \vec{b} \rightarrow_r \langle a' \rangle \vec{b}} \quad \frac{\vec{a} \rightarrow_r \vec{a}'}{\langle b \rangle \vec{a} \rightarrow_r \langle b \rangle \vec{a}'} \\
\frac{a \rightarrow_r a'}{a \oplus b \rightarrow_r a' \oplus b} \quad \frac{a \rightarrow_r a'}{b \oplus a \rightarrow_r b \oplus a'} \quad \frac{a \rightarrow_r a'}{a :: \vec{b} \rightarrow_r a' :: \vec{b}} \quad \frac{\vec{a} \rightarrow_r \vec{a}'}{b :: \vec{a} \rightarrow_r b :: \vec{a}'}
\end{array}$$

Figure 1.9. – Reduction rules of the rigid resource calculus

1.6. Normalizing the Taylor Expansion

In this final section we leverage our results on the groupoid of rigid expressions and permutation expressions in order to achieve Steps 3 and 4. This allows us to complete the proof of commutation between Taylor expansion and normalization.

1.6.1. Normalizing Resource Expressions in a Uniform Setting

Lemma 1.5.18 is almost sufficient to obtain Step 3, as it fixes the coefficients in a hereditary head reduction step from a uniform expression:

Lemma 1.6.1. *Let $e \in \Delta_{\oplus}^{(1)}$ with $e \circ e$. If $e' \in \text{supp}(L_{\partial}(e))$ then $(L_{\partial}(e))_{e'} = \frac{m(e)}{m(e')}$.*

Proof. By induction on the structure of e applying Lemma 1.5.18 in the redex case: observe indeed that if $e = \lambda \vec{x}. \langle \lambda y.s \rangle \bar{t} \bar{u}_1 \cdots \bar{u}_k$ then $e' = \lambda \vec{x}. \langle v \rangle \bar{u}_1 \cdots \bar{u}_k$ with $v \in \text{supp}(\partial_y s \cdot \bar{t})$, and then $(L_{\partial}(e))_{e'} = (\partial_y s \cdot \bar{t})_v = \frac{m(s)m(\bar{t})}{m(v)}$ and we conclude since $\frac{m(e)}{m(e')} = \frac{m(s)m(\bar{t})}{m(v)}$. All the other cases follow directly from the induction hypothesis by multilinearity. \square

To iterate Lemma 1.6.1 along the reduction sequence to the normal form, it only remains to show that uniformity is preserved by L_{∂} . As before, we prefer to focus on the rigid setting first.

The reduction of the rigid resource calculus is the relation from rigid expressions to partial rigid expressions induced by the rules of Figure 1.9: these rules mimic those for the resource calculus. Considered as a binary relation on partial expressions, \rightarrow_r is again confluent and strongly normalizing. We write $\text{nf}(r)$ for the unique normal form of r , that is a normal rigid expression or 0. Again, we will only consider the

1. Taylor Expansion for λ -Terms with Choice Operator – 1.6. Normalizing the Taylor Expansion

hereditary head reduction strategy defined as follows:

$$\begin{aligned}
L_r(a \oplus \bullet) &:= L_r(a) \oplus \bullet & L_r(\bullet \oplus a) &:= \bullet \oplus L_r(a) \\
L_r(\lambda \vec{x}. \lambda y. (a \oplus \bullet)) &:= \lambda \vec{x}. (\lambda y. a \oplus \bullet) & L_r(\lambda \vec{x}. \lambda y. (\bullet \oplus a)) &:= \lambda \vec{x}. (\bullet \oplus \lambda y. a) \\
L_r(\lambda \vec{x}. \langle \langle a \oplus \bullet \rangle \vec{b} \rangle \vec{c}_1 \cdots \vec{c}_k) &:= \lambda \vec{x}. \langle \langle a \rangle \vec{b} \oplus \bullet \rangle \vec{c}_1 \cdots \vec{c}_k \\
L_r(\lambda \vec{x}. \langle \langle \bullet \oplus a \rangle \vec{b} \rangle \vec{c}_1 \cdots \vec{c}_k) &:= \lambda \vec{x}. \langle \bullet \oplus \langle a \rangle \vec{b} \rangle \vec{c}_1 \cdots \vec{c}_k \\
L_r(\lambda \vec{x}. \langle y \rangle \vec{a}_1 \cdots \vec{a}_k) &:= \lambda \vec{x}. \langle y \rangle L_r(\vec{a}_1) \cdots L_r(\vec{a}_k) \\
L_r((a_1, \dots, a_k)) &:= (L_r(a_1), \dots, L_r(a_k)) \\
L_r(\lambda \vec{x}. \langle \lambda y. a \rangle \vec{b} \vec{c}_1 \cdots \vec{c}_k) &:= \lambda \vec{x}. \langle a[\vec{b}/y] \rangle \vec{c}_1 \cdots \vec{c}_k
\end{aligned}$$

extended to partial rigid expressions by setting $L_r(0) := 0$. For any $r \in D^{(l)}$, there exists $k \in \mathbb{N}$ such that $\text{nf}(r) = L_r^k(r)$. Moreover, r is in normal form iff $L_r(r) = r$.

Lemma 1.6.2. *If $e \in \Delta_{\oplus}^{(l)}$ then:*

1. $\text{supp}(L_{\partial}(e)) = \{\|L_r(r)\| \mid r \triangleleft e \text{ and } L_r(r) \neq 0\}$;
2. $\text{supp}(\text{nf}(e)) = \{\|\text{nf}(r)\| \mid r \triangleleft e \text{ and } \text{nf}(r) \neq 0\}$.

Proof. We first prove that $r' \triangleleft e' \in \text{supp}(L_{\partial}(e))$ iff there exists $r \triangleleft e$ with $r' = L_r(r)$, which gives the first result: this is done by a straightforward induction on the structure of e , using Lemma 1.5.5 for the β -redex case.

Now fix $k \in \mathbb{N}$ such that $\text{nf}(e) = L_{\partial}^k(e)$: by iterating the previous result, we obtain $r' \triangleleft e' \in \text{supp}(\text{nf}(e))$ iff there exists $r \triangleleft e$ with $r' = L_r^k(r)$. Then we conclude, observing that if $r' \triangleleft e'$, then r' is in normal form iff e' is. \square

Lemma 1.6.3. *If $r \supset r'$ and $\vec{b} \supset \vec{b}'$ with $n_x(r) = \text{len}(\vec{b})$ and $n_x(r') = \text{len}(\vec{b}')$ then $r[\vec{b}/x] \supset r'[\vec{b}'/x]$.*

Proof. By a straightforward induction on r . \square

Lemma 1.6.4. *For all $r, r' \in D^{(l)}$ such that $r \supset r'$:*

1. if $L_r(r) \neq 0$ and $L_r(r') \neq 0$ then $L_r(r) \supset L_r(r')$;
2. if $\text{nf}(r) \neq 0$ and $\text{nf}(r') \neq 0$ then $\text{nf}(r) \supset \text{nf}(r')$.

Proof. The first item is easily established by induction on r , using Lemma 1.6.3 in the case of a β -redex. Having fixed k such that both $\text{nf}(r) = L_r^k(r)$ and $\text{nf}(r') = L_r^k(r')$, the second item follows by iterating the first one. \square

Lemma 1.6.5. *For all $e, e' \in \Delta_{\oplus}^{(l)}$ such that $e \supset e'$, both $\text{supp}(L_{\partial}(e)) \cup \text{supp}(L_{\partial}(e'))$ and $\text{supp}(\text{nf}(e)) \cup \text{supp}(\text{nf}(e'))$ are cliques.*

Proof. This is a direct consequence of Lemma 1.6.4, via Lemma 1.6.2. \square

We can finally establish:

1. Taylor Expansion for λ -Terms with Choice Operator – 1.6. Normalizing the Taylor Expansion

Theorem 1.6.6 (Step 3). *Let $e \in \Delta_{\oplus}^{(1)}$ with $e \supset e$ and let $e' \in \text{supp}(NF(e))$. Then*

$$(NF(e))_{e'} = \frac{m(e)}{m(e')} .$$

Proof. Thanks to Lemma 1.6.5, we can iterate Lemma 1.6.1 on any sequence e_0, \dots, e_n such that $e_0 \supset e_0$ and $e_i \in \text{supp}(L_{\partial}(e_{i-1}))$ for $1 \leq i \leq n$. \square

Lemma 1.6.5 entails that the normal form of a clique of expressions is also a clique: Step 4 amounts to the fact that distinct elements in a clique have disjoint normal forms. In other words, if the normal forms of two coherent expressions intersect on a common element, then they must coincide:

Lemma 1.6.7. *For all $r, r' \in D^{(1)}$ such that $r \supset r'$:*

1. *if $L_r(r) \cong L_r(r')$ then $r \cong r'$;*
2. *if $\text{nf}(r) \cong \text{nf}(r')$ then $r \cong r'$.*

Proof. Observe that \cong is defined on rigid expressions only so that if, e.g., $L_r(r) \cong L_r(r')$ then in particular $L_r(r) \neq 0 \neq L_r(r')$. The first item is established by induction on r , using Corollary 1.5.14 in the case of a β -redex. Having fixed k such that both $\text{nf}(r) = L_r^k(r)$ and $\text{nf}(r') = L_r^k(r')$, the second item follows by iterating the first one, thanks to Lemma 1.6.4. \square

Theorem 1.6.8 (Step 4). *Let $e, e' \in \Delta_{\oplus}^{(1)}$ be such that $e \supset e'$. If $\text{supp}(\text{nf}(e)) \cap \text{supp}(\text{nf}(e')) \neq \emptyset$ then $e = e'$.*

Proof. Let $e_0 \in \text{supp}(\text{nf}(e)) \cap \text{supp}(\text{nf}(e'))$ and fix $r_0 \triangleleft e_0$. By Lemma 1.6.2, there are $r \triangleleft e$ and $r' \triangleleft e'$ such that $r_0 = \text{nf}(r) = \text{nf}(r')$. Since $e \supset e'$, we have $r \supset r'$ and, since $r_0 \cong r_0$, we obtain $r \cong r'$ by Lemma 1.6.7, hence $e = e'$. \square

1.6.2. Commutation

By assembling all our previous results, we obtain the desired commutation theorem:

Theorem 1.6.9. *Let $M \in \Lambda_{\oplus}$. Then $BT(M)^* = \text{nf}(M^*)$.*

Proof. By Theorem 1.4.3

$$M^* = \sum_{s \in T(M)} \frac{1}{m(s)} s$$

and by Theorem 1.4.1 and Theorem 1.6.8 we are allowed to form

$$\text{nf}(M^*) = \sum_{s \in T(M)} \frac{1}{m(s)} \text{nf}(s) = \sum_{s \in T(M)} \sum_{u \in \text{supp}(\text{nf}(s))} \frac{\text{nf}(s)_u}{m(s)} u$$

1. Taylor Expansion for λ -Terms with Choice Operator – 1.6. Normalizing the Taylor Expansion

the inner sums having pairwise disjoint supports. Then, if $u \in \text{supp}(\text{nf}(M^*))$, there is a unique $s \in \mathcal{T}(M)$ such that $u \in \text{supp}(\text{nf}(s))$ and we obtain $\text{nf}(M^*)_u = \frac{\text{nf}(s)_u}{m(s)} = \frac{1}{m(u)}$ by Theorem 1.6.6. We conclude since $\text{supp}(\text{nf}(M^*)) = \mathcal{T}(BT(M))$ by Theorem 1.4.7. \square

2. Categorical Interlude

In this chapter we present some categorical concepts that we shall need for the semantic investigations led in the rest of the thesis. The results presented are not original, with the exception of some minor ones, that are however corollary of already known theorems and methods. We presuppose familiarity with basic category theory and with the categorical semantics of linear logic, in particular the notion of Seely category [Mel09][pp. 145-150].

An important conceptual point is made in Section 2.6, where our point of view about an algebraic theory of resources is explained. We build on the ideas presented in [MZ18] and on the fundamental intuition of linear logic categorical semantics. Even if our semantic standpoint is deeply rooted in intuitions from linear logic, we shall keep this aspect rather implicit, since we focus on the denotational semantics of λ -calculus. For this reason, we are happy with cartesian closed bicategories (Section 2.2.1) and we do not introduce all the imposing technical machinery linked with monoidal bicategories.

The main categorical tool that we shall present here is a method, introduced in [Fio+17], that allows to lift 2-monads over the 2-category Cat of small categories, functors and natural transformations to (relative) pseudomonads over the bicategory Dist of small categories, distributors and natural transformations, without the need of introducing distributive laws in the bicategorical setting.

The Kleisli bicategories for an appropriate class of lifted 2-monads, the *resource monads* (Section 2.6.3) constitute a family of cartesian closed bicategories (Section 2.8.2.1), hence, in particular, bicategorical models of the λ -calculus. From this family of Kleisli bicategories we build our denotational semantics in Chapters 3 and 4.

These constructions depend on a considerable amount of theorems of advanced category theory, for this reason we try to recall most of the basic ingredients needed. However, we do not intend in any way to give an exhaustive presentation of these results, that would be beyond the scope of this thesis.

Structure of the Chapter We begin recalling some basic notions of monoidal categories. We introduce both monoidal and monoidal unbiased versions.

We then introduce bicategories, with some related structures, such as pseudofunctors, pseudonatural transformations, pseudoadjoints etc. Particularly important for our purposes is the definition of cartesian closed bicategories (Section 2.2.1).

Section 2.3 introduces some elements of two-dimensional monad theory and some algebras constructions for (relative) pseudomonads. Particularly important is the

notion of lifting to pseudoalgebras for a relative pseudomonad (Section 2.3.3).

Section 2.4 just recalls some facts about two well-known universal categorical constructions, (Left) Kan extensions and coends. In the following chapters, we shall use very often the fundamental theorems of coend calculus, such as Theorems 2.1 and 2.4.4.

Section 2.5 deals with some properties of presheaves. In particular, we recall that the presheaves construction determines free cocompletions of small categories. We then introduce a tensor product on presheaves, the Day convolution. We recall some properties of the Day convolution together with the strong monoidal structure induced on the Yoneda embedding. The universal property of the Day convolution (Theorem 2.5.5) is crucial for the results of Sections 2.6 and 2.8.

Section 2.6 introduces fundamental intuitions for our semantic investigations. We briefly recall the Boom hierarchy of data types and we give a natural monadic interpretation of it. We then define the categories of integers and lists. Finally, we consider a collection of doctrines, the resource monads, giving a concrete characterization of them in terms of lists and showing that the relative pseudomonad of presheaves lifts to resource monads pseudoalgebras.

Section 2.7 contains some particular categorical structures that determine models of linear logic. We start by recalling the well-known relational model and its preorder-induced generalization. We define the bicategory of distributors, which generalizes in a categorified setting the former two one-dimensional structures.

We conclude with Section 2.8, proving that the resource monads can be lifted to relative pseudomonads over the bicategory of distributors, exploiting much of the technical machinery introduced in former sections. We consider the family of Kleisli bicategories for these relative pseudomonads and we prove how their opposite bicategories are cartesian closed.

2.1. Monoidal Categories

In this section we present several notions of monoidal categories.

Definition 2.1.1. *A monoidal category is a 6-tuple $\mathbb{A} = \langle C, \otimes, 1, \alpha, \lambda, \rho \rangle$, where C is a category, $- \otimes - : C \times C \rightarrow C$ is a bifunctor, $1 \in C$ is called the unit of the tensor product and α, λ, ρ are natural isomorphisms*

$$\alpha_{a,b,c}(a \otimes b) \otimes c \cong a \otimes (b \otimes c)$$

$$\lambda_a : a \otimes 1 \cong a \quad \rho_a : 1 \otimes a \cong a$$

called, respectively, the associator, left unitor and right unitor, such that the

2. Categorical Interlude – 2.1. Monoidal Categories

following diagrams commute:

$$\begin{array}{ccc}
 & a \otimes (b \otimes (c \otimes d)) & \\
 & \swarrow 1 * \alpha \quad \searrow \alpha & \\
 a \otimes ((b \otimes c) \otimes d) & & (a \otimes b) \otimes (c \otimes d) \\
 & \swarrow \alpha \quad \searrow \alpha & \\
 (a \otimes (b \otimes c)) \otimes d & \xrightarrow{\alpha * 1} & ((a \otimes b) \otimes c) \otimes d
 \end{array}$$

$$\begin{array}{ccc}
 (a \otimes 1) \otimes b & \xrightarrow{\alpha} & a \otimes (1 \otimes b) \\
 \rho * 1 \downarrow & \swarrow 1 * \lambda & \\
 a \otimes b & &
 \end{array}$$

A monoidal category is *strict* if the natural isomorphisms α, λ, ρ are identities.

Definition 2.1.2. A lax monoidal functor $\mathbb{F} : \mathbb{A} = \langle C, \otimes, 1, \alpha, \lambda, \rho \rangle \rightarrow \mathbb{B} = \langle C', \otimes', 1', \alpha', \lambda', \rho' \rangle$ is a functor $F : A \rightarrow B$ together with natural transformations $\phi_{a,b}^1 : F(a) \otimes' F(b) \rightarrow F(a \otimes b)$ and $\phi^0 : 1' \rightarrow F(1)$ such that the following diagrams commute:

$$\begin{array}{ccc}
 (F(a) \otimes' F(b)) \otimes' F(c) & \xrightarrow{\alpha'_{a,b,c}} & F(a) \otimes' (F(b) \otimes' F(c)) \\
 \phi^1_{(a,b) \otimes 1} \downarrow & & \downarrow 1 \otimes \phi^1_{(b,c)} \\
 F(a \otimes b) \otimes' F(c) & & F(a) \otimes' F(b \otimes c) \\
 \phi^1_{(a \otimes b) \otimes c} \downarrow & & \downarrow \phi^1_{a \otimes (b \otimes c)} \\
 F((a \otimes b) \otimes c) & \xrightarrow{F(\alpha_{a,b,c})} & F(a \otimes (b \otimes c))
 \end{array}$$

$$\begin{array}{ccc}
 1' \otimes F(a) & \xrightarrow{\lambda'_a} & F(a) \\
 \phi^0 \downarrow & & \downarrow F(\lambda_a) \\
 F(1) \otimes F(a) & \xrightarrow{\phi^1_{1,a}} & F(1 \otimes a)
 \end{array}$$

2. Categorical Interlude – 2.1. Monoidal Categories

$$\begin{array}{ccc}
 F(a) \otimes 1' & \xrightarrow{\rho'_a} & F(a) \\
 1 \otimes \phi^0 \downarrow & & \downarrow F(\rho_a) \\
 F(a) \otimes F(1) & \xrightarrow{\phi_{a,1}^1} & F(a \otimes 1)
 \end{array}$$

A *strong monoidal functor* is a lax monoidal functor such that the arrows $\phi_{a,b}^1, \phi^0$ are isomorphisms.

Definition 2.1.3. A symmetric monoidal category is a monoidal category $\mathbb{A} = \langle C, \otimes, 1, \alpha, \lambda, \rho \rangle$ equipped with a natural isomorphism $\sigma_{a,b} : a \otimes b \cong b \otimes a$, called the symmetry, such that the following diagrams commute:

$$\begin{array}{ccc}
 (a \otimes b) \otimes c & \xrightarrow{\sigma_{a,b} \otimes 1_c} & (b \otimes a) \otimes c \\
 \downarrow \alpha_{a,b,c} & & \downarrow \alpha_{b,a,c} \\
 a \otimes (b \otimes c) & & b \otimes (a \otimes c) \\
 \downarrow \sigma_{a,b \otimes c} & & \downarrow 1_b \otimes \sigma_{a,c} \\
 (b \otimes c) \otimes a & \xrightarrow{\alpha_{b,c,a}} & b \otimes (c \otimes a)
 \end{array}$$

$$\begin{array}{ccc}
 a \otimes 1 & \xrightarrow{\sigma_{a,1}} & 1 \otimes a \\
 \searrow \lambda_a & & \downarrow \rho_a \\
 & & a
 \end{array}
 \qquad
 \begin{array}{ccc}
 a \otimes b & \xrightarrow{\sigma_{a,b}} & b \otimes a \\
 \searrow & & \downarrow \sigma_{b,a} \\
 & & a \otimes b
 \end{array}$$

A *symmetric monoidal functor* is then a monoidal functor that preserves the structure on the nose. More precisely, given $\mathbb{A} = \langle C, \otimes, 1, \alpha, \lambda, \rho, \sigma \rangle, \mathbb{B} = \langle C', \otimes', 1', \alpha', \lambda', \rho', \sigma' \rangle$ a monoidal functor is symmetric if the following diagram commutes:

$$\begin{array}{ccc}
 F(a) \otimes' F(b) & \xrightarrow{\sigma'_{a,b}} & F(b) \otimes' F(a) \\
 \phi_{a,b} \downarrow & & \downarrow \phi_{b,a} \\
 F(a \otimes b) & \xrightarrow{F(\sigma_{a,b})} & F(b \otimes a)
 \end{array}$$

Definition 2.1.4. A relevant monoidal category is a symmetric monoidal category $\mathbb{A} = \langle C, \otimes, 1, \alpha, \lambda, \rho, \sigma \rangle$ equipped with a natural transformation $c_a : a \rightarrow a \otimes a$, called the diagonal, such that the following diagrams commute:

$$\begin{array}{ccc}
 a & \xrightarrow{c_a} & a \otimes a & \xrightarrow{c_a \otimes 1_a} & (a \otimes a) \otimes a \\
 \parallel \downarrow & & & & \downarrow \alpha_{a,a,a} \\
 a & \longrightarrow & a \otimes a & \xrightarrow{1_a \otimes c_a} & a \otimes (a \otimes a)
 \end{array}$$

2. Categorical Interlude – 2.1. Monoidal Categories

$$\begin{array}{ccc}
 1 & \xrightarrow{c_1} & 1 \otimes 1 \\
 & \searrow & \downarrow \rho = \lambda \\
 & & 1
 \end{array}
 \qquad
 \begin{array}{ccc}
 a & \xrightarrow{c_a} & a \otimes a \\
 & \searrow c_a & \downarrow \sigma_{a,a} \\
 & & a \otimes a
 \end{array}$$

A *relevant monoidal functor* is then a symmetric monoidal functor that preserves the structure on the nose. More precisely, given $\mathbb{A} = \langle C, \otimes, 1, \alpha, \lambda, \rho, \sigma, c \rangle$, $\mathbb{B} = \langle C', \otimes', 1', \alpha', \lambda', \rho', \sigma', c' \rangle$ a symmetric monoidal functor is relevant if the following diagram commutes:

$$\begin{array}{ccc}
 F(a) & \xrightarrow{c'_a} & F(a) \otimes' F(a) \\
 \phi_{a,b} \downarrow & & \downarrow \phi_{b,a} \\
 F(a) & \xrightarrow{F(c_a)} & F(a \otimes a)
 \end{array}$$

Definition 2.1.5. A *semicartesian monoidal category* is a symmetric monoidal category $\mathbb{A} = \langle C, \otimes, 1, \alpha, \lambda, \rho, \sigma \rangle$ such that the unit is a terminal object. We write then $e_a : a \rightarrow 1$ for the terminal morphism.

A *semicartesian monoidal functor* is then a monoidal functor that preserves the structure on the nose.

Definition 2.1.6 (Colimits and monoidality). Given A_1, \dots, A_n categories, we say that $F : \prod_{i=1}^n A_i \rightarrow C$ is *separately cocontinuous* if for all $i \in [n]$ and $a_1 \in A_1, \dots, a_i \in A_i, \dots, a_n \in A_n$ the functors $F(a_1, \dots, a_{i-1}, -, a_{i+1}, \dots, a_n) : A_i \rightarrow C$ are cocontinuous.

Given a monoidal category $\mathbb{A} = \langle A, \otimes, 1 \rangle$ such that A is a cocomplete category, we say that \mathbb{A} is *monoidally cocomplete* if \otimes is a separately cocontinuous functor.

Definition 2.1.7. Given monoidal functors $\mathbb{F} = \langle F, \phi^1, \phi^0 \rangle$, $\mathbb{G} = \langle G, \psi^1, \psi^0 \rangle$ a monoidal natural transformation $\alpha : \mathbb{F} \rightarrow \mathbb{G}$ is just a natural transformation $\alpha : F \rightarrow G$ such that the following diagrams commute:

$$\begin{array}{ccc}
 F(a) \otimes' F(b) & \xrightarrow{\phi_{a,b}^1} & F(a \otimes b) \\
 \alpha_a \otimes' \alpha_b \downarrow & & \downarrow \alpha_{a,b} \\
 G(a) \otimes' G(b) & \xrightarrow{\psi_{a,b}^1} & G(a \otimes b)
 \end{array}$$

$$\begin{array}{ccc}
 1' & \xrightarrow{\phi^0} & F(1) \\
 & \searrow \psi_0 & \downarrow \alpha_1 \\
 & & G(1)
 \end{array}$$

Remark 2.1.8. In any *semicartesian monoidal category* we can define canonical projections:

$$\pi_1 : a \otimes b \rightarrow a \qquad \pi_2 : a \otimes b \rightarrow b$$

$$\pi_1 = 1_a \otimes e_b \quad \pi_2 = e_a \otimes 1_b.$$

If a monoidal category is both relevant and semicartesian then its tensor product is a cartesian product.

2.1.1. Unbiased Monoidal Categories

In a standard monoidal category we can define n -ary tensor products in the natural way, exploiting the binary functor $- \otimes - : C \times C \rightarrow C$. For this reason we can say that the standard definition of monoidal category is "biased" toward the arity $n = 2$. It is then possible, and useful, to present also an unbiased definition of monoidal category. Standard monoidal categories and unbiased ones are equivalent [Lei03].

Definition 2.1.9. An unbiased lax monoidal category is the collection of the following data:

- a Category C .
- A family of functors $\otimes_n : C^n \rightarrow C$ called the n -fold tensor product

$$\langle a_1, \dots, a_n \rangle \mapsto (a_1 \otimes \dots \otimes a_n).$$

- For $n, n_1, \dots, n_{k_n} \in \mathbb{N}$ and double sequence of objects of A ,

$$\langle \langle a_{1,1}, \dots, a_{1,k_1} \rangle, \dots, \langle a_{n,1}, \dots, a_{n,k_n} \rangle \rangle$$

a family of arrows

$$\begin{aligned} \gamma_{\langle \langle a_{1,1}, \dots, a_{1,k_1} \rangle, \dots, \langle a_{n,1}, \dots, a_{n,k_n} \rangle \rangle} : & ((a_{1,1} \otimes \dots \otimes a_{1,k_1}) \otimes \dots \otimes (a_{n,1} \otimes \dots \otimes a_{n,k_n})) \\ \rightarrow & (a_{1,1} \otimes \dots \otimes a_{1,k_1} \otimes \dots \otimes a_{n,1} \otimes \dots \otimes a_{n,k_n}) \end{aligned}$$

natural in the $a_{j,i}$.

- a family of arrows

$$\iota_a : a \rightarrow (a)$$

natural in $a \in C$.

The families γ, ι must satisfy associativity and identity laws [Lei03]. We call $\otimes_0(*) = ()$ the unit.

An unbiased monoidal category is a lax monoidal category where the arrows $\gamma_{\langle \langle a_{1,1}, \dots, a_{1,k_1} \rangle, \dots, \langle a_{n,1}, \dots, a_{n,k_n} \rangle \rangle}, \iota_a$ are isomorphisms. An unbiased strict monoidal category is a lax monoidal category where the arrows $\gamma_{\langle \langle a_{1,1}, \dots, a_{1,k_1} \rangle, \dots, \langle a_{n,1}, \dots, a_{n,k_n} \rangle \rangle}, \iota_a$ are identities.

Definition 2.1.10. An unbiased symmetric monoidal category is an unbiased monoidal category $\mathbb{C} = \langle C, \otimes_n, \gamma, \iota \rangle$ equipped, for all $n \in \mathbb{N}$ and $\sigma \in \mathfrak{S}_n$ with a

2. Categorical Interlude – 2.1. Monoidal Categories

family of arrows

$$\sigma_{\langle a_1, \dots, a_n \rangle} : (a_1 \otimes \cdots \otimes a_n) \rightarrow (a_{\sigma(1)} \otimes \cdots \otimes a_{\sigma(n)})$$

for $a_1, \dots, a_n \in C$, natural in the a_i and functorial, i.e. $\sigma_{\langle a_1, \dots, a_n \rangle} \circ \tau_{\langle a_1, \dots, a_k \rangle} = (\sigma \circ \tau)_{\langle a_1, \dots, a_k \rangle}$ and $\text{id}_{[n]} = 1_{\langle a_1, \dots, a_k \rangle}$. Moreover, we impose the following coherence conditions:

— For all $\sigma_1 \in \mathfrak{S}_{k_1}, \dots, \sigma_n \in \mathfrak{S}_{k_n}$

$$\begin{array}{ccc} ((a_{1,1} \otimes \cdots \otimes a_{1,k_1}) \otimes \cdots \otimes (a_{n,1} \otimes \cdots \otimes a_{n,k_n})) & \xrightarrow{\gamma} & (a_{1,1} \cdots \otimes a_{n,k_n}) \\ \downarrow (\sigma_1 \otimes \cdots \otimes \sigma_n) & & \downarrow \sigma \\ ((a_{1,\sigma_1(1)} \otimes \cdots \otimes a_{1,\sigma_1(k_1)}) \otimes \cdots \otimes (a_{n,\sigma_n(1)} \otimes \cdots \otimes a_{n,\sigma_n(k_n)})) & \xrightarrow{\sigma_{\sum_{i=1}^n k_i}} & (a_{1,\sigma_1(1)} \otimes \cdots \otimes a_{n,\sigma_n(k_n)}) \end{array}$$

— For all $\tau \in \mathfrak{S}_n$

$$\begin{array}{ccc} ((a_{1,1} \otimes \cdots \otimes a_{1,k_1}) \otimes \cdots \otimes (a_{n,1} \otimes \cdots \otimes a_{n,k_n})) & \xrightarrow{\gamma} & (a_{1,1} \cdots \otimes a_{n,k_n}) \\ \downarrow \tau & & \downarrow \tau \circ (1_{k_1}, \dots, 1_{k_n}) \\ ((a_{\tau(1),1} \otimes \cdots \otimes a_{\tau(1),k_{\tau(1)}}) \otimes \cdots \otimes (a_{\tau(n),1} \otimes \cdots \otimes a_{\tau(n),k_{\tau(n)}})) & \xrightarrow{\alpha} & (a_{\tau(1),1} \otimes \cdots \otimes a_{\tau(n),k_{\tau(n)}}) \end{array}$$

where we recall that $\tau \circ (1_{k_1}, \dots, 1_{k_n})$ is the multiplexing operation (Subsection 1.5.3).

Definition 2.1.11. An unbiased relevant monoidal category is an unbiased symmetric monoidal category $\mathbb{C} = \langle C, \otimes_n, \gamma, \iota, \sigma \rangle$ equipped, for all $n \in \mathbb{N}$, with a family of arrows

$$c_a^n : a \rightarrow \overbrace{(a \otimes \cdots \otimes a)}^{n \text{ times}}$$

for $a \in C$, such that the following diagrams commute

$$\begin{array}{ccc} a & \xrightarrow{c^n} & (a \otimes \cdots \otimes a) \xrightarrow{c^{k_1} \otimes \cdots \otimes c^{k_n}} (a^{k_1} \otimes \cdots \otimes a^{k_n}) \\ & \searrow & \downarrow \gamma \\ & & a^{\sum_{i=1}^n k_i} \end{array}$$

$$\begin{array}{ccc} a & \xrightarrow{c^n} & (a \otimes \cdots \otimes a) \\ & \searrow c^n & \downarrow \sigma \\ & & (a \otimes \cdots \otimes a) \end{array}$$

for $\sigma \in \mathfrak{S}_n$, letting $a^n = \overbrace{(a \otimes \cdots \otimes a)}^{n \text{ times}}$.

Definition 2.1.12. An unbiased semicartesian monoidal category is an unbiased symmetric monoidal category $\mathbb{C} = \langle C, \otimes_n, \gamma, \iota, \sigma \rangle$ where the unit $()$ is terminal.

Definition 2.1.13. An unbiased cartesian monoidal categories is an unbiased symmetric monoidal category $\mathbb{C} = \langle C, \otimes_n, \gamma, \iota, \sigma \rangle$ where the functors \otimes_n are finite products.

2.2. Bicategories

Bicategories are weak 2-dimensional categorical structures, where we have objects, arrows between objects (called *1-cells*) and arrows between arrows (called *2-cells*). Identity and associativity laws for compositions of 1-cells hold only up to coherent isomorphisms, for this reason bicategories are weak structures. We refer to [Bor94] for the basics of bicategory theory.

A *bicategory* \mathcal{C} is the collection of the following data:

- A set of *objects*, also called *0-cells* $Obj(\mathcal{C}) \ni A, B, C \dots$;
- for all $A, B \in Obj(\mathcal{C})$, a category $\mathcal{C}(A, B)$. Objects of these categories are called *1-cells* or *morphisms*, while arrows are called *2-cells* or *2-morphisms*. Composition of 2-cells is generally called *vertical composition*;
- For every $A, B, C \in Obj(\mathcal{C})$ a functor

$$\begin{aligned} \circ_{A,B,C} : \mathcal{C}(B, C) \times \mathcal{C}(A, B) &\rightarrow \mathcal{C}(A, C) \\ (G, F) &\mapsto G \circ F \\ (\beta, \alpha) &\mapsto \beta \star \alpha \end{aligned}$$

called horizontal composition;

- for every $A \in ob(\mathcal{C})$ a functor $1_A : 1 \rightarrow \mathcal{C}(A, A)$. With an abuse of notation we identify $1_A(*)$ with 1_A and we call it the identity of A ;
- For all $A, B, C, D \in ob(\mathcal{C})$ a natural isomorphism $\alpha_{A,B,C,D}$:

$$\begin{array}{ccc} \mathcal{C}(C, D) \times \mathcal{C}(B, C) \times \mathcal{C}(A, B) & \xrightarrow{1 \times \circ_{A,B,C}} & \mathcal{C}(D, C) \times \mathcal{C}(A, C) \\ \downarrow \circ_{B,C,D} \times 1 & \swarrow \alpha_{A,B,C,D} & \downarrow \circ_{A,C,D} \\ \mathcal{C}(B, D) \times \mathcal{C}(A, B) & \xrightarrow{\circ_{A,B,D}} & \mathcal{C}(A, D) \end{array}$$

- For all $A \in ob(\mathcal{C})$, natural isomorphisms λ_A and ρ_A :

2. Categorical Interlude – 2.2. Bicategories

$$\begin{array}{ccc}
 \mathcal{C}(A, B) \times 1 & \longrightarrow & \mathcal{C}(A, B) \\
 \downarrow 1 \times 1_A & \swarrow \rho_A & \uparrow \\
 \mathcal{C}(A, B) \times \mathcal{C}(A, A) & \xrightarrow{\circ_{A,A,B}} &
 \end{array}
 \qquad
 \begin{array}{ccc}
 1 \times \mathcal{C}(A, B) & \longrightarrow & \mathcal{C}(A, B) \\
 \downarrow 1_B \times 1 & \swarrow \lambda_A & \uparrow \\
 \mathcal{C}(B, B) \times \mathcal{C}(A, B) & \xrightarrow{\circ_{A,A,B}} &
 \end{array}$$

— Finally, we impose two coherence conditions, *i.e.* the commutation of the following two diagrams:

$$\begin{array}{ccc}
 & K \circ (H \circ (G \circ F)) & \\
 & \swarrow K \circ \alpha & \searrow \alpha \\
 K \circ ((H \circ G) \circ F) & & (K \circ H) \circ (G \circ G) \\
 \downarrow \alpha & & \downarrow \alpha \\
 (K \circ (H \circ G)) \circ F & \xrightarrow{\alpha \circ F} & ((K \circ H) \circ G) \circ F
 \end{array}$$

$$\begin{array}{ccc}
 (G \circ 1) \circ F & \xrightarrow{\alpha} & G \circ (1 \circ F) \\
 \downarrow \rho \circ F & \swarrow G \circ \lambda & \\
 G \circ F & &
 \end{array}$$

Given a bicategory \mathcal{C} , there exists its *opposite bicategory*, where we reverse only the 1-cells, denoted as \mathcal{C}^{op} .

A *2-category* is a bicategory where the natural isomorphisms α, ρ, λ are identities.

Example 2.2.1. We list some examples of bicategories that we will use in what follows.

— The associativity and identity laws for bicategories look very much alike the conditions that one imposes over the associativity and unity natural transformations of monoidal categories. Indeed, we have that all monoidal categories are one object bicategories (easy check of the definitions).

2. Categorical Interlude – 2.2. Bicategories

- For this work, the most important example of bicategory is the bicategory of distributors (Section 2.7.3), where objects are small categories, 1-cells are functors $F : B^o \times A \rightarrow \text{Set}$ and 2-cells are natural transformations.
- Categories, functors and natural transformations determine a 2-category, CAT. We write Cat for its full sub-2-category of small categories.
- Locally small monoidal categories, strong monoidal functors and monoidal natural transformations determine a 2-category, MON. We write Mon for its full sub-2-category of small monoidal categories.
- Locally small symmetric monoidal categories, strong symmetric monoidal functors and monoidal natural transformations determine a 2-category, SMMON. We write SMMon for its full sub-2-category of small symmetric monoidal categories.
- Locally small monoidally cocomplete categories, strong monoidal cocontinuous functors and monoidal natural transformations determine a 2-category, MONCOC.
- Locally small semicartesian monoidal categories, strong semicartesian monoidal functors and monoidal natural transformations determine a 2-category, SCMON. We write SCMon for its full sub-2-category of small semicartesian monoidal categories. relevant monoidal categories, strong relevant monoidal functors and monoidal natural transformations determine a 2-category, RMON. We write RMon for its full sub-2-category of small relevant monoidal categories.

Definition 2.2.2. Let \mathcal{A}, \mathcal{B} be two bicategories. A lax functor $\Phi : \mathcal{A} \rightarrow \mathcal{B}$ is the collection of the following data:

- A function $\Phi : \text{ob}(\mathcal{A}) \rightarrow \text{ob}(\mathcal{B})$.
- For each pair of objects A, B a functor $\Phi_{A,B} : \mathcal{A}(A, B) \rightarrow \mathcal{B}(\Phi_0(A), \Phi_0(B))$.
- For all $A, B, C \in \text{ob}(\mathcal{A})$ a natural transformation $\phi_{A,B,C}$

$$\begin{array}{ccc}
 \mathcal{A}(B, C) \times \mathcal{A}(A, B) & \xrightarrow{\Phi_{A,B} \times \Phi_{B,C}} & \mathcal{B}(\Phi B, \Phi C) \times \mathcal{B}(\Phi A, \Phi C) \\
 \downarrow \circ_{A,B,C} & \swarrow \phi_{A,B,C} & \downarrow \circ_{\Phi A, \Phi B, \Phi C} \\
 \mathcal{A}(A, C) & \xrightarrow{\Phi_{A,C}} & \mathcal{B}(\Phi A, \Phi C)
 \end{array}$$

with components $\phi_{F,G} : \Phi(G) \circ \Phi(F) \rightarrow \Phi(G \circ F)$.

- For all $A \in \text{ob}(\mathcal{C})$ a natural transformation ϕ_A

2. Categorical Interlude – 2.2. Bicategories

$$\begin{array}{ccc}
 1 & \xrightarrow{1_A} & \mathcal{A}(A, A) \\
 \parallel & \searrow \phi_A & \downarrow \Phi_{A,A} \\
 1 & \xrightarrow{1_{\Phi A}} & \mathcal{B}(\Phi A, \Phi A)
 \end{array}$$

with components $\Phi_A : \Phi(1_A) \rightarrow 1_{\Phi A}$

Finally, we impose three coherence axioms (see [Lei98][p.4]).

If the two natural transformations are isomorphisms, Φ is called a pseudofunctor.

A 2-functor is a pseudo-functor where the two natural isomorphisms are identities.

Definition 2.2.3. Let $\Phi, \Psi : \mathcal{A} \rightarrow \mathcal{B}$ be two morphisms. A pseudonatural transformation $P : \Phi \Rightarrow \Psi$ is the collection of the following data:

- A family of 1-cells $(P_A : \Phi A \rightarrow \Psi A)_{A \in \mathcal{A}}$.
- for each 1-cell $F : A \rightarrow B$ an invertible 2-cell P_F :

$$\begin{array}{ccc}
 \Phi A & \xrightarrow{P_A} & \Psi A \\
 \downarrow \Phi F & \Downarrow P_F & \downarrow \Psi F \\
 \Phi B & \xrightarrow{P_B} & \Psi B
 \end{array}$$

- We impose the following conditions:

1. For all $F : A \rightarrow B$ and $G : B \rightarrow C$ we have

$$\begin{array}{ccc}
 \Phi A & \xrightarrow{P_A} & \Psi A \\
 \downarrow \Phi F & \Downarrow P_F & \downarrow \Psi F \\
 \Phi B & \xrightarrow{P_B} & \Psi B \\
 \downarrow \Phi G & \Downarrow P_G & \downarrow \Psi G \\
 \Phi C & \xrightarrow{P_C} & \Psi C
 \end{array}
 =
 \begin{array}{ccc}
 \Phi A & \xrightarrow{P_A} & \Psi A \\
 \downarrow \Phi_{G \circ F} & \Downarrow P_{G \circ F} & \downarrow \Psi_{G \circ F} \\
 \Phi C & \xrightarrow{P_B} & \Psi C
 \end{array}$$

2. for all $A \in \mathcal{A}$

$$\begin{array}{ccc}
 \Phi A & \xrightarrow{P_A} & \Psi A \\
 \downarrow \Phi 1_A & \Downarrow P_{1_A} & \downarrow \Psi 1_A \\
 \Phi A & \xrightarrow{P_A} & \Psi A
 \end{array}
 =
 \begin{array}{ccc}
 \Phi A & \xrightarrow{P_A} & \Psi A \\
 \downarrow 1_{\Phi A} & & \downarrow 1_{\Psi A} \\
 \Phi A & \xrightarrow{P_A} & \Psi A
 \end{array}$$

A *2-natural transformation* is a pseudonatural transformation where the invertible 2-cells P_F are identities.

Definition 2.2.4. Let $P, Q : \Phi \Rightarrow \Psi$ be two pseudo-natural transformations. A modification $\sigma : P \Rightarrow Q$ consists in a family of 2-cells $\sigma_A : P(A) \rightarrow Q(A)$ such that

$$\begin{array}{ccc}
 \Phi A & \xrightarrow{P_A} & \Psi A \\
 \downarrow \Phi F & \Downarrow \sigma & \downarrow \Psi F \\
 \Phi B & \xrightarrow{Q_B} & \Psi B
 \end{array}
 =
 \begin{array}{ccc}
 \Phi A & \xrightarrow{P_A} & \Psi A \\
 \downarrow \Phi F & \Downarrow P_F & \downarrow \Psi F \\
 \Phi B & \xrightarrow{P_B} & \Psi B
 \end{array}$$

2.2.0.1. Pseudoadjunctions

We can now introduce the notion of pseudoadjunction. We do so by giving a definition based on *biuniversal arrows* [Fio06] [Sav20].

Definition 2.2.5. Let $\Phi : \mathcal{C} \rightarrow \mathcal{D}$ be a pseudofunctor and $D \in \mathcal{D}$. A biuniversal arrow from Φ to D consists of a pair $\langle C \in \mathcal{C}, q_C : \Phi C \rightarrow D \rangle$ and for every $B \in \mathcal{C}$ an adjoint equivalence of categories

$$\begin{array}{ccc}
 \mathcal{D}(\Phi B, D) & \xleftarrow{q_C \circ \Phi(-)} & \mathcal{C}(B, C) \\
 & \perp & \\
 & \xrightarrow{(-)^b} &
 \end{array}$$

Definition 2.2.6 ((Right) pseudoadjoint). Let $\Phi : \mathcal{A} \rightarrow \mathcal{B}$ be a pseudo-functor. A right pseudoadjoint $\Phi \dashv \Psi$ is given by the following data:

- A function $\Psi : \text{ob}(\mathcal{B}) \rightarrow \text{ob}(\mathcal{A})$.
- For every $B \in \mathcal{B}$ a universal arrow $\langle \Psi B, q_A : \Phi \Psi B \rightarrow B \rangle$ from Φ to B .

These data univocally determine a pseudo-functor $\Psi : \mathcal{B} \rightarrow \mathcal{A}$ such that $\Phi \dashv \Psi$. For example, the functor $\psi_{B,C} : \mathcal{B}(B, C) \rightarrow \mathcal{A}(\Psi B, \Psi C)$ is defined as $(- \circ q_C)^b$, where

the action on 2-cells is given by precomposition with the identity $id_{q_C} : q_C \Rightarrow q_C$. The former definition of pseudoadjunction is particularly suitable for calculations, this is the main reason for which we choose such presentation.

It is possible to give a formalization of basic category theory in the context of an arbitrary bicategory. More precisely, we can define the notions of equivalence, adjunctions, monads exploiting the structure of 1-cells and 2-cells. We give a particular example of this process of formalization with the following definition of a *retraction*.

Definition 2.2.7 (Retraction). *Let D, E be 0-cells in a bicategory \mathcal{C} . A retraction of D to E is a couple of 1-cells $i : E \rightarrow D$, $j : D \rightarrow E$ together with an invertible 2-cell β such that the diagram below commute. We write $E \triangleleft D$ if there is a retraction of D to E .*

$$\begin{array}{ccc} E & \xrightarrow{i} & D \\ 1_E \downarrow & \swarrow \beta & \uparrow j \\ E & \xleftarrow{j} & D \end{array}$$

If we have both $E \triangleleft D$ and $D \triangleleft E$ we say that E is equivalent to D , in symbols $E \simeq D$, and we call the pair (i, j) an equivalence.

2.2.1. Cartesian Closed Structure

Cartesian closed bicategories are of particular interest for our purpose, since the classic categorical semantics of λ -calculus is given by its one-dimensional analogue. This kind of structures has been thoroughly studied in [Sav20; FS19; FS20], where one of the main results is an important theorem of coherence for cartesian closed bicategories, proven via a type theoretic method¹. In the definition of cartesian closed bicategory we mostly follow [Sav20; GJ17].

Given bicategories $\mathcal{A}_1, \dots, \mathcal{A}_n$ with $n \in \mathbb{N}$ there exists the finite product bicategory $\prod_{i=1}^n \mathcal{A}_i$, defined in the natural way. Given a bicategory \mathcal{C} , we define the n -ary

diagonal pseudo-functor $\Delta_{\mathcal{C}}^n : \mathcal{C} \rightarrow \overbrace{\mathcal{C} \times \dots \times \mathcal{C}}^{n \text{ times}}$ as follows:

- On objects $\Delta^n(C) = \langle C, \dots, C \rangle$.
- For all $C, D \in \mathcal{C}$,

$$\begin{aligned} \Delta_{\mathcal{C}, D}^n : \mathcal{C}(C, D) &\rightarrow \mathcal{C} \times \mathcal{C}(\Delta^n(C), \Delta^n(D)) \\ (F, G) &\mapsto ((F, \dots, F), (G, \dots, G)) \end{aligned}$$

1. More precisely, a notion of type theory for cartesian closed bicategories is introduced and it is used to prove that in the free cartesian closed bicategory over a set X there is at most one 2-cell between two different 1-cells. This is a very important coherence result, since intuitively means that the cartesian closed structure on a bicategory can be studied "forgetting" about coherences conditions on 2-arrows.

2. Categorical Interlude – 2.3. Two-Dimensional Monad Theory

$$(\alpha, \beta) \mapsto ((\alpha, \dots, \alpha), (\beta, \dots, \beta))$$

The natural isomorphisms $\Delta_{F,G}^n$ and Δ_A^n are the identities. We say that a bicategory *admits all finite products* if for every $n \in \mathbb{N}$ the pseudofunctor Δ^n admits a right pseudoadjoint. If we spell this out, following Definition 2.2.6, we demand the following:

- A function $\prod_n(-) : \text{ob}(\mathcal{C}^n) \rightarrow \text{ob}(\mathcal{C})$.
- For every $A_1, \dots, A_n \in \text{ob}(\mathcal{C})$, a 1-cell $\pi_{n,i} : \prod_n \langle A_1, \dots, A_n \rangle \rightarrow A_i$ called the *i-th projection*.
- For every $B \in \text{ob}(\mathcal{C})$ an adjoint equivalence

$$\begin{array}{ccc} & \langle \pi_{n,1} \circ -, \dots, \pi_{n,n} \circ - \rangle & \\ \swarrow & & \searrow \\ \prod_{i=1}^n \mathcal{C}(B, A_i) & \perp & \mathcal{C}(B, \prod_n \langle A_1, \dots, A_n \rangle) \\ \searrow & & \swarrow \\ & (-) & \end{array}$$

where the right adjoint $(-)$ is called the *tupling*. We shall write $\prod_n(A_1, \dots, A_n) = \&_{i=1}^n A_i$ and, for $n = 2$, $\prod_2(A, B) = A \& B$.

A bicategory that admits finite products is *closed* if the pseudofunctor $- \& B : \mathcal{C} \rightarrow \mathcal{C}$ admits a right pseudoadjoint $-^B$. If we spell this out, we get the following:

- For every $B \in \text{ob}(\mathcal{C})$ function $-^B : \text{ob}(\mathcal{C}) \rightarrow \text{ob}(\mathcal{C})$.
- For every $A, B \in \text{ob}(\mathcal{C})$, a 1-cell $\text{ev}_{A,B} : A^B \& A \rightarrow B$ called the *evaluation morphism*.
- For every $A, B, C \in \text{ob}(\mathcal{C})$ an adjoint equivalence

$$\begin{array}{ccc} & \text{ev}_{B,C} \circ (- \times B) & \\ \swarrow & & \searrow \\ \prod_{i=1}^n \mathcal{C}(A \& B, C) & \perp & \mathcal{C}(A, C^B) \\ \searrow & & \swarrow \\ & \lambda(-) & \end{array}$$

the counit of the former adjunction $\epsilon_f : \text{ev}_{B,C} \circ (\lambda(f) \times B) \cong f$ models β -reduction.

2.3. Two-Dimensional Monad Theory

Two-dimensional monad theory is the study of algebraic constructions in the framework of bicategories. Monads over bicategories are called *pseudomonads*. Of particular relevance are the possible algebras structure that we can associate to pseudomonads. While in the one-dimensional framework the notion of algebra is unique, in the two-dimensional one we can consider *lax algebras*, *pseudotalgebras*

and *strict algebras*². We will focus on strict and pseudoalgebras. The main reference on two-dimensional monad theory is the classic [BKP89].

2.3.1. Pseudomonads and Lax Algebras

Definition 2.3.1. A pseudomonad over a bicategory \mathcal{A} is the collection of the following data:

- A triple $\langle T, \eta, \mu \rangle$ where $T : \mathcal{A} \rightarrow \mathcal{A}$ is a pseudo-functor, and $\eta : 1_{\mathcal{A}} \rightarrow T$ and $\mu : T^2 \rightarrow T$ are pseudonatural transformations.
- Invertible modifications

$$\begin{array}{ccc}
 T^3 & \xrightarrow{T\mu} & T^2 \\
 \downarrow \mu T & \Downarrow \sigma & \downarrow \mu \\
 T^2 & \xrightarrow{\quad} & T
 \end{array}
 \qquad
 \begin{array}{ccc}
 T & \xrightarrow{T\eta} & T^2 \\
 \eta T \downarrow & \swarrow \tau_1 & \downarrow \mu \\
 T^2 & \xrightarrow{\quad} & T \\
 & \nearrow \tau_2 & \\
 & \mu &
 \end{array}$$

We also impose two additional coherence conditions [Lac00][p.180].

A *2-monad* is a pseudomonad $(T, \eta, \mu, \sigma, \tau_1, \tau_2)$ where T is a 2-functor, η, μ are 2-natural transformations and σ, τ_1, τ_2 are identity modifications.

Given a pseudomonad $\langle S : \mathcal{C} \rightarrow \mathcal{C}, \eta, \mu, \sigma, \tau_1, \tau_2 \rangle$ we can build the *2-category of lax algebras of S* , $S\text{-LAlg}_{\mathcal{C}}$ as follows:

- An object of $S\text{-LAlg}_{\mathcal{C}}$ is given by an object $A \in \mathcal{C}$, called the *underlying object*, a 1-cell $h_A : SA \rightarrow A$ called the *structure map* and 2-cells ι_1, ι_2 :

$$\begin{array}{ccc}
 SSA & \xrightarrow{\mu_A} & SA \\
 h_{SA} \downarrow & \swarrow \iota_1 & \downarrow h_A \\
 SA & \xrightarrow{\quad} & A
 \end{array}
 \qquad
 \begin{array}{ccc}
 A & \xrightarrow{\eta_{SA}} & SA \\
 \swarrow \iota_2 & \searrow & \downarrow h_{SA} \\
 & & A
 \end{array}$$

We impose two additional coherence conditions on the 2-cells [Luc18]. If the 2-cells ι_1, ι_2 are isos, \mathbb{A} is called a *pseudoalgebra*. If they are identities, \mathbb{A} is a *strict algebra*.

We generally denote lax algebras by $\mathbb{A}, \mathbb{B}, \dots$

- For lax algebras \mathbb{A}, \mathbb{B} a 1-cell or morphism $\varphi : \mathbb{A} \rightarrow \mathbb{B}$ is a morphism $F : A \rightarrow B$ together with an invertible 2-cell

2. Moreover, one can choose between various degree of strictness also for 1-cells.

2. Categorical Interlude – 2.3. Two-Dimensional Monad Theory

$$\begin{array}{ccc}
 SA & \xrightarrow{SF} & SB \\
 \downarrow h_A & \swarrow \zeta & \downarrow h_B \\
 A & \xrightarrow{F} & B
 \end{array}$$

required to satisfy two coherence conditions [BKP89][p.3]. If ζ is an isomorphism, then the morphism is called a *pseudomorphism*. If ζ is the identity, then the morphism is called a *strict* morphism.

- Given two lax morphisms $\varphi = \langle F, \zeta \rangle, \psi = \langle F', \zeta' \rangle : \mathbb{A} \rightarrow \mathbb{B}$, a 2-cell $\bar{\alpha} : \mathbb{A} \rightarrow \mathbb{B}$ consists of a 2-cell $\alpha : F \rightarrow F'$ satisfying

$$\begin{array}{ccc}
 \begin{array}{ccc}
 SA & \xrightarrow{SF} & SB \\
 \downarrow h_A & \Downarrow \alpha & \downarrow h_B \\
 SA & \xrightarrow{SF'} & SB \\
 \downarrow h_A & \Downarrow \zeta' & \downarrow h_B \\
 A & \xrightarrow{F'} & B
 \end{array} & = & \begin{array}{ccc}
 SA & \xrightarrow{SF} & SB \\
 \downarrow h_A & \Downarrow \zeta & \downarrow h_B \\
 A & \xrightarrow{F} & B \\
 \downarrow h_A & \Downarrow \alpha & \downarrow h_B \\
 A & \xrightarrow{F'} & B
 \end{array}
 \end{array}$$

We denote the 2-category of pseudoalgebras and strict algebras as respectively $S\text{-PAlg}_{\mathcal{C}}$ and $S\text{-Alg}_{\mathcal{C}}$, in both cases the 1-cell considered are pseudomorphisms. Clearly we have that $S\text{-Alg}_{\mathcal{C}}$ is a full 2-subcategory of $S\text{-PAlg}_{\mathcal{C}}$. We will often write $\mathbb{A} = \langle A, h_A \rangle$ for a pseudo S -algebra, keeping the 2-arrows implicit.

Example 2.3.2. We give some examples of (pseudo)algebras for some 2-monads on the 2-category CAT that are important for the present work (Section 2.6).

- Consider the 2-monad S for monoidal categories. Strict algebras for this 2-monad are monoidal categories. More precisely, the 2-category $S\text{-Alg}_{\text{CAT}}$ is equivalent to the 2-category MON . A similar result holds for the strict, symmetric, relevant, semicartesian and cartesian cases. This example expresses very well why we chose pseudomorphisms of strict algebras as the 1-cells of $S\text{-Alg}_{\mathcal{C}}$. Strict morphisms are too restrictive: the structure of strong monoidal functors corresponds to pseudomorphisms.
- Consider the 2-monad S for strict monoidal categories. A pseudoalgebra for this kind of 2-monad consists exactly of an unbiased monoidal category [Lei03][pp. 69-70]. We briefly detail the proof in the case where S is the 2-monad for symmetric monoidal categories. Let $\langle A, h_A, \iota_1, \iota_2 \rangle$ be a pseudoal-

2. Categorical Interlude – 2.3. Two-Dimensional Monad Theory

gebra for S , by definition, this means that we have the following

$$\begin{array}{ccc}
 SSA & \xrightarrow{\mu_A} & SA \\
 h_{SA} \downarrow & \swarrow \iota_1 & \downarrow h_A \\
 SA & \xrightarrow{h_A} & A
 \end{array}
 \qquad
 \begin{array}{ccc}
 A & \xrightarrow{\eta_{SA}} & SA \\
 \swarrow \iota_2 & \searrow & \downarrow h_{SA} \\
 & & A
 \end{array}$$

we want to define the n -fold tensor products on A . Let $\langle \vec{a}_1, \dots, \vec{a}_k \rangle \in SSA$. By the first diagram and definition of S we have

$$h_A\left(\bigoplus_{i=1}^k \vec{a}_i\right) \cong h_A(\langle h_A(\vec{a}_1), \dots, h_A(\vec{a}_k) \rangle).$$

Then we set $\otimes_n(a_1, \dots, a_k) = h_A(\langle a_1, \dots, a_k \rangle)$. If we denote as natural

$$\otimes_n(a_1, \dots, a_n) = (a_1 \otimes \dots \otimes a_n),$$

we get, just rephrasing the former formula,

$$\iota_1 : (a_{1,1} \otimes \dots \otimes a_{1,k_1}) \otimes \dots \otimes (a_{n,1} \otimes \dots \otimes a_{n,k_n}) \cong (a_{1,1} \otimes \dots \otimes a_{n,k_n})$$

hence the natural isomorphism ι_1 will give the associativity isomorphism. The natural isomorphism ι_2 will instead give $\iota_2 : a \rightarrow (a)$, just imposing $h_A(\eta_A(a)) = (a)$. The coherence conditions for associativity and identity are satisfied just writing down the associativity and identity conditions for pseudoalgebras. For what concerns symmetries, lets consider the symmetries between lists

$$\sigma : \langle a_1, \dots, a_k \rangle \rightarrow \langle a_{\sigma(1)}, \dots, a_{\sigma(k)} \rangle.$$

Then we set $\sigma : (a_1 \otimes \dots \otimes a_k) \rightarrow (a_{\sigma(1)} \otimes \dots \otimes a_{\sigma(k)})$ just as $h_A(\sigma)$. The functoriality is then immediately derived by functoriality of h_A .

2.3.2. Relative Pseudomonads

It is well-known given a pair of adjoint functors, we get a monad. We want now to extend this result to the two-dimensional setting. In order to do so, we relax the notion of pseudoadjunction (Def 2.2.6) to the one of relative pseudoadjunction, that is more useful for our purposes. For the content of this section we refer to [Fio+17].

Definition 2.3.3 (Relative (left) pseudoadjunction). *Let $\Psi : \mathcal{E} \rightarrow \mathcal{D}, J : \mathcal{C} \rightarrow \mathcal{D}$*

2. Categorical Interlude – 2.3. Two-Dimensional Monad Theory

be two pseudofunctors. A relative left pseudoadjoint Φ to Ψ over J , denoted

$$\begin{array}{ccc} & & \mathcal{E} \\ & \nearrow \Phi & \downarrow \Psi \\ \mathcal{C} & \xrightarrow{J} & \mathcal{D} \end{array}$$

is the collection of the following data:

- for all $A \in \mathcal{C}$, an object $\Phi A \in \mathcal{E}$;
- for all $A \in \mathcal{C}$, a family of morphisms $i_A : JA \rightarrow \Psi\Phi A$;
- for all $A \in \mathcal{C}$ and $B \in \mathcal{E}$ a family of adjoint equivalences

$$\begin{array}{ccc} & \xrightarrow{(-)^{\flat}} & \\ \mathcal{D}(JA, \Psi B) & \perp & \mathcal{E}(\Phi A, B) \\ & \xleftarrow{\Psi(-) \circ i_A} & \end{array}$$

We perform the same kind of generalization also for the notion of pseudomonad.

Definition 2.3.4 (Relative pseudomonad). *Let $J : \mathcal{C} \rightarrow \mathcal{D}$ be a pseudofunctor between 2-categories. A relative pseudomonad T over J is the collection of the following data:*

- for $A \in \mathcal{C}$, an object $TA \in \mathcal{D}$;
- for $A, B \in \mathcal{C}$, a functor $(-)^*_{A,B} : \mathcal{D}(JA, TB) \rightarrow \mathcal{D}(TA, TB)$;
- for $A \in \mathcal{C}$, a morphism $i_A : JA \rightarrow TA$;
- for $F : A \rightarrow B$ and $G : B \rightarrow C$ a family of invertible two-cells $\mu_{F,G} : (G^* \circ F)^* \cong G^* \circ F^*$;
- for $F : JA \rightarrow TB$ a family of invertible two cells $\eta_F : F \cong F^* \circ i_x$;
- a family of invertible two cells $\theta_A : i_A^* \cong 1_{TA}$.

This data has also to satisfy two coherence conditions [Fio+17].

We set $(-)^* = \langle (-)^*_{A,B} \rangle_{A,B \in \mathcal{C}}$, $i = \langle i_A \rangle_{A \in \mathcal{C}}$, $\mu = \langle \mu_{F,G} \rangle_{F,G \in \mathcal{C}(A,B)}$, $\eta = \langle \eta_F \rangle_{F \in \mathcal{D}(JA, TB)}$ and $\theta = \langle \theta_A \rangle_{A \in \mathcal{C}}$.

Remark 2.3.5 (Relative Pseudomonads from Relative Pseudoadjunctions). *Given a relative (left) pseudoadjunction we can define its associated relative pseudomonads, following the construction presented in [Fio+17][Theorem 3.8]. Given a left pseudoadjunction*

$$\begin{array}{ccc} & & \mathcal{E} \\ & \nearrow \Phi & \downarrow \Psi \\ \mathcal{C} & \xrightarrow{J} & \mathcal{D} \end{array}$$

2. Categorical Interlude – 2.3. Two-Dimensional Monad Theory

we set $TA = \Psi\Phi A$. The data of the relative pseudoadjunction gives 1-cells $i_A : A \rightarrow TA$ and adjoint equivalences

$$\begin{array}{ccc}
 & \xrightarrow{(-)^b} & \\
 \mathcal{D}(JA, \Psi\Phi B) & \perp & \mathcal{E}(\Phi A, \Phi B) \\
 & \xleftarrow{\Psi(-) \circ i_A} &
 \end{array}$$

The 2-cells are build form the unit and counit of the family of adjoint equivalences that determines the pseudoadjunction.

The relative pseudomonads have Kleisli bicategories, built in a completely natural way:

Proposition 2.3.6 (Kleisli Bicategory, Fiore-Gambino-Hyland-Winskel). *Given a relative pseudomonad $T = \langle \langle TA \rangle_{A \in \mathcal{C}}, (-)^{ast}, i, \mu, \eta, \theta \rangle$ over $J : \mathcal{C} \rightarrow \mathcal{D}$ there is a bicategory $Kl(T)$ called the Kleisli Bicategory of T such that*

$$\text{ob}(Kl(T)) = \text{ob}(\mathcal{C}) \quad Kl(T)(A, B) = \mathcal{D}(JA, TB)$$

identities are given by the family i and, for $F \in Kl(T)(A, B), G \in Kl(T)(B, C)$ horizontal composition is given as follows:

$$G \circ_{Kl(T)} F = G^* \circ_{\mathcal{D}} F.$$

with associativity and identities isomorphisms:

$$\begin{array}{ccc}
 (H^* \circ G)^* \circ F & \xrightarrow{\mu_{H \circ G, F}} & (H^* \circ G^*) \circ F \xrightarrow{\alpha_{H^* \circ G^*, F}^{\mathcal{D}}} (H^* \circ (G^* \circ F)) \\
 i_A^* \circ F & \xrightarrow{\theta_A \circ F} 1_{TA} \circ F \xrightarrow{\lambda_F^{\mathcal{D}}} F & F \circ i_A^* \xrightarrow{F \circ \theta_A} F \circ 1_{TA} \xrightarrow{\rho_F^{\mathcal{D}}} F
 \end{array}$$

2.3.3. (Pseudo) Algebras Lifting of Relative Pseudomonads

Given a pseudofunctor $S : \mathcal{D} \rightarrow \mathcal{D}$ and a pseudofunctor $J : \mathcal{C} \rightarrow \mathcal{D}$ we say that S restricts to \mathcal{C} if for all $A \in \mathcal{C}, SA \in \mathcal{C}$. We denote $S_{\mathcal{C}}$ the restriction of S to \mathcal{C} , defined in the natural way as

$$S_{\mathcal{C}}(A) = S(A).$$

2. Categorical Interlude – 2.3. Two-Dimensional Monad Theory

We say that a 2-monad S *restricts along* J if its endofunctor restricts to \mathcal{C} , $\mu_A \in \mathcal{C}(SSA, SA)$, $\eta_A \in \mathcal{C}(A, SA)$ and the following diagram commutes

$$\begin{array}{ccc} \mathcal{C} & \xrightarrow{J} & \mathcal{D} \\ S|_{\mathcal{C}} \downarrow & & \downarrow S \\ \mathcal{C} & \xrightarrow{J} & \mathcal{D} \end{array}$$

If S restricts along J , the restriction induces pseudofunctors $J^{S,PsA} : S\text{-PAlg}_{\mathcal{C}} \rightarrow S\text{-PAlg}_{\mathcal{D}}$, $J^{S,A} : S\text{-Alg}_{\mathcal{C}} \rightarrow S\text{-Alg}_{\mathcal{D}}$ making the following diagram commute

$$\begin{array}{ccc} S\text{-Alg}_{\mathcal{C}} & \xrightarrow{J^{S,A}} & S\text{-Alg}_{\mathcal{D}} \\ \downarrow & & \downarrow \\ S\text{-PAlg}_{\mathcal{C}} & \xrightarrow{J^{S,PsA}} & S\text{-PAlg}_{\mathcal{D}} \\ \Upsilon \downarrow & & \downarrow \Upsilon \\ \mathcal{C} & \xrightarrow{J} & \mathcal{D} \end{array}$$

Where Υ denotes the evident forgetful functors. From now on, we shall call just J the former two pseudofunctors.

Definition 2.3.7. *Given a relative pseudomonad T over the pseudofunctor $J : \mathcal{C} \rightarrow \mathcal{D}$ and a 2-monad S over \mathcal{D} that restricts along J , A lifting of T to pseudoalgebras of S is the collection of the following data:*

- For every $\mathbb{A} = \langle A, h_A \rangle \in S\text{-PAlg}_{\mathcal{C}}$ a pseudoalgebra structure on $T\mathbb{A}$, denoted as $T\mathbb{A} \in S\text{-PAlg}_{\mathcal{D}}$.
- For every pseudo morphism $\varphi = \langle f, \sigma \rangle : \mathbb{A} = \langle A, h_A \rangle \rightarrow \mathbb{B} = \langle B, h_B \rangle$ a pseudomorphism structure on $f^* : J\mathbb{A} \rightarrow T\mathbb{B}$, denoted as $\varphi^* : J\mathbb{A} \rightarrow T\mathbb{B}$.
- For every $\mathbb{A} = \langle A, h_A \rangle$, a pseudomorphism structure on $i_A : A \rightarrow T\mathbb{A}$, denoted as $i_{\mathbb{A}} : \mathbb{A} \rightarrow T\mathbb{A}$ such that
- $\mu_{f,g} : (g^* \circ f)^* \rightarrow g^* \circ f^*$ is an algebra 2-cell for every pseudomorphism $\langle f, \sigma \rangle^* : J\mathbb{A} \rightarrow T\mathbb{B}$, $\langle g, \tau \rangle^* : J\mathbb{B} \rightarrow T\mathbb{C}$.
- $\eta_f : f \rightarrow f^* \circ i_A$ is an algebra 2-cell for every pseudomorphism $\langle f, \sigma \rangle : J\mathbb{A} \rightarrow T\mathbb{B}$.
- $\theta_A : i_A^* \rightarrow 1_{T\mathbb{A}}$ is an algebra 2-cell for every $\mathbb{A} = \langle A, h_A \rangle \in S\text{-PAlg}_{\mathcal{C}}$.

If a pseudomonad T over $J : \mathcal{C} \rightarrow \mathcal{D}$ admits a lifting to (pseudo) algebras of S , then T induces an evident relative pseudomonad \bar{T} over $J : S\text{-PAlg}_{\mathcal{C}} \rightarrow S\text{-PAlg}_{\mathcal{D}}$. The data of this relative pseudomonad are given directly by the definition of lifting and of pseudoalgebras. One only needs to check that the construction $(-)^*$ induces a functor and that the coherences hold.

2. Categorical Interlude – 2.3. Two-Dimensional Monad Theory

The following theorem is the fundamental technical tool that we are going to use in Section 2.8 to build the family of Kleisli bicategories of distributors that will constitute the mathematical setting for our semantic investigations in Chapter 3 and 4.

Theorem 2.3.8 (Fiore-Gambino-Hyland-Winskel). *If a relative pseudomonad T over $J : \mathcal{C} \rightarrow \mathcal{D}$ admits a lifting to either algebras or pseudoalgebras of $S = \langle S, \eta, \mu \rangle$, then S can be extended to a relative pseudomonad over the identity $1_{Kl(T)} : Kl(T) \rightarrow Kl(T)$.*

Proof. We recall the proof of [Fio+17], since the construction of the considered pseudomonad is relevant for Section 2.8. We do the case of pseudoalgebras, the strict one being completely analogous. Consider the Kleisli bicategory of the relative pseudomonad \bar{T} over $J : S\text{-PAlg}_{\mathcal{C}} \rightarrow S\text{-PAlg}_{\mathcal{D}}$. Objects of $Kl(\bar{T})$ are strict algebras and homcategories are defined as

$$Kl(\bar{T})(\mathbb{A}, \mathbb{B}) = S\text{-PAlg}_{\mathcal{D}}(J\mathbb{A}, T\mathbb{B}).$$

We remark that there is an evident forgetful pseudofunctor $\Upsilon : Kl(\bar{T}) \rightarrow Kl(T)$. In particular then, we have families of functors

$$\begin{array}{ccc} Kl(\bar{T})(\mathbb{A}, \mathbb{B}) & \xrightarrow{\Upsilon_{\mathbb{A}, \mathbb{B}}} & Kl(T)(\Upsilon\mathbb{A}, \Upsilon\mathbb{B}) \\ \parallel & & \parallel \\ S\text{-PAlg}_{\mathcal{D}}(J\mathbb{A}, T\mathbb{B}) & \xrightarrow{\Upsilon_{\mathbb{A}, \mathbb{B}}} & \mathcal{D}(JA, TB) \end{array}$$

where we suppose that $\mathbb{A} = \langle A, h_A \rangle$ and $\mathbb{B} = \langle B, h_B \rangle$. We claim that Υ has a left biadjoint Φ . We build the appropriate data for Φ .

- For $A \in Kl(T)$, we set $\Phi A = \langle SA, h_{SA} : SSA \rightarrow SA \rangle$, that is the free algebra on A , which, in particular, is a pseudoalgebra.
- For $A \in Kl(T)$ we define a morphism $\tilde{e}_A \in \mathcal{C}(A, SA)$ as the composite

$$JA \xrightarrow{\eta_{JA}} SJA = JSA \xrightarrow{i_{SA}} TSA$$

in \mathcal{D} , where η_{JA} is the unit of the 2-monad S . We have the following diagram

$$\begin{array}{ccc} Kl(T)(A, B) & \xleftarrow{\Upsilon(-) \circ \tilde{e}_A} & Kl(\bar{T})(\Phi A, \mathbb{B}) \\ \parallel & & \parallel \\ \mathcal{D}(JA, TB) & \xleftarrow{\Upsilon(-) \circ e_A} & S\text{-PAlg}_{\mathcal{D}}(\Phi JA, T\mathbb{B}) \end{array}$$

where we suppose that the underling object of \mathbb{B} is B . The former diagram commutes up to natural isomorphism by the following property of \tilde{e}_A . Given

2. Categorical Interlude – 2.4. Kan Extensions and Coends

a pseudomorphism $\varphi = \langle F, \zeta \rangle : \Phi JA \rightarrow T\mathbb{B}$ we get the following isomorphism

$$F \circ \tilde{e}_A = F^* \circ \tilde{e}_A = (F^* \circ i_{SA}) \circ \eta_A \cong F \circ \eta_A$$

now, since $\Upsilon(-) \circ e_A : S\text{-PAlg}_{\mathcal{D}}(\Phi JA, T\mathbb{B}) \rightarrow \mathcal{D}(JA, TB)$ defines an equivalence of categories, we get in particular a relative left pseudoadjunction

$$\begin{array}{ccc} & & Kl(\bar{T}) \\ & \nearrow \Phi & \downarrow \Upsilon \\ Kl(T) & \longleftarrow & Kl(T) \end{array}$$

then following the construction presented in Remark 2.3.5, we get our relative pseudomonad. □

2.4. Kan Extensions and Coends

We recall in this section two standard universal categorical constructions: *coends* and (*left*) *Kan extensions*. Both construction arises fairly often when reasoning about categories and they are strictly related to some colimit constructions. They are also interrelated concept: under some conditions, we can compute left Kan extension with a coend formula.

2.4.1. Coends

Definition 2.4.1. Let $F : C^o \times C \rightarrow D$ be a functor. A *cowedge* of F is an object $T \in D$ together with a family of morphisms $\langle w_c : F(c, c) \rightarrow T \rangle_{c \in C}$ such that the following diagram commutes

$$\begin{array}{ccc} F(c', c) & \xrightarrow{F(f, 1)} & F(c, c) \\ \downarrow F(1, f) & & \downarrow w_c \\ F(c', c') & \xrightarrow{w_{c'}} & T \end{array}$$

for $f : c \rightarrow c'$.

Definition 2.4.2. Let $F : C^o \times C \rightarrow D$ be a functor. A *coend* of F is a cowedge $\langle T, w \rangle$ of F such that for any other cowedge $\langle K, u \rangle$ of F there exists a unique morphism $h : K \rightarrow T$ that makes the following diagram commutes:

2. Categorical Interlude – 2.4. Kan Extensions and Coends

$$\begin{array}{ccc}
 F(c, c') & \xrightarrow{F(c, f)} & F(c, c) \\
 \downarrow F(f, c') & & \downarrow u_c \\
 F(c', c') & \xrightarrow{u_{c'}} & K \\
 & & \dashrightarrow h \\
 & & \int^{c \in C} F(c, c)
 \end{array}
 \begin{array}{l}
 \curvearrowright w_c \\
 \curvearrowleft w_{c'}
 \end{array}$$

Coends are unique up to isomorphism. We denote the coend of F as $\int^{c \in C} F(c, c)$. The integral notation is justified by the formal calculus connected with this notion³.

A coend of a functor $F : C^o \times C \rightarrow D$ is a kind of colimit, precisely the following coequalizer:

$$\sum_{c, c' \in C} C(c', c) \times F(c, c') \begin{array}{c} \xrightarrow{f} \\ \xrightarrow{g} \end{array} \sum_{c \in C} F(c, c) \rightarrow \int^{c \in C} F(c, c).$$

Where the two parallel morphisms are the coproduct of the two following families of morphisms

$$\begin{aligned}
 f_{c, c'} : C(c', c) \times F(c, c') &\rightarrow F(c, c) \\
 \langle f, x \rangle &\mapsto F(c, f)(x)
 \end{aligned}$$

and

$$\begin{aligned}
 g_{c, c'} : C(c', c) \times F(c, c') &\rightarrow F(c', c') \\
 \langle f, x \rangle &\mapsto F(f, c')(x)
 \end{aligned}$$

Since we will work in the Set enriched setting, it is useful to explicitly compute that coequalizer in the case when $D = \text{Set}$. We get the following quotient:

$$\sum_{c \in C} F(c, c) / \sim$$

where \sim is the smallest equivalence relation generated as follows: $\langle c, x \rangle \sim \langle c', x' \rangle$ if there exists $f : c' \rightarrow c$ and $y \in F(c, c')$ such that $F(c, f)(y) = x$ and $F(f, c')(y) = x'$. In particular, given a functor $F : C^o \times C \times D \rightarrow \text{Set}$, we can canonically build a functor $\int^{c \in C} F(c, c, -) : D \rightarrow \text{Set}$ as follows :

$$\begin{aligned}
 e &\mapsto \int^{c \in C} F(c, c, e) \\
 \int^{c \in C} F(c, c, f) : \int^{c \in C} F(c, c, e) &\rightarrow \int^{c \in C} F(c, c, e')
 \end{aligned}$$

3. For a proper introduction to coend calculus see [Lor15].

2. Categorical Interlude – 2.4. Kan Extensions and Coends

$$\widetilde{\langle c, x \rangle} \mapsto \langle c, F(1, 1, f)(x) \rangle$$

where $f : e \rightarrow e'$. The action on morphisms is well-defined since if $\langle c, x \rangle \sim \langle c', x' \rangle$ then $\langle c, F(c, c, f)(x) \rangle \sim \langle c', F(c, c, f)(x') \rangle$ ⁴.

Lemma 2.4.3. *Every cocontinuous functor preserves coends.*

Proof. Immediate, since a coend is a coequalizer. □

Lemma 2.4.4 (Fubini). *Let $F : C^o \times C \times D^o \times D \rightarrow E$ be a functor. We have*

$$\int^{\langle c, d \rangle} F(c, c, d, d) \cong \int^c \int^d F(c, c, d, d) \cong \int^d \int^c F(c, c, d, d)$$

Proof. We refer to [Lor15]. □

Theorem 2.4.5 (Yoneda, Density Theorem). *Let $K, H : C \rightarrow D$ be, respectively, a contravariant and a covariant functor. We have the following natural isomorphisms*

$$K(-) \cong \int^{c \in C} K(c) \times C(-, c). \quad (2.1)$$

$$H(-) \cong \int^{c \in C} H(c) \times C(c, -). \quad (2.2)$$

Proof. For the general proof we refer to [Lor15]. However, we build the natural isomorphism explicitly in the particular case where $D = \text{Set}$. We prove (2.1), the two being dual to each other.

Let $b \in C$. We define a function

$$f_b : K(b) \rightarrow \int^{c \in C} K(c) \times C(b, c)$$

$$x \mapsto \widetilde{\langle b, x, 1_b \rangle}.$$

The injectivity of f_b is immediate by definition. We prove its surjectivity. Let $\langle c, x, f \rangle \in \int^{c \in C} K(c) \times C(b, c)$. By definition of equivalence, $\langle c, x, f \rangle \sim \langle b, K(f)(x), 1_b \rangle$. Then we have that $f_b(K(f)(x)) = \widetilde{\langle c, x, f \rangle}$. In order to prove the naturality, we need to show that the following diagram commutes:

$$\begin{array}{ccc} K(b') & \xrightarrow{K(f)} & K(b) \\ \downarrow f_{b'} & & \downarrow f_b \\ F(b') & \xrightarrow{F(f)} & F(b) \end{array}$$

4. This functor trivially corresponds to the one given by the universal property of the coend construction.

for any $f : b' \rightarrow b$ and $F(-) = \int^{c \in C} K(c) \times C(-, c)$. By definition we have

$$F(f) : \int^{c \in C} K(c) \times C(b', c) \rightarrow \int^{c \in C} K(c) \times C(b, c)$$

$$\widetilde{\langle c, x, g \rangle} \mapsto \widetilde{\langle c, x, g \circ f \rangle}$$

then $F(f) \circ \widetilde{f_b}(x) = \widetilde{\langle b', x, f \rangle}$ and $f_b \circ K(f)(x) = \widetilde{\langle b, K(f)(x), 1_b \rangle}$. By definition of equivalence we have $\widetilde{\langle b, K(f)(x), 1_b \rangle} \sim \widetilde{\langle b', x, f \rangle}$. We can then conclude. \square

In what follows we will constantly refer to the former result as the "Yoneda Lemma". The context will make it clear if we are referring to the standard Yoneda Lemma or to the former result on coends.

2.4.2. Kan Extensions

Definition 2.4.6. *Let $G : A \rightarrow C$ and $F : A \rightarrow B$ be two functors. A left Kan extension of F along G is a functor $L_G(F) : C \rightarrow B$ together with a natural transformation $\eta : F \rightarrow L_G(F) \circ G$ such that, for any $K : C \rightarrow B$ and natural transformation $\theta : F \rightarrow K \circ G$ there exists a unique natural transformation $\sigma : L_G(F) \rightarrow K$ such that $\theta = \eta \star (\sigma \circ G)$.*

The following is a classic result on the existence of Kan extensions:

Theorem 2.4.7. *Let $F : A \rightarrow B$ and $G : A \rightarrow C$ be functors. If A is small and B cocomplete there exists the left Kan extension $\langle L_G(F), \eta \rangle$. Moreover, if G is full and faithful we have $\eta : F \cong L_G(F) \circ G$.*

It is possible to explicitly compute left Kan extensions as a special kind of coends. We first define the notion of *copower* in our Set-enriched setting.

