Universitat Autònoma de Barcelona
Departament de Filologia Catalana

Doctorat en
ciència cognitiva i llenguatge

Tesi Doctoral

Theory of Discontinuous Lambek Calculus

Autor:
Oriol Valentín Gallart

Director de tesi:
Dr. Glyn Morrill

Tutora:
Dra M. Teresa Espinal

2012

Contents

Agraïments

Estic infinitament agraït a Glyn Morrill. Ell m'ha ajudat a endinsar-me en una disciplina apassionant com és la Gramàtica Lògica de Tipus, o en anglès Type Logical Grammar (TLG). He tingut a més a més l'oportunitat de treballar en seminaris, i molts articles en conjunt, on he intentat posar tot el millor de mi. Li estic també molt agraït per la seva humanitat que ha demostrat tantes vegades en hores baixes de la meva vida com ara quan va morir el meu germà Luis Valentín.

D'altra banda, també voldria agrair molt a M. Teresa Espinal el fet d'haver-me donat l'oportunitat de matricular-me al doctorat de ciències cognitives i llenguatge. També m'ha ajudat en moments en què no anava enlloc i no feia les tasques acadèmiques en el seu corresponent moment.

Hi ha unes quantes persones que m'han ajudat en TLG a més a més de Glyn Morrill: Michael Moortgat, Wojciech Buszkowski, Philippe de Groote, Mario Fadda i Claudia Casadio.

A nivell laboral, agraeixo molt a Toni Badia pel fet d'haver-me integrat en un grup de recerca en processament de llenguatge natural (a la UPF i, més tard, a Barcelona Media) des del 2001 fins al 2011. No oblido els meus col·legues Stefan Bott, Martí Quixal, Gemma Boleda i Judith Domingo.

A nivell personal, dec molt a la meva estimada companya Nuria Iranzo, a qui dedico aquesta tesi. Els meus pares Luis Valentín i Maria Teresa Gallart m'han donat un suport no quantificable. Gràcies! Montserrat Consul i Montserrat Arcas, les Montses, m'han sabut entendre i ajudar molt durant tot aquest any de redacció d'aquesta tesi. Finalment, Luis, parapoeta, te añoro.

List of Figures

Chapter 1

Type Logical Grammar and the Quest for Discontinuity

The point of departure of this thesis is Lambek's seminal paper 'The Mathematics of Sentence Structure' published in 1958 by the mathematician Joachim Lambek (Lambek (1958)) in which he presented the celebrated 'Syntactic Calculus', which we write **L**. Lambek was inspired by Ajdukiewicz (Ajdukiewicz (1935)) and Bar-Hillel (Bar-Hillel (1953)). The syntactic calculus was formulated by an axiomatic presentation of a calculus based on the concept of *residuation*. In order to prove the decidability of the calculus he formulated it also as a Gentzen sequent system. In this way he was able to prove the Cut elimination theorem which was the key to show the decidability.

We can say that Lambek formulated a system which could essentially capture the logic of string concatenation. **L** can be considered as the first formulation of a substructural logic, namely the intuitionistic multiplicative fragment of non-commutative Linear logic (Girard (1987)). Of course, **L** is a very rudimentary logic because it has no additive conjunction and disjunction. Moreover, due to the fact that **L** is intuitionistic, the multiplicative disjunction of linear logic (the so-called *par* connective $\invamp$) is not present in **L**. Nevertheless, from a linguistic point of view **L** turned out to be quite expressive and intuitive as we shall see.

But the logic of concatenation is a priori limited to express mismatches between functors and dependents. In fact, it seems that a pervasive characteristic of natural languages is that functors/dependents are very frequently not adjacent (see Bresnan (2001)). This phenomenon can be named the *problem of discontinuity* of natural languages. Even the so-called *configurational* languages (like for example English) that at a first sight could be considered essentially continuous, have many phenomena which contradict the apparent continuity. **L** seems not to be able to account for the syntax and semantics of, for example in English:

(1)
- Discontinuous idioms (*Mary gave the man the cold shoulder*).
- Quantification (John gave every book to Mary; Mary thinks someone left; Everyone loves someone).
- VP ellipsis (*John slept before Mary did; John slept and Mary did too*).

- Medial extraction (*drummer that John Coltrane saw in 1960*).
- Pied-piping (mountain the painting of which by Cezanne John sold for \$10,000,000).
- Appositive relativization (*John, who jogs, sneezed*).
- Parentheticals (*Fortunately, John has perserverance; John, fortunately, has perserverance; John has, fortunately, perseverance; John has perseverance, fortunately*).
- Gapping (*John studies logic, and Charles, phonetics*).
- Comparative subdeletion (*John ate more donuts than Mary bought bagels*).
- Reflexivization (*John sent himself flowers*).

In this thesis the examples itemized in (1) will be analyzed in depth. In the literature there are compelling arguments against the context-freeness of many natural languages, like for example Shieber (1985). But Pentus (1992) proved that **L** is weakly context-free. Hence, **L** contains a serious limitation for the task of the analysis of the syntax/semantics of natural languages.

Nevertheless, **L** is a very elegant calculus which can easily describe phenomena like coordination. In the next section we see its basic properties and we address with more detail the weak points of **L** w.r.t. discontinuity.

1.1 The (Continuous) Lambek Calculus L

We take as basic type logical grammar (henceforth TLG) **L**, by which we mean the system of Lambek (1958) with type-logical semantics along the lines of van Benthem (1983) and Lambek (1988).

(2) **Definition** (*Basic Syntactical Algebra*)

A *basic syntactical algebra* is an algebra $(L, +, 0)$ of arity $(2, 0)$ which is a free monoid. I.e. L is a set, and $+$ is a binary operation on L such that for all $s_1, s_2, s_3 \in L$,

$$\begin{array}{rcll} s_1+(s_2+s_3) & = & (s_1+s_2)+s_3 & \text{associativity;} \\ 0+s = & s & = s+0 & \text{identity} \end{array}$$

furthermore, up to associativity every element of L has a unique factorization into primes (freeness).[1]

We call $+$ concatenation.

We use now the term "syntactical" in the sense in which Morrill in the 90's called "prosodic" algebra.

(3) **Definition** (*Types of* **L**)

[1] Factors of an element s are elements $s_1, \ldots, s_n$ such that $s = s_1 + \cdots + s_n$; a prime is an element which has no factorization other than into just itself.

The set $\mathcal{F}$ of *types* of **L** is defined on the basis of a set $\mathcal{P}$ of primitive or atomic basic types as follows:

$$\mathcal{F} ::= \mathcal{P} \mid \mathcal{F}\bullet\mathcal{F} \mid \mathcal{F}\backslash\mathcal{F} \mid \mathcal{F}/\mathcal{F}$$

The connective $\bullet$ is called (continuous) 'product', $\backslash$ is called 'under', and / is called 'over'.

(4) **Definition** (*Syntactical Interpretation of* **L**)

A *syntactical interpretation of* **L** is a function $[\![\cdot]\!]$ mapping each type $A \in \mathcal{F}$ into a subset of L such that:

$$\begin{array}{lcl} [\![A\backslash C]\!] & = & \{s_2 \mid \forall s_1 \in [\![A]\!], s_1{+}s_2 \in [\![C]\!]\} \\ [\![C/B]\!] & = & \{s_1 \mid \forall s_2 \in [\![B]\!], s_1{+}s_2 \in [\![C]\!]\} \\ [\![A\bullet B]\!] & = & \{s_1{+}s_2 \mid s_1 \in [\![A]\!] \ \& \ s_2 \in [\![B]\!]\} \end{array}$$

Such a formalized interpretation appears to have been first made explicit in Buszkowski (1982) (in Lambek (1958) the syntactical interpretation was implicit). Observe that the (continuous) product $\bullet$ inherits associativity from the basic syntactical algebra:

(5) $A\bullet(B\bullet C) = (A\bullet B)\bullet C$

(6) **Remark**
Although the syntactical interpretation is made in monoids we have not incorporated in the set of types $\mathcal{F}$ the product unit type I which satisfies:

$$I\bullet A = A = A\bullet I$$

Note also that $(\backslash, \bullet, /; \subseteq)$ constitutes a residuated triple, i.e.

(7) $B \subseteq A\backslash C$ iff $A\bullet B \subseteq C$ iff $A \subseteq C/B$

(8) **Definition** (*Configurations and Sequents of* **L**)

The set $\mathcal{O}$ of *configurations* of **L** is defined as follows:

$$\mathcal{O} ::= \Lambda \mid \mathcal{F}, \mathcal{O}$$

The set Σ of *sequents* of **L** is defined as follows:

$$\Sigma ::= \mathcal{O} \Rightarrow \mathcal{F}$$

$\mathcal{O}$ is called the *antecedent* configuration and $\mathcal{F}$ is called the *succedent* type.

Notice that contrarily to the standard Lambek calculus the non-empty antecedent constraint is dropped, i.e. empty antecedents in sequents are allowed.[2]

[2]Although it is known in the type logical community that the non-empty antecedent constraint rules out ungrammatical expressions like `a`+`very`+`man`, it is maybe not widely known that the possibility of sequents with empty antecedent allows that one single type assignment for relative pronouns like `which` accounts for relative sentences with or without pied-piping (see Chapter 6).

(9) **Definition** (*Syntactical Interpretation of Configurations and Validity of Sequents in* **L**)

We extend the interpretation of types to include configurations as follows:

$$\begin{array}{rcl} [\![\Lambda]\!] & = & \{0\} \\ [\![A, \Gamma]\!] & = & \{s_1{+}s_2 |\ s_1 \in [\![A]\!]\ \&\ s_2 \in [\![\Gamma]\!]\} \end{array}$$

A sequent $\Gamma \Rightarrow A$ is *valid* iff $[\![\Gamma]\!] \subseteq [\![A]\!]$ in every interpretation.

(10) **Definition** (*sequent calculus for* **L**)

The *sequent calculus* for **L** is as follows, where $\Delta(\Gamma)$ indicates a configuration Δ with a distinguished subconfiguration Γ:

$$\frac{}{A \Rightarrow A}\,id \qquad \frac{\Gamma \Rightarrow A \qquad \Delta(A) \Rightarrow B}{\Delta(\Gamma) \Rightarrow B}\,Cut$$

$$\frac{\Gamma \Rightarrow A \qquad \Delta(C) \Rightarrow D}{\Delta(\Gamma, A\backslash C) \Rightarrow D}\,\backslash L \qquad \frac{A, \Gamma \Rightarrow C}{\Gamma \Rightarrow A\backslash C}\,\backslash R$$

$$\frac{\Gamma \Rightarrow B \qquad \Delta(C) \Rightarrow D}{\Delta(C/B, \Gamma) \Rightarrow D}\,/L \qquad \frac{\Gamma, B \Rightarrow C}{\Gamma \Rightarrow C/B}\,/R$$

$$\frac{\Delta(A, B) \Rightarrow D}{\Delta(A{\bullet}B) \Rightarrow D}\,{\bullet}L \qquad \frac{\Gamma \Rightarrow A \qquad \Delta \Rightarrow B}{\Gamma, \Delta \Rightarrow A{\bullet}B}\,{\bullet}R$$

In **L**, a *theorem* is a sequent which is derivable in this calculus.

Observe that in the sequent calculus for **L**, for each connective there is a left (L) rule introducing it in the antecedent, and a right (R) rule introducing it in the succedent; the type in which this connective occurs is called *active*; the other types are called *side formulas*. The L and R rules reflect respectively sufficient conditions for use, and necessary conditions for proof, of a type so-built. The sequent calculus fully modularizes the inferential properties of connectives: it deals with a single occurrence of a single connective at a time.

(11) **Proposition** (*soundness of* **L**)

In **L**, every theorem is valid.

Proof. Straightforward induction on the length of sequent proofs. □

(12) **Theorem** (*Cut-elimination for* **L**)

In **L**, every theorem has a Cut-free sequent proof.

Proof. Lambek (1958). Where A is a type, let $d(A)$ be the number of separate occurrences of the connectives $\backslash$, $\bullet$, $/$ in A, and let $d(A_1, A_2, \ldots, A_n) = d(A_1) + d(A_2) + \cdots + d(A_n)$. The *degree* of an instance of Cut

$$\frac{\Gamma \Rightarrow A \qquad \Delta_1, A, \Delta_2 \Rightarrow B}{\Delta_1, \Gamma, \Delta_2 \Rightarrow B}$$

is defined to be $d(\Gamma)+d(\Delta_1)+d(\Delta_2)+d(A)+d(B)$. It is shown that in any Cut the premises of which have been proved without Cut, the conclusion is either identical with one of the premises, or else the Cut can be replaced by one or two Cuts of smaller degree. Therefore since no degree is negative, every theorem has a Cut-free proof. □

(13) **Corollary** (*subformula property for* **L**)

In **L**, every theorem has a sequent proof containing only its subformulas.

Proof. Every rule except Cut has the property that all the types in the premises are either in the conclusion (side formulas) or are the immediate subtypes of the active formula, and Cut itself is eliminable. □

(14) **Corollary** (*decidability of* **L**)

In **L**, it is decidable whether a sequent is a theorem.

Proof. By backward-chaining in the finite Cut-free sequent search space. □

We see now the completeness of **L** w.r.t. the implicative fragment (see Buszkowski (1982)).

(15) **Theorem** (*Completeness of the* $\mathbf{L}[\backslash,/]$ *fragment*)

If $\Delta \Rightarrow A$ is a valid **L** sequent with types ranging in the so-called implicative fragment$\mathcal{F}[\backslash,/]$ then it is **L** provable.

Proof. It is worth seeing the simple proof of this result. A notational convention: we will write $\vdash$ instead of $\vdash_{\mathbf{L}}$. Consider the basic syntactical algebra $\mathcal{A} = \langle (\mathcal{F}[\backslash,/])^*, (,), \Lambda \rangle$. Define the following valuation on the atomic types:

$$v(A) \quad := \quad \{\Delta : \vdash \Delta \Rightarrow A\}$$

We show that for every implicative type C:

$$[\![C]\!]_v = \{\Delta : \vdash \Delta \Rightarrow C\}$$

We proceed by induction on the structure of the implicative types.

- Base case. True by definition.
- Inductive case. Let us suppose that $C = B/A$:

  - [$\subseteq$]: suppose $\Delta \in [\![B/A]\!]_v$. Let Γ_A be an arbitrary configuration belonging to $[\![A]\!]_v$. By induction hypothesis (i.h.), $\vdash \Gamma_A \Rightarrow A$. By hypothesis $\Delta, \Gamma_A \in [\![B]\!]_v$. Again, by i.h. $\vdash \Delta, \Gamma_A \Rightarrow B$. Then

$$\cfrac{\cfrac{\Gamma_A \Rightarrow A \qquad \Delta, \Gamma_A \Rightarrow B}{\Delta, A \Rightarrow B}\,Cut}{\Delta \Rightarrow B/A}\,/R$$

  Hence $\Delta \in \{\Gamma : \vdash \Gamma \Rightarrow B/A\}$.

- [$\supseteq$]: Let Δ be such that $\vdash \Delta \Rightarrow B/A$. Let $\Gamma_A \in [\![A]\!]_v$. By i.h. $\Gamma_A \Rightarrow A$. Then:

$$\frac{\Delta \Rightarrow B/A \qquad \Gamma_A \Rightarrow A}{\Delta, \Gamma_A \Rightarrow B}\,/E$$

Hence $\Delta \in [\![B/A]\!]$.

□

(16) **Theorem** (*Completeness of* **L**)

In **L**, every valid sequent is a theorem.

Proof. By the reasoning of Pentus (1993), which goes via "quasimodels". □

1.1.1 Type-Logical Semantics

(17) **Definition** (*semantic types*)

The set $\mathcal{T}$ of semantic types is defined on the basis of a set δ of primitive semantic types by:

$$\mathcal{T} ::= \delta \mid \mathcal{T}\&\mathcal{T} \mid \mathcal{T} \to \mathcal{T}$$

(18) **Definition** (*semantic frame*)

A *semantic frame* is a $\mathcal{T}$-indexed family of non-empty sets $\{D_\tau\}_{\tau\in\mathcal{T}}$ such that:

$$\begin{array}{rcll} D_{\tau_1\&\tau_2} & = & D_{\tau_1} \times D_{\tau_2} & \text{cartesian product} \\ D_{\tau_1\to\tau_2} & = & D_{\tau_2}^{D_{\tau_1}} & \text{functional exponentiation} \end{array}$$

For example, we might select as basic types a type e of entities, $\delta(\text{e})$ a nonempty set of individuals, and a basic type 2 of two truth values, $\delta(2) = \{\emptyset, \{\emptyset\}\}$.

(19) **Definition** (*semantic terms*)

The sets Φ_τ of *semantic terms* of type τ for each type τ are defined on the basis of a set C_τ of constants of type τ and an denumerably infinite set V_τ of variables of type τ for each type τ as follows:

$$\begin{array}{rcl} \Phi_\tau & ::= & C_\tau \mid V_\tau \mid (\Phi_{\tau'\to\tau}\ \Phi_{\tau'}) \mid \pi_1\Phi_{\tau\&\tau'} \mid \pi_2\Phi_{\tau'\&\tau} \\ \Phi_{\tau\to\tau'} & ::= & \lambda V_\tau \Phi_{\tau'} \\ \Phi_{\tau\&\tau'} & ::= & (\Phi_\tau, \Phi_{\tau'}) \end{array}$$

(We allow ourselves to abbreviate $((\phi\ \psi)\ \chi)$ as $(\phi\ \psi\ \chi)$, etc.[3]) An occurrence of a variable x in a term is *free* iff it does not fall within any part of the term of the form $\lambda x\cdot$; otherwise it is *bound* (by the closest λx within the scope of which it falls). Each term $\phi \in \Phi_\tau$ receives a semantic value $[\phi]^g \in D_\tau$ with respect to a valuation f sending each constant in C_τ to an element in D_τ, and an assignment g sending each variable in V_τ to an element in D_τ, as follows:

[3]Likewise, $((\phi, \psi), \chi)$ would be abbreviated (ϕ, ψ, χ), etc.

(20)
$$\begin{array}{rcll}
[c]^g & = & f(c) & \text{for } c \in C_\tau \\
[x]^g & = & g(x) & \text{for } x \in V_\tau \\
[(\phi\ \psi)]^g & = & [\phi]^g([\psi]^g) & \text{functional application} \\
[\pi_1\phi]^g & = & \mathbf{fst}([\phi]^g) & \text{first projection} \\
[\pi_2\phi]^g & = & \mathbf{snd}([\phi]^g) & \text{second projection} \\
[\lambda x_\tau \phi]^g & = & D_\tau \ni d \mapsto [\phi]^{(g-\{(x,g(x))\cup\{(x,d)\}}} & \text{functional abstraction} \\
[(\phi, \psi)]^g & = & \langle [\phi]^g, [\psi]^g \rangle & \text{ordered pair formation}
\end{array}$$

The result $\phi\{\psi/x\}$ of substituting term ψ (of type τ) for variable x (of type τ) in a term ϕ is the result of replacing by ψ every free occurrence of x in ϕ. The application of the substitution is *free* iff no variable free in ψ is bound in its new location. (Manipulations can be pathological if substitution is not free.) The following laws of lambda-conversion obtain:

(21)
$$\begin{array}{rcll}
\lambda y \phi & = & \lambda x(\phi\{x/y\}) & \text{if } x \text{ is not free in } \phi \text{ and } \phi\{x/y\} \text{ is free}
\end{array}$$
α-conversion

$$\begin{array}{rcll}
(\lambda x \phi\ \psi) & = & \phi\{\psi/x\} & \text{if } \phi\{\psi/x\} \text{ is free} \\
\pi_1(\phi, \psi) & = & \phi & \\
\pi_2(\phi, \psi) & = & \psi &
\end{array}$$
β-conversion

$$\begin{array}{rcll}
\lambda x(\phi\ x) & = & \phi & \text{if } x \text{ is not free in } \phi \\
(\pi_1\phi, \pi_2\phi) & = & \phi &
\end{array}$$
η-conversion

(22) **Definition** (*semantic type map for* **L**)

The *semantic type map* for **L** is a homomorphism T from syntactic types $\mathcal{F}$ to semantic types $\mathcal{T}$ such that:

$$\begin{array}{rcl}
T(A \bullet B) & = & T(A) \& T(B) \\
T(A \backslash C) & = & T(A) \to T(C) \\
T(C/B) & = & T(B) \to T(C)
\end{array}$$

Categorial semantics, Curry-Howard type-logical semantics, works because under such a type map categorial derivations are homomorphically sent to intuitionistic proofs, i.e. pure terms of the typed lambda calculus. These compose lexical semantics expressed as terms of higher-order logic into meanings in higher-order logic of projected expressions. Montague (1970) observed that algebraically, compositionality is a homomorphism from syntax to semantics. TLG goes further in asserting that it is a homomorphism from syntactic *proofs* to semantic *proofs*.
In this sense, we see that **L** adheres to the following theoretical principles:

(23)
- Strong homomorphism from the syntactic dimension into the semantic dimension.
- Direct semantics: the different semantic readings of a linguistic expression Φ are strictly controlled by the different syntactic derivations of a provable sequent associated to Φ.
- Intuitionistic regime.

- Language models (interpretating **L** in basic syntactical algebras) constitute an important characteristic of TLG which measures the quality of a type logical system w.r.t. soundness and completeness. Kripke-like interpretations with abstract ternary relations are not considered a good measure of the logical machinery.
- **L** has a sequent system whitout structural rules.

We realize then that **L** is for our approach the starting point for further logical/linguistic evolutions. A classical criticism against **L** is its inability to project constituents like in the generative paradigm. A radically different view of **L** is to identify it with the non-associative Lambek calculus (Lambek (1961)) extended with the non-logical postulate of associativity: **NL** + **Assc**. Nevertheless, the **L** invisibility of the constituent structure can be remediated with the use of modal unary connectives (see Morrill (1994) and Chapter 6 from this thesis). Hence there is no reason to drop **L** as an initial theoretical construct for type logical investigations.

We will see in the next section that the landscape of type logical linguistic theories have a rich variety of different theoretical evolutions (or even revolutions!) from the principles formulated in (23).

1.2 Type Logical Linguistic Theories Facing the Problem of Discontinuity

In this section we formulate a general theoretical test which we think constitutes a good technical device to measure the *goodness* of a TLG theory facing the problem of discontinuity:

(24) Given a type logical theory **T** and an arbitrary linguistic expression $\Phi[\Psi]$ with a distinguished occurrence of a linguistic subexpression Ψ, in **T** there is a derivation such that:

$$\Phi[\Psi] \to \Phi[] \widehat{\circ} \Psi$$

It is illustrative to see how the principle (24) beahaves in **L**. The way a type logical linguistic theory accounts for extraction in relatives is a good instance of test (24). Consider the following contrast:

(25) a. Man that Peter saw
$\lambda x[(man\ x) \wedge (saw\ x\ peter)]$

b. Man that Peter saw today
$\lambda x[(man\ x) \wedge (today\ (saw\ x\ peter))]$

Let us see the standard **L** type assignments:

(26)
`man` : $CN : man$
`that` : $(CN\backslash CN)/(S/N) : \lambda P.\lambda Q.\lambda z((Q\ z) \wedge (P\ z))$
`loves` : $(N\backslash S)/N : love$
`Peter` : $N : peter$

In (25), given the standard lexicon (26), a straightforward derivation in **L** accounts for the grammaticality of a) and its reading. Nevertheless, the grammatical common noun in (25.b) is not derivable in **L**. We can build an **L** model $(\mathcal{A}, v)$ which falsifies (25.b).
Let $\mathcal{A} = \langle \{d, m, p, s, t\}^*, (,), \Lambda \rangle$ be a basic syntactical algebra and let v be a valuation which is defined on atomic types as follows:

$$\begin{array}{lcl} v(N) & := & \{p\} \\ v(CN) & := & \{m + (t + p + s)^n : n \in \omega\} \\ v(S) & := & \{p + s + p + d^n : n \in \omega\} \end{array}$$

From the interpretation on the atomic types we can then compute the interpretations of the lexical types:

$$\begin{array}{lcl} [\![(CN\backslash CN)/(S/N)]\!] & := & \{t\} \\ [\![(N\backslash S)/N]\!] & := & \{s\} \\ [\![S\backslash S]\!] & := & \{d^n : n \in \omega\} \end{array}$$

Since $m + t + p + s + d \in [\![CN, (CN\backslash CN)/(S/N), N, (N\backslash S)/N, S\backslash S]\!]_v$, but $m + t + p + s + d \notin \{m + (t + p + s)^n : n \in \omega\} = [\![CN]\!]_v$, it follows that:

$$\not\models CN, (CN\backslash CN)/(S/N), N, (N\backslash S)/N, S\backslash S \Rightarrow CN$$

The type of the relative pronoun of our examples has as a subtype S/N which simply means that it must combine with a linguistic expression which is a prefix of a sentence, i.e. a context. But, crucially here the combination with the context inside the relative sentence is blocked because it is no longer peripheral due to the presence of the adverbial `today`.

In recent years (2000-2012), scholars have developed a variety of type logical theories which diverge from the ones that were studied in the 90's like the widely kown influential works of Morrill (1994) and Moortgat (1997). The former author, who can be considered the scholar who invented the term *type logical grammar*, assumed that in principle any type logical theory should have **L** as a basic underlying theory. The latter author instead considered the non-associative Lambek calculus (Lambek (1961)) **NL** as the basic theory for further theoretical developments like multimodal categorial grammar (henceforth MMCG). Morrill's and Moortgat's theories adhered to the principles itemized in (23). In the case of Moortgat's work (*op. cit.*) Krike frames (with ternary relations) were considered instead of language models.

The type logical theories from the period 2000 to 2012 are not easy to classify because they diverge and share properties in many aspects. It seems then convenient to give some classifying theoretical axes in order to grasp the new type logical landscape:

(27)
1) Intuitionistic regime versus classical regime.
2) Semantic readings controlled in some way by syntax versus what we could call the *Curry approach* in which there is an abstract syntax which is then mapped into diverse linguistic dimensions: the syntactic dimension, the semantic dimension, etc.

3) Direct semantics versus the so-called continuation semantics or the use of extensions of the lambda calculus (like for example the $\lambda\mu\tilde{\mu}$-calculus; Herbelin (2005); Curien and Herbelin (2000)) which are not confluent calculi.
4) Complete use of a (substructural) logic versus the controlled or limited use of the rules of for example a sequent calculus.

Notice that every item of (27) contains the word *versus*, meaning that the item is *polarized*. For instance, if a theory **T** is such that it does adhere to direct semantic we will say that **T** adheres to the third item 3) from (27), i.e. $3)^{+}$. Otherwise we say **T** satifies $3)^{-}$. In case that a theory **T** is such it has variants that adhere respectively to say $2)^{+}$ and $2)^{-}$, we will write $2)^{+,-}$. We will also write $+, -$ in case that the theory **T** does not adhere exactly to any feature of the points itemized in (27). Let us see then some representative theories which have arisen in the last 12 years (for an excellent survey see Lecomte (2011)):[4]

- Symmetric categorial grammar Moortgat (2009): $1)^{-}$, $2)^{+}$, $3)^{-}$, $4)^{+}$.
- Abstract categorial grammar De Groote (2001): $1)^{+}$, $2)^{-}$, $3)^{+}$, $4)^{+}$.
- Categorial grammar for minimalism Lecomte (2005): $1)^{+}$, $2)^{+,-}$, $3)^{+,-}$, $4)^{-}$.
- Convergent grammars Pollard (2007): $1)^{+}$, $2)^{-}$, $3)^{+}$, $4)^{-}$.
- Linear grammars Anoun and Lecomte (2006): $1)^{+}$, $2)^{-}$, $3)^{+}$, $4)^{-}$.
- Barker's continuation semantics program Barker (2002): $1)^{+}$, $2)^{+,-}$, $3)^{-}$, $4)^{+}$.