Definition 2.4.8. *Let C be a category and $c \in C, X \in \text{Set}$. The copower of c by X is an object $X \odot c$ such that we have an isomorphism*

$$C(X \odot c, Y) \cong \text{Set}(x, C(X, Y))$$

natural in $Y \in \text{Set}$.

The copower is unique up to isomorphism. In particular we have $(X \times Y) \odot c \cong (X \odot (Y \odot c))$.

Proposition 2.4.9 ([Lor15][Proposition 2.3.5]). *Let $F : A \rightarrow B$ and $G : A \rightarrow C$. Suppose that $L_G(F)$ exists and for all $c \in C$, copowers $C(G(c), -) \odot F(c)$ exist. Then*

$$L_G(F)(-) \cong \int^{c \in C} C(G(c), -) \odot F(c).$$

2.5. The Category of Presheaves

For a small category A define $PA = [A^o, \text{Set}]$, the category of *presheaves* of A and natural transformations. We write y_A for the *Yoneda embedding of the category A in PA* defined as the functor

$$y_A : A \rightarrow PA$$

$$a \mapsto A(-, a)$$

and the action of y_A on morphisms is given by composition. It is well-known that the former functor is full and faithful as a corollary of the Yoneda Lemma. For this reason A can be seen as a full subcategory of PA . The category of presheaves is in particular cocomplete, as corollary of what is known as the *density theorem*, *i.e.* any presheaf is a canonical colimit of representable functors (Theorem 2.1).

2.5.1. Free Cocompletion of (Product) Categories

We present the classic results of [IK86] on the relationship between monoidality and presheaves. We start by proving that PA is the *free cocompletion* of a small category A . This means that given any cocomplete category B and functor F , there exists a unique cocontinuous functor $(F)^*$ such that the following diagram commutes up to natural isomorphism:

$$\begin{array}{ccc} A & \xrightarrow{y_A} & PA \\ F \downarrow & \searrow & \uparrow \\ B & \xleftarrow{(F)^*} & \end{array}$$

The functor $(F)^*$ is the left Kan extension of F along the Yoneda embedding $L_Y(F)$, which, by Theorem 2.4.7 always exists. We are now going to recall the proof of this classic and crucial fact in a slightly more general setting, where A is a finite product category and F is separately cocontinuous.

Let A_1, \dots, A_k be small category and let C be a cocomplete category. Separately cocontinuous functors (Definition 2.1.6) $F : \prod_{i=1}^k P(A_i) \rightarrow C$ together with natural transformations forms a category $\text{SCoc}(\prod_{i=1}^k P(A_i), C)$. We define functors

$$R^k : \text{SCoc}\left(\prod_{i=1}^k P(A_i), C\right) \rightarrow \text{Cat}\left(\prod_{i=1}^k A_i, C\right)$$

$$F \mapsto F \circ \left(\prod_{i=1}^k y_{A_i}\right)$$

2. Categorical Interlude – 2.5. The Category of Presheaves

and

$$L^k : \text{Cat}\left(\prod_{i=1}^k A_i, C\right) \rightarrow \text{SCoc}\left(\prod_{i=1}^k P(A_i), C\right)$$

$$S \mapsto \int^{a_1 \in A_1, \dots, a_k \in A_k} -(a_1) \times \cdots \times -(a_k) \odot S(a_1, \dots, a_k)$$

The functor $L^k(S)(-) : \prod_{i=1}^k P(A_i) \rightarrow C$ is separately cocontinuous (Definition 2.1.6), i.e. for any $i \in [k]$, the functors $L^k(S)(P_1, \dots, P_{i-1}, -, P_{i+1}, \dots, P_k) : PA_i \rightarrow C$ are cocontinuous, since colimits commutes with colimits. We remark that in the case where $k = 1$ we have

$$L^1(S)(P) = \int^{a \in A} P(a) \odot S(a)$$

that is the left Kan extension of S along the Yoneda embedding y_A . We set $R^1 = R$ and $L^1 = L$.

Proposition 2.5.1. *For all $k \in \mathbb{N}$ the functors R^k and L^k describes an equivalence of categories*

$$\begin{array}{ccc} & \xrightarrow{R^k} & \\ \text{SCoc}\left(\prod_{i=1}^k P(A_i), C\right) & & \text{Cat}\left(\prod_{i=1}^k A_i, C\right) \\ & \xleftarrow{L^k} & \end{array}$$

Proof. We follow the proof of [IK86]. First we prove that we have a natural isomorphism

$$\eta_S : RL^k S \cong S. \quad (2.3)$$

By definition and Yoneda we have $R^k L^k S(a'_1, \dots, a'_k) =$

$$\int^{a_1 \in A_1, \dots, a_k \in A_k} y_A(a'_1)(a_1) \times \cdots \times y_A(a'_k)(a_k) \odot S(a_1, \dots, a_k) \cong S(a'_1, \dots, a'_k).$$

Now for arbitrary $P_i \in PA_i$ with $i \in [k]$ we have by Yoneda

$$P_i(-) \cong \int^{a_i \in A_i} P_i(a_i) \times A_i(-, a_i)$$

this means that, if $F : \prod_{i=1}^k PA_i \rightarrow C$, is separately cocontinuous we have

$$F(P_1, \dots, P_k) \cong \int^{a_1 \in A_1, \dots, a_k \in A_k} \prod_{i=1}^k P_i(a_i) \odot F(y_{A_1}(a_1), \dots, y_{A_k}(a_k))$$

then we have a natural isomorphism

$$\epsilon_F : L^k R^k F \cong F. \quad (2.4)$$

□

2.5.2. Day Convolution

In the case where A exhibits a structure of monoidal category, we can define a tensor product on the category $P(A)$, called the *Day convolution* [Day70], exploiting the construction presented in the former paragraph.

Definition 2.5.2 (Day Convolution). *Let $\mathbb{A} = \langle A, \otimes, 1, \alpha, \rho, \lambda \rangle$ be a monoidal category. We define the Day convolution $\hat{\otimes} : PA \times PA \rightarrow PA$ as*

$$L^2(y_A \circ \otimes).$$

By definition,

$$P\hat{\otimes}Q(-) = \int^{a_1, a_2 \in A} P(a_1) \times Q(a_2) \times A(-, a_1 \otimes a_2)$$

since in Set the copower \odot trivially collapses on the cartesian product. Exploiting the unbiased version of the tensor product of A , one can straightforwardly define an n -ary version of the Day convolution:

$$(P_1 \hat{\otimes} \dots \hat{\otimes} P_n)(-) = \int^{a_1, \dots, a_n \in A} \prod_{i \in [n]} P_i(a_i) \times A(-, \bigotimes_{i=1}^n a_i).$$

Proposition 2.5.3. *Let $\mathbb{A} = \langle A, \otimes, 1, \alpha, \lambda, \rho \rangle$ be a monoidal category. The following statements hold.*

1. *The Day convolution exhibits a structure of tensor product on the presheaf category PA .*
2. *If the product of \mathbb{A} is symmetric (resp. semicartesian, relevant, cartesian) then the Day convolution is symmetric (resp. semicartesian, relevant, cartesian).*

Proof. 1. Corollary of Proposition 2.5.1. By definition we have that $y_A \circ \otimes \in \text{Cat}(A \times A, PA)$. Now, consider $R^2L^2(y_A \circ \otimes)(a, b) =$

$$\int^{a_1, a_2 \in A} y_A(a)(a_1) \times y_A(b)(a_2) \times y_A(-)(a_1 \otimes a_2) = y_A(a)(-) \hat{\otimes} y_A(b)(-).$$

Hence, the natural isomorphism (2.3) gives

$$\iota^1(a, b) : y_A(a) \hat{\otimes} y_A(b) \cong y_A(a \otimes b).$$

Let $\otimes_{3,1} = (- \otimes -) \otimes -$ and $\otimes_{3,2} = - \otimes (- \otimes -)$. By Proposition 2.5.1 we know that $L^3(y_A \circ \otimes_{3,1}) \cong L^3R^3L^3(y_A \circ \otimes_{3,1})$, $L^3(y_A \circ \otimes_{3,2}) \cong L^3R^3L^3(y_A \circ \otimes_{3,2})$.

Since

$$R^3L^3(y_A \circ \otimes_{3,1}) = (y_A(-) \hat{\otimes} y_A(-)) \hat{\otimes} y_A(-)$$

2. Categorical Interlude – 2.5. The Category of Presheaves

$$R^3L^3(y_A \circ \otimes_{3,2}) = y_A(-) \hat{\otimes} (y_A(-) \hat{\otimes} y_A(-))$$

if we find a natural family of isomorphisms

$$\hat{\alpha}_{a,b,c} : (y_A(a) \hat{\otimes} y_A(b)) \hat{\otimes} y_A(c) \cong y_A(a) \hat{\otimes} (y_A(b) \hat{\otimes} y_A(c))$$

we get

$$L^3(\hat{\alpha}) : L^3((y_A \hat{\otimes} y_A) \hat{\otimes} y_A) = L^3(y_A \circ \otimes_{3,1}) \cong L^3(y_A \hat{\otimes} (y_A \hat{\otimes} y_A)) = L^3(y_A \circ \otimes_{3,2}).$$

Then we define $\hat{\alpha}_{a,b,c}$ as the composite

$$\begin{array}{ccc} (y_A(a) \hat{\otimes} y_A(b)) \hat{\otimes} y_A(c) & \xrightarrow{\hat{\alpha}_{a,b,c}} & y_A(a) \hat{\otimes} (y_A(b) \hat{\otimes} y_A(c)) \\ \downarrow \iota_{a,b} \hat{\otimes} 1 & & \downarrow 1 \hat{\otimes} \iota_{b,c} \\ y_A(a \otimes b) \hat{\otimes} y_A(c) & & y_A(a) \hat{\otimes} y_A(b \otimes c) \\ \downarrow \iota_{a \otimes b, c} & & \downarrow \iota_{a, b \otimes c} \\ y_A((a \otimes b) \otimes c) & \xrightarrow{y_A(\alpha_{a,b,c})} & y_A(a \otimes (b \otimes c)) \end{array}$$

we take as unit $\hat{1} = y_A(1)$. We set

$$\iota_0 : \hat{1} \cong y_A(1)$$

as the identity. We then define the natural isomorphisms $\hat{\lambda}, \hat{\rho}$ exploiting Proposition 2.5.1 in a similar way as for associativity case. The commutation of coherence diagrams follows directly by the former construction, exploiting again Proposition 2.5.1. We remark that the natural isomorphisms ι^0, ι^1 gives to the Yoneda embedding the structure of a strong monoidal functor.

2. We need to respectively define a natural isomorphism

$$\hat{\sigma}_{P,Q} : P \hat{\otimes} Q \rightarrow Q \hat{\otimes} P$$

and natural transformations

$$\hat{e}_P : P \rightarrow \hat{1} \quad \hat{c}_P : P \rightarrow P \hat{\otimes} P$$

satisfying the appropriate coherence conditions. We proceed as for the first point of this lemma, defining the appropriate structure on representables. For instance, the symmetry $\hat{\sigma}_{y_A(a), y_A(b)} : y_A(a) \hat{\otimes} y_A(b) \rightarrow y_A(b) \hat{\otimes} y_A(a)$ is given as

2. Categorical Interlude – 2.5. The Category of Presheaves

the composite

$$\begin{array}{ccc} y_A(a) \hat{\otimes} y_A(b) & \xrightarrow{\hat{\sigma}_{y_A(a), y_A(b)}} & y_A(b) \hat{\otimes} y_A(a) \\ \iota_{a,b}^1 \downarrow & & \downarrow \iota_{b,a}^1 \\ y_A(a \otimes b) & \xrightarrow{y_A(\sigma_{a,b})} & y_A(b \otimes a) \end{array}$$

and the diagonal as the composite

$$\begin{array}{ccc} y_A(a) & \xrightarrow{\hat{c}_{y_A(a)}} & y_A(a) \hat{\otimes} y_A(a) \\ \iota_a^1 \downarrow & & \downarrow \iota_{b,a}^1 \\ y_A(a) & \xrightarrow{y_A(c_a)} & y_A(a \otimes a) \end{array}$$

For the cartesian case, we have the following isomorphism

$$\begin{aligned} P \hat{\otimes} Q &= \int^{a_1, a_2 \in A} P(a_1) \times Q(a_2) \times A(-, a_1 \otimes a_2) \\ &\cong P \hat{\otimes} Q = \int^{a_1, a_2 \in A} P(a_1) \times Q(a_2) \times A(-, a_1) \times A(-, a_2) \end{aligned}$$

given by the fact that the cartesian product \otimes is left adjoint to the diagonal functor. Then we can apply Yoneda twice and conclude

$$\cong P(-) \times Q(-)$$

that is the cartesian product of presheaves. □

We set $P\mathbb{A} = \langle PA, \hat{\otimes}, \hat{1}, \hat{\alpha}, \hat{\lambda}, \hat{\rho} \rangle$. For a monoidal category $\mathbb{A} = \langle A, \otimes, 1 \rangle$ we set $Y_{\mathbb{A}} = \langle Y_A, \iota_0, \iota_1 \rangle$.

Proposition 2.5.4. *Let $\mathbb{A} = \langle A, \otimes, 1 \rangle$ be a monoidal category. If \mathbb{A} is symmetric (resp. semicartesian, relevant, cartesian) then the Yoneda embedding is a strong symmetric (resp. semicartesian, relevant, cartesian) monoidal functor.*

Proof. We already mentioned in the proof of Proposition 2.5.3 that Y_A is strong monoidal. The results are immediate by definition of symmetries, diagonals, terminal morphisms in the category of presheaves PA . For instance, in the case of symmetries we need to prove that the following diagram commutes

$$\begin{array}{ccc} y_A(a) \hat{\otimes} y_A(b) & \xrightarrow{\hat{\sigma}_{a,b}} & y_A(b) \hat{\otimes} y_A(a) \\ \iota_{a,b}^1 \downarrow & & \downarrow \iota_{b,a}^1 \\ y_A(a \otimes b) & \xrightarrow{y_A(\sigma_{a,b})} & y_A(b \otimes a) \end{array}$$

but the former diagram is exactly the definition of symmetries for representables in the presheaf category. The same happens for diagonals. The semicartesian and cartesian cases follow by the fact that the Yoneda embedding is continuous. \square

Theorem 2.5.5. *Let \mathbb{A} be a small monoidal category and \mathbb{B} be a locally small monoidally cocomplete category (Definition 2.1.6). We have the following equivalence*

$$\begin{array}{ccc} & L & \\ & \curvearrowright & \\ \text{MON}(\mathbb{A}, \mathbb{B}) & & \text{MONCOC}(P\mathbb{A}, \mathbb{B}) \\ & \curvearrowleft & \\ & R & \end{array}$$

The former equivalence restricts to the symmetric (resp. semicartesian, relevant, cartesian) case.

Proof. The proof for the basic case and the symmetric one is given in [IK86]. The other cases follow as a simple corollary of Proposition 2.5.1. The proof follows the following structure:

1. We prove that the functor $R = U(-) \circ Y_{\mathbb{A}}$ is full and faithful. In order to do so we exploit Proposition 2.5.1.
2. We prove that the left Kan extension $L_{Y_A}(F) = L(F)$ can be equipped with a strong monoidal structure, exploiting the natural isomorphism $\eta_F : RL(F) \cong F$ given in the proof of Proposition 2.5.1. This makes the functor R essentially surjective on objects.
3. We observe that, again by Proposition 2.5.1, it is enough to define the appropriate structure for $L(F)$ restricting its domain to representable presheaves. Then the symmetric (resp. semicartesian, relevant, cartesian) structure on $L(F)$ is defined each time just as a composite determined by the image of the Yoneda embedding.

\square

Lemma 2.5.6 (Fiore-Gambino-Hyland-Winskel). *The presheaf construction PA induces a relative pseudomonad on the inclusion 2-functor $J : \text{Cat} \rightarrow \text{CAT}$.*

Proof. The proof consists in presenting the necessary structure. We detail some of it. For a small category $A \in \text{Cat}$, we set PA as the category of presheaf of A . For $A, B \in \text{Cat}$ the functor $(-)^*_{A,B} : \text{CAT}(JA, PB) \rightarrow \text{CAT}(PA, PB)$ is induced by the left Kan extension construction along the Yoneda embedding, *i.e.* $(F)^* = L_Y(F)$. The morphism $i_A : JA \rightarrow PA$ is the Yoneda embedding of A . \square

2.6. Monads and Resources

The goal of this section is to present some monadic construction that we shall utilise to model resource consumption. We start by recalling the *Boom Hierarchy*

of *data types* [Bun94], for which we give an intuitive monadic interpretation. We then introduce the categories of integers and lists presenting some of their basic structure. Finally we introduce a collection of 2-monads that we call resource monads, which we will use (Section 2.8) to build a family of Kleisli bicategories over the bicategory of distributors (Section 2.7.3).

2.6.1. Boom Hierarchy of Data Types

The Boom Hierarchy [Bun94] is an algebraic model of fundamental data types in computer science. We recall it in this paragraph, since it is useful to understand our semantic standpoint.

We consider sets equipped with a binary operation \oplus , that can satisfy one or more of the following conditions:

1. For all $x, y, z \in X$, $(x \oplus y) \oplus z = x \oplus (y \oplus z)$. (Associativity)
2. There exists $1 \in X$ such that for all $x \in X$, $1 \oplus x = x = x \oplus 1$. (Unity)
3. For all $x, y \in X$, $x \oplus y = y \oplus x$. (Commutativity)
4. For all $x \in X$, $x \oplus x = x$. (Idempotency)

If \oplus satisfy only (1), then we are dealing with semigroups, if it satisfies (1,2) with monoids, if it satisfies (1,2,3) with commutative monoids and if it satisfies the four of them with commutative idempotent monoid. However, in principle, one could consider just operation which satisfy, *e.g.*, the unital condition (2) and not the others. Each condition generates then a variety of algebras. Using this very broad framework, it is possible to give a simple algebraic model for the 4 basic *data structures* : trees, lists, multisets and sets.

When we are talking about algebraic constructions we are talking about monads. The standard combinations of the former conditions that we recalled, semigroups, monoids, etc. determine monads on the category of Set. Each data structure can be then identified with an appropriate monadic construction.

- Given a set X , the *tree structure on X* is given by $\mathcal{T}X$, where $\mathcal{T} : \text{Set} \rightarrow \text{Set}$ is the endofunctor associated to the monad corresponding to the condition (2).
- Given a set X , the *list structure on X* is given by $\mathcal{L}X$, where $\mathcal{L} : \text{Set} \rightarrow \text{Set}$ is the endofunctor associated with the free monoid monad. The unit element is the *empty list* and the operation \oplus performs list concatenation.
- Given a set X , the *multiset structure on X* is given by $\mathcal{M}X$, where $\mathcal{M} : \text{Set} \rightarrow \text{Set}$ is the endofunctor associated with the free commutative monoid monad. The unit element is the *empty multiset* and the operation \oplus performs multiset sum.
- Given a set X , the *set structure on X* is given by $\mathcal{S}X$, where $\mathcal{S} : \text{Set} \rightarrow \text{Set}$ is the endofunctor associated with the free idempotent commutative monoid

monad. The unit element is the empty set and the operation \oplus performs the set union.

A *data structure value* is then an element of the former free constructions. This means that, in all generality, a value is either a unit 1 or a "join" of two other values $x \oplus y$ (eventually a singleton join, when one of the two values is the unit).

The former model is very simple and effective, but, due to its one-dimensional nature, cannot directly express the possible operation that one can perform on the data values. Consider the case of lists: given a set X , and $\vec{a} = \langle a_1, \dots, a_k \rangle, \vec{b} = \langle a'_1, \dots, a'_{k'} \rangle \in \mathcal{L}X$, we know that the concatenation $\vec{a} \oplus \vec{b}$ is not commutative, but clearly one can eventually perform a *shuffle operation* $\sigma_{\vec{a}, \vec{b}} : \vec{a} \oplus \vec{b} \rightarrow \vec{b} \oplus \vec{a}$. This works for any $\vec{a}, \vec{b} \in \mathcal{L}X$. Again, the concatenation is non-idempotent, but clearly we can perform a *copying operation* $c_{\vec{a}} : \vec{a} \rightarrow \vec{a} \oplus \vec{a}$ and *deleting operations* $\pi_1 : \vec{a}_1 \oplus \vec{a}_2 \rightarrow \vec{a}_1, \pi_2 : \vec{a}_1 \oplus \vec{a}_2 \rightarrow \vec{a}_2$. We shall see that these operations gives to \oplus some special kind of *tensor product* structure. Moreover, this new framework can model *resource sensitivity*. If we just allow a shifting operation on lists but neither copying or deleting, the data can be used only once, hence we do not have a potentially infinite access to resources. Instead, if we allow the copying operation, we can eventually duplicate data and, for this reason we can use resources as many times as we want.

In what follows, we are going to present a categorical model for this refined framework in the setting of two-dimensional monad theory, that was first introduced in [MZ18]. We shall focus on the list data structure, keeping the operational point of view discussed above.

2.6.2. Integers and Lists

We define the category \mathbb{O}_f of *finite ordinals* and functions.

1. $\text{ob}(\mathbb{O}_f) = \{[n] = \{1, \dots, n\} \mid n \in \mathbb{N}\}$.
2. $\mathbb{O}_f([n], [m]) = [m]^{[n]}$.
3. the category \mathbb{O}_f is symmetric strict monoidal, with tensor product given by addition:

$$[n] \oplus [m] = [n + m]$$

Let $\alpha : [k_1] \rightarrow [k'_1]$ and $\beta : [k_2] \rightarrow [k'_2]$, then

$$(\alpha \oplus \beta)(i) = \begin{cases} \alpha(i) & \text{if } i \leq k_1 \\ \beta(i - k_1) + k'_1 & \text{otherwise.} \end{cases}$$

Let k_1, \dots, k_n be integers and $\alpha : [m] \rightarrow [n]$ we define $\bar{\alpha} : [\sum_{j=1}^m k_{\alpha(j)}] \rightarrow [\sum_{i=1}^n k_i]$ as follows:

2. Categorical Interlude – 2.6. Monads and Resources

$$\bar{\alpha}\left(\sum_{j=1}^{l-1} k_{\alpha(j)} + p\right) = \sum_{i=1}^{\alpha(l)-1} k_i + p$$

with $l \in [m]$, and $1 \leq p \leq k_{\alpha(l)}$. We remark that this operation is just a generalization of the notion of multiplexing from Subsection 1.5.3.

From \mathbb{O}_f we can build categories of indexed families of objects over finite ordinals. Let $\langle a_1, \dots, a_k \rangle$ be a list of elements of A . We write $\text{len}(\vec{a})$ for its length. We denote lists as $\vec{a}, \vec{b}, \vec{c} \dots$. Given a list $\vec{a} = \langle a_1, \dots, a_k \rangle$ and a function $\alpha : [k] \rightarrow [k']$ we define the *right action of α on \vec{a}* as $\vec{a}\{\alpha\} = \langle a_{\alpha(1)}, \dots, a_{\alpha(k)} \rangle$. Given a category A , we define the category $\mathbb{O}_f A$ of *lists* of A , as follows:

1. $\text{Obj}(\mathbb{O}_f A) = \{\langle a_1, \dots, a_n \rangle \mid a_i \in A\}$.
2. $\mathbb{O}_f A(\langle a_1, \dots, a_n \rangle, \langle b_1, \dots, b_m \rangle) = \{\langle \alpha, f_1, \dots, f_m \rangle \mid \alpha : [m] \rightarrow [n] \text{ and } f_i : a_{\alpha(i)} \rightarrow b_i\}$.
3. For $\langle \alpha, \vec{f} \rangle : \vec{a} \rightarrow \vec{b}$ and $\langle \beta, \vec{g} \rangle : \vec{b} \rightarrow \vec{c}$, composition is given by

$$\langle \beta, \vec{g} \rangle \circ \langle \alpha, \vec{f} \rangle = \langle \alpha \circ \beta, \vec{g} \circ \vec{f}\{\alpha\} \rangle$$

The category $\mathbb{O}_f A$ is monoidal strict, with tensor product given by list concatenation. We consider also the category of *tuples* of A , A^n . We denote as $\vec{f}, \vec{g} \dots$ the morphisms of this category, being simply tuple of morphisms.

4. We exhibit some of the structure of $\mathbb{O}_f A$ that we will use later.
 - a) *Symmetries*: for $\vec{a}_1, \dots, \vec{a}_k \in \mathbb{O}_f A$ with $\text{len}(\vec{a}_i) = k_i$ and $\sigma \in S_k$ symmetries

$$\sigma^* : \bigoplus_{i=1}^k \vec{a}_i \rightarrow \bigoplus_{i=1}^k \vec{a}_{\sigma(i)}$$

are defined as $\langle \vec{\sigma}, 1_{a_1}, \vec{1}_{\bigoplus_{i=1}^k \vec{a}_{\sigma(i)}} \rangle$.

- b) *Diagonals*: for $\vec{a} = \langle a_1, \dots, a_n \rangle \in \mathbb{O}_f A$ there is a morphism

$$c_{\vec{a}} = \langle c_{[n]}, 1_{a_1}, \dots, 1_{a_n}, 1_{a_1}, \dots, 1_{a_n} \rangle : \vec{a} \rightarrow \vec{a} \oplus \vec{a}$$

where $c_{[n]} : [n] \oplus [n] \rightarrow [n]$ is a surjective function, defined in the natural way as

$$c_{[n]}(i) = \begin{cases} i & \text{if } i \leq n \\ i - n & \text{otherwise.} \end{cases}$$

- c) *Terminal morphisms*: for $\vec{a} \in \mathbb{O}_f A$, there is a unique morphism

$$\top_{\vec{a}} : \vec{a} \rightarrow \langle \rangle$$

in particular, the empty list is a terminal object.

2. Categorical Interlude – 2.6. Monads and Resources

- d) *Projections*: for $\vec{a} = \langle a_1, \dots, a_n \rangle, \vec{b} = \langle b_1, \dots, b_m \rangle \in \mathbb{O}_f A$, there are canonical morphisms

$$\pi_{1,2} = \langle p_{1,2}, 1_{a_1}, \dots, 1_{a_n} \rangle : \vec{a} \oplus \vec{b} \rightarrow \vec{a}$$

$$\pi_{2,2} = \langle p_{2,2}, 1_{b_1}, \dots, 1_{b_m} \rangle : \vec{a} \oplus \vec{b} \rightarrow \vec{b}$$

where $p_{1,2} : [n] \rightarrow [n] \oplus [m]$ and $p_{2,2} : [m] \rightarrow [n] \oplus [m]$ are the inclusion functions. The former data give to \oplus the structure of a cartesian product.

- e) *Structural morphisms*: For $\langle \vec{a}_1, \dots, \vec{a}_n \rangle$ and $\alpha : [m] \rightarrow [n]$ with $\text{len}(\vec{a}_i) = k_i$ we define

$$\alpha^* : \bigoplus_{i=1}^n \vec{a}_i \rightarrow \bigoplus_{j=1}^m \vec{a}_{\alpha(j)}$$

as $\alpha^* = \langle \bar{\alpha}, \vec{1}_{\bigoplus_{i=1}^k \vec{a}_{\alpha(i)}} \rangle$. Structural morphisms intuitively are all morphisms generated by composition of the free structure of $\mathbb{O}_f A$ (symmetries, projections, etc). We remark that α^* is a natural transformation $\alpha^* = \{ \alpha_{\langle \vec{a}_1, \dots, \vec{a}_k \rangle}^* \mid \alpha_{\langle \vec{a}_1, \dots, \vec{a}_k \rangle}^* : \bigotimes_{j=1}^k \vec{a}_j \rightarrow \bigotimes_{j'=1}^{k'} \vec{a}_{\alpha(j')} \}$. We shall constantly keep the list parameter implicit.

Remark 2.6.1. *We remark that finite product categories $\mathbb{O}_f A_1 \times \dots \times \mathbb{O}_f A_n$ admit a strict tensor product, defined as*

$$\langle \vec{a}_1, \dots, \vec{a}_n \rangle \otimes \langle \vec{a}'_1, \dots, \vec{a}'_n \rangle = \langle \vec{a}_1 \oplus \vec{a}'_1, \dots, \vec{a}_n \oplus \vec{a}'_n \rangle.$$

The unit is clearly $\langle \rangle = \langle \langle \rangle, \dots, \langle \rangle \rangle$. In particular, the former tensor product inherits all the structure from \oplus in the natural way. We use capital Greek letters $\Gamma, \Delta \dots$ to denote tuples. Given a tuple $\Gamma = \langle \vec{a}_1, \dots, \vec{a}_n \rangle$ and a sequence of tuples $\vec{\Gamma} = \langle \Gamma_1 = \langle \vec{a}_{1,1}, \dots, \vec{a}_{1,n} \rangle, \dots, \Gamma_k = \langle \vec{a}_{k,1}, \dots, \vec{a}_{k,n} \rangle \rangle$ we set

$$\Gamma \otimes \vec{\Gamma} = \Gamma \otimes \bigotimes_{i=1}^k \Gamma_i.$$

The $(-)^*$ construction on lists lifts to this setting in the natural way. We then have a natural transformation $\alpha^* = \{ \alpha_{\langle \Gamma_1, \dots, \Gamma_k \rangle}^* \mid \alpha_{\langle \Gamma_1, \dots, \Gamma_k \rangle}^* : \bigotimes_{j=1}^k \Gamma_j \rightarrow \bigotimes_{j'=1}^{k'} \Gamma_{\alpha(j')} \}$. Given a permutation τ , we shall constantly denote τ^* as $\vec{\tau}$. We shall constantly keep the list parameter implicit.

We introduce some notation on lists that we shall use extensively in Chapters 3 and 4. Let $\vec{a} = \langle a_1, \dots, a_k \rangle$. We write $\vec{a}^{\nearrow a_i}$ meaning that a_i is the i -th element of \vec{a} . When we do not want to make explicit all the elements of a list \vec{a} , we will denote the i -th element of \vec{a} as $\vec{a}(i)$. Given $b \in A$, we set \vec{a}^{+b_i} as the list such that

$\text{len}(\vec{a}^{+b_i}) = \text{len}(\vec{a}) + 1$ and

$$\vec{a}^{+b_i}(j) = \begin{cases} \vec{a}(j) & \text{if } j < i \\ b & \text{if } i = j \\ \vec{a}(j - 1) & \text{if } i < j. \end{cases}$$

We set $\vec{a}^{-a_i} = \langle a_1, \dots, a_{i-1}, a_{i+1}, \dots, a_k \rangle$.

2.6.3. Resource Monads

We start by giving a canonical presentation of some free monoidal constructions.

Proposition 2.6.2. *For $A \in \text{Cat}$ and $\vec{a}, \vec{b} \in \mathbb{O}_f A$ with $n = l(\vec{a}), m = l(\vec{b})$ we define*

$$\mathbb{O}_f A^*(\vec{a}, \vec{b}) = \sum_{\alpha: [m] \rightarrow [n]} \prod_{i \in [m]} A(a_{\alpha(i)}, b_i)$$

for $\alpha: [m] \rightarrow [n]$ being restricted either to general functions, bijections, surjections, injections or identities. The following holds:

1. If α is restricted to identities, then $\mathbb{O}_f A^*(\vec{a}, \vec{b})$ is the homset of the free strict monoidal category on A .
2. If α is restricted to bijections, then $\mathbb{O}_f A^*(\vec{a}, \vec{b})$ is the homset of the free symmetric strict monoidal category on A .
3. If α is restricted to injections, then $\mathbb{O}_f A^*(\vec{a}, \vec{b})$ is homset of the free semi-cartesian monoidal strict category on A .
4. If α is restricted to surjections, then $\mathbb{O}_f A^*(\vec{a}, \vec{b})$ is the homset of free relevant monoidal strict category on A .
5. If α is a general function then $\mathbb{O}_f A^*(\vec{a}, \vec{b})$ is the homset of the free cartesian monoidal strict category on A .

Proof. The proof exploits the fact that each $\mathbb{O}_f A^*(\vec{a}, \vec{b})$ defines a subcategory of $\mathbb{O}_f A$. The unit $\eta_A: A \rightarrow \mathbb{O}_f A^*$ is given by the singleton embedding

$$a \mapsto \langle a \rangle$$

we give the general structure of the proof.

Let \mathbb{B} be a strict monoidal category with the appropriate structure and F be a functor. We need to define a unique strong monoidal functor F^\sharp that preserves the structure on the nose and makes the following diagram commute

$$\begin{array}{ccc} A & \xrightarrow{\eta_A} & \mathbb{O}_f A^* \\ F \downarrow & & \uparrow F^\sharp \\ \mathbb{B} & & \end{array}$$

The general form of $F^\sharp : \mathbb{O}_f A^* \rightarrow \mathbb{B}$ is

$$F^\sharp(\langle a_1, \dots, a_n \rangle) = \bigotimes_{i \in [n]} F(a_i).$$

The action of F^\sharp on morphisms is defined case by case. The unicity is proved pointwise. \square

We present a list of 2-monads over CAT, the 2-category of categories, functors and natural transformations. We call these monads *resource monads*. The intuition is that each of these monadic constructions gives a particular notion of resource management.

1. The *strict monoidal resource monad*: the 2-monad over CAT that sends a category A to its free strict monoidal completion;
2. The *linear resource monad*: the 2-monad over CAT that sends a category A to its free symmetric strict monoidal completion;
3. The *semicartesian resource monad*: the 2-monad over CAT that sends a category A to the free semicartesian strict monoidal category on A ;
4. The *relevant resource monad*: the 2-monad over CAT that sends a category A to the free relevant strict monoidal category on A ;
5. The *cartesian resource monad*: the 2-monad over CAT that sends a category A to its free cartesian strict monoidal completion. One can see it also as the free semicartesian strict monoidal completion with diagonals⁵.

The unit of each resource monad is given by the free construction. Multiplication is just list concatenation: for $\langle \vec{a}_1, \dots, \vec{a}_k \rangle \in SSA$,

$$\mu_A : SSA \rightarrow SA$$

$$\langle \vec{a}_1, \dots, \vec{a}_k \rangle \mapsto \bigoplus_{i=1}^k \vec{a}_i.$$

We call the non strict version of the former 2-monads *non-strict resource monads*.

For S resource monad, we call the *tensor product of S* the tensor product on SA . We call *S -monoidal functor* a functor that preserves the structure on the nose. We denote as S -MON the 2-category of locally small S -monoidal categories, strong S -monoidal functors and monoidal transformations and as S -Mon its full 2-subcategory of small S -monoidal categories. We denote as S -MONCOC the 2-category of locally small S -monoidally cocomplete categories, strong S -monoidal cocontinuous functors and monoidal natural transformations.

Theorem 2.6.3. *Let S be a non-strict resource monad. The relative pseudomonad of presheaves P (Lemma 2.5.6) admits a lifting to the strict algebras of S .*

5. Since products are limits, this construction is equivalent to its non-strict version.

Proof. Corollary of the Theorem 2.5.5. The equivalence

$$\begin{array}{ccc}
 & L & \\
 & \curvearrowright & \\
 S\text{-MON}(\mathbb{A}, \mathbb{B}) & \perp & S\text{-MONCOC}(P\mathbb{A}, \mathbb{B}) \\
 & \curvearrowleft & \\
 & R &
 \end{array}$$

gives a left relative pseudoadjunction

$$\begin{array}{ccc}
 & S\text{-MONCOC} & \\
 & \nearrow P & \downarrow \Upsilon \\
 S\text{-Mon} & \longrightarrow & S\text{-MON}
 \end{array}$$

where Υ is the evident forgetful pseudofunctor, which provide the lifting of P to the pseudoalgebras of S . \square

Theorem 2.6.4. *Let S be a resource monad. The relative pseudomonad of presheaves P (Lemma 2.5.6) admits a lifting to the pseudoalgebras of S .*

Proof. Let S' the 2-monad for the unbiased version of an arbitrary non-strict resource monad S . As remarked in Example 2.3.2, the pseudoalgebras of S are unbiased monoidal categories. In particular, we have an equivalence of 2-categories $S\text{-PAlg}_{\mathcal{D}} \simeq S'\text{-Alg}_{\mathcal{D}}$ ⁶. Moreover, $S\text{-Alg}_{\mathcal{D}}$ is a full 2-subcategory of $S'\text{-Alg}_{\mathcal{D}}$. Hence the lifting of P obtained by the former Theorem restricts to the embedding $S\text{-Alg}_{\mathcal{D}} \hookrightarrow S'\text{-Alg}_{\mathcal{D}}$. \square

Proposition 2.6.5 (Seely Equivalences). *Let $A, B \in \text{Cat}$ and S be a resource monad. If the tensor product of S is symmetric, then we have*

$$\mathbf{1} \cong S\emptyset \tag{2.5}$$

$$S(A \sqcup B) \simeq SA \times SB. \tag{2.6}$$

Proof. The first one is immediate, since $S\emptyset = \{\langle \rangle\}$. We build the equivalence, exploiting the universal property of SA . We define a functor $F : A \sqcup B \rightarrow SA \times SB$ as follows

$$F(\iota_i(c)) = \begin{cases} \langle \langle c \rangle, \langle \rangle \rangle & \text{if } i = 1 \\ \langle \langle \rangle, \langle c \rangle \rangle & \text{if } i = 2. \end{cases}$$

Then the universal property of $S(A \sqcup B)$ gives a functor $\mu_0 = F^\sharp : S(A \sqcup B) \rightarrow SA \times SB$. We define a functor $\mu_1 : SA \times SB \rightarrow S(A \sqcup B)$ by concatenation. The symmetry hypothesis is used to prove that $\mu_1 \circ \mu_0 \cong 1_{S(A \sqcup B)}$. \square

6. This passage is actually more subtle than it seems and depends on the fact that non-strict resource monads are flexible monads, in the sense of [BKP89]. Resource monads fail to be flexible, due to the strictness of their tensor product.

Remark 2.6.6. *We can extend the former proposition to finite products and coproducts of categories*

$$S(A_1 \sqcup \cdots \sqcup A_n) \simeq SA_1 \times \cdots \times SA_n \quad (2.7)$$

we denote the two components of the former equivalence as respectively

$$\mu_0 : S(A_1 \sqcup \cdots \sqcup A_n) \rightarrow SA_1 \times \cdots \times SA_n$$

$$\mu_1 : SA_1 \times \cdots \times SA_n \rightarrow S(A_1 \sqcup \cdots \sqcup A_n).$$

2.7. Relations, Preorders, Distributors

We sketch the structure of some (bi) categories providing categorical models of linear logic. With linear logic categorical model we mean the given of a Seely category [Mel09]. All these (bi) categories are "relations-like categories", in the sense that morphisms are given by some kind of generalization of relations between sets.

2.7.1. The Category of Sets and Relations

A simple model of linear logic is the category Rel of sets and relations. It is a prototype of *quantitative* semantics: the interpretation of a program gives information about its resource consumption during computation.

Objects of Rel are sets, X, Y, \dots and $\text{Rel}(X, Y) = \mathcal{P}(X \times Y)$. Identities are diagonal relations $1_X = \{\langle x, x \rangle \mid x \in X\}$. Composition of morphisms in Rel is the usual composition of relations

$$g \circ f = \{\langle x, z \rangle \mid \exists y \in Y : \langle x, y \rangle \in f, \langle y, z \rangle \in g\} \text{ for } f \subseteq X \times Y \text{ and } g \subseteq Y \times Z.$$

For $X_1, X_2 \in \text{ob}(\text{Rel})$, the cartesian product $X_1 \& X_2$ in Rel is the disjoint union of sets $X_1 \sqcup X_2 = (\{1\} \times X_1) \cup (\{2\} \times X_2)$, where projections $\pi_i : X_1 \& X_2 \rightarrow X_i$ (for $i \in \{1, 2\}$) are defined using the canonical coproduct injections $\{\langle \langle i, x \rangle, x \rangle \mid x \in X_i\}$, and the terminal object \top is the empty set \emptyset .

Rel is a *symmetric monoidal* category, where the tensor $X \otimes Y$ is the cartesian product of sets $X \times Y$ and its unit $\mathbf{1}$ is an arbitrary singleton set. It is *closed*, with $X \multimap Y = X \times Y$ and evaluation $\text{ev}_{X,Y} : (X \multimap Y) \times X \rightarrow Y$ defined by $\{\langle \langle \langle x, y \rangle, x \rangle, y \rangle \mid x \in X, y \in Y\}$.

Rel can be equipped with an *exponential comonad* $\langle M, \text{der}, \text{dig} \rangle$. The set MX is the free commutative monoid over X , (cfr. Section 2.6.1) which has a very well-known description

$$MX = \mathcal{M}_f(X) = \{[a_1, \dots, a_k] \mid k \in \mathbb{N}, a_i \in X\}$$

that is the set of finite multisets over X . The action on morphisms is given as follows: $f \in \text{Rel}(X, Y)$, $Mf = \{\langle [x_1, \dots, x_n], [y_1, \dots, y_n] \rangle \mid n \in \mathbb{N}, \langle x_1, y_1 \rangle, \dots, \langle x_n, y_n \rangle \in f\}$. Dereliction $\text{der}_X \in \text{Rel}(MX, X)$ is $\{\langle [x], x \rangle \mid x \in X\}$, and digging $\text{dig}_X \in \text{Rel}[MX, MMX]$ is $\{\langle m_1 + \dots + m_k, [m_1, \dots, m_k] \rangle \mid m_1, \dots, m_k \in MX\}$ (for two finite multisets $\bar{a} = [a_1, \dots, a_k]$ and $\bar{b} = [b_1, \dots, b_n]$, we set $\bar{a} + \bar{b} = [a_1, \dots, a_k, b_1, \dots, b_n]$). We denote as MRel the coKleisli category of the comonad $!$.

2.7.2. The Category of Preorders and Monotonic Relations

To work within a more informative setting, providing not only *quantitative*, but also *qualitative* information, consider the category Polr of preordered sets and monotonic relations [Ehr12a; Ehr16]. All the constructions in Polr are refinements and generalizations of the ones for Rel .

In Polr , objects are preordered sets; a morphism f from $\mathcal{X} = \langle |\mathcal{X}|, \leq_{\mathcal{X}} \rangle$ to $\mathcal{Y} = \langle |\mathcal{Y}|, \leq_{\mathcal{Y}} \rangle$ is a *monotonic* relation⁷ from $|\mathcal{X}|$ to $|\mathcal{Y}|$, *i.e.*, if $\langle x, y \rangle \in f$ with $x' \leq_{\mathcal{X}} x$ and $y \leq_{\mathcal{Y}} y'$ then $\langle x', y' \rangle \in f$. The identity at \mathcal{X} is $\{\langle x, x' \rangle \mid x \leq_{\mathcal{X}} x'\}$. Composition preserves monotonicity.

In Polr the cartesian product $\mathcal{X}_1 \& \mathcal{X}_2$ is the disjoint union of sets $|\mathcal{X}_1| \sqcup |\mathcal{X}_2|$ with the preorder $\leq_{\mathcal{X}_1 \& \mathcal{X}_2}$ defined as $\langle i, x \rangle \leq_{\mathcal{X}_1 \& \mathcal{X}_2} \langle j, y \rangle$ if $i = j$ and $x \leq_{\mathcal{X}_i} y$. The terminal object is \emptyset with the empty order. Projections $\pi_i: \mathcal{X}_1 \& \mathcal{X}_2 \rightarrow \mathcal{X}_i$ are $\pi_i = \{\langle \langle i, x \rangle, x' \rangle \mid x \leq_{\mathcal{X}_i} x'\}$.

Polr has a symmetric monoidal structure. The tensor $\mathcal{X}_1 \otimes \mathcal{X}_2$ is the cartesian product of sets with the product order. The endofunctor $\mathcal{X} \otimes _$ admits a right adjoint $_ \multimap \mathcal{Y}$ defined as follows: $|\mathcal{X} \multimap \mathcal{Y}| = |\mathcal{X}| \times |\mathcal{Y}|$ and $\langle x, y \rangle \leq_{\mathcal{X} \multimap \mathcal{Y}} \langle x', y' \rangle$ if $x' \leq_{\mathcal{X}} x$ and $y \leq_{\mathcal{Y}} y'$. The evaluation morphism $\text{ev}_{\mathcal{X}_1, \mathcal{X}_2}: (\mathcal{X}_1 \multimap \mathcal{X}_2) \& \mathcal{X}_1 \rightarrow \mathcal{X}_2$ is $\{\langle \langle \langle x, y \rangle, x' \rangle, y' \rangle \mid x \leq_{\mathcal{X}_1} x', y \leq_{\mathcal{X}_2} y' \rangle\}$.

Rel is the full subcategory of Polr where objects are sets equipped with the discrete order.

Exponential Comonads for Polr The richer setting of Polr allows several interesting possible choices of exponential comonads. We present two particular comonadic constructions, already considered in [Ehr12a]. The First one is a completely straightforward extension of the exponential in Rel , given as follows. We define a comonad $\langle \mathcal{M}, \text{der}, \text{dig} \rangle$ based again on the free commutative monoid construction. The endofunctor $\mathcal{M}: \text{Polr} \rightarrow \text{Polr}$ is given by $\mathcal{M}\mathcal{X} = \langle \mathcal{M}_f(|\mathcal{X}|), \leq_{\mathcal{X}} \rangle$ with $[x_1, \dots, x_n] \leq_{\mathcal{M}\mathcal{X}} [x'_1, \dots, x'_{n'}]$ if $n = n'$ and there is $\sigma \in S_n$ such that $x_i \leq_{\mathcal{X}} x'_{\sigma(i)}$ for all $1 \leq i \leq n$; for $f \in \text{Polr}(\mathcal{X}, \mathcal{Y})$, we set $\mathcal{M}f = \{\langle [x_1, \dots, x_n], [y_1, \dots, y_k] \rangle \mid \langle x_i, y_i \rangle \in f, k \in \mathbb{N}\}$. Dereliction $\text{der}_{\mathcal{X}}: \mathcal{M}\mathcal{X} \rightarrow \mathcal{X}$ is $\{\langle [x], x' \rangle \mid x \leq_{\mathcal{X}} x'\}$, and digging $\text{dig}_{\mathcal{X}}: \mathcal{M}\mathcal{X} \rightarrow \mathcal{M}\mathcal{M}\mathcal{X}$ is $\{\langle m, [m_1, \dots, m_k] \rangle \mid m \leq_{\mathcal{X}} m_1 + \dots + m_k\}$.

We denote as MPolr the coKleisli category of the comonad \mathcal{M} . MRel is the full subcategory of MPolr where objects are sets equipped with the discrete order.

7. Our framework is actually dual to [Ehr12a; Ehr16].

2. Categorical Interlude – 2.7. Relations, Preorders, Distributors

We now consider the second comonad $\langle \mathcal{C}, \text{der}, \text{dig} \rangle$. The endofunctor action on objects $\mathcal{C} : \text{Polr} \rightarrow \text{Polr}$ is given as $\mathcal{C}\mathcal{X} = \langle \mathcal{M}_f(|\mathcal{X}|), \leq_{\mathcal{X}} \rangle$ with $[x_1, \dots, x_n] \leq_{\mathcal{C}\mathcal{X}} [x'_1, \dots, x'_{n'}]$ if for all $j \in [n']$ there exists $i \in [n]$ such that $x_i \leq_{\mathcal{X}} x_j$. For $f \in \text{Polr}(\mathcal{X}, \mathcal{Y})$, we set

$$\mathcal{C}f = \{ \langle m, m' \rangle \mid \text{for all } y \in m' \text{ there exists } x \in m \text{ s.t. } \langle x, y \rangle \in f \}.$$

Dereliction $\text{der}_{\mathcal{X}} : \mathcal{C}\mathcal{X} \rightarrow \mathcal{X}$ is $\{ \langle \bar{x}, x \rangle \mid \bar{x} \leq_{\mathcal{C}\mathcal{X}} [x] \}$, and digging $\text{dig}_{\mathcal{X}} : \mathcal{C}\mathcal{X} \rightarrow \mathcal{C}\mathcal{C}\mathcal{X}$ is $\{ \langle m, [m_1, \dots, m_k] \rangle \mid m \leq_{\mathcal{C}\mathcal{X}} m_1 + \dots + m_k \}$.

The former two comonadic constructions have a nice interpretation in the framework of resource structures (cfr. Section 2.6.1). Both functors \mathcal{M} and \mathcal{C} builds free preorder constructions on multisets. Given a preorder \mathcal{X} , that could be thought as arbitrary atomic data, the construction $\mathcal{M}\mathcal{X}$ gives a canonical way to build a preorder relation on finite collections (multisets) of this data. The relation on \mathcal{M} can be seen as *resource sensitive*: multisets of different sizes are incomparable. Moreover, two multisets \bar{x}, \bar{y} of the same size are comparable iff given $x \in \bar{x}$ there exists *exactly* one element $y \in \bar{y}$ such that $x \leq_{\mathcal{X}} y$ and vice-versa. The construction \mathcal{C} instead allows a more liberal preorder. In particular, it is easy to prove that $\mathcal{C}\mathcal{X}$ admits a greatest element, that is the empty multiset $[\]$. The structure in this case loses resource sensitivity: multisets of different size are in general comparable and we have $[a] \cong [a, a]$, *i.e.* up to isomorphism idempotency. In particular, these are the basic properties satisfied by the preorder relation of $\mathcal{C}\mathcal{X}$:

$$\begin{aligned} [a] \leq_{\mathcal{C}\mathcal{X}} [\] \quad [a, b] \leq_{\mathcal{C}\mathcal{X}} [a] \quad [a, b] \leq_{\mathcal{C}\mathcal{X}} [b] \\ [a, a] \cong [a] \end{aligned}$$

The former conditions make $\mathcal{C}\mathcal{X}$ equivalent to the free bounded meet-semilattice on \mathcal{X} ⁸.

From Rel and Polr to Dist. We recall a basic but pivotal fact: a relation $f \subseteq X \times Y$ can be identified with its characteristic function $\chi_f : X \times Y \rightarrow \mathbf{2}$ where $\mathbf{2} = \{0, 1\}$ is the two-element boolean algebra with sum and product. Composition is then defined as

$$\chi_{g \circ f}(x, z) = \bigvee_{y \in Y} \chi_g(y, z) \wedge \chi_f(x, y) \quad \text{where } \chi_f : X \times Y \rightarrow \mathbf{2} \text{ and } \chi_g : Y \times Z \rightarrow \mathbf{2}. \quad (2.8)$$

All the constructions in Rel and Polr can be reformulated in this characteristic function perspective. For instance, in Rel, the identity at X becomes the characteristic function of X .

In Polr, a monotonic relation f from $\mathcal{X} = \langle |\mathcal{X}|, \leq_{\mathcal{X}} \rangle$ to $\mathcal{Y} = \langle |\mathcal{Y}|, \leq_{\mathcal{Y}} \rangle$ can be

⁸. Indeed, as remarked in [Ehr12a], the construction $\mathcal{C}\mathcal{X}$ could be replaced with an equivalent one, where instead to consider multisets one considers finite sets.

seen as a monotonic characteristic function $\chi_f: \mathcal{X}^o \times \mathcal{Y} \rightarrow \mathbf{2}$, where $\mathcal{X}^o = \langle X, \geq_{\mathcal{X}} \rangle$ and $\mathbf{2}$ is endowed with the boolean order. Any preorder $\mathcal{X} = \langle |\mathcal{X}|, \leq_{\mathcal{X}} \rangle$ forms a category where $\text{ob}(\mathcal{X}) = |\mathcal{X}|$ and $\mathcal{X}(x, x')$ is a singleton (if $x \leq_{\mathcal{X}} x'$) or the empty set (otherwise), so \mathcal{X}^o is the opposite category of \mathcal{X} . Thus, $\chi_f: \mathcal{X}^o \times \mathcal{Y} \rightarrow \mathbf{2}$ is a bifunctor, contravariant in \mathcal{X} and covariant in \mathcal{Y} .

It is then natural to generalize the characteristic function viewpoint to generic categories, which gives rise to the notion of *distributor* (also known as *profunctors*).

2.7.3. The Bicategory of Distributors

For two small categories A, B , a *distributor* $F: A \dashv\vdash B$ is a functor $F: B^o \times A \rightarrow \text{Set}$.

Distributors determine a bicategory, Dist . For a complete presentation of the structure of this bicategory we refer to [Bor94; Bén00; GJ17].

- 0-cells are small categories $A, B, C \dots$. 1-cells $F: A \dashv\vdash B$ are distributors, *i.e.* functors $F: B^o \times A \rightarrow \text{Set}$. 2-cells $\alpha: F \Rightarrow G$ are natural transformations.
- Given any 0-cells A and B , 1-cells and 2-cells are organized as a category $\text{Dist}(A, B)$. Vertical composition $\alpha \star \beta$ is given by the usual composition of natural transformations.
- For $A \in \text{Dist}$, the identity $1_A: A \dashv\vdash A$ is Yoneda's embedding $1_A(a', a) = A(a', a)$;
- For 1-cells $F: A \dashv\vdash B$ and $G: B \dashv\vdash C$, their horizontal composition is given by

$$(G \circ F)(a, c) = \int^{b \in B} G(b, c) \times F(a, b)$$

Note the analogy with (2.8). Composition associativity and identity laws hold only up to canonical isomorphisms, that one can easily derives via coend manipulations. For this reason Dist is a bicategory.

- We define the *zero distributor* $\emptyset_{A,B} \in \text{ob}(\text{Dist}(A, B))$ as $\emptyset_A(a, b) = \emptyset$ for all $a \in \text{ob}(A)$ and $b \in \text{ob}(B)$.

The cartesian closed structure of the 2-category CAT gives, in particular, the following isomorphism

$$\text{CAT}(B^o \times A, \text{Set}) \cong \text{CAT}(A, PB)$$

where we recall that PB is the category of presheaves of B . Hence we have a correspondence

$$\frac{F: B^o \times A \rightarrow \text{Set}}{\lambda(F): A \rightarrow PB}$$

From this basic fact, one would be able to see the bicategory of distributors as a Kleisli bicategory for a pseudomonad of presheaves on Cat . Unfortunately, this is

not possible since for a small category A , PA is not small any more. However, the notion of relative pseudomonads (Definition 2.3.4) comes to help (Lemma 2.5.6).

Lemma 2.7.1 (Fiore-Gambino-Hyland-Winskel). *Distributors are the Kleisli bicategory for the relative pseudomonad of presheaves P over the inclusion functor $j : \text{Cat} \rightarrow \text{CAT}$.*

2.8. The Bicategories $S\text{-Dist}$ and $S\text{-CatSym}$

The content of this section is a corollary of the constructions and results presented in [GJ17; Fio+08; Fio+17].

In [Fio+17] is introduced a method to extend 2-monads over Cat to (relative) pseudomonads over Dist . The construction is based on the intuition that the bicategory of distributors is the Kleisli bicategory for a suitable (relative) pseudomonad of presheaf on the 2-category Cat (Lemma 2.7.1) and the notion of lifting of (pseudo) algebras for a 2-monad (Section 2.3.3).

We work with an arbitrary resource monad S over CAT and we suppose that its tensor product is symmetric. Let $J : \text{Cat} \rightarrow \text{CAT}$ be the inclusion functor and P the relative pseudomonad of presheaves over J (cfr. Lemma 2.5.6). By Theorem 2.6.4, the relative pseudomonad of presheaves P admits a lifting to the pseudoalgebras of S , where we recall that the pseudoalgebras of S are unbiased S -monoidal categories. Hence, by Theorem 2.3.8, the 2-monad S lifts to a relative pseudomonad \tilde{S} over the identity $1_{Kl(P)} : Kl(P) \rightarrow Kl(P)$. We recall that $Kl(P) = \text{Dist}$. We want to explicitly define the Kleisli bicategory of \tilde{S} , which we denote as $S\text{-Dist}$. In order to do so, we need an explicit definition of \tilde{S} . This is the relative pseudomonad associated to the relative pseudoadjunction

$$\begin{array}{ccc} & & Kl(\bar{P}) \\ & \nearrow \Phi & \downarrow \Upsilon \\ Kl(P) & \longleftarrow & Kl(P) \end{array}$$

Where Υ is the forgetful pseudofunctor $\Upsilon : Kl(\tilde{P}) \rightarrow Kl(P)$, and \bar{P} is the relative pseudomonad associated to the lifting of P to pseudoalgebras of S . The former (relative) pseudoadjunction consists of the following:

- for $A \in \text{Cat}$ we have that the free algebra $\Phi A = (SA, h_A : SSA \rightarrow SA)$ is clearly an object of $Kl(\bar{P})$, since, in particular, ΦA is a strict algebra, hence a pseudoalgebra (this derives by the fact that S is a 2-monad).
- A family of morphisms $\tilde{e}_A : A \rightarrow \Upsilon \Phi A = SA$, defined as the composite

$$JA \xrightarrow{\eta_A} SJA = JSA \xrightarrow{y_{SA}} PSA$$

2. Categorical Interlude – 2.8. The Bicategories $\mathcal{S}\text{-Dist}$ and $\mathcal{S}\text{-CatSym}$

where we recall that η_A is the unit of the 2-monad S and y_A is the Yoneda embedding, such that, for $A \in \text{Cat}$ and $\mathbb{B} = \langle B, h_b \rangle \in S\text{-PAlg}_{\text{CAT}}$, we have the adjoint equivalence of categories

$$\begin{array}{ccc} & \xrightarrow{(-)^b} & \\ \text{CAT}(JA, PB) & \perp & S\text{-PAlg}_{\text{CAT}}(\Phi JA, P\mathbb{B}) \\ & \xleftarrow{\Upsilon(-) \circ \tilde{e}_A} & \end{array}$$

where for $F \in \text{Dist}(A, B) = \text{CAT}(JA, PB)$ we have $F^b \in S\text{-PAlg}_{\text{CAT}}(\Phi JA, P\mathbb{B})$ is induced by the essentially unique functor defined by the universal property of the free construction

$$\begin{array}{ccc} SSA & \longrightarrow & SPB \\ h_{SA} \downarrow & \swarrow \sigma & \downarrow h_{PB}^P \\ SA & \xrightarrow{F^b} & PB \end{array}$$

PB being equipped with the Day convolution, it admits a (unbiased) S -monoidal structure. Being non-strict, the former diagram commutes only up to invertible (coherent) 2-cell σ . Hence F^b is a pseudomorphism. Concretely

$$F^b(\langle a_1, \dots, a_n \rangle) = \otimes_{i \in [n]} F(a_i)$$

then, here, it will be the n -ary Day convolution

$$\langle a_1, \dots, a_n \rangle \rightarrow F_{a_1} \hat{\otimes} \dots \hat{\otimes} F_{a_n}.$$

By Remark 2.3.5, we get a relative pseudomonad \tilde{S} on the identity $1_{Kl(P)} : Kl(P) \rightarrow Kl(P)$ with the following basic structure:

- For $A \in \text{Cat}$, $\tilde{S}A = SA$.
- For $A \in \text{Cat}$, $i_A : A \rightarrow \tilde{S}A$ is given by the 1-cell \tilde{e}_A .
- For $F \in \text{CAT}(JA, PSB)$, $F^* \in \text{CAT}(SJA, PSB)$ is defined as the underling morphism of the pseudoalgebra F^b , that is given by the n -ary Day convolution.

2.8.1. The Bicategory $\mathcal{S}\text{-Dist}$

We give an explicit presentation of the Kleisli bicategory for the relative pseudomonad \tilde{S} , applying Theorem 2.3.6. We call this bicategory $\mathcal{S}\text{-Dist}$. The explicit structure of $\mathcal{S}\text{-Dist}$ is as follows:

1. $\text{ob}(\mathcal{S}\text{-Dist}) = \text{Obj}(\text{Dist}) = \text{Obj}(\text{Cat})$.
2. For $A, B \in \mathcal{S}\text{-Dist}$, we have $\mathcal{S}\text{-Dist}(A, B) = \text{Dist}(A, SB)$.

2. *Categorical Interlude – 2.8. The Bicategories $S\text{-Dist}$ and $S\text{-CatSym}$*

3. The identity S -distributor is defined as

$$Y_{SA} \circ \eta_A$$

Explicitly, $1_A(\vec{a}, a) = SA(\vec{a}, \langle a \rangle)$.

4. For $F : A \rightarrow PSB$, we have $F^\flat : SA \rightarrow PSB$, that is given by the n -ary Day convolution, as seen in the former section.
5. For $F : A \rightrightarrows SB$ and $G : B \rightrightarrows SC$, composition is given as follows

$$(G \circ F)(\vec{c}, a) = \int^{\vec{b} \in SB} G^\flat(\vec{c}, \vec{b}) \times F(\vec{b}, a).$$

2.8.2. The Bicategory $S\text{-CatSym}$

We define the bicategory of *S -categorical symmetric sequences* by $S\text{-CatSym} = S\text{-Dist}^{op}$. It is useful to give an explicit definition of the relevant structure of $S\text{-CatSym}$.

1. $\text{ob}(S\text{-CatSym}) = \text{ob}(\text{Cat})$.
2. For $A, B \in S\text{-Dist}$, we have $S\text{-CatSym}(A, B) = S\text{-Dist}(B, A) = \text{Dist}(B, SA)$.
3. The identity is again defined as

$$1_A(\vec{a}, a) = SA(\vec{a}, \langle a \rangle).$$

4. For $F : A \rightsquigarrow B$ and $G : B \rightsquigarrow C$ S -categorical symmetric sequences, composition is given by considering F and G as S -distributors:

$$(G \circ F)(\vec{a}, c) = \int^{\vec{b} \in SB} G(\vec{b}, c) \times F^\flat(\vec{a}, \vec{b}).$$

5. $S\text{-CatSym}$ is cartesian (Proposition 2.8.1). The cartesian product is the disjoint union $A \& B = A \sqcup B$ and the projections are defined as follows:

$$\pi_{i,2}(\vec{c}, a) = S(A \sqcup B)(\vec{c}, \langle \iota_i(a) \rangle).$$

The terminal object is the empty category.

6. We shall see that the bicategory $S\text{-CatSym}$ is cartesian closed (Theorem 2.8.2.1). Indeed, if the tensor product on SA is symmetric, we have the following Seely equivalence (Proposition 2.6.5):

$$S(A \sqcup B) \simeq S(A) \times S(B).$$

From that, one can build the following chain of equivalences, that gives the right pseudoadjoint to the cartesian product:

2. Categorical Interlude – 2.8. The Bicategories $S\text{-Dist}$ and $S\text{-CatSym}$

$$\begin{aligned} S\text{-CatSym}(A \& B, C) &= \text{Dist}(C, S(A \sqcup B)) = \\ \text{CAT}(S(A \sqcup B)^o \times C, \text{Set}) &\simeq \text{CAT}(SA^o \times (SB^o \times C), \text{Set}) = \\ S\text{-Dist}(SB^o \times C, A) &= S\text{-CatSym}(A, SB^o \times C) \end{aligned}$$

This chain of equivalences suggests to consider $SB^o \times C$ as the exponential object.

2.8.2.1. The Cartesian Closed Structure of $S\text{-CatSym}$

In this section we extend the results of [Fio+08] and [GJ17] to our parameterized setting.

Proposition 2.8.1. *The bicategory $S\text{-CatSym}$ is cartesian.*

Proof. We prove it by building a right biadjoint to the diagonal pseudofunctor. For small categories $(A_i)_{i \in [n]}$ we set

$$\&_{i \in [n]} A_i = \sum_{i \in [n]} A_i$$

We define the projections

$$\begin{aligned} \pi_{i,n} : \&_{i \in [n]} A_i &\rightsquigarrow A_i \\ \pi_{i,n}(\vec{c}, a) &= S(\&A_i)(\vec{c}, \langle \iota_i(a) \rangle) \end{aligned}$$

we define the pairing

$$\langle F_1, \dots, F_n \rangle(\vec{v}, \iota_i(a)) = F_i(\vec{v}, a)$$

We now prove that we have an equivalence

$$\begin{array}{ccc} & \xrightarrow{(\pi_{1,n} \circ (-), \dots, \pi_{n,n} \circ (-))} & \\ S\text{-CatSym}(B, \&_{i \in [n]} A_i) & & \prod_{i \in [n]} S\text{-CatSym}(A_i, B) \\ & \xleftarrow{\langle - \rangle} & \end{array}$$

In order to do so, we compute the unit and counit isomorphisms. For $F : B \rightsquigarrow \&_{i \in [n]} A_i$, the components of the unit are given by a natural isomorphism

$$F \cong \langle \pi_{1,n} \circ F, \dots, \pi_{n,n} \circ F \rangle$$

We prove it by coend manipulations.

$$\langle \pi_{1,n} \circ F, \dots, \pi_{n,n} \circ F \rangle(\vec{b}, \iota_i(a)) = \pi_{i,n} \circ F(\vec{b}, \iota_i(a))$$

2. Categorical Interlude – 2.8. The Bicategories $S\text{-Dist}$ and $S\text{-CatSym}$

$$\pi_{i,n} \circ F(\vec{b}, a) = \int^{\vec{c} \in \&A_i} \pi_{i,n}(\vec{c}, \iota_i(a)) \times F^b(\vec{b}, \vec{c})$$

We develop

$$\int^{\vec{c} \in \&A_i} S(\&A_i)(\vec{c}, \langle \iota_i(a) \rangle) \times F^b(\vec{b}, \vec{c})$$

Then we apply Yoneda and we conclude. The counit case is again by coend manipulations. \square

Theorem 2.8.2. *The bicategory $S\text{-CatSym}$ is cartesian closed.*

Proof. First, we consider the equivalence of categories (Proposition 2.6.5)

$$S(A \sqcup B) \simeq SA \times SB$$

We denote the components as $\mu_0 : S(A \sqcup B) \rightarrow SA \times SB$ and $\mu_1 : SA \times SB \rightarrow S(A \sqcup B)$. We have the following corresponding distributors:

$$\begin{aligned} \bar{\mu}_0 : S(A \sqcup B) &\rightarrow SA \times SB \\ (\langle \vec{a}, \vec{b} \rangle, \vec{c}) &\mapsto (SA \times SB)(\langle \vec{a}, \vec{b} \rangle, \mu_0(\vec{c})) \end{aligned}$$

$$\begin{aligned} \bar{\mu}_1 : SA \times SB &\rightarrow S(A \sqcup B) \\ (\vec{c}, \langle \vec{a}, \vec{b} \rangle) &\mapsto (S(A \sqcup B))(\vec{c}, \mu_1(\vec{a}, \vec{b})) \end{aligned}$$

We set

$$B^A = SA^\circ \times B$$

Then we define the S -categorical symmetric sequence $ev_{A,B} : B^A \& A \rightsquigarrow B$ as

$$ev_{A,B} = \bar{\mu}_1 \circ 1_{B^A}$$

Explicitly,

$$\begin{aligned} ev_{A,B}(\vec{c}, b) &= \int^{\vec{d} \in S(SA^\circ \times B)} \int^{\vec{a} \in SA} S((SA^\circ \times B) \sqcup A)(\vec{c}, \vec{d} \oplus \vec{a}) \\ &\quad \times S(SA^\circ \times B)(\vec{d}, \langle \vec{a}, b \rangle) \\ &\quad \cong \\ &\quad \int^{\vec{a} \in SA} S((SA^\circ \times B) \sqcup A)(\vec{c}, \langle \vec{a}, b \rangle \oplus \vec{a}) \end{aligned}$$

Then we proceed to show that the cartesian product pseudofunctor admits a right biadjoint. For $G : A \& B \rightsquigarrow C$ we define its currying as

$$\lambda_B(G)(\vec{a}, \langle \vec{b}, c \rangle) = G(\vec{a} \oplus \vec{b}, c)$$

2. Categorical Interlude – 2.8. The Bicategories $S\text{-Dist}$ and $S\text{-CatSym}$

We now prove that there is an equivalence

$$\begin{array}{ccc}
 & \xleftarrow{ev_B \circ (- \& B)} & \\
 S\text{-CatSym}(A \& B, C) & & S\text{-CatSym}(A, C^B) \\
 & \xrightarrow{\lambda_B(-)} &
 \end{array}$$

In order to do so, we compute the unit and counit isomorphisms. For $F : A \rightsquigarrow C^B$ the components of the unit are given by a natural isomorphism

$$F \cong \lambda_B(ev_{B,C} \circ (F \& B))$$

We show it by coend manipulations.

$$\begin{aligned}
 \lambda_B(ev_{B,C} \circ (F \& B))(\vec{a}, \langle \vec{b}, c \rangle) &= ev_{B,C} \circ (F \& B)(\vec{a} \oplus \vec{b}, c) \\
 &= \int^{\vec{d} \in S((SB^o \times C) \sqcup B)} ev_{B,C}(\vec{d}, c) \times (F \& B)^b(\vec{a} \oplus \vec{b}, \vec{d})
 \end{aligned}$$

By definition of ev and Yoneda

$$\cong \int^{\vec{b}' \in SB} (F \& B)^b(\vec{a} \oplus \vec{b}, \langle \vec{b}, c \rangle \oplus \vec{b})$$

By definition of the product pseudofunctor

$$\cong \int^{\vec{b}' \in SB} F(\vec{a}, \langle \vec{b}', c \rangle) \times \int^{\vec{b}_i \in SB} \prod_{i \in [l(\vec{b})]} SB(\vec{b}_i, \langle b'_i \rangle) \times SB(\vec{b}, \bigoplus \vec{b}_i)$$

Applying several times Yoneda we can then conclude

$$\cong F(\vec{a}, \langle \vec{b}, c \rangle)$$

The counit case is again by lengthy coend manipulations. \square

Remark 2.8.3. *In the case where S is the linear resource monad, $S\text{-CatSym}$ is the bicategory of Categorical Symmetric Sequences [GJ17; Fio+17]. This bicategory is biequivalent, via an appropriate dualizing pseudofunctor, to the bicategory of Generalized Species of Structures [Fio+08]. Generalized species of structures are a very rich framework which categorifies both Joyal’s Combinatorial Species [Joy86] and the cartesian closed category MPolr (Section 2.7.2).*

3. Intersection Type Distributors

3.1. Introduction

In this Chapter we present a *categorification* of the classic correspondence between intersection type assignments and categorical semantics of λ -calculus induced by (generalizations of) the category of sets and relations. It is well known [Car07; Ehr12a; Ter12] that in several coKleisli categories of Polr (which contains Rel, Section 2.7.2) the interpretation of a λ -term M is given by a monotonic relation that behaves as follows:

$$\llbracket M \rrbracket = \{ \langle \Delta, a \rangle \mid \Delta \vdash M : a \}$$

Where Δ is a type context, a is a type and the judgment $\Delta \vdash M : a$ refers to an appropriate intersection type system¹. The former relation corresponds, taking the characteristic functions point of view, to

$$\llbracket M \rrbracket(\Delta, a) = \begin{cases} 1 & \text{if } \Delta \vdash M : a \\ 0 & \text{otherwise.} \end{cases}$$

We shall present a denotational semantics in the parametric bicategory $S\text{-CatSym}$ which improves and refines the former intersection type semantics in several ways.

1. In our setting, the denotation of a λ -term M is given, up to isomorphisms, by a distributor (Theorem 3.4.10) that behaves as follows:

$$\llbracket M \rrbracket(\Delta, a) = \left\{ \begin{array}{c} \tilde{\pi} \\ \vdots \\ \Delta \vdash M : a \end{array} \right\}$$

where π is a type derivation in an appropriate intersection type system (Figure 3.3) whose intersection type \cap is given by the *list construction* and the equivalence relation is generated by the specific horizontal composition of $S\text{-CatSym}$. More precisely, the intersection type constructor corresponds to a *tensor product*, whose particular structure depends on the resource monad S . We call $\llbracket M \rrbracket$ the *S-intersection type distributor of M*. The former bicategorical semantics is then *proof-relevant*: the intersection type distributor of M not

1. Where the intersection type $a \cap b$, which corresponds to the product of the appropriate free commutative monoid construction, that can be *non-idempotent* [Car07] or *idempotent* [Ehr12a; Ter12].

3. Intersection Type Distributors – 3.1. Introduction

only testes, given as input a type context Δ and a type a , the typability relation $\Gamma \vdash M : a$, but it returns the set of all (equivalence class of) type derivations that witness that relation. Moreover, if $M \rightarrow_\beta N$, the equivalence relation on type derivations induces a natural isomorphism

$$\alpha : \llbracket M \rrbracket \cong \llbracket N \rrbracket.$$

For this reason, intersection type distributors are a bicategorical *denotational* semantics.

2. Types live in a *category* (Figure 3.3) that we denote as D , where morphisms gives a notion of generalized subtyping relation. This is a completely natural generalization of what happens in the category Polr , where types lives in a preorder. However, this more general kind of subtyping is sensitive to the particular *witness* of the relation $a \leq b$, that is a morphism $f \in D(a, b)$. This allows the definition of *left* and *right* actions on type derivations (Figures 3.6 and 3.5). In the standard intersection type systems with subtyping, if $\Delta \vdash M : a$ and $\Delta' \leq \Delta, a \leq a'$ then $\Delta' \vdash M : a'$. In our framework the action of morphisms $\eta : \Delta' \rightarrow \Delta, f : a \rightarrow a'$ produces a type derivation of conclusion $\Delta' \vdash M : a'$.
3. Usually, the behaviour of the intersection type is characterized by the fact that the intersection operation is either associative, commutative or idempotent. Our bicategorical setting refines this point-of-view. As in [MZ18], the emphasis is shifted from the *strictness* of commutativity and idempotency to the exhibition of morphisms (Section 2.6). More explicitly, the standard commutativity condition $a \cap b = b \cap a$ is replaced by the existence of *symmetries* $\sigma_{a,b} : a \cap b \cong b \cap a$ in the category SD . The standard idempotency condition is replaced by the existence of *diagonals* $c_a : a \rightarrow a \cap a$ in the category SD . Throughout the chapter we shall make an extensive use of the notations and operations on lists introduced in Section 2.6.2.

Structure of the Chapter First we present the bicategorical semantics of simply typed λ -calculus induced by the bicategory $S\text{-CatSym}$. We prove that this semantics is equivalent, up to isomorphism, to another interpretation for simply typed λ -terms, called the *denotation*, that we shall use for defining intersection type distributors. We extend the two interpretations to the untyped case and we prove that the typed one embed in this latter. We then define the intersection type system E_A^S and S -intersection type distributors, proving that they are naturally isomorphic to denotations of λ -terms. We study intersection type distributors under reduction, showing that they induce a proof-relevant bicategorical denotational semantics. We then prove several normalization theorems, where the standard normalization properties are characterized trough intersection type distributors. The proofs follow from an adaptation of standard reducibility techniques to our categorified setting.

We conclude presenting two concrete examples of our parametric construction: the cases where S is respectively linear and cartesian.

Discussion of Related Work In this chapter we define four intersection type systems, corresponding to the four resource monads that make S -CatSym a cartesian closed bicategory *i.e.* the linear, semicartesian, relevant and cartesian resource monads. A similar phenomenon appears in [MPV18], where a general categorical interpretation of intersection type disciplines is presented. The setting of [MPV18] is 2-dimensional and the λ -calculus is there presented as a 2-operad, a construction akin to other 2-dimensional generalizations of λ -calculus [See87; Hir13]. In that framework, the authors define linear, affine, relevant and cartesian intersection type systems, giving an elegant presentation of these systems in terms of a special kind of *fibrations*. However, their intersection type systems are *discrete*, in the sense that they do not exhibit subtyping. Duplication and erasing is simulated *via* inference rules. We already discussed how in our setting instead the structural behaviour of the intersection type is completely determined by type morphisms. This fact makes our systems syntax directed, contrary to the systems of [MPV18]. One can also straightforwardly embed the type systems of [MPV18] in our type systems. We believe that an extension of Mazza’s point of view to our setting would be of great interest². However, this would be far from trivial. Mazza’s construction relies on the correspondence between a special kind of resource calculi, the *polyadic terms*, and an appropriate class of intersection type systems. This correspondence fails in our case, as shown in the preliminary discussion of Chapter 4 of this thesis. For this reason one needs to introduce *subtyping-aware polyadic terms*, an extension of standard polyadic calculi that is sensitive to the subtyping feature of intersection type distributors.

3.2. Models for the Simply Typed λ -calculus

In this section we present two denotational bicategorical models for simply typed λ -calculus and we prove that they are isomorphic up to Seely equivalence. The first model is just the standard bicategorical semantics induced by the bicategory S -CatSym, parametric over resource monads, where S admits a symmetric tensor product. The second one will be fundamental in the definition of the intersection types semantics and it is inspired by Linear Logic. When we write SA^n (resp. SA°)

2. As we already discussed in the main introduction, their construction deals only with relational discrete distributors. Their category of types is just a set. However, they clearly stated the possibility of extending their approach to the non-discrete setting, taking subtyping into consideration [Maz17][p.66]. We present such an extension. We deal with standard distributors, where natural transformations are given by functions, not relations. Moreover, if $M \rightarrow_\beta N$ not only we get natural transformation $\beta : \llbracket M \rrbracket \rightarrow \llbracket N \rrbracket$, but β is also *invertible*. This happens thanks to a non-trivial quotient on type derivations, that is naturally induced by the structure of our bicategories.

3. Intersection Type Distributors – 3.2. Models for the Simply Typed λ -calculus

$$\begin{array}{c}
 A, B := a \mid A \Rightarrow B \\
 \\
 \frac{}{\Gamma, x : A \vdash x : A} \quad \frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x.M : A \Rightarrow B} \\
 \\
 \frac{\Gamma \vdash M : A \Rightarrow B \quad \Gamma \vdash N : A}{\Gamma \vdash MN : B}
 \end{array}$$

Figure 3.1. – Simply typed λ -calculus

we always mean $(SA)^n$ (resp. $(SA)^o$). Let At be a set, called the *set of atoms*. We define the *simple types* by the following grammar:

$$A, B ::= o \in At \mid A \Rightarrow B$$

Figure 3.1 recalls the syntax of simply typed λ -calculus. We fix an evaluation function $\rho : At \rightarrow \text{ob}(\text{Cat})$. We recall that in the bicategory $\mathcal{S}\text{-CatSym}$ the product $A \& B$ is given by the coproduct of small categories $A \sqcup B$ and the exponential object A^B by $SA^o \times B$ (Section 2.8.2). The model follows the standard one-dimensional definition:

1. On types:

$$\begin{aligned}
 \llbracket o \rrbracket_\rho &= \rho(o) & \llbracket A \Rightarrow B \rrbracket_\rho &= S[\llbracket A \rrbracket_\rho^o \times \llbracket B \rrbracket_\rho] \\
 \llbracket \Gamma = A_1, \dots, A_n \rrbracket_\rho &= \llbracket A_1 \rrbracket_\rho \sqcup \dots \sqcup \llbracket A_n \rrbracket_\rho
 \end{aligned}$$

We will often write $\llbracket A \rrbracket$ instead of $\llbracket A \rrbracket_\rho$, keeping the parameter ρ implicit.

2. We associate to each simply typed λ -term $\Gamma \vdash M : A$ a \mathcal{S} -categorical symmetric sequence $\llbracket \Gamma \vdash M : A \rrbracket : \llbracket \Gamma \rrbracket_\rho \rightsquigarrow \llbracket A \rrbracket_\rho$ by induction as follows:

$$\begin{aligned}
 \llbracket x_1 : A_1, \dots, x_n : A_n \vdash x_i : A_i \rrbracket(\vec{c}, a) &= \pi_{n,i}(\vec{c}, a) = S\left(\bigsqcup_{j=1}^n \llbracket A_j \rrbracket_\rho\right)(\vec{c}, \iota_i(a)) \\
 \llbracket \Gamma \vdash \lambda x.M : A \Rightarrow B \rrbracket(\vec{c}, \langle \vec{a}, b \rangle) &= \lambda(\llbracket \Gamma, x : A \vdash M : B \rrbracket)(\vec{c}, \langle \vec{a}, b \rangle) = \\
 &= \llbracket \Gamma, x : A \vdash M : B \rrbracket_\rho(\vec{c} \oplus \vec{a}, b) \\
 \llbracket \Gamma \vdash PQ : B \rrbracket(\vec{c}, b) &= \text{ev}_{A,B} \circ \langle \llbracket \Gamma \vdash P : A \Rightarrow B \rrbracket, \llbracket \Gamma \vdash Q : A \rrbracket \rangle(\vec{c}, b) = \\
 &= \int^{\vec{d} \in S(\llbracket A \Rightarrow B \rrbracket \sqcup \llbracket A \rrbracket)} \int^{\vec{a} \in S\llbracket A \rrbracket_\rho} S(\llbracket A \Rightarrow B \rrbracket_\rho \sqcup \llbracket A \rrbracket_\rho)(\vec{d}, \langle \langle \vec{a}, b \rangle \rangle \oplus \vec{a}) \\
 &\quad \times (\langle \llbracket \Gamma \vdash P : A \Rightarrow B \rrbracket, \llbracket \Gamma \vdash Q : A \rrbracket \rangle)^\flat(\vec{c}, \vec{d})
 \end{aligned}$$

3. Intersection Type Distributors – 3.2. Models for the Simply Typed λ -calculus

$$\begin{aligned}
[[x]]_{\Gamma}^{A_i}(\Delta, a_i) &= [[\Gamma]]_{dn}(\Delta, \langle \langle \rangle, \dots, \langle a_i \rangle, \dots, \langle \rangle \rangle) \\
[[\lambda x.M]]_{\Gamma}^{A \Rightarrow B}(\Delta, \langle \vec{a}, b \rangle) &= [[M]]_{\Gamma, A}^B(\Delta \oplus \langle \vec{a} \rangle, b) \\
[[MN]]_{\Gamma}^B(\Delta, B) &= \\
\int^{\vec{a} = \langle a_1, \dots, a_k \rangle \in S[[A]]_{dn}} \int^{\Gamma_0, \dots, \Gamma_k \in [[\Gamma]]_{dn}} &[[M]]_{\Gamma}^{A \Rightarrow B}(\Gamma_0, \langle \vec{a}, b \rangle) \times \prod_{i=1}^k [[N]]_{\Gamma}^A(\Gamma_i, a_i) \times [[\Gamma]]_{dn}(\Delta, \bigotimes_{i=0}^k \Gamma_i)
\end{aligned}$$

Figure 3.2. – Denotation of simply typed λ -terms.