According to (27), the approach of this thesis would be: $1)^{+}$, $2)^{+}$, $3)^{+}$, $4)^{+}$. The discontinuous Lambek calculus has deep precedents in the 90's, which in the next subsection are summarized

1.2.1 Precedents of the Discontinuous Lambek Calculus in the 90's: the wrapping approach

The idea of discontinuity operators for categorial grammar appears to originate in Bach (1981) and Bach (1984). Where $s = a_1 + \cdots + a_n$ is the factorization of s into primes, let us define:

(28)
$$\begin{array}{rcl} FIRST(s) & =_{df} & a_1 \\ RREST(s) & =_{df} & a_2 + \cdots + a_n \\ LAST(s) & =_{df} & a_n \\ LREST(s) & =_{df} & a_1 + \cdots + a_{n-1} \end{array}$$

Bach (1984) defined the operations $RWRAP$ and $LWRAP$, and their converses $LINFIX$ and $RINFIX$ respectively, as follows:

(29)
$$\begin{array}{rcccl} RWRAP(s_1, s_2) & = & LINFIX(s_2, s_1) & = & FIRST(s_1) + s_2 + RREST(s_1) \\ LWRAP(s_1, s_2) & = & RINFIX(s_2, s_1) & = & LREST(s_1) + s_2 + LAST(s_1) \end{array}$$

[4] Lecomte (2011) is detailed and exhaustive. In case of doubts we refer the reader to this book.

Bach had in mind such applications as a characterization of the object equi *persuade*/subject equi *promise* distinction in terms of alternative argument order, but here we would assume a coding of control properties in lexical semantics. He also proposed 'long-distance' functors in relation to Dutch word order, which we will address, but in terms of wrapping.

The first type-logical formulation of discontinuity, i.e. with an interpretation of types and with a sequent calculus, appeared in Moortgat (1988). Moortgat defined discontinuous types as follows (we modify his notation):

(30) $\mathcal{F} \quad ::= \quad \mathcal{F} \downarrow_\forall \mathcal{F} \mid \mathcal{F} \uparrow_\exists \mathcal{F}$

(31)
$$\begin{array}{lcl} [\![A \downarrow_\forall C]\!] & = & \{s \mid \forall s_1{+}s_2 \in [\![A]\!], s_1{+}s{+}s_2 \in [\![C]\!]\} \\ [\![C \uparrow_\exists B]\!] & = & \{s \mid \exists s_1, s_2, s = s_1{+}s_2 \;\&\; \forall s' \in [\![B]\!], s_1{+}s'{+}s_2 \in [\![C]\!]\} \end{array}$$

The following sequent rules were given:

(32)
$$\frac{\Gamma \Rightarrow A \qquad \Delta(C) \Rightarrow D}{\Delta(\Gamma(A \downarrow_\forall C)) \Rightarrow D} \downarrow_\forall L \qquad \frac{\Gamma, B, \Delta \Rightarrow C}{\Gamma, \Delta \Rightarrow C \uparrow_\exists B} \uparrow_\exists R$$

Thus e.g. medial extraction, not otherwise derivable in the Lambek calculus, is obtained from a relative pronoun type R/(S$\uparrow_\exists$N). And S(neg)$\downarrow_\forall$S(pos) would be the type of a freely floating negation particle, if there were really such an element. However, the other sequent rules cannot be formulated, so the logic is incomplete.[5]

Moortgat (1991)[6] defined a three-place in-situ binder type-constructor Q for e.g. quantifier phrases, Q(S, N, S), and subject-oriented reflexives, Q(N\S, N, N\S). The left sequent rule is:

(33)
$$\frac{\Gamma(A) \Rightarrow B \qquad \Delta(C) \Rightarrow D}{\Delta(\Gamma(Q(B, A, C))) \Rightarrow D} QL$$

However the best that can be managed on the right is:

(34)
$$\frac{\Gamma \Rightarrow A}{\Gamma \Rightarrow Q(B, A, B)} QR$$

This is insufficient to derive e.g. Q(S, N, S) $\Rightarrow$ Q(N\S, N, N\S) (that a quantifier phrase can occur in a verb phrase conjunct, H. Hendriks, p.c.) so the logic is incomplete again. Moortgat (1991) indicated that $Q(B, A, C)$ might be decomposed into something like $(B{\uparrow_\exists}A){\downarrow_\forall}C$, but he did not have a calculus ensuring that the two points of discontinuity would be one and the same, as is required in order to ensure, for example, that a quantifier phrase only binds the position it occupies.[7]

[5] We resolve this by decomposing Moortgat's connectives into ones for which both rules of proof and use can be given, as follows: $A{\downarrow_\forall}C = \check{~}A{\downarrow}C$ and $C{\uparrow_\exists}B = \hat{~}(C{\uparrow}B)$.

[6] Moortgat (1991) also proposed a substring product:

(i) $[\![A{\odot_\exists}B]\!] = \{s_1{+}s_2{+}s_3 \mid s_1{+}s_3 \in [\![A]\!] \;\&\; s_2 \in [\![B]\!]\}$

(ii)
$$\frac{\Gamma_1, \Gamma_2 \Rightarrow A \qquad \Delta \Rightarrow B}{\Gamma_1, \Delta, \Gamma_2 \Rightarrow A{\odot_\exists}B} \odot_\exists R$$

But again a left rule cannot be given. We resolve this by decomposing thus: $A{\odot_\exists}B = \check{~}A{\odot}B$.

[7] We resolve this by realizing exactly the decomposition $Q(B, A, C) = (B{\uparrow}A){\downarrow}C$.

Versmissen (1991) observed that we want in some way to mark points of discontinuity. Algebraic formulations, developed without knowledge of the head grammars of Pollard (1984), were as follows:

- Solias Arís (1992): syntactical algebra $(L, +, 0, \langle\cdot,\cdot\rangle)$ where $(L, +, 0)$ is a free monoid and $(L, \langle\cdot,\cdot\rangle)$ is a free groupoid. Wrap was a partial operation defined by $\langle s_1, s_3\rangle W s_2 =_{df} s_1{+}s_2{+}s_3$.

- Morrill and Solias (1993): syntactical algebra $(L, +, 0, \langle\cdot,\cdot\rangle, 1, 2)$ where $(L, +, 0)$ is a monoid, $(L, \langle\cdot,\cdot\rangle)$ is a groupoid and $1\langle s_1, s_2\rangle = s_1, 2\langle s_1, s_2\rangle = s_2$ and $\langle 1s, 2s\rangle = s$. Wrap was a total operation defined by $sWs' =_{df} 1s{+}s'{+}2s$.

- Morrill (1994) and Morrill (1995): syntactical algebra $(L, +, 0, (\cdot,\cdot), W)$ where $(L, +, 0)$ is a monoid, $(L, (\cdot,\cdot))$ and (L, W) are groupoids and there is the structural rule of interaction $(s_1, s_3)W s_2 = s_1{+}s_2{+}s_3$. Wrap was a primitive total operation.

In Solias Arís (1992) wrapping was derived and partial. In Morrill and Solias (1993) it was derived and total. In Morrill (1994) and Morrill (1995) it was primitive and total. But in all three cases, the representation of discontinuous expressions in an (unsorted) algebra introduced syntactical terms in which points of discontinuity, because embedded, could necessarily never wrap, e.g. $s_1{+}(s_2, s_3)$, so the syntactical ontology contained much junk.

Morrill and Merenciano (1996) cleared this up admitting only n-tuples of strings sorted by their arity. But in the generalized case (i.e. with no upper bound on the number of points of discontinuity), both pairing and the empty tuple would be required for the construction of unboundedly long tuples. Here we reduce the machinery to a single operator of arity zero (i.e. a constant); cf. also Moortgat (1996) for a constant operator, but here we use it to get the generalized case of discontinuity. The nullary operator is in general internal to concatenation, but whenever it is embedded it can be considered (by virtue of the associativity of concatenation) to be immediately embedded, and as such, always useful to undergo wrap.

1.2.2 Challenges of this Thesis w.r.t. its 90's Precedents

In this thesis, the discontinuous Lambek calculus, in notation **D**, represents the natural evolution of the 90's wrapping approach. The challenges that **D** faces in this thesis are the following:

- To have a clear class of algebras in which **D** is formulated: the so-called class of *standard displacement algebras* **FreeDisp** and the class of *general displacement algebras* **Disp** (c.f. Chapter 2). These algebras are close to the so-called language models of the Lambek calculus.

- To discover an equational theory $\mathcal{E}_D$ for **Disp** in which the notion of (linguistic) context can be manipulated in order to build over it a type logical theory. The notion of extraction of a subterm in a term is studied in depth and the conditions for extractability are stated and proved. This enables handling the notion of general contexts.

- To formulate a (sorted) multimodal calculus **mD** (c.f. Chapter 3) which has precisely as structural postulates the equations of $\mathcal{E}_D$.
- To build a data-structure (see the notion of *hyperconfiguration* in Chapter 3) that allows us to have a (sorted) sequent calculus without structural rules, which in fact has absorbed the structural postulates of **mD**. We call this new sequent calculus without structural rules the *hypersequent calculus*, in notation **hD**.
- To give a faithfull embedding translation between the multimodal calculus and the hypersequent calculus.
- To give for the first time a proof of the Cut elimination theorem for the hypersequent calculus, and hence (via the referred embedding translation) for the multimodal calculus. The proof is very similar to the one Lambek (1958)) gave for the syntactic calculus.
- To prove several completeness results for the first time for the wrapping approach.
- To extend the calculus with the linear logic (Girard (1987)) additive conjunction and disjuntion. To consider new synthetic connectives which allow for example extend the discontinous lambek Calculus to a nondeterministic version. To prove the Cut elimination theorem for these extensions.
- For the first time, to study and to prove results on the (weak) generative capacity of the discontinuous Lambek calculus.

The road map for the wrapping approach for discontinuity in this thesis is divided in 7 chapters:

- This chapter.
- Chapter 2 studies in depth the model-theoretical foundations of discontinuity.
- Chapter 3 studies the different calculi: the categorical, the multimodal and the hypersequent calculi, which has no structural rules. The Cut elimination theorem is proved for the discontinuous Lambek calculus and its extensions. Several faithful embedding translations between the different calculi are proved.
- Chapter 4 studies and proves several soundness and completeness results for the discontinuous Lambek calculus.
- Chapter 5 studies the generative expressivity of the discontinuous Lambek calculus. Several interesting results are proved.
- In Chapter 6, we show the new calculi at work: linguistic applications. The discontinuous phenomena of (1) are accounted for. Moreover a detailed study of anaphora in English is carried out. The case of some subtle Romance reflexive binding properties are studied in depth.
- Finally, in Chapter 7 we find the conclusions and identify further studies related to the discontinuous Lambek calculus.

1.2.3 Contributions of this Thesis

This thesis presents new material as well as the work of several years of this author on the discontinuous Lambek calculus, as co-author with Glyn Morrill and Mario Fadda. Several papers on the subject are listed:

- Morrill, Fadda, and Valentín (2007). In this paper the nondeterministic discontinuous Lambek calculus is formulated with some linguistic applications. The joint development (of the three authors) of a new sequent syntax (main feature of the paper), which is called hypersequent calculus, was formulated. This new sequent syntax corresponds to the segmented (or string-based) hypersequent calculus (cf. Chapter 3). The calculus allows an unbounded number of points of discontinuity.
- Morrill, Valentín, and Fadda (2009). This paper contains a detailed type logical account of processing issues in Dutch, namely the so-called cross-serial dependencies. The calculus which is used in this analysis is a version of the 1-**DLC** based on Valentín (2006). The main contribution of the author in this paper is the formulation of unary synthetic connectives (cf. Girard (2006)) and their hypersequent rules. These synthetic connectives allows the authors to give an elegant linguistic account of a fragment of Dutch. The 1-**DLC** is a proper subsystem of the hypersequent calculus (cf. Chapter 3 and Chapter 6).

In the Spring of 2010 the author had the idea of a new sequent syntax, the so-called *tree-based* hypersequent syntax. This idea concluded satisfactorily a whole series of considerations on categorial discontinuity. It is essential to all the following articles, which it enabled and precipitated.

- Morrill and Valentín (2010a). This paper contains the first published formulation of the tree-based hypersequent calculus (cf. Chapter 3). This calculus has an unbounded number of connectives but it does not contain the nondeterministic connectives. The main contribution of the author was the proof à la Lambek (1958) of the Cut elimination theorem for the tree-based hypersequent calculus.
- Morrill and Valentín (2010c). In this paper the principles A, B and C of Binding theory are accounted for with the discontinuous Lambek calculus enriched with the modality $S4$, Jäger's logic (cf. Jäger (2005)) of limited contraction, and finally the *negation as failure*. The main contribution of the author was a series of theoretical considerations on the so-called negation as failure. Computer generated examples of binding theory are given. Interestingly, there is no use of features like *nominative* or *accusative* because English is a configurational language.
- Morrill and Valentín (2010b). Here a variant of the discontinuous Lambek calculus is proposed. It is called the edge displacement logic. This variant has two characteristic features: it allows an unbounded number of points of discontinuity but the number of connectives is finite (compare with the discontinuous Lambek calculus presented in this thesis and in Morrill and Valentín (2010a)). The main contribution of the author was the formulation of binary synthetic connectives and the discussion of the spurious

ambiguity in the hypersequent calculus. These synthetic connectives were defined with the so-called additive conjunction and disjunction of Linear Logic (cf. Girard (1987)).

- Morrill and Valentín (2010d). This paper contains some interesting results of the (weak) generative capacity of the discontinuous Lambek calculus. The main contribution of the author was the formulation and proof of the fronting lemma and the theorem which states the recognition of the permutation closure of context-free languages.

- Morrill and Valentín (2011). In this paper the main contribution of the author was the Cut elimination admissibility of the extended logic of Morrill and Valentín (2010c): it was proved that a significative part of the calculus (the whole system except the negation as failure) enjoys Cut elimination. The problem of introducing a negation in type logical grammar is addressed. Finally an exhaustive account of anaphora (cataphora phenomena are studied) ends the paper.

- Morrill, Valentín, and Fadda (2011). This paper formulates the edge displacement calculus introduced before. Several linguistic applications are given and the Cut elimination theorem is proved. The main contribution of the author was a proof of the Cut elimination theorem.

Chapter 2

Model Theoretical Foundations for Discontinuity

The principal purpose of this chapter is to set out the model-theoretical foundations (mainly from universal algebra) underlying our approach to discontinuity in natural language. The point of departure is the class of free monoids. From free monoids we define the class of ω-sorted (standard) *displacement algebras* **FreeDisp**, which we claim is an appropriate class of sorted algebras modelling the segmented strings we defined in the introduction of this thesis. We propose an equational theory denoted $\mathbf{Eq_D}$[1] which axiomatizes the class of *general displacement algebras*, in notation **Disp**, which properly contains **FreeDisp**. Interestingly, the equations which hold of the class of standard displacement algebras are derivable in the equational theory $\mathbf{Eq_D}$. In other words, $\mathbf{Eq_D}$ is proved to be sound and complete w.r.t. **FreeDisp**.

In the introduction of this work we noted the relevance of having formal tools to deal with what we called contexts for linguistic subexpressions in linguistic expressions. As we saw in our type-logical agenda for discontinuity, formal tools to work with contexts (of linguistic expressions) have been revealed to be fundamental for the analysis of e.g. quantifier expressions and the problem of quantifier scope, binding, negative polarities and so on. The equational theory $\mathbf{Eq_D}$ we present in this chapter gives a way to express the metalogical notion of context at an object level. This ability of $\mathbf{Eq_D}$ will be called *visibility for extraction* and some related results will be stated and proved. The main properties of $\mathbf{Eq_D}$ will be used extensively in the next two chapters.

[1] In fact we propose two equational theories, namely $\mathbf{Eq_D}$ and $\mathbf{Eq_{D2}}$.

2.1 Some technical preliminaries

A *signature* Σ is a graduated set, i.e., a set $\Sigma = \bigcup_{n \in \omega} \Sigma_n$. Each set Σ_n is considered the set of n-ary functions. In particular, Σ_0 is the set of constants. Every signature Σ has associated a function $ar : \Sigma \longrightarrow \omega$, called the arity function. Given $f \in \Sigma_n$, we put $ar(f) = n$. Given $n, m \in \omega$ with $n \neq m$, Σ_n and Σ_m are of course disjoint. If Σ is finite then it is called a *ranked alphabet.*

Let X be a (finite or infinite) set such that Σ and X are disjoint. Elements of X are called *variables.* We define the set of formal terms $T_\Sigma[X]$ with signature Σ. Elements of $T_\Sigma[X]$ are called Σ-*terms.* $T_\Sigma[X]$ is the least set such that:

(35) i) $\Sigma_0 \subseteq T_\Sigma[X]$

ii) $X \subseteq T_\Sigma[X]$

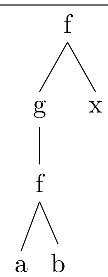
iii) If $f \in \Sigma_n$ with $n > 0$ and $t_1, \cdots, t_n \in T_\Sigma[X]$ then $f(t_1, \cdots, t_n) \in T_\Sigma[X]$.

This impredicative definition can be *generated* using the standard Backus Naur Form (henceforth BNF) recursive definitions:

(36) $T_\Sigma[X] ::= X \mid \Sigma_0 \mid \Sigma_n(\underbrace{T_\Sigma[X], \cdots, T_\Sigma[X]}_{n})$ for $n > 0$

(37) **Example**
Let $\Sigma = \Sigma_0 \cup \Sigma_1 \cup \Sigma_2$ where $\Sigma_0 = \{a, b\}$, $\Sigma_1 = \{g\}$ and $\Sigma_2 = \{f\}$. Here, x denotes an arbitrary variable. Σ-terms can be depicted graphically as labelled trees:

(38)

Textual representation of the term	Tree representation of the term
$g(a)$	
$f(g(f(a,b)),x)$	

(39) **Definition** (Σ-*Algebras*)

A Σ-Algebra $\mathcal{A} = \langle A, (f^{\mathcal{A}})_{f \in \Sigma} \rangle$ comprises a set (the universe) and the collection of the interpreted constants and functions of Σ in A, i.e:

- If $ar(f) = 0$ then $f^{\mathcal{A}} \in A$.
- If $ar(f) = n > 0$ then $f^{\mathcal{A}} = A^n \longrightarrow A$.

A Σ-homomorphism or simply a homomorphism is a function $\alpha : \mathcal{A} \to \mathcal{B}$ between two Σ-algebras $\mathcal{A}$ and $\mathcal{B}$ such that for every $n > 0$ $\alpha(f^{\mathcal{A}}(a_1, \cdots, a_n)) = f^{\mathcal{B}}(\alpha(a_1), \cdots, \alpha(a_n))$, and if $n = 0$ $\alpha(f^{\mathcal{A}}) = f^{\mathcal{B}}$. A Σ-monomorphism is an injective Σ-homomorphism. A Σ-epimorphism is an exhaustive Σ-homomorphism. Finally a Σ-isomorphism is a Σ-homomorphism which is bijective.

In this chapter and the subsequent chapters we will need the notion of *sorted* signatures. Sorted signatures are more fine-grained than standard signatures and they constitute a useful way to introduce partiality. A *sorted signature* is a pair $(\mathbf{S}, \Sigma)$ where $\mathbf{S}$ is called the set of *sorts* and Σ is, like in the case of unsorted signatures, a graduated set $\Sigma = \bigcup_{n \in \omega} \Sigma_n$. For $n > 0$ every Σ_n is called the set of *n-ary functions* and if $n = 0$ then Σ_n is called the set of *constants*. Every sorted signature is associated a function $\Omega : \Sigma \longrightarrow \mathbf{S}^+$, where $\mathbf{S}^+$ is the set of non-empty finite lists of sort elements. Ω is such that for every $f \in \Sigma_0$, if $\Omega(f) = s$ (i.e. $\Omega(f)$ is a list containing a single sort element) then s is the sort of the constant f, and if $\Omega(f) = \langle s_1, \cdots, s_n, s_{n+1} \rangle$, f is a functional symbol which must be thought of as a typed function as follows:

(40) $f : s_1, \cdots, s_n \longrightarrow s_{n+1}$

We say that f is of *sort functionality* $s_1, \cdots, s_n \longrightarrow s_{n+1}$. Intuitively f is a function of arity $n > 0$ which needs n arguments of the appropriate sorts s_i's giving a value of sort s_{n+1}.

Let X be an $\mathbf{S}$-graduated set $X = \bigcup_{s \in \mathbf{S}} X_s$. We suppose that X and Σ are disjoint. Intuitively X is the set of $\mathbf{S}$-sorted variables. Like in the unsorted case, we can define the set of formal terms $T_\Sigma[X]$ with signature $(\mathbf{S}, \Sigma)$. Elements of $T_\Sigma[X]$ are called $(\mathbf{S}, \Sigma)$-*terms*, or simply Σ-terms. $T_\Sigma[X]$ is the least set such that:

(41) i) $\Sigma_0 \subseteq T_\Sigma[X]$

ii) $X \subseteq T_\Sigma[X]$.

iii) If $f \in \Sigma_n$ with $n > 0$ and sort functionality $s_1, \cdots, s_n \to s_{n+1}$, and t_i are of sort s_i $(i = 1, \cdots, n)$, then $f(t_1, \cdots, t_n) \in T_\Sigma[X]$. Of course, $\Omega(f(t_1, \cdots, t_n)) = s_{n+1}$

$T_\Sigma[X]$ can be thought of as a graduated set, i.e. $T_\Sigma[X] = \bigcup_{s \in \mathbf{S}} (T_\Sigma[X])_s$. Each set $(T_\Sigma[X])_s$ is the set of Σ-terms of sort s. We say that $s_1, \cdots, s_n \to s$ is a *sort functionality* of Σ if there exist at least one $f \in \Sigma$ such that $\Omega(f) = s_1, \cdots, s_n \to s$. We can give a BNF definition of $T_\Sigma[X]$:

(42) **Definition** (*Sorted Σ-Algebras*)

A sorted $(\mathbf{S}, \Sigma)$ algebra is the following tuple $\mathcal{A} = \langle \{L_s\}_{s \in \mathbf{S}}, \{f_L^{\mathcal{A}}\}_{L = \Omega(f)} \rangle$. The $\mathbf{S}$-indexed set $(L_s)_{s \in \mathbf{S}}$ is called the set of *sort domains*. The interpretations of elements of Σ are such that:

- If $n = 0$ then $f^{\mathcal{A}} \in L_s$ where $s = \Omega(f)$.
- If $n > 0$, where $\Omega(f) = (s_1, \cdots, s_n, s_{n+1})$, then $f^{\mathcal{A}}$ is an interpreted function as follows:

  (43) $f^{\mathcal{A}} : L_{s_1} \times \ldots \times L_{s_n} \longrightarrow L_{s_{n+1}}$

We present two examples of sorted Σ-algebras:

(44) **Example**
An example from computer science, taken from Lalement (1990): consider the *set* data-structure. Lalement proposes a set of sorts with only two elements, $\mathbf{S} := \{\mathbf{element}, \mathbf{set}\}$. The signature is $\Sigma = \{\emptyset, \#\}$, where $\emptyset$ is a constant of sort **set** and $\#$ is a function of sort functionality $\Omega(\#) = \mathbf{element}, \mathbf{set} \to \mathbf{set}$. The axioms proposed for the *set* data-structure are:

- $a\#(a\#x) \approx a\#x$
- $a\#(b\#x) \approx b\#(a\#x)$
  Where a and b are of sort **element**, and x is of sort **set**. $\#$ must be interpreted as the function of *insertion* of an element in a set.

(45) **Example**
Consider the class $\mathbf{R_L}$-**Mod** of (not necessarily unital) left R-modules where R denotes an arbitrary ring. Giving an equational axiomatization for $\mathbf{R_L}$-**Mod** in standard unsorted universal algebra is complex. Scalars must be simulated with the help of R-indexed unary functions $(f_r)_{r\in A}$. But in sorted universal algebra the axiomatization is very clear and easy. Consider the sorts **scalar** and **vector** and the sorted $\{\mathbf{vector}, \mathbf{scalar}\}$-signature:

(46) $\Sigma = \{0_{vector}, +_{vector}, -_{vector}, 0_{scalar}, +_{scalar}, -_{scalar}, \cdot_{scalar}, \cdot_{scalar\times vector}\}$

Sort functionalities of Σ and the axiomatization of $\mathbf{R_L}$-**Mod** are:[2]

$$\begin{array}{lll}
0_{vector} : \mathbf{vector} & & \\
0_{scalar} : \mathbf{scalar} & & \\
+_{vector} : \mathbf{vector} \times \mathbf{vector} & \longrightarrow & \mathbf{vector} \\
\quad (v, w) & \mapsto & v +_{vector} w \\
-_{vector} : \mathbf{vector} & \longrightarrow & \mathbf{vector} \\
\quad v & \mapsto & -_{vector} v \\
+_{scalar} : \mathbf{scalar} \times \mathbf{scalar} & \longrightarrow & \mathbf{scalar} \\
\quad (a, b) & \mapsto & a +_{scalar} b \\
-_{scalar} : \mathbf{scalar} & \longrightarrow & \mathbf{scalar} \\
\quad r & \mapsto & -_{scalar} r \\
\cdot_{scalar} : \mathbf{scalar} \times \mathbf{scalar} & \longrightarrow & \mathbf{scalar} \\
\quad (r, s) & \mapsto & r \cdot_{scalar} s \\
\cdot_{scalar\times vector} : \mathbf{scalar} \times \mathbf{vector} & \longrightarrow & \mathbf{scalar} \\
\quad (r, v) & \mapsto & r \cdot_{scalar\times vector} v
\end{array}$$

$\langle \mathrm{L}_{\mathbf{vector}}, +_{\mathrm{vector}}, -_{\mathrm{vector}}, 0_{\mathrm{vector}} \rangle$ is an abelian group

$\langle \mathrm{L}_{\mathbf{scalar}}, +_{\mathrm{scalar}}, \cdot_{\mathrm{scalar}}, -_{\mathrm{scalar}}, 0_{\mathrm{scalar}} \rangle$ is a ring

$r(v +_{vector} w) \approx rv +_{vector} rw$
$(r +_{scalar} s)v \approx rv +_{vector} sw$
$r(sv) \approx (r \cdot_{scalar} s)v$

[2]Usually, $r \cdot_{scalar\times vector} v$ is denoted rv omitting $\cdot_{scalar\times vector}$, where r is of scalar sort and v is of vector sort.

Note that in the previous example, the operations could have been presented as sort-polymorphic.

2.1.1 Notion of occurrence of a subterm in an unsorted or a sorted Σ-term

In what follows we describe the notion of context for an unsorted or sorted term. The definition we give is almost the same for both unsorted and sorted cases. Let s be a term (unsorted or sorted) in which occurs a subterm r. In the tree representation of s we can see what is a occurrence of a subterm in a term: an address in the tree representation of the term s.[3]

A simple way to deal with contexts is considering the use of linear occurrences of variables. Let t be a (sorted or unsorted) term with a linear occurrence of the variable x. We define $t[r]$ as follows:

(47) $t[r] \stackrel{def}{=} \sigma_{r/x}(t)$

Where $\sigma_{r/x}$ is the term substitution map which replaces x by r. If the term t is sorted then of course x is also sorted, and therefore r is required to have the same sort of x, i.e. $\Omega(x) = \Omega(r)$, where Ω is the associated map to an arbitrary sorted signature $(\mathbf{S}, \Sigma)$. It is interesting to signal that the following holds:

$$t[x] = t, \text{ for } t[x] = \sigma_{x/x}(t) = t$$

The bracket notation $t[r]$ which signals the distinguished occurrence of the subterm r in s presupposes that t has a linear occurrence of a variable which is replaced by r. Usually, when proving results which use contexts, without loss of generality we assume that x's occurrence in $t[x]$ is linear. The notion of context we have defined is very simple and works correctly for the unsorted/sorted case.

As will shall see in section (2.3), we will define what we call *visibility for extraction* of a term. As we have seen, contexts need a metalogical point of view. Here we will be able to render this point of view at the *object level* (see the following sections on displacement algebras (2.2) and visibility for extraction (2.3)).
More concretely we will have the following duality:

$$(48)\quad \begin{cases} t[r] & = & \sigma_{r/x}(t[x]) \text{ where } x's \text{ occurrence is linear} \\ t[x] & \approx & t' \circ_i x \text{ where } t' \text{ is a term and } \circ_i \text{ is a binary function} \end{cases}$$

As we see in (48), the equation $t[x] \approx t' \circ_i x$ gives a term t' which can be considered the context of x in $t[x]$ at the object level. Therefore, the substitution operation σ which is at a metalevel is transformed into a term level operation: $\circ_i x$:

$t[x]$	$\approx$	$\underbrace{t'}_{\text{Object level context}}$	$\underbrace{\circ_i x}_{\text{Operation at object level}}$

[3]In the literature there are different ways to make explicit the notion of occurrence of a term in a tree (or even in more complex (data) structures). A quite common way is the use of addresses through the so called *tree domains*.