We now define a family of S -distributors that we will prove to be isomorphic, up to Seely equivalence, to the bicategorical semantics of λ -terms.

We define a *denotation* for types as follows:

$$[[A]]_{dn}^{\rho} = [[A]]_{\rho} \quad [[\Gamma = A_1, \dots, A_n]]_{dn}^{\rho} = S[[A_1]]_{dn}^{\rho} \times \dots \times S[[A_n]]_{dn}^{\rho}$$

By the Seely equivalence (Remark 2.6.6) we have $[[A_1 \dots A_n]]_{dn}^{\rho} \simeq S([[A_1, \dots, A_n]]_{\rho})$.

We will often write $[[A]]_{dn}$ instead of $[[A]]_{dn}^{\rho}$, again keeping the parameter ρ implicit. Given $\Gamma \vdash M : A$, we define the *typed denotation* of M , $[[M]]_{\Gamma}^A : [[\Gamma]]_{dn}^o \times [[A]]_{dn} \rightarrow \text{Set}$ by induction in Figure 3.2. We recall (see Section 2.6.1) that the tensor product $\Gamma_1 = \langle \vec{a}_1, \dots, \vec{a}_{\text{len}(\Gamma)} \rangle \otimes \Gamma_2 = \langle \vec{a}'_1, \dots, \vec{a}'_{\text{len}(\Gamma)} \rangle$, for $\Gamma_1, \Gamma_2 \in [[\Gamma]]_{dn}$, is defined as

$$\Gamma_1 \otimes \Gamma_2 = \langle \vec{a}_1 \oplus \vec{a}'_1, \dots, \vec{a}_n \oplus \vec{a}'_n \rangle.$$

We also recall that we denote as μ_1 the component of the Seely equivalence that goes from $SA \times SB$ to $S(A \sqcup B)$. We denote as $\bar{\mu}$ the corresponding distributor of μ_1 , defined as $\bar{\mu}(\vec{c}, \langle \vec{a}, \vec{b} \rangle) = S(A \sqcup B)(\vec{c}, (\mu_1(\vec{a}, \vec{b}) = \vec{a} \oplus \vec{b}))$.

Theorem 3.2.1. *Let $M \in \Lambda$ and $\Gamma \vdash M : A$. We have a natural isomorphism*

$$[[M]]_{\Gamma}^A \cong [[\Gamma \vdash M : A]] \circ_{\text{Dist}} \bar{\mu}_1.$$

Proof. By induction on the structure of M , *via* lengthy but straightforward coend manipulations.

If $M = x$ then we have $[[\Gamma \vdash x_i : A_i]](\vec{c}, a_i) = S([[\Gamma]]) (\vec{c}, \langle \iota_i(a_i) \rangle)$ with $\Gamma = A_1, \dots, A_i, \dots, A_n$ and $a_i \in [[A_i]]$. Then

$$\begin{aligned}
[[\Gamma \vdash M : A_i]] \circ_{\text{Dist}} \mu_1(\Delta, a_i) &= \\
\int^{\vec{c} \in [[\Gamma]]} &S([[\Gamma]]) (\vec{c}, \langle \iota_i(a_i) \rangle) \times [[\Gamma]]_{dn}(\Delta, \mu_1(\vec{c}))
\end{aligned}$$

We apply Yoneda and we conclude

3. Intersection Type Distributors – 3.2. Models for the Simply Typed λ -calculus

$$\cong \llbracket \Gamma \rrbracket_{dn}(\Delta, \mu_1(\langle \iota_i(a) \rangle)) = \llbracket \Gamma \rrbracket_{dn}(\Delta, (\langle \rangle, \dots, \langle a_i \rangle, \dots, \langle \rangle)).$$

If $M = \lambda x.M'$, we have

$$\llbracket \Gamma \vdash \lambda x.M' : A \Rightarrow B \rrbracket(\vec{c}, \langle \vec{a}, b \rangle) = \lambda(\llbracket \Gamma, x : A \vdash M' : B \rrbracket)(\vec{c} \oplus \vec{a}, b).$$

Now we precompose with $\bar{\mu}_1$:

$$\int^{\vec{a} \in S[A]} \llbracket \Gamma \rrbracket(\Delta, \mu_1(\vec{c})) \times \llbracket \Gamma, x : A \vdash M' : B \rrbracket(\vec{c} \oplus \vec{a}, b)$$

By IH

$$\cong \llbracket \Gamma, A \rrbracket BM(\Delta \oplus \vec{a}, b) = \llbracket M \rrbracket_{\Gamma}^{A \Rightarrow B}(\Delta, \langle \vec{a}, b \rangle).$$

If $M = PQ$ we have $\llbracket \Gamma \vdash M : B \rrbracket = ev_{A \Rightarrow B, A} \circ \langle \llbracket \Gamma \vdash P : A \Rightarrow B \rrbracket, \llbracket \Gamma \vdash Q : A \rrbracket \rangle$.
By definition $\llbracket \Gamma \vdash M : B \rrbracket(\vec{c}, b) =$

$$\int^{\vec{d} \in S([A \Rightarrow B] \sqcup [A])} \int^{\vec{a} \in S[A]} S([A \Rightarrow B] \sqcup [A])(\vec{d}, \langle \langle \vec{a}, b \rangle \rangle \oplus \vec{a}) \\ \times (\langle \llbracket \Gamma \vdash P : A \Rightarrow B \rrbracket, \llbracket \Gamma \vdash Q : A \rrbracket \rangle)^b(\vec{c}, \vec{d})$$

By Yoneda

$$\cong \int^{\vec{a} \in S[A]} (\langle \llbracket \Gamma \vdash P : A \Rightarrow B \rrbracket, \llbracket \Gamma \vdash Q : A \rrbracket \rangle)^b(\vec{c}, \langle \langle \vec{a}, b \rangle \rangle \oplus \vec{a})$$

If we precompose with $\bar{\mu}_1$ and we apply Yoneda we get

$$\int^{\vec{a} \in S[A]} \int^{\vec{a}_0, \dots, \vec{a}_l(\vec{a}) \in S[A]} \llbracket \Gamma \vdash P : A \Rightarrow B \rrbracket(\vec{a}_0, \langle \vec{a}, b \rangle) \times \\ \prod_{i \in [l(\vec{a})]} \llbracket \Gamma \vdash Q : A \rrbracket(\vec{a}_i, a_i) \times \llbracket \Gamma \rrbracket_{dn}(\Delta, \bigotimes \mu_1(\vec{a}_j))$$

Now we develop the other side.

$$\llbracket PQ \rrbracket_{\Gamma}^B(\Delta, b) = \int^{\vec{a} \in S[A]} \int^{\Gamma_j \in [\Gamma]_{dn}} \llbracket P \rrbracket_{\Gamma}^{A \Rightarrow B}(\Gamma_0, \langle \vec{a}, b \rangle) \times \\ \prod \llbracket Q \rrbracket_{\Gamma}^{A_i}(\Gamma_i, a_i) \times \llbracket \Gamma \rrbracket_{dn}(\Delta, \bigotimes \Gamma_j)$$

By the IH we can conclude, applying Yoneda several times. \square

Remark 3.2.2. We observe that the construction $\llbracket M \rrbracket_{\Gamma}^A$ is not formally an S -symmetric sequence, but a distributor $\llbracket M \rrbracket_{\Gamma}^A : [A]_{dn} \rightarrow [\Gamma]_{dn}$. However, by the former theorem, one can canonically build a S -categorical sequence from $\llbracket M \rrbracket_{\Gamma}^A$, just precomposing the former distributor with the other component of the Seely equivalence, i.e. $\bar{\mu}_2$. We chose to make this passage from S -categorical sequences to

distributors because the corresponding intersection type system (Figure 3.3) has a more type-theoretic familiar structure than the one which can be build directly from the S -categorical sequence $\llbracket \Gamma \vdash M : A \rrbracket$ ³. However both choices are legitimate and equivalent.

3.3. Models for pure λ -calculus

We build a family of non-extensional bicategorical models for pure λ -calculus.

Definition 3.3.1. *Let A be a small category. We define by induction a family of small categories as follows:*

$$D_0 = A \quad D_{n+1} = (SD_n^o \times D_n) \sqcup A$$

We define by induction on $n \in \mathbb{N}$ a sequence of inclusions $\iota_n : D_n \hookrightarrow D_{n+1}$:

$$\iota_0 = \iota_A \quad \iota_{n+1} = (S(\iota_n)^o \times \iota_n) \sqcup 1_A$$

Then we set $D_A = \varinjlim_{n \in \mathbb{N}} D_n$ ⁴.

We denote as $\xi_n : SD_n^o \times D_n \rightarrow D_{n+1}$ the canonical inclusion, for all $n \in \mathbb{N}$. The category D_A is the filtered colimit for the diagram $(D_n \hookrightarrow D_{n+1})_{n \in \mathbb{N}}$. If we set $\langle a_1, \dots, a_k \rangle \Rightarrow a ::= \iota(\langle a_1, \dots, a_k \rangle, a)$, we can give a completely type-theoretic presentation of the category D_A as in Figure 3.3.

Proposition 3.3.2. *There exists a canonical inclusion functor $\iota : SD_A^o \times D_A \hookrightarrow D_A$.*

Proof. The result derives directly from the filtered colimit construction. We remark that

$$SD_A^o \times D_A \cong \varinjlim_{n \in \mathbb{N}} SD_n^o \times \varinjlim_{n \in \mathbb{N}} D_n.$$

Hence we can define $\iota : SD_A^o \times D_A \hookrightarrow D_A$ as

$$\iota(\langle a_1, \dots, a_k \rangle, a) = y_{j+1}(\xi_j(\langle a_1, \dots, a_k \rangle, a))$$

where $j = \min\{n \in \mathbb{N} \mid (\langle a_1, \dots, a_k \rangle, a) \in !D_n^o \times D_n\}$ and $y_{j+1} : D_{j+1} \rightarrow D$ is the canonical injection of D_{j+1} . \square

3. Choosing the other presentation we would end up with intersection type systems where in the context side of a derivation a variable could appear more than once, as for the systems presented in [MPV18].

4. This definition is a generalisation of the standard construction for reflexive objects in relational models. Both constructions are actually a special case of the standard free-algebra construction for an (unpointed) endofunctor [Kel80]. In our case the endofunctor is $S(-)^o \times (-) : \text{Cat} \rightarrow \text{Cat}$.

3. Intersection Type Distributors – 3.3. Models for pure λ -calculus

Types:

$$a := o \in A \mid \langle a_1, \dots, a_k \rangle \Rightarrow a$$

Morphisms:

$$\frac{\frac{f \in A(o, o')}{f : o \rightarrow o'} \quad \frac{\langle \alpha, \vec{f} \rangle : \vec{a}' \rightarrow \vec{a} \quad f : a \rightarrow a'}{\langle \alpha, \vec{f} \rangle \Rightarrow f : (\vec{a} \Rightarrow a) \rightarrow (\vec{a}' \Rightarrow a')}}{\alpha : [k] \rightarrow [k'] \quad f_1 : a_{\alpha(1)} \rightarrow a'_1 \quad \cdots \quad f_k : a_{\alpha(k)} \rightarrow a'_k} \quad \frac{\langle \alpha, f_1, \dots, f_k \rangle : \langle a_1, \dots, a_k \rangle \rightarrow \langle a'_1, \dots, a'_k \rangle}{\langle \alpha, f_1, \dots, f_k \rangle : \langle a_1, \dots, a_k \rangle \rightarrow \langle a'_1, \dots, a'_k \rangle}}$$

Derivations:

$$\frac{\frac{f_1 : \vec{a}_1 \rightarrow \langle \rangle, \dots, f_i : \vec{a}_i \rightarrow \langle a \rangle, \dots, f_n : \vec{a}_n \rightarrow \langle \rangle}{x_1 : \vec{a}_1, \dots, x_i : \vec{a}_i, \dots, x_n : \vec{a}_n \vdash x_i : a} \quad \frac{\Gamma_0 \vdash M : \langle a_1, \dots, a_k \rangle \Rightarrow a \quad (\Gamma_i \vdash N : a_i)_{i=1}^k \quad \eta : \Delta \rightarrow \bigotimes_{i=0}^k \Gamma_i}{\Delta \vdash MN : a}}{\frac{\Delta, x : \vec{a} \vdash M : a}{\Delta \vdash \lambda x. M : \vec{a} \Rightarrow a}}$$

Figure 3.3. – Category of Types D_A and parametric Intersection Type System E_A^S .

We remark that $D_A \cong (SD_A^o \times D_A) \sqcup A$. We now define our retraction pair in $\mathcal{S}\text{-CatSym}$:

$$i : (S(SD_A^o \times D_A))^o \times D_A \rightarrow \text{Set}$$

$$\langle \vec{d}, a \rangle \mapsto SD_A(S(\iota)(\vec{d}), \langle a \rangle)$$

$$j : SD_A^o \times (SD_A^o \times D_A) \rightarrow \text{Set}$$

$$\langle \vec{a}', \langle \vec{a}, a \rangle \rangle \mapsto SD_A(\vec{a}', \langle \iota(\vec{a}, a) \rangle)$$

Theorem 3.3.3. *We have that $j \circ i \cong 1_{D_A}$.*

The *interpretation* of a λ -term is defined by induction in the usual way, following the definition given in [AC98, Section 4.6]. We fix a constant type D such that $D = D \Rightarrow D$.⁵

1. On types:

$$\llbracket D \rrbracket = D_A$$

⁵ It is worth noting that we do not require for this equation to be semantically satisfied, *i.e.* we consider non-extensional models.

3. Intersection Type Distributors – 3.4. Intersection Types as Distributors

$$\llbracket \Gamma = A_1, \dots, A_n \rrbracket = \overbrace{\llbracket D \rrbracket \& \dots \& \llbracket D \rrbracket}^{\text{n times}}$$

2. On terms:

$$\begin{aligned} \llbracket x_1 : D, \dots, x_n : D \vdash x_i : D \rrbracket &= \pi_{i,n} \\ \llbracket \Gamma \vdash \lambda x. M : D \rrbracket &= i \circ \lambda(\llbracket \Gamma, x : D \vdash M : D \rrbracket) \\ \llbracket \Gamma \vdash PQ : D \rrbracket &= ev_{D,D} \circ \langle j \circ \llbracket \Gamma \vdash P : D \rrbracket, \llbracket \Gamma \vdash Q : D \rrbracket \rangle. \end{aligned}$$

$$\begin{aligned} \llbracket x \rrbracket_{\vec{x}}(\Delta, a) &= SD^n(\Delta, \langle \rangle, \dots, \langle a \rangle, \dots, \langle \rangle) \\ \llbracket \lambda x. M \rrbracket_{\vec{x}}(\Delta, a) &= \begin{cases} \llbracket M \rrbracket_{\vec{x} \oplus \langle x \rangle}(\Delta \oplus \langle \vec{a}' \rangle, a') & \text{if } a = \iota(\vec{a}', a') \\ \emptyset & \text{otherwise.} \end{cases} \end{aligned}$$

$$\llbracket MN \rrbracket_{\vec{x}}(\Delta, a) = \int^{\vec{a} = \langle a_1, \dots, a_k \rangle \in SD} \int^{\Gamma_0, \dots, \Gamma_k \in SD^n} \llbracket M \rrbracket_{\vec{x}}(\Gamma_0, \iota(\vec{a}, a)) \times \prod_{i=1}^k \llbracket N \rrbracket_{\vec{x}}(\Gamma_i, a_i) \times SD^n(\Delta, \bigotimes_{i=0}^k \Gamma_i)$$

Figure 3.4. – Denotation of λ -terms.

Remark 3.3.4. We recall that SD_A is the free S -monoidal category build from D_A . Hence, given a morphism $\langle \alpha, \vec{f} \rangle = \langle \alpha, f_1, \dots, f_{k'} \rangle : \langle a_1, \dots, a_k \rangle \rightarrow \langle b_1, \dots, b_{k'} \rangle$ we have the following canonical decomposition

$$\begin{array}{ccc} \langle a_1, \dots, a_k \rangle & \xrightarrow{\alpha^*} & \langle a_{\alpha(1)}, \dots, a_{\alpha(k')} \rangle \\ \langle \alpha, \vec{f} \rangle \downarrow & \swarrow \langle 1, f_1, \dots, f_{k'} \rangle & \\ \langle b_1, \dots, b_{k'} \rangle & & \end{array}$$

Where we recall that $\alpha^* = \langle \alpha, 1_{a_{\alpha(1)}}, \dots, 1_{a_{\alpha(k')}} \rangle$. We define the canonical representation of f as the couple $cr(f) = \langle \langle 1, f_1, \dots, f_{k'} \rangle, \alpha \rangle$. We define the structural function of f as the function $sm(f) = \alpha$. We recall that $\vec{a}\{\alpha\} = \langle a_{\alpha(1)}, \dots, a_{\alpha(k')} \rangle$. We have that $sm(g \circ f) = sm(f) \circ sm(g)$ and $sm(f \oplus g) = sm(f) \oplus sm(g)$. We also remark that we can canonically decompose a morphism with trivial structural function $f = \langle 1, \vec{f} \rangle : \vec{a} \rightarrow \bigoplus_{i \in [k]} \vec{b}_i$ as $\bigoplus_{i \in [k]} f_i : \bigoplus_{i \in [k]} \vec{a}_i \rightarrow \bigoplus_{i \in [k]} \vec{b}_i$ where $\bigoplus_{i \in [k]} f_i = \vec{f}$ and $\bigoplus_{i \in [k]} \vec{a}_i = \vec{a}$. These decompositions naturally extend to morphisms between tuples of type lists $\Delta = \langle \vec{a}_1, \dots, \vec{a}_n \rangle$.

3.4. Intersection Types as Distributors

Theorem 3.3.3 says that the category D_A is a non-extensional model for pure λ -calculus. We will denote, with a small abuse of language, SD_A as SD and D_A

3. Intersection Type Distributors – 3.4. Intersection Types as Distributors

as D . We now want to make explicit the idea that the semantics induced by this category is an intersection type system. We proceed as for the simply typed case.

We call *intersection type contexts*, or contexts for short, the objects of SD^n . We recall that SD^n admits a tensor product, that we denote as \otimes (Section 2.6.1). This tensor product inherits all the structure from \oplus , *i.e.*, if \oplus is symmetric (resp., semicartesian, relevant, cartesian) then also \otimes is so.

We define the *denotation* of a λ -term by induction in Figure 3.4. In general we have that $\llbracket M \rrbracket_{\vec{x}} : D \dashv\dashv SD^n$.

Remark 3.4.1. *The denotation of an application deserves some commentaries. Consider the functor*

$$\begin{aligned} F : SD^o \times SD \times (SD^n)^o \times D &\rightarrow \text{Set} \\ \langle \vec{a}, \vec{b} = \langle b_1, \dots, b_k \rangle, \Delta, a \rangle &\mapsto \\ \left(\llbracket M \rrbracket_{\vec{x}}(-, \vec{a} \Rightarrow a) \hat{\otimes} \widehat{\bigotimes}_{i=1}^k \llbracket N \rrbracket_{\vec{x}}(-, b_i) \right) &(\Delta) \end{aligned}$$

Then we can write the denotation of an application in a very suggestive way, via the Day convolution tensor product:

$$\begin{aligned} \llbracket MN \rrbracket_{\vec{x}}(\Delta, a) = \\ \left(\int^{\langle a_1, \dots, a_k \rangle \in SD} \llbracket M \rrbracket_{\vec{x}}(-, \langle a_1, \dots, a_k \rangle \Rightarrow a) \hat{\otimes} \widehat{\bigotimes}_{i=1}^k \llbracket N \rrbracket_{\vec{x}}(-, a_i) \right) &(\Delta). \end{aligned}$$

We observe that, for what we have seen in Section 2.4.1, we have that the denotation of an application can be rewritten as a coequalizer. Since the formula contains two coends, we will first make explicit the coend produced by the Day convolution. We fix $\vec{a}, \langle b_1, \dots, b_k \rangle \in SD, \Delta \in SD^n$ and $a \in D$. We have that

$$\begin{aligned} \left(\llbracket M \rrbracket_{\vec{x}}(-, \vec{a} \Rightarrow a) \hat{\otimes} \widehat{\bigotimes}_{i=1}^k \llbracket N \rrbracket_{\vec{x}}(-, b_i) \right) &(\Delta) = \\ \int^{\Gamma_0, \dots, \Gamma_k \in SD^n} \llbracket M \rrbracket_{\vec{x}}(\Gamma_0, \vec{a} \Rightarrow a) \times \prod_{i=1}^k \llbracket N \rrbracket_{\vec{x}}(\Gamma_i, b_i) \times SD^n(\Delta, \bigotimes_{j=0}^k \Gamma_j). & \quad (3.1) \end{aligned}$$

(3.1) is the coend of the functor

$$\begin{aligned} F_k : ((SD^n)^{k+1})^o \times (SD^n)^{k+1} &\rightarrow \text{Set} \\ \langle \langle \Gamma_0, \dots, \Gamma_k \rangle, \langle \Delta_0, \dots, \Delta_k \rangle \rangle &\mapsto \\ \llbracket M \rrbracket_{\vec{x}}(\Gamma_0, \vec{a} \Rightarrow a) \times \prod_{i=1}^k \llbracket N \rrbracket_{\vec{x}}(\Gamma_i, b_i) \times SD^n(\Delta, \bigotimes_{j=0}^k \Delta_j). & \end{aligned}$$

3. Intersection Type Distributors – 3.4. Intersection Types as Distributors

(3.1) is the coequalizer of the following diagram:

$$\sum_{\vec{\Gamma}, \vec{\Gamma}' \in SD^n} SD^n(\vec{\Gamma}', \vec{\Gamma}) \times F_k(\vec{\Gamma}, \vec{\Gamma}') \rightrightarrows \sum_{\vec{\Gamma} \in SD^n} F_k(\vec{\Gamma}, \vec{\Gamma}) \rightarrow \int^{\vec{\Gamma} \in SD^n} F_k(\vec{\Gamma}, \vec{\Gamma})$$

where the components of the two parallel arrows are

$$f_{\vec{\Gamma}, \vec{\Gamma}'} : SD^n(\vec{\Gamma}', \vec{\Gamma}) \times F_k(\vec{\Gamma}, \vec{\Gamma}') \rightarrow F_k(\vec{\Gamma}, \vec{\Gamma})$$

$$\langle \langle \eta_1, \dots, \eta_k \rangle, \varphi, \langle \psi_1, \dots, \psi_k \rangle, \eta \rangle \mapsto \langle \varphi, \langle \psi_1, \dots, \psi_k \rangle, (\bigotimes_{j=0}^k \eta_j) \circ \eta \rangle$$

and

$$g_{\vec{\Gamma}, \vec{\Gamma}'} : SD^n(\vec{\Gamma}', \vec{\Gamma}) \times F_k(\vec{\Gamma}, \vec{\Gamma}') \rightarrow F_k(\vec{\Gamma}', \vec{\Gamma}')$$

$$\langle \langle \eta_1, \dots, \eta_k \rangle, \varphi, \langle \psi_1, \dots, \psi_k \rangle, \eta \rangle \mapsto$$

$$\langle \llbracket M \rrbracket_{\vec{x}}(\eta_0, \vec{\Gamma}')(\varphi), \prod_{i=1}^k \llbracket N \rrbracket_{\vec{x}}(\langle \eta_1, \dots, \eta_k \rangle)(\langle \psi_1, \dots, \psi_k \rangle), \eta \rangle$$

where $\vec{\Gamma} = \langle \Gamma_1, \dots, \Gamma_k \rangle$, $\vec{\Gamma}' = \langle \Gamma'_1, \dots, \Gamma'_k \rangle$, $\varphi \in \llbracket M \rrbracket_{\vec{x}}(\Gamma_0, \vec{a} \Rightarrow a)$, $\psi_i \in \llbracket N \rrbracket_{\vec{x}}(\Gamma_i, a_i)$, $\eta : \Delta \rightarrow \bigotimes_{j=0}^k \Gamma'_j$ and $\eta_j : \Gamma'_j \rightarrow \Gamma_j$.

The coend

$$\left(\int^{\langle a_1, \dots, a_k \rangle \in SD} \llbracket M \rrbracket_{\vec{x}}(-, \langle a_1, \dots, a_k \rangle \Rightarrow a) \hat{\otimes} \widehat{\bigotimes}_{i=1}^k \llbracket N \rrbracket_{\vec{x}}(-, a_i) \right) (\Delta)$$

corresponds to the coequalizer of the following diagram:

$$\sum_{\vec{a}, \vec{a}' \in SD^n} SD(\vec{a}', \vec{a}) \times F(\vec{a}, \vec{a}', \Delta, a) \rightrightarrows \sum_{\vec{a} \in SD} F(\vec{a}, \vec{a}, \Delta, a) \rightarrow \int^{\vec{a} \in SD} F(\vec{a}, \vec{a}, \Delta, a)$$

where the components of the two parallel arrows are

$$f'_{\vec{a}, \vec{a}'} : SD(\vec{a}', \vec{a}) \times F(\vec{a}, \vec{a}', \Delta, a) \rightarrow \sum_{\vec{a} \in SD} F(\vec{a}, \vec{a}, \Delta, a)$$

$$\langle f = \langle \alpha, \vec{f} = f_1, \dots, f_{k'} \rangle, \varphi, \langle \psi_1, \dots, \psi_{k'} \rangle, \eta \rangle \mapsto$$

$$\langle \varphi, \prod_{i=\alpha(1)}^{\alpha(k')} \llbracket N \rrbracket_{\vec{x}}(\Gamma_{\alpha(1)}, \dots, \Gamma_{\alpha(k')}) (\vec{f}) (\langle \psi_1, \dots, \psi_{k'} \rangle), (1_{\Gamma_0} \otimes \alpha^*) \circ \eta \rangle$$

and

$$g'_{\vec{a}, \vec{a}'} : SD(\vec{a}', \vec{a}) \times F(\vec{a}, \vec{a}', \Delta, a) \rightarrow \sum_{\vec{a} \in SD} F(\vec{a}', \vec{a}', \Delta, a)$$

$$\langle f = \langle \alpha, \vec{f} = f_1, \dots, f_{k'} \rangle, \varphi, \langle \psi_1, \dots, \psi_{k'} \rangle, \eta \rangle \mapsto \langle \llbracket M \rrbracket_{\vec{x}}(\Gamma_0, f)(\varphi), \langle \psi_1, \dots, \psi_{k'} \rangle, \eta \rangle$$

3. Intersection Type Distributors – 3.4. Intersection Types as Distributors

where $\varphi \in \llbracket M \rrbracket_{\vec{x}}(\Gamma_0, \vec{a} \Rightarrow a)$, $\psi_i \in \llbracket N \rrbracket_{\vec{x}}(\Gamma_i, a'_i)$, for $i \in [k']\eta : \Delta \rightarrow \bigotimes_{j=0}^k \Gamma'_j$ and $f : \vec{a} \rightarrow \vec{a}' = \langle a'_1, \dots, a'_{k'} \rangle$.

Hence, we can rewrite the denotation of an application as the following quotient:

$$\left(\left(\sum_{\vec{a}=\langle a_1, \dots, a_k \rangle \in SD} \sum_{\Gamma_0, \dots, \Gamma_k \in SD^n} \llbracket M \rrbracket_{\vec{x}}(\Gamma_0, \iota(\vec{a}, a)) \times \prod_{i=1}^k \llbracket N \rrbracket_{\vec{x}}(\Gamma_i, a_i) \times SD^n(\Delta, \bigotimes_{i=0}^k \Gamma_i) \right) / \sim \right) / \sim'$$

where the two equivalence relations are generated by, respectively, the two following rules:

$$\langle \vec{a}, \langle \Gamma_0, \dots, \Gamma_{\text{len}(\vec{a})} \rangle, \varphi, \langle \psi_1, \dots, \psi_{\text{len}(\vec{a})} \rangle, \left(\bigotimes_{j=0}^k \eta_j \right) \circ \eta \rangle \sim$$

$$\langle \vec{a}, \langle \Gamma'_0, \dots, \Gamma'_{\text{len}(\vec{a})} \rangle, \llbracket M \rrbracket_{\vec{x}}(\eta_0, 1)(\varphi_0), \prod_{i=1}^{\text{len}(\vec{a})} \llbracket N \rrbracket_{\vec{x}}(\langle \eta_1, \dots, \eta_{\text{len}(\vec{a})} \rangle, 1)(\langle \psi_1, \dots, \psi_{\text{len}(\vec{a})} \rangle), \eta \rangle$$

where $\eta_j : \Gamma'_j \rightarrow \Gamma_j$ for $0 \leq j \leq \text{len}(\vec{a})$.

$$\langle \vec{a}, \vec{\Gamma} = \langle \Gamma_0, \dots, \Gamma_{\text{len}(\vec{a})} \rangle, \llbracket M \rrbracket_{\vec{x}}(\Gamma_0, \langle \alpha, \vec{f} \rangle)(\varphi), \langle \psi_1, \dots, \psi_{\text{len}(\vec{a})} \rangle, \eta \rangle \sim'$$

$$\langle \vec{a}', \langle \Gamma_0, \Gamma_{\alpha(1)} \dots, \Gamma_{\alpha(\text{len}(\vec{a}))} \rangle, \varphi, \prod_{i=\alpha(1)}^{\alpha(\text{len}(\vec{a}))} \llbracket N \rrbracket_{\vec{x}}(\vec{\Gamma}\{\alpha\}, \langle f_1, \dots, f_{\text{len}(\vec{a})} \rangle)(\langle \psi_{\alpha(1)}, \dots, \psi_{\alpha(\text{len}(\vec{a}))} \rangle), (1_{\Gamma_0} \otimes \alpha^*) \circ \eta \rangle$$

with $\vec{f} = \langle f_1, \dots, f_{\text{len}(\vec{a})} \rangle$ and $\langle \alpha, \vec{f} \rangle : \vec{a} \rightarrow \vec{a}'$.

3.4.1. The Denotation is Isomorphic to the Semantics

Also the untyped denotation of a term is isomorphic to its bicategorical interpretation *via* the Seely equivalence (Remark 2.6.6).

Theorem 3.4.2. *Let $M \in \Lambda$, $\vec{x} \supset FV(M)$ and $\Gamma \vdash M : D$ such that $\text{supp}(\Gamma) = \vec{x}$. We have a natural isomorphism*

$$\llbracket M \rrbracket_{\vec{x}} \cong \llbracket \Gamma \vdash M : D \rrbracket \circ_{Dist} \bar{\mu}_1.$$

Proof. Straightforward adaptation of the proof of Theorem 3.2.1. \square

Embedding of the Simply Typed Denotation in the Untyped One In this paragraph we establish a relationship between the typed and untyped denotations of a simply typed λ -term. Consider a grammar for simple types. Let ρ be an evaluation function $At \rightarrow \text{ob}(\text{Cat})$. We set $\rho(At) = \bigsqcup_{o \in AT} \rho(o)$.

Proposition 3.4.3. *Let A be a simple type. The category $\llbracket A \rrbracket_{\rho}$ is a full subcategory of $D_{\rho(At)}$.*

3. Intersection Type Distributors – 3.4. Intersection Types as Distributors

$$\begin{aligned}
& [g : a \rightarrow b] \left(\frac{f_1 : \vec{a}_1 \rightarrow \langle \rangle, \dots, f_i = \langle \alpha, f \rangle : \vec{a}_i \rightarrow \langle a \rangle, \dots, f_n : \vec{a}_n \rightarrow \langle \rangle}{x_1 : \vec{a}_1, \dots, x_i : \vec{a}_i, \dots, x_n : \vec{a}_n \vdash x_i : a} \right) = \\
& \frac{f_1 : \vec{a}_1 \rightarrow \langle \rangle, \dots, \langle g \rangle \circ f_i = \langle \alpha, g \circ f \rangle : \vec{a}_i \rightarrow \langle b \rangle, \dots, f_n : \vec{a}_n \rightarrow \langle \rangle}{x_1 : \vec{a}_1, \dots, x_i : \vec{a}_i, \dots, x_n : \vec{a}_n \vdash x_i : b} \\
& [\langle \alpha, \vec{g} \rangle \Rightarrow g : \vec{a} \Rightarrow a \rightarrow \vec{b} \Rightarrow b] \left(\frac{\frac{\pi}{\Delta, x : \vec{a} \vdash M : a}}{\Delta \vdash \lambda x. M : \vec{a} \Rightarrow a} \right) = \frac{[g] \pi \{ \langle 1, \langle \alpha, \vec{g} \rangle \rangle \}}{\Delta, x : \vec{b} \vdash M : b} \\
& \frac{\Delta \vdash \lambda x. M : \vec{b} \Rightarrow b}{\Delta \vdash \lambda x. M : \vec{b} \Rightarrow b} \\
& [g : a \rightarrow b] \left(\frac{\frac{\frac{\pi_0}{\Gamma_0 \vdash M : \vec{a} \Rightarrow a} \quad \left(\frac{\pi_i}{\Gamma_i \vdash M : a_i} \right)_{i=1}^k}{\Delta \vdash MN : a}}{\frac{[1 \Rightarrow g] \pi_0 \quad \left(\frac{\pi_i}{\Gamma_i \vdash M : a_i} \right)_{i=1}^k}{\Gamma_0 \vdash M : \vec{a} \Rightarrow b}} \quad \eta : \Delta \rightarrow \otimes_0^k \Gamma_j}{\Delta \vdash MN : a} \right) = \\
& \frac{\frac{[1 \Rightarrow g] \pi_0 \quad \left(\frac{\pi_i}{\Gamma_i \vdash M : a_i} \right)_{i=1}^k}{\Gamma_0 \vdash M : \vec{a} \Rightarrow b} \quad \eta : \Delta \rightarrow \otimes_{j=0}^k \Gamma_j}{\Delta \vdash MN : b}
\end{aligned}$$

Where $\vec{a} = \langle a_1, \dots, a_k \rangle$.

Figure 3.5. – Left action on derivations.

Proof. By induction on A , we exploit the isomorphism $D_{\rho(At)} \cong SD_{\rho(At)}^o \times D_{\rho(At)} \sqcup \rho(At)$. If $A = o$ the result is immediate by definition. If $A = B \Rightarrow C$ then, by IH, $\llbracket A \rrbracket_\rho \hookrightarrow D_{\rho(At)}$, $\llbracket B \rrbracket_\rho \hookrightarrow D_{\rho(At)}$. It is easy to see that we have $S\llbracket A \rrbracket_\rho^o \hookrightarrow SD_{\rho(At)}^o$. Thus

$$S\llbracket A \rrbracket_\rho^o \times \llbracket B \rrbracket_\rho \hookrightarrow SD_{\rho(At)}^o \times D_{\rho(At)} \xrightarrow{\iota} D_{\rho(At)}.$$

□

By the former proposition, we have that, for $\Gamma \vdash M : A$ and $\vec{x} = \text{supp}(\Gamma)$,

$$\llbracket M \rrbracket_\Gamma^A \cong (\llbracket M \rrbracket_{\vec{x}})_{|\Gamma, A}$$

where $(\llbracket M \rrbracket_{\vec{x}})_{|\Gamma, A}(\Delta, a)$ is the restriction functor. Then the simply typed denotation is just the restriction of the untyped denotation. For this reason, in what follows we will focus just on the untyped denotation.

3.4.2. The Denotation as an Intersection Type System

We now give a type-theoretic description of the denotation of a λ -term. We define the intersection type system E_A^S , where types and morphisms live in the category D_A (Figure 3.3). Thanks to this type theoretic description, we can present

3. Intersection Type Distributors – 3.4. Intersection Types as Distributors

the denotation's action on morphism as right and left actions on type derivations:

$$\begin{array}{ccc} \begin{array}{c} \pi \\ \vdots \\ \Delta \vdash M : a \end{array} & \rightsquigarrow & \begin{array}{c} ([f] \pi) \{\eta\} \\ \vdots \\ \Delta' \vdash M : a' \end{array} \end{array}$$

with $f : a \rightarrow a'$ and $\eta : \Delta' \rightarrow \Delta$. The actions are inductively defined in Figures 3.5 and 3.6.

Remark 3.4.4. *We observe that in the variable rule of our system (Figure 3.3) the morphisms $f_j : \vec{a}_j \rightarrow \langle \rangle$ for $j \neq i \in [n]$, are unique, by the structure of resource monads. In particular, if S is irrelevant (cartesian or semicartesian resource monad), f_j is the terminal morphism $\top_{\vec{a}_j} : \vec{a}_j \rightarrow \langle \rangle$. Otherwise (linear or relevant resource monad) f_j is the identity $1_{\langle \rangle} : \langle \rangle \rightarrow \langle \rangle$. We shall denote this unique universal morphism as $\diamond_{\vec{a}_j}$.*

Lemma 3.4.5 (Actions). *Let $\pi \in E_A^S$ of conclusion $\Delta \vdash M : a$ and let $\eta : \Delta' \rightarrow \Delta, \eta' : \Delta'' \rightarrow \Delta', g : a \rightarrow a', f' : a' \rightarrow a''$. The following statements hold.*

1. $(\pi\{\eta\})\{\eta'\} = \pi\{\eta \circ \eta'\}$.
2. $[f](\pi\{\eta\}) = ([f]\pi)\{\eta\}$.
3. $[f']([f]\pi) = [f' \circ f]\pi$.

Proof. 1. By induction on the structure of M . In the variable and application case the result is given by morphisms composition, while in the abstraction case is an immediate application of the IH.

2. By induction on the structure of M . If $M = x_i$ Then $\pi =$

$$\frac{f_1 : \vec{a}_1 \rightarrow \langle \rangle, \dots, f_i = \langle \alpha, g \rangle : \vec{a}_i \rightarrow \langle a \rangle, \dots, f_n : \vec{a}_n \rightarrow \langle \rangle}{x_1 : \vec{a}_1, \dots, x_i : \vec{a}_i, \dots, x_n : \vec{a}_n \vdash x_i : a}$$

Let $\eta = \langle g_1, \dots, g_n \rangle$, then $\pi\{\eta\} =$

$$\frac{f_1 \circ g_1 : \vec{b}_1 \rightarrow \langle \rangle, \dots, f_i \circ g_i : \vec{b}_i \rightarrow \langle a \rangle, \dots, f_n \circ g_n : \vec{b}_n \rightarrow \langle \rangle}{x_1 : \vec{b}_1, \dots, x_i : \vec{b}_i, \dots, x_n : \vec{b}_n \vdash x_i : a}$$

and $[f](\pi\{\eta\}) =$

$$\frac{f_1 \circ g_1 : \vec{b}_1 \rightarrow \langle \rangle, \dots, \langle f \rangle \circ (f_i \circ g_i) : \vec{b}_i \rightarrow \langle a \rangle, \dots, f_n \circ g_n : \vec{b}_n \rightarrow \langle \rangle}{x_1 : \vec{b}_1, \dots, x_i : \vec{b}_i, \dots, x_n : \vec{b}_n \vdash x_i : a}$$

then the result derives by associativity of morphisms composition, since $\langle f \rangle \circ (f_i \circ g_i) = (\langle f \rangle \circ f_i) \circ g_i$.

3. Intersection Type Distributors – 3.4. Intersection Types as Distributors

If $M = \lambda x.M'$ then $\pi =$

$$\frac{\begin{array}{c} \pi' \\ \vdots \\ \Delta, x : \vec{a} \vdash M : a \end{array}}{\Delta \vdash \lambda x.M : \vec{a} \Rightarrow a}$$

then $\pi\{\eta\} =$

$$\frac{\begin{array}{c} \pi'\{\eta \oplus \langle 1_{\vec{a}} \rangle\} \\ \vdots \\ \Delta', x : \vec{a} \vdash M : a \end{array}}{\Delta' \vdash \lambda x.M : \vec{a} \Rightarrow a}$$

Let $f = \langle \alpha, \vec{g} \rangle \Rightarrow g : \vec{a} \Rightarrow a \rightarrow \vec{a}' \Rightarrow a'$. Then $[f](\pi\{\eta\}) =$

$$\frac{\begin{array}{c} ([g](\pi'\{\eta \oplus \langle 1_{\vec{a}} \rangle\}))\{1_{\Delta'} \oplus \langle \langle \alpha, \vec{g} \rangle \rangle\} \\ \vdots \\ \Delta', x : \vec{a}' \vdash M : a' \end{array}}{\Delta' \vdash \lambda x.M : \vec{a}' \Rightarrow a'}$$

By IH we have $([g](\pi'\{\eta \oplus \langle 1_{\vec{a}} \rangle\}))\{1_{\Delta'} \oplus \langle \langle \alpha, \vec{g} \rangle \rangle\} = (([g]\pi')\{\eta \oplus \langle 1_{\vec{a}} \rangle\})\{1_{\Delta'} \oplus \langle \langle \alpha, \vec{g} \rangle \rangle\}$. Since $(\eta \oplus \langle 1_{\vec{a}} \rangle) \circ (1_{\Delta'} \oplus \langle \langle \alpha, \vec{g} \rangle \rangle) = \eta \oplus \langle \langle \alpha, \vec{g} \rangle \rangle = (1_{\Delta} \oplus \langle \langle \alpha, \vec{g} \rangle \rangle) \circ (\eta \oplus \langle 1_{\vec{a}} \rangle)$ we apply the former point of the lemma and we conclude.

If $M = PQ$ the result is immediate by definition.

3. We prove the abstraction case, the others being trivial applications of the definition and the IH. Let $M = \lambda x.M'$ and

$$\pi = \frac{\begin{array}{c} \pi' \\ \vdots \\ \Delta, x : \vec{a} \vdash M' : a \end{array}}{\Delta \vdash \lambda x.M' : \vec{a} \Rightarrow a}$$

Let $f = \langle \alpha, \vec{g} \rangle \Rightarrow g : \vec{a} \Rightarrow a \rightarrow \vec{a}' \Rightarrow a'$ and $f' = \langle \alpha', \vec{g}' \rangle \Rightarrow g' : \vec{a}' \Rightarrow a' \rightarrow \vec{a}'' \Rightarrow a''$. By definition

$$[f']([f]\pi) = \frac{\begin{array}{c} ([g'](([g]\pi'))\{1_{\Delta} \oplus \langle \langle \alpha, \vec{g} \rangle \rangle\})\{1_{\Delta} \oplus \langle \langle \alpha', \vec{g}' \rangle \rangle\}) \\ \vdots \\ \Delta, x : \vec{a}'' \vdash M' : a'' \end{array}}{\Delta \vdash \lambda x.M' : \vec{a}'' \Rightarrow a''}$$

By the former point of this lemma we have $([g'](([g]\pi'))\{1_{\Delta} \oplus \langle \langle \alpha, \vec{g} \rangle \rangle\})\{1_{\Delta} \oplus \langle \langle \alpha', \vec{g}' \rangle \rangle\} = (([g'](([g]\pi'))\{1_{\Delta} \oplus \langle \langle \alpha, \vec{g} \rangle \rangle\})\{1_{\Delta} \oplus \langle \langle \alpha', \vec{g}' \rangle \rangle\})$ and by the first point of this lemma $(([g'](([g]\pi'))\{1_{\Delta} \oplus \langle \langle \alpha, \vec{g} \rangle \rangle\})\{1_{\Delta} \oplus \langle \langle \alpha', \vec{g}' \rangle \rangle\}) = ([g'](([g]\pi'))\{1_{\Delta} \oplus \langle \langle \alpha, \vec{g} \rangle \circ \langle \alpha', \vec{g}' \rangle \rangle\})$. We then apply the IH and we get $([g' \circ g]\pi')\{1_{\Delta} \oplus \langle \langle \alpha, \vec{g} \rangle \circ \langle \alpha', \vec{g}' \rangle \rangle\}$.

$$\begin{array}{c}
 \left(\frac{f_1 : \vec{a}_1 \rightarrow \langle \rangle, \dots, f_i : \vec{a}_i \rightarrow \langle a \rangle, \dots, f_n : \vec{a}_n \rightarrow \langle \rangle}{x_1 : \vec{a}_1, \dots, x_i : \vec{a}_i, \dots, x_n : \vec{a}_n \vdash x_i : a} \right) \{\eta\} = \\
 \frac{f_1 \circ g_1 : \vec{b}_1 \rightarrow \langle \rangle, \dots, f_i \circ g_i : \vec{b}_i \rightarrow \langle a \rangle, \dots, f_n \circ g_n : \vec{b}_n \rightarrow \langle \rangle}{x_1 : \vec{b}_1, \dots, x_i : \vec{b}_i, \dots, x_n : \vec{b}_n \vdash x_i : a} \\
 \left(\frac{\pi}{\Delta, x : \vec{a} \vdash M : a} \right) \{\eta\} = \frac{\pi \{\eta \oplus \langle 1 \rangle\}}{\Delta', x : \vec{a} \vdash M : a} \\
 \frac{\Delta \vdash \lambda x. M : \vec{a} \Rightarrow a}{\Delta \vdash \lambda x. M : \vec{a} \Rightarrow a} \\
 \left(\frac{\pi_1}{\Gamma_1 \vdash M : \vec{a} \Rightarrow a} \quad \left(\frac{\pi_i}{\Gamma_i \vdash M : a_i} \right)_{i=1}^k \quad \theta : \Delta \rightarrow \bigotimes_{j=0}^k \Gamma_j \right) \{\eta\} = \\
 \frac{\Delta \vdash MN : a}{\Delta \vdash MN : a} \\
 \frac{\pi_1}{\Gamma_1 \vdash M : \vec{a} \Rightarrow a} \quad \left(\frac{\pi_i}{\Gamma_i \vdash M : a_i} \right)_{i=1}^k \quad \theta \circ \eta : \Delta' \rightarrow \bigotimes_{j=0}^k \Gamma_j}{\Delta' \vdash MN : a}
 \end{array}$$

Where $\vec{a} = \langle a_1, \dots, a_k \rangle$ and $\eta = \langle g_1, \dots, g_n \rangle : \Delta' \rightarrow \Delta$.

Figure 3.6. – Right action on derivations.

$\langle \alpha', \vec{g}' \rangle \rangle \}} \}} \}. Then we can conclude, since by definition of left action$

$$\begin{array}{c}
 ([g' \circ g] \pi') \{1_{\Delta} \oplus \langle \langle \alpha, \vec{g} \rangle \circ \langle \alpha', \vec{g}' \rangle \rangle \} \\
 [f' \circ f] \pi = \frac{\Delta, x : \vec{a}'' \vdash M' : a''}{\Delta \vdash \lambda x. M' : \vec{a}'' \Rightarrow a''}
 \end{array}$$

□

Congruence on Type Derivations The definition of denotation of an application MN depends on the notion of coend. As we saw (Remark 3.4.1), in the Set enriched setting this notion boils down to an appropriate quotient sum of sets. Hence, if we want to give a syntactic presentation of the denotation *via* the intersection type system E_A^S , we shall need to translate the quotient in the setting of type derivations.

A relation $R \subseteq (E_A^S)^2$ is called a *congruence* if it is an equivalence relation and it satisfies the following additional conditions:

1. If

$$\frac{\pi}{\Delta, x : \vec{a} \vdash M : a} \quad R \quad \frac{\pi'}{\Delta, x : \vec{a} \vdash M : a}$$

3. Intersection Type Distributors – 3.4. Intersection Types as Distributors

then

$$\frac{\begin{array}{c} \pi \\ \vdots \\ \Delta, x : \vec{a} \vdash M : a \end{array}}{\Delta \vdash \lambda x.M : \vec{a} \Rightarrow a} \quad R \quad \frac{\begin{array}{c} \pi' \\ \vdots \\ \Delta, x : \vec{a} \vdash M : a \end{array}}{\Delta \vdash \lambda x.M : \vec{a} \Rightarrow a} .$$

2. If

$$\frac{\begin{array}{c} \pi_0 \\ \vdots \\ \Delta \vdash M : \vec{a} = \langle a_1, \dots, a_k \rangle \Rightarrow a \end{array}}{\Delta \vdash M : \vec{a} = \langle a_1, \dots, a_k \rangle \Rightarrow a} \quad R \quad \frac{\begin{array}{c} \pi'_0 \\ \vdots \\ \Delta \vdash M : \vec{a} = \langle a_1, \dots, a_k \rangle \Rightarrow a \end{array}}{\Delta \vdash M : \vec{a} = \langle a_1, \dots, a_k \rangle \Rightarrow a}$$

and

$$\left(\begin{array}{c} \pi_i \\ \vdots \\ \Gamma_i \vdash N : a_i \end{array} \right)_{i=1}^k \quad R \quad \left(\begin{array}{c} \pi'_i \\ \vdots \\ \Gamma_i \vdash N : a'_i \end{array} \right)_{i=1}^k$$

then

$$\frac{\begin{array}{c} \pi_0 \\ \vdots \\ \Delta \vdash M : \vec{a} \Rightarrow a \end{array} \quad \left(\begin{array}{c} \pi_i \\ \vdots \\ \Gamma_i \vdash N : a_i \end{array} \right)_{i=1}^k \quad \eta : \Delta \rightarrow \bigotimes_{j=0}^k \Gamma_j \quad R}{\Delta \vdash MN : a}$$

$$\frac{\begin{array}{c} \pi'_0 \\ \vdots \\ \Delta \vdash M : \vec{a} \Rightarrow a \end{array} \quad \left(\begin{array}{c} \pi'_i \\ \vdots \\ \Gamma_i \vdash N : a'_i \end{array} \right)_{i=1}^k \quad \eta : \Delta \rightarrow \bigotimes_{j=0}^k \Gamma_j}{\Delta \vdash MN : a}$$

We set $\tilde{\pi}$ as the equivalence class of π for the smallest congruence generated by following two rules:

$$\frac{\begin{array}{c} \pi_0 \\ \vdots \\ \Gamma_0 \vdash M : \vec{b} \Rightarrow a \end{array} \quad \left(\begin{array}{c} [f_i] \pi_{\alpha(i)} \\ \vdots \\ \Gamma_{\alpha(i)} \vdash M : b_i \end{array} \right)_{i=1}^{k'} \quad (1 \otimes \alpha^*) \circ \eta : \Delta \rightarrow \Gamma_0 \otimes \bigotimes_{i=1}^{k'} \Gamma_{\alpha(i)}}{\Delta \vdash MN : a} \quad \sim \quad (3.2)$$

$$\frac{\begin{array}{c} [\langle \alpha, \vec{f} \rangle \Rightarrow 1] \pi_0 \\ \vdots \\ \Gamma_0 \vdash M : \vec{a} \Rightarrow a \end{array} \quad \left(\begin{array}{c} \pi_i \\ \vdots \\ \Gamma_i \vdash N : a_i \end{array} \right)_{i=1}^k \quad \eta : \Delta \rightarrow \bigotimes_{j=0}^k \Gamma_j}{\Delta \vdash MN : a}$$

3. Intersection Type Distributors – 3.4. Intersection Types as Distributors

$$\frac{\begin{array}{c} \pi_0\{\theta_0\} \\ \vdots \\ \Gamma_0 \vdash M : \vec{a} \Rightarrow a \end{array} \quad \left(\begin{array}{c} \pi_i\{\theta_i\} \\ \vdots \\ \Gamma_i \vdash N : a_i \end{array} \right)_{i=1}^k \quad \eta : \Delta \rightarrow \bigotimes_{j=0}^k \Gamma_j}{\Delta \vdash MN : a} \sim \quad (3.3)$$

$$\frac{\begin{array}{c} \pi_0 \\ \vdots \\ \Gamma'_0 \vdash M : \vec{a} \Rightarrow a \end{array} \quad \left(\begin{array}{c} \pi_i \\ \vdots \\ \Gamma'_i \vdash N : a_i \end{array} \right)_{i=1}^k \quad (\bigotimes_{j=0}^k \theta_j) \circ \eta : \Delta \rightarrow \bigotimes_{j=0}^k \Gamma'_j}{\Delta \vdash MN : a}$$

Where $\langle \alpha, f_1, \dots, f_{k'} \rangle : \vec{a} = \langle a_1, \dots, a_k \rangle \rightarrow \vec{b} = \langle b_1, \dots, b_{k'} \rangle$ and $\theta_i : \Gamma_i \rightarrow \Gamma'_i$.
We prove that the actions are stable under congruence.

Lemma 3.4.6. *Let $\pi \in E_A^S$ of conclusion $\Delta \vdash M : a$ and let $\theta : \Delta' \rightarrow \Delta, g : a \rightarrow a'$. The following statements hold.*

1. *If $\pi \sim \pi'$ then $\pi\{\theta\} \sim \pi'\{\theta\}$.*
2. *If $\pi \sim \pi'$ then $[g]\pi \sim [g]\pi'$.*

Proof. By induction on π . We just prove the cases (3.2) and (3.3), the others being trivial applications of the definitions the IH.

1. (3.2) Let $\pi =$

$$\frac{\begin{array}{c} \pi_0 \\ \vdots \\ \Gamma_0 \vdash M : \vec{b} \Rightarrow a \end{array} \quad \left(\begin{array}{c} [f_i]\pi_{\alpha(i)} \\ \vdots \\ \Gamma_{\alpha(i)} \vdash N : b_i \end{array} \right)_{i=1}^{k'} \quad (1 \otimes \alpha^*) \circ \eta : \Delta \rightarrow \Gamma_0 \otimes \bigotimes_{i=1}^{k'} \Gamma_{\alpha(i)}}{\Delta \vdash MN : a}$$

and $\pi' =$

$$\frac{\begin{array}{c} [\langle \alpha, \vec{f} \rangle \Rightarrow 1]\pi_0 \\ \vdots \\ \Gamma_0 \vdash M : \vec{a} \Rightarrow a \end{array} \quad \left(\begin{array}{c} \pi_i \\ \vdots \\ \Gamma_i \vdash N : a_i \end{array} \right)_{i=1}^k \quad \eta : \Delta \rightarrow \bigotimes_{j=0}^k \Gamma_j}{\Delta \vdash MN : a}$$

by definition $\pi\{\theta\} =$

$$\frac{\begin{array}{c} \pi_0 \\ \vdots \\ \Gamma_0 \vdash M : \vec{b} \Rightarrow a \end{array} \quad \left(\begin{array}{c} [f_i]\pi_{\alpha(i)} \\ \vdots \\ \Gamma_{\alpha(i)} \vdash N : b_i \end{array} \right)_{i=1}^{k'} \quad ((1 \otimes \alpha^*) \circ \eta) \circ \theta : \Delta' \rightarrow \Gamma_0 \otimes \bigotimes_{i=1}^{k'} \Gamma_{\alpha(i)}}{\Delta' \vdash MN : a}$$

3. Intersection Type Distributors – 3.4. Intersection Types as Distributors

and $\pi'\{\theta\} =$

$$\frac{\begin{array}{c} \langle \alpha, \vec{f} \rangle \Rightarrow 1] \pi_0 \\ \vdots \\ \Gamma_0 \vdash M : \vec{a} \Rightarrow a \end{array} \quad \left(\begin{array}{c} \pi_i \\ \vdots \\ \Gamma_i \vdash N : a_i \end{array} \right)_{i=1}^k \quad \eta \circ \theta : \Delta' \rightarrow \bigotimes_{j=0}^k \Gamma_j}{\Delta' \vdash MN : a}$$

then we can conclude by associativity of composition.

(3.3) Let $\pi =$

$$\frac{\begin{array}{c} \pi_0\{\theta_0\} \\ \vdots \\ \Gamma_0 \vdash M : \vec{a} \Rightarrow a \end{array} \quad \left(\begin{array}{c} \pi_i\{\theta_i\} \\ \vdots \\ \Gamma_i \vdash N : a_i \end{array} \right)_{i=1}^k \quad \eta : \Delta \rightarrow \bigotimes_{j=0}^k \Gamma_j}{\Delta \vdash MN : a}$$

and $\pi' =$

$$\frac{\begin{array}{c} \pi_0 \\ \vdots \\ \Gamma'_0 \vdash M : \vec{a} \Rightarrow a \end{array} \quad \left(\begin{array}{c} \pi_i \\ \vdots \\ \Gamma'_i \vdash N : a_i \end{array} \right)_{i=1}^k \quad (\bigotimes_{j=0}^k \theta_j) \circ \eta : \Delta \rightarrow \bigotimes_{j=0}^k \Gamma'_j}{\Delta \vdash MN : a}$$

we have $\pi\{\theta\} =$

$$\frac{\begin{array}{c} \pi_0\{\theta_0\} \\ \vdots \\ \Gamma_0 \vdash M : \vec{a} \Rightarrow a \end{array} \quad \left(\begin{array}{c} \pi_i\{\theta_i\} \\ \vdots \\ \Gamma_i \vdash N : a_i \end{array} \right)_{i=1}^k \quad \eta \circ \theta : \Delta \rightarrow \bigotimes_{j=0}^k \Gamma_j}{\Delta' \vdash MN : a}$$

and $\pi'\{\theta\} =$

$$\frac{\begin{array}{c} \pi_0 \\ \vdots \\ \Gamma'_0 \vdash M : \vec{a} \Rightarrow a \end{array} \quad \left(\begin{array}{c} \pi_i \\ \vdots \\ \Gamma'_i \vdash N : a_i \end{array} \right)_{i=1}^k \quad ((\bigotimes_{j=0}^k \theta_j) \circ \eta) \circ \theta : \Delta \rightarrow \bigotimes_{j=0}^k \Gamma'_j}{\Delta' \vdash MN : a}$$

we conclude again by associativity of composition.

2. (3.2) Let $\pi =$

$$\frac{\begin{array}{c} \pi_0 \\ \vdots \\ \Gamma_0 \vdash M : \vec{b} \Rightarrow a \end{array} \quad \left(\begin{array}{c} [f_i]\pi_{\alpha(i)} \\ \vdots \\ \Gamma_{\alpha(i)} \vdash N : b_i \end{array} \right)_{i=1}^{k'} \quad (1 \otimes \alpha^*) \circ \eta : \Delta \rightarrow \Gamma_0 \otimes \bigotimes_{i=1}^{k'} \Gamma_{\alpha(i)}}{\Delta \vdash MN : a}$$

3. Intersection Type Distributors – 3.4. Intersection Types as Distributors

and $\pi' =$

$$\frac{\begin{array}{c} [\langle \alpha, \vec{f} \rangle \Rightarrow 1] \pi_0 \\ \vdots \\ \Gamma_0 \vdash M : \vec{a} \Rightarrow a \end{array} \quad \left(\begin{array}{c} \pi_i \\ \vdots \\ \Gamma_i \vdash N : a_i \end{array} \right)_{i=1}^k \quad \eta : \Delta \rightarrow \bigotimes_{j=0}^k \Gamma_j}{\Delta' \vdash MN : a}$$

by definition $[g]\pi =$

$$\frac{\begin{array}{c} [1 \Rightarrow g] \pi_0 \\ \vdots \\ \Gamma_0 \vdash M : \vec{b} \Rightarrow a \end{array} \quad \left(\begin{array}{c} [f_i] \pi_{\alpha(i)} \\ \vdots \\ \Gamma_{\alpha(i)} \vdash N : b_i \end{array} \right)_{i=1}^{k'} \quad (1 \otimes \alpha^*) \circ \eta : \Delta \rightarrow \Gamma_0 \otimes \bigotimes_{i=1}^{k'} \Gamma_{\alpha(i)}}{\Delta \vdash MN : a'}$$

and $[g]\pi' =$

$$\frac{\begin{array}{c} [1 \Rightarrow g]([\langle \alpha, \vec{f} \rangle \Rightarrow 1] \pi_0) \\ \vdots \\ \Gamma_0 \vdash M : \vec{a} \Rightarrow a \end{array} \quad \left(\begin{array}{c} \pi_i \\ \vdots \\ \Gamma_i \vdash N : a_i \end{array} \right)_{i=1}^k \quad \eta \circ \theta : \Delta' \rightarrow \bigotimes_{j=0}^k \Gamma_j}{\Delta \vdash MN : a'}$$

then we can conclude by Lemma 3.4.5, since $[1 \Rightarrow g]([\langle \alpha, \vec{f} \rangle \Rightarrow 1] \pi_0) = [\langle \alpha, \vec{f} \rangle \Rightarrow 1]([1 \Rightarrow g] \pi_0)$.

(3.3) Let $\pi =$

$$\frac{\begin{array}{c} \pi_0 \{\theta_0\} \\ \vdots \\ \Gamma_0 \vdash M : \vec{a} \Rightarrow a \end{array} \quad \left(\begin{array}{c} \pi_i \{\theta_i\} \\ \vdots \\ \Gamma_i \vdash N : a_i \end{array} \right)_{i=1}^k \quad \eta : \Delta \rightarrow \bigotimes_{j=0}^k \Gamma_j}{\Delta \vdash MN : a}$$

and $\pi' =$

$$\frac{\begin{array}{c} \pi_0 \\ \vdots \\ \Gamma'_0 \vdash M : \vec{a} \Rightarrow a \end{array} \quad \left(\begin{array}{c} \pi_i \\ \vdots \\ \Gamma'_i \vdash N : a_i \end{array} \right)_{i=1}^k \quad (\bigotimes_{j=0}^k \theta_j) \circ \eta : \Delta \rightarrow \bigotimes_{j=0}^k \Gamma'_j}{\Delta \vdash MN : a}$$

we have $[g]\pi =$

$$\frac{\begin{array}{c} [1 \Rightarrow g](\pi_0 \{\theta_0\}) \\ \vdots \\ \Gamma_0 \vdash M : \vec{a} \Rightarrow a \end{array} \quad \left(\begin{array}{c} \pi_i \{\theta_i\} \\ \vdots \\ \Gamma_i \vdash N : a_i \end{array} \right)_{i=1}^k \quad \eta : \Delta \rightarrow \bigotimes_{j=0}^k \Gamma_j}{\Delta \vdash MN : a'}$$

3. Intersection Type Distributors – 3.4. Intersection Types as Distributors

and $[g]\pi' =$

$$\frac{\begin{array}{c} [1 \Rightarrow g]\pi_0 \\ \vdots \\ \Gamma'_0 \vdash M : \vec{a} \Rightarrow a \end{array} \quad \left(\begin{array}{c} \pi_i \\ \vdots \\ \Gamma'_i \vdash N : a_i \end{array} \right)_{i=1}^k \quad ((\otimes_{j=0}^k \theta_j) \circ \eta) \circ \theta : \Delta \rightarrow \otimes_{j=0}^k \Gamma'_j}{\Delta \vdash MN : a'}$$

then we can conclude again by Lemma 3.4.5, since $[1 \Rightarrow g](\pi_0\{\theta_0\}) = ([1 \Rightarrow g]\pi_0)\{\theta_0\}$. □

Definition 3.4.7. Let $\vec{x} \supset \text{fv}(M)$ and $\text{len}(\vec{x}) = n$. We now define the S-intersection type distributor of M , $T_D(M)_{\vec{x}} : D \rightarrow SD^n$, as follows:

1. on objects

$$T_D(M)_{\vec{x}}(\Delta, a) = \left\{ \begin{array}{c} \tilde{\pi} \\ \vdots \\ \Delta \vdash M : a \end{array} \right\}$$

2. on morphisms

$$T_D(M)_{\vec{x}}(f, \eta) : T_D(M)_{\vec{x}}(\Delta, a) \rightarrow T_D(M)_{\vec{x}}(\Delta', a')$$

$$\tilde{\pi} \mapsto [f]\widetilde{\pi\{\eta\}}$$

We will often use the term intersection type distributor, keeping the parameter implicit. Given a simply typed λ -term $\Gamma \vdash M : A$, its intersection type distributor is just $T(M)_{\Gamma}^A = (T_{D_{\rho(A)}}(M)_{\text{supp}(\Gamma)})_{|\Gamma, A}$.

Lemma 3.4.8. $T_D(M)_{\vec{x}}$ is a functor.

Proof. The result is a corollary of Lemmas 3.4.5 and 3.4.6. Indeed, given $\eta : \Delta' \rightarrow \Delta$ and $f : a \rightarrow a'$ the function

$$T_D(M)_{\vec{x}}(f, \eta) : T_D(M)_{\vec{x}}(\Delta, a) \rightarrow T_D(M)_{\vec{x}}(\Delta', a')$$

$$\tilde{\pi} \mapsto [f]\widetilde{\pi\{\eta\}}$$

is well-defined since actions are stable under equivalence. Moreover, given $\eta' : \Delta'' \rightarrow \Delta'$ and $f' : a' \rightarrow a''$ we have that

$$T_D(M)_{\vec{x}}(f' \circ f, \eta \circ \eta') = T_D(M)_{\vec{x}}(f', \eta) \circ T_D(M)_{\vec{x}}(f, \eta')$$

pointwise, again by Lemmas 3.4.5 and 3.4.6. The preservation of identities is immediate by definition. □

We now prove that S-intersection type distributors are isomorphic to the bicategorical semantics of λ -calculus induced by S-categorical symmetric sequences, up to Seelye equivalence.

3. Intersection Type Distributors – 3.4. Intersection Types as Distributors

Remark 3.4.9. Let us denote as \sim_1 and \sim_2 respectively the smallest congruences generated by Rule 3.2 and Rule 3.3. For $\tilde{\pi}, \tilde{\pi}' \in E_A^S / \sim_1$ we set $\tilde{\pi} \sim_2 \tilde{\pi}'$ when we have $\pi \sim_2 \pi'$. Then we have the following isomorphism

$$(E_A^S / \sim_1) / \sim_2 \cong E_A^S / \sim.$$

Consider now the intersection type distributor of an application MN . If we develop the definition, by the former observation we get $T_D(MN)_{\vec{x}}(\Delta, a) \cong$

$$\sum_{\vec{a} \in SD} \left(\left(\sum_{\Gamma_0, \dots, \Gamma_k \in SD^n} T_D(P)_{\vec{x}}(\Gamma_0, \vec{a} \Rightarrow a) \times \prod_{i=1}^k T_D(Q)_{\vec{x}}(\Gamma_i, a_i) \times SD^n(\Delta, \bigotimes_{j=0}^k \Gamma_j) \right) / \sim_1 \right) / \sim_2$$

where the functor

$$\langle \vec{b}, \vec{a} = \langle a_1, \dots, a_k \rangle, \Delta \rangle \mapsto \left(\sum_{\Gamma_0, \dots, \Gamma_k \in SD^n} T_D(P)_{\vec{x}}(\Gamma_0, \vec{b} \Rightarrow a) \times \prod_{i=1}^k T_D(Q)_{\vec{x}}(\Gamma_i, a_i) \times SD^n(\Delta, \bigotimes_{j=0}^k \Gamma_j) \right) / \sim_1$$

acts on morphisms $g : \vec{b}' \rightarrow \vec{b}, f = \langle \alpha, f_1, \dots, f_{k'} \rangle : \vec{a} \rightarrow \vec{a}' = \langle a'_1, \dots, a'_{k'} \rangle$ as follows

$$\begin{aligned} & \langle \langle \Gamma_0, \dots, \Gamma_k \rangle, \widetilde{\pi_0}, \langle \pi_1, \dots, \pi_k \rangle, \eta \rangle \mapsto \\ & \langle \langle \Gamma_0, \dots, \Gamma_{\alpha(k')} \rangle, ([g \Rightarrow 1]\pi_0), \langle [f_1]\pi_{\alpha(1)}, \dots, [f_{k'}]\pi_{\alpha(k')} \rangle, (1 \circ \alpha^*) \circ \eta \rangle. \end{aligned}$$

Theorem 3.4.10. Let $M \in \Lambda$. We have a natural isomorphism

$$\llbracket M \rrbracket_{\vec{x}}(\Delta, a) \cong T_D(M)_{\vec{x}}(\Delta, a).$$

Proof. We set $\text{len}(\vec{x}) = n$. We prove it by induction on the structure of M . If $M = x$ then the result is immediate, since $\llbracket M \rrbracket_{\vec{x}}(\Delta, a)$ is just the hom-set functor $SD^n(\Delta, \langle \langle \rangle, \dots, \langle a \rangle, \dots, \langle \rangle \rangle)$. If $M = \lambda x.M'$ then the result follows immediately by IH. If $M = PQ$ we exploit the Remark 1. let $p \in \llbracket M \rrbracket_{\vec{x}}(\Delta, a)$ then $p = (b_0, \langle b_1, \dots, b_k \rangle, \vec{\beta})$ with $a_0 \in \llbracket P \rrbracket_{\vec{x}}(\Gamma_0, \iota(\langle \vec{a}, a \rangle))$, $\langle b_1, \dots, b_k \rangle \in \prod_{i=1}^k \llbracket Q \rrbracket_{\vec{x}}(\Gamma_i, a_i)$ for some $\vec{a} = \langle a_1, \dots, a_k \rangle \in SD$ and $\Gamma_j \in SD^n$. Following the definitions,

$$\begin{aligned} \llbracket PQ \rrbracket_{\vec{x}}(\Delta, a) &= \\ & \int^{(a_1, \dots, a_k) \in SD} \int^{\Gamma_0, \dots, \Gamma_k \in SD^n} \llbracket P \rrbracket_{\vec{x}}(\Gamma_0, \iota(\langle \vec{a}, a \rangle)) \\ & \times \prod_{i \in [k]} \llbracket Q \rrbracket_{\vec{x}}(\Gamma_i, a_i) \times SD^n(\Delta, \bigotimes_{j=0}^k (\Gamma_j)) \end{aligned}$$

3. Intersection Type Distributors – 3.4. Intersection Types as Distributors

By Remark 3.4.1, we get that:

$$\begin{aligned} & \llbracket PQ \rrbracket_{\vec{x}}(\Delta, a) \cong \\ & \sum_{\langle a_1, \dots, a_k \rangle \in SD} \left(\left(\sum_{\Gamma_0, \dots, \Gamma_k \in SD^n} \llbracket P \rrbracket_{\vec{x}}(\Gamma_0, \iota(\vec{a}, a)) \right. \right. \\ & \left. \left. \times \prod_{i \in [k]} \llbracket Q \rrbracket_{\vec{x}}(\Gamma_i, a_i) \times SD^n(\Delta, \bigotimes_{j=0}^k \Gamma_j) \right) / \sim \right) / \sim' \end{aligned}$$

The equivalences are defined as follows:

- let $b_0 \in \llbracket P \rrbracket_{\vec{x}}(\Gamma_0, \iota(\vec{a}, a))$, $\langle b_1, \dots, b_k \rangle \in \prod_{i=1}^k \llbracket Q \rrbracket_{\vec{x}}(\Gamma_i, a_i)$ and $\eta \in SD^n(\Delta, \bigotimes_{j=0}^k \Gamma_j)$.
- Let $\langle \alpha, \vec{f} \rangle = \langle \alpha, f_1, \dots, f_k \rangle : \langle a_1, \dots, a_k \rangle \rightarrow \langle a'_1, \dots, a'_{k'} \rangle$ and $\langle \theta_0, \dots, \theta_k \rangle : \langle \Gamma'_0, \dots, \Gamma'_k \rangle \rightarrow \langle \Gamma_0, \dots, \Gamma_k \rangle$.

Then we have

$$\langle \llbracket P \rrbracket_{\vec{x}}(\langle \alpha, \vec{f} \rangle)(b_0), \langle b_1, \dots, b_k \rangle, \eta \rangle$$

~

$$\langle b_0, \langle \llbracket Q \rrbracket_{\vec{x}}(f_1)(b_{\alpha(1)}), \dots, \llbracket Q \rrbracket_{\vec{x}}(f_n)(b_{\alpha(k')}) \rangle, \alpha^* \circ \eta \rangle$$

and

$$\langle \llbracket P \rrbracket_{\vec{x}}(\theta_0)(b_0), \langle \llbracket Q \rrbracket_{\vec{x}}(\theta_1)(b_1), \dots, \llbracket Q \rrbracket_{\vec{x}}(\theta_n)(b_n) \rangle, \eta \rangle$$

~'

$$\langle b_0, \langle b_1, \dots, b_n \rangle, \left(\bigotimes_{j=0}^k \theta_j \right) \circ \eta \rangle$$

BY IH, there is a natural isomorphism

$$f_{\Gamma, a}^P \llbracket P \rrbracket_{\vec{x}}(\Gamma, a) \cong T_D(P)_{\vec{x}}(\Gamma, a)$$

for $\vec{a} \in SD, a \in D, \Gamma \in SD^n$.

Again, by IH, Lemma 3.4.6 and a simple inspection of the definitions we get a natural isomorphism

$$F_{\Gamma, a}^Q : \llbracket Q \rrbracket_{\vec{x}}(\Gamma, a) \cong T_D(Q)_{\vec{x}}(\Gamma, a).$$

For $\vec{a} = \langle a_1, \dots, a_k \rangle, \vec{b} \in SD, \Delta \in SD^n$ we have then a natural isomorphism

$$\begin{aligned} & \int^{\Gamma_0, \dots, \Gamma_k \in SD^n} \llbracket P \rrbracket_{\vec{x}}(\Gamma_0, \vec{b} \Rightarrow a) \times \prod_{i=1}^k \llbracket Q \rrbracket_{\vec{x}}(\Gamma_i, a_i) \times SD^n(\Delta, \bigotimes_{j=0}^k \Gamma_j) \cong \\ & \left(\sum_{\Gamma_0, \dots, \Gamma_k} T_D(P)_{\vec{x}}(\Gamma_0, \vec{b} \Rightarrow a) \times \prod_{i=1}^k T_D(Q)_{\vec{x}}(\Gamma_i, a_i) \times SD^n(\Delta, \bigotimes_{j=0}^k \Gamma_j) \right) / \sim_1 \end{aligned}$$

3. Intersection Type Distributors – 3.4. Intersection Types as Distributors

where the congruence \sim_1 is defined in Remark 3.4.9. Then we can make the same kind of reasoning to get the following natural isomorphism

$$\int^{\vec{a} \in SD} \int^{\Gamma_0, \dots, \Gamma_k \in SD^n} \llbracket P \rrbracket_{\vec{x}}(\Gamma_0, \vec{a} \Rightarrow a) \times \prod_{i=1}^k \llbracket Q \rrbracket_{\vec{x}}(\Gamma_i, a_i) \times SD^n(\Delta, \bigotimes_{j=0}^k \Gamma_j) \cong$$

$$\sum_{\vec{a} \in SD} \left(\left(\sum_{\Gamma_0, \dots, \Gamma_k} T_D(P)_{\vec{x}}(\Gamma_0, \vec{a} \Rightarrow a) \times \prod_{i=1}^k T_D(Q)_{\vec{x}}(\Gamma_i, a_i) \times SD^n(\Delta, \bigotimes_{j=0}^k \Gamma_j) \right) / \sim_1 \right) / \sim_2 .$$

Then we can conclude by Remark 3.4.9. □

3.4.3. Type Derivations under Reduction

In this section we will prove that

$$\llbracket M \rrbracket_{\vec{x}}(\Delta, a) \cong \llbracket N \rrbracket_{\vec{x}}(\Delta, a)$$

when $M \rightarrow_\beta N$, refining the standard subject reduction and expansion for intersection types. Indeed, we recall that, by Theorem 3.4.2,

$$\llbracket M \rrbracket_{\vec{x}} \cong T_D(M)_{\vec{x}}$$

hence, if we prove that

$$\llbracket M \rrbracket_{\vec{x}}(\Delta, a) \cong \llbracket N \rrbracket_{\vec{x}}(\Delta, a)$$

when $M \rightarrow_\beta N$, in particular we have $T_D(M)_{\vec{x}} \cong T_D(N)_{\vec{x}}$. This means that we have a natural bijection between the set of equivalence classes of type derivations with conclusion $\Delta \vdash M : a$ and the set of equivalence classes of type derivations with conclusion $\Delta \vdash N : a$, that is what we called a *proof relevant* denotational semantics. In the next chapter we will see as the witness of the former natural isomorphism can be seen as a *reduction relation* on (equivalence classes of) type derivations.

Let $M, N \in \Lambda$, $(\text{fv}(M)/\{x\}) \cup \text{fv}(N) \subseteq \vec{x}$ and $x \notin \vec{x}$. We set $Sub_{\vec{x}}^{M,x,N}(\Delta, a) =$

$$\int^{\vec{a} \in SD} \int^{\Gamma_1, \dots, \Gamma_{\text{len}(\vec{a})} \in SD^n} \llbracket M \rrbracket_{\vec{x} \oplus \langle \vec{x} \rangle}(\Gamma_0 \oplus \langle \vec{a} \rangle, a) \times \prod_{i=1}^{l(\vec{a})} \llbracket N \rrbracket_{\vec{x}}(\Gamma_i, a_i) \times SD^n(\Delta, \bigotimes_{j=0}^k \Gamma_j).$$

We can now state the following substitution lemma:

Lemma 3.4.11. *Let $M, N \in \Lambda$, $(\text{fv}(M)/\{x\}) \cup \text{fv}(N) \subseteq \vec{x}$ and $x \notin \vec{x}$. We have a natural isomorphism*

$$\llbracket M[N/x] \rrbracket_{\vec{x}}(\Delta, a) \cong Sub_{\vec{x}}^{M,x,N}(\Delta, a).$$

3. Intersection Type Distributors – 3.4. Intersection Types as Distributors

Proof. The proof is detailed in Section A.4. \square

Remark 3.4.12. *Reconstructing all the passages of the proof of the former lemma and exploiting the constructions of Theorems 2.4.4 and 2.4.5, we give an explicit definition of the natural isomorphism*

$$\mathbf{sub}^{M,x,N} : \mathit{Sub}_{\vec{x}}^{M,x,N} \cong \llbracket M[N/x] \rrbracket_{\vec{x}}.$$

This will be very useful to define the interpretation of β -reduction steps.

Let $n = \text{len}(\vec{x})$. We work up to the natural isomorphism established in Theorem 3.4.10, hence considering elements of $\llbracket M \rrbracket_{\vec{x}}$ as (equivalence classes of) type derivations.

– *If $M = x$ then*

$$\begin{aligned} \mathit{Sub}_{\vec{x}}^{M,x,N}(\Delta, a) &= \\ \int^{\vec{a}, \Gamma_j} SD^{n+1}(\Gamma_0 \oplus \langle \vec{a} \rangle, \langle \langle \rangle, \dots, \langle \rangle, \langle a \rangle \rangle) \times \prod_{i \in [\text{len}(\vec{a})]} \llbracket N \rrbracket_{\vec{x}}(\Gamma_i, a_i) \times SD^n(\Delta, \bigotimes_{j=0}^{\text{len}(\vec{a})} \Gamma_j) \\ \mathbf{sub}_{\Delta, a}^{M,x,N} : \mathit{Sub}_{\vec{x}}^{M,x,N}(\Delta, a) &\cong \llbracket M[N/x] \rrbracket_{\vec{x}}(\Delta, a) \\ \langle \vec{a}, \langle \Gamma_j \rangle_{j=0}^{\text{len}(\vec{a})}, \langle \diamond_{\Gamma_0}, \langle \alpha, f \rangle \rangle, \langle \varphi_1, \dots, \varphi_{\text{len}(\vec{a})} \rangle, \eta \rangle &\mapsto [f] \varphi_{\alpha(1)} \{ \pi_{n, \alpha(1)} \circ \eta \}. \end{aligned}$$

where $\pi_{n, \alpha(1)} : \bigotimes_{j=0}^{\text{len}(\vec{a})} \Gamma_j \rightarrow \Gamma_{\alpha(1)}$ is indeed the projection in the case where S is an irrelevant resource monad, while it is simply the identity morphism $1_{\Gamma_{\alpha(1)}} : \Gamma_{\alpha(1)} \rightarrow \Gamma_{\alpha(1)}$ otherwise ⁶.

– *If $M = \lambda y.M'$ and $a = \vec{b} \Rightarrow b$ ⁷ then $\mathit{Sub}_{\vec{x}}^{M,x,N}(\Delta, \vec{b} \Rightarrow b) =$*

$$\begin{aligned} \int^{\vec{a} = \langle a_1, \dots, a_k \rangle \in SD} \int^{\Gamma_j \in SD^n} \llbracket M' \rrbracket_{\vec{x} \oplus \langle x \rangle \oplus \langle y \rangle}(\Gamma_0 \oplus \langle \vec{a} \rangle \oplus \langle \vec{b} \rangle, b) \times \prod_{i=1}^{l(\vec{a})} \llbracket N \rrbracket_{\vec{x}}(\Gamma_i, a_i) \times SD^n(\Delta, \bigotimes_{j=0}^k \Gamma_j) \\ \mathbf{sub}_{\Delta, a}^{M,x,N} : \mathit{Sub}_{\vec{x}}^{M,x,N}(\Delta, a) &\cong \llbracket M[N/x] \rrbracket_{\vec{x}}(\Delta, a) \\ \langle \vec{a}, \langle \Gamma_j \oplus \langle \langle \rangle \rangle \rangle_{j=0}^{\text{len}(\vec{a})}, \varphi, \langle \varphi_1, \dots, \varphi_{\text{len}(\vec{a})} \rangle, \eta \rangle &\stackrel{8}{\mapsto} \\ \mathbf{sub}_{\Delta \oplus \langle \vec{b} \rangle, b}^{M',x,N}(\langle \vec{a}, \langle \Gamma_j \oplus \langle \langle \rangle \rangle \rangle_{j=0}^{\text{len}(\vec{a})}, \varphi, \langle \varphi_1, \dots, \varphi_{\text{len}(\vec{a})} \rangle, \eta \rangle) \end{aligned}$$

where

$$\mathbf{sub}^{M',x,N} : \mathit{Sub}_{\vec{x} \oplus \langle y \rangle}^{M',x,N} \cong \llbracket M[N/x] \rrbracket_{\vec{x} \oplus \langle y \rangle}$$

is the natural isomorphism given by the IH.

6. It is easy to check that in the relevant and linear cases, \vec{a} is necessarily a singleton.

7. We recall that if the type is not an implication then the structures considered are empty.

8. We are working up the the straightforward "embedding" $\llbracket M \rrbracket_{\vec{x}}(\Delta, a) \rightarrow \llbracket M \rrbracket_{\vec{x} \oplus \langle y \rangle}(\Delta \oplus \langle \langle \rangle \rangle, a)$, where we suppose that $y \notin \text{fv}(M)$.

3. Intersection Type Distributors – 3.4. Intersection Types as Distributors

– If $M = PQ$ then

$$\text{Sub}_{M,x,N}(\Delta, a) = \int^{\vec{a}=\langle a_1, \dots, a_k \rangle} \int^{\Gamma_0, \Gamma_i} \llbracket PQ \rrbracket_{\vec{x} \oplus \langle x \rangle}(\Gamma_0 \oplus \langle \vec{a} \rangle, a) \times \prod_{i=1}^k \llbracket N \rrbracket_{\vec{x}}(\Gamma_i, a_i) \times SD^n(\Delta, \bigotimes_{j=0}^k \Gamma_j).$$

and we recall that by definition of substitution of an application and by IH

$$\begin{aligned} & \llbracket M\{N/x\} \rrbracket_{\vec{x}}(\Delta, a) \cong \\ & \int^{\vec{a}} \int^{\Gamma_j} \left(\int^{\vec{a}_0} \int^{\Gamma_{0,0}, \vec{\Gamma}_0} \llbracket P \rrbracket_{\vec{x} \oplus \langle x \rangle}(\Gamma_{0,0} \oplus \langle \vec{a}_0 \rangle, \vec{a} \Rightarrow a) \times \llbracket N \rrbracket_{\vec{x}}(\vec{\Gamma}, \vec{a}_0) \times SD^n(\Gamma_0, \Gamma_{0,0} \otimes \vec{\Gamma}_0) \right) \times \\ & \prod_{i=1}^{\text{len}(\vec{a})} \left(\int^{\vec{a}_i} \int^{\Gamma_{i,0}, \vec{\Gamma}_i} \llbracket Q \rrbracket_{\vec{x} \oplus \langle x \rangle}(\Gamma_i \oplus \langle \vec{a}_i \rangle, a_i) \times \llbracket N \rrbracket_{\vec{x}}(\vec{\Gamma}_i, \vec{a}_i) \times SD^n(\Gamma_i, \Gamma_{0,i} \otimes \vec{\Gamma}_i) \right) \\ & \times SD^n(\Delta, \bigotimes_{j=0}^{\text{len}(\vec{a})} \Gamma_j) \end{aligned}$$

The natural isomorphism is then given by the following family of functions

$$\begin{aligned} & \text{sub}_{\Delta, a}^{M,x,N} : \text{Sub}_{\vec{x}}^{M,x,N}(\Delta, a) \cong \llbracket M\{N/x\} \rrbracket_{\vec{x}}(\Delta, a) \\ & \langle \vec{a}, \vec{\Gamma} = \langle \Gamma \rangle_{j=0}^{\text{len}(\vec{a})}, \langle \vec{b}, \langle \Gamma'_{j'} \oplus \langle \vec{a}_{j'} \rangle \rangle_{j'=1}^{\text{len}(\vec{b})}, \langle \varphi, \vec{\psi}, \eta \oplus \langle f \rangle \rangle, \vec{\varphi}, \theta \rangle \mapsto \\ & \langle \vec{b}, \langle \Gamma'_{j',0} \otimes ([f]\vec{\Gamma})_{j'} \rangle_{j'=0}^{\text{len}(\vec{b})}, \text{sub}_{\Gamma'_{0,0} \otimes ([f]\vec{\Gamma})_0, \vec{b} \Rightarrow a}^{P,x,N} \left(\langle \Gamma'_{0,0} \otimes \Gamma_{\vec{a}_1}, \varphi, ([f]\vec{\varphi})_0, 1 \rangle \right), \\ & \langle \text{sub}_{\Gamma'_{0,i'} \otimes ([f]\vec{\Gamma})_{i'}, b_{i'}}^{Q,x,N} \left(\langle \Gamma'_{0,i'} \otimes ([f]\vec{\Gamma})_{i'}, \psi_{i'}, ([f]\vec{\varphi})_{i'}, 1 \rangle \right)_{i'=1}^{\text{len}(\vec{b})}, \\ & (\vec{\tau} \circ (\eta \otimes \text{sm}(f)^*)) \rangle \end{aligned}$$

where

$$\begin{aligned} & \text{sub}^{P,x,N} : \text{Sub}_{\vec{x}}^{P,x,N} \cong P\{N/x\} \\ & \text{sub}^{Q,x,N} : \text{Sub}_{\vec{x}}^{Q,x,N} \cong Q\{N/x\} \end{aligned}$$

are the isos given by the IH. and $\vec{\tau}$ is the permutation

$$\vec{\tau} : \bigotimes_{j'=0}^{\text{len}(\vec{b})} \Gamma'_{j',0} \otimes (\vec{\Gamma}\{\alpha\}) \rightarrow \bigotimes_{j'=0}^{\text{len}(\vec{b})} (\Gamma'_{j',0} \otimes (\vec{\Gamma}\{\alpha\})_{j'}).$$

Theorem 3.4.13. Let $M, N \in \Lambda, \vec{x} \supseteq \text{fv}(M) \cup \text{fv}(N)$ and $M \rightarrow_{\beta} N$. We have a natural isomorphism

$$\llbracket M \rrbracket_{\vec{x}}(\Delta, a) \cong \llbracket N \rrbracket_{\vec{x}}(\Delta, a).$$

Proof. By induction on the reduction step $M \rightarrow_{\beta} N$. We suppose that $\text{len}(\vec{x}) = n$.

3. Intersection Type Distributors – 3.5. Normalization Theorems

Let $M = (\lambda x.P)Q$ and $N = M\{N/x\}$. Then $\llbracket M \rrbracket_{\vec{x}}(\Delta, a) =$

$$\int^{\vec{a}=\langle a_1, \dots, a_k \rangle \in SD} \int^{\Gamma_0, \dots, \Gamma_k \in SD^n} \llbracket P \rrbracket_{\vec{x} \oplus \langle x \rangle}(\Gamma_0 \oplus \langle \vec{a} \rangle, a) \times \prod_{i=1}^k \llbracket Q \rrbracket_{\vec{x}}(\Gamma_i, a_i) \times SD^n(\Delta, \bigotimes_{j=0}^k \Gamma_j).$$

Then we conclude applying the former lemma.

If $M = \lambda x.M'$ then $N = \lambda x.N'$ with $M' \rightarrow_\beta N'$. By definition we have

$$\llbracket \lambda x.M' \rrbracket_{\vec{x}}(\Delta, \vec{a} \Rightarrow a) = \llbracket M' \rrbracket_{\vec{x} \oplus \langle x \rangle}(\Delta \oplus \langle \vec{a} \rangle, a)$$

then we apply the IH and conclude.

If $M = PQ$ then either $N = P'Q$ and $P \rightarrow_\beta P'$ or $N = PQ'$ and $Q \rightarrow_\beta Q'$. Both cases are a direct consequence of the IH. \square

Remark 3.4.14. *Simply applying Remark 3.4.12 and the former theorem, we can extend the denotation to β -reduction steps, by induction as follows:*

— If $M \rightarrow_\beta N = (\lambda x.M)N \rightarrow_\beta M\{N/x\}$ then

$$\llbracket M \rightarrow_\beta N \rrbracket_{\vec{x}} = \text{sub}^{M,x,n}$$

— If $M \rightarrow_\beta N = \lambda y.M' \rightarrow_\beta \lambda y.N'$ then

$$\llbracket M \rightarrow_\beta N \rrbracket_{\vec{x}} = \beta_{M',N'}$$

where $\beta_{M',N'} : \llbracket M' \rrbracket_{\vec{x} \oplus \langle y \rangle} \cong \llbracket N' \rrbracket_{\vec{x} \oplus \langle y \rangle}$ is given by the IH.

— If $M \rightarrow_\beta N = PQ \rightarrow_\beta P'Q$ with $P \rightarrow_\beta P'$ then

$$\llbracket M \rightarrow_\beta N \rrbracket_{\vec{x}} = \int^{\vec{a}=\langle a_1, \dots, a_k \rangle, \Gamma_j} \beta_{\Gamma_0, \vec{a} \Rightarrow -}^{P,P'} \times \prod_{i=1}^k \llbracket Q \rrbracket_{\vec{x}}(\Gamma_i, a_i) \times SD^n(-, \bigotimes_{j=1}^k \Gamma_j)$$

where $\beta^{P,P'} : \llbracket P \rrbracket_{\vec{x}} \cong \llbracket P' \rrbracket_{\vec{x}}$ is given by the IH.

— If $M \rightarrow_\beta N = PQ \rightarrow_\beta PQ'$ with $Q \rightarrow_\beta Q'$ then

$$\llbracket M \rightarrow_\beta N \rrbracket_{\vec{x}} = \int^{\vec{a}=\langle a_1, \dots, a_k \rangle, \Gamma_j} \llbracket P \rrbracket_{\vec{x}}(\Gamma_0, \vec{a} \Rightarrow -) \times \prod_{i=1}^k \beta_{\Gamma_i, a_i}^{Q,Q'} \times SD^n(-, \bigotimes_{j=1}^k \Gamma_j)$$

where $\beta^{Q,Q'} : \llbracket Q \rrbracket_{\vec{x}} \cong \llbracket Q' \rrbracket_{\vec{x}}$ is given by the IH.

3.5. Normalization Theorems

In this section we present a parametric normalization theorem for our systems, adapting the reducibility argument of [Car07; Kri93] to our categorified setting.

3. Intersection Type Distributors – 3.5. Normalization Theorems

The construction of the argument is classical, but there is a technical improvement to be made in order to lift it to a category-theoretic perspective.

We shall consider the three fundamental notions of normalization for λ -calculus, that is head-normalization, β -normalization and strong normalization.

Lemma 3.5.1. *Let $M \in \Lambda$ be a head-normal form. Then*

$$\llbracket M \rrbracket_{\vec{x}} \neq \emptyset_{D_A, SD_A^{\text{len}(\vec{x})}}.$$

Proof. We have that $M = \lambda x_1 \dots \lambda x_m. x Q_1 \dots Q_n$. We prove it for $x Q_1 \dots Q_n$, choosing as list of variables $\vec{y} = \vec{x} \oplus \langle x_1, \dots, x_m \rangle = \langle y_1, \dots, y_k \rangle$ where $k = m + \text{len}(\vec{x})$, the extension being immediate.

Let $b = \langle \rangle \Rightarrow \dots \Rightarrow \langle \rangle \Rightarrow a$. It is enough to take the following type derivation $\pi =$

$$\frac{\frac{1_{\langle \rangle}, \dots, 1_{\langle b \rangle}, \dots, 1_{\langle \rangle}}{y_1 : \langle \rangle, \dots, x : \langle b \rangle, \dots, y_k : 1_{\langle \rangle} \vdash x : \langle \rangle \Rightarrow \dots \Rightarrow \langle \rangle \Rightarrow a}}{y_1 : \langle \rangle, \dots, x : \langle b \rangle, \dots, y_k : \langle \rangle \vdash x Q_1 \dots Q_n : a}}$$

Then $T_D(M)_{\vec{y}}(\langle \langle \rangle, \dots, \langle b \rangle, \dots, \langle \rangle \rangle, a)$ is non-empty for all types $a \in D$. By Theorem 3.4.10 we have $\llbracket M \rrbracket_{\vec{y}}(\langle \langle \rangle, \dots, \langle b \rangle, \dots, \langle \rangle \rangle, a) \neq \emptyset$. □

Corollary 3.5.2. *Let $M \in \Lambda$. If M is head-normalizable then $\llbracket M \rrbracket_{\vec{x}} \neq \emptyset_{D_A, SD_A^{\text{len}(\vec{x})}}$.*

Proof. Corollary of the former lemma and Theorem 3.4.13. □

Definition 3.5.3. *We inductively define two subsets Pos, Neg of $\text{ob}(D)$:*

- $\text{ob}(A) \subset Pos$ and $\text{ob}(A) \subset Neg$;
- if $\vec{a} \in Neg^!$ and $a \in Pos$ then $\vec{a} \Rightarrow a \in Pos$.
- if $\vec{a} \in Pos^!$ such that $\vec{a} \neq \langle \rangle$ and $a \in Neg$ then $(\vec{a}, a) \in Neg$.

We remark that the two considered subset defines two full subcategories of D in the natural way. If $a \in Pos$ (resp. $a \in Neg$) we say that a is *positive* (resp. *negative*). For $\Gamma \in SD^n$ for some $n \in \mathbb{N}$, we say that it is *positive* (resp. *negative*) if all its components are. We set $(\llbracket M \rrbracket_{\vec{x}})_{Pos} : ((S(Neg))^{\text{len}(\vec{x})})^o \times Pos \rightarrow Set$ for the evident restriction of the distributor $\llbracket M \rrbracket_{\vec{x}}$ to the two categories.

We also define another subset of $D^+ \subseteq \text{ob}(D)$ as the smallest set generated by the following grammar:

$$o \in A \mid \langle a_0, \dots, a_k \rangle \Rightarrow a$$

hence if $\vec{a} \Rightarrow a \in D^+$ then $\vec{a} \neq \langle \rangle$. Clearly also D^+ defines a full-subcategory of D in the natural way. We set $(\llbracket M \rrbracket_{\vec{x}})_{D^+}$ for the evident restriction of $\llbracket M \rrbracket_{\vec{x}}$ to D^+ .

Lemma 3.5.4. *Let $M \in \Lambda$ be a β -normal form. Then*

$$(\llbracket M \rrbracket_{\vec{x}})_{Pos} \neq \emptyset_{Pos, (S(Neg))^{\text{len}(\vec{x})}}.$$

3. Intersection Type Distributors – 3.5. Normalization Theorems

Proof. By induction on the size of $M = \lambda x_1 \dots \lambda x_m. x Q_1 \dots Q_n$. We set $\vec{y} = \vec{x} \oplus \langle x_1, \dots, x_m \rangle$. We prove the result for $M' = x Q_1 \dots Q_n$, the extension being immediate. By IH we have that $(\llbracket Q_i \rrbracket_{\vec{y}})_{Pos}(\Gamma_i, a_i) \neq \emptyset$ for some $\Gamma_i \in S(Neg)^{\text{len}(\vec{y})}$, $a_i \in Pos$ for $i \in [n]$. Consider the type $b = \langle a_1 \rangle \Rightarrow \dots \Rightarrow \langle a_n \rangle \Rightarrow o$ for $a \in Pos$. Since $a_1, \dots, a_n \in Pos$ and $o \in Neg$ we have that $b \in Neg$.

Let $\Gamma_0 = \langle \langle \rangle, \dots \langle b \rangle, \dots \langle \rangle \rangle$. Then we have by definition

$$(\llbracket M' \rrbracket_{\vec{y}})_{Pos}(\bigotimes_{j=0}^n \Gamma_j, o) \neq \emptyset.$$

□

Corollary 3.5.5. *Let $M \in \Lambda$. If M is β -normalizable then*

$$(\llbracket M \rrbracket_{\vec{x}})_{Pos} \neq \emptyset_{Pos, (S(Neg))^{\text{len}(\vec{x})}}.$$

Proof. Corollary of the former lemma and Theorem 3.4.13. □

Lemma 3.5.6. *Let $M \in \Lambda$ be a β -normal form and S be an irrelevant resource monad. Then*

$$(\llbracket M \rrbracket_{\vec{x}})_{D^+} \neq \emptyset_{D^+, (S(D^+))^{\text{len}(\vec{x})}}.$$

Proof. By induction on the size of $M = \lambda x_1 \dots \lambda x_m. x Q_1 \dots Q_n$. We set $\vec{y} = \vec{x} \oplus \langle x_1, \dots, x_m \rangle$. We prove the result for $M' = x Q_1 \dots Q_n$, the extension being immediate. If $n = 0$ we use the irrelevancy of the resource monad and we type x as follows

$$\frac{\diamond_{\vec{a}_1}, \dots, 1_a, \dots, \diamond_{\vec{a}_{\text{len}(\vec{y})}}}{y_1 : \vec{a}_1, \dots, x : \langle a \rangle, \dots, y_{\text{len}(\vec{y})} \vdash x : a}$$

and by Theorem 3.4.10 we conclude. If $n \neq 0$ then the proof follows the same pattern as the one of Lemma 3.5.4. □

Corollary 3.5.7. *Let $M \in \Lambda$ and S be an irrelevant resource monad. If M is β -normalizable then*

$$(\llbracket M \rrbracket_{\vec{x}})_{D^+} \neq \emptyset_{D^+, (S(D^+))^{\text{len}(\vec{x})}}.$$

Proof. Corollary of the former lemma and Theorem 3.4.13. □

We are now ready to present our reducibility argument. For a set $\mathcal{X} \subseteq \Lambda$ we say that \mathcal{X} is *saturated* when if $M\{N/x\}N_1 \dots N_n \in \mathcal{X}$ then $((\lambda x.M)N)N_1 \dots N_n \in \mathcal{X}$. Given $\mathcal{X}_1, \mathcal{X}_2 \subseteq \Lambda$, we write $\mathcal{X}_1 \Rightarrow \mathcal{X}_2 = \{M \in \Lambda \mid \text{for all } N \in \mathcal{X}_1, MN \in \mathcal{X}_2\}$.

Given a small category A an *interpretation* is a functor $I : A \rightarrow ((\wp\Lambda)^*, \subseteq)$, where $(\wp\Lambda)^* = \{\mathcal{X} \subseteq \Lambda \mid \mathcal{X} \text{ is saturated}\}$. Given $\delta \in D_A \sqcup SD_A$ we define the *set of realizers* of δ by induction as follows:

$$\llbracket o \rrbracket_I = I(o) \quad \llbracket \langle \rangle \rrbracket_I = \Lambda \quad \llbracket \langle a_0, \dots, a_k \rangle \rrbracket_I = \bigcap_{i=0}^k \llbracket a_i \rrbracket_I$$

3. Intersection Type Distributors – 3.5. Normalization Theorems

$$\llbracket \vec{a} \Rightarrow a \rrbracket_I = \llbracket \vec{a} \rrbracket_I \Rightarrow \llbracket a \rrbracket_I$$

by construction we have that $\llbracket \delta \rrbracket_I$ is saturated.

Lemma 3.5.8. *Let $\delta, \delta' \in D_A \sqcup SD_A$. If $f : \delta \rightarrow \delta'$ then $\llbracket \delta \rrbracket_I \subseteq \llbracket \delta' \rrbracket_I$.*

Proof. By induction on the structure of δ' . If $\delta' = o \in A$ then $\delta = o' \in A$ and the result follows immediately by functoriality of I . If $\delta' = \vec{a} \Rightarrow a$ then $\delta = \vec{a}' \Rightarrow a', \langle \alpha, \vec{g} \rangle : \vec{a}' \rightarrow \vec{a}$ and $g : a \rightarrow a'$ with $f = \langle \alpha, \vec{g} \rangle \Rightarrow g$. By IH, $\llbracket \vec{a}' \rrbracket_I \subseteq \llbracket \vec{a} \rrbracket_I$ and $\llbracket a \rrbracket_I \subseteq \llbracket a' \rrbracket_I$. By a simple inspection of the definitions we can conclude, since if $M \in \llbracket \vec{a} \Rightarrow a \rrbracket_I$ then for all $N \in \llbracket \vec{a}' \rrbracket_I \subseteq \llbracket \vec{a} \rrbracket_I$ one has that $MN \in \llbracket a \rrbracket_I \subseteq \llbracket a' \rrbracket_I$. If $\delta' = \langle \rangle$ then the result is immediate since $\llbracket \langle \rangle \rrbracket_I = \Lambda$. If $\delta' = \langle a_0, \dots, a_k \rangle$ then $\delta = \langle a'_0, \dots, a'_k \rangle$ and $f = \langle \alpha, f_0, \dots, f_k \rangle$ with $f_i : a'_{\alpha(i)} \rightarrow a_i$. By inductive hypothesis then $\llbracket a_{\alpha(i)} \rrbracket_I \subseteq \llbracket a'_i \rrbracket_I$. Since $\bigcap_{j=0}^{k'} \llbracket a'_j \rrbracket_I \subseteq \bigcap_{i=0}^k \llbracket a'_{\alpha(i)} \rrbracket_I$ we can conclude. \square

Lemma 3.5.9. *Let $M, N_1, \dots, N_n \in \Lambda$ and I be an interpretation. If $x_1 : \vec{a}_1, \dots, x_n : \vec{a}_n \vdash M : a$ and $N_i \in \llbracket \vec{a}_i \rrbracket_I$ then $M\{N_1, \dots, N_n/x_1, \dots, x_n\} \in \llbracket a \rrbracket_I$.*

Proof. If $M = x_i$ then $M\{N_1, \dots, N_n/x_1, \dots, x_n\} = N_i \in \llbracket a \rrbracket_I$. We have

$$\frac{f_1 : \vec{a}_1 \rightarrow \langle \rangle, \dots, f_i : \vec{a}_i \rightarrow \langle a \rangle, \dots, f_n : \vec{a}_n \rightarrow \langle \rangle}{x_1 : \vec{a}_1, \dots, x_i : \vec{a}_i, \dots, x_n : \vec{a}_n \vdash x_i : a}$$

By hypothesis $N_i \in \llbracket \vec{a}_i \rrbracket_I$ then, by Lemma 3.5.8 we can conclude. The abstraction case is immediate by IH and the application case is again a corollary of Lemma 3.5.8 and the IH. \square

We define

- $\mathcal{HN} = \{M \in \Lambda \mid \text{The head-reduction of } M \text{ ends}\}$ and $\mathcal{HN}_0 = \{xN_1 \dots N_n \mid N_i \in \Lambda\}$.
- $\mathcal{N} = \{M \in \Lambda \mid \text{the left reduction of } M \text{ ends}\}$ and $\mathcal{N}_0 = \{xN_1 \dots N_n \mid N_i \in \mathcal{N}\}$.
- $\mathcal{SN} = \{M \in \Lambda \mid M \text{ is strongly normalizable}\}$, $\mathcal{SN}_0 = \{xN_1 \dots N_n \mid N_i \in \mathcal{SN}\}$.

Lemma 3.5.10. *The following statements hold.*

1. \mathcal{HN} is saturated.
2. \mathcal{N} is saturated.
3. \mathcal{SN} is saturated.

Proof. — Immediate by definition.

- Immediate by observing that $\lambda x_1. \dots \lambda x_m. P\{Q/x\}Q_1 \dots Q_n$ is exactly the left reduct of $\lambda x_1. \dots \lambda x_m. (\lambda x. P)QQ_1 \dots Q_n$.

3. Intersection Type Distributors – 3.5. Normalization Theorems

— Direct consequence of the definition of strong normalization, it is proven in [Kri93]. □

We set $I_{HN} : A \rightarrow ((\varphi\Lambda)^*, \subseteq)$ to be the functor such that for all $a \in A$, $I_{HN}(a) = \mathcal{HN}$, the action on morphisms being the trivial one. We define in the same way I_N and I_{SN} .

Lemma 3.5.11. *For all $a \in D_A$ we have*

1. $\mathcal{HN}_0 \subseteq \llbracket a \rrbracket_{I_{HN}} \subseteq \mathcal{HN}$.
2. If a is positive (resp. negative) then $\mathcal{N}_0 \subseteq \llbracket a \rrbracket_{I_N}$ (resp. $\llbracket a \rrbracket_{I_N} \subseteq \mathcal{N}$).
3. if $a \in D^+$ then $\mathcal{SN}_0 \subseteq \llbracket a \rrbracket_{I_{SN}} \subseteq \mathcal{SN}$.

Proof. 1. By induction on a . If $a = o$ then the result is immediate by definition.

If $a = \vec{a} \Rightarrow a'$, with $\vec{a} = \langle a_1, \dots, a_k \rangle$. by IH $\mathcal{HN}_0 \subseteq \llbracket a_i \rrbracket_{I_{HN}} \subseteq \mathcal{HN}$ for $0 \leq i \leq k$ and $\mathcal{HN}_0 \subseteq \llbracket a' \rrbracket_{I_{HN}} \subseteq \mathcal{HN}$. By definition, if $M \in \llbracket a \rrbracket_{I_{HN}}$ then for all $N \in \llbracket \vec{a} \rrbracket_{I_{HN}}$ we have $MN \in \llbracket a' \rrbracket_{I_{HN}} \subseteq \mathcal{HN}$. Take $xN_1 \dots N_n \in \mathcal{HN}_0$. Then for $Q \in \llbracket \vec{a} \rrbracket_{I_{HN}}$ we have $xN_1 \dots N_n Q \in \llbracket a' \rrbracket_{I_{HN}}$.

2. By induction on a . If $a = o$ then the result is immediate by definition. If $a = \vec{a} \Rightarrow a'$ then suppose that a is positive. This means that \vec{a} (resp. a') is negative (resp. positive). Then, by IH, $\llbracket \vec{a} \rrbracket_{I_N} \subseteq \mathcal{N}$ and $\mathcal{N}_0 \subseteq \llbracket a' \rrbracket_{I_N}$. Then for $xN_1 \dots N_n \in \mathcal{N}_0$ take $Q \in \llbracket \vec{a} \rrbracket_{I_N}$. By IH, $Q \in \mathcal{N}$ and $xN_1 \dots N_n Q \in \llbracket a' \rrbracket_{I_N}$. Then by definition, $xN_1 \dots N_n \in \llbracket a \rrbracket_{I_N}$.

Suppose that a is negative. Then $\vec{a} \neq \langle \rangle$ is positive and a is negative. By IH $\llbracket a' \rrbracket_{I_N} \subseteq \mathcal{N}$ and $\mathcal{N}_0 \subseteq \llbracket \vec{a} \rrbracket_{I_N}$. Given $M \in \llbracket a \rrbracket_{I_N}$ we have that for all $Q \in \llbracket \vec{a} \rrbracket_{I_N} \subseteq \mathcal{N}$ $MQ \in \llbracket a' \rrbracket_{I_N} \subseteq \mathcal{N}$. Then the left reduction of MQ ends. In particular, this means that the left reduction of M ends. Then $M \in \mathcal{N}$.

3. By induction on a . If $a = o$ then the result is immediate by definition. If $a = \vec{a} \Rightarrow a'$, since $a \in D^+$ we have $\vec{a} = \langle a_0, \dots, a_k \rangle$ and by IH $\mathcal{SN}_0 \subseteq \llbracket a_i \rrbracket_{I_{SN}} \subseteq \mathcal{SN}$ for $0 \leq i \leq k$ and $\mathcal{SN}_0 \subseteq \llbracket a' \rrbracket_{I_{SN}} \subseteq \mathcal{SN}$. By definition, if $M \in \llbracket a \rrbracket_{I_{SN}}$ then for all $N \in \llbracket \vec{a} \rrbracket_{I_{SN}}$ we have $MN \in \llbracket a' \rrbracket_{I_{SN}} \subseteq \mathcal{SN}$. Take $xN_1 \dots N_n \in \mathcal{SN}_0$. Then for $Q \in \llbracket \vec{a} \rrbracket_{I_{SN}}$ we have $xN_1 \dots N_n Q \in \llbracket a' \rrbracket_{I_{SN}}$. □

Lemma 3.5.12. *Let $M \in \Lambda$. The following statements hold.*

1. If M is typable in the system E_A^S then the head-reduction of M ends.
2. Let Γ be a negative context and a a positive type. If $\Gamma \vdash_{E_A^S} M : a$ then the left reduction of M ends.
3. Let $\Gamma \in (SD^+)^n$ for some $n \in \mathbb{N}$ and $a \in D^+$. If $\Gamma \vdash_{E_A^S} M : a$ then M is strongly normalizable.

3. Intersection Type Distributors – 3.5. Normalization Theorems

Proof. 1. By lemma 3.5.11 we have $x_i \in \llbracket a_i \rrbracket_{I_{HN}}$. By Lemma 3.5.9 we have that

$$M\{x_1, \dots, x_n/x_1, \dots, x_n\} = M \in \llbracket a \rrbracket_{I_{HN}}.$$

By Lemma 3.5.11 we have $\llbracket a \rrbracket_{I_{HN}} \subseteq \mathcal{HN}$. Then $M \in \mathcal{HN}$.

2. By lemma 3.5.11 we have $x_i \in \llbracket a_i \rrbracket_{I_N}$. By Lemma 3.5.9 we have that

$$M\{x_1, \dots, x_n/x_1, \dots, x_n\} = M \in \llbracket a \rrbracket_{I_N}.$$

By Lemma 3.5.11 we have $\llbracket a \rrbracket_{I_N} \subseteq \mathcal{HN}$. Then $M \in \mathcal{N}$.

3. By lemma 3.5.11 we have $x_i \in \llbracket a_i \rrbracket_{I_{SN}}$. By Lemma 3.5.9 we have that

$$M\{x_1, \dots, x_n/x_1, \dots, x_n\} = M \in \llbracket a \rrbracket_{I_{SN}}.$$

By Lemma 3.5.11 we have $\llbracket a \rrbracket_{I_{SN}} \subseteq \mathcal{SN}$. Then $M \in \mathcal{SN}$. □

Theorem 3.5.13. *Let $M \in \Lambda$. The following statements are equivalent.*

1. $\llbracket M \rrbracket_{\bar{x}} \neq \emptyset_{D, SD^{\text{len}(\bar{x})}}$.
2. *The head-reduction of M ends.*
3. *M is head-normalizable.*

Proof. (1) \Rightarrow (2) Corollary of Theorems 3.4.10 and Lemma 3.5.12. (2) \Rightarrow (3) immediate by definition. (3) \Rightarrow (1) by Theorem 3.4.2 and Corollary 3.5.2. □

Theorem 3.5.14. *Let $M \in \Lambda$. The following statements are equivalent.*

1. $(\llbracket M \rrbracket_{\bar{x}})_{|Pos} \neq \emptyset_{Pos, (S(Neg))^{\text{len}(\bar{x})}}$.
2. *The left reduction of M ends.*
3. *M is β -normalizable.*

Proof. (1) \Rightarrow (2) Corollary of Theorems 3.4.10 and Lemma 3.5.12. (2) \Rightarrow (3) immediate by definition. (3) \Rightarrow (1) by Theorem 3.4.2 and Corollary 3.5.5. □

Theorem 3.5.15. *Let $M \in \Lambda$ and S be an irrelevant resource monad. The following statements are equivalent.*

1. $(\llbracket M \rrbracket_{\bar{x}})_{|D^+} \neq \emptyset_{D^+, (S(D^+))^{\text{len}(\bar{x})}}$.
2. *M is strongly normalizable.*

Proof. (1) \Rightarrow (2) Corollary of Theorems 3.4.10 and Lemma 3.5.12. (2) \Rightarrow (3) immediate by definition. (3) \Rightarrow (1) by Theorem 3.4.2 and Corollary 3.5.7. □

3.6. Some Worked Out Examples

We present two concrete constructions of the distributor-induced denotational semantics that we introduced in the previous sections.

We chose the examples of the linear resource monad (symmetric monoidal strict completion) and of the cartesian one (cartesian strict completion). Those two examples are particularly relevant since they correspond to the categorification of the two best known intersection type systems: the linear logic induced Gardner-De Carvalho System \mathcal{R} [Gar94; Car07] and the original Coppo-Dezani System $D\Omega$ [CD78]. The first one is non-idempotent, the second one is idempotent. In our setting, the idempotency issue is replaced by an operational one: which operations do we allow on intersections?

3.6.1. Example 1: Linear Resources

We present a refinement of the standard Gardner-De Carvalho non-idempotent intersection type system \mathcal{R} . That system has a categorical counterpart in the linear logic induced relational model for pure λ -calculus [Car07]. The intersection type is given by multisets. In our case, we achieve a non-idempotent and commutative (up to isos) intersection type system applying our construction in the special case where the resource monad S is the 2-monad for symmetric strict monoidal categories. The corresponding intersection type system is system R_A in Figure 3.7.

In the linear case, we can prove the head-normalization theorem in a combinatorial way⁹. We set $\llbracket M \rrbracket_{\vec{x}}^{RA}$ to be the denotation of M in the case where S is the linear resource monad. We define the *size* of a type derivation by induction as follows:

$$\begin{aligned}
 & s\left(\frac{\alpha : a' \rightarrow a}{x_1 : \langle \rangle, \dots, x_i : \langle a' \rangle, \dots, x_n : \langle \rangle \vdash x_i : a}\right) = 0 \\
 & s\left(\frac{\pi}{\Delta, x : \vec{a} \vdash M : a}\right) = s(\pi) \\
 & s\left(\frac{\pi_0 \quad \left(\frac{\pi_i}{\Gamma_i \vdash M : a_i}\right)_{i=1}^k \quad \eta : \Delta \rightarrow \otimes_{j=0}^k \Gamma_j}{\Gamma_0 \vdash M : \vec{a} \Rightarrow a \quad \Delta \vdash MN : a}\right) = s(\pi_0) + \sum_{i \in [k]} s(\pi_i) + 1
 \end{aligned}$$

By an easy inspection of the definitions, we have that the size is stable under actions and under congruence: if $\pi, \pi' \in R_A$ and $\pi \sim \pi'$ then $s(\pi) \sim s(\pi')$.

Let $\rho_{\Delta, a} : \llbracket M \rrbracket_{\vec{x}}^{RA}(\Delta, a) \cong T_D(M)_{\vec{x}}^{RA}(\Delta, a)$ be the isomorphism given by Theorem 3.4.2. Then for $\alpha \in \llbracket M \rrbracket_{\vec{x}}(\Delta, a)$ we set $s(\alpha) = s(\rho_{\Delta, a}(\alpha))$. Given $\vec{\alpha} = \langle \alpha_1, \dots, \alpha_k \rangle$ with $\alpha_i \in \llbracket M \rrbracket_{\vec{x}}$, we set $s(\vec{\alpha}) = \sum_{i=1}^k s(\alpha_i)$.

Lemma 3.6.1. *Let M, N be two λ -terms, $\vec{x} \supset \text{fv}(M) \cup \text{fv}(N)$ with $x \notin \vec{x}$ and*

$$\text{sub}_{\Delta, a}^{M, x, N} : \text{Sub}_{\vec{x}}^{M, x, N}(\Delta, a) \cong \llbracket M\{N/x\} \rrbracket_{\vec{x}}^{RA}(\Delta, a)$$

9. Actually, the same argument can be used also in the semicartesian case.

3. Intersection Type Distributors – 3.6. Some Worked Out Examples

be the natural isomorphism defined in Remark 3.4.12. For all $\alpha = \langle \pi, \widetilde{\vec{\psi}}, \eta \rangle \in \text{Sub}_{\vec{x}}^{M,x,N}(\Delta, a)$, we have

$$s\left(\text{sub}_{\Delta,a}^{M,x,N}(\alpha)\right) = s(\pi) + s(\vec{\psi}).$$

Proof. By induction on the structure of $M\{N/x\}$, exploiting Remark 3.4.12. If $M = x$ then $M\{N/x\} = N$ and

$$\text{Sub}_{\vec{x}}^{M,x,N}(\Delta, a) \cong \int^{a' \in D} \int^{\Gamma \in \text{SD}^n} \text{SD}^n(\langle \langle \rangle, \dots, \langle \rangle, \langle a' \rangle, \langle a \rangle \rangle \times \llbracket \vec{x} \rrbracket_N(\Gamma, a') \times \text{SD}^n(\Delta, \Gamma)).$$

By Remark 3.4.12 the natural isomorphism $\text{sub}_{\Delta,a}^{M,x,N}$ is given by the following function:

$$\langle a', \Gamma, \langle \widetilde{1_{\langle \rangle}}, \langle f \rangle \rangle, \langle \rho \rangle, \eta \mapsto [f]\rho\{\eta\}.$$

Since the size is stable under actions, we can conclude, since $s(\langle \rho \rangle) + s(\langle \widetilde{1_{\langle \rangle}}, \langle f \rangle \rangle) = s(\rho)$.

If $M = y$ with $y \neq x$ the result is immediate by definition. If $M = \lambda y.M'$ the result is immediate by IH. If $M = PQ$ we have

$$\begin{aligned} \text{sub}_{\Delta,a}^{M,x,N} : \text{Sub}_{\vec{x}}^{M,x,N}(\Delta, a) &\cong \llbracket M[N/x] \rrbracket_{\vec{x}}(\Delta, a) \\ \langle \vec{a}, \vec{\Gamma} = \langle \Gamma \rangle_{j=0}^{\text{len}(\vec{a})}, \langle \vec{b}, \langle \Gamma'_{j'} \oplus \langle \vec{a}_{j'} \rangle \rangle_{j'=0}^{\text{len}(\vec{b})}, \langle \varphi, \vec{\psi}, \eta \oplus \langle f \rangle \rangle \rangle, \vec{\varphi}, \theta &\mapsto \\ \langle \vec{b}, \langle \Gamma'_{j',0} \otimes ([f]\vec{\Gamma})_{j'} \rangle_{j'=0}^{\text{len}(\vec{b})}, \text{sub}_{\Gamma'_{0,0} \otimes ([f]\vec{\Gamma})_0, \vec{b} \Rightarrow a}^{P,x,N} \left(\langle \Gamma'_{0,0} \otimes \Gamma_{\vec{a}_1}, \varphi, ([f]\vec{\varphi})_0, 1 \rangle \right), & \\ \langle \text{sub}_{\Gamma'_{0,i'} \otimes ([f]\vec{\Gamma})_{i'}, b_{i'}}^{Q,x,N} \left(\Gamma'_{0,i'} \otimes ([f]\vec{\Gamma})_{i'}, \psi_{i'}, ([f]\vec{\varphi})_{i'}, 1 \right) \rangle_{i'=1}^{\text{len}(\vec{b})}, & \\ (\tau \circ (\eta \otimes \text{sm}(f)^*)) & \end{aligned}$$

where

$$\text{sub}^{P,x,N} : \text{Sub}_{\vec{x}}^{P,x,N} \cong P\{N/x\}$$

and

$$\text{sub}^{Q,x,N} : \text{Sub}_{\vec{x}}^{Q,x,N} \cong Q\{N/x\}$$

are the isos given by the IH and $\vec{\tau}$ is the appropriate permutation. By IH, we have that

$$s\left(\text{sub}_{\Gamma'_{0,0} \otimes ([f]\vec{\Gamma})_0, \vec{b} \Rightarrow a}^{P,x,N} \left(\langle \Gamma'_{0,0} \otimes \Gamma_{\vec{a}_1}, \varphi, ([f]\vec{\varphi})_0, 1 \rangle \right)\right) = s(\varphi) + s((([f]\vec{\varphi})_0))$$

and

$$\text{sub}_{\Gamma'_{0,i'} \otimes ([f]\vec{\Gamma})_{i'}, b_{i'}}^{Q,x,N} \left(\Gamma'_{0,i'} \otimes ([f]\vec{\Gamma})_{i'}, \psi_{i'}, ([f]\vec{\varphi})_{i'}, 1 \right) = s(\psi_{i'}) + s((([f]\vec{\varphi})_{i'}))$$

3. Intersection Type Distributors – 3.6. Some Worked Out Examples

now, by commutativity of the sum we have $s([f]\vec{\varphi}) = \sum_{j'=0}^{\text{len}(\vec{b})} s([f]\vec{\varphi}_{j'})$. Since the size is stable under left action, we get $s([f]\vec{\varphi}) = s(\vec{\varphi})$. Let

$$\begin{aligned} \beta = & \langle \vec{b}, \langle \Gamma'_{j',0} \otimes ([f]\vec{\Gamma})_{j'} \rangle_{j'=0}^{\text{len}(\vec{b})}, \text{sub}_{\Gamma'_{0,0} \otimes ([f]\vec{\Gamma})_0, \vec{b} \Rightarrow a}^{P,x,N} \left(\langle \Gamma'_{0,0} \otimes \Gamma_{\vec{a}_1}, \varphi, ([f]\vec{\varphi})_0, 1 \rangle \right), \\ & \langle \text{sub}_{\Gamma'_{0,i'} \otimes ([f]\vec{\Gamma})_{i'}, b_{i'}}^{Q,x,N} \left(\langle \Gamma'_{0,i'} \otimes ([f]\vec{\Gamma})_{i'}, \psi_{i'}, ([f]\vec{\varphi})_{i'}, 1 \rangle \right)_{i'=1}^{\text{len}(\vec{b})}, \\ & (\tau \circ (\eta \otimes \text{sm}(f)^*)) \rangle. \end{aligned}$$

By definition of the size of a derivation of an application we get

$$\begin{aligned} s(\beta) = & (s(\varphi) + s([f]\vec{\varphi}_0)) + \left(\sum_{i'=1}^{\text{len}(\vec{b})} s(\psi_{i'}) + s([f]\vec{\varphi}_{i'}) \right) + 1 = \\ & (s(\varphi) + s(\langle \psi_1, \dots, \psi_{\text{len}(\vec{b})} \rangle) + 1) + s(\vec{\varphi}). \end{aligned}$$

□

We denote the head-reduct of M as $H(M)$.

Theorem 3.6.2. *Let $M \in \Lambda$. We have a natural isomorphism*

$$\varphi_{\Delta,a} : \llbracket M \rrbracket_{\vec{x}}^{RA}(\Delta, a) \cong \llbracket H(M) \rrbracket_{\vec{x}}^{RA}(\Delta, a)$$

such that for $\alpha \in \llbracket M \rrbracket_{\vec{x}}^{RA}(\Delta, a)$, $s(\varphi_{\Delta,a}(\alpha)) \preceq s(\alpha)$.

Proof. Direct corollary of the former lemma. □

Theorem 3.6.3. *Let $M \in \Lambda$. If $\llbracket M \rrbracket_{\vec{x}}^{RA} \neq \emptyset_{D,SD^{\text{len}(\vec{x})}}$ the head reduction of M ends.*

Proof. We have that, for $\varphi : \llbracket M \rrbracket_{\vec{x}}^{RA} \cong \llbracket H(M) \rrbracket_{\vec{x}}^{RA}$. If $\llbracket M \rrbracket_{\vec{x}}^{RA} \neq \emptyset_{D,SD^{\text{len}(\vec{x})}}$ then $\llbracket H(M) \rrbracket_{\vec{x}}^{RA} \neq \emptyset_{D,SD^{\text{len}(\vec{x})}}$. We consider $\alpha \in \llbracket M \rrbracket_{\vec{x}}^{RA}(\Delta, a)$ for some $\langle \Delta, a \rangle \in SD^{\text{len}(\vec{x})} \times D$. Then, by the former theorem, $s(\varphi_{\Delta,a}) < s(\alpha)$. Then we can apply the IH and conclude. □

Example 3.6.4. *We provide some example of type derivations in system R_A , giving also some intuition for what concerns the congruence on type derivations.*

1. *We fix two types a, a' and a morphism $f : a' \rightarrow a$ between them. One can think of them as, e.g. $a = \langle *, \langle * \rangle \Rightarrow * \rangle$ and $a' = \langle \langle * \rangle \Rightarrow *, * \rangle$ with $f = \sigma \Rightarrow 1$ being the obvious permutation.*

Let us type the term xx :

$$\frac{\frac{1 : \langle a \rangle \Rightarrow a \rightarrow \langle a \rangle \Rightarrow a}{x : \langle \langle a \rangle \Rightarrow a \rangle \vdash x : \langle a \rangle \Rightarrow a} \quad \frac{1 : a \rightarrow a}{x : \langle a \rangle \vdash x : a}}{x : \langle a', \langle a \rangle \Rightarrow a \rangle \vdash xx : a} \quad \langle (1, 2), f, 1 \rangle$$

3. Intersection Type Distributors – 3.6. Some Worked Out Examples

System R_A :

$$\frac{\frac{\alpha : a' \rightarrow a}{x_1 : \langle \rangle, \dots, x_i : \langle a' \rangle, \dots, x_n : \langle \rangle \vdash x_i : a} \quad \frac{\Delta, x : \vec{a} \vdash M : a}{\Delta \vdash \lambda x.M : \vec{a} \Rightarrow a}}{\Gamma_0 \vdash M : \vec{a} \Rightarrow a \quad (\Gamma_i \vdash N : a_i)_{i \in [k]} \quad \eta : \Delta \rightarrow \bigotimes_{j=0}^k \Gamma_j} \quad \Delta \vdash MN : a$$

System C_A :

$$\frac{\frac{f_1 : \vec{a}_1 \rightarrow \langle \rangle, \dots, f_i : \vec{a} \rightarrow \langle a \rangle, \dots, f_n : \vec{a}_n \rightarrow \langle \rangle}{x_1 : \vec{a}_1, \dots, x_i : \vec{a}, \dots, x_n : \vec{a}_n \vdash x_i : a} \quad \frac{\Delta, x : \vec{a} \vdash M : a}{\Delta \vdash \lambda x.M : \vec{a} \Rightarrow a}}{\Delta \vdash M : \vec{a} \Rightarrow a \quad (\Delta \vdash N : a_i)_{i \in [k]}} \quad \Delta \vdash MN : a$$

Where $\vec{a} = \langle a_1, \dots, a_k \rangle$.

Figure 3.7. – Intersection type systems R_A and C_A .

We call π the previous derivation. Now consider the following type derivation π'

$$\frac{\frac{1 : (\langle a \rangle \Rightarrow a) \rightarrow (\langle a \rangle \Rightarrow a)}{x : \langle \langle a \rangle \Rightarrow a \rangle \vdash x : \langle a \rangle \Rightarrow a} \quad \frac{f : a' \rightarrow a}{x : \langle a' \rangle \vdash x : a} \quad \langle (1, 2), 1, 1 \rangle}{x : \langle a', \langle a \rangle \Rightarrow a \rangle \vdash xx : a}$$

We produced π' by anticipating the morphism induced by α in π . The equivalence induced by the semantics (3.3) says that we have the right to consider π and π' as the same type derivation.

2. Let us type the λ -term $(\lambda x.x)z$:

$$\pi = \frac{\frac{f : a' \rightarrow a}{x : \langle a' \rangle \vdash x : a} \quad \frac{1 : a' \rightarrow a'}{z : \langle a' \rangle \vdash z : a'} \quad 1}{z : \langle a' \rangle \vdash (\lambda x.x)z : a}$$

Now consider the following derivation

$$\pi' = \frac{\frac{1 : a \rightarrow a}{x : \langle a \rangle \vdash x : a} \quad \frac{f : a' \rightarrow a}{z : \langle a' \rangle \vdash z : a} \quad 1}{z : \langle a' \rangle \vdash (\lambda x.x)z : a}$$

3. Intersection Type Distributors – 3.6. Some Worked Out Examples

We produced π' by swapping the role of the morphism f in π . The equivalence induced by the semantics (3.2) says that we have the right to consider π and π' as the same type derivation.

Example 3.6.5. We provide a simple example of reduction of type derivations to ease the understanding of the congruence's role in establishing the natural isomorphisms. Consider $M = (\lambda x.x)y$. We type it with the following type derivations:

$$\pi_1 = \frac{\frac{h \circ f : a \rightarrow b}{x : \langle a \rangle \vdash x : b} \quad \frac{g : c \rightarrow a}{y : \langle c \rangle \vdash y : a} \quad 1}{\vdash \lambda x.x : \langle a \rangle \Rightarrow b} \quad \frac{\frac{h \circ f' : d \rightarrow b}{x : \langle a \rangle \vdash x : b} \quad \frac{g' : c \rightarrow d}{y : \langle c \rangle \vdash y : d} \quad 1}{\vdash \lambda x.x : \langle d \rangle \Rightarrow b} \quad 1}{y : \langle c \rangle \vdash (\lambda x.x)y : b} \quad 1$$

suppose that $f \circ g = f' \circ g'$ and $h : b \rightarrow b, f : a \rightarrow b, f' : d \rightarrow b$. We have that $\pi_1 \sim \pi_2$. Indeed, by the first rule of Figure 3.2:

$$\pi_1 \sim \frac{\frac{h : b \rightarrow b}{x : \langle b \rangle \vdash x : b} \quad \frac{f \circ g : c \rightarrow b}{y : \langle c \rangle \vdash y : b} \quad 1}{\vdash \lambda x.x : \langle b \rangle \Rightarrow b} \quad 1}{y : \langle c \rangle \vdash (\lambda x.x)y : b}$$

$$\pi_2 \sim \frac{\frac{h : b \rightarrow b}{x : \langle b \rangle \vdash x : b} \quad \frac{f' \circ g' : c \rightarrow b}{y : \langle c \rangle \vdash y : b} \quad 1}{\vdash \lambda x.x : \langle b \rangle \Rightarrow b} \quad 1}{y : \langle c \rangle \vdash (\lambda x.x)y : b}$$

and by the hypothesis that $f \circ g = f' \circ g'$ we can conclude by transitivity. In particular, this means that the quotient identify all couple of morphisms leading to the same composition. Now, we have that $M \rightarrow y$. Consider the following type derivation of y :

$$\pi_3 = \frac{h \circ (f \circ g) : c \rightarrow b}{y : \langle c \rangle \vdash y : b}$$

By an easy inspection of the definitions we have that for $\varphi_{\langle c \rangle, b} : \llbracket M \rrbracket_{\langle y \rangle}(\langle c \rangle, b) \cong \llbracket y \rrbracket_{\langle y \rangle}(\langle c \rangle, b)$, $\varphi_{\langle c \rangle, b}(\tilde{\pi}_1) = \pi_3$, where we keep implicit the isomorphism given by Theorem 3.4. There is then a nice correspondence between substitution on the term side and composition on the morphism side, that validates the basic intuition of categorical semantics¹⁰.

10. The natural isomorphism $\varphi_{\langle c \rangle, b} : \llbracket M \rrbracket_{\langle y \rangle}(\langle c \rangle, b) \cong \llbracket y \rrbracket_{\langle y \rangle}(\langle c \rangle, b)$ is a particular instance of the Yoneda Lemma for coends (Theorem 2.4.5).

3.6.2. Example 2: Cartesian Resources

In this section we focus on the type theoretic semantics induced by the cartesian resource monad. In this framework, a resource can be copied and deleted at will.

When SA is cartesian, the Day convolution on PSA is isomorphic to the cartesian product. Hence, we have the following natural isomorphism (see Proposition 2.5.3)

$$G \circ F(\vec{a}, c) \cong \int^{(b_1, \dots, b_k) \in SB} G(\vec{b}, c) \times \prod_{i \in [n]} F(\vec{a}, b_i).$$

By straightforward coend manipulations¹¹, we derive the type system C_A described in Figure 3.7. Actions on type derivations are defined in the straightforward way. The equivalence on type derivation in this case is generated only by (3.2), since now the coend on contexts disappeared. It is worth noting that the cartesian category SD_A admits all the basic axioms imposed on the preorder over idempotent intersection types [ABD06]. This means that our construction generalizes the standard subtyping relation, as expected. However, the two conditions

$$\pi_{i,2} : \vec{a}_1 \oplus \vec{a}_2 \rightarrow \vec{a}_i \quad c_{\vec{a}} : \vec{a} \rightarrow \vec{a} \oplus \vec{a}$$

do not determine an idempotency $\vec{a} \oplus \vec{a} \cong \vec{a}$. In our categorified setting, idempotency is replaced by the possibility to perform two operations on resources: copying and deleting.

Example 3.6.6. *We provide some example of type derivations in system C_A , giving also some intuition for what concerns the congruence on type derivations.*

1. *Let us type the term $M = (\lambda x.(xx)x)$. Let $b = \langle a \rangle \Rightarrow \langle a \rangle \Rightarrow a$. Consider the following type derivation π :*

$$\frac{\frac{\frac{\pi_1 : \langle b, a \rangle \rightarrow \langle b \rangle}{x : \langle b, a \rangle \vdash x : \langle a \rangle \Rightarrow \langle a \rangle \Rightarrow a} \quad \frac{\pi_2 : \langle b, a \rangle \rightarrow \langle a \rangle}{x : \langle b, a \rangle \vdash x : a}}{x : \langle b, a \rangle \vdash xx : \langle a \rangle \Rightarrow a} \quad \frac{\pi_2 : \langle b, a \rangle \rightarrow \langle a \rangle}{x : \langle b, a \rangle \vdash x : a}}{x : \langle b, a \rangle \vdash (xx)x : a} \quad \frac{}{\vdash \lambda x.(xx)x : \langle b, a \rangle \Rightarrow a}$$

Now consider the following type derivation ρ :

$$\frac{\begin{array}{c} \pi \\ \vdots \\ \vdash \lambda x.(xx)x : \langle b, a \rangle \Rightarrow a \quad \vdash N : b \quad \vdash N : a \end{array}}{\vdash (\lambda x.(xx)x)N : a}$$

and π' :

11. Simply observing that the tensor product over SD^n is cartesian.

3. Intersection Type Distributors – 3.6. Some Worked Out Examples

$$\begin{array}{c}
 \frac{\pi_1 : \langle b, a, a \rangle \rightarrow \langle b \rangle}{x : \langle b, a, a \rangle \vdash x : \langle a \rangle \Rightarrow \langle a \rangle \Rightarrow a} \quad \frac{\pi_2 : \langle b, a, a \rangle \rightarrow \langle a \rangle}{x : \langle b, a, a \rangle \vdash x : a} \quad \frac{\pi_3 : \langle b, a, a \rangle \rightarrow \langle a \rangle}{x : \langle b, a, a \rangle \vdash x : a} \\
 \frac{x : \langle b, a, a \rangle \vdash xx : \langle a \rangle \Rightarrow a}{x : \langle b, a, a \rangle \vdash (xx)x : a} \\
 \frac{x : \langle b, a, a \rangle \vdash (xx)x : a}{\vdash \lambda x.(xx)x : \langle b, a, a \rangle \Rightarrow a}
 \end{array}$$

We have that $\pi = [c^* \Rightarrow 1]\pi'$ where $c^* = 1_{\langle b \rangle} \oplus c_{\langle a \rangle}$. If we consider then the following derivation ρ' :

$$\begin{array}{c}
 \pi' \\
 \vdots \\
 \frac{\vdash \lambda x.(xx)x : \langle b, a, a \rangle \Rightarrow a \quad \vdash N : b \quad \vdash N : a \quad \vdash N : a}{\vdash (\lambda x.(xx)x)N : a}
 \end{array}$$

We have that $\rho \sim \rho'$ by the first rule of congruence (3.2).

4. Subtyping-Aware Polyadic Calculus and Rigid Expansion of Ordinary λ -Terms

4.1. Introduction

In [TAO17] the *rigid Taylor Expansion* of λ -terms is introduced and studied in a simply typed η -long framework with fixpoints. The rigid Taylor expansion is a bicategorical denotational semantics for λ -terms in the particular setting of generalized species of structures. A term is interpreted as a distributor that takes as inputs a type context Δ and a type a and returns sets of well-typed linear approximants¹, roughly

$$\llbracket M \rrbracket(\Delta, a) = \{\tilde{p} \mid p \triangleleft M \text{ and } \Delta \vdash p : a\}$$

where the equivalence relation on approximants depends on the coend formula quotient and $p \triangleleft M$ means that p approximates M .

In this chapter we extend that notion to the pure λ -calculus and we establish a formal relationship between rigid Taylor Expansion and Intersection Type Distributors. In particular, we will not limit ourself to the linear case, but we will carry out a parametric construction over resource monads, as we have already done for intersection type distributors.

The main goal of the chapter is to find an appropriate calculus of approximants for ordinary λ -terms that could be seen as the *term language* for the intersection type systems introduced in the previous chapter. We will find out that the right notion of approximant in this context is an hybrid between type derivations and a standard resource calculus (Section 4.2.4). These approximants are connected to λ -terms *via* a *rigid expansion*, that can be seen as the structure which "corresponds" in our setting to the ordinary Taylor expansion.

We will finally obtain (Definition 4.2.3) a reduction relation on approximants, that makes explicit the *dynamical content* of the natural isomorphisms defined in Remark 3.4.14. Hence, modulo the isomorphism between intersection type distributors and rigid expansion (Theorem 4.3.4), we have that the 2-cell

$$\beta^{M,N} : \llbracket M \rrbracket_{\vec{x}} \cong \llbracket N \rrbracket_{\vec{x}}$$

1. The typing is given in a standard linear intersection type system.

associated to a β -reduction step $M \rightarrow_\beta N$ is given by performing a corresponding reduction step on approximants.

Throughout the chapter we shall make an extensive use of the notations and operations on lists introduced in Section 2.6.2.

Structure of the Chapter First we introduce the *subtyping-aware* polyadic calculus, that has to be understood as the term language for our parametric intersection type system E_A^S , defined in Chapter 3 (Figure 3.3). We present a type assignment for these terms and we define right and left morphisms actions. We introduce a congruence on polyadic type derivations which mimics the one already introduced in Chapter 3 (3.3). We prove that a polyadic type derivation is univocally determined by its *point*, *i.e.*, by the morphism and term associated to it. For this reason, we can restrict our consideration to points. We define an appropriate reduction relation on points. We then introduce the *rigid expansion* of ordinary λ -terms, that is a distributor that, given as input a context $\Delta \in SD^n$ and a type $a \in D$, returns sets of well-typed approximants of ordinary λ -terms, *i.e.*, subtyping-aware polyadic terms together with morphisms information. We prove that the rigid expansion of a λ -term is naturally isomorphic to its intersection type distributor. We study this structure under reduction, extending basic results already known about the Taylor expansion (see Chapter 1) to this finer setting. Finally, we study a particular case of our construction, that is the linear one. We present a relationship between rigid expansion and ordinary Taylor expansion in this setting.

Discussion of Related Work In [TAO17] the rigid Taylor Expansion is introduced *via* a specific class of resource terms, the *polyadic terms* [MPV18]. However, since we deal with greater generality, we need to define a new polyadic calculus, what we call *subtyping-aware polyadic terms*. In particular, we shall see that, while in the η -long fragment one can reason simply considering standard polyadic terms, this does not work any more already for the whole simply typed λ -calculus². Before starting with the technicalities, we want to briefly and informally explain why standard polyadic calculi are not enough in our setting.

Consider *linear polyadic terms*, generated by the following grammar:

$$p ::= x \mid \lambda \langle x_1, \dots, x_k \rangle . p \mid p\vec{q}$$

2. It is easy to see that the corresponding distributor is not well-defined if we relax the hypothesis about η -longness. Let us denote as $\mathcal{T}_{\text{rig}}(M)_{\text{std}}^{\Gamma, A}(\Delta, a)$ the distributor defined in [TAO17][p.8] associated to $\Gamma \vdash M : A$, without supposing M being η -long. Just consider $\mathcal{T}_{\text{rig}}(x)_{\text{std}}^{\Gamma, A}(\langle y : \langle \langle \star \rangle \Rightarrow \star, \langle \rangle \Rightarrow \star \rangle \Rightarrow \star, \langle \langle \star \rangle \Rightarrow \star, \langle \rangle \Rightarrow \star \rangle \Rightarrow \star) = \{y\}$ where $M = x, \Gamma = x : (o \Rightarrow o) \Rightarrow o$ and $A = (o \Rightarrow o) \Rightarrow o$. Then take $\sigma \Rightarrow 1 : \langle \langle \star \rangle \Rightarrow \star, \langle \rangle \Rightarrow \star \rangle \Rightarrow \star \rightarrow \langle \langle \rangle \Rightarrow \star, \langle \star \rangle \Rightarrow \star \rangle \Rightarrow \star$, with σ being the symmetry $\sigma : \langle \langle \rangle \Rightarrow \star, \langle \star \rangle \Rightarrow \star \rangle \rightarrow \langle \langle \star \rangle \Rightarrow \star, \langle \rangle \Rightarrow \star \rangle$. By definition $\mathcal{T}_{\text{rig}}(x)_{\text{std}}^{\Gamma, A}(\langle y : \langle \langle \star \rangle \Rightarrow \star, \langle \rangle \Rightarrow \star \rangle \Rightarrow \star, \langle \langle \rangle \Rightarrow \star, \langle \star \rangle \Rightarrow \star \rangle \Rightarrow \star) = \emptyset$ and then $\mathcal{T}_{\text{rig}}(x)_{\text{std}}^{\Gamma, A}(-, -)$ cannot be a functor, since there is no function from a set to the empty set.

4. Subtyping-Aware Polyadic Calculus and Rigid Expansion of Ordinary λ -Terms –
4.1. Introduction

$$\vec{q} = \langle \rangle \mid \langle p \rangle \oplus \vec{q} \mid \perp$$

Where each variable in a term must appear exactly once. The operational semantics is given contextually by the following base case:

$$(\lambda \vec{x}.p)\vec{q} \rightarrow \begin{cases} p\{\vec{q}/\vec{x}\} & \text{if } \text{len}(\vec{q}) = \text{len}(\vec{x}) \\ \perp & \text{otherwise.} \end{cases}$$

Where the substitution operation is linear. We consider terms up to renaming of bound variables and up to linearity wrt \perp , *i.e.*, $\perp = \lambda \vec{x}.\perp = \perp \vec{q}$ etc.

Since we want to link a calculus of approximants to intersection type distributors, the first thing to check is that the calculus satisfies subject reduction and expansion for system E_A^S . We give the following naive type assignment:

$$\frac{\frac{\frac{f : a' \rightarrow a}{\langle \rangle : \langle \rangle, \dots, \langle x \rangle : \langle a' \rangle, \dots, \langle \rangle : \langle \rangle \vdash x : a} \quad \zeta \oplus \langle \vec{x} \rangle : \Delta \oplus \langle \vec{a} \rangle \vdash p : a}{\zeta : \Delta \vdash \lambda \vec{x}.p : \vec{a} \Rightarrow a}}{\zeta_0 : \Gamma_0 \vdash p : \langle a_1, \dots, a_k \rangle \Rightarrow a \quad (\zeta_i : \Gamma_i \vdash q_i)_{i \in [k]} \quad \eta : \Delta \rightarrow \bigotimes_{j=0}^k \Gamma_j}}{\left(\bigotimes_{j=0}^k \zeta_j\right)\{\eta\} : \Delta \vdash p\langle q_1, \dots, q_k \rangle : a}$$

Where in the application case the right action $(\bigotimes_{j=0}^k \zeta_j)\{\eta\}$ only means that we have to rearrange the positions of free variables in context along the permutation induced by the morphism η . This is reasonable and necessary, since the morphism η can in general rearrange the position of types. We do not formally define this action for the moment, since the problem with the former type assignment is even more fundamental. It is easy to see that \perp is not typable in the former system.

Example 4.1.1. *We present a counter-example for the subject reduction of the former system. Take the polyadic term $p = (\lambda \langle x \rangle.x \langle \lambda \langle \rangle.y_1 \langle \rangle, \lambda \langle f \rangle.y_2 \langle f \rangle \rangle) \langle \lambda \langle z_1, z_2 \rangle.z_1 \langle z_2 \rangle \rangle$. This term clearly reduces to \perp , but it is typable in the former type system. Let $\pi =$*

$$\frac{\frac{\frac{g : b' \rightarrow b}{\langle x \rangle : \langle b' \rangle, \langle \rangle \vdash x : b} \quad \langle \rangle : \langle \rangle, \langle y_1 \rangle : \langle \langle \rangle \Rightarrow a \rangle \vdash \lambda \langle \rangle.y_1 \langle \rangle : \langle \rangle \Rightarrow a \quad \langle \rangle : \langle \rangle, \langle y_1 \rangle : \langle \langle c \rangle \Rightarrow a \rangle \vdash \lambda \langle f \rangle.y_1 \langle f \rangle : \langle c \rangle \Rightarrow a}{\langle x \rangle : \langle b' \rangle, \langle y_1, y_2 \rangle : \langle \langle \rangle \Rightarrow a, \langle c \rangle \Rightarrow a \rangle \vdash x \langle \lambda \langle \rangle.y_1 \langle \rangle, \lambda \langle f \rangle.y_2 \langle f \rangle \rangle : a}}{\langle y_1, y_2 \rangle : \langle \langle \rangle \Rightarrow a, \langle c \rangle \Rightarrow a \rangle \vdash \lambda \langle x \rangle.x \langle \lambda \langle \rangle.y_1 \langle \rangle, \lambda \langle f \rangle.y_2 \langle f \rangle \rangle : \langle b' \rangle \Rightarrow a}}$$

Where $c = \langle \rangle \Rightarrow a$ and $b' = \langle \langle c \rangle \Rightarrow a, \langle \rangle \Rightarrow a \rangle \Rightarrow a$ and $b = \langle \langle \rangle \Rightarrow a, \langle c \rangle \Rightarrow a \rangle \Rightarrow a$ the morphism g being of the shape $\langle \sigma, 1_{\langle \rangle \Rightarrow a}, 1_{\langle a \rangle \Rightarrow a} \rangle \Rightarrow 1$ with σ being the obvious permutation. Consider $\rho =$

$$\frac{\frac{\langle z_1 \rangle : \langle \langle c \rangle \Rightarrow a \rangle \vdash z_1 : \langle c \rangle \Rightarrow a \quad \langle z_2 \rangle : \langle c \rangle \vdash z_2 : c}{\langle z_1, z_2 \rangle : \langle \langle c \rangle \Rightarrow a, c \rangle \vdash z_1 \langle z_2 \rangle : a}}{\vdash \lambda \langle z_1, z_2 \rangle.z_1 \langle z_2 \rangle : \langle \langle c \rangle \Rightarrow a, c \rangle \Rightarrow a}}$$

4. Subtyping-Aware Polyadic Calculus and Rigid Expansion of Ordinary λ -Terms –
 4.2. The Calculus of Subtyping-Aware Polyadic Terms

Now take $\pi' =$

$$\frac{\begin{array}{c} \pi \\ \vdots \end{array} \quad \begin{array}{c} \rho \\ \vdots \end{array} \quad \langle y_1, y_2 \rangle : \langle \langle \rangle \Rightarrow a, \langle a \rangle \Rightarrow a \rangle \vdash \lambda \langle x \rangle . x \langle \lambda \langle \rangle . y_1 \langle \rangle , \lambda \langle f \rangle . y_2 \langle f \rangle \rangle : \langle b' \rangle \Rightarrow a \quad \vdash \lambda \langle z_1, z_2 \rangle . z_1 \langle z_2 \rangle : b'}{\langle y_1, y_2 \rangle : \langle \langle \rangle \Rightarrow a, \langle a \rangle \Rightarrow a \rangle \vdash p : a}$$

Then $p \rightarrow \perp$ and p is typable, while \perp it is not. The problem relies completely in the variable rule: the subtyping feature of the system is not detected by the syntax of the standard polyadic calculus. If we want to find an appropriate term language for our system, whose elements are also approximants of ordinary λ -terms, we need to take seriously the qualitative information produced by the subtyping.

4.2. The Calculus of Subtyping-Aware Polyadic Terms

We fix a countable set of variables \mathcal{V} . Given a set A we denote the free non-commutative monoid on A as $A^!$. Then $A^!$ is just the set of lists over A . We call *variable lists* the elements of $\mathcal{V}^!$.

We introduce the syntax of the subtyping-aware polyadic calculus by the following grammar:

$$p \in \Xi_A^S ::= x \mid \lambda \langle x_1, \dots, x_n \rangle : f.p \mid p\vec{q} \quad \text{with } f \in \text{mrp}(SD_A)$$

$$\vec{q} ::= \langle \rangle \mid \langle p \rangle \oplus \vec{q}$$

The set of *standard polyadic terms* is then generated by the same rule, just ignoring the morphism in the abstraction case. In what follows we will constantly write Ξ instead of Ξ_A^S , keeping the parameters implicit³. We consider polyadic terms up to renaming of bound variables. Given a polyadic term p , we use $\text{fv}(p)$ for the set of *free variables* of p . Given a variable list $\vec{x} = \langle x_1, \dots, x_k \rangle$ we write $\vec{x} \in \text{fv}(p)$ meaning that for all $i \in [k]$, $x_i \in \text{fv}(p)$. A polyadic term p is

- *affine* if S is the semicartesian resource monad;
- *relevant* if S is the relevant resource monad;
- *linear* if S is the linear resource monad;
- *cartesian* if S is the cartesian resource monad.

Given a variable list $\vec{x} = \langle x_1, \dots, x_k \rangle$ we define the *list of occurrences of \vec{x} in p* by induction as follows:

3. It is also worth noting that, if the small category A admits homsets of uncountable cardinality, the set of terms Ξ_A will also be uncountable. This is not a standard feature of calculi, but it does not affect our results. However, if the reader is more comfortable with Ξ_A being countable, one can restrict without problem to any small category with countable homsets he/she likes the most.

4. Subtyping-Aware Polyadic Calculus and Rigid Expansion of Ordinary λ -Terms –
4.2. The Calculus of Subtyping-Aware Polyadic Terms

- if $p = x$ and $x \in \vec{x}$ then $\text{occ}_{\vec{x}}(p) = \langle x \rangle$. If $x \notin \vec{x}$ then $\text{occ}_{\vec{x}}(p) = \langle \rangle$;
- if $p = \lambda \vec{y} : f.p'$ then $\text{occ}_{\vec{x}}(p) = \text{occ}_{\vec{x}}(p')$;
- if $p = s\vec{t}$ then $\text{occ}_{\vec{x}}(p) = \text{occ}_{\vec{x}}(s) \oplus \text{occ}_{\vec{x}}(\vec{t})$;
- if $p = \langle q_1, \dots, q_k \rangle$ then $\text{occ}_{\vec{x}}(p) = \bigoplus_{i=1}^k \text{occ}_{\vec{x}}(q_i)$.

We define the *size* of a polyadic term by induction as follows:

$$s(x) = 1 \quad s(\lambda \vec{x} : f.p) = s(p) + 1 \quad s(s\langle t_1, \dots, t_k \rangle) = s(s) + \sum_{i \in [k]} s(t_i) + 1$$

The following remark is central for most of the technicalities that follows.

Notions of Substitution for Polyadic Terms We introduce standard and linear substitution operations for our calculus. While the standard one can perform structural operation on the term, like copying and deleting, the linear one cannot.

Definition 4.2.1. Let $p \in \Xi$, $\vec{q} = \langle q_1, \dots, q_k \rangle \in \Xi^!$ and $\vec{x} = \langle x_1, \dots, x_k \rangle$ be a repetitions-free variable list. We define the substitution of \vec{x} in p by \vec{q} by induction as follows:

$$\begin{aligned} x_i[\langle q_1, \dots, q_k \rangle / \vec{x}] &= q_i & y[\langle q_1, \dots, q_k \rangle / \vec{x}] &= y \\ (\lambda \vec{y} : f.p)[\vec{q} / \vec{x}] &= \lambda \vec{y} : f.(p[\vec{q} / \vec{x}]) & (s\vec{t})[\vec{q} / \vec{x}] &= s[\vec{q} / \vec{x}]t[\vec{q} / \vec{x}] \\ \langle p_1, \dots, p_n \rangle \{ \vec{q} / \vec{x} \} &= \langle p_1 \{ \vec{q} / \vec{x} \}, \dots, p_n \{ \vec{q} / \vec{x} \} \rangle \end{aligned}$$

whenever $\vec{y} \cap \vec{x} = \emptyset$ and $y \notin \vec{x}$.

The *n-linear substitution operation* on polyadic terms is defined by induction as follows:

Definition 4.2.2. Let $p \in \Xi$, $\vec{q} \in \Xi^!$ and $\vec{x} = \text{occ}_{\vec{x}}(p)$ with $\text{len}(\vec{q}) = \text{len}(\vec{x})$.

$$\begin{aligned} x\{\langle q \rangle / \langle x \rangle\} &= q & y\{\langle \rangle / \langle \rangle\} &= y \\ (\lambda \vec{y} : f.p)\{\vec{q} / \vec{x}\} &= \lambda \vec{y} : f.(p\{\vec{q} / \vec{x}\}) & (s\vec{t})\{\vec{q}_1 \oplus \vec{q}_2 / \vec{x}_1 \oplus \vec{x}_2\} &= s\{\vec{q}_1 / \vec{x}_1\}t\{\vec{q}_2 / \vec{x}_2\} \\ \langle p_1, \dots, p_n \rangle \{ \vec{q}_1 \oplus \dots \oplus \vec{q}_n / \vec{x}_1 \oplus \dots \oplus \vec{x}_n \} &= \langle p_1 \{ \vec{q}_1 / \vec{x}_1 \}, \dots, p_n \{ \vec{q}_n / \vec{x}_n \} \rangle \end{aligned}$$

whenever $\vec{y} \neq \vec{x}$, $y \notin \vec{x}$.

$$\begin{aligned} \vec{x} &= \vec{x}_1 \oplus \vec{x}_2 \text{ and } \vec{x}_1 = \text{occ}_{\vec{x}}(s), \vec{x}_2 = \text{occ}_{\vec{x}}(\vec{t}). \\ \vec{x} &= \bigoplus_{i \in [n]} \vec{x}_i \text{ and } \vec{x}_i = \text{occ}_{\vec{x}}(q_i). \end{aligned}$$

We define $\text{tlen}(\langle \vec{q}_1, \dots, \vec{q}_n \rangle) = \langle \text{len}(\vec{q}_1), \dots, \text{len}(\vec{q}_n) \rangle$. Given a repetitions-free tuple of variable lists $\zeta = \langle \vec{x}_1, \dots, \vec{x}_n \rangle$, $\vec{\rho} = \langle \vec{q}_1, \dots, \vec{q}_n \rangle \in (\Xi^!)^n$ and a polyadic term $p \in \Xi$ we define $\text{occ}_{\zeta}(p)$ in the natural way. For the substitution being well-defined, we require that $\text{tlen}(\zeta) = \text{tlen}(\vec{\rho})$. We also extend the notion of substitution and *n-linear substitution* to tuples of variables in the natural way, supposing that $\text{occ}_{\zeta}(p) = \zeta$.

4. Subtyping-Aware Polyadic Calculus and Rigid Expansion of Ordinary λ -Terms –
4.2. The Calculus of Subtyping-Aware Polyadic Terms

Proposition 4.2.3. *Let $p \in \Xi$. The following statements hold.*

1. *Given $\vec{x}, \vec{y}, \vec{z} \in \mathcal{V}^!$ we have $p\{\vec{y}/\vec{x}\}\{\vec{z}/\vec{y}\} = p\{\vec{z}/\vec{x}\}$.*
2. *Given $\zeta, \zeta_1, \zeta_2 \in (\mathcal{V}^!)^n$ we have $p\{\zeta_1/\zeta\}\{\zeta_2/\zeta_1\} = p\{\zeta_2/\zeta\}$.*

Proposition 4.2.4. *Let $p, p' \in \Xi$ such that $\text{fv}(p) = \text{fv}(p')$. The following statements hold.*

1. *Given $\vec{x}, \vec{y} \in \mathcal{V}^!$ if $p\{\vec{y}/\vec{x}\} = p'\{\vec{y}/\vec{x}\}$ then $p = p'$.*
2. *Given $\zeta_1, \zeta_2 \in (\mathcal{V}^!)^n$ if $p\{\zeta_2/\zeta_1\} = p'\{\zeta_2/\zeta_1\}$ then $p = p'$.*

Proposition 4.2.5. *Let $p \in \Xi, \vec{q} \in \Xi^!, \vec{x} \in \mathcal{V}^!$ with $\alpha : [\text{len}(\text{occ}_{\vec{x}}(p))] \rightarrow [\text{len}(\vec{x})]$ such that $x_i = y_{\alpha(i)}$ and $\text{len}(\vec{q}) = \text{len}(\vec{x})$. We have $p[\vec{q}/\vec{x}] = p\{\vec{q}\{\alpha\}/\text{occ}_{\vec{x}}(p)\}$.*

Proof. By induction on the structure of p . If $p = x$ with $x \in \vec{x}$, then $\text{occ}_{\vec{x}}(p) = \langle x \rangle$. By definition, $\alpha(1)$ is the index of x in \vec{x} . Hence $p[\vec{q}/\vec{x}] = q_{\alpha(1)}$. We conclude since $\vec{q}\{\alpha\} = \langle q_{\alpha(1)} \rangle$ and $p\{\vec{q}\{\alpha\}/\text{occ}_{\vec{x}}(p)\} = q_{\alpha(1)}$. If $p = x$ and $x \notin \vec{x}$ the result is immediate by definition. If $p = \lambda \vec{y} : f.p'$ then the result is a direct consequence of the IH. If $p = s\vec{t}$ then $\text{occ}_{\vec{x}}(p) = \text{occ}_{\vec{x}}(s) \oplus \text{occ}_{\vec{x}}(\vec{t})$. Let $\alpha|_s$ and $\alpha|_{\vec{t}}$ respectively the restrictions of α to $\text{occ}_{\vec{x}}(s)$ and $\text{occ}_{\vec{x}}(\vec{t})$. By IH we have that $s[\vec{q}/\vec{x}] = s\{\vec{q}\{\alpha|_s\}/\text{occ}_{\vec{x}}(s)\}$ and $\vec{t}[\vec{q}/\vec{x}] = \vec{t}\{\vec{q}\{\alpha|_{\vec{t}}\}/\text{occ}_{\vec{x}}(s)\}$. Then we can conclude simply applying the IH, observing that $\vec{q}\{\alpha\} = \vec{q}\{\alpha|_s\} \oplus \vec{q}\{\alpha|_{\vec{t}}\}$. \square

4.2.0.1. Type Assignment for Polyadic Terms

In order to give an operational semantics, we firstly need to introduce a typing assignment for our calculus. Indeed, types will clarify the computational role of morphisms.

We define SD^* , an extension of the category SD , as follows:

- $\text{ob}(SD^*) = \{\vec{x} : \vec{a} \mid \vec{a} \in SD, \vec{x} \in \mathcal{V}^! \text{ and } \text{len}(\vec{x}) = \text{len}(\vec{a})\}$.
- $(SD^*)^*(\vec{x} : \vec{a}, \vec{x}' : \vec{a}') = SD(\vec{a}, \vec{a}')$.

We define $(SD^n)^*$, an extension of the category SD^n , as follows:

- $\text{ob}((SD^n)^*) = \{\zeta : \Delta \mid \Delta \in SD^n, \zeta \in (\mathcal{V}^!)^n \text{ and } \text{tlen}(\zeta) = \text{tlen}(\Delta)\}$.
- $(SD^n)^*(\zeta : \Delta, \zeta' : \Delta') = SD^n(\Delta, \Delta')$.

It is worth noting that $SD^* = SD \times \mathcal{V}^!$. Moreover, we have $SD \simeq SD^*$.

By the former definition, one obtains several copies of the same morphism, indexed by tuples of variable lists. In particular, there will be fake identities $1_{\Delta} : \zeta : \Delta \rightarrow \zeta' : \Delta$, while the categorical identities clearly are $1_{\Delta} : \zeta : \Delta \rightarrow \zeta : \Delta$. For the rest of the chapter, when we consider a morphism $\zeta : \Delta \rightarrow \zeta' : \Delta'$ we suppose ζ being repetitions-free, while ζ' could admits repetitions, if not stated otherwise. We have an evident forgetful functor $U^* : (SD^n)^* \rightarrow SD^n$.

Let $\eta = \langle f_1, \dots, f_n \rangle \in (SD^n)^*(\langle \vec{x}_1, \dots, \vec{x}_n \rangle : \langle \vec{a}_1, \dots, \vec{a}_n \rangle, \zeta : \Delta)$. The tuple $\Sigma = \vec{x}_1 : f_1 : \vec{a}_1, \dots, \vec{x}_n : f_n : \vec{a}_n$ is a *polyadic type context*. For the rest of the

4. Subtyping-Aware Polyadic Calculus and Rigid Expansion of Ordinary λ -Terms –
 4.2. The Calculus of Subtyping-Aware Polyadic Terms

chapter, when we say type context we mean polyadic type context if not specified otherwise. We set $\vec{x} : \vec{a} ::= \vec{x} : \mathbf{1}_{\vec{a}} : \vec{a}$. We write $\zeta : \eta : \Delta$ to denote the three components of a context.