This ability to deal with the context at the object level (with the help of an equational theory) will be crucial on our road to discontinuity. The next sections give the necessary machinery to work with contexts. This material will be *exported* to the type-logical setting in the next chapter. In this way, as promised in the previous chapter we build the tools for managing discontinuity.

2.2 Towards Discontinuity: Displacement Algebras and their sorted equational theory

As we said in the introduction of this chapter our point of departure on the road to discontinuity is the class of free monoids with a distinguished element different from the neutral element which we call *separator* which is in fact a *prime* (see Morrill (2002)). We need to define the concept of *prime* in a free monoid:

(49) **Definition** (*Prime*)

Let $M = (L, +, 0, 1)$ be a free monoid with a distinguished constant 1. We say that 1 is a *prime* iff 1 does not have other factors than of itself and of the 0, i.e.:

For every x, y if $1 = x{+}y$ then either $x = 1$ and $y = 0$ or $x = 0$ and $y = 1$

(50) **Definition** (*Syntactical Algebra*)

A *syntactical algebra* is a free algebra $(L, +, 0, 1)$ of arity $(2, 0, 0)$ such that $(L, +, 0)$ is a monoid and 1 is a prime. I.e. L is a set, $0 \in L$ and $+$ is a binary operation on L such that for all $s_1, s_2, s_3, s \in L$,

$$\begin{array}{rcll} s_1{+}(s_2{+}s_3) & = & (s_1{+}s_2){+}s_3 & \text{associativity} \\ 0{+}s = & s & = s{+}0 & \text{identity} \end{array}$$

The distinguished constant 1 is called a *separator*.

(51) **Definition** (*Sorts of Elements in a Syntactical Algebra*)

The sort $S(s)$ of an element s of a syntactical algebra $(L, +, 0, 1)$ is defined by the homomorphism of monoids S to the additive monoid of naturals defined thus:

$$\begin{array}{rcll} S(1) & = & 1 & \\ S(a) & = & 0 & \text{for a prime } a \neq 1 \\ S(s_1{+}s_2) & = & S(s_1) + S(s_2) & \end{array}$$

I.e. the sort of a syntactical element is simply the number of separators it contains; we require the separator 1 to be a prime and the syntactical algebra to be free in order to ensure that this induction is well-defined.

By using the concept of syntactical algebra, we now define one of the main classes of (sorted) algebras which will be used in this thesis, the class of standard *displacement algebras*, in notation **FreeDisp**. Let us consider the ω-sorted signature of displacement algebras $\Sigma_{\mathbf{D}} = (\oplus, \{\otimes_{i+1}\}_{i\in\omega}, 0, 1)$ of sort functionality $(\{i, j \longrightarrow i+j\}_{i,j\in\omega}, \{k+1, l \longrightarrow k+l\}_{k,l\in\omega}, 0, 1)$. It must be observed that the operations $\oplus$ and $\{\otimes_{i+1}\}_{i\in\omega}$ are sort-polymorphic.

The ω-sorted set of variables for the algebra of Σ_D-terms is the following:

$$\begin{aligned} X &= \bigcup_{i\,\in\,\omega} X_i, \text{ where } X_i \text{ is defined for every } i \text{ as follows:} \\ X_i &= (x_{ij})_{j\,\in\,\omega} \end{aligned}$$

As can be observed, for any $i \in \omega$ there is an infinite set of variables of sort i X_i. For any $i \in \omega$, X_i is denumerable, although it could have another cardinal. For the necessities of this thesis, the denumerable sets $X_i, i \in \omega$ are sufficient.

(52) **Definition** (*Sort Domains*)

Where $(L, +, 0, 1)$ is a syntactical algebra, the *sort domains* L_i of sort i of generalized discontinuous Lambek calculus are defined as follows:

$$L_i = \{s|S(s) = i\}, i \geq 0$$

The Σ_D-operations are defined as follows:

(53) **Definition** (*Displacement Algebra*)

The *displacement algebra* $\mathcal{A}$ with ω-sorted signature Σ_D defined by a syntactical algebra $(L, +, 0, 1)$ is the following:

$$(\{L_i\}_{i\in\omega}, +, \{\times_{k+1}\}_{k\in\omega}, 0, 1)$$

where $+$ and $\{\times_k\}_{k\in\omega}$ correspond respectively to the formal term constructors of the signature Σ_D, $\oplus$ and $(\otimes_{k+1})_{k\in\omega}$. The reader has to notice that we write $+$ and $\times_i, i > 0$ than rather the more rigorous notation $\oplus^{\mathcal{A}}$ and $\otimes_i^{\mathcal{A}}$. The constants 0 and 1 are simply written in the algebra 0 and 1 instead of $0^{\mathcal{A}}$ and $1^{\mathcal{A}}$. The interpretation of the Σ_D-terms in the displacement algebra is as follows:

operation	is such that
$+ : L_i \times L_j \to L_{i+j}$	as in the syntactical algebra
$\times_k : L_{i+1} \times L_j \to L_{i+j}$	$\times_k(s, t)$ is the result of replacing the k-th separator in s by t
$0 : L_0$	0 is the neutral element of the corresponding syntactical algebra
$1 : L_1$	1 is the prime defined in the corresponding syntactical algebra

The class of displacement algebras will be denoted **FreeDisp**. Algebras of **FreeDisp** are also called *standard* displacement algebras. This is to be contrasted with *general* displacement algebras which we will present in a few lines. An important property which holds of **FreeDisp** is the *separation property*:

$$(54)\ \begin{cases} (1)\ \forall x \in \mathrm{L}_i \text{ with } i > 0, \text{ there exists } \langle a_0, \cdots, a_i\rangle \in L_0^{i+1} \\ \text{such that } x = a_0 + 1 + a_1 + 1 + \cdots + a_{i-1} + 1 + a_i \\ \\ (2)\ \text{If } a_0 + 1 + a_1 + 1 + \cdots + a_{i-1} + 1 + a_i = \\ b_0 + 1 + b_1 + 1 + \cdots + b_{i-1} + 1 + b_i \text{ then } \forall i\ a_i = b_i \end{cases} \quad \textbf{(Separation property)}$$

The separation property will be extensively used in many results of this chapter.

(55) **Definition** (*Standard Displacement Models*)

Let $\mathcal{A} = (M, +, \{\times_{i+1}\}_{i\in\omega}, 0, 1)$ be a standard displacement algebra. Let v be a mapping, called a *valuation*, between the set of variables X to the set M; then:

$$\begin{array}{lll} 0 & \overset{v}{\mapsto} & 0 \\ 1 & \overset{v}{\mapsto} & 1 \\ x_{ij} & \overset{v}{\mapsto} & v(x_{ij}) \in L_i \text{ for every } i, j \in \omega \\ t \oplus s & \overset{v}{\mapsto} & v(t) + v(s) \\ t \otimes_i s & \overset{v}{\mapsto} & v(t) \times_i v(s) \end{array}$$

Here $x_{ij} \in X$ and $t, s \in T_{\mathbf{\Sigma_D}}[X]$. $(\mathcal{A}, v)$ is called a Σ_D-model.

As it can be seen, the mapping v is a homomorphism between Σ_D algebras, namely the (sorted) term algebra $T_{\mathbf{\Sigma_D}}[X]$ and the displacement algebra $\mathcal{A}$.

(56) **Definition** (*Displacement Model of an Equation*)

Let $(\mathcal{A}, v)$ be a displacement model:

$$(\mathcal{A}, v) \models t \approx s \text{ iff } v(t) = v(s)$$

We say that the equation $t \approx s$ is satisfied in the model $(\mathcal{A}, v)$.

(57) **Definition** (*Satisfaction of an equation in* **FreeDisp**)

Let $t, s \in T_{\mathbf{\Sigma_D}}[X]$. We say that:

FreeDisp $\models t \approx s$ iff for every $(\mathcal{A}, v)$ with $\mathcal{A} \in$ **FreeDisp**, $(\mathcal{A}, v) \models t \approx s$

We propose two ω-sorted sets of equations $\mathbf{Eq_D}$ and $\mathbf{Eq_{D2}}$ (see figures 2.2 and 2.3). We consider the extensions of $\mathbf{Eq_D}$ and $\mathbf{Eq_{D2}}$ which are the smallest congruence closed by substitutions which contain respectively $\mathbf{Eq_D}$ and $\mathbf{Eq_{D2}}$ (see Lalement (1990)). We denote these extensions with the same names, i.e. $\mathbf{Eq_D}$ and $\mathbf{Eq_{D2}}$. As we know (*op. cit.*), these types of sets of equations are called *equational theories* which are also characterized as $\mathbf{Eq}(\mathbf{C})$ for some class $\mathbf{C}$ of Σ_D-algebras, where the $\mathbf{Eq}(\cdot)$ operator applied to a class of algebras $\mathbf{C}$ denotes the set of valid equations in $\mathbf{C}$. The converse operator of $\mathbf{Eq}(\cdot)$ is $\mathbf{Mod}(\cdot)$. If $\mathcal{E}$ is a set of Σ_D-equations, then $\mathbf{Mod}(\mathcal{C})$ denotes the class of Σ_D-algebras which are models of the equations of $\mathcal{E}$. We denote the class of $\mathbf{Mod}(\mathbf{Eq_D})$ as **Disp**, which will be called the class of *general displacement algebras*. At the end of the chapter we will see that **FreeDisp** $\subsetneq$ **Disp** by exhibiting two **Disp** algebras which are not **FreeDisp** algebras. It is important to remark that the separation property which we presented before does not hold of all displacement algebras of **Disp**. We will see this at the end of the chapter. As we will prove later, $\mathbf{Eq_D}$ and $\mathbf{Eq_D}_2$ are syntactically equivalent, whence $\mathbf{Mod}(\mathbf{Eq_D}_2)$ is equal to $\mathbf{Mod}(\mathbf{Eq_D})$, i.e. to **Disp**. We can also then identify $\mathbf{Eq_D}$ and $\mathbf{Eq_D}_2$ for they contain the same equations. An important result of this chapter is that $\mathbf{Eq_D} = \mathbf{Eq}(\mathbf{FreeDisp})$.

It is time to show the inference system of (sorted) equational logic. Metaterms appearing in Figure 2.1 range over a set of Σ-terms $T_\Sigma[X]$ for a given unsorted or sorted signature Σ. Semantic inferences are denoted as usual with $\models$. Derivations in equational logic w.r.t. a system of equations $\mathcal{E}$ are denoted by $\vdash_\mathcal{E}$.

$$\frac{}{M \approx M}\,\mathbf{R} \qquad \frac{M \approx N}{N \approx M}\,\mathbf{S}$$

$$\frac{M \approx M' \qquad M' \approx M''}{M \approx M''}\,\mathbf{T}$$

$$\frac{M_1 \approx N_1 \qquad \cdots \qquad M_n \approx N_n,\, n > 0}{f(M_1, \cdots, M_n) = f(N_1, \cdots, N_n)}\,\mathbf{Cong}$$

$$\frac{M \approx N}{\sigma(M) \approx \sigma(N)}\,\mathbf{Subst}$$

Figure 2.1: Inference system for equational logic

A few words on notation. Arbitrary variables of X will usually be denoted x, y instead of x_{ik} for $i, k \in \omega$ and we will generally drop in formal derivations of equations the reference to the equational theories $\mathbf{Eq_D}$ or $\mathbf{Eq_D}_2$. The reader has to keep in mind that equations appearing in Figure 2.2 and in Figure 2.3 are sorted. The following technical definition is crucial for our work in this chapter.[4]

Definition 1 *Given the term* $(t_1 \otimes_i t_2) \otimes_j t_3$ *with* $t_i \in T_{\mathbf{\Sigma_D}}[X]$ *with* $i = 1, 2, 3$, *we say that:*

(P1) $t_2 \prec_{t_1} t_3$ *iff* $i + S(t_2) - 1 < j$.

(P2) $t_3 \prec_{t_1} t_2$ *iff* $j < i$.

(O) $t_2 \between_{t_1} t_3$ *iff* $i \leq j \leq i + S(t_2) - 1$.

Observe that in a term like $(t_1 \otimes_i t_2) \otimes_j t_3$, if $(P1)$ or $(P2)$ hold, then (O) does not apply. Conversely, if (O) is applicable, neither $(P1)$ nor $(P2)$ hold. If $t_2 \prec_{t_1} t_3$, we say that t_2 and t_3 *permute* in t_1 (similarly in the case of $(P3)$). Otherwise, if (O) holds, we say that t_2 *wraps* t_3 in t_1.

[4]The intuition of this definition will be apparent in the following sections.

Continuous associativity

$x \oplus (y \oplus z) \approx (x \oplus y) \oplus z$ (**Assc**)

Discontinuous associativity

$x \otimes_i (y \otimes_j z) \approx (x \otimes_i y) \otimes_{i+j-1} z$ with $y \between_x z$ (**Assc$_\mathbf{d}$1**)
$(x \otimes_i y) \otimes_j z \approx x \otimes_i (y \otimes_{j-i+1} z)$ (**Assc$_\mathbf{d}$2**)

Mixed permutation 1 case $y \prec_x z$

$(x \otimes_i y) \otimes_j z \approx (x \otimes_{j-S(y)+1} z) \otimes_i y$ (**MixPerm1**)
$(x \otimes_i z) \otimes_j y \approx (x \otimes_j y) \otimes_{i+S(y)-1} z$ (**MixPerm1**)

Mixed permutation 2 case $z \prec_x y$

$(x \otimes_i y) \otimes_j z \approx (x \otimes_j z) \otimes_{i+S(z)-1} y$ (**MixPerm2**)
$(x \otimes_i z) \otimes_j y \approx (x \otimes_{j-S(z)+1} y) \otimes_i z$ (**MixPerm2**)

SplitWrap

$x \oplus y \approx (x + 1) \otimes_{S(x)+1} y$
$x \oplus y \approx (1 + y) \otimes_1 x$

Continuous unit and discontinuous unit

$0 \oplus x \approx x \approx x \oplus 0$ and $1 \otimes_1 x \approx x \approx x \otimes_i 1$

Figure 2.2: $\mathbf{Eq_D}$

A simple examination of the operations of displacement algebras gives that both $\mathbf{Eq_D}$ and $\mathbf{Eq_{D}}_2$ hold of in the class **FreeDisp**.

(58) **Theorem** (*Soundness of* $\mathbf{Eq_D}$ *and* $\mathbf{Eq_{D}}_2$ *w.r.t.* **FreeDisp**)

Let $t, s \in T_{\mathbf{\Sigma_D}}[X]$. We have

$\vdash_{\mathbf{Eq_D}} t \approx s$ then **FreeDisp** $\models t \approx s$
$\vdash_{\mathbf{Eq_{D_2}}} t \approx s$ then **FreeDisp** $\models t \approx s$

Proof. By a straightforward inspection of all the equations of $\mathbf{Eq_D}$ and $\mathbf{Eq_{D}}_2$. □

Now we see that both $\mathbf{Eq_D}$ and $\mathbf{Eq_{D}}_2$ are syntactically equivalent, i.e.:

(59) **Theorem** (*Equivalence between* $\mathbf{Eq_D}$ *and* $\mathbf{Eq_{D}}_2$)

Let $t \approx s$ be an equation in the signature of Σ_D. Then:

Continuous associativity

$x \oplus (y \oplus z) \approx (x \oplus y) \oplus z$ (**Assc**)

Discontinuous associativity

$x \otimes_i (y \otimes_j z) \approx (x \otimes_i y) \otimes_{i+j-1} z$ with $y \between_x z$ (**Assc_d1**)
$(x \otimes_i y) \otimes_j z \approx x \otimes_i (y \otimes_{j-i+1} z)$ (**Assc_d2**)

Mixed permutation 1 case $y \prec_x z$

$(x \otimes_i y) \otimes_j z \approx (x \otimes_{j-S(y)+1} z) \otimes_i y$ (**MixPerm1**)
$(x \otimes_i z) \otimes_j y \approx (x \otimes_j y) \otimes_{i+S(y)-1} z$ (**MixPerm1**)

Mixed permutation 2 case $z \prec_x y$

$(x \otimes_i y) \otimes_j z \approx (x \otimes_j z) \otimes_{i+S(z)-1} y$ (**MixPerm2**)
$(x \otimes_i z) \otimes_j y \approx (x \otimes_{j-S(z)+1} y) \otimes_i z$ (**MixPerm2**)

Mixed associativity

$(x \oplus y) \otimes_i z \approx (x \otimes_i y) \oplus z$ iff $1 \leq i \leq S(x)$
$(x \oplus y) \otimes_i z \approx x \oplus (y \otimes_{i-S(x)} z)$ iff $x + 1 \leq i \leq S(x) + S(y)$

Continuous unit and discontinuous unit

$0 \oplus x \approx x \approx x \oplus 0$ and $1 \otimes_1 x \approx x \approx x \otimes_i 1$

Figure 2.3: $\mathbf{Eq_{D}}_2$

$$\vdash_{\mathbf{Eq_D}} t \approx s \text{ iff } \vdash_{\mathbf{Eq_{D_2}}} t \approx s$$

Proof. All the axioms of both theories are equal except **Split-Wrap** and **Mixed Associativity**.

- If part:

  We want to prove that the Split-Wrap is derivable in $\mathbf{Eq_D}_2$, i.e.:

  $$\begin{array}{lcl} \vdash_{\mathbf{Eq_{D_2}}} x \oplus y & \approx & (x \oplus 1) \otimes_{S(x)+1} y \\ \vdash_{\mathbf{Eq_{D_2}}} x \oplus y & \approx & (1 \oplus y) \otimes_1 x \end{array}$$

  We have the following $\vdash_{\mathbf{Eq_{D_2}}}$ derivations:

  $$\begin{array}{lcll} x \oplus y & \approx & x \oplus (1 \otimes_1 y) & \text{by } \textbf{Discontinuous Unit} \\ & \approx & (x \oplus 1) \otimes_{S(x)+1} y & \text{by } \textbf{Mixed Associativity} \end{array}$$

  $$\begin{array}{lcll} x \oplus y & \approx & (1 \otimes_1 x) \oplus y & \text{by } \textbf{Discontinuous Unit} \\ & \approx & (1 \oplus y) \otimes_1 x & \text{by } \textbf{Mixed Associativity} \end{array}$$

We have proved that the Split-Wrap rule is derivable in $\mathbf{Eq_{D}}_2$.

- Only if:

  We want to prove that the Mixed Associativity rule is derivable in $\mathbf{Eq_D}$:

$$\begin{array}{lll} \vdash_{\mathbf{Eq_D}} (x \oplus y) \otimes_i z & \approx & (x \otimes_i z) \oplus y \text{ if } i \leq S(x) \\ \vdash_{\mathbf{Eq_D}} (x \oplus y) \otimes_i z & \approx & x \oplus (y \otimes_{i-S(x)} z) \text{ if } i > S(x) \end{array}$$

  We consider two cases:

  1. $i \leq S(x)$

     We have the following $\vdash_{\mathbf{Eq_D}}$ derivation:

$$\begin{array}{llll} (x \oplus y) \otimes_i z & \approx & ((1 \oplus y) \otimes_1 x) \otimes_i z & \text{by } \textbf{Split Wrap} \\ & \approx & (1 \oplus y) \otimes_1 (x \otimes_i z) & \text{by } \textbf{Mixed Associativity} \text{ for } x \between_{1\oplus y} z \\ & \approx & (x \otimes_i z) \oplus y & \text{by } \textbf{Split-Wrap} \end{array}$$

     Here $x \between_{1\oplus y} z$ because $x \between_{1\oplus y} z$ is equivalent to $1 \leq i \leq 1+S(x)-1 = S(x)$, which is the case.

  2. $S(x) < i \leq S(x) + S(y)$

$$\begin{array}{llll} (x \oplus y) \otimes_i z & \approx & ((x \oplus 1) \otimes_{S(x)+1} y) \otimes_i z & \text{by } \textbf{Split Wrap} \\ & \approx & (x \oplus 1) \otimes_{S(x)+1} (y \otimes_{i-S(x)\not{-}1+\not{1}} z) & \text{by } \textbf{Mixed Associativity} \text{ for } x \between_{1\oplus} \\ & \approx & (x \oplus 1) \otimes_{S(x)+1} (y \otimes_{i-S(x)} z) & \text{by rewriting the above equation} \\ & \approx & x \oplus (y \otimes_{i-S(x)} z) & \text{by } \textbf{Split-Wrap} \end{array}$$

     Here $x \between_{1\oplus y} z$ holds for it is equivalent to $S(x) + 1 \leq i \leq S(x) + 1 + S(y) - 1 = S(x) + S(y)$. The proof is complete.

□

2.2.1 A useful mapping: $\llbracket \cdot \rrbracket$

Let $V_D = (a^k_{ij})_{i,j \in \omega,\ 0 \leq k \leq i}$ be an infinite alphabet. Two new useful notations are introduced: $\overrightarrow{a_{ij}}^n$ and $\overrightarrow{a_{ij}}^n_m$ for $n \geq 0$ and $0 \leq k \leq i$. The first one is the following:

$$\text{For every } n \geq 0,\ \overrightarrow{a_{ij}}^n = \begin{cases} a^0_{ij} \text{ if } n = 0 \\ \overrightarrow{a_{ij}}^{n-1} + 1 + a^n_{ij} & \text{if } 0 < n \leq i \end{cases}$$

For a given $(a^k_{ij})_{0 \leq k \leq i}$ we define $\overrightarrow{a_{ij}}$ as follows:

$$\begin{cases} \overrightarrow{a_{0j}} = a_{0j} \text{ if } i = 0 \\ \overrightarrow{a_{ij}} = \displaystyle\sum_{k=0}^{i-1} (a^k_{ij} + 1) + a^i_{ij} & \text{if } i > 0 \end{cases}$$

Notice that in fact we have that $\overrightarrow{a_{ij}}$ corresponds to $\overrightarrow{a_{ij}}^i$. For example $\overrightarrow{a_{32}}$, $\overrightarrow{a_{00}}$ and $\overrightarrow{a_{11}}$ are respectively:

$$\begin{cases} \overrightarrow{a_{32}} = a_{32}^0 + 1 + a_{32}^1 + 1 + a_{32}^2 + 1 + a_{32}^3 \\ \overrightarrow{a_{00}} = a_{00} \\ \overrightarrow{a_{11}} = a_{11}^0 + 1 + a_{11}^1 \end{cases}$$

We now introduce the second notation we mentioned above, i.e. $\overrightarrow{a_{ij}}{}_m^n$, with $0 \leq m \leq n \leq i$. We have that for every n, m with $0 \leq m \leq n \leq i$:

$$\overrightarrow{a_{ij}}{}_m^n = \begin{cases} a_{ij}^n \text{ if } n = m \\ \overrightarrow{a_{ij}}{}_m^{n-1} + 1 + a_{ij}^n \quad \text{if } 0 \leq m < n \leq i \end{cases}$$

With this notation $\overrightarrow{a_{ij}}$ can also be written as $\overrightarrow{a_{ij}}{}_0^i$.

We recall the definition of the ω-sorted graduated set X of variables of arbitrary sort:

$$\begin{array}{lcl} X & = & \bigcup_{i \in \omega} X_i, \text{ where } X_i \text{ is defined for every } i \text{ as follows:} \\ X_i & = & (x_{ij})_{j \in \omega} \end{array}$$

We define the following Σ_D-homomorphism from $T_{\Sigma_D}[X]$ to the standard displacement algebra defined by the syntactical algebra $\langle (V_D \cup \{1\})^*, +, \Lambda, 1 \rangle$ (we call this algebra $\mathcal{A}_{V_D}$):

$$\begin{array}{llll} [\![\cdot]\!]: & T_{\Sigma_D}[X] & \longrightarrow & \mathcal{A}_{V_D} \\ & 0 & \mapsto & \Lambda \\ & 1 & \mapsto & 1 \\ & x_{ij} & \mapsto & \overrightarrow{a_{ij}}^i \end{array}$$

$[\![\cdot]\!]$ is then defined recursively as follows:

$$\begin{array}{lcl} t \oplus s & \overset{[\![\cdot]\!]}{\mapsto} & [\![t]\!] + [\![s]\!] \\ t \otimes_i s & \mapsto & [\![t]\!] \times_i [\![s]\!] \end{array}$$

The mapping $[\![\cdot]\!]$ is a Σ_D-homomorphism between the term algebra $T_{\mathbf{\Sigma_D}}[X]$ and a special displacement algebra of **freeDisp**. It follows then that by soundness:

(60) **Lemma**
if $\vdash_{\mathbf{Eq_D}} t \approx s$ or $\vdash_{\mathbf{Eq_{D_2}}} t \approx s$ then:

$$[\![t]\!] = [\![s]\!]$$

Let us see an example:

(61) **Example**

$$\begin{array}{lcl} [\![x_{05}]\!] & = & a_{05} \\ [\![x_{10}]\!] & = & a_{10}^0 + 1 + a_{10}^1 \\ [\![x_{21}]\!] & = & a_{21}^0 + 1 + a_{21}^1 + 1 + a_{21}^2 \\ [\![x_{21} \otimes_2 0]\!] & = & a_{21}^0 + 1 + a_{21}^1 + a_{21}^2 \end{array}$$

Observe that $Im([\![\cdot]\!])$ is a general displacement algebra.

2.2.2 A normal form for $T_{\Sigma_{\mathbf{D}}}[X]$ terms in $\mathbf{Eq_D}$

We now define a relation $\rhd$ at the level of $T_{\Sigma_{\mathbf{D}}}[X]$ which will be crucial for the main results of this chapter.

(62) **R1** Unit elimination: if $t = 0 \oplus s$ or $t = s \oplus 0$ transform t to s: $t \rhd s$. If $t = 1 \otimes_1 s$ or $t = s \otimes_i 1$ transform t to s: $t \rhd s$.

R2 If $t = s \oplus k$ then if $s = s_1 \oplus s_2$, then $t \rhd s_1 \oplus (s_2 \oplus k)$.

R3 If $t = s \otimes_i k$. If $s = s_1 \oplus s_2$, then $t \rhd (s_1 \otimes_i k) \oplus s_2$, if $1 \leq i \leq S(s_1)$, or $t \rhd s_1 \oplus (s_2 \otimes_{i-s_1} k)$, if $S(s_1) < i$.

R4 If $t = (s_1 \otimes_i s_2) \otimes_j k$, and in case that $k \prec_{s_1} s_2$, which holds iff $j < S(k)$, then $t \rhd (s_1 \otimes_i k) \otimes_{k+S(s_2)-1} s_2$.

R5 If $t = (s_1 \otimes_i s_2) \otimes_j k$, and in case that $k \between_{s_1} s_2$, which holds iff $i \leq j \leq i + S(s_2) - 1$, then $t \rhd s_1 \otimes_i (s_2 \otimes_{j-i+1} k)$

$\rhd^*$ is defined to be the reflexive and transitive closure of $\rhd$. The reductions **R1-R5** for $\rhd$ are motivated by the following facts:

(63)

- **R1** is motivated by the rules of the continuous and discontinuous units:

$$0 \oplus x \approx x \quad x \oplus 0 \approx x$$
$$1 \otimes_1 x \approx x \quad x \otimes_i 1 \approx x,\ 1 \leq i \leq S(x)$$

- **R2** is motivated by continuous associativity.
- **R3** is motivated by $\mathbf{Eq_D^2}$ displacement axiom mixed associativity **MixAssc**:[5]

$$(s_1 \oplus s_2) \otimes_i k \approx (s_1 \otimes_i k) \oplus s_2 \text{ if } i \leq S(s_1)$$
$$(s_1 \oplus s_2) \otimes_i k \approx s_1 \oplus (s_2 \otimes_{i-S(s_1)} k) \text{ if } i > S(s_1)$$

- **R4** is motivated by the mixed permutation rule **MixPerm**.
- **R5** is motivated by the discontinuous associativity rule.

(64) **Definition** (*Normal form for* $T_{\Sigma_{\mathbf{D}}}[X]$ *terms*)

Given a term $t \in T_{\mathbf{D}}[X]$, we say that t is in $\mathbf{Eq_D}$ *normal form* iff no reduction from the reductions **R1-R5** can be applied to it.

In the next section we take a look at some properties of $T_{\Sigma_{\mathbf{D}}}[X]$ terms in normal form. We will usually write a *term in normal form* instead of the more formal *term* in $\mathbf{Eq_D}$ *normal form*.

[5] As we saw before $\mathbf{Eq_D}$ and $\mathbf{Eq_D}2$ are equivalent: **MixAssc** is a derived rule in $\mathbf{Eq_D}$ (cf. the previous section).