Given two type contexts $\Psi_1 = \vec{x}_1 : \mathbf{f}_1 : \vec{a}_1, \dots, \vec{x}_n : \mathbf{f}_n : \vec{a}_1$ and $\Psi_2 = \vec{y}_1 : \mathbf{g}_1 : \vec{b}_1, \dots, \vec{y}_n : \mathbf{g}_n : \vec{b}_n$ we define the tensor product as follows $\Psi_1 \otimes \Psi_2 = \vec{x}_1 \oplus \vec{y}_1 : \mathbf{f}_1 \oplus \mathbf{g}_1 : \vec{a}_1 \oplus \vec{b}_1, \dots, \vec{x}_n \oplus \vec{y}_n : \mathbf{f}_n \oplus \mathbf{g}_n : \vec{a}_n \oplus \vec{b}_n$. We give the following type assignment:

$$\frac{\frac{\frac{f_1 : \vec{a}_1 \rightarrow \langle \rangle, \dots, f_i : \vec{a}_i \rightarrow \langle a \rangle, \dots, f_n : \vec{a}_n \rightarrow \langle \rangle}{\vec{x}_1 : \mathbf{f}_1 : \vec{a}_1, \dots, \vec{x}_i : \mathbf{f}_i : \vec{a}_i, \dots, \vec{x}_n : \mathbf{f}_n : \vec{a}_n \vdash x_{i, \text{sm}(f_i)(1)} : a} \quad \zeta \oplus \langle \vec{z} \rangle : \eta \oplus \langle \mathbf{f} \rangle : \Delta \oplus \langle \vec{a} \rangle \vdash p : a}{\zeta : \eta : \Delta \vdash \lambda \vec{z} : \mathbf{f}. p : \vec{a} \Rightarrow a}}{\zeta_0 : \eta_0 : \Gamma_0 \vdash p : \langle a_1, \dots, a_k \rangle \Rightarrow a \quad (\zeta_i : \eta_i : \Gamma_i \vdash q_i : a_i)_{i=1}^k \quad \eta : \zeta : \Delta \rightarrow \bigotimes_{j=0}^k \zeta_j : \bigotimes_{j=0}^k \Gamma_j} \zeta : (\bigotimes_{j=0}^k \eta_j) \circ \eta : \Delta \vdash (p \langle q_1, \dots, q_k \rangle)^n : a}$$

Where in the application case ζ must be repetitions-free and

$$p^n = p[\zeta\{\text{sm}(\eta)\} / \bigotimes_{j=0}^k \zeta_j].$$

We remark that the inference rules preserves the properties of context variable tuples being repetitions-free. Context variable lists admit two types. The first one is a morphism in the category SD_A , the second one is the source of this morphism, *i.e.*, the standard type. From the syntactic point of view, the subtyping-aware polyadic calculus is just the standard polyadic calculus typed *à la Church* in the former intersection type system. However the Church-style typing concerns morphisms and it is not standard at all. Indeed, we shall see that the morphisms typing deeply affects the dynamics of subtyping-aware polyadic terms ⁴.

It is easy to check that

- if p is affine and it is typable, then a variable appears at most once in its body;
- if p is relevant and it is typable, then for any subterm of p that is an abstraction $\lambda \vec{x} : \mathbf{f}. p'$, we have $\vec{x} \in \text{fv}(p')$.
- if p is linear and it is typable, it satisfies the two former conditions.

If we need to make explicit an i -th element of a context, such as in $\zeta = (\zeta_1 \oplus \langle \vec{x}_i \rangle \oplus \zeta_2) : \eta = (\eta_1 \oplus \langle \mathbf{f}_i \rangle \oplus \eta_2) : \Delta = (\Delta_1 \oplus \langle \vec{a}_i \rangle \oplus \Delta_2) \vdash p : a$, we write $\zeta \curvearrowright^{\vec{x}_i} : \eta \curvearrowright^{\mathbf{f}_i} : \Delta \curvearrowright^{\vec{a}_i} \vdash p : a$. When π is a type derivation of conclusion $\zeta : \eta : \Delta \vdash p : a$ we write $\pi \vdash_{\zeta : \Delta}^a \langle \eta, p \rangle$.

4. This fact will determine subject reduction wrt the standard typing (but *not* wrt the morphisms typing) for subtyping-aware polyadic terms, see Section 4.2.3.

4. Subtyping-Aware Polyadic Calculus and Rigid Expansion of Ordinary λ -Terms –
4.2. The Calculus of Subtyping-Aware Polyadic Terms

Lemma 4.2.6. *Let $\zeta \circlearrowleft \vec{x}_i : \eta \circlearrowleft f_i : \Delta \circlearrowleft \vec{a}_i \vdash p : a$ we have $\text{occ}_{\vec{x}_i}(p) = \vec{x}_i\{\text{sm}(f_i)\}$ ⁵.*

Proof. By induction on the structure of p . Let $p = x_{i,\text{sm}(f_i)(1)}$ and let $x_i \in \vec{x}_i$. The result is then immediate by definition. If $x_i \notin \vec{x}_i$, then $f_i = \diamond_{\vec{a}_i}$ and we can conclude, since $\vec{x}_i\{\text{sm}(\diamond_{\vec{a}_i})\} = \langle \rangle$. The abstraction case is an immediate consequence of the IH. If $p = st$ then there exist $s' \in \Xi, \vec{t}' \in \Xi^!, \zeta_j : \Gamma_j \in (SD^{\text{len}(\zeta)})^*, \eta : \zeta : \Delta \rightarrow \otimes_{j=0}^k \zeta_j : \otimes_{j=0}^k \Gamma_j$ such that

$$\frac{\zeta_i \circlearrowleft \vec{y}_{0,i} : \eta_0 \circlearrowleft f_{0,i} : \Gamma_0 \circlearrowleft \vec{a}_{0,i} \vdash s' : \langle a_1, \dots, a_k \rangle \Rightarrow a \quad (\zeta_i \circlearrowleft \vec{y}_{l,i} : \eta_l \circlearrowleft f_{l,i} : \Gamma_l \circlearrowleft \vec{a}_{l,i} \vdash t'_l : a_l)_{l=1}^k \quad \eta' \circlearrowleft f_i : \zeta : \Delta \rightarrow \otimes_{j=0}^k \zeta_j : \otimes_{j=0}^k \Gamma_j}{\zeta \circlearrowleft \vec{x}_i : (\otimes_{j=0}^k \eta_j \circlearrowleft f_{j,i}) \circ \eta \circlearrowleft f'_i : \Delta \circlearrowleft \vec{a}_i \vdash p = (s' \langle t'_1, \dots, t'_k \rangle)^\eta : a}$$

where $f_i = (\oplus_{j=0}^k f_{j,i}) \circ f'_i$. By IH, $\text{occ}_{\vec{y}_{0,i}}(s') = \vec{y}_{0,i}\{\text{sm}(f_{0,i})\}$ and $\text{occ}_{\vec{y}_{l,i}}(t'_l) = \vec{y}_{l,i}\{\text{sm}(f_{l,i})\}$. Then $\text{occ}_{\oplus_{j=0}^k \vec{y}_{j,i}}(s' \vec{t}') = \oplus_{j=0}^k \vec{y}_{j,i}\{\text{sm}(\oplus_{j=0}^k f_{j,i})\}$. By definition $f'_i : \vec{x}_i : \vec{a}_i \rightarrow \oplus_{j=0}^k \vec{y}_{j,i} : \oplus_{j=0}^k \vec{a}_{j,i}$. By definition $(s' \vec{t}')^{f'_i} = s' \vec{t}'[\vec{x}_i\{\text{sm}(f'_i)\} / \oplus_{j=0}^k \vec{y}_{j,i}]$ and by Proposition 4.2.5 and definition of right action

$$s' \vec{t}'[\vec{x}_i\{\text{sm}(f'_i)\} / \oplus_{j=0}^k \vec{y}_{j,i}] = s' \vec{t}'\{\vec{x}_i\{\text{sm}(f_i)\} / \text{occ}_{\oplus_{j=0}^k \vec{y}_{j,i}}(s' \vec{t}')\}.$$

Since ζ is repetitions-free, we have that $\text{occ}_{(\vec{x}_i)}(s' \vec{t}')^\eta = \vec{x}_i\{\text{sm}(f_i)\}$. \square

We can generalize the observation to variable tuples: if $\zeta : \eta : \Delta \vdash p : a$ then $\text{occ}_\zeta(p) = \zeta\{\text{sm}(\eta)\}$. We also get that, given a type derivation $\pi \vdash_{\zeta:\Delta}^a \langle \eta, p \rangle$ with $\theta : \zeta' : \Delta' \rightarrow \zeta : \Delta$ we have $p[\zeta'\{\text{sm}(\theta)\} / \zeta] = p\{\zeta'\{\text{sm}(\eta \circ \theta)\} / \text{occ}_\zeta(p)\}$.

We denote as \mathcal{TD} the set of all type derivations. We define $TD_\zeta(\Delta, a) = \{\pi \in \mathcal{TD} \mid \pi \vdash_{\zeta:\Delta}^a \langle \eta, p \rangle \text{ for some } \eta \in SD^{\text{len}(\zeta)} \text{ and } p \in \Xi\}$. We clearly have that the sets $TD_\zeta(\Delta, a)$ are pairwise disjoint. Hence we can write

$$\mathcal{TD} = \bigsqcup_{n \in \mathbb{N}} \bigsqcup_{\langle \zeta:\Delta, a \rangle \in (SD^{\text{len}(\zeta)})^* \times D} TD_\zeta(\Delta, a).$$

We introduce some syntactic sugar for type derivations.

Let

$$\pi = \begin{array}{c} \pi \\ \vdots \\ \zeta \oplus \langle \vec{z} \rangle : \eta \oplus \langle f \rangle : \Delta \oplus \langle \vec{a} \rangle \vdash p : a \end{array}$$

We set

$$\lambda \vec{z} : f . \pi = \frac{\begin{array}{c} \pi \\ \vdots \\ \zeta \oplus \langle \vec{z} \rangle : \eta : \Delta \oplus \langle \vec{a} \rangle \vdash p : a \end{array}}{\zeta : \eta : \Delta \vdash \lambda \vec{z} : f . p : \vec{a} \Rightarrow a}$$

⁵. It is worth noting that, in general, while \vec{x} in this case is repetitions-free, $\vec{x}_i\{\text{sm}(f_i)\}$ is not. Just consider the case where $\vec{x}_i = \langle x \rangle$ and $f_i = \langle \alpha, \vec{f} \rangle = \langle c_{[2]}, 1, 1 \rangle : \langle a \rangle \rightarrow \langle a, a \rangle$ is the diagonal.

4. Subtyping-Aware Polyadic Calculus and Rigid Expansion of Ordinary λ -Terms –
4.2. The Calculus of Subtyping-Aware Polyadic Terms

Let

$$\pi_0 = \begin{array}{c} \pi_0 \\ \vdots \\ \zeta_0 : \eta_0 : \Gamma_0 \vdash p : \langle a_1, \dots, a_k \rangle \Rightarrow a \end{array} \quad \langle \pi_1, \dots, \pi_k \rangle = \left(\begin{array}{c} \pi_i \\ \vdots \\ \zeta_i : \eta_i : \Gamma_i \vdash q_i : a_i \end{array} \right)_{i=1}^k$$

and let $\eta : \zeta : \Delta \rightarrow \bigotimes_{j=0}^k \zeta_j : \bigotimes_{j=0}^k \Gamma_j$. We set $(\pi_0 \langle \pi_1, \dots, \pi_k \rangle) \circ \eta =$

$$\frac{\begin{array}{c} \pi_0 \\ \vdots \\ \zeta_0 : \eta_0 : \Gamma_0 \vdash p : \langle a_1, \dots, a_k \rangle \Rightarrow a \end{array} \quad \left(\begin{array}{c} \pi_i \\ \vdots \\ \zeta_i : \eta_i : \Gamma_i \vdash q_i : a_i \end{array} \right)_{i=1}^k \quad \eta : \zeta : \Delta \rightarrow \bigotimes_{j=0}^k \zeta_j : \bigotimes_{j=0}^k \Gamma_j}{\zeta : (\bigotimes_{j=0}^k \eta_j) \circ \eta : \Delta \vdash (p \langle q_1, \dots, q_k \rangle)^\eta : a}$$

Let $\pi \vdash_{\zeta:\Delta}^a \langle \eta, p \rangle$. We set $\text{term}(\pi) = p$, $\text{mrp}(\pi) = \eta$, $\text{supp}(\pi) = \zeta : \Delta$. Let $x \in \mathcal{V}$, we say that x is *relevant* (resp. *irrelevant*) to π if $x \in \text{fv}(\text{term}(\pi))$ (resp. $x \notin \text{fv}(\text{term}(\pi))$). Let $\vec{x} \in \mathcal{V}^!$, we say that \vec{x} is *relevant* (resp. *irrelevant*) to π if $\vec{x} \in \text{fv}(\text{term}(\pi))$ (resp. for all $x \in \vec{x}$, $x \notin \text{fv}(\text{term}(\pi))$).

Example 4.2.7. We provide some examples of type assignment for polyadic terms.

- Let $p = (\lambda \langle x_1, x_2 \rangle : \langle \sigma, f, 1_{\langle a \rangle \Rightarrow a} \rangle . x_2 \langle x_1 \rangle)$ where σ is the permutation that performs the swap and $f : a' \rightarrow a$. We type it as follows:

$$\frac{\frac{1}{\langle z_1 \rangle : \langle \langle a \rangle \Rightarrow a \rangle \vdash z_1 : \langle a \rangle \Rightarrow a} \quad \frac{\langle f \rangle : \langle a' \rangle \rightarrow \langle a \rangle}{\langle z_2 \rangle : \langle f \rangle : \langle a' \rangle \vdash z_2 : a} \quad \langle \sigma \rangle : \langle \langle x_1, x_2 \rangle \rangle : \langle a', \langle a \rangle \Rightarrow a \rangle \rightarrow \langle \langle z_1, z_2 \rangle \rangle : \langle \langle a \rangle \Rightarrow a, a' \rangle}}{\frac{\langle x_1, x_2 \rangle : \langle \sigma, f, 1 \rangle : \langle a', \langle a \rangle \Rightarrow a \rangle \vdash x_2 \langle x_1 \rangle : a}{\vdash \lambda \langle x_1, x_2 \rangle : \langle \sigma, f, 1 \rangle . x_2 \langle x_1 \rangle : \langle a', \langle a \rangle \Rightarrow a \rangle \Rightarrow a}}$$

- Let $p = x \langle y, y \rangle$ and $b = \langle a, a \rangle \Rightarrow a$. We recall that $c_a : \langle a \rangle \rightarrow \langle a, a \rangle$ denotes the diagonal morphism.

$$\frac{\frac{\langle x \rangle : \langle b, \langle \rangle \rangle : \langle \rangle \vdash x : \langle a, a \rangle \Rightarrow a}{\langle \rangle : \langle \rangle, \langle z_i \rangle : \langle a \rangle \vdash z_i : a} \quad \langle 1, c_a \rangle : \langle \langle x \rangle, \langle y \rangle \rangle : \langle \langle b, \langle a \rangle \rangle \rangle \rightarrow \langle \langle x \rangle, \langle z_1, z_2 \rangle \rangle : \langle \langle b, \langle a, a \rangle \rangle \rangle}}{\langle x \rangle : \langle b, \langle y \rangle \rangle : \langle a \rangle \vdash x \langle y, y \rangle : a}$$

- Let $p = \lambda \langle x_1, \dots, x_k \rangle : \pi_i . x_i$ with $\pi_i : \langle a_1, \dots, a_k \rangle \rightarrow \langle a_i \rangle$ being the i -th projection.

$$\frac{\pi_i : \langle a_1, \dots, a_k \rangle \rightarrow \langle a_i \rangle}{\frac{\langle x_1, \dots, x_k \rangle : \pi_i : \langle a_1, \dots, a_k \rangle \vdash x_i : a_i}{\vdash \lambda \langle x_1, \dots, x_k \rangle : \pi_i . x_i : a_i}}$$

4.2.1. Actions on Polyadic Type Derivations

We can extend the definition of morphisms action on type derivation given in Chapter 3 to the polyadic framework.

4. Subtyping-Aware Polyadic Calculus and Rigid Expansion of Ordinary λ -Terms –
 4.2. The Calculus of Subtyping-Aware Polyadic Terms

$$\begin{array}{c}
 \left(\frac{f_1 : \vec{a}_1 \rightarrow \langle \rangle, \dots, f_i = \langle \alpha_i, f \rangle : \vec{a}_i \rightarrow \langle a \rangle, \dots, f_n : \vec{a}_n \rightarrow \langle \rangle}{\vec{x}_1 : \vec{f}_1 : \vec{a}_1, \dots, \vec{x}_i : \vec{f}_i : \vec{a}_i, \dots, \vec{x}_n : \vec{f}_n : \vec{a}_n \vdash x_{i, \alpha_i(1)} : a} \right) \{\eta\} = \\
 \frac{f_1 \circ g_1 : \vec{b}_1 \rightarrow \langle \rangle, \dots, (f_i \circ g_i) = \langle \beta_i \circ \alpha_i, (\langle f \rangle \{\beta_i\}) \circ \vec{g}_i \rangle : \vec{b}_i \rightarrow \langle a \rangle, \dots, f_n \circ g_n : \vec{b}_n \rightarrow \langle \rangle}{\vec{y}_1 : \vec{f}_1 \circ g_1 : \vec{b}_1, \dots, \vec{y}_i : \vec{f}_i \circ g_i : \vec{b}_i, \dots, \vec{y}_n : \vec{f}_n \circ g_n : \vec{b}_n \vdash y_{i, (\beta \circ \alpha)_i(1)} : a} \\
 \left(\frac{\begin{array}{c} \pi \\ \vdots \\ \zeta \oplus \langle \vec{x} \rangle : \eta \oplus \langle f \rangle : \Delta \oplus \langle \vec{a} \rangle : \vec{a} \vdash p : a \end{array}}{\zeta : \eta : \Delta \vdash \lambda \vec{x} : \vec{f}. p : \vec{a} \Rightarrow a} \right) \{\eta\} = \frac{\begin{array}{c} \pi \{\langle \eta \rangle \oplus \langle 1_{\vec{a}} \rangle\} \\ \vdots \\ \zeta' \oplus \langle \vec{x} \rangle : \eta' \oplus \langle f \rangle : \Delta' \oplus \langle \vec{a} \rangle \vdash p^\eta : a \end{array}}{\zeta' : \eta' : \Delta' \vdash \lambda \vec{x} : \vec{f}. (p^\eta) : \vec{a} \Rightarrow a} \\
 \left(\frac{\begin{array}{c} \pi_1 \\ \vdots \\ \zeta_1 : \eta_1 : \Gamma_1 \vdash p : \vec{a} \Rightarrow a \end{array} \quad \left(\zeta_i : \eta_i : \Gamma_i \vdash q_i : a_i \right)_{i=1}^k \quad \theta : \zeta : \Delta \rightarrow \bigotimes_{j=0}^k \zeta_j : \bigotimes_{j=0}^k \Gamma_j}{\zeta : (\bigotimes_{j=0}^k \eta_j) \circ \theta : \Delta \vdash (p \langle q_1, \dots, q_k \rangle)^\eta : a} \right) \{\eta\} = \\
 \frac{\begin{array}{c} \pi_1 \\ \vdots \\ \zeta_1 : \eta_1 : \Gamma_1 \vdash p : \vec{a} \Rightarrow a \end{array} \quad \left(\zeta_i : \eta_i : \Gamma_i \vdash q_i : a_i \right)_{i=1}^k \quad \theta \circ \eta : \zeta' : \Delta' \rightarrow \bigotimes_{j=0}^k \zeta_j : \bigotimes_{j=0}^k \Gamma_j}{\zeta' : (\bigotimes_{j=0}^k \eta_j) \circ (\theta \circ \eta) : \Delta' \vdash (p \langle q_1, \dots, q_k \rangle)^{\theta \circ \eta} : a}
 \end{array}$$

Where $\vec{a} = \langle a_1, \dots, a_k \rangle$, $\eta = \langle g_1, \dots, g_n \rangle : \zeta = \langle \vec{y}_1, \dots, \vec{y}_n \rangle : \Delta' = \langle \vec{b}_1, \dots, \vec{b}_n \rangle \rightarrow \zeta' = \langle \vec{x}_1, \dots, \vec{x}_n \rangle : \Delta = \langle \vec{a}_1, \dots, \vec{a}_n \rangle$ with $g_i = \langle \beta_i, \vec{g}_i \rangle$. In the abstraction case we take $\langle 1_{\vec{a}} \rangle : \langle \vec{x} \rangle : \langle \vec{a} \rangle \rightarrow \langle \vec{x} \rangle : \langle \vec{a} \rangle$.

Figure 4.1. – Right action on polyadic type derivations.

Given $\pi \vdash_{\zeta : \Delta}^a \langle \theta, p \rangle$ and a morphism $\eta : \zeta' : \Delta' \rightarrow \zeta : \Delta$ where ζ' is repetitions-free⁶, we define the *right action of η on π* by induction in Figure 4.1, that performs the following transformation on type derivations:

$$\left(\frac{\begin{array}{c} \pi \\ \vdots \\ \zeta : \theta : \Delta \vdash p : a \end{array}}{\zeta : \theta : \Delta \vdash p : a} \right) \{\eta\} = \frac{\begin{array}{c} \pi \{\eta\} \\ \vdots \\ \zeta : \theta \circ \eta : \Delta \vdash p^\eta : a \end{array}}{\zeta : \theta \circ \eta : \Delta \vdash p^\eta : a}$$

The former action can be seen as a generalization of what happens in the typing rule of an application, where a morphism of contexts acts on the free variables of a polyadic term. It essentially consists of free variables substitution and context morphisms precomposition. This substitution performs *structural* operations on the term's free variables: they can eventually be duplicated or shifted.

Let $f : a \rightarrow a'$. We define the *left action of f on π* by induction in Figure 4.2, that performs the following transformation on type derivations:

6. We remark that, by convention and definition of the type assignment, both tuples ζ, ζ' are repetitions-free.

4. Subtyping-Aware Polyadic Calculus and Rigid Expansion of Ordinary λ -Terms –
4.2. The Calculus of Subtyping-Aware Polyadic Terms

$$\begin{aligned}
& [g : a \rightarrow b] \left(\frac{f_1 : \vec{a}_1 \rightarrow \langle \rangle, \dots, f_i = \langle \alpha_i, f \rangle : \vec{a}_i \rightarrow \langle a \rangle, \dots, f_n : \vec{a}_n \rightarrow \langle \rangle}{\vec{x}_1 : \mathbf{f}_1 : \vec{a}_1, \dots, \vec{x}_i : \mathbf{f}_i : \vec{a}_i, \dots, \vec{x}_n : \mathbf{f}_n : \vec{a}_n \vdash x_{i, \alpha_i(1)} : a} \right) = \\
& \frac{f_1 : \vec{a}_1 \rightarrow \langle \rangle, \dots, \langle g \rangle \circ f_i = \langle \alpha, g \circ f \rangle : \vec{a}_i \rightarrow \langle b \rangle, \dots, f_n : \vec{a}_n \rightarrow \langle \rangle}{\vec{x}_1 : \mathbf{f}_1 : \vec{a}_1, \dots, \vec{x}_i : \langle g \rangle \circ \mathbf{f}_i : \vec{a}_i, \dots, \vec{x}_n : \mathbf{f}_n : \vec{a}_n \vdash x_{i, \alpha_i(1)} : b} \\
& [\langle \tau, \vec{g} \rangle \Rightarrow g : (\vec{a} \Rightarrow a) \rightarrow (\vec{b} \Rightarrow b)] \left(\frac{\left(\frac{\pi}{\Sigma, \vec{x} : \mathbf{f} : \vec{a} \vdash p : a} \right)}{\Sigma \vdash \lambda \vec{x} : \mathbf{f} . p : \vec{a} \Rightarrow a} \right) = \frac{([g]\pi)\{1_\Delta \oplus \langle \langle \tau, \vec{g} \rangle \rangle\}}{\vdots} \\
& \frac{[g]\Sigma, \vec{y} : ([g]\mathbf{f}) \circ \langle \tau, \vec{g} \rangle : \vec{b} \vdash [g]p[\vec{y}\{\tau\}/\vec{x}] : b}{[g]\Sigma \vdash \lambda \vec{y} : ([g]\mathbf{f}) \circ \langle \tau, \vec{g} \rangle . ([g]p[\vec{y}\{\tau\}/\vec{x}]) : \vec{b} \Rightarrow b} \\
& [g : a \rightarrow b] \left(\frac{\left(\frac{\pi_1}{\zeta_1 : \eta_1 : \Gamma_1 \vdash p : \vec{a} \Rightarrow a} \right) \left(\frac{\pi_i}{\zeta_i : \eta_i : \Gamma_i \vdash q_i : a_i} \right)_{i=1}^k \quad \eta : \zeta : \Delta \rightarrow \bigotimes_{j=0}^k \zeta_j : \bigotimes_{j=0}^k \Gamma_j}{\zeta : (\bigotimes_{j=0}^k \eta_j) \circ \eta : \Delta \vdash (p\langle q_1, \dots, q_k \rangle)^n : a} \right) = \\
& \frac{[1 \Rightarrow g]\pi_1}{\vdots} \left(\frac{\left(\frac{\pi_i}{\zeta_i : \eta_i : \Gamma_i \vdash q_i : a_i} \right)_{i=1}^k \quad \eta : \zeta : \Delta \rightarrow \bigotimes_{j=0}^k \zeta_j : \bigotimes_{j=0}^k \Gamma_j}{\zeta : (1 \Rightarrow g)\eta_0 \otimes \bigotimes_{j=0}^k \eta_i \circ \eta : \Delta \vdash (([1 \Rightarrow g]p)\langle q_1, \dots, q_k \rangle)^n : b} \right) \\
& \zeta_1 : [1 \Rightarrow g]\eta_1 : \Gamma_1 \vdash [1 \Rightarrow g]p : \vec{a} \Rightarrow b \quad \left(\frac{\pi_i}{\zeta_i : \eta_i : \Gamma_i \vdash q_i : a_i} \right)_{i=1}^k \quad \eta : \zeta : \Delta \rightarrow \bigotimes_{j=0}^k \zeta_j : \bigotimes_{j=0}^k \Gamma_j
\end{aligned}$$

Where, $\vec{a} = \langle a_1, \dots, a_k \rangle$ and in the abstraction case, \vec{y} is a fresh variable list s.t. $\text{len}(\vec{y}) = \text{len}(\vec{a}')$.

Figure 4.2. – Left action on polyadic type derivations.

$$[f] \left(\frac{\pi}{\zeta : \theta : \Delta \vdash p : a} \right) = \frac{[f]\pi}{\zeta : [f]\theta : \Delta \vdash [f]p : a'}$$

It is worth noting that the left action depends on the right one in the abstraction case. This is due to the contravariant nature of a morphism between implications: a morphism $\langle \alpha, \vec{f} \rangle \Rightarrow f : \vec{a} \Rightarrow a \rightarrow \vec{a}' \Rightarrow a'$ consist of $\langle \alpha, \vec{f} \rangle : \vec{a}' \rightarrow \vec{a}$ and $f : a \rightarrow a'$. The left action changes the type of the leftmost variable of a polyadic term (Remark 4.2.8) and transforms lambda abstractions⁷. Hence, on the contrary to what happened with the actions defined in Chapter 3, now the structure of terms is affected. This is completely in accordance to the intuition that our polyadic calculus is the *term language* of the intersection type system E_A^S .

Remark 4.2.8. We define the leftmost variable of a polyadic term p by induction as follows:

$$lv(x) = x \quad lv(\lambda \vec{x} : \mathbf{f} . p) = lv(p) \quad lv(s\vec{t}) = lv(s)$$

Given $\pi \vdash_{\zeta : \Delta}^a \langle \eta, p \rangle$, supposing that $lv(p)$ is free, we set $lv(\zeta)$ as the unique variable list $\vec{x} \in \zeta$ such that $lv(p) \in \vec{x}$. We denote its typing as $lv(\zeta) : lv(\eta) : lv(\Delta)$. We

7. Notice that the size of the abstracted variable list can eventually change.

4. Subtyping-Aware Polyadic Calculus and Rigid Expansion of Ordinary λ -Terms –
4.2. The Calculus of Subtyping-Aware Polyadic Terms

remark that for $\theta = \langle g_1, \dots, g_n \rangle$ we have $[f]\theta = \langle g_1, \dots, [f]lw(\theta), \dots, g_n \rangle$ if $lw(p)$ is free, otherwise $[f]\theta = \theta$. Hence the left action transforms the typing of the leftmost variable of a polyadic term.

We prove that actions respect compositionality of arrows.

Lemma 4.2.9. *Let $\pi \vdash_{\zeta:\Delta}^a \langle \eta, p \rangle$ and $\theta_1 : \zeta_1 : \Delta_1 \rightarrow \zeta : \Delta, \theta_2 : \zeta_2 : \Delta_2 \rightarrow \zeta_1 : \Delta_1, f_1 : a \rightarrow a_1, f_2 : a_1 \rightarrow a_2$. The following statements hold.*

1. $(\pi\{\theta_1\})\{\theta_2\} = \pi\{\theta_1 \circ \theta_2\}$.
2. $[f_1](\pi\{\theta_1\}) = ([f_1]\pi)\{\theta_1\}$.
3. $[f_2]([f_1]\pi) = [f_2 \circ f_1]\pi$.

Proof. The proof of both statements follow a specular path to the one already seen for actions on type derivations in Chapter 3 (Lemma 3.4.5). However, in this case the term can be transformed by the actions, so we need to check that everything still works fine.

1. The result derives from a completely straightforward induction on π by associativity of morphisms composition, observing that given $\pi \vdash_{\zeta:\Delta}^a \langle \eta, p \rangle, \pi\{\theta_1\} \vdash_{\zeta_1:\Delta_1}^a \langle \eta \circ \theta_1, p^{\theta_1} \rangle$ and $(\pi\{\theta_1\})\{\theta_2\} \vdash_{\zeta_2:\Delta_2}^a \langle (\eta \circ \theta_1) \circ \theta_2, (p^{\theta_1})^{\theta_2} \rangle$ then $(p^{\theta_1})^{\theta_2} = p^{\theta_1 \circ \theta_2}$. Indeed, by definition $(p^{\theta_1})^{\theta_2} = p[\zeta_1\{\text{sm}(\theta_1)\}/\zeta][\zeta_2\{\text{sm}(\theta_2)\}/\zeta_1]$. By Lemma we can rewrite it as $p\{\zeta_1\{\text{sm}(\theta_1) \circ \text{sm}(\eta)\}/\text{occ}_\zeta(p)\}\{\zeta_2\{\text{sm}(\theta_2) \circ (\text{sm}(\theta_1) \circ \text{sm}(\eta))\}/\text{occ}_{\zeta_1}(p^\theta)\}$. By definition, $\text{sm}((\eta \circ \theta) \circ \theta') = \text{sm}(\theta_2) \circ (\text{sm}(\theta_1) \circ \text{sm}(\eta))$. Then by associativity of composition and definition of right action we can conclude, since $p^{\theta_1 \circ \theta_2} = p[\zeta_2\{\text{sm}(\theta_2) \circ \text{sm}(\theta_1)\}/\zeta] = p\{\zeta_2\{(\text{sm}(\theta_2) \circ \text{sm}(\theta_1)) \circ \eta\}/\text{occ}_\zeta(p)\}$.
2. The variable case derives by associativity of morphisms composition and the application case is immediate by definition of actions. We prove the abstraction case. Let $\pi =$

$$\frac{\begin{array}{c} \pi' \\ \vdots \\ \zeta \oplus \langle \vec{z} \rangle : \eta \oplus \langle f \rangle : \Delta \oplus \langle \vec{a} \rangle \vdash p : a \end{array}}{\zeta : \eta : \Delta \vdash \lambda \vec{z} : f.p : \vec{a} \Rightarrow a}$$

By definition $\pi\{\theta_1\} =$

$$\frac{\begin{array}{c} \pi'\{\theta_1 \oplus \{1_{\vec{a}}\}\} \\ \vdots \\ \zeta_1 \oplus \langle \vec{z} \rangle : (\eta \circ \theta_1) \oplus \langle f \rangle : \Delta_1 \oplus \langle \vec{a} \rangle \vdash p^{\theta \oplus (1_{\vec{a}})} : a \end{array}}{\zeta_1 : \eta \circ \theta_1 : \Delta_1 \vdash \lambda \vec{z} : f.(p^{\theta \oplus (1_{\vec{a}})}) : \vec{a} \Rightarrow a}$$

Let $f_1 = \langle \alpha, \vec{g} \rangle \Rightarrow g : \vec{a} \Rightarrow a \rightarrow \vec{a}' \Rightarrow a'$ and $\text{len}(\vec{z}') = \text{len}(\text{source}(\langle \alpha, \vec{g} \rangle))$. By

4. Subtyping-Aware Polyadic Calculus and Rigid Expansion of Ordinary λ -Terms –
4.2. The Calculus of Subtyping-Aware Polyadic Terms

definition $[f_1](\pi\{\theta_1\}) =$

$$\frac{([g](\pi'\{\theta_1 \oplus \langle 1_{\vec{a}} \rangle\}))\{1_{\Delta'} \oplus \langle \langle \alpha, \vec{g} \rangle \rangle\}}{\vdots} \\ \zeta_1 \oplus \langle \vec{z}' \rangle : [g](\eta \circ \theta_1) \oplus \langle ([g]f) \circ \langle \alpha, \vec{g} \rangle \rangle : \Delta_1 \oplus \langle \vec{a}' \rangle \vdash ([g](p)^{\theta \oplus (1)})^{\bar{1} \oplus \langle \langle \alpha, \vec{g} \rangle \rangle} : a'$$

$$\zeta_1 : [g](\eta \circ \theta_1) : \Delta_1 \vdash \lambda \vec{z}' : ([g]f) \circ \langle \alpha, \vec{g} \rangle . [g]((p)^{\theta \oplus (1)})^{\bar{1} \oplus \langle \langle \alpha, \vec{g} \rangle \rangle} : \vec{a}' \Rightarrow a'$$

by IH $[g](\pi'\{\theta_1 \oplus \langle 1_{\vec{a}} \rangle\})\{1_{\Delta'} \oplus \langle \langle \alpha, \vec{g} \rangle \rangle\} = (([g]\pi')\{\theta_1 \oplus \{1\}\})\{\bar{1} \oplus \langle \langle \alpha, \vec{g} \rangle \rangle\}$. By the former point of this lemma we have $(([g]\pi')\{\theta_1 \oplus \{1_{\vec{a}}\}\})\{1_{\Delta'} \oplus \langle \langle \alpha, \vec{g} \rangle \rangle\} = ([g]\pi')\{\theta \oplus \langle \langle \alpha, \vec{g} \rangle \rangle\}$. Now, consider $([f_1]\pi)\{\theta_1\} =$

$$(([g]\pi')\{1_{\Delta} \oplus \langle \langle \alpha, \vec{g} \rangle \rangle\})\{\theta_1 \oplus \langle 1_{\vec{a}'} \rangle\}$$

$$\vdots$$

$$\zeta_1 \oplus \langle \vec{z}' \rangle : ((([g]\eta) \circ \theta_1) \oplus \langle ([g]f) \circ \langle \alpha, \vec{g} \rangle \rangle) : \Delta_1 \oplus \langle \vec{a}' \rangle \vdash ((([g]p)^{\bar{1} \oplus \langle \langle \alpha, \vec{g} \rangle \rangle})^{\theta \oplus (1)}) : a'$$

$$\zeta_1 : [g](\eta) \circ \theta_1 : \Delta_1 \vdash \lambda \vec{z}' : ([g]f) \circ \langle \alpha, \vec{g} \rangle . ((([g]p)^{\bar{1} \oplus \langle \langle \alpha, \vec{g} \rangle \rangle})^{\theta \oplus (1)}) : \vec{a}' \Rightarrow a'$$

Again, by the former point, we get $(([g]\pi')\{1_{\Delta} \oplus \langle \langle \alpha, \vec{g} \rangle \rangle\})\{\theta_1 \oplus \langle 1_{\vec{a}'} \rangle\} = ([g]\pi')\{\theta \oplus \langle \langle \alpha, \vec{g} \rangle \rangle\}$. We then apply the IH and conclude.

3. We prove the abstraction case. Let $f_1 = \langle \alpha_1, \vec{g}_1 \rangle : \vec{a} \Rightarrow a \rightarrow \vec{a}_1 \Rightarrow a_1$ and $f_2 = \langle \alpha_2, \vec{g}_2 \rangle : \vec{a}_1 \Rightarrow a_1 \rightarrow \vec{a}_2 \Rightarrow a_2$. Let $\text{len}(\vec{z}_1) = \text{len}(\vec{a}_1)$ and $\text{len}(\vec{z}_2) = \text{len}(\vec{a}_2)$. Let $\pi =$

$$\pi'$$

$$\vdots$$

$$\zeta \oplus \langle \vec{z} \rangle : \eta \oplus \langle f \rangle : \Delta \oplus \langle \vec{a} \rangle \vdash p : a$$

$$\zeta : \eta : \Delta \vdash \lambda \vec{z} : f.p : \vec{a} \Rightarrow a$$

We have $[f_1]\pi =$

$$([g_1]\pi')\{1_{\Delta} \oplus \langle \langle \alpha_1, \vec{g}_1 \rangle \rangle\}$$

$$\vdots$$

$$\zeta \oplus \langle \vec{z}_1 \rangle : ([g_1]\eta) \oplus \langle [g_1]f \circ \langle \alpha_1, \vec{g}_1 \rangle \rangle : \Delta \oplus \langle \vec{a}_1 \rangle \vdash ([g_1]p)^{1_{\Delta} \oplus \langle \langle \alpha_1, \vec{g}_1 \rangle \rangle} : a_1$$

$$\zeta : [g_1]\eta : \Delta \vdash \lambda \vec{z}_1 : \langle [g_1]f \circ \langle \alpha_1, \vec{g}_1 \rangle \rangle . ([g_1]p)^{1_{\Delta} \oplus \langle \langle \alpha_1, \vec{g}_1 \rangle \rangle} : \vec{a} \Rightarrow a$$

Then $[f_2]([f_1]\pi) =$

$$([g_2]((([g_1]\pi')\{1_{\Delta} \oplus \langle \langle \alpha_1, \vec{g}_1 \rangle \rangle\}))\{1_{\Delta} \oplus \langle \langle \alpha_2, \vec{g}_2 \rangle \rangle\})$$

$$\vdots$$

$$\zeta \oplus \langle \vec{z}_2 \rangle : [g_2]([g_1]\eta) \oplus \langle ([g_2]([g_1]f) \circ \langle \alpha_1, \vec{g}_1 \rangle)) \circ \langle \alpha_2, \vec{g}_2 \rangle \rangle : \Delta \oplus \langle \vec{a}_2 \rangle \vdash ([g_2]([g_1]p)^{1_{\Delta} \oplus \langle \langle \alpha_1, \vec{g}_1 \rangle \rangle})^{1_{\Delta} \oplus \langle \langle \alpha_2, \vec{g}_2 \rangle \rangle} : a_2$$

$$\zeta : [g_2]([g_1]\eta) : \Delta \vdash \lambda \vec{z}_1 : \langle ([g_2]([g_1]f) \circ \langle \alpha_1, \vec{g}_1 \rangle)) \circ \langle \alpha_2, \vec{g}_2 \rangle \rangle . ([g_2]([g_1]p)^{1_{\Delta} \oplus \langle \langle \alpha_1, \vec{g}_1 \rangle \rangle})^{1_{\Delta} \oplus \langle \langle \alpha_2, \vec{g}_2 \rangle \rangle} : \vec{a}_2 \Rightarrow a_2$$

by the former point of this lemma we get $([g_2]((([g_1]\pi')\{1_{\Delta} \oplus \langle \alpha_1, \vec{g}_1 \rangle \rangle\}))\{1_{\Delta} \oplus \langle \langle \alpha_2, \vec{g}_2 \rangle \rangle\} = (([g_2]([g_1]\pi')\{1_{\Delta} \oplus \langle \langle \alpha_1, \vec{g}_1 \rangle \rangle\}))\{1_{\Delta} \oplus \langle \langle \alpha_2, \vec{g}_2 \rangle \rangle\}$. Then, by IH: $(([g_2]([g_1]\pi')\{1_{\Delta} \oplus \langle \langle \alpha_1, \vec{g}_1 \rangle \rangle\}))\{1_{\Delta} \oplus \langle \langle \alpha_2, \vec{g}_2 \rangle \rangle\} = (([g_2 \circ g_1]\pi')\{1_{\Delta} \oplus \langle \langle \alpha_1, \vec{g}_1 \rangle \rangle\})\{1_{\Delta} \oplus$

4. Subtyping-Aware Polyadic Calculus and Rigid Expansion of Ordinary λ -Terms –
 4.2. The Calculus of Subtyping-Aware Polyadic Terms

$\langle\langle\alpha_2, \vec{g}_2\rangle\rangle\}$. By the first point of this lemma we can rewrite it as $([g_2 \circ g_1]\pi')\{1_\Delta \oplus \langle\langle\alpha_2, \vec{g}_2\rangle \circ \langle\alpha_2, \vec{g}_2\rangle\rangle\}$. Then we can apply the IH and conclude. \square

4.2.2. Congruence on Polyadic Type Derivations

Let $R \subseteq \mathcal{TD}^2$. We say that R is a *congruence* if it is an equivalence relation that satisfies the following contextual rules:

1. If $\pi \vdash_{\zeta \oplus \vec{x} : \Delta \oplus \langle \vec{a} \rangle}^a \langle \eta \oplus \langle f \rangle, p \rangle$ and $\pi' \vdash_{\zeta \oplus \vec{x} : \Delta \oplus \langle \vec{a} \rangle}^a \langle \eta' \oplus \langle f' \rangle, p' \rangle$ such that $\pi R \pi'$ then $\lambda \vec{x} : f.\pi R \lambda \vec{x} : f'.\pi'$.
2. If $\pi_0 \vdash_{\zeta_0 : \Gamma_0}^{\vec{a} \Rightarrow a} \langle \eta_0, s \rangle$, $\pi'_0 \vdash_{\zeta'_0 : \Gamma'_0}^{\vec{a} \Rightarrow a} \langle \eta'_0, s' \rangle$ and $\pi_i \vdash_{\zeta_i : \Gamma_i}^{a_i} \langle \eta_i, t_i \rangle$, $\pi'_i \vdash_{\zeta'_i : \Gamma'_i}^{a_i} \langle \eta'_i, t'_i \rangle$ for $i \in [k]$ such that $\pi_0 R \pi'_0$ and $\pi_i R^{\zeta_i : \Gamma_i, a_i} \pi'_i$ with $\eta : \Delta \rightarrow \bigotimes_{j=0}^k \Gamma_j$ then

$$(\pi_0 \langle \pi_1, \dots, \pi_k \rangle) \circ \eta R (\pi'_0 \langle \pi'_1, \dots, \pi'_k \rangle) \circ \eta.$$

We define a congruence on polyadic type derivations as the smallest congruence generated by the following rule:

$$\frac{\begin{array}{c} \pi_0 \{ \eta_0 \} \\ \vdots \\ \zeta_0 : \theta_0 \circ \eta_0 : \Gamma_0 \vdash s^{\eta_0} : \vec{a} \Rightarrow a \end{array} \quad \left(\begin{array}{c} \pi_i \{ \eta_i \} \\ \vdots \\ \zeta_i : \theta_i \circ \eta_i : \Gamma_i \vdash q_i^{\eta_i} : a_i \end{array} \right)_{i=1}^k \quad \theta : \zeta : \Delta \rightarrow \bigotimes_{j=0}^k \zeta_j : \bigotimes_{j=0}^k \Gamma_j}{\zeta : (\bigotimes_{j=0}^k \theta_j) \circ (\bigotimes_{j=0}^k \eta_j) \circ \theta : \Delta \vdash (s^{\eta_0} \langle q_1^{\eta_1}, \dots, q_k^{\eta_k} \rangle)^\theta : a} \sim \frac{\begin{array}{c} \pi_0 \\ \vdots \\ \zeta'_0 : \theta_0 : \Gamma'_0 \vdash s : \vec{a} \Rightarrow a \end{array} \quad \left(\begin{array}{c} \pi_i \\ \vdots \\ \zeta'_i : \theta_i : \Gamma'_i \vdash q_i : a_i \end{array} \right)_{i=1}^k \quad (\bigotimes_{j=0}^k \eta_j) \circ \theta : \zeta : \Delta \rightarrow \bigotimes_{j=0}^k \zeta'_j : \bigotimes_{j=0}^k \Gamma'_j}{\zeta : (\bigotimes_{j=0}^k \theta_j) \circ (\bigotimes_{j=0}^k \eta_j) \circ \theta : \Delta \vdash (s\vec{q})^{\bigotimes_{j=0}^k \eta_j \circ \theta} : a}$$

Where $\vec{q} = \langle q_1, \dots, q_k \rangle$ and $\eta_j : \zeta_j : \Gamma_j \rightarrow \zeta'_j : \Gamma'_j$. By the former equivalence, we get that any choice of free variable tuples in the application rule is actually equivalent. Indeed:

$$\frac{\begin{array}{c} \pi_0 \{ 1_{\Gamma'_0} \} \\ \vdots \\ \zeta_0 : \theta_0 \circ 1_{\Gamma'_0} : \Gamma_0 \vdash s^{1_{\Gamma'_0}} : \vec{a} \Rightarrow a \end{array} \quad \left(\begin{array}{c} \pi_i \{ 1_{\Gamma'_i} \} \\ \vdots \\ \zeta_i : \theta_i \circ 1_{\Gamma'_i} : \Gamma'_i \vdash q_i^{1_{\Gamma'_i}} : a_i \end{array} \right)_{i=1}^k \quad \theta : \zeta : \Delta \rightarrow \bigotimes_{j=0}^k \zeta_j : \bigotimes_{j=0}^k \Gamma'_j}{\zeta : (\bigotimes_{j=0}^k \theta_j) \circ \theta : \Delta \vdash (s^{1_{\Gamma'_0}} \langle q_1^{1_{\Gamma'_1}}, \dots, q_k^{1_{\Gamma'_k}} \rangle)^\theta : a} \sim \frac{\begin{array}{c} \pi_0 \\ \vdots \\ \zeta'_0 : \theta_0 : \Gamma'_0 \vdash s : \vec{a} \Rightarrow a \end{array} \quad \left(\begin{array}{c} \pi_i \\ \vdots \\ \zeta'_i : \theta_i : \Gamma'_i \vdash q_i : a_i \end{array} \right)_{i=1}^k \quad (\bigotimes_{j=0}^k 1_{\Gamma'_j}) \circ \theta : \zeta : \Delta \rightarrow \bigotimes_{j=0}^k \zeta'_j : \bigotimes_{j=0}^k \Gamma'_j}{\zeta : (\bigotimes_{j=0}^k \theta_j) \circ \theta : \Delta \vdash (s\vec{q})^{\bigotimes_{j=0}^k 1_{\Gamma'_j} \circ \theta} : a}$$

4. Subtyping-Aware Polyadic Calculus and Rigid Expansion of Ordinary λ -Terms –
4.2. The Calculus of Subtyping-Aware Polyadic Terms

Where we took $1_{\Gamma'_j} : \zeta_j : \Gamma'_j \rightarrow \zeta'_j : \Gamma'_j$. Then, from now on, we do not need to care too much about the choice that we made in the typing of an application, since all working choices are equivalent.

We prove that the basic operations on type derivations are stable under congruence.

Lemma 4.2.10. *Let $\pi \vdash_{\zeta:\Delta}^a \langle \eta, p \rangle$ and $\theta : \zeta' : \Delta' \rightarrow \zeta : \Delta, f : a \rightarrow a'$. The following statements hold.*

1. *If $\pi \sim \pi'$ then $[f]\pi \sim [f]\pi'$.*
2. *If $\pi \sim \pi'$ then $\pi\{\theta\} \sim \pi'\{\theta\}$.*

Proof. The two statements follow from a completely straightforward induction on π , applying Lemma 4.2.9. We prove the application case of the first point.

1. Let $\pi =$

$$\frac{\begin{array}{c} \pi_0\{\eta_0\} \\ \vdots \\ \zeta_0 : \theta_0 \circ \eta_0 : \Gamma_0 \vdash s^{\eta_0} : \vec{a} \Rightarrow a \end{array} \quad \left(\begin{array}{c} \pi_i\{\eta_i\} \\ \vdots \\ \zeta_i : \theta_i \circ \eta_i : \Gamma_i \vdash q_i^{\eta_i} : a_i \end{array} \right)_{i=1}^k \quad \theta : \zeta : \Delta \rightarrow \otimes_{j=0}^k \zeta_j : \otimes_{j=0}^k \Gamma_j}{\zeta : (\otimes_{j=0}^k \theta_j) \circ (\otimes_{j=0}^k \eta_j) \circ \theta : \Delta \vdash (s^{\eta_0} \langle q_1^{\eta_1}, \dots, q_k^{\eta_k} \rangle)^\theta : a}$$

Then by definition of left action and by Lemma 4.2.9 we have $[f]\pi =$

$$\frac{\begin{array}{c} ([1 \Rightarrow f]\pi_0)\{\eta_0\} \\ \vdots \\ \zeta_0 : \theta_0 \circ \eta_0 : \Gamma_0 \vdash s^{\eta_0} : \vec{a} \Rightarrow a' \end{array} \quad \left(\begin{array}{c} \pi_i\{\eta_i\} \\ \vdots \\ \zeta_i : \theta_i \circ \eta_i : \Gamma_i \vdash q_i^{\eta_i} : a_i \end{array} \right)_{i=1}^k \quad \theta : \zeta : \Delta \rightarrow \otimes_{j=0}^k \zeta_j : \otimes_{j=0}^k \Gamma_j}{\zeta : (\otimes_{j=0}^k \theta_j) \circ (\otimes_{j=0}^k \eta_j) \circ \theta : \Delta \vdash (s^{\eta_0} \langle q_1^{\eta_1}, \dots, q_k^{\eta_k} \rangle)^\theta : a'}$$

By definition of congruence we conclude, since $[f]\pi' =$

$$\frac{\begin{array}{c} [1 \Rightarrow f]\pi_0 \\ \vdots \\ \zeta'_0 : \theta_0 : \Gamma'_0 \vdash s : \vec{a} \Rightarrow a' \end{array} \quad \left(\begin{array}{c} \pi_i \\ \vdots \\ \zeta'_i : \theta_i : \Gamma'_i \vdash q_i : a_i \end{array} \right)_{i=1}^k \quad (\otimes_{j=0}^k 1_{\Gamma'_j}) \circ \theta : \zeta : \Delta \rightarrow \otimes_{j=0}^k \zeta'_j : \otimes_{j=0}^k \Gamma'_j}{\zeta : (\otimes_{j=0}^k \theta_j) \circ \theta : \Delta \vdash (s\vec{q})^{(\otimes_{j=0}^k 1_{\Gamma'_j})^\theta} : a'}$$

□

We now define a construction on type derivations that account for *weakening*.

Given $\pi \vdash_{\zeta:\Delta}^a \langle \eta, p \rangle$ we define $\pi \bullet_j \langle \rangle \vdash_{\zeta+\langle \rangle_j, \Delta+\langle \rangle_j}^a \langle \eta^{+\langle \rangle_j}, p \rangle$ for $j \in \text{len}(\zeta)$ by induction on π as follows:

$$\frac{\left(\frac{f_1 : \vec{b}_1 \rightarrow \langle \rangle, \dots, f_i : \vec{b}_i \rightarrow \langle a \rangle, \dots, f_n : \vec{b}_n \rightarrow \langle \rangle}{\vec{x}_1 : f_1 : \vec{b}_1, \dots, \vec{x}_i : f_i : \vec{b}_i, \dots, \vec{x}_n : f_n : \vec{b}_n \vdash x_{i, \alpha_i(1)} : a} \right) \bullet_j \langle \rangle =}{f_1 : \vec{b}_1 \rightarrow \langle \rangle, \dots, f_i : \vec{b}_i \rightarrow \langle a \rangle, \dots, 1_{\langle \rangle} : \langle \rangle \rightarrow \langle \rangle, \dots, f_n : \vec{b}_n \rightarrow \langle \rangle}{\vec{x}_1 : f_1 : \vec{b}_1, \dots, \vec{x}_i : f_i : \vec{b}_i, \dots, \langle \rangle : 1_{\langle \rangle} : \langle \rangle, \dots, \vec{x}_n : f_n : \vec{b}_n \vdash x_{i, \alpha_i(1)} : a}$$

4. Subtyping-Aware Polyadic Calculus and Rigid Expansion of Ordinary λ -Terms –
4.2. The Calculus of Subtyping-Aware Polyadic Terms

$$(\lambda \vec{x} : f.\pi') \bullet_j \langle \rangle = \lambda \vec{x} : f.(\pi' \bullet_j \langle \rangle)$$

$$((\pi_0 \langle \pi_1, \dots, \pi_k \rangle) \circ \eta) \bullet_j \langle \rangle = ((\pi_0 \bullet_j \langle \rangle \langle \pi_1 \bullet_j \langle \rangle, \dots, \pi_k \bullet_j \langle \rangle \rangle) \circ \eta^{(1 \circ)_j})$$

Given a list of type derivations $\vec{\rho} = \langle \rho_1, \dots, \rho_k \rangle$ we write $\vec{\rho} \bullet_j \langle \rangle$ meaning $\langle \rho_1 \bullet_j \langle \rangle, \dots, \rho_k \bullet_j \langle \rangle \rangle$. We use the same notations for the components of the conclusions of these derivations, *e.g.*, if $\text{mrp}(\vec{\rho}) = \vec{\theta} = \langle \theta_1, \dots, \theta_k \rangle$ then $\vec{\theta} \bullet_j \langle \rangle = \langle \theta_1^{\langle j \rangle}, \dots, \theta_k^{\langle j \rangle} \rangle$.

Proposition 4.2.11. *Let $\pi \vdash_{\zeta:\Delta}^a \langle \eta, p \rangle$ and $\pi' \vdash_{\zeta':\Delta'}^{a'} \langle \eta', p' \rangle$. The following statements hold.*

1. *If $\pi \sim \pi'$ then $\eta = \eta'$ and $p = p'$.*
2. *If $\eta = \eta'$ and $p = p'$ then $\Delta = \Delta'$ and $a = a'$. If moreover $\zeta = \zeta'$ then $\pi \sim \pi'$.*

Proof. The result depends on a fine-grained analysis of the congruence on polyadic type derivations. In particular, we introduce and use an appropriate notion of *canonical form* of type derivations wrt the congruence. The proof is detailed in Section B.2. \square

4.2.3. Substitution Operation on Type Derivations

We fix some notation that we will use in what follows. Let $\vec{q} = \langle q_1, \dots, q_k \rangle \in \Xi^!$ and let $\rho_i \vdash_{\zeta_i:\Delta_i}^{a_i} \langle \theta_i, q_i \rangle$. We set $\vec{\rho} = \langle \rho_1, \dots, \rho_k \rangle, \vec{\theta} = \langle \theta_1, \dots, \theta_k \rangle$. We write $\vec{\rho} \vdash_{\vec{\zeta}:\vec{\Delta}}^{\vec{a}} \langle \vec{\theta}, \vec{q} \rangle$. Given $f = \langle \alpha, f_1, \dots, f_{k'} \rangle : \vec{a} = \langle a_1, \dots, a_k \rangle \rightarrow \vec{a}' = \langle a'_1, \dots, a'_{k'} \rangle$, we extend the left action to lists of type derivations as follows $[f]\vec{\rho} = \langle [f_1]\rho_{\alpha(1)}, \dots, [f_{k'}]\rho_{\alpha(k')} \rangle$. We define in the same way $[f]\vec{\theta}, [f]\vec{q}$, that depend on $[f]\vec{\rho}$. In particular, by definition of left action, one has $[f_j]\rho_{\alpha(j)} \vdash_{\zeta_{\alpha(j)}:\Delta_{\alpha(j)}}^{a'_j} \langle [f_j]\theta_{\alpha(j)}, [f_j]q_{\alpha(j)} \rangle$ for $j \in [k']$. If $\vec{a}' = \langle a'_1, \dots, a'_{k'} \rangle = \bigoplus_{j=1}^l \vec{a}_j$ then we denote $\langle ([f]\rho)_1, \dots, ([f]\rho)_l \rangle$ the canonical decomposition of $[f]\rho$ that the former decomposition $\langle \vec{a}_1, \dots, \vec{a}_l \rangle$ of \vec{a}' determines.

If $f_i : \vec{b}_i \rightarrow \vec{a}_i$ for $i \in [k]$ and $\vec{\rho} \vdash_{\vec{\zeta}:\vec{\Delta}}^{\vec{a}} \langle \vec{\theta}, \vec{q} \rangle$ then $[\bigoplus_{i=1}^k f_i]\vec{\rho} = \bigoplus_{i=1}^k [f_i](\vec{\rho})_i$ where $(\vec{\rho})_i$ are the univocally determined list of derivations such that $(\vec{\rho})_i \vdash_{(\vec{\zeta})_i:(\vec{\Delta})_i}^{\vec{a}_i} \langle (\vec{\theta})_i, (\vec{q})_i \rangle$ with $\vec{\zeta} = \bigoplus_{i=1}^k (\vec{\zeta})_i, \vec{\Delta} = \bigoplus_{i=1}^k (\vec{\Delta})_i, \vec{\theta} = \bigoplus_{i=1}^k (\vec{\theta})_i$ and $\vec{q} = \bigoplus_{i=1}^k (\vec{q})_i$.

We also extend the definition of right action to list of type derivations. Given $\vec{\eta} = \langle \eta_1, \dots, \eta_k \rangle$ with $\eta_i : \zeta'_i : \Delta'_i \rightarrow \zeta_i : \Delta_i$ we set $\vec{\rho}\{\vec{\eta}\} = \langle \rho_1\{\eta_1\}, \dots, \rho_k\{\eta_k\} \rangle$.

Now we are ready to define the substitution operation on type derivation.

Definition 4.2.12. *Let $\pi \vdash_{\zeta \circ \vec{x}_i : \Delta \circ \vec{a}_i}^a \langle \eta \circ f_i, p \rangle$ and $\vec{\rho} \vdash_{\vec{\zeta}:\vec{\Gamma}}^{\vec{a}_i} \langle \vec{\theta}, \vec{q} \rangle$ for $j \in [\text{len}(\vec{a}_i)], i \in [\text{len}(\Delta)]$. We inductively define*

$$\pi[\vec{\rho}/\vec{x}_i] \vdash_{(\zeta^{-\vec{x}_i \otimes \vec{\zeta}}):\Delta^{-\vec{a}_i \otimes \vec{\Gamma}}}^a \langle \eta \otimes_p^{\vec{x}_i} \vec{\theta}, p[\vec{q}/\vec{x}_i]_{f_i} \rangle$$

as follows:

4. Subtyping-Aware Polyadic Calculus and Rigid Expansion of Ordinary λ -Terms –
4.2. The Calculus of Subtyping-Aware Polyadic Terms

– If

$$\pi = \frac{f_1 : \vec{a}_1 \rightarrow \langle \rangle, \dots, f_i = \langle \alpha_i, f \rangle : \vec{a}_i \rightarrow \langle a \rangle, \dots, f_n : \vec{a}_n \rightarrow \langle \rangle}{\vec{x}_1 : \mathbf{f}_1 : \vec{a}_1, \dots, \vec{x}_i : \mathbf{f}_i : \vec{a}_i, \dots, \vec{x}_n : \mathbf{f}_n : \vec{a}_n \vdash x_{i, \alpha_i(1)} : a}$$

then

$$\begin{aligned} \pi[\vec{\rho}/\vec{x}_i] &= ([f]\vec{\rho}_{\alpha_i(1)})\{\pi_{\alpha_i(1), \text{len}(\vec{a}_i)+1}\} \\ p[\vec{q}/\vec{x}_i]_{f_i} &= ([f]q_{\alpha_i(1)})^{\pi_{\alpha_i(1), \text{len}(\vec{a}_i)+1}} \\ \eta \otimes_p^{\vec{x}_i} \vec{\theta} &= ([f]\theta_{\alpha_i(1)}) \circ \pi_{\alpha_i(1), \text{len}(\vec{a}_i)+1} \end{aligned}$$

where the morphism $\pi_{\alpha(1), \text{len}(\vec{a})+1}$ is either the projection

$$\pi_{\alpha(1), \text{len}(\vec{a})+1} : \zeta^{-\vec{x}_i} \otimes \vec{\zeta} : \Delta^{-\vec{a}_i} \otimes \vec{\Gamma} \rightarrow \zeta_{\alpha(1)} : \Gamma_{\alpha(1)}$$

in the case where S is an irrelevant resource monad, or the identity.

– If

$$\pi = \frac{f_1 : \vec{a}_1 \rightarrow \langle \rangle, \dots, f_j = \langle \alpha_j, f \rangle : \vec{a}_j \rightarrow \langle a \rangle, \dots, f_n : \vec{a}_n \rightarrow \langle \rangle}{\vec{x}_1 : \mathbf{f}_1 : \langle \vec{a}_1 \rangle, \dots, \vec{x}_i : \mathbf{f}_j : \vec{a}_j, \dots, \vec{x}_n : \mathbf{f}_n : \vec{a}_n \vdash x_{j, \alpha_j(1)} : a}$$

with $j \neq i$ then

$$\begin{aligned} \pi[\vec{\rho}/\vec{x}_i] &= \pi^{-\vec{a}_j} \{\pi_{0, \text{len}(\vec{a}_i)+1}\} \\ p[\vec{q}/\vec{x}_i]_{f_i} &= x_{j, \alpha_j(1)} \\ \eta \otimes_p^{\vec{x}_i} \vec{\theta} &= \eta^{-f_i} \circ \pi_{0, \text{len}(\vec{a}_i)+1} \end{aligned}$$

where the morphism $\pi_{0, \text{len}(\vec{a})+1}$ is either the projection

$$\pi_{0, \text{len}(\vec{a})+1} : \zeta^{-\vec{x}_i} \otimes \vec{\zeta} : \Delta^{-\vec{a}_i} \otimes \vec{\Gamma} \rightarrow \zeta^{-\vec{x}_i} : \Delta^{-\vec{a}_i}$$

in the case where S is an irrelevant resource monad, or the identity.

– If $\pi = \lambda \vec{x} : \mathbf{g}.\pi'$, $\text{term}(\pi) = \lambda \vec{y} : \mathbf{g}.p'$ and $\text{mrp}(\pi) = \eta$ then

$$\begin{aligned} \lambda \vec{y} : \mathbf{g}.\pi'[\vec{\rho}/\vec{x}_i] &= \lambda \vec{x} : \mathbf{g}'.(\pi'[\vec{\rho} \bullet \langle \rangle / \vec{x}_i]) \\ \lambda \vec{y} : \mathbf{g}.p'[\vec{q}/\vec{x}_i]_{f_i} &= \lambda \vec{x} : \mathbf{g}'.(p'[\vec{q}/\vec{x}_i]_{f_i}) \\ \eta \otimes_p^{\vec{x}_i} \vec{\theta} &= ((\eta \oplus \langle g \rangle) \otimes_p^{\vec{x}_i} (\vec{\theta} \bullet \langle \rangle)) - \langle g' \rangle \end{aligned}$$

where we suppose that $\vec{y} \neq \vec{x}_i$ and

$$(\eta \oplus \langle g \rangle) \otimes_p^{\vec{x}_i} (\vec{\theta} \bullet \langle \rangle) = \eta' \oplus \langle g' \rangle$$

for some morphism η' .

– If $\pi = (\pi_0 \langle \pi_1, \dots, \pi_k \rangle) \circ \eta$, $p = (s(t_1, \dots, t_k))^\theta$ and $\text{mrp}(\pi) = (\otimes_{j=0}^k \eta_j^{\otimes f_{j,i}}) \circ \theta^{\otimes f'_{i,i}}$ then

$$(\pi_0 \langle \pi_1, \dots, \pi_k \rangle) \circ \eta[\vec{\rho}/\vec{x}_i] =$$

4. Subtyping-Aware Polyadic Calculus and Rigid Expansion of Ordinary λ -Terms –
4.2. The Calculus of Subtyping-Aware Polyadic Terms

$$\begin{aligned}
& (\pi_0([f'_i]\vec{\rho})_0/\vec{y}_0)\langle\pi_1([f'_i]\vec{\rho})_1/\vec{y}_1, \dots, \pi_k([f'_i]\vec{\rho})_k/\vec{y}_k\rangle) \circ \theta^* \\
& \quad (s(t_1, \dots, t_k))^\theta [\vec{q}/\vec{x}]_{f_i} = \\
& (s([f'_i]\vec{q})/\vec{y}_0)_{f'_{0,i}}\langle s([f'_i]\vec{q})_0/\vec{y}_0 \rangle_{f'_{0,i}}\langle t_1([f'_i]\vec{q})_1/\vec{y}_1 \rangle_{f'_{1,i}}, \dots, t_k([f'_i]\vec{q})_k/\vec{y}_k \rangle_{f'_{k,i}} \rangle^{\theta^*} \\
& \quad \eta \otimes_p^{\vec{x}_i} \vec{\theta} = \\
& \quad ((\eta_0 \otimes_s^{\vec{y}_0} ([f'_i]\vec{\theta})_0) \otimes \bigotimes_{l=1}^k (\eta_l \otimes_{t_l}^{\vec{y}_l} ([f'_i]\vec{\theta})_l)) \circ \theta^*
\end{aligned}$$

where $f_i = (\bigoplus_{j=0}^k f'_{j,i}) \circ f'_i$ and $\theta^* =$

$$\vec{\tau} \circ (\eta^{-f'_i} \otimes sm(f'_i)^*) :$$

$$\zeta^{-\vec{x}_i} \otimes \vec{\zeta} : \Delta^{-\vec{a}_i} \otimes \vec{\Gamma} \rightarrow \bigotimes_{j=0}^k (\zeta_j^{-\vec{y}_{j,i}} \otimes \vec{\zeta}_j \{sm(f'_i)\}) : \bigotimes_{j=0}^k (\text{source}(\eta_j)^{-\vec{a}_{j,i}} \otimes \vec{\Gamma}_j \{sm(f'_i)\}).$$

Where $\vec{\tau}$ is the permutation

$$\vec{\tau} : \bigotimes_{j=0}^k \text{source}(\eta_j)^{-\vec{a}_{j,i}} \otimes (\vec{\Gamma} \{sm(f'_i)\}) \rightarrow \bigotimes_{j=0}^k (\text{source}(\eta_j)^{-\vec{a}_{j,i}} \otimes (\vec{\Gamma} \{sm(f'_i)\})_j).$$

The definition given above is quite dense and for this reason it deserves some comments.

The idea behind the former definition derives from the fine-grained study of type derivations under reduction made in Chapter 3 (Remark 3.4.12). The intuition that the invertible 2-cell of Remark 3.4.12 describes a kind of substitution is then vindicated.

In the variable and application cases the left action on type derivations plays a central role: this is due to the fact that the type of a variable list is not symmetric, *i.e.*, it depends on its type morphism. One could say that the "true" type of a variable list is the target of its morphism, since it represents the typing of the occurrences of the list in the considered term. Hence, in order to perform the substitution we need a well-typed derivation list and we produce it by acting on $\vec{\rho}$ with the morphism of \vec{x}_i . In the application case this operation is quite intricate: the morphism of a variable list is given by composition between a concatenation of morphisms in the various components of the term and a new morphism.

It is worth noting that the substitution operation performs a concatenation of variable tuples and type tuples, while its action on *morphisms* and *terms* is far more complicated. Indeed, we will see that the former definition grants directly *subject expansion* and *subject reduction* for what concerns the standard typing, while it does not for the morphism typing. Intuitively this means that not only terms, but also *morphisms* are subject to the dynamics of reduction. This is a completely new phenomenon wrt standard resource calculi.

4. Subtyping-Aware Polyadic Calculus and Rigid Expansion of Ordinary λ -Terms –
4.2. The Calculus of Subtyping-Aware Polyadic Terms

For what concerns the abstraction case, the inductive hypothesis does not allow *per se* to prove that g' is actually the same of g . However, this is a corollary of the following lemma:

Lemma 4.2.13. *Let $\pi \vdash_{\zeta^{\circlearrowleft} \bar{x}_i : \Delta^{\circlearrowleft} \bar{a}_i \oplus \bar{a}}^a \langle \eta^{f_i, f_j}, p \rangle$ with $i \neq j$ and $\vec{\rho} \vdash_{\zeta^{\circlearrowleft} \bar{y}_j : \vec{\Gamma}^{\circlearrowleft} \bar{y}_j}^{\bar{a}_i} \langle \vec{\theta}^{\circlearrowleft} \bar{y}_j, \vec{q} \rangle$. Then the j -th component of $\text{mrp}(\pi[\vec{\rho}/\vec{x}_i])$ is f_j .*

Proof. By induction on the structure of π . The proof is detailed in Section B.1. \square

Then, as corollary, we can characterize the action of substitution in the term component of a type derivation completely by the linear substitution on polyadic terms:

Proposition 4.2.14. *Let $\pi \vdash_{\zeta^{\circlearrowleft} \bar{x}_i : \Delta^{\circlearrowleft} \bar{a}_i \oplus \bar{a}}^a \langle \eta^{f_i}, p \rangle$ and $\vec{\pi} \vdash_{\vec{\zeta} : \vec{\Gamma}}^{\bar{a}_i} \langle \vec{\theta}, \vec{q} \rangle$. We have $\text{term}(\pi[\vec{\rho}/\vec{x}_i]) = p\{[f_i]\vec{q}/\text{occ}_{\bar{x}_i}(p)\}$.*

Remark 4.2.15. *By the former proposition we have that*

$$\text{point}(\pi\{\vec{\rho}/\vec{x}_i\}) = \langle \text{mrp}(\pi) \otimes_{\text{term}(\pi)}^{\bar{x}_i} \text{mrp}(\vec{\rho}), \text{term}(\pi)\{[f_i]\text{term}(\vec{\rho})/\vec{x}\} \rangle.$$

Lemma 4.2.16. *Let $\pi \vdash_{\zeta^{\circlearrowleft} \bar{x}_i : \Delta^{\circlearrowleft} \bar{a}_i \oplus \bar{a}}^a \langle \eta^{f_i}, p \rangle$. The following statements hold.*

1. *Let $\vec{\rho} \vdash_{\vec{\zeta} : \vec{\Gamma}}^{\bar{a}'_i} \langle \vec{\theta}, \vec{q} \rangle$ and $\theta^{\circlearrowleft} g_i : \zeta_1^{\circlearrowleft} \bar{y}_i : \Delta_1^{\circlearrowleft} \bar{a}'_i \rightarrow \zeta^{\circlearrowleft} \bar{x}_i : \Delta^{\circlearrowleft} \bar{a}_i$. We have*

$$\pi\{\theta\}[\vec{\rho}/\vec{y}_i] = (\pi[[g_i]\vec{\rho}/\vec{x}_i])\{\theta^{\circlearrowleft -g_i} \otimes \text{sm}(g_i)^*\}.$$

2. *Let $\vec{\rho} \vdash_{\vec{\zeta} : \vec{\Gamma}}^{\bar{a}_i} \langle \vec{\theta}, \vec{q} \rangle$ and $\vec{\eta} = \langle \eta_1, \dots, \eta_{\text{len}(\bar{a}_i)} \rangle : \vec{\zeta} : \vec{\Gamma} \rightarrow \vec{\zeta}' : \vec{\Gamma}'$ we have*

$$\pi[\vec{\rho}\{\vec{\eta}\}/\vec{x}_i] \sim \pi[\vec{\rho}/\vec{x}_i]\{1_{\Delta} \otimes \bigotimes_{l=1}^k \eta_l\}.$$

3. *Let $\vec{\rho} \vdash_{\vec{\zeta} : \vec{\Gamma}}^{\bar{a}_i} \langle \vec{\theta}, \vec{q} \rangle$ and $f : a \rightarrow a'$ we have*

$$[f]\pi[\vec{\rho}/\vec{x}_i] = [f](\pi[\vec{\rho}/\vec{x}_i]).$$

Proof. By induction on the structure of π . The proof is detailed in Section B.2. \square

We are now ready to prove that the substitution operation is stable under congruence.

Proposition 4.2.17. *Let $\pi \vdash_{\zeta^{\circlearrowleft} \bar{x}_i : \Delta^{\circlearrowleft} \bar{a}_i \oplus \bar{a}}^a \langle \eta^{f_i}, p \rangle$, $\pi' \vdash_{\zeta^{\circlearrowleft} \bar{x}_i : \Delta^{\circlearrowleft} \bar{a}_i \oplus \bar{a}}^a \langle \eta^{f_i}, p \rangle$ such that $\pi \sim \pi'$ and $\vec{\rho} \vdash_{\vec{\zeta}^{\circlearrowleft} : \vec{\Gamma}^{\circlearrowleft}}^{\bar{a}_i} \langle \vec{\theta}^{\circlearrowleft}, \vec{q} \rangle$, $\vec{\rho}' \vdash_{\vec{\zeta}'^{\circlearrowleft} : \vec{\Gamma}'^{\circlearrowleft}}^{\bar{a}_i} \langle \vec{\theta}'^{\circlearrowleft}, \vec{q}' \rangle$ such that $\vec{\rho} \sim \vec{\rho}'$. Then $\pi[\vec{\rho}/\vec{x}_i] \sim \pi'[\vec{\rho}'/\vec{x}_i]$.*

Proof. By induction on the structure of π . The proof is detailed in Section B.3. \square

4. Subtyping-Aware Polyadic Calculus and Rigid Expansion of Ordinary λ -Terms –
 4.2. The Calculus of Subtyping-Aware Polyadic Terms

Reduction of Type Derivations We define a reduction relation on type derivations $\rightarrow_d \subseteq \mathcal{TD}^2$ such that if $\pi \in TD_\zeta(\Delta, a)$ and $\pi \rightarrow_d \pi'$ then $\pi' \in TD_\zeta(\Delta, a)$ inductively as follows:

1. Let $\pi =$

$$\frac{\frac{\begin{array}{c} \pi_0 \\ \vdots \\ \zeta_0 \oplus \langle \vec{x} \rangle : \eta_0 \oplus \langle f \rangle : \Gamma_0 \oplus \langle \vec{a} \rangle \vdash s : a \end{array}}{\zeta_0 : \eta_0 : \Gamma_0 \vdash \lambda \vec{x} : f.s : \vec{a} \Rightarrow a} \quad \left(\begin{array}{c} \pi_l \\ \vdots \\ \zeta_l : \eta_l : \Gamma_l \vdash t_l : a_l \end{array} \right)_{l=1}^k \quad \eta : \zeta : \Delta \rightarrow \bigotimes_{l=0}^k \zeta_l : \bigotimes_{l=0}^k \Gamma_l}{\zeta : (\bigotimes_{l=0}^k \eta_j) \circ \eta : \Delta \vdash (\lambda \vec{x} : f.s \langle t_1, \dots, t_k \rangle)^n}$$

Then $\pi \rightarrow_d \pi_0[\langle \pi_1, \dots, \pi_k \rangle / \langle \vec{x}_i \rangle \{ \eta \}]$.

2. If $\pi \vdash_{\zeta \oplus \langle \vec{x} \rangle : \Delta \oplus \langle \vec{a} \rangle}^a \langle \eta \oplus \langle f \rangle, p \rangle$ and $\pi' \vdash_{\zeta \oplus \langle \vec{x} \rangle : \Delta \oplus \langle \vec{a} \rangle}^a \langle \eta' \oplus \langle f' \rangle, p' \rangle$ such that $\pi \rightarrow_d \pi'$ then $\lambda \vec{x} : f.\pi \rightarrow_d \lambda \vec{x} : f'.\pi'$.
3. If $\pi_0 \vdash_{\zeta_0 : \Gamma_0}^{\vec{a} \Rightarrow a} \langle \eta_0, s \rangle$, $\pi'_0 \vdash_{\zeta'_0 : \Gamma'_0}^{\vec{a} \Rightarrow a} \langle \eta'_0, s' \rangle$ such that $\pi_0 \rightarrow_d \pi'_0$ and $\pi_i \vdash_{\zeta_i : \Gamma_i}^{a_i} \langle \eta_i, t_i \rangle$, $\pi'_i \vdash_{\zeta'_i : \Gamma'_i}^{a_i} \langle \eta'_i, t'_i \rangle$ for $i \in [k]$ such that $\pi_i \rightarrow_d \pi'_i$ with $\eta : \zeta : \Delta \rightarrow \bigotimes_{j=0}^k \Gamma_j$ then

$$\pi_0 \langle \pi_1, \dots, \pi_k \rangle \rightarrow_d \pi'_0 \langle \pi'_1, \dots, \pi'_k \rangle.$$

The reduction \rightarrow_d univocally determines a family of reductions $\rightarrow_d^{\zeta : \Delta, a} \subseteq TD_\zeta(\Delta, a)$ in the natural way.

There is an unusual phenomenon in the contextual abstraction case. The *morphism* typing changes under reduction. This can appear surprising at first, but it is actually a direct consequence to the fact that not only term, but also morphisms represent the dynamic content of a type derivation.

Example 4.2.18. We give some examples of type derivations reduction.

— Let $b = \langle a, a \rangle \rightarrow a$ and $\pi =$

$$\frac{\frac{1}{\langle x \rangle : \langle b \rangle \vdash x : b} \quad \frac{\pi_1 : \langle a, a \rangle \rightarrow \langle a \rangle}{\langle \rangle, \langle z_1, z_2 \rangle : \pi_1 : \langle a, a \rangle \vdash z_1 : a} \quad \frac{1}{\langle \rangle, \langle z_3 \rangle : \langle a \rangle \vdash z_3 : a} \quad \langle 1, c_a^3 \rangle}{\frac{\langle x \rangle : \langle b \rangle, \langle y \rangle : (\pi_1 \oplus 1_a) \circ c_a^3 : \langle a \rangle \vdash x \langle y, y \rangle : a}{\langle x \rangle : \langle b \rangle \vdash \lambda \langle y \rangle : (\pi_1 \oplus 1_a) \circ c_a^3.x \langle y, y \rangle : \langle a \rangle \Rightarrow a}}$$

where $\langle 1, c_a^3 \rangle : \langle \langle x \rangle, \langle y \rangle \rangle : \langle \langle b \rangle, \langle a \rangle \rangle \rightarrow \langle \langle x \rangle, \langle z_1, z_2, z_3 \rangle \rangle : \langle \langle b \rangle, \langle a, a, a \rangle \rangle$. Now consider $\pi' =$

$$\frac{\frac{\pi \bullet \langle \rangle}{\vdots} \quad \frac{1}{\langle \rangle, \langle w \rangle : \langle a \rangle} \quad 1}{\langle x \rangle : \langle b \rangle, \langle w \rangle : \langle a \rangle \vdash (\lambda \langle y \rangle : (\pi_1 \oplus 1_a) \circ c_a^3.x \langle y, y \rangle) \langle w \rangle : a}$$

we have that $\pi' \rightarrow_d \pi''$ where $\pi'' =$

4. Subtyping-Aware Polyadic Calculus and Rigid Expansion of Ordinary λ -Terms –
4.2. The Calculus of Subtyping-Aware Polyadic Terms

$$\frac{\frac{1}{\langle x \rangle : \langle b \rangle \vdash x : b} \quad \frac{\pi_1 : \langle a, a \rangle \rightarrow \langle a \rangle}{\langle \rangle, \langle w, z_2 \rangle : \pi_1 : \langle a, a \rangle \vdash w : a} \quad \frac{1}{\langle \rangle, \langle w \rangle : \langle a \rangle \vdash w : a}}{\langle x \rangle : \langle b \rangle, \langle w \rangle : (\pi_1 \oplus 1_a) \circ c_a^3 : \langle a \rangle \vdash x \langle w, w \rangle : a} \quad \langle 1, c_a^3 \rangle$$

where $\langle 1, c_a^3 \rangle : \langle \langle x \rangle, \langle w \rangle \rangle : \langle \langle b \rangle, \langle a \rangle \rangle \rightarrow \langle \langle x \rangle, \langle w, z_2, w \rangle \rangle : \langle \langle b \rangle, \langle a, a, a \rangle \rangle$ and we used the fact that $1_a \oplus \diamond_b = \pi_1 : \langle a, b \rangle \rightarrow \langle a \rangle$. Hence the morphism typing does not enjoy subject reduction, as expected.

- Just take π' and π'' from the former example. Consider $\lambda \langle w \rangle : 1_a \cdot \pi'$ and $\lambda \langle w \rangle : (\pi_1 \oplus 1_a) \circ c_a^3 \cdot \pi''$. We have by contextuality that $\lambda \langle w \rangle : 1_a \cdot \pi' \rightarrow_d \lambda \langle w \rangle : (\pi_1 \oplus 1_a) \circ c_a^3 \cdot \pi''$.

Lemma 4.2.19. Let $\pi \vdash_{\zeta : \Delta}^a \langle \eta, p \rangle$ with $\eta : \zeta' : \Delta' \rightarrow \zeta : \Delta$ and $f : a \rightarrow a'$. The following statements hold.

- If $\pi \rightarrow_d \pi'$ then $\pi \{ \eta \} \rightarrow_d \pi' \{ \eta \}$.
- If $\pi \rightarrow_d \pi'$ then $[f] \pi \rightarrow_d [f] \pi'$.

Proof. The two results follow from a completely straightforward induction on the reduction step, applying Lemma 4.2.16. \square

4.2.4. Points of Type Derivations

Proposition 4.2.11 gives an important characterization of equivalent type derivations that shares the same context: two derivations π and π' are equivalent iff their morphism and term is the same. This fact says that the quotient $TD_\zeta(\Delta, a) / \sim$ is in bijection with the set $\Omega_\zeta(\Delta, a) = \{ \langle \eta, p \rangle \mid \text{for some } \pi \in TD_\zeta(\Delta, a), \pi \vdash_{\zeta : \Delta}^a \langle \eta, p \rangle \}$. For this reason, we will now study couples of morphisms and terms, that we call the *points of type derivations* and see them as equivalence classes of type derivation.

Given $\langle \zeta : \Delta, a \rangle \in (SD^n)^* \times D$ with $n \in \mathbb{N}$ and $\zeta = \langle \vec{x}_1, \dots, \vec{x}_n \rangle, \Delta = \langle \vec{a}_1, \dots, \vec{a}_n \rangle$ we define the *set of points* along $\langle \zeta : \Delta, a \rangle, \Omega_\zeta(\Delta, a)$ by induction as follows:

- If $\eta = \langle f_1, \dots, f_i = \langle \alpha_i, f \rangle, \dots, f_n \rangle : \zeta : \langle \vec{a}_1, \dots, \vec{a}_i, \dots, \vec{a}_n \rangle \rightarrow \langle \langle \rangle, \dots, \langle a \rangle, \dots, \langle \rangle \rangle$ then $\langle \eta, x_{i, \alpha_i(1)} \rangle \in \Omega_\zeta(\Delta, a)$.
- If $\langle \eta \oplus \langle f \rangle, p \rangle \in \Omega_{\zeta \oplus \langle \vec{x} \rangle}(\Delta \oplus \langle \vec{a} \rangle, a)$ then $\langle \eta, \lambda \vec{x} : f \cdot p \rangle \in \Omega_\zeta(\Delta, \vec{a} \Rightarrow a)$.
- If $\langle \eta_0, s \rangle \in \Omega_{\zeta_0}(\Gamma_0, \langle a_1, \dots, a_k \rangle \Rightarrow a)$, $\langle \eta_i, t_i \rangle \in \Omega_{\zeta_i}(\Gamma_i, a_i)$ for $i \in [k]$ and $\eta : \zeta : \Delta \rightarrow \otimes_{j=0}^k \zeta_j : \otimes_{j=1}^k \Gamma_j$ then $\langle (\otimes_{j=1}^k \eta_j) \circ \eta, (s \langle t_1, \dots, t_k \rangle)^n \rangle \in \Omega_\zeta(\Delta, a)$.

Let $\varphi = \langle \eta, p \rangle$ be a point. We say that φ is a variable (resp. an abstraction an application) when p is.

We range over points with Greek letters as φ, ψ, \dots . Let $\vec{\varphi} = \langle \varphi_1, \dots, \varphi_k \rangle, \vec{\Gamma} = \langle \Gamma_1, \dots, \Gamma_k \rangle, \vec{a} = \langle a_1, \dots, a_k \rangle, \vec{\zeta} = \langle \zeta_1, \dots, \zeta_k \rangle$ such that $\varphi_i \in \Omega_{\zeta_i}(\Gamma_i, a_i)$. We then write $\vec{\varphi} \in \Omega_{\vec{\zeta}}(\vec{\Gamma}, \vec{a})$. Trivially we have

$$\Omega_\zeta(\Delta, a) = \{ \langle \eta, p \rangle \mid \text{there exists a type derivation } \pi \text{ s.t. } \pi \vdash_{\zeta : \Delta}^a \langle \eta, p \rangle \}.$$

4. Subtyping-Aware Polyadic Calculus and Rigid Expansion of Ordinary λ -Terms –
 4.2. The Calculus of Subtyping-Aware Polyadic Terms

We define the *set of points*

$$\Omega = \bigcup_{n \in \mathbb{N}} \bigcup_{\langle \zeta : \Delta, a \rangle \in (SD^n)^* \times D} \Omega_\zeta(\Delta, a).$$

Given $\varphi = \langle \eta, p \rangle \in \Omega_\zeta(\Delta, a)$ with $\zeta = \langle \vec{z}_1, \dots, \vec{z}_n \rangle$. Let $x_{i,j} \in \vec{z}_i$ for some $i \in [n]$ and $j \in [\text{len}(\vec{z}_i)]$. We say that the variable $x_{i,j}$ is *relevant* (resp. *irrelevant*) for φ if $x_{i,j} \in \text{fv}(p)$ (resp. $x_{i,j} \notin \text{fv}(p)$). We extend this definition to variable lists in the natural way.

For a point $\varphi = \langle \eta, p \rangle$, we set $\text{mrp}(\varphi) = \eta$ and $\text{term}(\varphi) = p$. For a list of points $\vec{\psi} = \langle \psi_1, \dots, \psi_k \rangle$ with $\psi_i = \langle \eta_i, p_i \rangle$, we set $\text{mrp}(\vec{\psi}) = \langle \eta_1, \dots, \eta_k \rangle$ and $\text{source}(\text{mrp}(\vec{\psi})) = \langle \text{source}(\eta_1), \dots, \text{source}(\eta_k) \rangle$. We define some syntactic sugar for points as follows:

$$\lambda \vec{x} : f. \langle \eta \oplus \langle f \rangle, p \rangle = \langle \eta, \lambda \vec{x} : f.p \rangle \quad \langle \eta_0, s \rangle \langle \langle \eta_i, t_i \rangle \rangle_{i=1}^k = \langle \bigotimes_{i=0}^k \eta_j, s \langle t_1, \dots, t_k \rangle \rangle$$

$$\langle \eta, p \rangle \circ \theta = \langle \eta \circ \theta, p^\theta \rangle$$

Where $\langle \eta \oplus \langle f \rangle, p \rangle \in \Omega_{\zeta \oplus \langle \vec{x} \rangle}(\Delta \oplus \langle \vec{a} \rangle, a)$, $\langle \eta_0, s \rangle \in \Omega_\zeta(\Gamma_0, \langle a_1, \dots, a_k \rangle \Rightarrow a)$, $\langle \eta_i, t_i \rangle \in \Omega_\zeta(\Gamma_0, a_i)$ with $\text{len}(\zeta_i) = \text{len}(\zeta_j)$ for $i \neq j$ and $\text{trg}(\eta) = \text{source}(\theta)$. Intuitively, points represent all the dynamical information of derivations.

Lemma 4.2.20. *Let $\varphi = \langle \eta, p \rangle \in \Omega_\zeta(\Delta, a) \cap \Omega_{\zeta'}(\Delta, a)$. Then $\text{occ}_\zeta(p) = \text{occ}_{\zeta'}(p)$.*

Definition 4.2.21. *Let $\varphi \in \Omega_\zeta(\Delta, a)$. We define the set $\text{RP}(\varphi)(\zeta : \Delta, a)$ of representations of φ along $\zeta : \Delta, a$ by induction on φ as follows:*

- If $\varphi = \langle \eta \stackrel{\circ}{\eta} f_i, x_{i, \text{sm}(f_i)(1)} \rangle$ then $\text{RP}(\varphi)(\zeta : \Delta, a) = \{ \langle \zeta, \langle \eta \stackrel{\circ}{\eta} f_i, x_{i, \text{sm}(f_i)(1)} \rangle \}$.
- If $\varphi = \lambda \vec{x} : f. \varphi'$ then $\text{RP}(\varphi)(\zeta : \Delta, a) = \{ \langle \zeta, \lambda \vec{x} : f. \alpha \rangle, \text{ s.t. } \alpha \in \text{RP}(\varphi)(\zeta \oplus \langle \vec{x} \rangle : \Delta \oplus \langle \text{source}(f) \rangle, a) \}$.
- If $\varphi = (\psi_0 \langle \psi_1, \dots, \psi_k \rangle) \circ \eta$ then $\text{RP}(\varphi)(\zeta : \Delta, a) = \{ \langle \zeta, \langle \alpha, \langle \beta_1, \dots, \beta_k \rangle, \theta \rangle \rangle \mid \alpha \in \text{RP}(\psi_0)(\zeta_0 : \Gamma_0, \vec{a} \Rightarrow a), \beta_i \in \text{RP}(\psi'_i)(\zeta_i : \Gamma_i, a_i), \theta : \Delta \rightarrow \bigotimes \Gamma_j, \text{ for some } \psi'_j \in \Omega, \Gamma_j \in (SD^{\text{len}(\zeta)})^*, \vec{a} \in SD \text{ s.t. } (\psi'_0 \langle \psi'_1, \dots, \psi'_k \rangle) \circ \theta = (\psi_0 \langle \psi_1, \dots, \psi_k \rangle) \circ \eta \}$.

Given a representation $\alpha \in \text{RP}(\varphi)(\zeta : \Delta, a)$ we associate to it its point by induction as follows:

$$\text{point}(\langle \zeta, \langle \eta \stackrel{\circ}{\eta} f_i, x_{i, \text{sm}(f_i)(1)} \rangle \rangle) = \langle \eta \stackrel{\circ}{\eta} f_i, x_{i, \text{sm}(f_i)(1)} \rangle$$

$$\text{point}(\langle \zeta, \lambda \vec{x} : f. \alpha \rangle) = \lambda \vec{x} : f. \text{point}(\alpha)$$

$$\text{point}(\langle \zeta, \langle \alpha, \vec{\beta}, \eta \rangle \rangle) = (\text{point}(\alpha) \text{point}(\vec{\beta})) \circ \eta$$

Trivially, if $\alpha \in \text{RP}(\varphi)(\zeta : \Delta, a)$ then $\text{point}(\alpha) = \varphi$. Each representation of a point φ corresponds to a type derivation $\pi \in \mathcal{TD}$ such that $\text{point}(\pi) = \varphi$ in a natural way. Indeed, the former definition can be seen as just a formalization of the fact

4. Subtyping-Aware Polyadic Calculus and Rigid Expansion of Ordinary λ -Terms –
4.2. The Calculus of Subtyping-Aware Polyadic Terms

that points are equivalence classes of type derivations (Proposition 4.2.11). When we write a point we are implicitly choosing a representation of it.

Given an application point $(\varphi\langle\psi_1, \dots, \psi_k\rangle) \circ \eta \in \Omega_{\zeta \triangleright \vec{x}_i}(\Delta \triangleright \vec{a}_i, a)$ such that $\varphi \in \Omega_{\zeta_0 \triangleright \vec{x}_{0,i}}(\Gamma_0 \triangleright \vec{a}_{0,i}, \langle b_1, \dots, b_k \rangle \Rightarrow b)$, $\psi_l \in \Omega_{\zeta_l \triangleright \vec{x}_{l,i}}(\Gamma_l \triangleright \vec{a}_{l,i}, b_l)$, $\eta : \Delta \rightarrow \bigotimes_{j=0}^k \Gamma_j$ for $l \in [k]$ and some $n \in \mathbb{N}$, $\zeta : \Delta, \zeta_j \Gamma_j \in (SD^n)^*$, $\langle b_1, \dots, b_k \rangle \Rightarrow b \in D$, we say that $\vec{x}_{l,i}$ (resp. $\vec{a}_{l,i}$) for all $l \in [k]$, $i \in [n]$ is an *antecedent* of \vec{x}_i (resp. \vec{a}_i). Clearly, each representation of a point gives rise to a family of antecedents for \vec{x}_i, \vec{a}_i .

Example 4.2.22. *We present some examples of representations for points. Each representation corresponds to an element of the equivalence class of type derivations associated to the point.*

1. Let $\varphi = \langle\langle\sigma, g, f, 1_{\langle a, a \rangle \Rightarrow a}\rangle\rangle, x_3 \langle x_2, x_1 \rangle$ where $f : b \rightarrow a, g : c \rightarrow a$ where σ is the permutation $(3, 2, 1)$. We consider $\varphi \in \Omega_{\langle\langle x_1, x_2, x_3 \rangle\rangle}(\langle\langle c, b, \langle a, a \rangle \Rightarrow a \rangle\rangle, a)$. Now we have the possible following representation of φ : $\varphi' = \langle\langle 1_{\langle a, a \rangle \Rightarrow a}, z_1 \rangle \in \Omega_{\langle\langle z_1 \rangle\rangle}(\langle\langle a, a \rangle \Rightarrow a \rangle, \langle a, a \rangle \Rightarrow a), \psi_1 = \langle\langle f \rangle, z_2 \rangle \in \Omega_{\langle\langle z_2 \rangle\rangle}(\langle\langle b \rangle\rangle, a), \psi_2 = \langle\langle g \rangle, z_3 \rangle \in \Omega_{\langle\langle z_3 \rangle\rangle}(\langle\langle c \rangle\rangle, a)$ and $\eta = \langle\langle \sigma \rangle\rangle : \langle\langle x_1, x_2, x_3 \rangle\rangle : \langle\langle c, b, \langle a, a \rangle \Rightarrow a \rangle\rangle \rightarrow \langle\langle z_1, z_2, z_3 \rangle\rangle : \langle\langle \langle a, a \rangle \Rightarrow a, b, c \rangle\rangle$. Another possible representation is then the following: $\varphi' = \langle\langle 1_{\langle a, a \rangle \Rightarrow a}, z_1 \rangle \in \Omega_{\langle\langle z_1 \rangle\rangle}(\langle\langle a, a \rangle \Rightarrow a \rangle, \langle a, a \rangle \Rightarrow a), \psi_1 = \langle\langle 1_a \rangle, z_2 \rangle \in \Omega_{\langle\langle z_2 \rangle\rangle}(\langle\langle a \rangle\rangle, a), \psi_2 = \langle\langle 1_b \rangle, z_3 \rangle \in \Omega_{\langle\langle z_3 \rangle\rangle}(\langle\langle c \rangle\rangle, a)$ and $\eta = \langle\langle \sigma, g, f, 1 \rangle\rangle : \langle\langle x_1, x_2, x_3 \rangle\rangle : \langle\langle c, b, \langle a, a \rangle \Rightarrow a \rangle\rangle \rightarrow \langle\langle z_1, z_2, z_3 \rangle\rangle : \langle\langle \langle a, a \rangle \Rightarrow a, a, a \rangle\rangle$.
2. Let $\varphi = \langle\langle 1_{\langle b \rangle}, c_{\langle a \rangle} \rangle, x \langle y, y \rangle$ where $b = \langle a, a \rangle \Rightarrow a$ and we recall that $c_{\langle a \rangle} : \langle a \rangle \rightarrow \langle a, a \rangle$ is the diagonal. A representation of φ is then: $\varphi' = \langle\langle 1_{\langle b \rangle} \rangle, x \rangle \in \Omega_{\langle\langle x \rangle\rangle}(\langle\langle b \rangle\rangle, b), \psi_1 = \langle\langle 1_a \rangle, z \rangle, \psi_2 = \langle\langle 1_a \rangle, z \rangle \in \Omega_{\langle\langle z \rangle\rangle}(\langle\langle a \rangle\rangle, a)$ and $\eta = \langle\langle 1_{\langle b \rangle}, \langle c_a \rangle \rangle : \langle\langle x \rangle, \langle y \rangle \rangle : \langle\langle b \rangle, \langle a \rangle \rangle \rightarrow \langle\langle x \rangle, \langle z, z \rangle \rangle : \langle\langle b \rangle, \langle a, a \rangle \rangle$.
3. Consider the point $\varphi = \langle\langle \langle \vec{a} \rangle, 1_{\langle a \rangle} \rangle, x \rangle$. This point is a weakening point. Points cannot detect the non relevant variables, i.e., $\varphi \in \Omega_{\langle\langle \vec{z}, \langle x \rangle \rangle\rangle}(\langle\langle \vec{a} \rangle, \langle a \rangle \rangle, a) \cap \Omega_{\langle\langle \vec{y}, \langle x \rangle \rangle\rangle}(\langle\langle \vec{a} \rangle, \langle a \rangle \rangle, a)$ for any repetitions-free \vec{x}, \vec{y} of the same length as \vec{a} .

Thanks to Proposition 4.2.11, we can extend all the operations on type derivations that are stable under congruence to operations on points. This passage in general simplifies the technicalities, since points are just equivalence classes of type derivations. In particular, for the right action we get a much more concise definition: given $\eta : \zeta' : \Delta' \rightarrow \zeta : \Delta, \langle \theta, p \rangle \{ \eta \} = \langle \theta \circ \eta, p \eta \rangle$. For $g : a \rightarrow a'$ and $\langle \eta, p \rangle \in \Omega_{\zeta}(\Delta, a)$ The definition of left action $[g] \langle \eta, p \rangle = \langle [g] \eta, [g] p \rangle \in \Omega_{\zeta}(\Delta, a')$ is given in Figure 4.2.4. Trivially we have that if $\text{point}(\pi) = \langle \eta, p \rangle$ then $\text{point}([g] \pi) = [g] \langle \eta, p \rangle$. The same is true for the right action.

We extend the actions to lists of points. Let $\vec{\varphi} = \langle \varphi_1, \dots, \varphi_k \rangle \in \Omega_{\vec{\zeta}}(\vec{\Gamma}, \vec{a})$ and $f = \langle \alpha, f_1, \dots, f_{\text{len}(\vec{a}')} \rangle : \vec{a} \rightarrow \vec{a}'$. We set $[f] \vec{\varphi} = \langle [f_1] \varphi_{\alpha(1)}, \dots, [f_{\text{len}(\vec{a}')}] \varphi_{\alpha(\text{len}(\vec{a}'))} \rangle$. Let $\vec{\eta} = \langle \eta_1, \dots, \eta_k \rangle$, then $\vec{\varphi} \{ \vec{\eta} \} = \langle \varphi_1 \{ \eta_1 \}, \dots, \varphi_k \{ \eta_k \} \rangle$.

We fix some notations. Let $\vec{\psi} = \langle \psi_1, \dots, \psi_k \rangle, \vec{\zeta} = \langle \zeta_1, \dots, \zeta_k \rangle, \vec{\Gamma} = \langle \Gamma_1, \dots, \Gamma_k \rangle$ with $\psi_i \in \Omega_{\zeta_i}(\Gamma_i, a_i)$ for some $\zeta_i : \Gamma_i, a_i$. We set $\vec{\psi} \in \Omega_{\vec{\zeta}}(\vec{\Gamma}, \vec{a})$. Let $f = \langle \alpha, f_1, \dots, f'_k \rangle : \vec{a} = \langle a_1, \dots, a_k \rangle \rightarrow \vec{a}' = \langle a'_1, \dots, a'_{k'} \rangle$. We extend the definition of left action to list

4. Subtyping-Aware Polyadic Calculus and Rigid Expansion of Ordinary λ -Terms –
4.2. The Calculus of Subtyping-Aware Polyadic Terms

$$\begin{aligned} [g]\langle\langle f_1, \dots, f_i = \langle \alpha_i, f \rangle, \dots, f_n \rangle, x_{i, \alpha_i(i)} \rangle &= \langle\langle f_1, \dots, f_i = \langle \alpha_i, f \circ g \rangle, \dots, f_n \rangle, x_{i, \alpha_i(i)} \rangle \\ \langle\langle \beta, \vec{g} \rangle \Rightarrow g \rangle(\lambda \vec{x} : f \cdot \varphi) = \lambda \vec{y} : ([g]f) \circ \langle \beta, \vec{g} \rangle. &(([\varphi])\{1_\Delta \oplus \langle \langle \beta, \vec{g} \rangle \rangle\}) \\ [g](\langle \varphi \langle \psi_1, \dots, \psi_k \rangle \rangle \circ \eta) &= (([1 \Rightarrow g]\varphi)\langle \psi_1, \dots, \psi_k \rangle) \circ \eta \end{aligned}$$

Where we suppose that the points are well-defined and in the abstraction case $\text{len}(\vec{y}) = \text{len}(\text{source}(\langle \beta, \vec{g} \rangle))$.

Figure 4.3. – Left action on points.

of points as follows: $[f]\vec{\psi} = \langle [f_1]\psi_{\alpha(1)}, \dots, [f_{k'}]\psi_{\alpha(k')} \rangle \in \Omega_{\vec{\zeta}\{\text{sm}(f)\}}(\vec{\Gamma}\{\text{sm}(f)\}, \vec{a}')$. If $\vec{a}' = \bigoplus_{i=1}^l \vec{a}_i$ we denote as

$$\langle ([f]\vec{\psi})_1, \dots, ([f]\vec{\psi})_{l'} \rangle, \langle (\vec{\zeta}\{\text{sm}(f)\})_1, \dots, (\vec{\zeta}\{\text{sm}(f)\})_{l'} \rangle, \langle (\vec{\Gamma}\{\text{sm}(f)\})_1, \dots, (\vec{\Gamma}\{\text{sm}(f)\})_{l'} \rangle$$

the canonical partitions induced by the former partition of \vec{a}' . This means that $([f]\vec{\psi})_i \in \Omega_{(\vec{\zeta}\{\text{sm}(f)\})_i}((\vec{\Gamma}\{\text{sm}(f)\})_i, \vec{a}_i)$.

We define a *substitution* operation on points as follows:

Definition 4.2.23. Let $\varphi \in \Omega_{\zeta \ni \vec{x}_i}(\Gamma_0^{\vec{a}_i}, a)$ with $\vec{a}_i = \langle a_{i,1}, \dots, a_{i,k_i} \rangle$ and $\psi_l \in \Omega_{\zeta_i}(\Gamma_i, a_{i,j})$ with $l \in [k_i]$ and $\text{len}(\zeta) = \text{len}(\zeta_i)$. We set $\langle \psi_1, \dots, \psi_{k_i} \rangle = \vec{\psi}$. We define the substitution $\varphi[\vec{\psi}/\vec{x}_i] \in \Omega_{\zeta}(\bigotimes_{j=0}^{k_i} \Gamma_j, a)$ of \vec{x}_i in the point φ by $\vec{\psi}$ by induction on φ as follows:

$$\begin{aligned} \langle \eta^{f_i = \langle \alpha_i, f \rangle}, x_{i, \alpha_i(1)} \rangle [\vec{\psi}/\vec{x}_i] &= ([f]\psi_{\alpha_i(1)})\{\pi_{\alpha_i(1), \text{len}(\vec{a}_i)+1}\} \\ \langle \eta^{\triangleright f_j}, x_{j, \alpha_j(1)} \rangle [\vec{\psi}/\vec{x}_i] &= \langle \eta^{-f_j}, x_{j, \alpha_j(1)} \rangle \{\pi_{0, \text{len}(\vec{a}_i)+1}\} \\ \lambda \vec{y} : g \cdot \varphi[\vec{\psi}/\vec{x}_i] &= \lambda \vec{y} : g' \cdot \varphi[\vec{\psi} \bullet \langle \rangle / \vec{x}_i] \\ (\varphi_0 \langle \varphi_1, \dots, \varphi_k \rangle) \circ \eta^{\triangleright f'_i} [\vec{\psi}/\vec{x}_i] &= \\ (\varphi_0([\varphi'_i]\vec{\psi})_0 / \vec{y}_0) \langle \varphi_1([\varphi'_i]\vec{\psi})_1 / \vec{y}_1, \dots, \varphi_k([\varphi'_i]\vec{\psi})_k / \vec{y}_k \rangle \rangle & \circ (\vec{\tau} \circ ((\eta^{-\langle f'_i \rangle}) \otimes \text{sm}(f'_i)^*)). \end{aligned}$$

Where $i \neq j$ in the variable cases.

In the abstraction case, $\vec{y} \neq \vec{x}_i$ and g' is the morphism assigned to the variable list \vec{y} after performing the substitution.

In the application case we have $\vec{\tau} : (\bigotimes_{j=0}^k \text{supp}(\varphi_j)) \otimes ((\bigotimes_{l=1}^{k_i} \zeta_l) = \vec{\zeta}\{\text{sm}(f'_i)\}) : (\bigotimes_{j=0}^k \text{source}(\text{mrrp}(\varphi_j))) \otimes ((\bigotimes_{l=1}^{k_i} \Gamma_l) = \vec{\Gamma}\{\text{sm}(f'_i)\}) \rightarrow \bigotimes_{j=0}^k (\text{supp}(\varphi_j) \otimes (\vec{\zeta}\{\text{sm}(f'_i)\})_j) : \bigotimes_{j=0}^k (\text{source}(\text{mrrp}(\varphi_j)) \otimes (\vec{\Gamma}\{\text{sm}(f'_i)\})_j)$.

A straightforward inspection of the definitions gives

$$\text{point}(\pi[\rho/\vec{x}_i]) = \text{point}(\pi)[\text{point}(\rho)/\vec{x}_i].$$

The former substitution relation is actually just the restriction of Definition 4.2.12 to points. Hence given $\varphi = \langle \eta^{\triangleright f_i}, p \rangle$ and $\vec{\psi} = \langle \vec{\theta}, \vec{q} \rangle$ we can write, by Remark

4. Subtyping-Aware Polyadic Calculus and Rigid Expansion of Ordinary λ -Terms –
4.2. The Calculus of Subtyping-Aware Polyadic Terms

4.2.15 and the former consideration, $\varphi[\vec{\psi}/\vec{x}_i] = \langle \eta^{\gamma f_i} \otimes_p^{\vec{x}} \vec{\theta}, p\{[f_i]\vec{q}/\text{occ}_{\vec{x}_i}(p)\} \rangle$. In particular, we have that in the substitution abstraction case $g = g'$, as expected. This is an important feature of points, since it says that the substitution operation on the term part is completely determined the standard linear substitution on polyadic terms.

Definition 4.2.24. We define a reduction relation on points $\rightarrow_p \subseteq \Omega^2$ such that if $\varphi \in \Omega_\zeta(\Delta, a)$ and $\varphi \rightarrow_p \varphi'$ then $\varphi' \in \Omega_\zeta(\Delta, a)$ by induction as follows:

$$\begin{array}{c} ((\lambda \vec{x} : f.\varphi)\vec{\psi}) \circ \eta \rightarrow_p \varphi[\vec{\psi}/\vec{x}]\{\eta\} \\ \frac{\langle \langle \eta, f \rangle p \rangle \rightarrow_p \langle \langle \eta', f' \rangle, p' \rangle}{\langle \eta, \lambda \vec{x} : f.p \rangle \rightarrow_p \langle \eta', \lambda \vec{x} : f'.p' \rangle} \\ \frac{\varphi_0 \rightarrow_p \varphi'_0 \quad \varphi_i \rightarrow_p \varphi'_i \quad \eta : \Delta \rightarrow \bigotimes_{j=0}^k \text{source}(mrp(\varphi_j))}{(\varphi_0 \langle \varphi_1, \dots, \varphi_k \rangle) \circ \eta \rightarrow_p (\varphi'_0 \langle \varphi'_1, \dots, \varphi'_k \rangle) \circ \eta} \end{array}$$

We now define some dynamic properties of this reduction, that will rely entirely on the structure of polyadic terms. For $\varphi = \langle \eta, p \rangle \in \Omega_\zeta(\Delta, a)$, we say that φ is a *normal form* if p does not contain redexes as subterms. We say that φ *normalizes* if there exists $n \in \mathbb{N}$ and $\varphi_1, \dots, \varphi_n \in \Omega_\zeta(\Delta, a)$ such that $\varphi = \varphi_0 \rightarrow_p \varphi_1 \cdots \rightarrow_p \varphi_n$ and φ_n is a normal form. We say that φ *strongly normalizes* if there are no infinite reduction chains starting from φ .

Corollary 4.2.25. Let $\varphi = \langle \eta^{f_i}, p \rangle \in \Omega_{\zeta^{\gamma \vec{x}_i}}(\Delta^{\gamma \vec{a}_i}, a)$, $\vec{\psi} = \langle \vec{\theta}, \vec{q} \rangle \in \Omega_{\zeta}(\vec{\Gamma}, \vec{a}_i)$, $f : a \rightarrow a', g : a' \rightarrow a'', \theta^{\gamma g_i} : \zeta^{\gamma \vec{y}_i} : (\Delta')^{\gamma g_i} \rightarrow \zeta^{\gamma \vec{x}_i} : \Delta^{\gamma g_i}, \theta' : \zeta'' : \Delta'' \rightarrow \zeta' : \Delta'$ and $\vec{\eta} : \vec{\zeta}' : \vec{\Gamma}' \rightarrow \vec{\zeta} : \vec{\Gamma}$. The following statements hold.

1. We have $([f]\varphi)\{\theta\} = [f](\varphi\{\theta\})$, $[g \circ f]\varphi = [g]([f]\varphi)$ and $(\varphi\{\theta\})\{\theta'\} = \varphi\{\theta \circ \theta'\}$.
2. We have $([g]\varphi)[\vec{\psi}/\vec{x}_i] = [g]\varphi[\vec{\psi}/\vec{x}_i]$.
3. $\varphi\{\theta^{\gamma g_i}\}[\vec{\psi}/\vec{x}_i] = (\varphi[[g_i]\vec{\psi}/\vec{y}_i])\{\theta^{\gamma -g_i} \otimes sm(g_i)^\star\}$.
4. We have $\varphi[\vec{\psi}\{\vec{\eta}\}/\vec{x}_i] = \varphi[\vec{\psi}/\vec{x}_i]\{1_{\Delta - \vec{a}_i} \otimes \vec{\eta}\}$.
5. If $\varphi \rightarrow_p \varphi'$ then $[f]\varphi \rightarrow_p [f]\varphi'$ and $\varphi\{\theta\} \rightarrow_p \varphi'\{\theta\}$.

Proof. These results are an immediate corollary of Proposition 4.2.11 and the corresponding results already proven for type derivations. \square

Example 4.2.26. We present some examples of reduction for points.

- Let $\langle \langle 1, 1 \rangle, (\lambda \langle x, y \rangle : \langle \sigma, f_1, f_2 \rangle.x\langle y \rangle)\langle w, z \rangle \rangle \in \Omega_{\langle \langle w, z \rangle \rangle}(\langle \langle *, * \rangle \Rightarrow * \rangle, *)$ with σ being the permutations that perform the swap. We have $\langle \langle 1, 1 \rangle, (\lambda \langle x, y \rangle : \langle \sigma, f_1, f_2 \rangle.x\langle y \rangle)\langle w, z \rangle \rangle \rightarrow \langle \langle f_2, f_1 \rangle, z\langle w \rangle \rangle$.

4. Subtyping-Aware Polyadic Calculus and Rigid Expansion of Ordinary λ -Terms –
 4.2. The Calculus of Subtyping-Aware Polyadic Terms

- We give a more complicated example to understand the subtleties of the reduction. Let $\langle \eta, p \rangle =$

$$\langle \langle \langle 1, h_1, h_2 \rangle \rangle, (\lambda \langle w_1, w_2 \rangle : \langle 1, g_1, g_2 \rangle . (\lambda \langle x, y \rangle : \langle \sigma, f_1, f_2 \rangle . x \langle y \rangle)) \langle w_1, w_2 \rangle \rangle \langle z_1, z_2 \rangle \rangle \\ \in \Omega_{\langle \langle z_1, z_2 \rangle \rangle} (\langle \langle *, * \rangle \Rightarrow * \rangle, *)$$

with σ being the permutation that performs the swap. We have $\langle \eta, p \rangle \rightarrow^* \langle \langle \sigma, f_2 \circ (g_1 \circ h_1), f_1 \circ (g_2 \circ h_2) \rangle \rangle_{z_2 z_1}$.

- We introduced the new polyadic calculus to obtain a correspondence between approximants of a term M and intersection type derivations of M . Before going deeper in the technicalities, we present a very simple example to give some intuition. Take $M = (\lambda x.x)y$. Now consider the following type derivations:

$$\pi_1 = \frac{\frac{\langle f \rangle : \langle a \rangle \rightarrow \langle a \rangle}{x : \langle a \rangle \vdash x : a} \quad \frac{\langle g \rangle : \langle a' \rangle \rightarrow \langle a \rangle}{y : \langle a' \rangle \vdash y : a} \quad \langle h \rangle : \langle b \rangle \rightarrow \langle a' \rangle}{y : \langle b \rangle \vdash (\lambda x.x)y : a} \quad \pi_2 = \frac{\langle (g \circ f) \circ h \rangle : \langle b \rangle \rightarrow \langle a \rangle}{y : \langle b \rangle \vdash y : a}$$

We have that $M \rightarrow y$. We would like to say that $\pi_1 \rightarrow \pi_2$. If we consider the natural bijection $f : \llbracket M \rrbracket_{\langle y \rangle} \rightarrow \llbracket y \rrbracket_{\langle y \rangle}$ we have indeed that $f(\tilde{\pi}) = \tilde{\pi}'$. However, this kind of bijection is actually quite complicated to explicitly define. In order to give a workable explicit definition of the bijection, we shall use subtyping-aware polyadic terms via a Curry-Howard style correspondence.

Consider $p = \lambda x : \langle f \rangle . x \langle y \rangle$. We have that $\langle y \rangle \triangleleft y \vdash p \triangleleft M$. We would like to say $p \approx \pi_1$, but we lack the information about subtyping of free variables, i.e., y . Consider then the type derivations:

$$\pi_1 = \frac{\frac{\langle f \rangle : \langle a \rangle \rightarrow \langle a \rangle}{\langle x \rangle : \langle f \rangle : \langle a \rangle \vdash x : a} \quad \frac{\langle g \circ h \rangle : \langle b \rangle \rightarrow \langle a \rangle}{\langle y \rangle : \langle g \circ h \rangle : \langle b \rangle \vdash y : a}}{\langle y \rangle : \langle g \circ h \rangle : \langle b \rangle \vdash (\lambda \langle x \rangle : \langle f \rangle . x) \langle y \rangle : a} \quad \pi_2 = \frac{\langle f \circ (g \circ h) \rangle : \langle b \rangle \rightarrow \langle a \rangle}{\langle y \rangle : \langle f \circ (g \circ h) \rangle : \langle b \rangle \vdash y : a}$$

Again, we have $p \rightarrow y$. Hence, in order to capture π_1 we need the extra information about the subtyping of y . For this reason we are considering points of type derivations. We would like say $\langle \langle (g \circ h) \rangle, p \rangle \approx \pi_1$ and $\langle \langle f \circ (g \circ h) \rangle, y \rangle \approx \pi_2$ with $\langle \langle (g \circ h) \rangle, p \rangle \rightarrow \langle \langle f \circ (g \circ h) \rangle, y \rangle$, clearly taking also into account the equivalence. The former intuition is rooted in one of the basic ideas of categorical semantics: substitution corresponds to composition. In this case, a reduction step performs both a substitution in the term side and a composition in the morphism side. The aim of the next sections is to formally clarify this intuition.

Proposition 4.2.27. For all $n \in \mathbb{N}$ the family $\langle \Omega_\zeta(\Delta, a) \rangle_{\langle \zeta : \Delta, a \rangle \in (SD^n)^* \times D}$ defines

a functor

$$\begin{aligned} \Omega_-(-, -) &: ((SD^n)^*)^o \times D \rightarrow \text{Set} \\ \langle \zeta : \Delta, a \rangle &\mapsto \Omega_\zeta(\Delta, a) \end{aligned}$$

Proof. By Corollary 4.2.25. □

4.2.5. Congruence on Points

We say that R is a *congruence* on points if $R \subseteq \Omega^2$ and R is an equivalence relation that satisfies the following contextual rules:

$$\frac{\langle \eta \oplus \langle f \rangle \rangle = \varphi \sim \langle \eta' \oplus \langle f' \rangle \rangle = \varphi'}{\lambda \vec{x} : \mathbf{f} . \varphi \sim \lambda \vec{x} : \mathbf{f}' . \varphi'}$$

$$\frac{\varphi \sim \varphi' \quad \vec{\psi} \sim \vec{\psi}' \quad \eta}{(\varphi \vec{\psi}) \circ \eta \sim (\varphi' \vec{\psi}') \circ \eta} \quad \frac{\psi_1 \sim \psi'_1 \dots \psi_k \sim \psi'_k}{\langle \psi_1, \dots, \psi_k \rangle \sim \langle \psi'_1, \dots, \psi'_k \rangle}$$

where in the abstraction case we suppose that $\varphi, \varphi' \in \Omega_{\zeta \oplus \langle \vec{x} \rangle}(\Delta \oplus \langle \text{source}(f) = \text{source}(g) \rangle, a)$ for some $n \in \mathbb{N}, \zeta : \Delta \in (SD^n)^*, a \in D$. In the application case we suppose that $(\varphi \vec{\psi}) \circ \eta, (\varphi' \vec{\psi}') \circ \eta \in \Omega_\zeta(\Delta, a)$ for some $n \in \mathbb{N}, \zeta : \Delta \in (SD^n)^*, a \in D$ with $\eta : \Delta \rightarrow (\text{source}(\text{mrp}(\varphi)) = \text{source}(\text{mrp}(\varphi'))) \otimes (\text{source}(\text{mrp}(\vec{\psi})) = \text{source}(\text{mrp}(\vec{\psi}')))$.

Until now, we just considered a congruence generated by a rule on contexts, which corresponds to the rule (3.3) of the congruence on type derivations given in Chapter 3.⁸ We now present a congruence generated by the rule (3.2).

We define a congruence $\sim \subseteq \Omega^2$ as the smallest congruence generated by the following rule:

$$\varphi = (([\langle \alpha, \vec{f} \rangle] \Rightarrow 1 \langle \eta_0, p \rangle) \langle \langle \eta_1, q_1 \rangle, \dots, \langle \eta_{k'}, q_{k'} \rangle \rangle) \circ \eta \sim$$

$$\varphi' = (\langle \eta_0, p \rangle \langle [f_1] \langle \eta_{\alpha(1)}, q_{\alpha(1)} \rangle, \dots, [f_k] \langle \eta_{\alpha(k)}, q_{\alpha(k)} \rangle \rangle) \circ (1_\Delta \otimes \alpha^*) \circ \eta$$

Where $\varphi \in \Omega_\zeta(\Delta, a), \langle \eta_0, p \rangle \in \Omega_{\zeta_0}(\Gamma_0, \langle a_1, \dots, a_k \rangle \Rightarrow a), \langle \eta_i, q_i \rangle \in \Omega_{\zeta_i}(\Gamma_i, a'_i), i \in [k'], \langle \alpha, \vec{f} = \langle f_1, \dots, f_k \rangle \rangle : \langle a'_1, \dots, a'_{k'} \rangle \rightarrow \langle a_1, \dots, a_k \rangle, \eta : \zeta : \Delta \rightarrow \otimes \zeta_j : \otimes \Gamma_j$ for some $\zeta_j : \Gamma_j \in (SD^{\text{len}(\eta)})^*$ with $\Delta = \text{source}(\eta)$. We recall that $\alpha^* : \otimes \zeta_i : \otimes \Gamma_i \rightarrow \otimes \zeta_{\alpha(i)} : \otimes \Gamma_{\alpha(i)}$. We call the former rule the *application congruence base case*.

Example 4.2.28. *We give some examples of congruence over points.*

— *An informative is the following. In general, we have that*

$$((\lambda \vec{x} : \mathbf{f} . \varphi) \vec{\psi}) \circ \eta \sim ((\lambda \vec{y} : 1_{\text{trg}(f)} . (\varphi \{ \vec{y} / \text{occ}_p(\vec{x}) \})) [f] \vec{\psi}) \circ (1 \otimes \text{sm}(f)^*) \circ \eta$$

8. We did not consider the first rule since the congruence generated by the second rule has an interesting quotient, *i.e.*, the set of points of type derivations. The rule (3.2) instead affects the structure of points: if one wrote it down in the polyadic type derivations framework, you do not get any more that if $\pi \sim \pi'$ then $\text{point}(\pi) = \text{point}(\pi')$.

$$\begin{array}{c}
\frac{\vec{y}_1, \dots, \vec{y}_n \in \mathcal{V}^!, j \in [\text{len}(\vec{y}_i)]}{\vec{y}_1 \triangleleft x_1, \dots, \vec{y}_i \triangleleft x_i, \dots, \vec{y}_n \triangleleft x_n \vdash y_{i,j} \triangleleft x_i} \\
\frac{\zeta \triangleleft \chi, \langle x_1, \dots, x_k \rangle \triangleleft x \vdash p \triangleleft M \quad f \in \text{mrp}(SD)}{\zeta \triangleleft \chi \vdash \lambda \langle x_1, \dots, x_k \rangle : f.p \triangleleft \lambda x.M} \\
\frac{\zeta_0 \triangleleft \chi \vdash p \triangleleft M \quad (\zeta_i \triangleleft \chi \vdash q_i \triangleleft N)_{i=1}^k \quad \vec{\alpha} : \zeta \triangleleft \chi \rightarrow \bigotimes_{j_0}^k (\zeta_j \triangleleft \chi)}{\zeta \triangleleft \chi \vdash (p\vec{q})^{\vec{\alpha}} \triangleleft MN}
\end{array}$$

Figure 4.4. – Polyadic approximation of λ -terms.

where we take \vec{y} such that $\text{len}(\vec{y}) = \text{len}(\text{trg}(f))$.

- A concrete example of the former schema is given by $\langle \langle 1_a, 1_{a'} \rangle (\lambda \langle x_1, x_2 \rangle : \sigma.x_2 \langle x_1 \rangle) \langle y_1, y_2 \rangle \rangle \sim \langle \langle 1_{a'}, 1_a \rangle, (\lambda \langle x_2, x_1 \rangle : 1.x_2 \langle x_1 \rangle) \langle y_2, y_1 \rangle \rangle$, where σ is the permutation $\sigma : \langle a, a' \rangle \rightarrow \langle a', a \rangle$.
- We also have

$$(\lambda \vec{x} : 1.\varphi\vec{\psi}) \circ \eta \sim (\lambda(\vec{x}\{\sigma^{-1}\}) : \sigma.\varphi[\sigma^{-1}]\vec{\psi}) \circ (1 \otimes \vec{\sigma}^{-1}) \circ \eta$$

where $\sigma \in \mathfrak{S}_{\text{len}(\vec{x})}$, that is the main example of this kind of equivalence in [TAO17]⁹. The morphism that witnesses the former equivalence is $\sigma^{-1} \Rightarrow 1$.

Proposition 4.2.29. *Let $\varphi \in \Omega_\zeta(\Delta, a)$. Let $\eta : \zeta' : \Delta' \rightarrow \zeta : \Delta, f : a \rightarrow a'$. The following statements hold.*

1. If $\varphi \sim \varphi'$ then $[f]\varphi \sim [f]\varphi'$.
2. If $\varphi \sim \varphi'$ then $\varphi\{\eta\} \sim \varphi'\{\eta\}$.
3. If $\varphi \sim \varphi'$ and $\varphi \in \Omega_{\zeta \gamma \vec{x}_i}(\Delta^{\gamma \vec{a}_i}, a), \vec{\psi}, \vec{\psi}' \in \Omega_{\vec{\zeta}}(\vec{\Gamma}, \vec{a}_i)$ such that $\vec{\psi} \sim \vec{\psi}'$. Then $\varphi[\vec{\psi}/\vec{x}_i] \sim \varphi'[\vec{\psi}'/\vec{x}_i]$.

Proof. By induction on the structure of φ . The proof is detailed in Section B.7. \square

4.3. Rigid Expansion

In this section we introduce a semantics of approximants that generalizes the Taylor expansion. In our setting, the approximants are not just terms, but points of type derivations. The subtyping information of free variables is indeed *essential* for the operational semantics of our calculus, as we saw.

From now on we restrict the construction $(SD^n)^*$ to repetitions-free tuples of variable lists. Let $\chi = \langle x_1, \dots, x_n \rangle \supset \text{fv}(M)$. We define an *approximation assignment* for λ -terms by induction in Figure 4.4.

9. Notice the variance of σ in the second term of the equivalence.

4. Subtyping-Aware Polyadic Calculus and Rigid Expansion of Ordinary λ -Terms –
4.3. Rigid Expansion

Let $\zeta = \vec{z}_1, \dots, \vec{z}_n, \langle \zeta : \Delta, a \rangle \in (SD^n)^\star \times D$. We define the *rigid expansion* of M along $\langle \zeta : \Delta, a \rangle$ as follows:

$$\mathcal{T}_{\text{rig}}(M)_{\vec{x}}(\zeta : \Delta, a) = \{\langle \widetilde{\eta}, p \rangle \mid \vec{z}_1 \triangleleft x_1, \dots, \vec{z}_n \triangleleft x_n \vdash p \triangleleft M \text{ and } \zeta : \eta : \Delta \vdash p : a\}.$$

Remark 4.3.1. *If we explicit inductively the former compact definition we get*

$$\begin{aligned} \mathcal{T}_{\text{rig}}(x_i)_{\vec{x}}(\zeta : \Delta, a) &= \{\langle \eta^{f_i = \langle \alpha_i, f \rangle}, y_{i, \alpha_i(1)} \rangle \mid \langle \eta^{f_i = \langle \alpha_i, f \rangle}, y_{i, \alpha_i(1)} \rangle \in \Omega_\zeta(\Delta, a)\} \cong \\ &SD^{\text{len}(\vec{x})}(\Delta, \langle \langle \rangle, \dots, \langle a \rangle, \dots, \langle \rangle \rangle). \end{aligned}$$

$$\mathcal{T}_{\text{rig}}(\lambda x. M)_{\vec{x}}(\zeta : \Delta, \vec{a} \Rightarrow a) = \{\lambda \vec{x} : \widetilde{f} \cdot \varphi \mid \varphi \in \mathcal{T}_{\text{rig}}(M)_{\vec{x} \oplus \langle x \rangle}(\zeta \oplus \langle \vec{x} \rangle : \Delta \oplus \langle \vec{a} \rangle, a)\}$$

$$\begin{aligned} &\mathcal{T}_{\text{rig}}(PQ)_{\vec{x}}(\zeta : \Delta, a) = \\ &\left(\sum_{\vec{a} = \langle a_1, \dots, a_k \rangle \in SD} \sum_{\zeta_0 : \Gamma_0, \dots, \zeta_k : \Gamma_k \in (SD^n)^\star} \left(\mathcal{T}_{\text{rig}}(P)_{\vec{x}}(\zeta_0 : \Gamma_0, \vec{a} \Rightarrow a) \prod_{i=1}^k \mathcal{T}_{\text{rig}}(Q)_{\vec{x}}(\zeta_i : \Gamma_i, a_i) \right) \circ SD^n(\Delta, \bigotimes_{j=0}^k \Gamma_j) \right) / \sim \end{aligned}$$

Where in the application case

$$\begin{aligned} &\left(\mathcal{T}_{\text{rig}}(P)_{\vec{x}}(\zeta_0 : \Gamma_0, \vec{a} \Rightarrow a) \prod_{i=1}^k \mathcal{T}_{\text{rig}}(Q)_{\vec{x}}(\zeta_i : \Gamma_i, a_i) \right) \circ SD^n(\Delta, \bigotimes_{j=0}^k \Gamma_j) = \\ &\{(\varphi \langle \psi_1, \dots, \psi_k \rangle) \circ \eta \mid \varphi \in \mathcal{T}_{\text{rig}}(M)_{\vec{x}}(\zeta_0 : \Gamma_0, \vec{a} = \langle a_1, \dots, a_k \rangle \Rightarrow a), \\ &\psi_i \in \mathcal{T}_{\text{rig}}(P)_{\vec{x}}(\zeta_i : \Gamma_i, a_i) \text{ and } \eta : \zeta : \Delta \rightarrow \bigotimes \zeta_j : \bigotimes \Gamma_j\}. \end{aligned}$$

The former definition makes the rigid expansion look very much alike the denotation of λ -terms (Figure 3.4). Indeed, in the application case the considered quotient sum is clearly a coend.

Example 4.3.2. *We present some examples of elements in the rigid expansion of ordinary λ -terms.*

1. Let $M = \lambda x. xx$, and suppose that S is the cartesian resource monad. Let $a_0 = \langle a_1, \dots, a_k \rangle \Rightarrow a$ and $a_i = a$ for $i \in [k]$. We have

$$\begin{aligned} &\mathcal{T}_{\text{rig}}(M)_{\langle \rangle}(\langle \rangle, (\vec{a} = \langle a_0, \dots, a_k \rangle) \Rightarrow a) = \\ &\{\langle \langle \rangle, \lambda(\vec{x} = \langle x_1, \dots, x_k \rangle) : \widetilde{f} = \langle \alpha, \vec{f} \rangle \cdot x_{\alpha(1)} \langle x_{\alpha(2)}, \dots, x_{\alpha(k')} \rangle \rangle \mid \text{there exist} \\ &\quad f_0, f_1, \dots, f_k, f' \text{ such that} \\ &\quad f = \left(\bigoplus_{j=0}^{k'} f_j \right) \circ f' : \vec{x} : \vec{a} \rightarrow \bigoplus_{j=0}^{k'} \vec{y}_j : \bigoplus \vec{a}_j \text{ with } \langle \langle f_j \rangle, y_j \rangle \in \Omega_{\langle \vec{y}_j \rangle}(\vec{a}_j, a_j) \\ &\quad \text{for } 1 \leq j \leq k'\} / \sim . \end{aligned}$$

4. Subtyping-Aware Polyadic Calculus and Rigid Expansion of Ordinary λ -Terms –
4.3. Rigid Expansion

2. We have that $\langle\langle, \lambda\langle x, y \rangle : \widetilde{\mathbf{1}_{\langle a_0 \rangle} \oplus \mathbf{c}_{\langle a \rangle}}.x\langle y, y \rangle\rangle \in \mathcal{T}_{\text{rig}}(M)_{\langle, \langle a_0, a \rangle \Rightarrow a} \rangle$. Actually we can generalize it to points of the shape $\tilde{\varphi} = \langle\langle, \lambda\langle x, y^n \rangle : \widetilde{\mathbf{1}_{\langle a_0 \rangle} \oplus \mathbf{c}_{\langle a^n \rangle}}.x\langle y^n \rangle\rangle$ for all $n \in \mathbb{N}$. We also have that $\tilde{\varphi}' = \langle\langle, \lambda\langle x_0, x_1, \dots, x_k \rangle : \widetilde{\mathbf{1}_{\langle \vec{a} \rangle}}.x_0\langle x_1, \dots, x_k \rangle\rangle \in \mathcal{T}_{\text{rig}}(M)_{\langle, \langle \vec{a} = \langle a_0, \dots, a_k \rangle \Rightarrow a \rangle} \rangle$ for all $k \in \mathbb{N}$ and choice of variables x_0, \dots, x_k . There are also "degenerate cases", such as

$$\langle\langle, \lambda\langle x, y, z \rangle : \widetilde{\mathbf{1}_{\langle a_0 \rangle} \oplus \mathbf{c}_{\langle a \rangle} \oplus \mathbf{1}_a}.x\langle y, y, z \rangle\rangle \in \mathcal{T}_{\text{rig}}(M)_{\langle, \langle a_0, a, a \rangle \Rightarrow a \rangle} \rangle.$$

Proposition 4.3.3. *Let $M \in \Lambda$. The family $\langle \mathcal{T}_{\text{rig}}(M)_{\vec{x}}(\zeta : \Delta, a) \rangle_{\langle \zeta : \Delta, a \rangle \in (SD^n)^*}$ induces a functor $\mathcal{T}_{\text{rig}}(M)_{\vec{x}} : ((SD^n)^*)^o \times D \rightarrow \text{Set}$.*

Proof. We want to prove that, for $f : a \rightarrow a'$ and $\eta : \zeta' : \Delta' \rightarrow \zeta : \Delta$ we have a function

$$\mathcal{T}_{\text{rig}}(M)_{\vec{x}}(\eta, f) : \mathcal{T}_{\text{rig}}(M)_{\vec{x}}(\zeta : \Delta, a) \rightarrow \mathcal{T}_{\text{rig}}(M)_{\vec{x}}(\zeta' : \Delta', a')$$

Such that $\mathcal{T}_{\text{rig}}(M)_{\zeta}(\eta \circ \theta, g \circ f) = \mathcal{T}_{\text{rig}}(M)_{\zeta}(\theta, g) \circ \mathcal{T}_{\text{rig}}(M)_{\zeta}(\eta, f)$. We take

$$\langle \eta, p \rangle \rightarrow [f]\langle \eta, p \rangle\{\eta\}$$

By the Corollary 4.2.25 and the definition of actions we can conclude. \square

We are finally ready to prove the central theorem of the chapter, that is the isomorphism between intersection type distributors and the rigid expansion.

Theorem 4.3.4. *Let $M \in \Lambda$, $\vec{x} \supset \text{fv}(M)$. We have a natural isomorphism*

$$\llbracket M \rrbracket_{\vec{x}}(U^*(\zeta : \Delta), a) \cong \mathcal{T}_{\text{rig}}(M)_{\vec{x}}(\zeta : \Delta, a).$$

Proof. We work up to the natural isomorphism of Theorem 3.4.10. This means that we consider the elements of $\llbracket M \rrbracket_{\vec{x}}$ as type derivations. The variable and abstraction cases are immediate by definition and IH.

We prove the application case. If $M = PQ$ and $n = \text{len}(\vec{x})$. By IH we have that there exist natural bijections

$$\phi_{\Delta, a}^P : \llbracket P \rrbracket_{\vec{x}}(\Delta, a) \cong \mathcal{T}_{\text{rig}}(P)_{\vec{x}}(\zeta : \Delta, a)$$

$$\phi_{\Delta, a}^Q : \llbracket Q \rrbracket_{\vec{x}}(\Delta, a) \cong \mathcal{T}_{\text{rig}}(Q)_{\vec{x}}(\zeta : \Delta, a).$$

Given $\vec{a} = \langle a_1, \dots, a_k \rangle \in SD, \Gamma_0, \dots, \Gamma_k \in (SD^n)^*$ we can build a natural transformation

$$\phi_{\vec{a}, \Delta, a}^{PQ} : \sum_{\Gamma_j \in SD^n} \left(\mathcal{T}_{\text{rig}}(P)_{\vec{x}}(\Gamma_0, \vec{a} \Rightarrow a) \times \prod_{i=1}^k \mathcal{T}_{\text{rig}}(Q)_{\vec{x}}(\Gamma_i a_i) \right) \circ SD^n(\Delta, \bigotimes_{j=0}^k \Gamma_j)$$

$$\begin{aligned} &\rightarrow \int^{\Gamma_j \in SD^n} \llbracket P \rrbracket_{\vec{x}}(\Gamma_0, \vec{a} \Rightarrow a) \times \prod_{i=1}^k \llbracket Q \rrbracket_{\vec{x}}(\Gamma_i a_i) \times SD^n(\Delta, \bigotimes_{j=0}^k \Gamma_j) \\ &(\varphi \vec{\psi}) \circ \eta \mapsto \langle \phi_{\Gamma_0, \vec{a} \Rightarrow a}^P(\varphi), \widetilde{\prod_{i=1}^k \phi_{\Gamma_i, a_i}^Q(\vec{\psi})}, \eta \rangle \end{aligned}$$

we observe that the former map is well-defined, since by definitions of points given two different representations of $(\varphi \vec{\psi}) \circ \eta$ we get the same equivalence class¹⁰.

Let $\phi_{\Delta, a}^{PQ} = \sum_{\vec{a} \in SD} \phi_{\vec{a}, \Delta, a}^{PQ}$. By IH and Lemma 3.4.5, the former map is in particular a natural transformation. We prove that $\phi_{\Delta, a}^{PQ}$ is bijective and stable under the additional congruence induced by type lists $\vec{a} \in SD$. We use \sim for the corresponding congruence of points and \sim' for the corresponding congruence on type derivations generated by the rule (3.2).

We need to prove the following three conditions

1. If $\varphi \sim \varphi'$ then $\phi_{\Delta, a}^{PQ}(\varphi) \sim' \phi_{\Delta, a}^{PQ}(\varphi')$.
2. If $\phi_{\Delta, a}^{PQ}(\varphi) \sim' \phi_{\Delta, a}^{PQ}(\varphi')$ then $\varphi \sim \varphi'$.
3. For all π there exists φ such that $\pi = \phi_{\Delta, a}^{PQ}(\varphi)$.

The three former conditions are a direct consequence of naturality and the IH. \square

4.3.1. Reduction and Congruence

In this section we finally formalize the intuition that the invertible 2-cell associated to a β -reduction step can be described by the means of the operational semantics that we gave for our λ -terms approximants. More precisely, we shall prove that if $M \rightarrow_{\beta} N$, then we have

$$\beta^{M, N} : \mathcal{T}_{\text{rig}}(M)_{\vec{x}}(\Delta, a) \cong \mathcal{T}_{\text{rig}}(N)_{\vec{x}}(\Delta, a)$$

where $\beta_{\Delta, a}^{M, N}(\tilde{\varphi}) = \tilde{\psi}$, when $\tilde{\varphi} \rightarrow_p \tilde{\psi}$. This is a very strong result: the reduction relation on points that approximate an ordinary λ -term *collapses* into a function. Moreover, this function will be a bijection. The key ingredient of this result is the equivalence relation. We shall prove that if two approximants reduce to equivalent terms then they are equivalent to each other.

We start by proving two lemmas about substitution.

Lemma 4.3.5. *Let $\tilde{\varphi} \in \mathcal{T}_{\text{rig}}(M\{N/x\})_{\vec{x}}(\zeta : \Delta, a)$ with $\vec{x} \supseteq (\text{fv}(M)/\{x\}) \cup \text{fv}(N)$. There exist $\vec{a} = \langle a_1, \dots, a_k \rangle \in SD$, $\zeta_0 \oplus \langle \vec{y} \rangle : \Gamma_0 \oplus \langle \vec{a}_0 \rangle, \dots, \zeta_k : \Gamma_k \in (SD^n)^*$, $\eta : \zeta : \Delta \rightarrow \bigotimes \zeta_j : \Gamma_j$ and $\tilde{\varphi} \in \mathcal{T}_{\text{rig}}(M)_{\vec{x} \oplus \langle x \rangle}(\zeta_0 \oplus \langle \vec{y} \rangle : \Gamma_0 \oplus \langle \vec{a}_0 \rangle, a)$, $\tilde{\psi}_i \in \mathcal{T}_{\text{rig}}(N)_{\vec{x}}(\zeta_i : \Gamma_i, a_i)$ for $i \in [k]$ such that $\tilde{\varphi} = \varphi'[\vec{\psi} = \langle \psi_1, \dots, \psi_k \rangle / \vec{y}] \circ \eta$.*

¹⁰. This happens because points are indeed the equivalence classes for the considered equivalence relation. See Proposition 4.2.11 and Definition 4.2.21.

$$\begin{array}{c}
\hline
((\lambda \vec{x} : f.\varphi)\vec{\psi}) \circ \eta \rightarrow_p \varphi[\vec{\psi}/\vec{x}] \triangleleft (\lambda x.P)Q \rightarrow_\beta P\{Q/x\} \\
\varphi = \langle \eta \oplus \langle f \rangle, p \rangle \rightarrow_p \psi = \langle \eta \oplus \langle g \rangle, p \rangle \triangleleft M \rightarrow_\beta N \\
\hline
\lambda \vec{x} : f.\varphi \rightarrow_p \lambda \vec{x} : g.\psi \triangleleft \lambda x.M \rightarrow_\beta \lambda x.N \\
\varphi \rightarrow_p \varphi' \triangleleft P \rightarrow_\beta P' \\
\hline
(\varphi\vec{\psi}) \circ \eta \rightarrow_p (\varphi'\vec{\psi}) \circ \eta \triangleleft PQ \rightarrow_\beta P'Q \\
\vec{\psi} \rightarrow_p \vec{\psi}' \triangleleft Q \rightarrow_\beta Q' \\
\hline
(\varphi\vec{\psi}) \circ \eta \rightarrow_p (\varphi\vec{\psi}') \circ \eta \triangleleft PQ \rightarrow_\beta PQ' \\
Q \rightarrow_\beta Q' \\
\hline
(\varphi\langle \rangle) \circ \eta \rightarrow_p (\varphi\langle \rangle) \circ \eta \triangleleft PQ \rightarrow_\beta PQ' \\
\psi_1 \rightarrow_p \psi'_1 \triangleleft M \rightarrow_\beta N \dots \psi_l \rightarrow_p \psi'_l \triangleleft M \rightarrow_\beta N \\
\hline
\langle \psi_1, \dots, \psi_k \rangle \rightarrow_p \langle \psi'_1, \dots, \psi'_k \rangle \triangleleft M \rightarrow_\beta N
\end{array}$$

whenever the points are well-defined.

Figure 4.5. – Approximation of β -reduction steps.

Proof. By induction on the structure of M . The proof is detailed in Section B.11. \square

Lemma 4.3.6. *Let $M, N \in \Lambda, \vec{x} \supseteq (\text{fv}(M)/\{x\}) \cup \text{fv}(N)$ and $x \notin \vec{x}$. Let $\vec{a} = \langle a_1, \dots, a_k \rangle \in SD, \zeta_0 \oplus \langle \vec{y} \rangle : \Gamma_0 \oplus \langle \vec{a}_0 \rangle, \dots, \zeta_k : \Gamma_k \in (SD^n)^*, \eta : \zeta : \Delta \rightarrow \otimes \zeta_j : \Gamma_j, \tilde{\varphi} \in \mathcal{T}_{\text{rig}}(M)_{\vec{x} \oplus \langle x \rangle}(\zeta_0 \oplus \langle \vec{y} \rangle : \Gamma_0 \oplus \langle \vec{a}_0 \rangle, a), \tilde{\psi}_i \in \mathcal{T}_{\text{rig}}(N)_{\vec{x}}(\zeta_i : \Gamma_i, a_i)$ for $i \in [k]$. Then $\varphi\{\vec{\psi} = \langle \psi_1, \dots, \psi_k \rangle / \vec{x}\} \in \mathcal{T}_{\text{rig}}(M\{N/x\})_{\vec{x}}(\zeta : \Delta, a)$.*

Proof. By induction on the structure of M . The proof is detailed in Section B.12. \square

We define the approximation relation (Figure 4.4) directly over points.

We write $\varphi \triangleleft M$ if $\tilde{\varphi} \in \mathcal{T}_{\text{rig}}(M)_{\vec{x}}(\zeta : \Delta, a)$ for some $\langle \zeta : \Delta, a \rangle \in (SD^{\text{len}(\vec{x})})^* \times D$. In particular, by the former two lemmas we have that if $((\lambda \vec{x} : f.\varphi)\vec{\psi}) \circ \eta \triangleleft (\lambda x.M)N$ then $\varphi[\vec{\psi}/\vec{x}] \circ \eta \triangleleft M\{N/x\}$.

Let $\varphi \triangleleft M, \psi \triangleleft N$ with $M \rightarrow_\beta N$. We define $\varphi \rightarrow_p \psi \triangleleft M \rightarrow_\beta N$ by induction in Figure 4.5. In particular, we have that if $\varphi \rightarrow_p \psi \triangleleft M \rightarrow_\beta N$ then $\varphi \rightarrow_p^- \psi$.

Remark 4.3.7. *It is necessary to consider the reflexive closure of \rightarrow_p in Figure 4.5. Consider $M = x((\lambda y.y)z) \rightarrow_\beta N = xz$. Clearly, $\langle \langle 1 \rangle, x \rangle$, where we keep the typing implicit, is an approximant of both M and N , but it is a normal form.*

Proposition 4.3.8. *Let $\varphi_1 \rightarrow_p \varphi'_1 \triangleleft M \rightarrow_\beta N, \varphi_2 \rightarrow_p \varphi'_2 \triangleleft M \rightarrow_\beta N$. If $\varphi_1 \sim \varphi_2$ then $\varphi'_1 \sim \varphi'_2$.*

4. Subtyping-Aware Polyadic Calculus and Rigid Expansion of Ordinary λ -Terms –
4.3. Rigid Expansion

Proof. By induction on the β -reduction step $M \rightarrow_\beta N$. Let $M = (\lambda x.P)Q$ and $N = P\{Q/x\}$. Then $\varphi_1 = ((\lambda \vec{x}_1 : \mathbf{f}_1.\psi_1)\vec{\psi}_1) \circ \eta_1$ and $\varphi_2 = ((\lambda \vec{x}_2 : \mathbf{f}_2.\psi_2)\vec{\psi}_2) \circ \eta_2$. By hypothesis we have that $\varphi_1 \sim \varphi_2$. Suppose that they are equivalent by the application base case. Then $\varphi_1 = (([h \Rightarrow 1](\lambda \vec{x}_2 : \mathbf{f}_2.\psi_2))\vec{\psi}_1) \circ \eta_1$ and $\varphi_2 = ((\lambda \vec{x}_2 : \mathbf{f}_2.\psi_2)[h]\vec{\psi}_2) \circ (1 \otimes \text{sm}(h)) \circ \eta_1$ for some morphism h . Let \vec{y} such that $\text{len}(\vec{y}) = \text{len}(\text{source}(h))$. Then $[h \Rightarrow 1](\lambda \vec{x}_2 : \mathbf{f}_2.\psi_2) = \lambda \vec{y} : \mathbf{f}_2 \circ h.(\psi_2\{1 \oplus \langle h \rangle\})$. By Corollary 4.2.25 we get

$$\psi_2[[h]\vec{\psi}_1/\vec{x}] \circ (1 \otimes \text{sm}(h)) \circ \eta_1 = \psi_2\{1 \oplus \langle h \rangle\}[\vec{\psi}_1/\vec{y}] \circ \eta_1$$

Then we can conclude. If $\varphi \sim \varphi'$ by contextuality, the result follows directly by Proposition 4.2.29.

If $M = \lambda x.M' \rightarrow_\beta N = \lambda x.N'$ with $M' \rightarrow_\beta N'$ then $\varphi_1 = \lambda \vec{x}_1 : \mathbf{f}_1.\psi_1 \rightarrow_p \varphi'_1 = \lambda \vec{x} : \mathbf{f}'.\psi'_1$ and $\varphi_2 = \lambda \vec{x}_2 : \mathbf{f}_2.\psi_2 \rightarrow_p \varphi'_2 = \lambda \vec{x}_2 : \mathbf{f}'_2.\psi'_2$. By hypothesis we have that $\psi_1 \sim \psi_2$. Hence, by definition of congruence, $\vec{x}_1 = \vec{x}_2$. Then we conclude applying the IH.

If $M = PQ \rightarrow_\beta N = P'Q$ with $P \rightarrow_\beta P'$ the result is again an immediate corollary of congruence contextuality and the IH. The same happens for the other application case. \square

By the former proposition, the reduction relation for point that approximate ordinary λ -terms lifts to equivalence classes. Under these specific conditions, if $\varphi \rightarrow_p \psi$ we can then write $\tilde{\varphi} \rightarrow_p \tilde{\psi}$.

We now introduce a notion of *coherence* for points, adapting the corresponding notion on resource terms that we introduced in Chapter 1. Intuitively, two points are coherent if the respective associated terms represent the same syntactic tree. Hence, coherence is tightly connected to approximation.

Definition 4.3.9. We define a coherence relation on points $\circ \subseteq \Omega^2$ by induction as follows:

$$\frac{}{\langle \eta, x \rangle \circ \langle \theta, y \rangle} \quad \frac{\varphi \circ \varphi'}{\lambda \vec{x} : \mathbf{f}.\varphi \circ \lambda \vec{x} : \mathbf{f}'.\varphi'}$$

$$\frac{\varphi \circ \varphi' \quad \vec{\psi} \circ \vec{\psi}' \quad \eta, \eta'}{(\varphi\vec{\psi}) \circ \eta \circ (\varphi'\vec{\psi}') \circ \eta'} \quad \frac{\psi_i \circ \psi_j \text{ for } 1 \leq i, j \leq k + k'}{\langle \psi_1, \dots, \psi_k \rangle \circ \langle \psi_{k+1}, \dots, \psi_{k+k'} \rangle}$$

whenever the points are well-defined.

We observe that if $\varphi \sim \varphi'$ then $\varphi \circ \varphi'$ and $[f]\varphi \sim [f]\varphi', \varphi\{\eta\} \circ \varphi'\{\eta\}$ whenever those points are well-defined.

Proposition 4.3.10. Let $M \in \Lambda$, $\text{fv}(M) \subseteq \vec{x}$ and $\langle \zeta : \Delta, a \rangle \in (SD^{\text{len}(\vec{x})})^* \times D$. The set $\mathcal{T}_{\text{rig}}(M)_{\vec{x}}(\zeta : \Delta, a)$ is a clique for the coherence relation on points.

Proof. Straightforward induction on the structure of M . \square

Theorem 4.3.11 (Uniformity of Reduction). *Let $\varphi_1, \varphi_2 \in \Omega_\zeta(\Delta, a)$, $M, N \in \Lambda$ such that $\varphi_1 \rightarrow_p \varphi'_1 \triangleleft M \rightarrow_\beta N$ and $\varphi_2 \rightarrow_p \varphi'_2 \triangleleft M \rightarrow_\beta N$. If $\varphi'_1 \sim \varphi'_2$ then $\varphi_1 \sim \varphi_2$.*

Proof. The result is a corollary of an appropriate substitution lemma. The proof is detailed in Section B.9. \square

The former proposition can be seen as the computational meaning of the semantic equivalence on points and, more generally, on type derivations. Intuitively, we can rephrase it as follows: two approximants that represent the *same* dynamics must be considered as equivalent.

Proposition 4.3.12. *Let $M, N \in \Lambda$, $\vec{x} \supseteq \text{fv}(M) \cup \text{fv}(N)$ such that $M \rightarrow_\beta N$. We have*

1. *For all $\langle \zeta : \Delta, a \rangle \in (SD^{\text{len}(\vec{x})})^* \times D$ and $\tilde{\varphi} \in \mathcal{T}_{\text{rig}}(M)_{\vec{x}}(\zeta : \Delta, a)$ there exists $\tilde{\psi} \in \mathcal{T}_{\text{rig}}(N)_{\vec{x}}(\zeta : \Delta, a)$ such that $\tilde{\varphi} \rightarrow_p \tilde{\psi} \triangleleft M \rightarrow_\beta N$.*
2. *For all $\langle \zeta : \Delta, a \rangle \in (SD^{\text{len}(\vec{x})})^* \times D$ and $\tilde{\psi} \in \mathcal{T}_{\text{rig}}(N)_{\vec{x}}(\zeta : \Delta, a)$ there exists $\tilde{\varphi} \in \mathcal{T}_{\text{rig}}(M)_{\vec{x}}(\zeta : \Delta, a)$ such that $\tilde{\varphi} \rightarrow_p \tilde{\psi} \triangleleft M \rightarrow_\beta N$.*

Proof. Completely straightforward induction on the β -reduction step. The non-trivial case is the base case, that is a corollary of Lemmas 4.3.5 and 4.3.6. \square

Theorem 4.3.13. *Let $M, N \in \Lambda$, $\vec{x} \supseteq \text{fv}(M) \cup \text{fv}(N)$ such that $M \rightarrow_\beta N$. We have a natural isomorphism*

$$\beta_{\Delta, a}^{M, N} : \mathcal{T}_{\text{rig}}(N)_{\vec{x}}(\Delta, a) \cong \mathcal{T}_{\text{rig}}(M)_{\vec{x}}(\Delta, a)$$

induced by the reflexive closure of the reduction relation on points.

Proof. We can define the following function

$$\beta_{\Delta, a}^{M, N} : \mathcal{T}_{\text{rig}}(N)_{\vec{x}}(\Delta, a) \cong \mathcal{T}_{\text{rig}}(M)_{\vec{x}}(\Delta, a)$$

$$\beta_{\Delta, a}^{M, N}(\tilde{\varphi}) = \tilde{\psi} \text{ s.t. } \tilde{\varphi} \rightarrow_p \tilde{\psi} \triangleleft M \rightarrow_\beta N$$

Since, by Proposition 4.3.12, for all $\tilde{\varphi} \in \mathcal{T}_{\text{rig}}(M)_{\vec{x}}(\Delta, a)$ there exists $\tilde{\psi} \in \mathcal{T}_{\text{rig}}(N)_{\vec{x}}(\zeta : \Delta, a)$ such that $\tilde{\varphi} \rightarrow_p \tilde{\psi} \triangleleft M \rightarrow_\beta N$ and by Proposition 4.3.8 $\tilde{\psi}$ is unique. The former function is also natural in both Δ and a by Corollary 4.2.25 and Proposition 4.2.29. By Theorem 4.3.11 $\beta_{\Delta, a}^{M, N}$ is injective and by Proposition 4.3.12 $\beta_{\Delta, a}^{M, N}$ is surjective. \square

4.4. An Example: the Linear Case

In this section we present a concrete example of our construction, in the case when the resource monad S is the linear resource monad, *i.e.* the 2-monad where SA is the symmetric strict monoidal completion of A . As already seen in the context of intersection type distributors, this particular construction corresponds to the categorification of the standard relation semantics for linear logic.

Subtyping-Aware Linear Polyadic Terms The syntax of the calculus is just a particular instance of the parametric syntax for subtyping-aware polyadic terms.

A simple inspection of the definitions gives the following type assignment for polyadic terms:

$$\begin{array}{c}
 1_{\langle \rangle} : \langle \rangle \rightarrow \langle \rangle, \dots, \langle f \rangle : \langle a' \rangle \rightarrow \langle a \rangle, \dots, 1_{\langle \rangle} : \langle \rangle \rightarrow \langle \rangle \\
 \hline
 \langle \rangle : 1_{\langle \rangle} : \langle \rangle, \dots, \langle x \rangle : \langle f \rangle : \langle a' \rangle, \dots, \langle \rangle : 1_{\langle \rangle} : \langle \rangle \vdash x : a \\
 \vdots \\
 \zeta \oplus \langle \vec{z} \rangle : \eta \oplus \langle f \rangle : \Delta \oplus \langle \vec{a} \rangle \vdash p : a \\
 \hline
 \zeta : \eta : \Delta \vdash \lambda \vec{z} : f.p : \vec{a} \Rightarrow a \\
 \hline
 \zeta_0 : \eta_0 : \Gamma_0 \vdash p : \langle a_1, \dots, a_k \rangle \Rightarrow a \quad \left(\begin{array}{c} \pi_0 \\ \vdots \\ \pi_i \\ \vdots \\ \zeta_i : \eta_i : \Gamma_i \vdash q_i : a_i \end{array} \right)_{i=1}^k \quad \eta : \zeta : \Delta \rightarrow \bigotimes_{j=0}^k \zeta_j : \bigotimes_{j=0}^k \Gamma_j \\
 \hline
 \zeta : (\bigotimes_{j=0}^k \eta_j) \circ \eta : \Delta \vdash (p \langle q_1, \dots, q_k \rangle)^\eta : a
 \end{array}$$

We call R_1 the former type system. The first thing to notice is that, in the application case, the substitution $(p \langle q_1, \dots, q_k \rangle)^\eta = p[\zeta \{ \text{sm}(\eta) \} / \bigotimes_{j=0}^k \zeta_j]$ does not perform any copying operation, since the structural function of morphisms are just permutations. Hence this kind of substitution just boils down to linear substitutions where the position of free variables needs to be rearranged. This phenomenon suggests to restrict the typing of applications as follows:

$$\begin{array}{c}
 \zeta_0 : \eta_0 : \Gamma_0 \vdash p : \langle a_1, \dots, a_k \rangle \Rightarrow a \quad \left(\begin{array}{c} \pi_0 \\ \vdots \\ \pi_i \\ \vdots \\ \zeta_i : \eta_i : \Gamma_i \vdash q_i : a_i \end{array} \right)_{i=1}^k \quad (\bigotimes_{j=0}^k \Gamma_j) \{ \text{sm}(\eta)^{-1} \} : \zeta : \Delta \rightarrow \bigotimes_{j=0}^k \zeta_j : \bigotimes_{j=0}^k \Gamma_j \\
 \hline
 (\bigotimes_{j=0}^k \Gamma_j) \{ \text{sm}(\eta)^{-1} \} : (\bigotimes_{j=0}^k \eta_j) \circ \eta : \Delta \vdash (p \langle q_1, \dots, q_k \rangle)^\eta : a
 \end{array}$$

where we additionally impose that in a term $p \in \Xi$ each variable appears exactly once. We call R_2 the type system obtained by replacing the application rule of system R_1 with the former one. Clearly we have that the system R_2 is contained in R_1 . Then we observe that $(p \langle q_1, \dots, q_k \rangle)^\eta = p \langle q_1, \dots, q_k \rangle$. Hence the operation performed on contexts in that case is quite familiar and its just an exchange rule mixed within the application rule of our type system. This restriction is legitimate, in the sense that all the construction that we made in the previous sections does not depend on a particular choice of free variables in the application case.

It is completely straightforward to see that the two type systems are actually equivalent, in the sense that $\zeta : \eta \vdash_{R_1} p : a$ iff $\zeta : \eta \vdash_{R_2} p : a$. Moreover, they are equivalent in a stronger sense:

Lemma 4.4.1. *Let $\pi \in R_1$. Then there exists $\pi' \in R_2$ such that $\pi' \sim \pi$.*

Proof. By induction on the structure of π . The proof is detailed in Section B.13. \square

4.4.1. Points of Linear Type Derivations

Lemma 4.4.1 allows us to give an easier definition of points in the linear framework: Given $\langle \zeta : \Delta, a \rangle \in (SD^n)^* \times D$ with $n \in \mathbb{N}$ we define the *set of linear points* for $\langle \zeta : \Delta, a \rangle$, $\Omega_\zeta(\Delta, a)$ by induction as follows:

- If $\eta = \langle 1, \dots, f_i = \langle \alpha_i, f \rangle, \dots, 1 \rangle : \zeta : \langle \langle \rangle, \dots, \vec{a}_i, \dots, \langle \rangle \rangle \rightarrow \langle \langle \rangle, \dots, \langle a \rangle, \dots, \langle \rangle \rangle$ then $\langle \eta, x_i \rangle \in \Omega_\zeta(\langle \langle \rangle, \dots, \langle a' \rangle, \dots, \langle \rangle \rangle, a)$.
- If $\langle \eta \oplus \langle f \rangle, p \rangle \in \Omega_{\zeta \oplus (\vec{x})}(\Delta \oplus \langle \vec{a} \rangle, a)$ then $\langle \eta, \lambda \vec{x} : f.p \rangle \in \Omega_\zeta(\Delta, \vec{a} \Rightarrow a)$.
- If $\langle \eta_0, s \rangle \in \Omega_{\zeta_0}(\Gamma_0, \langle a_1, \dots, a_k \rangle \Rightarrow a)$, $\langle \eta_i, t_i \rangle \in \Omega_{\zeta_i}(\Gamma_i, a_i)$ for $i \in [k]$ and $\eta : (\otimes_{j=0}^k \zeta_j) \{ \text{sm}(\eta) \} : \Delta \rightarrow \otimes_{j=0}^k \zeta_j : \otimes_{j=1}^k \Gamma_j$ then $\langle (\otimes_{j=1}^k \eta_j) \circ \eta, (s \langle t_1, \dots, t_k \rangle) \rangle \in \Omega_\zeta(\Delta, a)$.

In particular, since the linear system is *relevant*, a point completely determines its typing, *i.e.* if $\varphi \in \Omega_\zeta(\Delta, a) \cap \Omega_{\zeta'}(\Delta', a')$ then $\zeta = \zeta'$, $\Delta = \Delta'$ and $a = a'$. Hence the set of linear points Ω^l can be written as

$$\bigsqcup_{n \in \mathbb{N}} \bigsqcup_{\langle \zeta : \Delta, a \rangle \in (SD^n)^*} \Omega_\zeta^l(\Delta, a).$$

If we consider the reduction relation $\rightarrow_p \subseteq \Omega^l$ we can prove that it is strongly normalizable in a completely *combinatorial way*, following the standard argument for strong normalization for resource terms [ER08]. We define the *size* of a point as follows $s(\langle \eta, p \rangle) = s(p)$.

Lemma 4.4.2. *Let $\varphi \in \Omega_{\zeta \circ \vec{x}_i}(\Delta \circ \vec{a}_i, a)$ and $\vec{\psi} \in \Omega_{\vec{\zeta}}(\vec{\Gamma}, \vec{a}_i)$. We have $s(\varphi[\vec{\psi}/\vec{x}_i]) = s(\varphi) + s(\vec{\psi})$.*

Proposition 4.4.3. *Let $\varphi \in \Omega$. If $\varphi \rightarrow_p \varphi'$ then $s(\varphi) = s(\varphi') - 1$.*

Theorem 4.4.4. *The reduction \rightarrow_p^l is strongly normalizing.*

Proof. By induction on the size of a term. Let $\varphi \rightarrow_p \varphi'$ then, by former lemma, we have that $s(\varphi') = s(\varphi) - 1$ and we can apply the IH and conclude. \square

We call the linear rigid expansion of a λ -term the *rigid Taylor expansion* of it.

Recovering the Ordinary Taylor expansion In this section we will denote standard resource terms (Definition 1.4.1) as s, t, \dots and polyadic terms as p, q, \dots . We want to establish an explicit relationship between standard resource terms and subtyping-aware polyadic terms.

We define the *approximation assignment* for resource terms by induction in Figure 4.6.