2.2.3 Properties of terms in normal form

We present some properties of terms in normal form. In the following lemma t denotes an arbitrary term in normal form.

(65) **Lemma** (*Properties of Terms in Normal Form*)

The following properties hold:

i) 0 and 1 are in normal form.

ii) Every subterm of t is in normal form.

iii) t is *unit-free*, i.e., no subterm of t can be $0 \oplus s$, $s \oplus 0$, $1 \otimes_1 s$ or $s \otimes_i 1$ $(1 \leq i \leq S(s))$.

iv) If the main constructor of t is a discontinuous product, say $\otimes_i$ for some i, then there exists a variable $x_{nj} \in X$, which we call the *foot* with sort $S(x_{nj}) = n > 0$, and terms $t_i, i = 1, \cdots, m$ in normal form such that:

$$t = (\cdots (x_{nj} \otimes_{i_1} t_1) \otimes_{i_2} t_2 \cdots) \otimes_{i_m} t_m$$

Moreover the terms t_k's *permute from left to right in* t meaning that:

$$(66) \quad \begin{array}{l} t_1 \prec_x t_2 \\ t_2 \prec_{x \times_{i_1} t_1} t_3, \text{ if } k \geq 3 \\ \vdots \\ t_{k-1} \prec_{(\cdots(x \times_{i_1} t_1) \cdots \times_{i_{k-2}} t_{k-2})} t_k, \text{ if } k \geq 3 \end{array}$$

If $k = 1$ then $t = x \otimes_{i_1} t_1$ and there is no constraint of the kind of (66). If $k = 2$ then in (66) we have only the first line, i.e.:

$$t_1 \prec_x t_2$$

v) 1 can only appear in $\oplus$ contexts, e.g. $t[r \oplus 1]$, whereas 0 can only appear in wrapping contexts, e.g. $t[s \otimes_i 0]$, for $i > 0$, for some subterms r and s.

vi) Terms in normal form are $\oplus$ right-associative. If $t = r \oplus s$ then either $r = x_{ij}$ (for some $x_{ij} \in X$) or 1, or $r = l \otimes_i k$, for terms r, k and index i. An interesting consequence is that if $t = r \oplus s$, then $l([\![r \oplus s]\!]) \geq 2$, where for any string α, $l(\alpha)$ is the length of α.

vii) If s is a term in normal form such that $[\![s]\!] = 0$, then $s = 0$.

viii) If s is a term in normal form such that $[\![s]\!] = 1$ or $[\![s]\!] = \overrightarrow{a_{ij}}$, then either $s = 1$ or $s = x_{ij}$ for a given variable $x_{ij} \in X$.

ix) If t is not of the form $r \oplus s$ for some terms r and s, then $[\![t]\!]$ is *prefix-free*, i.e. no term s in normal form is such that $[\![t]\!] = [\![s]\!] + \alpha$ with α different from 0.

x) No reduction step increases the complexity of a term.[6]

Proof.

[6] Contrast this with the simply typed lambda calculus where a β reduction of a term M, $M \rhd_\beta M'$, can increase the complexity of the original term, i.e., $|M'| > |M|$.

i), *ii*), *iii*) and *v*) are obvious. We prove *iv*).

iv) t cannot have the form:

$$t = (r \oplus s) \otimes_i t_2, \text{ for some } i > 0$$

because in that case t would not be in normal form for otherwise step **R3** would apply obtaining:

$$t \approx (r \otimes_i k) \oplus s \text{ if } i \leq S(t) \quad \text{or} \quad t \approx r \oplus (s \otimes_{i-S(s)} k) \text{ if } i > S(t)$$

By the same reasoning t must be equal to $(t_{11} \otimes_{i_2} t_{12}) \otimes_i t_2$ for some terms t_{11}, t_{12} and index i_2. Repeating this process (which is of course finite) we have that t is what we call a *product term*, i.e. t has the form:

$$t = (\cdots (t' \otimes_{i_1} t_1) \otimes_2 \cdots) \otimes_{i_m} t_m \text{ for some indexes } i_k, m \text{ and terms } t', t_{i_k}.$$

Now, since t is in normal form no subterm t_{i_k} can wrap its next term $t_{i_{k+1}}$ for in that case step **R4** would apply. By the same token we cannot have a precedence inequality such as the following:

$$t_k \prec_{(\cdots(x \times_{i_1} t_1)\cdots) \times_{i_{k-2}} t_{k-2}} t_{k-1}, \text{ if } k \geq 3$$

because in that case step **R4** would apply again. Finally, t' cannot be a discontinuous product (by assumption) nor a term of the form $r \oplus s$ for some terms r and s (in that case **R3** would apply!). Hence t' is equal either to a variable x_{nj} for some n, j with $n > 0$, or to 1. The latter case is not possible because in that case t would not be unit-free. It follows that $t' = x_{nj}$. We are done.

vi) Suppose $t = r \oplus s$ with $r = r_1 \oplus r_2$. This is not possible since **R2** would apply contradicting the fact that t is in normal form. Hence, r can be either 1 or a variable or a product term $r_1 \otimes_i r_2$ for some i and terms r_1, r_2.

Clearly $l(\llbracket r_1 \rrbracket) \geq 1$ as well as $l(\llbracket r_2 \rrbracket) \geq 1$. Hence $l(\llbracket r_1 \oplus r_2 \rrbracket) \geq 2$.

vii) We have that $l(\llbracket t \rrbracket) = 0$. Suppose:

- If $t = 1$ or $t = x$ for some $x \in X$ then $l(\llbracket t \rrbracket) \neq 0$.
- If $t = r \oplus s$ for some terms r, s.then by vi) we would have that $l(\llbracket t \rrbracket) \neq 0$.
- If $t = r \oplus_i s$ for some terms r, s and index i, then again $l(\llbracket t \rrbracket) \neq 0$ for by iv) t has a foot $x_{ij} \in X$ for some i, j $S(x) = i > 0$. In this case $l(\llbracket t \rrbracket)$ would contain elements of a^k_{ij}.

  In either case the length of $\llbracket t \rrbracket$ would be different from 0. We are done.

viii) • Suppose $[\![t]\!] = 1$.

Obviously t cannot be 0. Suppose t were a variable $x_{0j} \in X$ for some $j \in \omega$. We have that $l([\![x_{0j}]\!]) = 1$ but $[\![t]\!]$ should be equal to $[\![x_{0j}]\!]$, i.e. we would have $1 = a_{0j}$, which is impossible. If t were a variable x of sort greater than 0, we would have that $l([\![x]\!]) > 1$ contradicting that $l([\![t]\!]) = 1$.

Now, suppose $t = r \oplus s$ for some terms r and s. By vi) r should be equal to 1. But since s is not 0 (for in that case t would not be in normal form), it follows that $l([\![s]\!]) \neq 0$, and hence $l([\![t]\!]) > 1$. Contradiction.

Finally, if t were a product term (see iv)) t's foot would contribute to $[\![t]\!]$ with at least two elements $a_{ij}^{k_1}$ and $a_{ij}^{k_2}$. Again in this case, $l([\![t]\!])$ would be greater than 1. Contradiction.

It follows that t must be equal to 1.

• Suppose $[\![t]\!] = \overrightarrow{a_{ij}}$ for some $i, j \in \omega$.

It is obvious that t cannot be 1 nor 0.

If $t = r \oplus s$ for some terms r and s, then $[\![t]\!])$ could not be equal to $\overrightarrow{a_{ij}}$.

If $t = r \otimes_i s$ for some terms r, s and index i, then again $[\![t]\!])$ could not be equal to $\overrightarrow{a_{ij}}$.

Hence t is necessarily equal to x_{ij}.

ix) If t is equal to 0, 1, or $x_{0j} \in X$ ($S(x_{0j}) = 0$), then t is obviously prefix-free.

Suppose $t = r \otimes_i s$ for some terms r, s and index i. By iv) t has a foot $x_{nj} \in X$ for some indexes n, j such that $S(x_{nj}) = n > 0$. Applying the map $[\![\cdot]\!]$ to t we have that:

$$[\![t]\!] = a_{nj}^0 + \alpha_1 + a_{nj}^0 + \cdots + a_{nj}^{n-1} + \alpha_n + a_{nj}^n$$

Let s be such that $[\![t]\!] = [\![s]\!] + \beta$ with $\beta \neq \Lambda$. We prove that this is impossible by induction on the structure of terms s in normal form. Cases where $s = 0$ or $s = 1$ or $s = x_{0l}$ are obvious. Suppose $s = x_{kl}$. By comparing $[\![s]\!]$ and $[\![t]\!]$ we would have forcefully that $k = n$ and $l = j$, i.e. $x_{nj} = x_{kl}$. Clearly $[\![x_{nj}]\!]$ cannot be a prefix of $[\![t]\!]$.

Case: $s = s_1 \oplus s_2$. If $s_1 = 1$ or $s_1 = x_{lj}$ or s is a product term then again $[\![s]\!]$ cannot be a prefix of $[\![t]\!]$.

Finally, for the case where s is a product term, forcefully the foot of s should coincide with t's foot. Like before clearly $[\![s]\!]$ cannot be a prefix of $[\![t]\!]$. This ends the recursion on s.

Summing up, $[\![t]\!]$ is prefix-free. We are done.

x) It is obvious that no reduction step increases the complexity of a term.

□

2.2.4 On positions of terms in a product term which permute from left to right

Let x be an arbitrary variable of sort strictly positive. Then, $x = x_{ij}$ for some $i, j \in \omega$ with $i = S(x)$. We recall that product terms in normal permute also from left to right. Consider the term t:

$$t = (\cdots((x \otimes_{i_1} t_1) \otimes_{i_2} t_2) \cdots \otimes_{i_{k-1}} t_{k-1}) \otimes_{i_k} t_k$$

Suppose t permutes from left to right:

(67)
$$\begin{array}{l} t_1 \prec_x t_2 \\ t_2 \prec_{x \otimes_{i_1} t_1} t_3, \text{ if } k \geq 3 \\ \vdots \\ t_{k-1} \prec_{(\cdots(x \otimes_{i_1} t_1) \cdots \otimes_{i_{k-2}} t_{k-2})} t_k, \text{ if } k \geq 3 \end{array}$$

We define now the following function from the finite set of terms $\{t_1, \cdots, t_k\}$ to ω:

$$\begin{array}{llll} \mathbf{pos}^{\mathbf{t}}_{\mathbf{x}}: & \{t_1, \cdots, t_k\} & \longrightarrow & \omega \\ & t_1 & \mapsto & i_1 \\ & t_l & \mapsto & i_l - S(t_1) - \cdots - S(t_{l-1}) + l - 1 \text{ if } 1 < l \leq k \end{array}$$

(68) **Remark**
The reader has to keep in mind that this function $\mathbf{pos}^{\mathbf{t}}_{\mathbf{x}}$ only applies to product terms which permute from left to right.

In the following we will give the intuitive meaning of the function $\mathbf{pos}^{\mathbf{t}}_{\mathbf{x}}$ which we call *position*. $\mathbf{pos}^{\mathbf{t}}_{\mathbf{x}}(t_l)$ (for $1 \leq l \leq k$) is the position of the term t_l relatively to the foot[7] x of the term t. This function has notable properties:

(69) **Theorem** (*Properties of Positions of Terms*)

The following two properties hold:

i) Strict monotonicity:

For every $l, l' \in \{1, \cdots, k\}$ such that $l < l'$, $\mathbf{pos}^{\mathbf{t}}_{\mathbf{x}}(t_l) < \mathbf{pos}^{\mathbf{t}}_{\mathbf{x}}(t_{l'})$

ii) The range of $\mathbf{pos}^{\mathbf{t}}_{\mathbf{x}}$ is contained in the set:

$$\{1, \cdots, S(x)\}$$

[7] Recall that in a product term $(\cdots(x \otimes_{i_1} t_1) \cdots \otimes_{i_{n-1}} t_{n-1}) \otimes_{i_n} t_n$ x is called the *foot* of the product term.

Proof.

i) Strict Monotonicity: If $k = 1$ there is nothing to prove. Suppose $k = 2$. Hence:

$$\mathbf{pos}^{\mathbf{t}}_{\mathbf{x}}(t_1) < \mathbf{pos}^{\mathbf{t}}_{\mathbf{x}}(t_2)$$

Since the above inequation is equivalent to:

$$\begin{array}{c} i_1 < i_2 - S(t_1) + 1 \\ \text{iff} \\ i_1 + S(t_1) - 1 < i_2 \\ \text{iff} \\ t_1 \prec_x t_2 \end{array}$$

The last line is true because t_1 and t_2 permute from left to right in t.