Let $\langle \zeta : \Delta, a \rangle \in (SD^n)^* \times D$ for some $n \in \mathbb{N}$, s be a resource term and $\vec{x} \supset \text{fv}(s)$ such that $\text{len}(\vec{x}) = n$. We define the *rigid expansion* of s on $\langle \zeta : \Delta, a \rangle$ as follows

$$\mathcal{T}_{\text{rig}}(s)_{\vec{x}}(\zeta : \Delta, a) \{ \widetilde{\langle \eta, p \rangle} \mid \zeta \triangleleft \vec{x} \vdash p \triangleleft s \text{ and } \zeta : \eta : \Delta \vdash p : a \}.$$

4. Subtyping-Aware Polyadic Calculus and Rigid Expansion of Ordinary λ -Terms –
4.4. An Example: the Linear Case

$$\begin{array}{c}
\frac{y \in \mathcal{V}}{\langle \rangle \triangleleft x_1, \dots, \langle y_i \rangle \triangleleft x_i, \dots, \langle \rangle \triangleleft x_n \vdash y \triangleleft x} \\
\frac{\zeta \triangleleft \chi, \langle x_1, \dots, x_k \rangle \triangleleft x \vdash p \triangleleft s \quad f \in \text{mrp}(SD)}{\zeta \triangleleft \chi \vdash \lambda \langle x_1, \dots, x_k \rangle : f.p \triangleleft \lambda x.s} \\
\frac{\zeta_0 \triangleleft \chi \vdash p \triangleleft s \quad (\zeta_i \triangleleft \chi \vdash q_i \triangleleft t_{\sigma(i)})_{i=1}^k \quad \vec{\tau} : \zeta \triangleleft \chi \rightarrow \bigotimes_{j=0}^k (\zeta_j \triangleleft \chi) \quad \sigma \in S_k}{\zeta \triangleleft \chi \vdash (p(q_1, \dots, q_k))^{\vec{\tau}} \triangleleft s[t_1, \dots, t_k]}
\end{array}$$

Figure 4.6. – Polyadic approximation of resource terms.

If we develop inductively the former definition we get the following:

$$\mathcal{T}_{\text{rig}}(x)_{\vec{x}}(\zeta : \Delta, a) = \{\langle \eta, y \rangle \mid \langle \eta, y \rangle \in \Omega_{\zeta}(\Delta, a)\}.$$

$$\mathcal{T}_{\text{rig}}(\lambda x.s)_{\vec{x}}(\zeta : \Delta, a) = \{\widetilde{\lambda \vec{z} : f.\varphi} \mid \tilde{\varphi} \in \mathcal{T}_{\text{rig}}(s)_{\vec{x} \oplus \langle x \rangle}(\zeta \oplus \langle \vec{z} \rangle : \Delta \oplus \langle \text{source}(f) \rangle, a)\}.$$

$$\mathcal{T}_{\text{rig}}(\langle s \rangle[t_1, \dots, t_k])_{\vec{x}}(\zeta : \Delta, a) = \{(\varphi \langle \psi_1, \dots, \psi_k \rangle) \circ \eta \mid \tilde{\varphi} \in \mathcal{T}_{\text{rig}}(s)_{\vec{x}}(\zeta_0 : \Gamma_0, \vec{a} \Rightarrow a)\}$$

$$\tilde{\psi}_i \in \mathcal{T}_{\text{rig}}(t_{\sigma(i)})_{\vec{x}}(\zeta_i : \Gamma_i, a_i) \text{ for } i \in [k] \text{ and } \eta : \zeta : \Delta \rightarrow \bigotimes_{j=0}^k \zeta_j : \bigotimes_{j=0}^k \Gamma_j$$

for some $\zeta_0 : \Gamma_0, \zeta_i : \Gamma_i \in (SD^{\text{len}(\vec{x})})^*, \vec{a} = \langle a_1, \dots, a_k \rangle \in SD$ and $\sigma \in \mathfrak{S}_k$.

Theorem 4.4.5. *Let $M \in \Lambda$. We have*

$$\mathcal{T}_{\text{rig}}(M)_{\vec{x}}(\zeta : \Delta, a) = \sum_{s \in T(M)} \mathcal{T}_{\text{rig}}(s)_{\vec{x}}(\zeta : \Delta, a).$$

Proof. The variable and abstraction cases are immediate, we prove the application case. Let $M = PQ$ we have

$$\mathcal{T}_{\text{rig}}(PQ)_{\vec{x}}(\zeta : \Delta, a) = \{(\varphi \langle \psi_1, \dots, \psi_k \rangle) \circ \eta \mid \tilde{\varphi} \in \mathcal{T}_{\text{rig}}(P)_{\vec{x}}(\zeta_0 : \Gamma_0, \vec{a} \Rightarrow a)\}$$

$$\tilde{\psi}_i \in \mathcal{T}_{\text{rig}}(Q)_{\vec{x}}(\zeta_i : \Gamma_i, a_i) \text{ for } i \in [k] \text{ and } \eta : \zeta : \Delta \rightarrow \bigotimes_{j=0}^k \zeta_j : \bigotimes_{j=0}^k \Gamma_j$$

for some $\zeta_0 : \Gamma_0, \zeta_i : \Gamma_i \in (SD^{\text{len}(\vec{x})})^*, \vec{a} = \langle a_1, \dots, a_k \rangle \in SD$

and

$$\begin{aligned}
& \sum_{s \in T(M)} \mathcal{T}_{\text{rig}}(s)_{\vec{x}}(\zeta : \Delta, a) = \\
& \sum_{s \in T(P), [t_1, \dots, t_k] \in T(Q)} \mathcal{T}_{\text{rig}}(\langle s \rangle[t_1, \dots, t_k])_{\vec{x}}(\zeta : \Delta, a) =
\end{aligned}$$

4. Subtyping-Aware Polyadic Calculus and Rigid Expansion of Ordinary λ -Terms –
 4.4. An Example: the Linear Case

$$\{(\varphi\langle\psi_1, \dots, \psi_k\rangle) \circ \eta \mid \tilde{\varphi} \in \mathcal{T}_{\text{rig}}(s)_{\vec{x}}(\zeta_0 : \Gamma_0, \vec{a} \Rightarrow a)\}$$

$$\tilde{\psi}_i \in \mathcal{T}_{\text{rig}}(t_{\sigma(i)})_{\vec{x}}(\zeta_i : \Gamma_i, a_i) \text{ for } i \in [k] \text{ and } \eta : \zeta : \Delta \rightarrow \bigotimes_{j=0}^k \zeta_j : \bigotimes_{j=0}^k \Gamma_j$$

for some $\zeta_0 : \Gamma_0, \zeta_i : \Gamma_i \in (SD^{\text{len}(\vec{x})})^*, \vec{a} = \langle a_1, \dots, a_k \rangle \in SD$ and $\sigma \in S_k$.

We prove the result by double inclusion.

(\subseteq) Let $(\varphi\langle\psi_1, \dots, \psi_k\rangle) \circ \eta \in \mathcal{T}_{\text{rig}}(PQ)_{\vec{x}}(\zeta : \Delta, a)$, by IH we have $\varphi \in \mathcal{T}_{\text{rig}}(s)_{\vec{x}}(\zeta_0 : \Gamma_0, \vec{a} \Rightarrow a)$ and $\psi_i \in \mathcal{T}_{\text{rig}}(t_i)_{\vec{x}}(\zeta_i : \Gamma_i, a_i)$ for a unique $s \in T(M), t_i \in T(Q)$. Then, by definition

$$(\varphi\langle\psi_1, \dots, \psi_k\rangle) \circ \eta \in \mathcal{T}_{\text{rig}}(\langle s \rangle[t_1, \dots, t_k])_{\vec{x}}(\zeta : \Delta, a).$$

(\supseteq) Let $(\varphi\langle\psi_1, \dots, \psi_k\rangle) \circ \eta \in \mathcal{T}_{\text{rig}}(\langle s \rangle[t_1, \dots, t_k])_{\vec{x}}(\zeta : \Delta, a)$ for a unique $s \in T(M), t_i \in T(Q)$. Then for some $\sigma \in \mathfrak{S}_k, \varphi \in \mathcal{T}_{\text{rig}}(s)_{\vec{x}}(\zeta_0 : \Gamma_0, \vec{a} \Rightarrow a)$ and $\psi_i \in \mathcal{T}_{\text{rig}}(t_{\sigma(i)})_{\vec{x}}(\zeta_i : \Gamma_i, a_i)$. Then we apply the IH and conclude. \square

Conclusion

We conclude the present work by discussing ongoing further work and speculating about some interesting perspectives.

1. We are currently working on proving that the reduction relation on points (Definition 4.2.24) is both strongly normalizing and confluent. While strong normalization seems to follow from a non-trivial extension of the classic reducibility method, the confluence of the reduction is technically very challenging¹¹. Supposing that the reduction satisfies both properties, we believe that we can get a completely natural general commutation theorem between normal form of the rigid expansion and Böhm Trees. This would improve the standard result on Taylor expansion from a semantic point of view: the rigid expansion is now isomorphic to its normal form.
2. Moreover, we are also working to extend our framework to an untyped call-by-push value setting [OG21]. In this way, we can get, through a nice factorization of the semantics, both an interpretation of call-by-name and call-by-value dynamics. Moreover, this opens the path to the more general setting of Multiplicative Exponential Linear Logic.
3. The first question opened by this thesis that deserves to be further considered is the relationship between the bicategorical denotational semantics introduced in Chapters 3 and 4 and the more syntactic 2-operadic approach to intersection type systems and approximants of λ -terms studied in [MPV18; Maz17]. As already briefly observed, an extension of the results of Mazza *et al.* to our setting would overcome some limitations of their original framework. Indeed, we could probably obtain in this way an improvement of the *approximation presheaf* [Maz17][pp. 63-67]. Roughly speaking, the approximation presheaf is a (lax) morphism from the 2-operad of λ -terms¹² to the bicategory of relational distributors which takes a term M as input and returns a distributor $T_D(M)$ that works as follows: given Δ, a respectively a type context and a type in an appropriate (discrete) category of types D , we get the following set

$$T_D(M)(\Delta, a) = \{p \mid p \text{ is an approximant of } M \text{ s.t. } \Delta \vdash p : a\}.$$

11. However, since this reduction relation corresponds to the 2-cell associated to a β -reduction step; we believe that it is confluent. Maybe, some kind of "universal construction" approach could spare us a technical nightmare.

12. Actually, the construction works for any calculus that can be meaningfully embedded in Multiplicative Exponential Linear Logic.

4. Subtyping-Aware Polyadic Calculus and Rigid Expansion of Ordinary λ -Terms – 4.4. An Example: the Linear Case

The former intuitive presentation is very reminiscent of the structures considered in [TAO17] and, in greater generality, in the present work. We believe that an eventual extension of our approach to that framework would determine the lifting of the approximation presheaf from a lax morphism to an appropriate pseudofunctor. This would be a relevant improvement: in that case, the approximation presheaf would describe a bicategorical denotational semantics. Moreover, it would then be possible to exploit the Grothendieck construction of [Maz17][pp. 59-63] in all its generality, without the need to limit oneself to discrete categories, adding in his way the subtyping feature to the considered intersection type systems and approximants.

This perspective would be very beneficial also for our work: Mazza *et al.* framework is much broader than ordinary λ -calculus and the 2-operadic point of view is a very natural setting for bidimensional categorical considerations.

4. Another interesting issue arises in the context of resource monads. The relationship between the linear resource monad and the cartesian one is reminiscent of what happens in the one-dimensional framework between the relational model and the Scott model of linear logic [Ehr12b; Mel04]. If we lift to distributors, we get that the bicategory $\mathcal{S}\text{-CatSym}$, where \mathcal{S} is the cartesian resource monad, is biequivalent by dualization to the bicategory $\mathbf{Prof}_{\mathcal{C}}$ introduced in [Gal20], which generalizes the Scott model of linear logic. Hence, it is natural to consider the question of some kind of "extensional collapse" in the sense of [Ehr12b] in this categorified setting.
5. The rigid Taylor expansion semantics [TAO17] has been extended to quantum and probabilistic computation in [TAO18], where a notion of *weighted profunctor* is introduced. This result encourages a possible extension of the results of our thesis at least to the probabilistic λ -calculus [Lev16] and the algebraic one [Vau09]. Much of the classic results of advanced category theory presented in Chapter 2 have a natural generalization to enriched category theory. However, there would clearly be some highly non-trivial work on (relative) pseudomonads and enriched bicategories to be done.
6. Finally, an extension of the present semantics to the context of Multiplicative Exponential Linear Logic could eventually lead to the definition of a deterministic reduction relation on some kind of refined proof-nets *experiments* [Fal03; CPF11], shedding some light on a long-standing problem. Indeed, intuitively an experiment is to a proof-net what a type derivation in an appropriate intersection type system is to a λ -term. Hence, it is natural to think that the proof-nets semantics corresponding to our construction should somehow generalize the notion of experiment.

Bibliography

- [ABD06] F. Alessi, F. Barbanera, and M. Dezani-Ciancaglini. “Intersection types and lambda models”. In: *Theoretical Computer Science* 355.2 (2006). Logic, Language, Information and Computation, pp. 108–126. DOI: <https://doi.org/10.1016/j.tcs.2006.01.004> (cit. on p. 132).
- [AC98] R. M. Amadio and P.-L. Curien. *Domains and Lambda-calculi*. New York, NY, USA: Cambridge University Press, 1998. ISBN: 0-521-62277-8 (cit. on pp. 15, 102).
- [BM20] D. Barbarossa and G. Manzonetto. “Taylor subsumes Scott, Berry, Kahn and Plotkin”. In: *Proc. ACM Program. Lang.* 4.POPL (2020), 1:1–1:23. DOI: [10.1145/3371069](https://doi.org/10.1145/3371069) (cit. on p. 19).
- [Bén00] J. Bénabou. “Distributors at Work”. In: 2000. URL: <http://www.entretemps.asso.fr/maths/Distributors.pdf> (cit. on pp. 15, 88).
- [BL13] A. Bernadet and S. J. Lengrand. “Non-idempotent intersection types and strong normalisation”. In: *Logical Methods in Computer Science* Volume 9, Issue 4 (2013). DOI: [10.2168/LMCS-9\(4:3\)2013](https://doi.org/10.2168/LMCS-9(4:3)2013) (cit. on p. 12).
- [BKP89] R. Blackwell, G. M. Kelly, and A. J. Power. “Two-dimensional monad theory”. In: *Journal of Pure and Applied Algebra* 59.1 (1989), pp. 1–41. ISSN: 0022-4049. DOI: [https://doi.org/10.1016/0022-4049\(89\)90160-6](https://doi.org/10.1016/0022-4049(89)90160-6) (cit. on pp. 61, 62, 84).
- [Bor94] F. Borceux. *Handbook of Categorical Algebra*. Vol. 1. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 1994. DOI: [10.1017/CB09780511525858](https://doi.org/10.1017/CB09780511525858) (cit. on pp. 54, 88).
- [BKV17] A. Bucciarelli, D. Kesner, and D. Ventura. “Non-idempotent intersection types for the Lambda-Calculus”. In: *Logic Journal of the IGPL* 25.4 (2017), pp. 431–464. DOI: [10.1093/jigpal/jzx018](https://doi.org/10.1093/jigpal/jzx018) (cit. on pp. 12, 13).
- [Bun94] A. Bunkenburg. “The Boom Hierarchy”. In: *Functional Programming, Glasgow 1993: Proceedings of the 1993 Glasgow Workshop on Functional Programming, Ayr, Scotland, 5–7 July 1993*. Ed. by J. T. O’Donnell and K. Hammond. London: Springer London, 1994, pp. 1–8. ISBN: 978-1-4471-3236-3. DOI: [10.1007/978-1-4471-3236-3_1](https://doi.org/10.1007/978-1-4471-3236-3_1) (cit. on p. 78).

- [Car07] D. de Carvalho. “Semantique de la logique lineaire et temps de calcul”. In: PhD thesis, Aix-Marseille Université, 2007 (cit. on pp. 12, 13, 95, 121, 127).
- [CF16] D. de Carvalho and L. T. de Falco. “A semantic account of strong normalization in linear logic”. In: *Inf. Comput.* 248 (2016), pp. 104–129. DOI: [10.1016/j.ic.2015.12.010](https://doi.org/10.1016/j.ic.2015.12.010) (cit. on p. 13).
- [CPF11] D. de Carvalho, M. Pagani, and L. T. de Falco. “A semantic measure of the execution time in linear logic”. In: *Theoretical Computer Science* 412.20 (2011), pp. 1884–1902. ISSN: 0304-3975. DOI: <https://doi.org/10.1016/j.tcs.2010.12.017> (cit. on pp. 13, 173).
- [CW05] G. L. Cattani and G. Winskel. “Profunctors, open maps and bisimulation”. In: *Mathematical Structures in Computer Science* 15.3 (2005), pp. 553–614. DOI: [10.1017/S0960129505004718](https://doi.org/10.1017/S0960129505004718) (cit. on p. 15).
- [Cop+84] M. Coppo, M. Dezani-Ciancaglini, F. Honsell, and G. Longo. “Extended Type Structures and Filter Lambda Models”. In: *Logic Colloquium ’82*. Vol. 112. Studies in Logic and the Foundations of Mathematics. Elsevier, 1984, pp. 241–262. DOI: [https://doi.org/10.1016/S0049-237X\(08\)71819-6](https://doi.org/10.1016/S0049-237X(08)71819-6) (cit. on p. 13).
- [CD78] M. Coppo and M. Dezani-Ciancaglini. “A new type-assignment for lambda terms”. In: *Archiv für Mathematische Logik und Grundlagenforschung*. 1978, pp. 139–156 (cit. on pp. 12, 127).
- [Day70] B. Day. “On closed categories of functors”. In: *Reports of the Midwest Category Seminar IV*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1970, pp. 1–38. ISBN: 978-3-540-36292-0 (cit. on p. 74).
- [Ehr05] T. Ehrhard. “Finiteness spaces”. In: *Mathematical Structures in Computer Science* 15.4 (2005). DOI: [10.1017/S0960129504004645](https://doi.org/10.1017/S0960129504004645) (cit. on pp. 12, 23).
- [Ehr10] T. Ehrhard. “A Finiteness Structure on Resource Terms”. In: *LICS 2010*. 2010. DOI: [10.1109/LICS.2010.38](https://doi.org/10.1109/LICS.2010.38) (cit. on pp. 19, 23).
- [Ehr12a] T. Ehrhard. “Collapsing non-idempotent intersection types”. In: *Computer Science Logic (CSL’12) - 26th International Workshop/21st Annual Conference of the EACSL, CSL 2012*. Vol. 16. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2012, pp. 259–273. DOI: [10.4230/LIPIcs.CSL.2012.259](https://doi.org/10.4230/LIPIcs.CSL.2012.259) (cit. on pp. 86, 87, 95).
- [Ehr12b] T. Ehrhard. “The Scott model of linear logic is the extensional collapse of its relational model”. In: *Theoretical Computer Science* 424 (2012), pp. 20–45. ISSN: 0304-3975. DOI: <https://doi.org/10.1016/j.tcs.2011.11.027> (cit. on pp. 13, 173).

- [Ehr16] T. Ehrhard. “Call-By-Push-Value from a Linear Logic Point of View”. In: *Programming Languages and Systems*. Ed. by P. Thiemann. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016, pp. 202–228. ISBN: 978-3-662-49498-1 (cit. on pp. 13, 86).
- [ER03] T. Ehrhard and L. Regnier. “The differential lambda-calculus”. In: *Theoretical Computer Science* 309.1-3 (2003). DOI: [10.1016/S0304-3975\(03\)00392-X](https://doi.org/10.1016/S0304-3975(03)00392-X) (cit. on pp. 12, 23).
- [ER06a] T. Ehrhard and L. Regnier. “Böhm Trees, Krivine’s Machine and the Taylor Expansion of Lambda-Terms”. In: *CiE 2005*. 2006. DOI: [10.1007/11780342_20](https://doi.org/10.1007/11780342_20) (cit. on pp. 12, 19, 23).
- [ER06b] T. Ehrhard and L. Regnier. “Differential Interaction Nets”. In: *Theoretical Computer Science* 364.2 (2006). DOI: [10.1016/j.tcs.2006.08.003](https://doi.org/10.1016/j.tcs.2006.08.003) (cit. on p. 12).
- [ER08] T. Ehrhard and L. Regnier. “Uniformity and the Taylor Expansion of ordinary λ -terms”. In: *Theoretical Computer Science* 403.2-3 (2008). DOI: [10.1016/j.tcs.2008.06.001](https://doi.org/10.1016/j.tcs.2008.06.001) (cit. on pp. 12, 13, 16, 18, 19, 23, 27, 28, 30, 169).
- [Fal03] L. T. de Falco. “Obsessional Experiments For Linear Logic Proof-Nets”. In: *Mathematical Structures in Computer Science* 13.6 (2003), pp. 799–855. DOI: [10.1017/S0960129503003967](https://doi.org/10.1017/S0960129503003967) (cit. on p. 173).
- [Fio+08] M. Fiore, N. Gambino, M. Hyland, and G. Winskel. “The cartesian closed bicategory of generalised species of structures”. In: *Journal of the London Mathematical Society* (2008). DOI: [10.1112/jlms/jdm096](https://doi.org/10.1112/jlms/jdm096) (cit. on pp. 14, 15, 20, 33, 89, 92, 94).
- [Fio+17] M. Fiore, N. Gambino, M. Hyland, and G. Winskel. “Relative pseudomonads, Kleisli bicategories, and substitution monoidal structures”. In: *Selecta Mathematica* 24.3 (Nov. 2017), pp. 2791–2830. ISSN: 1420-9020. DOI: [10.1007/s00029-017-0361-3](https://doi.org/10.1007/s00029-017-0361-3) (cit. on pp. 14, 16, 47, 63, 64, 67, 89, 94).
- [FS19] M. Fiore and P. Saville. “A type theory for cartesian closed bicategories (Extended Abstract)”. In: *34th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2019, Vancouver, BC, Canada, June 24-27, 2019*. IEEE, 2019, pp. 1–13. ISBN: 978-1-7281-3608-0. DOI: [10.1109/LICS.2019.8785708](https://doi.org/10.1109/LICS.2019.8785708). URL: <https://ieeexplore.ieee.org/xpl/conhome/8765678/proceeding> (cit. on p. 59).
- [FS20] M. Fiore and P. Saville. “Coherence and Normalisation-by-Evaluation for Bicategorical Cartesian Closed Structure”. In: *Proceedings of the 35th Annual ACM/IEEE Symposium on Logic in Computer Science*. LICS ’20. Saarbrücken, Germany: Association for Computing Machinery, 2020,

- pp. 425–439. ISBN: 9781450371049. DOI: [10.1145/3373718.3394769](https://doi.org/10.1145/3373718.3394769) (cit. on p. 59).
- [Fio06] T. M. Fiore. “Pseudo limits, biadjoints, and pseudo algebras: categorical foundations of conformal field theory”. In: *Mem. Amer. Math. Soc* (2006) (cit. on p. 58).
- [Gal20] Z. Galal. “A Profunctorial Scott Semantics”. In: *5th International Conference on Formal Structures for Computation and Deduction (FSCD 2020)*. Ed. by Z. M. Ariola. Vol. 167. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2020, 16:1–16:18. ISBN: 978-3-95977-155-9. DOI: [10.4230/LIPIcs.FSCD.2020.16](https://doi.org/10.4230/LIPIcs.FSCD.2020.16) (cit. on p. 173).
- [GJ17] N. Gambino and A. Joyal. “On operads, bimodules and analytic functors”. In: *Memoirs of the American Mathematical Society* 249.1184 (Sept. 2017). ISSN: 1947-6221. DOI: [10.1090/memo/1184](https://doi.org/10.1090/memo/1184) (cit. on pp. 14, 59, 88, 89, 92, 94).
- [Gar94] P. Gardner. “Discovering needed reductions using type theory”. In: *Theoretical Aspects of Computer Software*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1994, pp. 555–574. ISBN: 978-3-540-48383-0 (cit. on pp. 12, 127).
- [Gir87] J.-Y. Girard. “Linear Logic”. In: *Theor. Comput. Sci.* 50 (1987), pp. 1–102 (cit. on pp. 11, 28).
- [Gir88] J.-Y. Girard. “Normal Functors, Power Series and Lambda-Calculus”. In: *Annals of Pure and Applied Logic* 37.2 (1988), p. 129 (cit. on p. 12).
- [Has02] R. Hasegawa. “Two applications of analytic functors”. In: *Theor. Comput. Sci.* 272.1-2 (2002), pp. 113–175. DOI: [10.1016/S0304-3975\(00\)00349-2](https://doi.org/10.1016/S0304-3975(00)00349-2) (cit. on p. 12).
- [Hir13] T. Hirschowitz. “Cartesian closed 2-categories and permutation equivalence in higher-order rewriting”. In: *Log. Methods Comput. Sci.* 9.3 (2013). DOI: [10.2168/LMCS-9\(3:10\)2013](https://doi.org/10.2168/LMCS-9(3:10)2013) (cit. on pp. 13, 97).
- [Hyl17] M. Hyland. “Classical lambda calculus in modern dress”. In: *Mathematical Structures in Computer Science* 27.5 (2017), pp. 762–781. DOI: [10.1017/S0960129515000377](https://doi.org/10.1017/S0960129515000377) (cit. on p. 13).
- [Hyl+06] M. Hyland, M. Nagayama, J. Power, and G. Rosolini. “A Category Theoretic Formulation for Engeler-style Models of the Untyped λ -Calculus”. In: *Electronic Notes in Theoretical Computer Science* 161 (2006). Proceedings of the Third Irish Conference on the Mathematical Foundations of Computer Science and Information Technology (MFCSIT 2004), pp. 43–57. DOI: <https://doi.org/10.1016/j.entcs.2006.04.024> (cit. on p. 13).

- [IK86] G. B. IM and G. Kelly. “A universal property of the convolution monoidal structure”. In: *Journal of Pure and Applied Algebra* 43.1 (1986), pp. 75–88. DOI: [https://doi.org/10.1016/0022-4049\(86\)90005-8](https://doi.org/10.1016/0022-4049(86)90005-8) (cit. on pp. 72, 73, 77).
- [Joy86] A. Joyal. “Foncteurs analytiques et espèces de structures”. In: *Combinatoire énumérative*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1986, pp. 126–159 (cit. on pp. 15, 94).
- [Kel80] G. M. Kelly. “A unified treatment of transfinite constructions for free algebras, free monoids, colimits, associated sheaves, and so on”. In: *Bulletin of the Australian Mathematical Society* 22.1 (1980), pp. 1–83. DOI: [10.1017/S0004972700006353](https://doi.org/10.1017/S0004972700006353) (cit. on p. 101).
- [Kri93] J.-L. Krivine. “Lambda-calculus, types and models”. In: *Ellis Horwood series in computers and their applications*. 1993 (cit. on pp. 12, 16, 121, 125).
- [Lac00] S. Lack. “A Coherent Approach to Pseudomonads”. In: *Advances in Mathematics* 152.2 (2000), pp. 179–202. ISSN: 0001-8708. DOI: <https://doi.org/10.1006/aima.1999.1881> (cit. on p. 61).
- [LL19] U. D. Lago and T. Leventis. “On the Taylor Expansion of Probabilistic λ -terms”. In: *4th International Conference on Formal Structures for Computation and Deduction, FSCD 2019, June 24-30, 2019, Dortmund, Germany*. Ed. by H. Geuvers. Vol. 131. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019, 13:1–13:16. DOI: [10.4230/LIPIcs.FSCD.2019.13](https://doi.org/10.4230/LIPIcs.FSCD.2019.13) (cit. on p. 20).
- [Lam92] F. Lamarche. “Quantitative Domains and Infinitary Algebras”. In: *Theor. Comput. Sci.* 94.1 (1992), pp. 37–62. DOI: [10.1016/0304-3975\(92\)90323-8](https://doi.org/10.1016/0304-3975(92)90323-8) (cit. on p. 12).
- [LS86] J. Lambek and P. J. Scott. *Introduction to Higher Order Categorical Logic*. USA: Cambridge University Press, 1986. ISBN: 0521246652 (cit. on p. 11).
- [Lan02] S. Lang. *Algebra*. Graduate Texts in Mathematics. Springer New York, 2002. ISBN: 9780387953854 (cit. on p. 22).
- [Lei98] T. Leinster. *Basic Bicategories*. 1998. arXiv: [math/9810017](https://arxiv.org/abs/math/9810017) [math.CT] (cit. on p. 57).
- [Lei03] T. Leinster. *Higher Operads, Higher Categories*. 2003. arXiv: [math/0305049](https://arxiv.org/abs/math/0305049) [math.CT] (cit. on pp. 52, 62).
- [Lev16] T. Leventis. “Probabilistic lambda-theories”. Theses. Aix-Marseille Université, Dec. 2016. URL: <https://tel.archives-ouvertes.fr/tel-01427279> (cit. on pp. 23, 173).

- [Lev19] T. Leventis. “A deterministic rewrite system for the probabilistic λ -calculus”. In: *Mathematical Structures in Computer Science* 29.10 (2019), pp. 1479–1512. DOI: [10.1017/S0960129519000045](https://doi.org/10.1017/S0960129519000045) (cit. on pp. 23, 24).
- [Lor15] F. Loregian. *Coend calculus*. 2015. arXiv: [1501.02503](https://arxiv.org/abs/1501.02503) [math.CT] (cit. on pp. 69–71).
- [Luc18] F. Lucatelli Nunes. “On lifting of biadjoints and lax algebras”. In: *Categories and General Algebraic Structures with Applications* 9.1 (2018), pp. 29–58. ISSN: 2345-5853. eprint: http://cgasa.sbu.ac.ir/article_50747_e7751692a69d525e49259ebe2763142f.pdf. URL: http://cgasa.sbu.ac.ir/article_50747.html (cit. on p. 61).
- [MZ18] D. Marsden and M. Zwart. “Quantitative Foundations for Resource Theories”. In: *27th EACSL Annual Conference on Computer Science Logic, CSL 2018, September 4-7, 2018, Birmingham, UK*. Ed. by D. R. Ghica and A. Jung. Vol. 119. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018, 32:1–32:17. ISBN: 978-3-95977-088-0. DOI: [10.4230/LIPIcs.CSL.2018.32](https://doi.org/10.4230/LIPIcs.CSL.2018.32) (cit. on pp. 14, 47, 79, 96).
- [Maz12] D. Mazza. “An Infinitary Affine Lambda-Calculus Isomorphic to the Full Lambda-Calculus”. In: *2012 27th Annual IEEE Symposium on Logic in Computer Science*. 2012, pp. 471–480 (cit. on pp. 15, 18, 20).
- [Maz17] D. Mazza. “Polyadic Approximations in Logic and Computation”. *Habilitation thesis*. Université Paris 13, 2017 (cit. on pp. 14, 15, 97, 172, 173).
- [MPV18] D. Mazza, L. Pellissier, and P. Vial. “Polyadic approximations, fibrations and intersection types”. In: 2018. DOI: [10.1145/3158094](https://doi.org/10.1145/3158094) (cit. on pp. 13, 14, 18, 20, 97, 101, 135, 172).
- [Mel04] P.-A. Melliès. “Comparing hierarchies of types in models of linear logic”. In: *Information and Computation* 189.2 (2004), pp. 202–234. ISSN: 0890-5401. DOI: <https://doi.org/10.1016/j.ic.2003.10.003> (cit. on p. 173).
- [Mel09] P.-A. Melliès. “Categorical semantics of linear logic”. In: *In: Interactive Models of Computation and Program Behaviour, Panoramas et Synthèses 27, Société Mathématique de France 1–196*. 2009 (cit. on pp. 11, 47, 85).
- [MZ15] P.-A. Melliès and N. Zeilberger. “Functors Are Type Refinement Systems”. In: *SIGPLAN Not.* 50.1 (Jan. 2015), pp. 3–16. ISSN: 0362-1340. DOI: [10.1145/2775051.2676970](https://doi.org/10.1145/2775051.2676970) (cit. on pp. 13, 14).
- [OA20] F. Olimpieri and L. V. Auclair. *On the Taylor expansion of λ -terms and the groupoid structure of their rigid approximants*. 2020. arXiv: [2008.02665](https://arxiv.org/abs/2008.02665) [cs.LG] (cit. on p. 18).

- [OG21] F. Olimpieri and G. Guerrieri. “Categorifying Non-Idempotent Intersection Types”. In: *To appear in the proceedings of the 29th International Conference Computer Science Logic, CSL 2021*. 2021 (cit. on p. 172).
- [Ong17] C.-H. L. Ong. “Quantitative semantics of the lambda calculus: Some generalisations of the relational model”. In: *32nd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2017, Reykjavik, Iceland, June 20-23, 2017*. IEEE Computer Society, 2017, pp. 1–12. DOI: [10.1109/LICS.2017.8005064](https://doi.org/10.1109/LICS.2017.8005064) (cit. on p. 13).
- [PTV16] M. Pagani, C. Tasson, and L. Vaux. “Strong Normalizability as a Finiteness Structure via the Taylor Expansion of λ -terms”. In: *FoSSaCS 2016*. 2016. DOI: [10.1007/978-3-662-49630-5_24](https://doi.org/10.1007/978-3-662-49630-5_24) (cit. on pp. 19, 23).
- [Sav20] P. Saville. *Cartesian closed bicategories: type theory and coherence*. 2020. arXiv: [2007.00624](https://arxiv.org/abs/2007.00624) [math.CT] (cit. on pp. 58, 59).
- [Sco70] D. S. Scott. “Outline of a mathematical theory of computation”. In: Oxford, United Kingdom: Oxford University Computing Laboratory, 1970 (cit. on p. 11).
- [SS71] D. S. Scott and C. Strachey. “Toward a mathematical semantics for computer languages”. In: *Oxford Programming Research Group Technical Monograph* (1971) (cit. on p. 11).
- [See87] R. A. G. Seely. “Modelling Computations: A 2-Categorical Framework”. In: *LICS*. 1987 (cit. on pp. 13, 97).
- [Suz82] M. Suzuki. *Group theory I*. Springer-Verlag Berlin ; New York, 1982. ISBN: 3642618065 (cit. on p. 22).
- [Ter12] K. Terui. “Semantic Evaluation, Intersection Types and Complexity of Simply Typed Lambda Calculus”. In: *23rd International Conference on Rewriting Techniques and Applications (RTA’12)*. Ed. by A. Tiwari. Vol. 15. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2012, pp. 323–338. ISBN: 978-3-939897-38-5. DOI: [10.4230/LIPIcs.RTA.2012.323](https://doi.org/10.4230/LIPIcs.RTA.2012.323) (cit. on p. 95).
- [TAO17] T. Tsukada, K. Asada, and C.-H. L. Ong. “Generalised Species of Rigid Resource Terms”. In: *Proceedings of the 32nd Annual ACM/IEEE Symposium on Logic in Computer Science*. LICS 2017. 2017. DOI: [10.1109/LICS.2017.8005093](https://doi.org/10.1109/LICS.2017.8005093) (cit. on pp. 14, 20, 25, 134, 135, 161, 173).

- [TAO18] T. Tsukada, K. Asada, and C.-H. L. Ong. “Species, Profunctors and Taylor Expansion Weighted by SMCC: A Unified Framework for Modelling Nondeterministic, Probabilistic and Quantum Programs”. In: *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science*. LICS '18. Oxford, United Kingdom, 2018, pp. 889–898. ISBN: 978-1-4503-5583-4. DOI: [10.1145/3209108.3209157](https://doi.org/10.1145/3209108.3209157) (cit. on pp. [20](#), [173](#)).
- [Vau09] L. Vaux. “The algebraic lambda calculus”. In: *Mathematical Structures in Computer Science* 19.5 (2009), pp. 1029–1059. DOI: [10.1017/S0960129509990089](https://doi.org/10.1017/S0960129509990089) (cit. on pp. [20](#), [173](#)).
- [Vau17] L. Vaux. “Taylor Expansion, β -Reduction and Normalization”. In: *Computer Science Logic 2017*. 2017. DOI: [10.4230/LIPIcs.CSL.2017.39](https://doi.org/10.4230/LIPIcs.CSL.2017.39) (cit. on pp. [19](#), [20](#), [23](#), [31](#)).

Appendix

A. Technical Proofs of Chapter 3

Lemma A.1. *Let $M \in \Lambda$, $\vec{x} \supset \text{fv}(M)$ and $x \in \mathcal{V}$. We have the following natural isomorphism*

$$\llbracket M \rrbracket_{\vec{x}}(\Delta, a) \cong \llbracket M \rrbracket_{\vec{x} \oplus \langle \rangle}(\Delta \oplus \langle \rangle, a)$$

Lemma A.2. *let $M \in \Lambda$, $\vec{x} \supset \text{fv}(M)$, $i, j \in \text{len}(\vec{x})$ and $\{\tau_{i,j}\} : \text{len}(\vec{x}) \rightarrow \text{len}(\vec{x})$ be the permutation that swaps i with j . We have the following natural isomorphism*

$$\llbracket M \rrbracket_{\vec{x}}(\Delta, a) \cong \llbracket M \rrbracket_{\vec{x}\{\tau_{i,j}\}}(\Delta\{\tau_{i,j}\}, a).$$

Lemma A.3. *Let $M \in \Lambda$, $\vec{x} \supset \text{fv}(M)$ and $y \notin \vec{x}$. We have the following natural isomorphism*

$$\llbracket M \rrbracket_{\vec{x}}(\Delta \oplus \langle \diamond_{\vec{b}} \rangle, a) : \llbracket M \rrbracket_{\vec{x} \oplus \langle y \rangle}(\Delta \oplus \langle \rangle, a) \cong \llbracket M \rrbracket_{\vec{x} \oplus \langle y \rangle}(\Delta \oplus \langle \vec{b} \rangle, a)$$

for all $\vec{b} \in SD$.

Proof. The proof is a straightforward induction on the structure of M , via coend manipulations. The hypothesis that $\diamond_{\vec{b}}$ is suitably universal is necessary, as expected. \square

Let $M, N \in \Lambda$, $(\text{fv}(M)/\{x\}) \cup \text{fv}(N) \supseteq \vec{x}$ and $x \notin \vec{x}$. We set $Sub_{\vec{x}}^{M,x,N}(\Delta, a) =$

$$\int^{\vec{a} \in SD} \int^{\Gamma_1, \dots, \Gamma_{\text{len}(\vec{a})} \in SD^n} \llbracket M \rrbracket_{\vec{x} \oplus \langle \vec{a} \rangle}(\Gamma_0 \oplus \langle \vec{a} \rangle, a) \times \prod_{i=1}^{l(\vec{a})} \llbracket N \rrbracket_{\vec{x}}(\Gamma_i, a_i) \times SD^n(\Delta, \bigotimes_{j=0}^k \Gamma_j).$$

We can now prove the following substitution lemma:

Lemma A.4. *Let $M, N \in \Lambda$, $(\text{fv}(M)/\{x\}) \cup \text{fv}(N) \subseteq \vec{x}$ and $x \notin \vec{x}$. We have a natural isomorphism*

$$\llbracket M[N/x] \rrbracket_{\vec{x}}(\Delta, a) \cong Sub_{\vec{x}}^{M,x,N}(\Delta, a).$$

Proof. By Induction on the structure of $M\{N/x\}$ via lengthy coend manipulations. It is worth noting that, in the proof of the application case, the hypothesis about the symmetry of the tensor product over SD is crucial, as expected. We do the variable case. Let $M = x$, then $M\{N/x\} = N$ and

$$\begin{aligned}
 & Sub_{\vec{x}}^{M,x,N}(\Delta, a) \cong \\
 & \int^{\vec{a}, \Gamma_j} SD^n(\Gamma_0, \langle \langle \rangle, \dots, \langle \rangle \rangle) \times SD(\vec{a}, \langle a \rangle) \\
 & \times \prod_{i \in [\text{len}(\vec{a})]} \llbracket N \rrbracket_{\vec{x}}(\Gamma_i, a_i) \times SD^n(\Delta, \bigotimes_{j=0}^{\text{len}(\vec{a})} \Gamma_j)
 \end{aligned}$$

We apply Yoneda twice and, since the number of contexts Γ_i depends on the length of the list, we get

$$\int^{\Gamma_0} \llbracket N \rrbracket_{\vec{x}}(\Gamma_0, a) \times SD^n(\Delta, \Gamma_0)$$

We can then conclude by applying Yoneda one more time.

If $M = y$ and $y \neq x$ then $M\{N/x\} = y$ and

$$\begin{aligned}
 & Sub_{\vec{x}}^{M,x,N}(\Delta, a) \cong \\
 & \int^{\vec{a}, \Gamma_j} SD^n(\Gamma_0, \langle \langle \rangle, \dots, \langle a \rangle, \dots, \langle \rangle \rangle) \times SD(\vec{a}, \langle \rangle) \\
 & \times \prod_{i \in [\text{len}(\vec{a})]} \llbracket N \rrbracket_{\vec{x}}(\Gamma_i, a_i) \times SD^n(\Delta, \bigotimes_{j=0}^{\text{len}(\vec{a})} \Gamma_j)
 \end{aligned}$$

we apply Yoneda twice and we get

$$\int^{\Gamma_0} SD^n(\Gamma_0, \langle \langle \rangle, \dots, \langle a \rangle, \dots, \langle \rangle \rangle) \times SD^n(\Delta, \Gamma_0)$$

we conclude applying Yoneda one more time.

If $M = \lambda x.M'$ then $Sub_{\vec{x}}^{M,x,N}(\Delta, \vec{b} \Rightarrow b) =$

$$\int^{\vec{a} = \langle a_1, \dots, a_k \rangle \in SD} \int^{\Gamma_j \in SD^n} \llbracket \lambda y.M' \rrbracket_{\vec{x} \oplus \langle x \rangle}(\Gamma_0 \oplus \langle \vec{a} \rangle, \vec{b} \Rightarrow b) \times \prod_{i=1}^{l(\vec{a})} \llbracket N \rrbracket_{\vec{x}}(\Gamma_i, a_i) \times SD^n(\Delta, \bigotimes_{j=0}^k \Gamma_j) =$$

$$\int^{\vec{a} \in SD} \int^{\Gamma_j \in SD^n} \llbracket M' \rrbracket_{\vec{x} \oplus \langle x \rangle \oplus \langle y \rangle}(\Gamma_0 \oplus \langle \vec{a} \rangle \oplus \langle \vec{b} \rangle, b) \times \prod_{i=1}^{l(\vec{a})} \llbracket N \rrbracket_{\vec{x}}(\Gamma_i, a_i) \times SD^n(\Delta, \bigotimes_{j=0}^k \Gamma_j)$$

by Lemma A.1

$$\cong \int^{\vec{a} \in SD} \int^{\Gamma_j \in SD^n} \llbracket M' \rrbracket_{\vec{x} \oplus \langle x \rangle \oplus \langle y \rangle}(\Gamma_0 \oplus \langle \vec{a} \rangle \oplus \langle \vec{b} \rangle, b) \times \prod_{i=1}^{l(\vec{a})} \llbracket N \rrbracket_{\vec{x}, y}(\Gamma_i \oplus \langle \rangle, a_i) \times SD^{n+1}(\Delta, \bigotimes_{j=0}^k \Gamma_j)$$

by Yoneda

$$\cong \int^{\vec{a}, \vec{c} \in SD} \int^{\Gamma_j \in SD^n} \llbracket M' \rrbracket_{\vec{x} \oplus (x) \oplus (y)}(\Gamma_0 \oplus \langle \vec{a} \rangle \oplus \langle \vec{c} \rangle, b) \times \prod_{i=1}^{l(\vec{a})} \llbracket N \rrbracket_{\vec{x}, y}(\Gamma_i \oplus \langle \rangle, a_i) \times SD^n(\Delta, \bigotimes_{j=0}^k \Gamma_j) \times SD(\vec{b}, \vec{c})$$

by the former lemma

$$\cong \int^{\vec{a}, \vec{c} \in SD} \int^{\Gamma_j \in SD^n} \llbracket M' \rrbracket_{\vec{x} \oplus (y) \oplus (x)}(\Gamma_0 \oplus \langle \vec{c} \rangle \oplus \langle \vec{a} \rangle, b) \times \prod_{i=1}^{l(\vec{a})} \llbracket N \rrbracket_{\vec{x}, y}(\Gamma_i \oplus \langle \rangle, a_i) \times SD^n(\Delta, \bigotimes_{j=0}^k \Gamma_j) \times SD(\vec{b}, \vec{c})$$

By IH we have

$$\begin{aligned} & \llbracket M' \{N/x\} \rrbracket_{\vec{x} \oplus (y)}(\Delta, a) \cong \\ & \int^{\vec{a}, \vec{b}_j \in SD} \int^{\Gamma_j \in SD^n} \llbracket M' \rrbracket_{\vec{x} \oplus (y) \oplus (x)}(\Gamma_0 \oplus \langle \vec{b}_0 \rangle \oplus \langle \vec{a} \rangle, b) \times \prod_{i=1}^{l(\vec{a})} \llbracket N \rrbracket_{\vec{x} \oplus (y)}(\Gamma_i \oplus \langle \vec{b}_i \rangle, a_i) \\ & \quad \times SD^n(\Delta, \bigotimes_{j=0}^k \Gamma_j) \times SD(\vec{b}, \bigoplus_{j=0}^k \vec{b}_j) \end{aligned}$$

Then we can conclude by Lemma A.3

$$\begin{aligned} & \cong \int^{\vec{a}, \vec{b}_j \in SD} \int^{\Gamma_j \in SD^n} \llbracket M' \rrbracket_{\vec{x} \oplus (y) \oplus (x)}(\Gamma_0 \oplus \langle \vec{b}_0 \rangle \oplus \langle \vec{a} \rangle, b) \times \prod_{i=1}^{l(\vec{a})} \llbracket N \rrbracket_{\vec{x} \oplus (y)}(\Gamma_i \oplus \langle \rangle, a_i) \\ & \quad \times SD^n(\Delta, \bigotimes_{j=0}^k \Gamma_j) \times SD(\vec{b}, \vec{b}_0). \end{aligned}$$

If $M = QP$ then

$$Sub_{M, x, N}(\Delta, a) = \int^{\vec{a} = \langle a_1, \dots, a_k \rangle} \int^{\Gamma_0, \Gamma_i} \llbracket QP \rrbracket_{\vec{x} \oplus (x)}(\Gamma_0 \oplus \langle \vec{a} \rangle, a) \times \prod_{i=1}^k \llbracket N \rrbracket_{\vec{x}}(\Gamma_i, a_i) \times SD^n(\Delta, \bigotimes_{j=0}^k \Gamma_j).$$

with $i \in [k]$ and $0 \leq j \leq k$. We develop $\llbracket QP \rrbracket_{\vec{x} \oplus (x)}(\Gamma_0 \oplus \langle \vec{a} \rangle, a)$:

$$\begin{aligned} & \int^{\vec{b} = \langle b_1, \dots, b_{k'} \rangle} \int^{\Gamma'_0 \oplus \langle \vec{a}_0 \rangle, \Gamma'_{i'} \oplus \langle \vec{a}_{i'} \rangle} \llbracket Q \rrbracket_{\vec{x} \oplus (x)}(\Gamma'_1 \oplus \langle \vec{a}_1 \rangle, \iota(\vec{b}, a)) \times \prod_{i'=1}^{k'} \llbracket P \rrbracket_{\vec{x} \oplus (x)}(\Gamma'_{i'} \oplus \langle \vec{a}_{i'} \rangle, \vec{b}) \\ & \quad \times SD^n(\Gamma_0 \oplus \langle \vec{a} \rangle, \bigotimes_{j'=0}^{k'} \Gamma'_{j'} \oplus \langle \vec{a}_{j'} \rangle) \end{aligned}$$

with $i' \in [k']$ and $0 \leq j' \leq k'$. By the structure of the product category we get

$$\cong \int^{\vec{b}} \int^{\Gamma'_1 \Gamma'_{i'}} \int^{\vec{a}_0, \vec{a}_{i'}} \llbracket Q \rrbracket_{\vec{x} \oplus (x)}(\Gamma'_1 \oplus \langle \vec{a}_0 \rangle, \iota(\vec{b}, a)) \times \prod_{i'=1}^{k'} \llbracket P \rrbracket_{\vec{x} \oplus (x)}(\Gamma'_{i'} \oplus \langle \vec{a}_{i'} \rangle, b_{i'})$$

$$\times SD^n(\Gamma_0, \bigotimes_{j'=0}^{k'} \Gamma'_{j'}) \times SD(\vec{a}, \bigoplus_{j'=0}^{k'} \vec{a}_{j'}).$$

We apply Yoneda on Γ_0 and on \vec{a} and we get

$$\begin{aligned} Sub_{M,x,N}(\Delta, a) &\cong \int^{\Gamma_i} \int^{\vec{b}} \int^{\Gamma'_{j'}} \int^{\vec{a}'_{j'}} \llbracket Q \rrbracket_{\vec{x} \oplus \langle x \rangle}(\Gamma'_0 \oplus \langle \vec{a}_0 \rangle, \iota(\vec{b}, a)) \times \\ &\quad \prod_{i'=1}^{k'} \llbracket P \rrbracket_{\vec{x} \oplus \langle x \rangle}(\Gamma'_{i'} \oplus \langle \vec{a}_{i'} \rangle, \vec{b}) \times \prod_{i=1}^l \llbracket N \rrbracket_{\vec{x}}(\Gamma_i, c_i) \\ &\quad \times SD^n(\Delta, (\bigotimes_{j'=0}^{k'} \Gamma'_{j'}) \otimes \bigotimes_{i=1}^l \Gamma_i) \end{aligned}$$

where $i \in [l]$ and $\bigoplus_{j'=0}^{k'} \vec{a}_{j'} = \langle c_1, \dots, c_l \rangle$. We denote $\vec{\Gamma}_{j'} = \langle \Gamma_{j',1}, \dots, \Gamma_{j',k_{j'}} \rangle$ for $j' \in [k']$ the partition of $\langle \Gamma_1, \dots, \Gamma_l \rangle$ induced by the partition $\vec{a}_{j'} = \langle a_{j',1}, \dots, a_{j',k_{j'}} \rangle$ of $\langle c_1, \dots, c_l \rangle$. We set $\llbracket N \rrbracket_{\vec{x}}(\vec{\Gamma}_{j'}, \vec{a}_{j'}) = \prod_{i_{j'}=1}^{k_{j'}} \llbracket N \rrbracket_{\vec{x}}(\Gamma_{i_{j'}}, a_{i_{j'}})$. Then we get

$$\begin{aligned} &\cong \int^{\Gamma_i} \int^{\vec{b}} \int^{\Gamma'_{j'}} \int^{\vec{a}'_{j'}} \llbracket Q \rrbracket_{\vec{x} \oplus \langle x \rangle}(\Gamma'_0 \oplus \langle \vec{a}_0 \rangle, \iota(\vec{b}, a)) \times \prod_{i'=1}^{k'} \llbracket P \rrbracket_{\vec{x} \oplus \langle x \rangle}(\Gamma'_{i'} \oplus \langle \vec{a}_{i'} \rangle, b_{i'}) \\ &\quad \times \llbracket N \rrbracket_{\vec{x}}(\vec{\Gamma}_0, \vec{a}_0) \times \prod_{i'=1}^{k'} \llbracket N \rrbracket_{\vec{x}}(\vec{\Gamma}_{i'}, \vec{a}_{i'}) \\ &\quad \times SD^n(\Delta, (\bigotimes_{j'=0}^{k'} \Gamma'_{j'}) \otimes (\bigotimes_{i=1}^l \Gamma_i)). \end{aligned}$$

Now, by the *symmetry* of the tensor product \otimes and by the fact that functors preserves isomorphisms, we get

$$\begin{aligned} &\cong \int^{\Gamma_i} \int^{\vec{b}} \int^{\Gamma'_{j'}} \int^{\vec{a}'_{j'}} \llbracket Q \rrbracket_{\vec{x} \oplus \langle x \rangle}(\Gamma'_0 \oplus \langle \vec{a}_0 \rangle, \iota(\vec{b}, a)) \times \prod_{i'=1}^{k'} \llbracket P \rrbracket_{\vec{x} \oplus \langle x \rangle}(\Gamma'_{i'} \oplus \langle \vec{a}_{i'} \rangle, b_{i'}) \\ &\quad \times \llbracket N \rrbracket_{\vec{x}}(\vec{\Gamma}_0, \vec{a}_0) \times \prod_{i'=1}^{k'} \llbracket N \rrbracket_{\vec{x}}(\vec{\Gamma}_{i'}, \vec{a}_{i'}) \\ &\quad \times SD^n(\Delta, (\bigotimes_{j'=0}^{k'} (\Gamma'_{j'} \otimes \vec{\Gamma}_{j'}))) \end{aligned}$$

where we set $\Gamma'_{j'} \otimes \vec{\Gamma}_{j'} = \Gamma'_{j'} \otimes (\bigotimes_{i_{j'}=1}^{k_{j'}} \Gamma_{i_{j'}})$.

Now, we apply Yoneda several times and we obtain

$$\begin{aligned}
 &\cong \int^{\Gamma_i} \int^{\vec{b}} \int^{\Gamma'_{j'}} \int^{\vec{a}'_{j'}} \int^{\Delta_{j'}} \llbracket Q \rrbracket_{\vec{x} \oplus \langle x \rangle}(\Gamma'_0 \oplus \langle \vec{a}_0 \rangle, \iota(\vec{b}, a)) \times \prod_{i'=1}^{k'} \llbracket P \rrbracket_{\vec{x} \oplus \langle x \rangle}(\Gamma'_{i'} \oplus \langle \vec{a}_{i'} \rangle, b_{i'}) \\
 &\quad \times \llbracket N \rrbracket_{\vec{x}}(\vec{\Gamma}_0, \vec{a}_0) \times \prod_{i'=1}^{k'} \llbracket N \rrbracket_{\vec{x}}(\vec{\Gamma}_{i'}, \vec{a}_{i'}) \\
 &\quad \times SD^n(\Delta, \bigotimes_{j'=0}^{k'} \Delta_{j'}) \times SD^n(\Delta_0, \Gamma'_0 \otimes \vec{\Gamma}_0) \times \cdots \times SD^n(\Delta_{k_{j'}}, \Gamma'_{k_{j'}} \otimes \vec{\Gamma}_{k_{j'}})
 \end{aligned}$$

Then by cocontinuity and commutativity we have

$$\begin{aligned}
 &\cong \int^{\vec{b}} \int^{\Delta_{j'}} \int^{\vec{a}_0} \int^{\Gamma'_0, \vec{\Gamma}_0} \llbracket Q \rrbracket_{\vec{x} \oplus \langle x \rangle}(\Gamma'_0 \oplus \langle \vec{a}_0 \rangle, \iota(\vec{b}, a)) \times \llbracket N \rrbracket_{\vec{x}}(\vec{\Gamma}_0, \vec{a}_0) \times SD^n(\Delta_0, \Gamma'_0 \otimes \vec{\Gamma}_0) \\
 &\times \llbracket N \rrbracket_{\vec{x}}(\vec{\Gamma}_0, \vec{a}_0) \times \prod_{i'=1}^{k'} \left(\int^{\vec{a}_{i'}} \int^{\Gamma'_{i'}, \vec{\Gamma}_{i'}} \llbracket P \rrbracket_{\vec{x} \oplus \langle x \rangle}(\Gamma'_{i'} \oplus \langle \vec{a}_{i'} \rangle, b_{i'}) \times \llbracket N \rrbracket_{\vec{x}}(\vec{\Gamma}_{i'}, \vec{a}_{i'}) \times SD^n(\Delta_{i'}, \Gamma'_{i'} \otimes \vec{\Gamma}_{i'}) \right) \\
 &\quad \times SD^n(\Delta, \bigotimes_{j'=0}^{k'} \Delta_{j'})
 \end{aligned}$$

By definition, the former coend is just

$$\int^{\vec{b}} \int^{\Delta_0, \dots, \Delta_{k'}} Sub_{\vec{x}}^{Q, x, N}(\Delta_0, \iota(\vec{b}, a)) \times \prod_{i'=1}^{k'} Sub_{\vec{x}}^{P, x, N}(\Delta_{i'}, b_{i'}) \times SD^n(\Delta, \bigotimes_{j'=0}^{k'} \Delta_{j'}).$$

By definition, we have

$$\begin{aligned}
 \llbracket QP\{N/x\} \rrbracket_{\vec{x}}(\Delta, a) &= \int^{\vec{b}} \int^{\Delta_0, \dots, \Delta_{k'}} \llbracket Q\{N/x\} \rrbracket_{\vec{x}}(\Delta_0, \iota(\vec{b}, a)) \\
 &\quad \times \prod_{i'=1}^{k'} \llbracket P\{N/x\} \rrbracket_{\vec{x}}(\Delta_{i'}, b_{i'}) \times SD^n(\Delta, \bigotimes_{j'=0}^{k'} \Delta_{j'}).
 \end{aligned}$$

Then applying the IH we get natural isos $\llbracket Q\{N/x\} \rrbracket_{\vec{x}}(\Delta_0, \iota(\vec{b}, a)) \cong Sub_{Q, x, N}(\Delta_0, \iota(\vec{b}, a))$ and $\llbracket P\{N/x\} \rrbracket_{\vec{x}}(\Delta_{i'}, b_{i'}) \cong Sub_{P, x, N}(\Delta_{i'}, b_{i'})$. We can then conclude, since isos are preserved by the coend construction. \square

B. Technical Proofs of Chapter 4

B.1. Actions and Congruence under Substitution of Type Derivations

Lemma B.1. *Let $\pi \vdash_{\zeta^{\triangleright \vec{x}_i : \Delta^{\triangleright \vec{a}_i \oplus \vec{a}}}}^a \langle \eta^{f_i, f_j}, p \rangle$ with $i \neq j$ and $\vec{\rho} \vdash_{\vec{\zeta}^{\triangleright \langle \rangle_j : \vec{\Gamma}^{\triangleright \langle \rangle_j}}}^{\vec{a}_i} \langle \vec{\theta}^{\triangleright \langle \rangle_j}, \vec{q} \rangle$. Then the j -th component of $\text{mrp}(\pi[\vec{\rho}/\vec{x}_i])$ is f_j .*

Proof. By induction on the structure of π . If $\pi =$

$$\frac{f_1 : \vec{a}_1 \rightarrow \langle \rangle, \dots, f_i : \vec{a}_i \rightarrow \langle a \rangle, \dots, f_j : \vec{a}_j \rightarrow \langle \rangle, \dots, f_n : \vec{a}_n \rightarrow \langle \rangle}{\vec{x}_1 : \mathbf{f}_1 : \langle \vec{a}_1 \rangle, \dots, \vec{x}_i : \mathbf{f}_i = \langle \alpha_i, f \rangle : \vec{a}_i, \dots, \vec{x}_n : \mathbf{f}_n : \vec{a}_n \vdash x_{i, \alpha_i(1)} : a}$$

Then $\pi[\vec{\rho}/\vec{x}_i] =$

$$([f] \rho_{\alpha_i(1)}) \{ \pi_{\alpha_i(1), \text{len}(\vec{a}_i)} \}$$

By definition of left action, we know that the only typing affected is the type of the leftmost variable of $q_{\alpha_i(1)}$, hence $([f] \theta_{\alpha_i(1)})(j) = 1_{\langle \rangle}$. Moreover, by definition of right action, $(\theta_{\alpha_i(1)} \circ \pi_{\alpha_i(1), \text{len}(\vec{a}_i)})(j) = f_j$. We can then conclude.

If \vec{x}_i is an irrelevant variable list, we have $\pi[\vec{\rho}/\vec{x}_i] =$

$$\pi^{\triangleright -\vec{a}_i} \{ \pi_{0, \text{len}(\vec{a}_i)} \}.$$

By hypothesis, $\Gamma_i^{\triangleright \langle \rangle_{i,j}}$, then we can conclude, since $\langle f_j \rangle \oplus \langle \rangle \oplus \dots \oplus \langle \rangle = \langle f_j \rangle$. If $\pi =$

$$\frac{\begin{array}{c} \pi' \\ \vdots \\ \zeta^{\triangleright \vec{x}_i} \oplus \langle \vec{y} \rangle : \eta^{f_i, f_j} \oplus \langle g \rangle : \Delta^{\triangleright \vec{a}_i} \oplus \langle \vec{a} \rangle \vdash p : a \end{array}}{\zeta^{\triangleright \vec{x}_i} : \eta^{f_i, f_j} : \Delta^{\triangleright \vec{a}_i} \vdash \lambda \vec{y} : g.p : \vec{a} \Rightarrow a}$$

Then $\pi[\vec{\rho}/\vec{x}_i] =$

$$\frac{\begin{array}{c} \pi'[\vec{\rho} \bullet \langle \rangle / \vec{x}_i] \\ \vdots \\ (\zeta \oplus \langle \vec{y} \rangle) \otimes (\vec{\zeta} \bullet \langle \rangle) : \eta^{f_i, f_j} \oplus \langle g \rangle \otimes_p^{\vec{x}_i} (\vec{\theta} \bullet \langle \rangle) : (\Delta^{\triangleright -\vec{a}_i} \oplus \langle \vec{a} \rangle) \otimes (\vec{\Gamma} \bullet \langle \rangle) \vdash p[\vec{q}/\vec{x}_i]_{f_i} : a \end{array}}{\zeta \otimes \vec{\zeta} : (\eta^{f_i, f_j} \oplus \langle g \rangle) \otimes_p^{\vec{x}_i} (\vec{\theta} \bullet 1_{\langle \rangle}) : \Delta^{\triangleright -\vec{a}_i} \otimes \vec{\Gamma} \vdash \lambda \vec{y} : g'.p[\vec{q}/\vec{x}]_f : a \Rightarrow a}$$

We apply the IH and we conclude.

If $\pi = \pi_0 \langle \pi_1, \dots, \pi_k \rangle \circ \eta^{f'_i, f'_j}$ then

$$\pi \{ \vec{\rho} / \vec{x} \} =$$

$$\pi_0 \{ ([f'_i] \vec{\rho})_0 / \vec{y}_{i,0} \} \langle \pi_1 \{ ([f'_i] \vec{\rho})_1 / \vec{y}_{i,1} \}, \dots, \pi_k \{ ([f'_i] \vec{\rho})_k / \vec{y}_{i,k} \} \rangle \circ \eta^*.$$

By IH we have that $(\text{mrp}(\pi_l \{ ([f'_i] \vec{\rho})_l / \vec{y}_{i,l} \}))(j) = (\text{mrp}(\pi_l))(j)$. Since the j -th

element of the Γ_l is empty, we have that $(\text{mrp}(\pi\{\vec{\rho}/\vec{x}\}))(j) = (\bigoplus_{l=0}^k (\text{mrp}(\pi_l))(j)) \circ f'_j = (\text{mrp}(\pi))(j)$. \square

Lemma B.2. *Let $\pi \vdash_{\zeta^{\circlearrowleft} \vec{x}_i : \Delta^{\circlearrowleft} \vec{a}_i \oplus \vec{a}} \langle \eta^{f_i}, p \rangle$. The following statements hold.*

1. *Let $\vec{\rho} \vdash_{\vec{\zeta}; \vec{\Gamma}}^{\vec{a}'_i} \langle \vec{\theta}, \vec{q} \rangle$ and $\theta^{\circlearrowleft} g_i : \zeta_1^{\circlearrowleft} \vec{y}_i : \Delta_1^{\circlearrowleft} \vec{\rho}^{\vec{a}'_i} \rightarrow \zeta^{\circlearrowleft} \vec{x}_i : \Delta^{\circlearrowleft} \vec{a}_i$. We have*

$$\pi\{\theta\}[\vec{\rho}/\vec{y}_i] = (\pi[[g_i]\vec{\rho}/\vec{x}_i])\{\theta^{\circlearrowleft -g_i} \otimes \text{sm}(g_i)^*\}.$$

2. *Let $\vec{\rho} \vdash_{\vec{\zeta}; \vec{\Gamma}}^{\vec{a}_i} \langle \vec{\theta}, \vec{q} \rangle$ and $\vec{\eta} = \langle \eta_1, \dots, \eta_{\text{len}(\vec{a}_i)} \rangle : \vec{\zeta} : \vec{\Gamma} \rightarrow \vec{\zeta}' : \vec{\Gamma}'$ we have*

$$\pi[\vec{\rho}\{\vec{\eta}\}/\vec{x}_i] \sim \pi[\vec{\rho}/\vec{x}_i]\{1_{\Delta} \otimes \bigotimes_{l=1}^k \eta_l\}.$$

3. *Let $\vec{\rho} \vdash_{\vec{\zeta}; \vec{\Gamma}}^{\vec{a}_i} \langle \vec{\theta}, \vec{q} \rangle$ and $f : a \rightarrow a'$ we have*

$$[f]\pi[\vec{\rho}/\vec{x}_i] = [f](\pi[\vec{\rho}/\vec{x}_i]).$$

Proof. 1. By induction on π . If $\pi =$

$$\frac{f_1 : \vec{a}_1 \rightarrow \langle \rangle, \dots, f_i = \langle \alpha_i, f \rangle : \vec{a}_i \rightarrow \langle a \rangle, \dots, f_n : \vec{a}_n \rightarrow \langle \rangle}{\vec{x}_1 : \vec{f}_1 : \vec{a}_1, \dots, \vec{x}_i : \vec{f}_i : \vec{a}_i, \dots, \vec{x}_n : \vec{f}_n : \vec{a}_n \vdash x_{i, \alpha_i(1)} : a}$$

Then $\pi\{\theta\} =$

$$\frac{f_1 \circ g_1 : \vec{b}_1 \rightarrow \langle \rangle, \dots, f_i \circ g_i = \langle \beta_i \circ \alpha_i, \langle f \rangle \circ (\vec{g}_i\{\alpha_i\}) \rangle : \vec{b}_i \rightarrow \langle a \rangle, \dots, f_n \circ g_n : \vec{b}_n \rightarrow \langle \rangle}{\vec{y}_1 : \vec{f}_1 \circ \vec{g}_1 : \vec{b}_1, \dots, \vec{y}_i : \vec{f}_i \circ \vec{g}_i : \vec{b}_i, \dots, \vec{y}_n : \vec{f}_n \circ \vec{g}_n : \vec{b}_n \vdash y_{i, (\beta_i \circ \alpha_i)(1)} : a}$$

Let $\vec{g}\{\alpha_i\} = g$. Then by definition we have

$$\pi\{\theta\}[\vec{\rho}/\vec{y}_i] = ([f \circ g]\rho_{(\beta_i \circ \alpha_i)(1)})\{\pi_{(\beta_i \circ \alpha_i)(1), \text{len}(\vec{b}_i)}\}$$

Consider now $[g_i]\rho = \langle [g_{i,1}]\rho_{\beta_i(1)}, \dots, [g_{i,k_i}]\rho_{\beta_i(k_i)} \rangle$. Then

$$\begin{aligned} (\pi[[g_i]\vec{\rho}/\vec{x}_i])\{\theta^{\circlearrowleft -g_i} \otimes \text{sm}(g_i)^*\} = \\ ([f]([g]\rho_{\beta_i(\alpha_i(1))}))\{\pi_{\beta_i(\alpha_i(1)), \text{len}(\vec{b}_i)}\} \end{aligned}$$

Then by Lemma 4.2.9 we can conclude. The second variable case is immediate by definition. The abstraction case is an immediate corollary of the IH.

If $\pi = (\pi\langle \pi_1, \dots, \pi_k \rangle) \circ \eta$, then

$$\pi\{\theta\} = ((\pi\langle \pi_1, \dots, \pi_k \rangle) \circ \eta) \circ \theta = ((\pi\langle \pi_1, \dots, \pi_k \rangle) \circ (\eta \circ \theta))$$

By definition $\pi\{\theta\}[\rho/\vec{x}_i] =$

$$(\pi_0([f'_i \circ g_i]\rho)_0/\vec{x}_{0,i})\langle \pi_1([f'_i \circ g_i]\rho)_{1,i}/\vec{x}_1, \dots, \pi_k([f'_i \circ g_i]\rho)_{k,i}/\vec{x}_k \rangle \circ (\eta \circ \theta)^*$$

where $\vec{x}_{j,i} \in \text{supp}(\pi_j)$, $f'_i \in \eta$.

Consider now $\pi[[g_i]\vec{\rho}/\vec{y}_i] =$

$$(\pi_0([g_i]\rho)_0/\vec{y}_{0,i})\langle \pi_1([g_i]\rho)_{1,i}/\vec{y}_1, \dots, \pi_k([g_i]\rho)_{k,i}/\vec{y}_k \rangle \circ \eta^*$$

We act on the right $\pi[[g_i]\vec{\rho}/\vec{y}_i]\{\theta^{-g_i} \otimes \text{sm}(g_i)^*\}$:

$$(\pi_0([g_i]\rho)_0/\vec{y}_{0,i})\langle \pi_1([g_i]\rho)_{1,i}/\vec{y}_1, \dots, \pi_k([g_i]\rho)_{k,i}/\vec{y}_k \rangle \circ \eta^* \circ (\theta^{-g_i} \otimes \text{sm}(g_i)^*).$$

It is easy to see that $([f'_i]([g_i]\vec{\rho}))_l = ([f'_i \circ g_i]\vec{\rho})_l$ since $[f'_i]([g_i]\vec{\rho}) = [f'_i \circ g_i]\vec{\rho}$. Then we only need to prove that $(\eta \circ \theta)^* = \eta^* \circ (\theta^{-g_i} \otimes \text{sm}(g_i)^*)$. But this follows directly by definition of symmetry group right action, monoidality and associativity of composition, since $(\eta \circ \theta)^* = \vec{\tau} \circ ((\eta^{\circ -f'_i} \circ \theta^{\circ -g_i}) \otimes \text{sm}(f'_i \circ g_i)^*)$ and $\eta^* = \vec{\tau} \circ (\eta^{\circ -f'_i} \otimes \text{sm}(f'_i)^*)$.

2. By induction on π . We prove the application congruence base case, the other cases being direct consequence of the IH and the definition of congruence. If

$$\pi = (\pi_0 \langle \pi_1, \dots, \pi_k \rangle) \circ \eta^{\circ h_i}$$

then

$$\begin{aligned} \pi\{\vec{\rho}\{1 \otimes \vec{\eta}\}/\vec{x}\} &= \\ \pi_0\{([h_i](\vec{\rho}\{\vec{\eta}\}))_0/\vec{y}_0\} & \\ \langle \pi_1\{([h_i](\vec{\rho}\{1 \otimes \vec{\eta}\}))_1/\vec{y}_1\}, \dots, \pi_k\{([h_i](\vec{\rho}\{1 \otimes \vec{\eta}\}))_k/\vec{y}_k\} \rangle \circ \eta^* & \end{aligned}$$

we have that $([h_i](\vec{\rho}\{1 \otimes \vec{\eta}\}))_j = ([h_i]\vec{\rho})_j\{(1 \otimes \vec{\eta}\{\text{sm}(h_i)\})_j\}$. Then, by IH

$$\pi_j\{([h_i](\vec{\rho}\{1 \otimes \vec{\eta}\}))_j/\vec{y}_j\} \sim \pi_j\{([h_i]\vec{\rho})_j/\vec{y}_j\}\{(1 \otimes \vec{\eta}\{\text{sm}(h_i)\})_j\}$$

then by definition of congruence

$$\begin{aligned} \pi\{\vec{\rho}\{1 \otimes \vec{\eta}\}/\vec{x}\} &\sim \\ \pi_0\{([h_i]\vec{\rho})_0/\vec{y}_0\} \langle \pi_1\{([h_i]\vec{\rho})_1/\vec{y}_1\}, \dots, \pi_k\{([h_i]\vec{\rho})_k/\vec{y}_k\} \rangle \circ \left(\bigotimes_{l=0}^k (1 \otimes \vec{\eta}\{\text{sm}(h_i)\})_l \right) \circ \eta^* & \\ = \pi\{\vec{\rho}/\vec{x}_i\}\{1 \otimes \bigotimes \eta_l\}. & \end{aligned}$$

3. By induction on π . If $\pi =$

$$\frac{g_1 : \vec{a}_1 \rightarrow \langle \rangle, \dots, g_i = \langle \alpha_i, g \rangle : \vec{a}_i \rightarrow \langle a \rangle, \dots, g_n : \vec{a}_n \rightarrow \langle \rangle}{\vec{x}_1 : \mathbf{g}_1 : \vec{a}_1, \dots, \vec{x}_i : \mathbf{g}_i : \vec{a}_i, \dots, \vec{x}_n : \mathbf{g}_n : \vec{a}_n \vdash x_{i,\alpha_i(1)} : a}$$

Then $[f]\pi =$

$$\frac{g_1 : \vec{a}_1 \rightarrow \langle \rangle, \dots, \langle f \rangle \circ g_i = \langle \alpha_i, f \circ g \rangle : \vec{a}_i \rightarrow \langle a' \rangle, \dots, g_n : \vec{a}_n \rightarrow \langle \rangle}{\vec{x}_1 : \mathbf{g}_1 : \vec{a}_1, \dots, \vec{x}_i : \langle f \rangle \circ \mathbf{g}_i : \vec{a}_i, \dots, \vec{x}_n : \mathbf{g}_n : \vec{a}_n \vdash x_{i, \alpha_i(1)} : a'}$$

By definition $[f]\pi[\vec{\rho}/\vec{x}_i] =$

$$([f \circ g]\rho_{(\beta_i \circ \alpha_i)(1)})\{\pi_{(\beta_i \circ \alpha_i)(1), \text{len}(\vec{a}_1+1)}\}$$

Then we can conclude by Lemma 4.2.9, since $[f](\pi[\vec{\rho}/\vec{x}_i]) =$

$$[f]([\mathbf{g}]\rho_{\beta_i(1)})\{\pi_{\beta_i(1), \text{len}(\vec{a}_1+1)}\}$$

The second variable case is immediate by definition of left action and Lemma 4.2.9.

If $\pi =$

$$\frac{\begin{array}{c} \pi' \\ \vdots \\ \zeta \oplus \langle \vec{x} \rangle : \eta \oplus \langle f' \rangle : \Delta \oplus \langle \vec{a} \rangle \vdash p : a \end{array}}{\zeta : \eta : \Delta \vdash \lambda \vec{x} : f'.p : \Delta \Rightarrow a}$$

then $f = \langle \alpha, \vec{g} \rangle \Rightarrow g$ and $[f]\pi =$

$$\frac{\begin{array}{c} ([g]\pi')\{1_\Delta \oplus \langle \langle \alpha, \vec{g} \rangle \rangle\} \\ \vdots \\ \zeta \oplus \langle \vec{y} \rangle : \eta \oplus \langle f' \circ \langle \alpha, \vec{g} \rangle \rangle : \Delta \oplus \langle \vec{a}' \rangle \vdash ([f]p)^{1_\Delta \oplus \langle \langle \alpha, \vec{g} \rangle \rangle} : a' \end{array}}{\zeta : \eta : \Delta \vdash \lambda \vec{y} : f' \circ \langle \alpha, \vec{g} \rangle. ([f]p)^{1_\Delta \oplus \langle \langle \alpha, \vec{g} \rangle \rangle} : \Delta \Rightarrow a'}$$

we have that $[f]\pi\{\vec{\rho}/\vec{x}_i\} = \lambda \vec{y} : f'. [f]\pi'\{1_\Delta \oplus \langle \langle \alpha, \vec{g} \rangle \rangle\}\{\vec{\rho} \bullet \langle \rangle / \vec{x}_i\}$. By IH and the first point of this lemma we have $[f]\pi'\{1_\Delta \oplus \langle \langle \alpha, \vec{g} \rangle \rangle\}\{\vec{\rho} \bullet \langle \rangle / \vec{x}_i\} = [f]\pi'\{\vec{\rho} \bullet \langle \rangle / \vec{x}_i\}\{(1_{\Delta \triangleright \vec{a}_i} \oplus \langle \langle \alpha, \vec{g} \rangle \rangle) \otimes 1\}$. We can then conclude.

If $\pi =$

$$\frac{\begin{array}{c} \pi_0 \\ \vdots \\ \zeta_0 \triangleright \vec{y}_{i,0} : \eta_0 : \Gamma_0 \vdash s : a \end{array} \quad \left(\begin{array}{c} \pi_l \\ \vdots \\ \zeta_l \triangleright \vec{y}_{i,l} : \eta_l : \Gamma_l \vdash t_l : a_l \end{array} \right)_{l=1}^k \quad \eta \triangleright h_i : \zeta : \Delta \rightarrow \bigotimes_{l=0}^k \zeta_l : \bigotimes_{l=0}^k \Gamma_l}{\zeta : (\bigotimes_{l=0}^k \eta_j) \circ \eta : \Delta \vdash (s\langle t_1, \dots, t_k \rangle)^\eta}$$

then $[f]\pi\{\vec{\rho}/\vec{x}_i\} =$

$$[1 \Rightarrow f]\pi_0\{([h_i]\vec{\rho})_{i,0}/\vec{y}_{i,0}\}\{\pi_1\{([h_i]\vec{\rho})_{1,i}/\vec{y}_{1,i}\}, \dots, \pi_k\{([h_i]\vec{\rho})_{k,i}/\vec{y}_{k,i}\}\} \circ \eta$$

then, by IH $[1 \Rightarrow f]\pi_0\{([h_i]\vec{\rho})_{i,0}/\vec{y}_{i,0}\} = [1 \Rightarrow f](\pi_0\{([h_i]\vec{\rho})_{i,0}/\vec{y}_{i,0}\})$ and we can then conclude.

□

Proposition B.3. *Let $\pi \vdash_{\zeta^{\circlearrowleft \vec{x}_i : \Delta^{\circlearrowleft \vec{a}_i \oplus \vec{a}}}^a} \langle \eta^{f_i}, p \rangle$, $\pi' \vdash_{\zeta^{\circlearrowleft \vec{x}_i : \Delta^{\circlearrowleft \vec{a}_i \oplus \vec{a}}}^a} \langle \eta^{f_i}, p \rangle$ such that $\pi \sim \pi'$ and $\vec{\rho} \vdash_{\vec{\zeta}^{\circlearrowleft \vec{f}} : \vec{\Gamma}^{\circlearrowleft}} \langle \vec{\theta}^{\circlearrowleft}, \vec{q} \rangle$, $\vec{\rho}' \vdash_{\vec{\zeta}^{\circlearrowleft \vec{f}} : \vec{\Gamma}^{\circlearrowleft}} \langle \vec{\theta}'^{\circlearrowleft}, \vec{q}' \rangle$ such that $\vec{\rho} \sim \vec{\rho}'$. Then $\pi[\vec{\rho}/\vec{x}_i] \sim \pi'[\vec{\rho}'/\vec{x}_i]$.*

Proof. We prove the application base case. Let $\pi =$

$$\frac{\zeta_0^{\circlearrowleft \vec{y}_{0,i}} : \theta_0 \circ \eta_0^{\circlearrowleft h_{g_0,i}} : \Delta_0 \vdash s^{\eta_0} : \vec{a} \Rightarrow a \quad \left(\begin{array}{c} \pi_l \{ \eta_l \} \\ \vdots \\ \zeta_l^{\circlearrowleft \vec{y}_{l,i}} : \theta_l \circ \eta_l^{\circlearrowleft h_{g_l,i}} : \Delta_l \vdash q_l^{\eta_l} : a_l \end{array} \right)_{l=1}^k}{\zeta : (\otimes_{j=0}^k \theta_j) \circ (\otimes_{j=0}^k \eta_j) \circ \theta : \Delta \vdash (s^{\eta_0} \langle q_1^{\eta_1}, \dots, q_k^{\eta_k} \rangle)^{\theta} : a}$$

and $\pi' =$

$$\frac{(\zeta_0')^{\circlearrowleft \vec{y}'_{0,i}} : \theta_0' \circ \eta_0'^{\circlearrowleft h_{g_0,i}} : \Delta_0' \vdash s : \vec{a} \Rightarrow a \quad \left((\zeta_l')^{\circlearrowleft \vec{y}'_{l,i}} : \theta_l' \circ \eta_l'^{\circlearrowleft h_{g_l,i}} : \Delta_l' \vdash q_l : a_l \right)_{l=1}^k}{\zeta : (\otimes_{j=0}^k \theta_j) \circ (\otimes_{j=0}^k \eta_j) \circ \theta : \Delta \vdash (s \vec{q})^{\otimes_{j=0}^k \eta_j \circ \theta} : a}$$

We have $\pi[\vec{\rho}/\vec{x}_i] =$

$$(\pi_0 \{ \eta_0 \} [([f'_i] \vec{\rho})_0 / \vec{y}_{0,i}] \langle \pi_1 \{ \eta_1 \} [([f'_i] \vec{\rho})_1 / \vec{y}_{1,i}], \dots, \pi_k \{ \eta_k \} [([f'_i] \vec{\rho})_k / \vec{y}_{k,i}] \rangle) \circ \theta^*$$

Now consider $\pi'[\vec{\rho}/\vec{x}_i] =$

$$(\pi_0 [((\bigoplus_{j=0}^k g_j) \circ f'_i) \vec{\rho}_0 / \vec{y}'_{0,i}] \langle \pi_1 [((\bigoplus_{j=0}^k g_j) \circ f'_i) \vec{\rho}_1 / \vec{y}'_{1,i}], \dots, \pi_k [((\bigoplus_{j=0}^k g_j) \circ f'_i) \vec{\rho}_k / \vec{y}'_{k,i}] \rangle) \circ \eta^*$$

where $\eta = (\otimes \eta_j) \circ \theta$.

By the former Lemma we know that

$$\pi_l \{ \eta^{\circlearrowleft g_{l,i}} \} [([f'_i] \vec{\rho})_l / \vec{y}_{l,i}] = \pi_l [g_{l,i} ([f'_i] \vec{\rho})_l / \vec{y}'_{l,i}] \{ \eta_l^{-g_{l,i}} \otimes \text{sm}(g_{l,i})^* \}.$$

By definition of left action we have $[g_{l,i} ([f'_i] \vec{\rho})_l] = [(\bigoplus_{j=0}^k g_{j,i}) \circ f'_i] \vec{\rho}_l$ since $[(\bigoplus_{j=0}^k g_{j,i}) \circ f'_i] \vec{\rho} = [\bigoplus_{j=0}^k g_{j,i}] ([f'_i] \vec{\rho})$.

Then we just need to prove that $(\otimes_{j=0}^k \eta_j) \circ \theta)^* = ((\otimes_{j=0}^k (\eta_l^{-g_{l,i}} \otimes \text{sm}(g_{l,i})^*)) \circ \theta^*$. By definition $(\otimes_{j=0}^k \eta_j) \circ \theta)^* = \vec{\tau}_1 \circ ((\otimes_{j=0}^k \eta_j^{-g_{j,i}}) \circ \theta^{-f'_i}) \otimes \text{sm}((\bigoplus_{j=0}^k g_{j,i}) \circ f'_i)^*$. Again, by definition $\theta^* = \vec{\tau}_2 \circ (\theta^{-f'_i} \circ \text{sm}(f'_i)^*)$. Now $((\otimes_{j=0}^k (\eta_l^{-g_{l,i}} \otimes \text{sm}(g_{l,i})^*)) \circ \theta^* = (\otimes_{j=0}^k (\eta_l^{-g_{l,i}} \otimes \text{sm}(g_{l,i})^*)) \circ (\vec{\tau}_2 \circ (\theta^{-f'_i} \circ \text{sm}(f'_i)^*))$. The explicit type of the two permutations is

$$\vec{\tau}_1 : \bigotimes_{j=0}^k \Delta'_j \otimes \left(\bigotimes_{j=0}^k \vec{\Gamma} \left\{ \bigoplus_{j=0}^k g_{j,i} \circ f'_i \right\} \right) \rightarrow \bigotimes_{j=0}^k (\Delta'_j \otimes (\vec{\Gamma} \left\{ \bigoplus_{j=0}^k g_{j,i} \circ f'_i \right\})_j)$$

$$\vec{\tau}_2 : \bigotimes_{j=0}^k \Delta_j \otimes \left(\bigotimes_{j=0}^k \vec{\Gamma}\{f'_i\} \right) \rightarrow \bigotimes_{j=0}^k (\Delta'_j \otimes (\vec{\Gamma}\{f'_i\})_j)$$

We set $(\bigotimes_{j=0}^k \eta_j^{-g_{j,i}}) \circ \theta^{-f'_i} \otimes \text{sm}((\bigoplus_{j=0}^k g_{j,i}) \circ f'_i)^* = \iota$ and $\bigoplus_{j=0}^k g_{j,i} \circ f'_i = h$. Hence we need to prove that the following diagram commutes:

$$\begin{array}{ccc} \Delta \otimes \vec{\Gamma} & \xrightarrow{\theta^{-f'_i} \otimes \text{sm}(f'_i)^*} & \bigotimes \Delta_j \otimes (\vec{\Gamma}\{f'_i\}) & \xrightarrow{\vec{\tau}_2} & \bigotimes (\Delta_j \otimes (\vec{\Gamma}\{f'_i\})_j) \\ \downarrow \iota & \swarrow & \bigotimes \eta_j^{-g_{j,i}} \otimes \text{sm}(g_{j,i})^* & \searrow & \bigotimes (\eta_j^{-g_{j,i}} \otimes \text{sm}(g_{j,i})^*) \\ \bigotimes \Delta'_j \otimes (\vec{\Gamma}\{h\}) & \xrightarrow{\vec{\tau}_1} & \bigotimes (\Delta'_j \otimes (\vec{\Gamma}\{h\})_j) & & \end{array}$$

The commutation of the former diagram is a direct corollary of tensor product functoriality (triangle diagram) and naturality of symmetries (square diagram). \square

B.2. Canonical Forms of Type Derivations

Erasing Subtyping Information Let π be a polyadic type derivation of conclusion $\zeta : \eta : \Delta \vdash p : a$ and $\eta = \langle g_1, \dots, g_n \rangle : \zeta = \langle \vec{x}_1, \dots, \vec{x}_n \rangle : \Delta = \langle \vec{b}_1, \dots, \vec{b}_n \rangle \rightarrow \zeta' = \langle \vec{y}_1, \dots, \vec{y}_n \rangle : \Gamma = \langle \vec{a}_1, \dots, \vec{a}_n \rangle$, where ζ' is a repetitions-free variable tuple. we define $\pi/\eta^{\zeta'}$ with conclusion $\zeta' : \mathbf{1}_\Gamma : \Gamma \vdash p^{\zeta'} : a$, where $p^{\zeta'} = p\{\zeta'/\text{occ}_\zeta(p)\}$, by induction as follows:

$$\left(\frac{f_1 : \vec{b}_1 \rightarrow \langle \rangle, \dots, f_i : \vec{b}_i \rightarrow \langle a \rangle, \dots, f_n : \vec{b}_n \rightarrow \langle \rangle}{\vec{x}_1 : \mathbf{f}_1 : \vec{b}_1, \dots, \vec{x}_i : \mathbf{f}_i : \vec{b}_i, \dots, \vec{x}_n : \mathbf{f}_n : \vec{b}_n \vdash x_{i,\alpha(1)} : a} \right) / \eta^{\zeta'} =$$

$$\frac{1 : \langle \rangle \rightarrow \langle \rangle, \dots, \langle 1_a \rangle : \langle a \rangle \rightarrow \langle a \rangle, \dots, 1_\langle \rangle : \langle \rangle \rightarrow \langle \rangle}{\langle \rangle : \mathbf{1}_\langle \rangle : \langle \rangle, \dots, \langle y \rangle : \langle \mathbf{1}_a \rangle : \langle a \rangle, \dots, \langle \rangle : \mathbf{1}_\langle \rangle : \langle \rangle \vdash y : a}$$

$$(\lambda \vec{x} : \mathbf{f} . \pi) / \eta^{\zeta'} = \lambda \vec{x} : \mathbf{f} . ((\pi / (\eta \oplus \langle f \rangle))^\zeta) \{ \mathbf{1} \oplus \langle f \rangle \}$$

$$((\pi_0 \langle \pi_1, \dots, \pi_k \rangle) \circ \theta) / \eta^{\zeta'} = ((\pi_0 / \text{mrp}(\pi_0))^{\zeta'_0} \langle \pi_1 / \text{mrp}(\pi_1)^{\zeta'_1}, \dots, \pi_k / \text{mrp}(\pi_k)^{\zeta'_k} \rangle) \circ \mathbf{1}$$

Where in the application case $\zeta' = \bigotimes_{j=0}^k \zeta'_j$. We shall constantly wrote π/η instead of $\pi/\eta^{\zeta'}$, keeping the parameter ζ' implicit.

The operation π/η erases the subtyping on the context side of a type derivation. However, it is important to keep in mind that the subtyping information of bound variables remains untouched.

Lemma B.4. *Let $\pi \vdash_{\zeta, \Delta}^a \langle \eta, p \rangle$ with $\eta : \zeta : \Delta \rightarrow \zeta' : \Delta'$ and $\theta : \zeta'' : \Delta'' \rightarrow \zeta' : \Delta'$. The following statements hold.*

1. $\pi \sim (\pi/\eta)\{\eta\}$.

$$\begin{aligned}
 & CF \left(\pi = \frac{f_1 : \vec{a}_1 \rightarrow \langle \rangle, \dots, f_i : \vec{a}_i \rightarrow \langle a \rangle, \dots, f_n : \vec{a}_n \rightarrow \langle \rangle}{\vec{x}_1 : \mathbf{f}_1 : \vec{a}_1, \dots, \vec{x}_i : \mathbf{f}_i : \vec{a}_i, \dots, \vec{x}_n : \mathbf{f}_n : \vec{a}_n \vdash x_{i,\alpha(1)} : a} \right) = \pi \\
 & CF \left(\frac{\pi}{\Sigma, \vec{x} : \mathbf{f} : \vec{a} \vdash p : a} \right) = \frac{CF(\pi)}{\Sigma, \vec{x} : \mathbf{f} : \vec{a} \vdash p : a} \\
 & CF \left(\frac{\pi}{\Sigma \vdash \lambda \vec{x} : \mathbf{f}. p : \vec{a} \Rightarrow a'} \right) = \frac{CF(\pi)}{\Sigma \vdash \lambda \vec{x} : \mathbf{f}. p : \vec{a} \Rightarrow a'} \\
 & CF \left(\frac{\left(\frac{\pi_0}{\zeta_0 : \eta_0 : \Gamma_0 \vdash p : \vec{a} \Rightarrow a} \quad \left(\frac{\pi_i}{\zeta_i : \eta_i : \Gamma_i \vdash q_i : a_i} \right)_{i=1}^k \quad \theta : \zeta : \Delta \rightarrow (\otimes_{j=0}^k \zeta_j) : \otimes_{j=0}^k \Gamma_j \right)}{\zeta : \eta = (\otimes_{j=0}^k \eta_j \circ \theta) : \Delta \vdash (pq)^\theta : a} \right) = \\
 & \frac{CF(\pi_i)/\eta_0^{\zeta'_0} \quad \left(\frac{CF(\pi_i)/\eta_i^{\zeta'_i}}{\zeta'_i : \mathbf{1}_{\Gamma'_i} : \Gamma'_i \vdash q_i^{\zeta'_i} : a_i} \right)_{i=1}^k \quad (\otimes_{j=0}^k \eta_j) \circ \eta : \zeta : \Delta \rightarrow (\otimes_{j=0}^k \zeta'_j) : (\otimes_{j=0}^k \Gamma'_j)}{\zeta : \eta = ((\otimes_{j=0}^k \eta_j) \circ \eta) : \Delta \vdash (p^{\zeta'_0} \langle q_1^{\zeta'_1}, \dots, q_k^{\zeta'_k} \rangle) (\otimes_{j=0}^k \eta_j)^{\circ \eta} : a}
 \end{aligned}$$

Where in the application case we took $\zeta' = \otimes \zeta'_j$.

Figure .1. – Canonical forms.

2. $\pi/\eta = (\pi\{\theta\})/(\eta \circ \theta)$.
3. if $\pi \sim \pi'$ then $\pi/\eta = \pi'/\eta'$.

Proof. By induction on the structure of π .

1. We prove the variable case. Let $\zeta = \langle \vec{x}_1, \dots, \vec{x}_n \rangle, \zeta' = \langle \vec{y}_1, \dots, \vec{y}_n \rangle, \Delta = \langle \vec{b}_1, \dots, \vec{b}_n \rangle$ and $\Delta' = \langle \vec{a}_1, \dots, \vec{a}_n \rangle$. Let

$$\pi = \frac{f_1 : \vec{b}_1 \rightarrow \langle \rangle, \dots, f_i : \vec{b}_i \rightarrow \langle a \rangle, \dots, f_n : \vec{b}_n \rightarrow \langle \rangle}{\vec{x}_1 : \mathbf{f}_1 : \vec{b}_1, \dots, \vec{x}_i : \mathbf{f}_i : \vec{b}_i, \dots, \vec{x}_n : \mathbf{f}_n : \vec{b}_n \vdash x_{i,\alpha(1)} : a}$$

Then $\pi/\eta =$

$$\frac{1 : \langle \rangle \rightarrow \langle \rangle, \dots, \langle 1_a \rangle : \langle a \rangle \rightarrow \langle a \rangle, \dots, 1_\langle \rangle : \langle \rangle \rightarrow \langle \rangle}{\langle \rangle : \mathbf{1}_\langle \rangle : \langle \rangle, \dots, \langle y \rangle : \langle 1_a \rangle : \langle a \rangle, \dots, \langle \rangle : \mathbf{1}_\langle \rangle : \langle \rangle \vdash y : a}$$

By definition $(\pi/\eta)\{\eta\} =$

$$\frac{1_\langle \rangle \circ f_1 : \vec{b}_1 \rightarrow \langle \rangle, \dots, (\langle 1_a \rangle \circ f_i) : \vec{b}_i \rightarrow \langle a \rangle, \dots, 1_\langle \rangle f_n : \vec{b}_n \rightarrow \langle \rangle}{\vec{x}_1 : \mathbf{f}_1 : \vec{b}_1, \dots, \vec{x}_i : \mathbf{f}_i : \vec{b}_i, \dots, \vec{x}_n : \mathbf{f}_n : \vec{b}_n \vdash y[\zeta\{\text{sm}(\eta)\} = x_{i,\alpha(1)}/\langle y \rangle] : a}$$

Then $(\pi/\eta)\{\eta\} = \pi$ and, in particular, they are equivalent.

We prove the application case. Let

$$\pi = \frac{\begin{array}{c} \pi_0 \\ \vdots \\ \zeta_0 : \eta_0 : \Gamma_0 \vdash p : \vec{a} \Rightarrow a \end{array} \quad \left(\begin{array}{c} \pi_i \\ \vdots \\ \zeta_i : \eta_i : \Gamma_i \vdash q_i : a_i \end{array} \right)_{i=1}^k \quad \theta : \zeta : \Delta \rightarrow (\otimes_{j=0}^k \zeta_j) : \otimes_{j=0}^k \Gamma_j}{\zeta : \eta = (\otimes_{j=0}^k \eta_j \circ \theta) : \Delta \vdash (p\langle q_1, \dots, q_k \rangle)^\theta : a}$$

Then, we consider $\eta_j = 1_{\Gamma_j} \circ \eta_j : \zeta_j : \Gamma_j \rightarrow \zeta'_j : \Gamma'_j$ with $1_{\Gamma'_j} : \zeta'_j : \Gamma'_j \rightarrow \zeta'_j : \Gamma'_j$.
By definition $\pi/\eta =$

$$\frac{\begin{array}{c} \pi_0/\eta_0 \\ \vdots \\ \zeta'_0 : 1_{\Gamma'_0} : \Gamma_0 \vdash p^{\zeta'_0} : \vec{a} \Rightarrow a \end{array} \quad \left(\begin{array}{c} \pi_i/\eta_i \\ \vdots \\ \zeta'_i : 1_{\Gamma'_i} : \Gamma'_i \vdash q_i^{\zeta'_i} : a_i \end{array} \right)_{i=1}^k \quad 1_{\otimes \Gamma'_j} : (\otimes_{j=0}^k \zeta'_j) : \otimes_{j=0}^k \Gamma'_j \rightarrow (\otimes_{j=0}^k \zeta_j) : \otimes_{j=0}^k \Gamma_j}{\otimes_{j=0}^k \zeta'_j : 1_{\otimes \Gamma'_j} : \otimes_{j=0}^k \Gamma'_j \vdash (p\langle q_1, \dots, q_k \rangle)^{\otimes_{j=0}^k \zeta'_j} : a}$$

consider $(\pi/\eta)\{\eta\} =$

$$\frac{\begin{array}{c} \pi_0/\eta_0 \\ \vdots \\ \zeta'_0 : 1_{\Gamma'_0} : \Gamma_0 \vdash p^{\zeta'_0} : \vec{a} \Rightarrow a \end{array} \quad \left(\begin{array}{c} \pi_i/\eta_i \\ \vdots \\ \zeta'_i : 1_{\Gamma'_i} : \Gamma'_i \vdash q_i^{\zeta'_i} : a_i \end{array} \right)_{i=1}^k \quad \eta \circ 1_{\otimes \Gamma'_j} : \zeta : \Delta \rightarrow (\otimes_{j=0}^k \zeta'_j) : \otimes_{j=0}^k \Gamma'_j}{\zeta : \eta : \Delta \vdash ((p\langle q_1, \dots, q_k \rangle)^{\otimes_{j=0}^k \zeta'_j})^\eta : a}$$

By definition of equivalence we have $(\pi/\eta)\{\eta\} \sim$

$$\frac{\begin{array}{c} (\pi_0/\eta_0)\{\eta_0\} \\ \vdots \\ \zeta_0 : \eta_0 : \Gamma_0 \vdash (p^{\zeta'_0})^{\eta_0} : \vec{a} \Rightarrow a \end{array} \quad \left(\begin{array}{c} (\pi_i/\eta_i)\{\eta_i\} \\ \vdots \\ \zeta_i : \eta_i : \Gamma_i \vdash (q_i^{\zeta'_i})^{\eta_i} : a_i \end{array} \right)_{i=1}^k \quad \theta : \zeta : \Delta \rightarrow (\otimes_{j=0}^k \zeta_j) : \otimes_{j=0}^k \Gamma_j}{\zeta : \eta : \Delta \vdash (((p^{\zeta'_0})^{\eta_0} \langle (q_1^{\zeta'_1})^{\eta_1}, \dots, (q_k^{\zeta'_k})^{\eta_k} \rangle))^\theta : a}$$

Then we apply the IH and we get $\pi_j \sim (\pi_j/\eta_j)\{\eta_j\}$. We conclude by congruence and transitivity.

2. Completely straightforward induction on the structure of π .
3. We prove the application case, the other cases being immediate by definition and IH. If $\pi \sim \pi'$ then

$$\begin{aligned} \pi &= \frac{\begin{array}{c} \pi_0\{\eta_0\} \\ \vdots \\ \zeta_0 : \theta_0 \circ \eta_0 : \Gamma_0 \vdash s^{\eta_0} : \vec{a} \Rightarrow a \end{array} \quad \left(\begin{array}{c} \pi_i\{\eta_i\} \\ \vdots \\ \zeta_i : \theta_i \circ \eta_i : \Gamma_i \vdash q_i^{\eta_i} : a_i \end{array} \right)_{i=1}^k \quad \theta : \zeta : \Delta \rightarrow \otimes_{j=0}^k \zeta_j : \otimes_{j=0}^k \Gamma_j}{\zeta : (\otimes_{j=0}^k \theta_j) \circ (\otimes_{j=0}^k \eta_j) \circ \theta : \Delta \vdash (s^{\eta_0} \langle q_1^{\eta_1}, \dots, q_k^{\eta_k} \rangle)^\theta : a} \\ &\sim \pi' = \frac{\begin{array}{c} \pi_0 \\ \vdots \\ \zeta'_0 : \theta_0 : \Gamma'_0 \vdash s : \vec{a} \Rightarrow a \end{array} \quad \left(\begin{array}{c} \pi_i \\ \vdots \\ \zeta'_i : \theta_i : \Gamma'_i \vdash q_i : a_i \end{array} \right)_{i=1}^k \quad (\otimes_{j=0}^k \eta_j) \circ \theta : \zeta : \Delta \rightarrow \otimes_{j=0}^k \zeta'_j : \otimes_{j=0}^k \Gamma'_j}{\zeta : (\otimes_{j=0}^k \theta_j) \circ (\otimes_{j=0}^k \vec{\eta}_j) \circ \theta : \Delta \vdash (s\vec{q})^{(\otimes_{j=0}^k \eta_j) \circ \theta} : a} \end{aligned}$$

By definition $\pi/\eta =$

$$\frac{\begin{array}{c} (\pi_0\{\eta_0\})/(\theta_0 \circ \eta_0) \\ \vdots \\ \zeta_0'' : \mathbf{1}_{\Gamma_0''} : \Gamma_0'' \vdash (s^{\eta_0})^{\zeta_0''} : \vec{a} \Rightarrow a \end{array}}{\otimes_{j=0}^k \zeta_j'' : \mathbf{1}_{\otimes \Gamma_j''} : \otimes_{j=0}^k \Gamma_j'' \vdash (s^{\eta_0})^{\zeta_0''} \langle (q_1^{\eta_1})^{\zeta_1''}, \dots, (q_k^{\eta_k})^{\zeta_k''} \rangle : a} \left(\begin{array}{c} (\pi_i\{\eta_i\})/(\theta_i \circ \eta_i) \\ \vdots \\ \zeta_i'' : \mathbf{1}_{\Gamma_i''} : \Gamma_i'' \vdash (q_i^{\eta_i})^{\zeta_i''} : a_i \end{array} \right)_{i=1}^k \quad \mathbf{1}_{\otimes \Gamma_j''} : \otimes_{j=0}^k \zeta_j'' : \otimes_{j=0}^k \Gamma_j'' \rightarrow \otimes_{j=0}^k \zeta_j'' : \otimes_{j=0}^k \Gamma_j''$$

and $\pi'/\eta =$

$$\frac{\begin{array}{c} (\pi_0)/\theta_0 \\ \vdots \\ \zeta_0'' : \mathbf{1}_{\Gamma_0''} : \Gamma_0'' \vdash s^{\zeta_0''} : \vec{a} \Rightarrow a \end{array}}{\otimes_{j=0}^k \zeta_j'' : \mathbf{1}_{\otimes \Gamma_j''} : \otimes_{j=0}^k \Gamma_j'' \vdash (s^{\zeta_0''} \langle q_1^{\zeta_1''}, \dots, q_k^{\zeta_k''} \rangle) : a} \left(\begin{array}{c} \pi_i/\theta_i \\ \vdots \\ \zeta_i'' : \mathbf{1}_{\Gamma_i''} : \Gamma_i'' \vdash (q_i)^{\zeta_i''} : a_i \end{array} \right)_{i=1}^k \quad \mathbf{1}_{\otimes \Gamma_j''} : \otimes_{j=0}^k \zeta_j'' : \otimes_{j=0}^k \Gamma_j'' \rightarrow \otimes_{j=0}^k \zeta_j'' : \otimes_{j=0}^k \Gamma_j''$$

Then by point (2) of this Lemma we can conclude. \square

Canonical Forms of Type Derivations We introduce the notion of *canonical form* of a polyadic type derivations. Given π of conclusion $\zeta : \eta : \Delta \vdash p : a$ and ζ' a repetitions-free variable tuple such that $\eta : \zeta : \Delta \rightarrow \zeta' : \Delta'$ we define $CF(\pi)_{\zeta'}$ by induction in Figure .1. Also this construction formally depends on a choice of free variables for the application case. The intuition is that canonical forms are canonical representative for the equivalence class of type derivations, up to the choice of variable tuple ζ' . Also this time, we shall constantly write $CF(\pi)$ keeping the parameters ζ', ζ'' implicit. We are justified in doing so by the congruence relation, that makes all possible choices equivalent (it's easy to see that $CF(\pi)_{\zeta} \sim CF(\pi)_{\zeta'}$ for all valid variable tuples ζ, ζ').

Lemma B.5. *We have $\pi \sim CF(\pi)_{\zeta'}$.*

Proof. We prove the application base case. Let $\pi =$

$$\frac{\begin{array}{c} \pi_0 \\ \vdots \\ \zeta_0 : \eta_0 : \Gamma_0 \vdash p : \vec{a} \Rightarrow a \end{array}}{\zeta : \eta = (\otimes_{j=0}^k \eta_j \circ \theta) : \Delta \vdash (pq)^\theta : a} \left(\begin{array}{c} \pi_i \\ \vdots \\ \zeta_i : \eta_i : \Gamma_i \vdash q : a_i \end{array} \right)_{i=1}^k \quad \theta : \zeta : \Delta \rightarrow (\otimes_{j=0}^k \zeta_j) : \otimes_{j=0}^k \Gamma_j$$

We take $\zeta' = \otimes \zeta_j'$. By Lemma B.4, definition of congruence and IH we have that $\pi \sim \pi'$ with $\pi' =$

$$\frac{\begin{array}{c} CF(\pi_i)/\eta_0^{\zeta_0'} \\ \vdots \\ \zeta_0' : \mathbf{1}_{\Gamma_0'} : \Gamma_0' \vdash p^{\zeta_0'} : \vec{a} \Rightarrow a \end{array}}{\zeta : \eta = ((\otimes_{j=0}^k \eta_j) \circ \eta) : \Delta \vdash (p^{\zeta_0'} \langle q_1^{\zeta_1'}, \dots, q_k^{\zeta_k'} \rangle) (\otimes_{j=0}^k \eta_j)^{\circ \eta} : a} \left(\begin{array}{c} CF(\pi_i)/\eta_i^{\zeta_i'} \\ \vdots \\ \zeta_i' : \mathbf{1}_{\Gamma_i'} : \Gamma_i' \vdash q_i^{\zeta_i'} : a_i \end{array} \right)_{i=1}^k \quad (\otimes_{j=0}^k \eta_j) \circ \eta : \zeta : \Delta \rightarrow (\otimes_{j=0}^k \zeta_j') : (\otimes_{j=0}^k \Gamma_j')$$

and $\pi' = CF_{\otimes \zeta'_j}(\pi)$.

□

Proposition B.6. *Let $\pi \vdash_{\zeta:\Delta}^a \langle \eta, p \rangle$ and $\pi' \vdash_{\zeta':\Delta'}^{a'} \langle \eta', p' \rangle$. The following statements hold.*

1. *If $\pi \sim \pi'$ then $\eta = \eta'$ and $p = p'$.*
2. *If $\eta = \eta'$ and $p = p'$ then $\Delta = \Delta'$ and $a = a'$. If moreover $\zeta = \zeta'$ then $\pi \sim \pi'$.*

Proof. We prove it by induction on π . The case of the variable and the lambda abstraction are immediate by congruence and IH. We prove the application case.

1. Let $\pi \vdash_{\zeta:\Delta}^a \langle \eta, p \rangle$ with $p = s\vec{t}$. If $\pi \sim \pi'$ then $a = a', \Delta = \Delta'$ and

$$\frac{\begin{array}{c} \pi = \\ \pi_0\{\eta_0\} \\ \vdots \\ \zeta_0 : \theta_0 \circ \eta_0 : \Gamma_0 \vdash s^{\eta_0} : \vec{a} \Rightarrow a \end{array} \quad \left(\begin{array}{c} \pi_i\{\eta_i\} \\ \vdots \\ \zeta_i : \theta_i \circ \eta_i : \Gamma_i \vdash q_i^{\eta_i} : a_i \end{array} \right)_{i=1}^k \quad \theta : \zeta : \Delta \rightarrow \otimes_{j=0}^k \zeta_j : \otimes_{j=0}^k \Gamma_j}{\zeta : (\otimes_{j=0}^k \theta_j) \circ (\otimes_{j=0}^k \eta_j) \circ \theta : \Delta \vdash (s^{\eta_0} \langle q_1^{\eta_1}, \dots, q_k^{\eta_k} \rangle)^\theta : a} \\ \sim \pi' = \\ \frac{\begin{array}{c} \pi_0 \\ \vdots \\ \zeta_0' : \theta_0 : \Gamma_0' \vdash s : \vec{a} \Rightarrow a \end{array} \quad \left(\begin{array}{c} \pi_i \\ \vdots \\ \zeta_i' : \theta_i : \Gamma_i' \vdash q_i : a_i \end{array} \right)_{i=1}^k \quad (\otimes_{j=0}^k \eta_j) \circ \theta : \zeta : \Delta \rightarrow \otimes_{j=0}^k \zeta_j' : \otimes_{j=0}^k \Gamma_j'}{\zeta : (\otimes_{j=0}^k \theta_j) \circ (\otimes_{j=0}^k \vec{\eta}_j) \circ \theta : \Delta \vdash (s\vec{q})^{(\otimes_{j=0}^k \eta_j) \circ \theta} : a}$$

We only need to prove that $(s^{\eta_0} \langle q_1^{\eta_1}, \dots, q_k^{\eta_k} \rangle)^\theta = (s\vec{q})^{(\otimes_{j=0}^k \eta_j) \circ \theta}$. By definition $(s^{\eta_0} \langle q_1^{\eta_1}, \dots, q_k^{\eta_k} \rangle)^\theta = s^{\eta_0} \langle q_1^{\eta_1}, \dots, q_k^{\eta_k} \rangle [\zeta \{ \text{sm}(\theta) \} / \otimes_{j=0}^k \zeta_j]$. By Lemma 4.2.6

$$s^{\eta_0} \langle q_1^{\eta_1}, \dots, q_k^{\eta_k} \rangle [\zeta \{ \text{sm}(\theta) \} / \otimes_{j=0}^k \zeta_j] =$$

$$s\vec{t} \left\{ \otimes_{j=0}^k \zeta_j \{ \text{sm}(\otimes_{j=0}^k \vec{\eta}_j) \circ \text{sm}(\otimes_{j=0}^k \theta_j) \} / \text{occ}_{\otimes_{j=0}^k \zeta_j'}(s\vec{t}) \right\} \{ \zeta \{ \text{sm}(\theta) \circ \text{sm}(\otimes_{j=0}^k \vec{\eta}_j) \circ \text{sm}(\otimes_{j=0}^k \theta_j) \} / \text{occ}_{\otimes_{j=0}^k \zeta_j}((s\vec{t})^*) \}.$$

Where $(s\vec{t})^* = s\vec{t} \{ \otimes_{j=0}^k \zeta_j \{ \text{sm}(\otimes_{j=0}^k \vec{\eta}_j) \circ \text{sm}(\otimes_{j=0}^k \theta_j) \} / \text{occ}_{\otimes_{j=0}^k \zeta_j'}(s\vec{t}) \}$ and $\text{occ}_{\otimes_{j=0}^k \zeta_j}(s\vec{t}) = \otimes_{j=0}^k \zeta_j \{ \text{sm}(\otimes_{j=0}^k \vec{\eta}_j) \circ \text{sm}(\otimes_{j=0}^k \theta_j) \}$.

By definition $(s\vec{q})^{(\otimes_{j=0}^k \eta_j) \circ \theta} = s\vec{t} [\zeta \{ \text{sm}(\otimes_{j=0}^k \eta_j) \circ \theta \} / \otimes_{j=0}^k \zeta_j']$. Again, by Proposition 4.2.6,

$$s\vec{t} [\zeta \{ \text{sm}(\otimes_{j=0}^k \eta_j) \circ \theta \} / \otimes_{j=0}^k \zeta_j'] = s\vec{t} \{ \zeta \{ \text{sm}(\otimes_{j=0}^k \theta_j) \circ \text{sm}(\otimes_{j=0}^k \eta_j) \circ \theta \} / \text{occ}_{\otimes_{j=0}^k \zeta_j'}(s\vec{t}) \}.$$

and by contravariance of right action $\text{sm}(\otimes_{j=0}^k \eta_j) \circ \theta = \text{sm}(\theta) \circ \text{sm}(\otimes_{j=0}^k \eta_j)$. Then we can conclude by Lemma 4.2.3.

2. Let

$$\pi = \frac{\zeta_0 : \eta_0 : \Gamma_0 \vdash s : \vec{a} \Rightarrow a \quad \left(\begin{array}{c} \pi_0 \\ \vdots \\ \zeta_i : \eta_i : \Gamma_i \vdash t_i : a_i \end{array} \right)_{i=1}^k \quad \eta : \zeta : \Delta \rightarrow \otimes_{j=0}^k \zeta_j : \otimes_{j=0}^k \Gamma_j}{\zeta : (\otimes_{j=0}^k \eta_j) \circ \eta : \Delta \vdash p = (s\langle t_1, \dots, t_k \rangle)^\eta : a}$$

Now let $\pi' =$

$$\zeta'_0 : \eta'_0 : \Gamma'_0 \vdash s' : \vec{a}' \Rightarrow a' \quad \left(\begin{array}{c} \pi'_0 \\ \vdots \\ \zeta'_i : \eta'_i : \Gamma'_i \vdash t'_i : a'_i \end{array} \right)_{i=1}^{k'} \quad \eta' : \zeta' : \Delta' \rightarrow \otimes_{j=0}^k \zeta'_j : \otimes_{j=0}^k \Gamma'_j \\ \hline \zeta' : (\otimes_{j=0}^k \eta'_j) \circ \eta' : \Delta' \vdash p' = (s'\langle t'_1, \dots, t'_{k'} \rangle)^{\eta'} : a'$$

By hypothesis we have $k = k'$. Let $\zeta' = \otimes_{j=0}^k \chi_j = \otimes_{j=0}^k \zeta''_j$ be a repetitions-free tuple and $\eta_j : \zeta_j : \Gamma_j \rightarrow \chi_j : \Delta'_j, \eta'_j : \zeta'_j : \Gamma'_j \rightarrow \zeta''_j : \Delta''_j$. Consider $CF(\pi)_{\zeta'} =$

$$\chi_0 : \mathbf{1}_{\Delta'_0} : \Delta'_0 \vdash s^{\chi_0} : \vec{a} \Rightarrow a \quad \left(\begin{array}{c} CF(\pi_0)/\eta_0^{\chi_0} \\ \vdots \\ \chi_i : \mathbf{1}_{\Delta'_i} : \Delta'_i \vdash t_i^{\chi_i} : a_i \end{array} \right)_{i=1}^k \quad (\otimes_{j=0}^k \eta_j) \circ \eta : \zeta : \Delta \rightarrow (\otimes_{j=0}^k \chi_j) : (\otimes_{j=0}^k \Delta'_j) \\ \hline \zeta : (\otimes_{j=0}^k \eta_j) \circ \eta : \Delta \vdash (s^{\chi_0} \langle t_1^{\chi_1}, \dots, t_k^{\chi_k} \rangle)^{(\otimes_{j=0}^k \eta_j) \circ \eta} : a$$

now consider $CF(\pi')_{\zeta'} =$

$$\zeta''_0 : \mathbf{1}_{\Delta''_0} : \Delta''_0 \vdash (s')^{\zeta''_0} : \vec{a}' \Rightarrow a' \quad \left(\begin{array}{c} CF(\pi'_0)/(\eta'_0)^{\zeta''_0} \\ \vdots \\ \zeta''_i : \mathbf{1}_{\Delta''_i} : \Delta''_i \vdash (t'_i)^{\zeta''_i} : a'_i \end{array} \right)_{i=1}^k \quad (\otimes_{j=0}^k \eta'_j) \circ \eta' : \zeta' : \Delta' \rightarrow (\otimes_{j=0}^k \zeta''_j) : (\otimes_{j=0}^k \Delta''_j) \\ \hline \zeta' : (\otimes_{j=0}^k \eta'_j) \circ \eta' : \Delta' \vdash ((s')^{\zeta''_0} \langle (t'_1)^{\zeta''_1}, \dots, (t'_k)^{\zeta''_k} \rangle)^{(\otimes_{j=0}^k \eta'_j) \circ \eta'} : a'$$

We set $\theta = (\otimes_{j=0}^k \eta_j) \circ \eta = (\otimes_{j=0}^k \eta'_j) \circ \eta'$. By hypothesis and the former point of this lemma

$$(s^{\chi_0} \langle t_1^{\chi_1}, \dots, t_k^{\chi_k} \rangle)^\theta = ((s')^{\zeta''_0} \langle (t'_1)^{\zeta''_1}, \dots, (t'_k)^{\zeta''_k} \rangle)^\theta.$$

$$\text{occ}_{\otimes \chi_j} ((s^{\chi_0} \langle t_1^{\chi_1}, \dots, t_k^{\chi_k} \rangle)) = \text{occ}_{\otimes \zeta''_j} \left((s^{\zeta''_0} \langle t_1^{\zeta''_1}, \dots, t_k^{\zeta''_k} \rangle) \right)$$

In particular then $(s^{\chi_0})^\theta = (s^{\zeta''_0})^\theta$ and $((t_i)^{\chi_i})^\theta = ((t'_i)^{\zeta''_i})^\theta$. We apply Proposition 4.2.4 and 4.2.6 we get

$$(s')^{\zeta''_0} = s^{\chi_0} \quad (t'_i)^{\zeta''_i} = t_i^{\chi_i}. \quad (1)$$

We now prove that $\Delta'_j = \Delta''_j$ for $0 \leq j \leq k$. Now consider $\vec{x}_i : \vec{a}_i \in \zeta : \Delta$, $\vec{x}'_i : \vec{a}_i \in \zeta' : \Delta$ and $f_i : \vec{x}_i : \vec{a}_i \rightarrow \bigoplus \vec{y}_{j,i} : \bigoplus \vec{a}_{j,i}$ with $f_i \in (\bigotimes_{j=0}^k \eta_j) \circ \eta = (\bigotimes_{j=0}^k \eta'_j) \circ \eta'$. Then $f_i : \vec{x}'_i : \vec{a}_i \rightarrow \bigoplus \vec{y}'_{j,i} : \bigoplus \vec{a}'_{j,i}$ with $\bigoplus \vec{y}'_{j,i} : \bigoplus \vec{a}'_{j,i} = \bigoplus \vec{y}_{j,i} : \bigoplus \vec{a}_{j,i} \in \bigotimes \Delta''_j$. We know that $\text{occ}_{\vec{x}_i}(p) = \text{occ}_{\vec{x}'_i}(p) = \vec{x}_i\{\text{sm}(f_i)\}$ and that we can decompose $\vec{x}_i\{\text{sm}(f_i)\} = \bigoplus_{j=0}^k \vec{x}_{j,i}$ with $\vec{x}_{0,i} = \text{occ}_{\vec{x}_i}(((s)^{x_0})^\theta) = \text{occ}_{\vec{x}_i}(((s')^{\zeta'_0})^\theta)$. Moreover f_i defines a morphism $f_i : \vec{x}_i : \vec{a}_i \rightarrow \vec{x}_i\{\text{sm}(f_i)\} = \bigoplus_{j=0}^k \vec{x}_{j,i} : \bigoplus \vec{a}_{j,i}$. Then $\vec{x}_{j,i} : \vec{a}_{j,i}$ and $\vec{x}'_{j,i} : \vec{a}'_{j,i}$. Now, since the type assignment is functional on context variable lists, we get $\vec{a}_{j,i} = \vec{a}'_{j,i}$. Then we have $\Delta'_j = \Delta''_j$ and we can apply the hypothesis and get $\vec{a} = \vec{a}'$, $a = a'$.

Suppose now that $\zeta = \zeta'$, by hypothesis we get

$$(CF(\pi_j)/\eta_j^{x_j})\{\eta'_j\} \sim (CF(\pi'_0)/(\eta'_0)^{\zeta'_0})\{\eta'_0\}.$$

By Congruence we can conclude that $\pi \sim \pi'$. □

B.3. Actions and Congruence under Reduction of Points

Erasing Subtyping in Points We can extend also the operation π/η^ζ to points in a completely straightforward way. If $\varphi = \langle \eta, p \rangle \in \Omega_\zeta(\Delta, a)$ then we fix a repetitions-free variable tuple ζ' such that $\eta : \zeta : \Delta \rightarrow \zeta' : \text{trg}(\eta)$. We then define $\varphi/\eta^{\zeta'} = \langle 1_{\text{trg}(\eta)}, p^{\zeta'} \rangle \in \Omega_{\zeta'}(\text{trg}(\eta), a)$. We recall that $p^{\zeta'} = p\{\zeta'/\text{occ}_\zeta(p)\}$. In particular, it is worth noting that now $(\varphi/\eta^\zeta)\{\eta\} = \varphi$.

Congruence over Points

Proposition B.7. *Let $\varphi \in \Omega_\zeta(\Delta, a)$. Let $\eta : \zeta' : \Delta' \rightarrow \zeta : \Delta$, $f : a \rightarrow a'$. The following statements hold.*

1. *If $\varphi \sim \varphi'$ then $\varphi\{\eta\} \sim \varphi'\{\eta\}$.*
2. *If $\varphi \sim \varphi'$ then $[f]\varphi \sim [f]\varphi'$.*
3. *If $\varphi \sim \varphi'$ and $\varphi \in \Omega_{\zeta \triangleright \vec{x}_i}(\Delta \triangleright \vec{a}_i, a)$, $\vec{\psi}, \vec{\psi}' \in \Omega_{\vec{\zeta}}(\vec{\Gamma}, \vec{a}_i)$ such that $\vec{\psi} \sim \vec{\psi}'$. Then $\varphi[\vec{\psi}/\vec{x}_i] \sim \varphi'[\vec{\psi}'/\vec{x}_i]$.*

Proof. By induction on the definition of congruence $\varphi \sim \varphi'$. We will prove only the application congruence base case, since the others are direct corollaries of the IH and Corollary 4.2.25.

1. Immediate by definition of congruence and morphisms precomposition.
2. Let $\varphi =$

$$\langle (([\langle \alpha, \vec{f} \rangle \Rightarrow 1_a] \eta_0) \otimes \bigotimes_{i=1}^{k'} \eta_i) \circ \eta, ((([\langle \alpha, \vec{f} \rangle \Rightarrow 1_a] p) \langle q_1, \dots, q_{k'} \rangle))^n \rangle$$

and $\varphi' =$

$$\langle (\eta_0 \otimes \bigotimes_{i=1}^{k'} ([f_i] \eta_{\alpha(i)})) \circ ((1_{\Gamma_0} \otimes \alpha^*) \circ \eta), (p([f_1] q_{\alpha(1)}, \dots, [f_{k'}] q_{\alpha(k')}))^{(1_{\Gamma_0} \otimes \alpha^*) \circ \eta} \rangle.$$

By definition of left action and corollary 4.2.25 we get $[f]\varphi =$

$$\langle ([\langle \alpha, \vec{f} \rangle \Rightarrow f] \eta_0) \otimes \bigotimes_{i=1}^{k'} \eta_i \circ \eta, ([\langle \alpha, \vec{f} \rangle \Rightarrow f] p) \langle q_1, \dots, q_{k'} \rangle \rangle^\eta$$

and $[f]\varphi' =$

$$\langle ([1 \Rightarrow f] \eta_0) \otimes \bigotimes_{i=1}^{k'} ([f_i] \eta_{\alpha(i)}) \circ ((1_{\Gamma_0} \otimes \alpha^*) \circ \eta), ([1 \Rightarrow f] p) \langle [f_1] q_{\alpha(1)}, \dots, [f_{k'}] q_{\alpha(k')} \rangle \rangle^{(1_{\Gamma_0} \otimes \alpha^*) \circ \eta}$$

then we can conclude, since by Corollary 4.2.25 we have $\langle [\langle \alpha, \vec{f} \rangle \Rightarrow f] \eta_0, [\langle \alpha, \vec{f} \rangle \Rightarrow f] p \rangle = \langle \alpha, \vec{f} \rangle \Rightarrow f \langle \eta_0, p \rangle = \langle \alpha, \vec{f} \rangle \Rightarrow 1_{a'} \langle [1 \Rightarrow f] \eta_0, [1 \Rightarrow f] p \rangle$.

3. Let $\varphi =$

$$([f \Rightarrow 1] \varphi'_0) \langle \varphi_1, \dots, \varphi_k \rangle \circ \eta^{\triangleright g_i}$$

and $\varphi' =$

$$(\varphi'_0 \langle [f_1] \varphi'_{\alpha(1)}, \dots, [f_{k'}] \varphi'_{\alpha(k')} \rangle) \circ (1 \otimes (\alpha^*)^{\triangleright \alpha_i}) \circ \eta^{\triangleright g_i}$$

for some $f = \langle \alpha, f_1, \dots, f_{k'} \rangle \in \text{mrp}(SD)$, $k' \in \mathbb{N}$ that satisfies the conditions of congruence. We recall that α^* is a natural transformation $\alpha^* = \{ \alpha_{\langle \Gamma_1, \dots, \Gamma_k \rangle}^* \mid \alpha_{\langle \Gamma_1, \dots, \Gamma_k \rangle}^* : \bigotimes_{j=1}^k \Gamma_j \rightarrow \bigotimes_{j'=1}^{k'} \Gamma_{\alpha(j')} \}$. Here, we keep the parameter implicit¹. Then $\varphi[\vec{\psi}/\vec{x}_i] =$

$$([f \Rightarrow 1] \varphi'_0 \langle ([g_i] \vec{\psi})_0 / \vec{y}_{0,i} \rangle \langle \varphi_1 \langle ([g_i] \vec{\psi})_1 / \vec{y}_{1,i} \rangle, \dots, \varphi_k \langle ([g_i] \vec{\psi})_k / \vec{y}_{k,i} \rangle \rangle) \circ \eta^*$$

and $\varphi'[\vec{\psi}'/\vec{x}'_i] =$

$$(\varphi'_0 \langle ([g_i] \vec{\psi}')_0 / \vec{y}'_{0,i} \rangle \langle [f_1] \varphi_{\alpha(1)} \langle ([g_i] \vec{\psi}')_{\alpha(1)} / \vec{y}'_{\alpha(1),i} \rangle, \dots, [f_{k'}] \varphi_{\alpha(k')} \langle ([g_i] \vec{\psi}')_{\alpha(k')} / \vec{y}'_{\alpha(k'),i} \rangle \rangle) \circ ((1 \otimes \alpha^*) \circ \eta)^*$$

since $[\alpha_i \circ g_i] \vec{\psi}' = \bigoplus_{l=1}^{k'} ([g_i] \vec{\psi}')_{\alpha(l)}$. By Corollary 4.2.25 we get $[f \Rightarrow 1] \varphi'_0 \langle ([g_i] \vec{\psi}')_0 / \vec{y}'_{0,i} \rangle = [f \Rightarrow 1] (\varphi'_0 \langle ([g_i] \vec{\psi}')_0 / \vec{y}'_{0,i} \rangle)$ and

$$[f_l] \varphi_{\alpha(l)} \langle ([g_i] \vec{\psi}')_{\alpha(l)} / \vec{y}'_{\alpha(l),i} \rangle = [f_l] (\varphi_{\alpha(l)} \langle ([g_i] \vec{\psi}')_{\alpha(l)} / \vec{y}'_{\alpha(l),i} \rangle)$$

for $l \in [k']$. By hypothesis and the first point of this lemma, since $\vec{\psi} \sim \vec{\psi}'$,

1. In this case, we are considering $\alpha_{\langle \text{source}(\text{mrp}(\varphi))_j \rangle_{j=1}^k}^*$.

we get $[f_l](\varphi_{\alpha(l)}[[g_i]\vec{\psi}']_{\alpha(l)}/\vec{y}_{\alpha(l),i}) \sim [f_l](\varphi_{\alpha(l)}[[g_i]\vec{\psi}]_{\alpha(l)}/\vec{y}_{\alpha(l),i})$. By definition $((1 \otimes \alpha^*) \circ \eta)^* = \vec{\tau}_1 \circ (((1 \otimes (\alpha^*)^{-\alpha_i}) \circ \eta^{-g_i}) \otimes \text{sm}((1 \otimes g_i)^*))$ and $\eta^* = \vec{\tau}_2 \circ (\eta^{-g_i} \otimes \text{sm}(g_i)^*)$ with $\tau_1 \in \mathfrak{S}_{k'+1}, \tau_2 \in \mathfrak{S}_{k+1}$ being appropriate permutations. We observe that $\text{sm}(g \circ f)^* = \text{sm}(g)^* \circ \text{sm}(f)^*$ for all composable morphisms f, g . Hence, by functoriality of tensor product, $((1 \otimes (\alpha^*)^{-\alpha_i}) \circ \eta^{-g_i}) \otimes \text{sm}((1 \otimes \alpha_i) \circ g_i)^* = ((1 \otimes (\alpha^*)^{-\alpha_i}) \otimes \text{sm}(1 \otimes \alpha_i)^*) \circ (\eta^{-g_i} \otimes \text{sm}(g_i)^*)$. We remark that $\text{sm}(\alpha_i)^* = (\alpha^*)^{-\alpha_i}_{\langle \text{source}(\text{mrp}(\varphi_j)) \otimes \text{source}(\text{mrp}(\vec{\psi}_j)) \rangle_{i=1}^k}$ where $(\alpha^*)^{-\alpha_i}_{\langle \text{source}(\text{mrp}(\varphi_j)) \otimes \text{source}(\text{mrp}(\vec{\psi}_j)) \rangle_{i=1}^k}$ is the natural transformation α^* evaluated in $\langle \text{source}(\text{mrp}(\varphi_j)) \otimes \text{source}(\text{mrp}(\vec{\psi}_j)) \rangle_{i=1}^k$. That is the following morphism:

$$\begin{aligned} \text{sm}(\alpha_i)^* : \bigotimes_{j=1}^k (\text{source}(\text{mrp}(\varphi))_j \otimes (\text{source}(\text{mrp}([g_i]\vec{\psi}'))_j)) \rightarrow \\ \bigotimes_{j'=1}^{k'} (\text{source}(\text{mrp}(\varphi))_{\alpha(j')} \otimes (\text{source}(\text{mrp}([g_i]\vec{\psi}'))_{\alpha(j')})). \end{aligned}$$

We can then conclude, since $\vec{\tau}_1 \circ ((1 \otimes (\alpha^*)^{-\alpha_i}) \otimes (1 \otimes (\alpha^*)^{-\alpha_i})) = (1 \otimes \text{sm}(\alpha_i)^*) \circ \vec{\tau}_2$.

□

Congruence and Reduction We introduce some structures that we shall use in dealing with the intricacies of substitution.

Let $Sub^\Omega(\zeta : \Delta, a) =$

$$\sum_{\vec{x}:\vec{a}=(a_1,\dots,a_k) \in SD} \sum_{\zeta_1:\Gamma_1,\dots,\zeta_k:\Gamma_k \in (SD^{\text{len}(\zeta)})^*} \Omega_{\zeta_0 \oplus \vec{x}}(\Gamma_0, \vec{a} \Rightarrow a) \times \prod_{i=1}^k \Omega_{\zeta_i}(\Gamma_i, a_i) \times SD^{\text{len}(\zeta)}(\Delta, \bigotimes_{j=0}^k \Gamma_j)$$

the former set is clearly reminiscent of the one introduced in Chapter 3 in order to deal with substitution of type derivations.

We consider the smallest equivalence relation on $Sub^\Omega(\zeta : \Delta, a)$ generate by the following two rules:

$$\langle \vec{x} : \vec{a}, \langle \zeta_0 : \Gamma_0, \dots, \zeta_{\text{len}(\vec{a})} : \Gamma_{\text{len}(\vec{a})} \rangle, \varphi, \langle \psi_1 \{ \eta_1 \}, \dots, \psi_{\text{len}(\vec{a})} \{ \eta_{\text{len}(\vec{a})} \} \rangle, \eta \rangle \sim \quad (.2)$$

$$\langle \vec{x} : \vec{a}, \langle \zeta'_0 : \Gamma'_0, \dots, \zeta'_{\text{len}(\vec{a})} : \Gamma'_{\text{len}(\vec{a})} \rangle, \varphi, \langle \psi_1, \dots, \psi_{\text{len}(\vec{a})} \rangle, (\bigotimes_{j=0}^{\text{len}(\vec{a})} \eta_j) \circ \eta \rangle$$

where $\eta_j : \Gamma'_j \rightarrow \Gamma_j$.

$$\langle \vec{x} : \vec{a}, \vec{\zeta} : \vec{\Gamma}, \varphi \{ 1_{\Gamma_0} \oplus \langle f \rangle \}, \langle \psi_1, \dots, \psi_{\text{len}(\vec{a})} \rangle, \eta \rangle \sim \quad (.3)$$

$$\langle \vec{x} : \vec{a}, (\vec{\zeta} : \vec{\Gamma}) \{ id \oplus \text{sm}(h) \}, \varphi, ([f] \langle \psi_1, \dots, \psi_{\text{len}(\vec{a})} \rangle), (1 \otimes \text{sm}(f)^*) \circ \eta \rangle$$

where $f : \vec{a} \rightarrow \vec{a}'$ and $\vec{\zeta} : \vec{\Gamma} = \langle \zeta_0 : \Gamma_0, \dots, \zeta_{\text{len}(\vec{a})} : \Gamma_{\text{len}(\vec{a})} \rangle$. We shall make an abuse of language and set $\varphi\{1_{\Gamma_0} \oplus \langle h \rangle\} = [h \Rightarrow 1]_{\vec{x}}\varphi$.

We additionally impose two standard contextual rules:

$$\frac{\varphi \sim \varphi' \quad \vec{\psi} \sim \vec{\psi}'}{\langle \varphi, \vec{\psi}, \eta \rangle \sim \langle \varphi', \vec{\psi}', \eta \rangle} \quad \frac{\psi_1 \sim \psi'_1 \dots \psi_k \sim \psi'_k}{\langle \psi_1, \dots, \psi_k \rangle \sim \langle \psi'_1, \dots, \psi'_k \rangle}$$

whenever the points are well-defined. Since we added contextual rules, we shall call the former equivalence relation a *congruence*.

Clearly, the former rules are just a variation of the kind of congruences to which we got used to. We need to impose also a condition on contexts because the elements of $Sub^\Omega(\zeta : \Delta, a)$ are not points. As should be clear by now, the former equivalence determines a coend.

We observe that if $\langle \vec{x} : \vec{a}, \langle \zeta_0 : \Gamma_0, \vec{\zeta} : \vec{\Gamma} \rangle, \varphi, \vec{\psi}, \eta \rangle \sim \langle \vec{x}' : \vec{a}, \langle \zeta'_0 : \Gamma'_0, \vec{\zeta}' : \vec{\Gamma}' \rangle, \varphi', \vec{\psi}', \eta' \rangle$ then $(\lambda \vec{x} : f.\varphi\vec{\psi}) \circ \eta \sim (\lambda \vec{x}' : f.\varphi'\vec{\psi}') \circ \eta$.

Lemma B.8. *Let $\langle \varphi, \vec{\psi}, \eta \rangle, \langle \varphi', \vec{\psi}', \eta' \rangle \in Sub_{\vec{x}:\vec{a}}^\Omega(\zeta : \Delta, a)$. If $\langle \varphi, \vec{\psi}, \eta \rangle \sim \langle \varphi', \vec{\psi}', \eta' \rangle$ then $\varphi[\vec{\psi}/\vec{x}] \sim \varphi'[\vec{\psi}'/\vec{x}']$.*

Proof. Straightforward induction on the structure of φ . □

Lemma B.9 (Uniformity of Substitution). *Let $\langle \varphi, \vec{\psi}, \eta \rangle \in Sub_{\vec{x}:\vec{a}}^\Omega(\zeta : \Delta, a)$, $\langle \varphi', \vec{\psi}', \eta' \rangle \in Sub_{\vec{x}':\vec{a}'}^\Omega(\zeta : \Delta, a)$ for some $n \in \mathbb{N}$, $\langle \zeta : \Delta, a \rangle \in (SD^n)^* \times D$, $\vec{x} : \vec{a}; \vec{x}' : \vec{a}' \in SD^*$. If $\varphi[\vec{\psi}/\vec{x}] \circ \eta \sim \varphi'[\vec{\psi}'/\vec{x}'] \circ \eta'$ and $\varphi \supset \varphi', \vec{\psi} \supset \vec{\psi}'$ then $\langle \varphi, \vec{\psi}, \eta \rangle \sim \langle \varphi', \vec{\psi}', \eta' \rangle$.*

Proof. By induction on φ . If $\varphi = \langle \eta \oplus \langle f \rangle, x_{\text{sm}(f)(1)} \rangle \in \Omega_{\zeta \oplus \langle \vec{x} \rangle}(\Delta \oplus \langle \text{source}(f) \rangle, a)$, since $\psi \supset \psi'$, we have that $\varphi' = \langle \eta' \oplus \langle f' \rangle, x'_{\text{sm}(f')(1)} \rangle \in \Omega_{\zeta \oplus \langle \vec{x}' \rangle}(\Delta \oplus \langle \text{source}(f') \rangle, a)$. Hence

$$\begin{aligned} \rho &= \langle \varphi, \vec{\psi}, \eta \rangle = \langle \langle \eta \oplus \langle f \rangle, x'_{\text{sm}(f)(1)} \rangle, \vec{\psi}, \eta \rangle \\ \rho' &= \langle \varphi', \vec{\psi}', \eta' \rangle = \langle \langle \eta' \oplus \langle f' \rangle, x'_{\text{sm}(f')(1)} \rangle, \vec{\psi}', \eta' \rangle. \end{aligned}$$

We fix a variable z . By definition of congruence, we have

$$\begin{aligned} \rho &\sim \rho_1 = \langle \langle \eta \oplus \langle 1_a \rangle, z \rangle, [f]\vec{\psi}, (1 \otimes \text{sm}(f))^* \circ \eta \rangle \\ \rho' &\sim \rho_2 = \langle \langle \eta' \oplus \langle 1_a \rangle, z \rangle, [f']\vec{\psi}', (1 \otimes \text{sm}(f'))^* \circ \eta' \rangle. \end{aligned}$$

Let $f = \langle \alpha, g \rangle, f' = \langle \alpha', g' \rangle$. Consider

$$\begin{aligned} \psi_1\{\vec{\psi} &= [g]\psi_{\alpha(1)}/\langle z \rangle\}(1 \otimes \text{sm}(f))^* \circ \eta = ([g]\psi_{\alpha(1)})\{\pi_{2,2}\} \circ (1 \otimes \alpha^*) \circ \eta \\ \psi'_1\{[f']\vec{\psi}' &= [g']\psi'_{\alpha'(1)}/\langle z \rangle\}(1 \otimes \text{sm}(f'))^* \circ \eta' = ([g']\psi'_{\alpha'(1)})\{\pi'_{2,2}\} \circ (1 \otimes (\alpha')^*) \circ \eta' \end{aligned}$$

By universal property we have

$$\pi_{2,2} \circ (1 \otimes \alpha^*) = \pi_{\alpha(1), \text{len}(\text{source}(f))+1}$$

$$\pi'_{2,2} \circ (1 \otimes \alpha'^*) = \pi_{\alpha'(1), \text{len}(\text{source}(f'))+1}$$

hence, by definition of right action

$$\psi_1\{[g]\psi_{\alpha(1)}/\langle z \rangle\}(1 \otimes \text{sm}(f)^*) \circ \eta = ([g]\psi_{\alpha(1)})\{\pi_{\alpha(1), \text{len}(\text{source}(f))+1} \circ \eta\}$$

$$\psi'_1\{[g']\psi'_{\alpha'(1)}/\langle z \rangle\}(1 \otimes \text{sm}(f')^*) \circ \eta' = ([g']\psi'_{\alpha'(1)})\{\pi_{\alpha'(1), \text{len}(\text{source}(f'))+1} \circ \eta'\}$$

by hypothesis and the former lemma we have $\psi_1\{[g]\psi_{\alpha(1)}/\langle z \rangle\}(1 \otimes \text{sm}(f)^*) \circ \eta \sim \psi_2\{[g']\psi'_{\alpha'(1)}/\langle z \rangle\}(1 \otimes \text{sm}(f')^*) \circ \eta'$. By definition of congruence

$$\rho_1 \sim \langle \langle \langle 1_{\vec{y}}, 1_a \rangle, z \rangle, ([g]\psi_{\alpha(1)})\{\pi_{\alpha(1), \text{len}(\text{source}(f))+1} \circ \eta\}, 1 \rangle$$

$$\rho_2 \sim \langle \langle \langle 1_{\vec{y}}, 1_a \rangle, z \rangle, ([g']\psi'_{\alpha'(1)})\{\pi_{\alpha'(1), \text{len}(\text{source}(f'))+1} \circ \eta'\}, 1 \rangle$$

Since by hypothesis $([g]\psi_{\alpha(1)})\{\pi_{\alpha(1), \text{len}(\text{source}(f))+1} \circ \eta\} \sim ([g']\psi'_{\alpha'(1)})\{\pi_{\alpha'(1), \text{len}(\text{source}(f'))+1} \circ \eta'\}$ we can conclude by transitivity and contextuality of congruence.

If $\varphi = \lambda \vec{y} : g.\psi_1$ with $\vec{y} \neq \vec{x}$ then $\varphi' = \lambda \vec{y}' : g'.\psi_2$ for some ψ_2 such that $\psi_1 \supset \psi_2$. The result is then a direct consequence of the IH.

If $\varphi = (\varphi_0 \langle \varphi_1, \dots, \varphi_l \rangle) \circ (\theta \oplus \langle g \rangle)$ then $\varphi' = (\varphi'_0 \langle \varphi'_1, \dots, \varphi'_{l'} \rangle) \circ (\theta' \oplus \langle g' \rangle)$. We made g, g' explicit because they are the morphism at the contextual position of \vec{x}, \vec{x}' and hence they will act on the lists of terms that will be substituted. By Hypothesis we have that

$$\begin{aligned} & (\varphi[\vec{\psi}/\vec{x}]) \circ \eta = \\ & (\varphi_0([g]\vec{\psi})_0/\vec{y}_0 \langle \varphi_1([g]\vec{\psi})_1/\vec{y}_1, \dots, \varphi_l([g]\vec{\psi})_l/\vec{y}_l \rangle) \circ \theta^* \circ \eta \sim \\ & (\varphi'[\vec{\psi}'/\vec{x}']) \circ \eta' = \\ & (\varphi'_0([g']\vec{\psi}')_0/\vec{y}'_0 \langle \varphi'_1([g']\vec{\psi}')_1/\vec{y}'_1, \dots, \varphi'_{l'}([g']\vec{\psi}')_{l'}/\vec{y}'_{l'} \rangle) \circ (\theta')^* \circ \eta' \end{aligned}$$

where $\vec{y}_j, \vec{y}'_{j'}$, for $0 \leq j \leq l, 0 \leq j' \leq l'$ are antecedents of respectively \vec{x}, \vec{x}' . Suppose that the congruence is witnessed by the application base rule². We then have that there exists $h \in \text{mrp}(SD)$, such that

$$\begin{aligned} & (\varphi[\vec{\psi}/\vec{x}]) \circ \eta = \\ & (([h \Rightarrow 1]\varphi'_0([g']\vec{\psi}')_0/\vec{y}'_0) \langle \varphi_1([g]\vec{\psi})_1/\vec{y}_1, \dots, \varphi_l([g]\vec{\psi})_l/\vec{y}_l \rangle) \circ \theta^* \circ \eta \\ & (\varphi'[\vec{\psi}'/\vec{x}']) \circ \eta' = \\ & (\varphi'_0([g]\vec{\psi}')_0/\vec{y}'_0) \langle [h] \langle \varphi_1([g]\vec{\psi})_1/\vec{y}_1, \dots, \varphi_l([g]\vec{\psi})_l/\vec{y}_l \rangle \rangle \circ (1 \otimes \text{sm}(h)^*) \circ ((\theta^*) \circ \eta) \end{aligned}$$

By definition of congruence (Rule .3), we get

$$\langle (\varphi_0 \langle \varphi_1, \dots, \varphi_l \rangle) \circ (\theta \oplus \langle g \rangle), \vec{\psi}, \eta \rangle \sim$$

2. We will not consider the contextual cases in what follows, since the results in that context are a direct corollary of the IH.

$$\rho = \langle (\varphi_0 \langle \varphi_1, \dots, \varphi_l \rangle) \circ (\theta \oplus \langle 1_{\text{trg}(g)} \rangle), [g] \vec{\psi}, (1 \otimes \text{sm}(g)^*) \circ \eta \rangle$$

and

$$\begin{aligned} & \langle (\varphi'_0 \langle \varphi'_1, \dots, \varphi'_l \rangle) \circ (\theta' \oplus \langle g' \rangle), \vec{\psi}', \eta' \rangle \sim \\ \rho' & = \langle (\varphi'_0 \langle \varphi'_1, \dots, \varphi'_l \rangle) \circ (\theta' \oplus \langle 1_{\text{trg}(g')} \rangle), [g'] \vec{\psi}', (1 \otimes \text{sm}(g')^*) \circ \eta' \rangle \end{aligned}$$

By definition of congruence and Proposition B.7, we then have the following

$$\rho_1 = \langle (([h \Rightarrow 1] \varphi'_0) \langle \varphi_1, \dots, \varphi_l \rangle) \circ (\theta \oplus \langle 1 \rangle), ([g'] \vec{\psi}')_0 \oplus \left(\bigoplus_{i=1}^l ([g] \vec{\psi})_i \right), (1 \otimes \text{sm}(g)^*) \circ \eta \rangle \sim$$

$$\rho_2 =$$

$$\langle (\varphi'_0([h] \langle \varphi_1, \dots, \varphi_l \rangle)) \circ ((1 \otimes \text{sm}(h)^*) \circ (\theta \oplus \langle 1 \rangle)), ([g'] \vec{\psi}')_0 \oplus \left(\bigoplus_{i'=\text{sm}(h)(1)}^{\text{sm}(h)(l')} ([g] \vec{\psi})_{\text{sm}(h)(i')} \right), (1 \otimes \text{sm}(g)^*) \circ \eta \rangle$$

If we prove that $\rho \sim \rho_1$ and $\rho' \sim \rho_2$ we can then conclude by transitivity.

We prove that $\rho \sim \rho_1$ By IH and Proposition B.7, we have

$$\langle \varphi_0, ([g] \vec{\psi})_0, 1 \rangle \sim \langle [h \Rightarrow 1] \varphi'_0, ([g'] \vec{\psi}')_0, 1 \rangle$$

then there exists $r_0 \in \text{mrp}(SD)$ such that either

$$\varphi_0 = ([r_0 \Rightarrow 1]_{\vec{y}'_0}([h \Rightarrow 1] \varphi'_0)) \quad ([g'] \vec{\psi})_0 = [r_0]([g] \vec{\psi})_0$$

or

$$[h \Rightarrow 1] \varphi'_0 = [r_0 \Rightarrow 1]_{\vec{y}'_0} \varphi_0 \quad [r_0]([g'] \vec{\psi})_0 = ([g] \vec{\psi})_0$$

we do the first case, the second one being completely symmetric. We set $r = r_0 \oplus 1 : \vec{y}'_0 \oplus \bigoplus_{i=1}^l \vec{y}'_i \rightarrow \vec{y}'_0 \oplus \bigoplus_{i=1}^l \vec{y}'_i$ ³. We set $\vec{y}' = \vec{y}'_0 \oplus \bigoplus_{i=1}^l \vec{y}'_i$. Then, by definition of congruence and IH we get

$$\begin{aligned} & (\varphi_0 \langle \varphi_1, \dots, \varphi_l \rangle) \circ (\theta \oplus \langle 1_{\text{trg}(g)} \rangle) = \\ & [r \Rightarrow 1]_{\vec{y}'}(([h \Rightarrow 1] \varphi'_0) \langle \varphi_1, \dots, \varphi_l \rangle) \circ (\theta \oplus \langle 1_{\text{trg}(g)} \rangle) \end{aligned}$$

and

$$[r]([g] \vec{\psi}) = \bigoplus_{i=1}^l ([g] \vec{\psi})_i.$$

We can then conclude that $\rho \sim \rho_1$. An analogous argument, applying the IH, works also for ρ', ρ_2 . We can then conclude. \square

Theorem B.10 (Uniformity of Reduction). *Let $\varphi_1, \varphi_2 \in \Omega_\zeta(\Delta, a)$, $M, N \in \Lambda$ such that $\varphi_1 \rightarrow_p \varphi'_1 \triangleleft M \rightarrow_\beta N$ and $\varphi_2 \rightarrow_p \varphi'_2 \triangleleft M \rightarrow_\beta N$. If $\varphi'_1 \sim \varphi'_2$ then $\varphi_1 \sim \varphi_2$.*

Proof. Direct corollary of the former lemma. \square

3. Where we keep the type list implicit to not overcharge the text.

Lemma B.11. *Let $\tilde{\varphi} \in \mathcal{T}_{\text{rig}}(M\{N/x\})_{\vec{x}}(\zeta : \Delta, a)$ with $\text{len}(\vec{x}) = n$. There exist $\vec{a} = \langle a_1, \dots, a_k \rangle \in SD, \zeta_0 \oplus \langle \vec{y} \rangle : \Gamma_0 \oplus \langle \vec{a}_0 \rangle, \dots, \zeta_k : \Gamma_k \in (SD^n)^*, \eta : \zeta : \Delta \rightarrow \otimes \zeta_j : \Gamma_j$ and $\tilde{\varphi}' \in \mathcal{T}_{\text{rig}}(M)_{\vec{x} \oplus \langle x \rangle}(\zeta_0 \oplus \langle \vec{y} \rangle : \Gamma_0 \oplus \langle \vec{a}_0 \rangle, a), \tilde{\psi}_i \in \mathcal{T}_{\text{rig}}(N)_{\vec{x}}(\zeta_i : \Gamma_i, a_i)$ for $i \in [k]$ such that $\tilde{\varphi} = \varphi'[\vec{\psi}/\vec{y}] \circ \eta$.*

Proof. By induction on the structure of M . If $M = x$ then $\mathcal{T}_{\text{rig}}(M\{N/x\})_{\vec{x}}(\zeta : \Delta, a) = \mathcal{T}_{\text{rig}}(N)_{\vec{x}}(\zeta : \Delta, a)$. Given $\tilde{\varphi} = \langle \eta, q \rangle \in \mathcal{T}_{\text{rig}}(N)_{\vec{x}}(\zeta : \Delta, a)$, we consider $\varphi' = \langle \langle 1_{\emptyset}, \dots, 1_{\emptyset}, 1_a \rangle, y \rangle$ and we take $\eta = 1_{\Delta} : \zeta : \Delta \rightarrow \zeta : \Delta$. By definition $\varphi'[\langle \varphi \rangle / \langle y \rangle] \circ \eta = \varphi$.

If $M = \lambda y.M'$ then by definition $\varphi \in \mathcal{T}_{\text{rig}}(M\{N/x\})_{\vec{x}}(\zeta : \Delta, a)$ is of the shape $\lambda \vec{z} : f.\varphi'$ with $\varphi' \in \mathcal{T}_{\text{rig}}(M\{N/x\})_{\vec{x}}(\zeta \oplus \langle \vec{z} \rangle : \Delta \oplus \langle \vec{a} \rangle, a')$ for $a = \vec{a} \Rightarrow a'$ for some variable list \vec{z} and morphism f . The result is then an immediate corollary of the IH.

If $M = PQ$ then by definition $\varphi \in \mathcal{T}_{\text{rig}}(M\{N/x\})_{\vec{x}}(\zeta : \Delta, a) = \mathcal{T}_{\text{rig}}(P\{N/x\}Q\{n/x\})_{\vec{x}}(\zeta : \Delta, a)$ is of the shape $\psi_0 \langle \psi_1, \dots, \psi_k \rangle \circ \eta$ for some $\langle b_1, \dots, b_{k'} \rangle \in SD, \zeta_0 : \Delta_0, \dots, \zeta_{k'} : \Delta_{k'} \in (SD^n)^*, \psi_0 \in \mathcal{T}_{\text{rig}}(P\{N/x\})_{\vec{x}}(\zeta_0 : \Delta_0, \langle b_1, \dots, b_{k'} \rangle \Rightarrow a), \psi_l \in \mathcal{T}_{\text{rig}}(Q\{N/x\})_{\vec{x}}(\zeta_l : \Delta_l, b_l)$ for $l \in [k']$. By IH there exist $\vec{a}_0 = \langle a_{0,1}, \dots, a_{0,k_0} \rangle \in SD, \zeta_{0,0} \oplus \langle \vec{y}_0 \rangle : \Gamma_{0,0} \oplus \langle \vec{a}_0 \rangle, \dots, \zeta_{0,k} : \Gamma_{0,k} \in (SD^n)^*, \eta_0 : \zeta_0 : \Delta_0 \rightarrow \otimes \zeta_{0,j_0} : \Gamma_{j_0}$ and $\varphi'_0 \in \mathcal{T}_{\text{rig}}(P)_{\vec{x} \oplus \langle x \rangle}(\zeta_{0,0} \oplus \langle \vec{y}_0 \rangle : \Gamma_{0,0} \oplus \langle \vec{a}_0 \rangle, \vec{b} \Rightarrow a), \psi_{0,i_0} \in \mathcal{T}_{\text{rig}}(N)_{\vec{x}}(\zeta_{0,i_0} : \Gamma_{0,i_0}, a_{0,i_0})$ for $i_0 \in [k_0]$ such that $\psi_0 = \varphi'_0[\vec{\psi}_0/\vec{y}_0] \circ \eta_0$. Again, by IH there exist $\vec{a}_i = \langle a_{i,1}, \dots, a_{i,k_i} \rangle \in SD, \zeta_{i,0} \oplus \langle \vec{y}_i \rangle : \Gamma_{i,0} \oplus \langle \vec{a}_i \rangle, \dots, \zeta_{i,k} : \Gamma_{i,k} \in (SD^n)^*, \eta_i : \zeta_i : \Delta_i \rightarrow \otimes \zeta_{i,j_i} : \Gamma_{j_i}$ and $\varphi'_i \in \mathcal{T}_{\text{rig}}(Q)_{\vec{x} \oplus \langle x \rangle}(\zeta_{i,0} \oplus \langle \vec{y}_i \rangle : \Gamma_{i,0} \oplus \langle \vec{a}_i \rangle, b_i), \psi_{0,i_i} \in \mathcal{T}_{\text{rig}}(N)_{\vec{x}}(\zeta_{i,i_i} : \Gamma_{i,i_i}, a_{i,i_i})$ for $i_i \in [k_i]$ such that $\psi_i = \varphi'_i[\vec{\psi}_i/\vec{y}_i] \circ \eta_i$. Then

$$\varphi = ((\varphi'_0[\vec{\psi}_0/\vec{y}_0] \circ \eta_0)(\vec{\varphi}'[\vec{\psi}/\vec{y}] \circ \vec{\eta})) \circ \eta$$

Where $\vec{\varphi}' = \langle \varphi'_1, \dots, \varphi'_{k'} \rangle, \vec{\psi} = \langle \psi_1, \dots, \psi_{k'} \rangle, \vec{y} = \langle \vec{y}_1, \dots, \vec{y}_{k'} \rangle, \vec{\eta} = \langle \eta_1, \dots, \eta_{k'} \rangle$. By definition of substitution $\varphi'_0[\vec{\psi}_0/\vec{y}_0] \in \Omega_{\zeta_{0,0} \otimes \vec{\zeta}_0}(\Gamma_{0,0} \otimes \vec{\Gamma}_0, \vec{b} \Rightarrow b)$ and $\varphi'_i[\vec{\psi}_i/\vec{y}_i] \in \Omega_{\zeta_{i,0} \otimes \vec{\zeta}_i}(\Gamma_{i,0} \otimes \vec{\Gamma}_i, b_i)$. Consider $\varphi' = \varphi'_0 \vec{\varphi}' \in \Omega_{\otimes \zeta_{j,0} \oplus \langle \vec{y}_j \rangle}(\otimes \Gamma_{j,0} \oplus \langle \vec{a}_j \rangle, a)$. By definition and IH $\varphi'_0 \vec{\varphi}' \in \mathcal{T}_{\text{rig}}(PQ)_{\vec{x}}(\otimes \zeta_{j,0} \oplus \langle \vec{y}_j \rangle : \otimes \Gamma_{j,0} \oplus \langle \vec{a}_j \rangle, a)$. We set $\vec{a} = \oplus \vec{a}_j$. Then we take $\vec{\psi}' = \oplus \psi_j$ and $\eta' = (\otimes \eta_j) \circ \eta$ and we can conclude, since

$$\varphi = \varphi'[\vec{\psi}' / \oplus \vec{x}_j] \circ \eta'.$$

□

Lemma B.12. *Let $M, N \in \Lambda, \vec{x} \supseteq (\text{fv}(M)/\{x\}) \cup \text{fv}(N)$ and $x \notin \vec{x}$. Let $\vec{a} = \langle a_1, \dots, a_k \rangle \in SD, \zeta_0 \oplus \langle \vec{y} \rangle : \Gamma_0 \oplus \langle \vec{a}_0 \rangle, \dots, \zeta_k : \Gamma_k \in (SD^n)^*, \eta : \zeta : \Delta \rightarrow \otimes \zeta_j : \Gamma_j, \tilde{\varphi} \in \mathcal{T}_{\text{rig}}(M)_{\vec{x} \oplus \langle x \rangle}(\zeta_0 \oplus \langle \vec{y} \rangle : \Gamma_0 \oplus \langle \vec{a}_0 \rangle, a), \tilde{\psi}_i \in \mathcal{T}_{\text{rig}}(N)_{\vec{x}}(\zeta_i : \Gamma_i, a_i)$ for $i \in [k]$.*

Then $\varphi\{\vec{\psi} = \langle \psi_1, \dots, \psi_k \rangle / \vec{x}\} \in \mathcal{T}_{\text{rig}}(M\{N/x\})_{\vec{x}}(\zeta : \Delta, a)$.

Proof. By induction on the structure of M . Let $M = x$ and $\tilde{\varphi} = \langle \eta \oplus \langle f \rangle, x_{\text{sm}(f)(1)} \rangle \in \mathcal{T}_{\text{rig}}(M)_{\vec{x} \oplus \langle x \rangle}(\zeta_0 \oplus \langle \vec{y} \rangle : \Gamma_0 \oplus \langle \vec{a}_0 \rangle, a)$ where $x_{\text{sm}(f)(1)} \in \vec{x}$. We have, by definition

$\varphi[\vec{\psi}/\vec{x}] \circ \eta = \psi_{\text{sm}(f)(1)}\{\pi_{\text{sm}(f)(1), \text{len}(\vec{a})}\}\{\eta\}$. By definition of right action

$$\psi_{\text{sm}(f)(1)}\{\pi_{\text{sm}(f)(1), \text{len}(\vec{a})}\}\{\eta\} \in \mathcal{T}_{\text{rig}}(N)_{\vec{x}}(\zeta : \Delta, a).$$

If $M = y$ with $x \neq y$ the result is immediate by definition. If $M = \lambda x.M'$ the result is a direct consequence of the IH. If $M = PQ$ then $\varphi = (\varphi_0 \langle \varphi_1, \dots, \varphi_l \rangle) \circ (\theta \oplus \langle g \rangle)$ and $\varphi[\vec{psi}/\vec{x}] =$

$$((\varphi_0([\vec{g}]\vec{\psi})_0/\vec{y}_0) \langle \varphi_0([\vec{g}]\vec{\psi})_1/\vec{y}_1, \dots, \varphi_l([\vec{g}]\vec{\psi})_l/\vec{y}_l \rangle) \circ \theta^* \circ \eta$$

then we apply the IH and conclude by definition of rigid expansion of an application. \square

B.4. Linear Case

Lemma B.13. *Let $\pi \in R_1$. Then there exists $\pi' \in R_2$ such that $\pi' \sim \pi$.*

Proof. The variable and abstraction cases are immediate. We prove the application case. We exploit Remark 3.3.4. Let $\pi =$

$$\frac{\zeta_0 : \eta_0 : \Gamma_0 \vdash p : \langle a_1, \dots, a_k \rangle \Rightarrow a \quad \left(\begin{array}{c} \pi_i \\ \vdots \\ \zeta_i : \eta_i : \Gamma_i \vdash q_i : a_i \end{array} \right)_{i=1}^k \quad \eta : \zeta : \Delta \rightarrow \bigotimes_{j=0}^k \zeta_j : \bigotimes_{j=0}^k \Gamma_j}{\zeta : (\bigotimes_{j=0}^k \eta_j) \circ \eta : \Delta \vdash (p \langle q_1, \dots, q_k \rangle)^\eta : a}$$

We know that $\eta = (\bigotimes_{j=0}^k \theta_j) \circ \text{sm}(\eta)^*$, where $\theta_j : \Gamma'_j \rightarrow \text{trg}(\eta_j)$, $\text{sm}(\eta)^* : \Delta \rightarrow (\Delta \{\text{sm}(\eta)\}) = \bigotimes_{j=0}^k \Gamma'_j$. We fix some repetitions-free variable tuples ζ'_j for $0 \leq j \leq k$ such that $\zeta'_j : \Gamma'_j \in (SD^{\text{len}(\zeta)})^*$ with $\zeta \{\text{sm}(\eta)\} = \bigotimes_{j=0}^k \zeta'_j$. Again, we fix some repetitions-free variable tuples ζ''_j for $0 \leq j \leq k$ such that $\zeta''_j : \text{trg}(\eta_j) \in (SD^{\text{len}(\zeta)})^*$. Then consider $\pi' =$

$$\frac{\zeta'_0 : \theta_0 : \Gamma'_0 \vdash (p^{\zeta'_0})^{\theta'_0} : \langle a_1, \dots, a_k \rangle \Rightarrow a \quad \left(\begin{array}{c} (\pi_i/\eta_i^{\zeta''_i})\{\theta_i^{\zeta'_i}\} \\ \vdots \\ \zeta'_i : \theta_i : \Gamma'_i \vdash (q_i^{\zeta''_i})^{\theta'_i} : a_i \end{array} \right)_{i=1}^k \quad \text{sm}(\eta) : \zeta : \Delta \rightarrow \bigotimes_{j=0}^k \zeta'_j : \bigotimes_{j=0}^k \Gamma'_j}{\zeta : (\bigotimes_{j=0}^k \theta_j) \circ \text{sm}(\eta)^* : \Delta \vdash ((p^{\zeta'_0})^{\theta'_0} \langle (q_1^{\zeta''_1})^{\theta'_1}, \dots, (q_k^{\zeta''_k})^{\theta'_k} \rangle)^{\text{sm}(\eta)^*} : a}$$

and it is easy to see that $\pi' \sim CF(\pi) \bigotimes \zeta''_j$. Then by Lemma 17 we can conclude. \square