Suppose then that $k \geq 3$. We prove that for every l such that $1 < l \leq k$:

$$\mathbf{pos}^{\mathbf{t}}_{\mathbf{x}}(t_{l-1}) < \mathbf{pos}^{\mathbf{t}}_{\mathbf{x}}(t_l)$$

By the definition of $\mathbf{pos}^{\mathbf{t}}_{\mathbf{x}}$ we have that:

$$\begin{array}{lcl} \mathbf{pos}^{\mathbf{t}}_{\mathbf{x}}(t_{l-1}) & = & i_{l-1} - S(t_1) - \cdots - S(t_{l-2}) + l - 1 \\ \mathbf{pos}^{\mathbf{t}}_{\mathbf{x}}(t_l) & = & i_l - S(t_1) - \cdots - S(t_{l-1}) + l \end{array}$$

Hence:

$$\begin{array}{l} i_{l-1} - S(t_1) - \cdots - S(t_{l-2}) + l - 1 < i_l - S(t_1) - \cdots - S(t_{l-1}) + l \\ \qquad\qquad \text{iff} \\ i_{l-1} - 1 < i_l - S(t_{l-1}) \\ \qquad\qquad \text{iff} \\ i_{l-1} + S(t_{l-1}) - 1 < i_l \\ \qquad\qquad \text{iff} \\ t_{l-1} \prec_{(\cdots((x \otimes_{i_1} t_1) \otimes_{i_2} t_2) \cdots) \otimes_{i_{l-2}} t_{l-2}} t_l (\star) \end{array}$$

Where the last line $(\star)$ is true by hypotesis, i.e., t_{l-1} and t_l permute from left to right. Hence $\mathbf{pos}^{\mathbf{t}}_{\mathbf{x}}(t_{l-1}) < \mathbf{pos}^{\mathbf{t}}_{\mathbf{x}}(t_l)$. We have therefore that:

(70) $\mathbf{pos}^{\mathbf{t}}_{\mathbf{x}}(t_1) < \mathbf{pos}^{\mathbf{t}}_{\mathbf{x}}(t_2) < \cdots < \mathbf{pos}^{\mathbf{t}}_{\mathbf{x}}(t_k)$

Hence, for every $l, l' \in \{1, \cdots, k\}$ such that $l < l'$ we have that:

$$\mathbf{pos}^{\mathbf{t}}_{\mathbf{x}}(t_l) < \mathbf{pos}^{\mathbf{t}}_{\mathbf{x}}(t_{l'})$$

This proves i).

ii) By definition, since $\mathbf{pos}^{\mathbf{t}}_{\mathbf{x}}(t_1) = i_1 \geq 1$ and the property of strict monotonicity property (70), it follows that for every l:

$$1 \leq \mathbf{pos}^{\mathbf{t}}_{\mathbf{x}}(t_l)$$

This proves the first inequality. Now, for every l we have that:

$$1 \leq i_l \leq S((\cdots(x \otimes_{i_1} t_1) \cdots) \otimes_{i_{l-1}} t_{l-1}) = S(x) + S(t_1) + \cdots + S(t_{l-1}) - l + 1$$

Since we recall that the term t has the form:

$$((\cdots(x \otimes_{i_1} t_1)\cdots) \otimes_{i_{l-1}} t_{l-1}) \otimes_{i_l} t_l$$

i_l must therefore satisfy:

$$\begin{array}{c} 1 \leq i_l \leq S(x) + S(t_1) + \cdots + S(t_{l-1}) - l + 1 \\ \text{iff} \\ i_l - S(t_1) - \cdots - S(t_{l-1}) + l - 1 \leq S(x) \end{array}$$

And the left hand side of the above inequality is:

$$\mathbf{pos^t_x}(t_l)$$

I.e, for any l such that $1 \leq l \leq k$:

$$\mathbf{pos^t_x}(t_l) \leq S(x)$$

It follows that the range of $\mathbf{pos^t_x}$ is contained in $\{1, \cdots, S(x)\}$. This proves ii).

□

(71) **Corollary** (*Number of Terms in a Product Term which Permutes from Left to Right*)

Let t be a product term which permutes from left to right:

$$t = (\cdots((x \otimes_{i_1} t_1) \otimes_{i_2} t_2) \cdots \otimes_{i_{k-1}} t_{k-1}) \otimes_{i_k} t_k$$

It follows that the number of terms $t_1, \cdots, t_k$ is bounded by the sort of the foot x of t, i.e.:

$$1 \leq k \leq S(x)$$

Proof. Suppose that $k > S(x)$. By lemma (71) $\mathbf{pos^t_x}(\{\mathbf{t_1}, \cdots, \mathbf{t_k}\})$ forms a strictly increasing sequence of natural numbers contained in the natural number interval $[1, S(x)]$. We have therefore a strictly increasing sequence of natural numbers of length k. The longest chain in this interval must be bounded above by $S(x)$. Contradiction because $\mathbf{pos^t_x}(\{\mathbf{t_1}, \cdots, \mathbf{t_k}\})$ forms a chain strictly greater than $S(x)$. □

It follows that if we have a product term t with foot x and l wrapped terms in the product term such that $l > S(x)$, then t cannot be in normal form.

What is the intuition behind the function $\mathbf{pos^t_x}$? We will prove that if we have a term $t = (\cdots((x \otimes_{i_1} t_1) \otimes_{i_2} t_2) \cdots \otimes_{i_{k-1}} t_{k-1}) \otimes_{i_k} t_k$, then applying the map $[\![\cdot]\!]$ to t there holds the following property:

$$\begin{array}{l} [\![t]\!] = a^0_{ij} + 1 + a^1_{ij} + \\ a^{\mathbf{pos^t_x}(t_1)-1}_{ij} + [\![t_1]\!] + a^{\mathbf{pos^t_x}(t_1)}_{ij} + \\ \cdots + a^{\mathbf{pos^t_x}(t_l)-1}_{ij} + [\![t_l]\!] + a^{\mathbf{pos^t_x}(t_l)}_{ij} + \cdots + \\ \cdots + a^{\mathbf{pos^t_x}(t_k)-1}_{ij} + [\![t_k]\!] + a^{\mathbf{pos^t_x}(t_k)}_{ij} + \cdots + a^i_{ij} \end{array}$$

where $x = x_{ij}$ with $i = S(x)$. We see then that the function $\mathbf{pos}^{\mathbf{t}}_{\mathbf{x}}$ gives the position of the string $\llbracket t_l \rrbracket$ relative to $(a^k_{ij})_{k=0,\cdots,i}$. Where $\alpha, \beta \in ((a^k_{ij})_{i,j\in\omega} \text{ and } {}_{0\leq k\leq i} \cup \{1\})^*$, we will say that $\llbracket t_l \rrbracket$ occurs in $\llbracket t \rrbracket$ *near* a^r_{ij} when:

$$\llbracket t \rrbracket = \alpha + a^{r-1}_{ij} + \llbracket t_l \rrbracket + a^r_{ij} + \beta$$

We note that if we apply an axiom of $\mathbf{Eq}_D$ to the term t, the occurrence of $\llbracket t_l \rrbracket$ does not change for we have that:

$$\text{If } t \approx t' \text{ then } \llbracket t \rrbracket = \llbracket t' \rrbracket$$

Let us prove formally the intuition behind $\mathbf{pos}^{\mathbf{t}}_{\mathbf{x}}$ we talked about above. We have the following lemma:

(72) **Theorem** (*Intuitive Meaning of the Function* $\mathbf{pos}^{\mathbf{t}}_{\mathbf{x}}$)

Let $t = (\cdots((x \otimes_{i_1} t_1) \otimes_{i_2} t_2) \cdots \otimes_{i_{k-1}} t_{k-1}) x \otimes_{i_k} t_k$. We suppose t permutes from left to right. It follows that for every $1 \leq l \leq k$, $\llbracket t_l \rrbracket$ occurs in t near $a^{\mathbf{pos_x}(t_l)}_{ij}$, i.e.:

$$\llbracket t \rrbracket = \alpha + a^{\mathbf{pos_x}(t_l)-1}_{ij} + \llbracket t_l \rrbracket + a^{\mathbf{pos_x}(t_l)}_{ij} + \beta \ (\star)$$

where as before, $\alpha, \beta \in V^*_D = ((a^k_{ij})_{i,j\in\omega} \text{ and } {}_{0\leq k\leq i} \cup \{1\})^*$.

Proof. We suppose that x occurs linearly in t. The result generalizes to nonlinear occurrences of x using the concept of distinguished occurrence of x.

We prove the result by induction on the number k of terms $t_1, \cdots, t_k$ (with $k \geq 1$).

- Case $k = 1$: In this case we have that $\mathbf{pos_x}(t_1) = i_1$.

  Then by definition of the map $\llbracket \cdot \rrbracket$, applying $\llbracket \cdot \rrbracket$ to $t = x \otimes_{i_1} t_1$ is the result of replacing the i_1-th occurrence of 1 in $\llbracket x \rrbracket$. We have hence that $\mathbf{pos}^{\mathbf{t}}_{\mathbf{x}}(t_1) = i_1$ occurs in t near $\mathbf{pos}^{\mathbf{t}}_{\mathbf{x}}(t_1)$. This proves the case $k = 1$.

- Case $k = 2$: In this case we have that:

  $$t = (x \otimes_{i_1} t_1) \otimes_{i_2} t_2$$

  Since t is in normal form, we have that:

  $$t_1 \prec_x t_2$$

  Hence t_1 and t_2 permute in t. So we have that the following equation holds:

  $$(x \otimes_{i_1} t_1) \otimes_{i_2} t_2 \approx (x \otimes_{i_2 - S(t_1)+1} t_2) \otimes_{i_1} t_1 =: t'$$

  $x \otimes_{i_2 - S(t_1)+1} t_2$ is in normal form and the number of terms wrapped in t is equal to 1. We can then apply the induction hypothesis (i.h.) whence:

t_2 occurs near $\mathbf{pos_x^{x\otimes_{i_2-S(t_1)+1}}}(t_2) = i_2{-}S(t_1){+}1$ in $(x{\otimes_{i_2-S(t_1)+1}}t_2){\otimes_{i_1}}t_1$

t_2 occurs also near $i_2 - S(t_1) + 1$ in t'. Since $[\![t]\!] = [\![t']\!]$ (because $t \approx t'$), it follows that t_2 occurs near $i_2 - S(t_1) + 1$ in t. Finally we realize that $\mathbf{pos_x^t}(t_2)$ is equal to $i_2 - S(t_1) + 1$. Therefore t_2 occurs near $\mathbf{pos_x^t}(t_2)$ in t. This proves the case $k = 2$.

- Case $k \geq 3$: We have that:

$$t = ((\cdots((x \otimes_{i_1} t_1)x \otimes_{i_2} t_2) \cdots \otimes_{i_{k-2}} t_{k-2}) \otimes_{i_{k-1}} t_{k-1}) \otimes_{i_k} t_k$$

Since t is in normal form it follows that:

(73) $t_{k-1} \prec_{(\cdots(x\otimes_{i_1}t_1)\cdots)\otimes_{i_{k-2}}t_{k-2}} t_k$

We can apply Mixed Permutation to the term t (because t is in normal form!) obtaining:

$t \approx ((\cdots((x{\otimes_{i_1}}t_1)x{\otimes_{i_2}}t_2)\cdots{\otimes_{i_{k-2}}}t_{k-2})){\otimes_{i_k-S(t_{k-1}+1}}t_k){\otimes_{i_{k-1}}}t_{k-1} =: t'$

The term:

$$A := (\cdots((x \otimes_{i_1} t_1)x \otimes_{i_2} t_2) \cdots \otimes_{i_{k-2}} t_{k-2}) \otimes_{i_k-S(t_{k-1})+1} t_k$$

has $k-1$ wrapped terms and moreover is in normal form since:

$$i_{k-2} + S(t_{k-2}) - 1 < i_{k-1} < i_l - S(t_{k-1}) + 1$$

We can then apply i.h. to the term A. And we know that t_k occurs near $\mathbf{pos_x^A}(t_k)$ in the term A: we calculate $\mathbf{pos_x^A}(t_k)$. Then:

(74) $\mathbf{pos_x^A}(t_k) = (i_k - S(t_{k-1}) + 1) - S(t_1) - \cdots - S(t_{k_2}) + k - 2 =$
$i_k - S(t_1) - \cdots - S(t_{k-2}) - S(t_{k-1}) + k - 1$

t_k occurs near $\mathbf{pos_x^A}(t_k)$ in A and hence it occurs near $\mathbf{pos_x^A}(t_k)$ in t' as well as in t. We see that $\mathbf{pos_x^A}(t_k)$ is precisely $\mathbf{pos_x^t}(t_k)$. It follows that t_k occurs near $\mathbf{pos_x^t}(t_k)$ in t. This completes the induction step. We are done.

□

(75) **Lemma**
Let $t = (\cdots((x \otimes_{i_1} t_1) \otimes_{i_2} t_2) \cdots) \otimes_{i_n} t_n$ with $n \geq 1$. t has a normal form.

Proof. We prove by induction on n that by applying a finite number of times the reducing step **R4** to t we get:

$t \approx (\cdots(x \otimes_{j_1} s_1) \otimes_{j_2} \cdots s_{m-1}) \otimes_{j_m} s_m$ for $m \geq 1$ and for some $j_1, \cdots, j_k \in \omega$ and terms s_{j_k}

Such that:

(76)
$$\begin{array}{l} s_1 \prec_x s_2 \\ s_2 \prec_{x \times_{i_1} s_1} s_3 \\ \vdots \\ s_{k-1} \prec_{(\cdots(x \otimes_{i_1} s_1)\cdots)\otimes_{i_{k-2}} s_{k-2}} s_k \end{array} \qquad (\star)$$

The case $n = 1$ is obvious. The case $n = 2$ is also trivial since either mixed permutation or mixed associativity apply.

Suppose $n \geq 3$. By induction hypothesis (i.h.) we have that:

$$(\cdots(x \otimes_{j_1} t_1) \otimes_{j_2} \cdots t_{n-2}) \otimes_{j_{n-1}} t_{n-1} \approx (\cdots(x \otimes_{j_1} s_1) \otimes_{j_2} \cdots) \otimes_{j_m} s_m$$

In such a way that $(\star)$ applies to the right hand side of the above equation. We now have the term:

$$t \approx ((\cdots(x \otimes_{j_1} s_1) \otimes_{j_2} \cdots) \otimes_{j_{m-1}} s_m) \otimes_{i_n} t_n$$

To the above equation we apply the reducing step **R4** (permuting t_n to the left) as many times as possible obtaining the two following possible results (where the term displaced to the left is underbraced for more clarity):

$$\begin{cases} ((\cdots(x \otimes_{j_1} s_1) \otimes_{j_2} \cdots) \otimes_{j_l} s_l) \underbrace{\otimes_{i_n} t_n} \otimes_{j_{l+1}+S(t_n)-1} s_{l+1} \otimes_{j_{l+2}+S(t_n)-1} \cdots) \otimes_{j_m+S(t_n)-1} s_m & (1) \\ \text{Or} \\ ((\cdots(x \otimes_{j_1} s_1) \otimes_{j_2} \cdots) \otimes_{j_l} (s_l \underbrace{\otimes_{j_l+i_n-1} t_n}) \otimes_{j_{l+1}+S(t_n)-1} s_{l+1} \otimes_{j_{l+2}+S(t_n)-1} \cdots) \otimes_{j_m+S(t_n)-1} s_m & (2) \end{cases}$$

We have to check that both (1) and (2) expressions permute from left to right. In the case of (1), by i.h. s_i with $i = 1, \cdots, l$ permute from left to right. We have that by assumption (t_n has been displaced to the left as many times as possible):

$$s_l \prec_{(\cdots(x \otimes_{j_1} s_1) \otimes_{j_2} \cdots) \otimes_{j_l} s_{l-1} s_{l-1}} t_n$$

In the case of:

$$t_n \prec_{(\cdots(x \otimes_{j_1} s_1) \otimes_{j_2} \cdots) \otimes_{j_l} s_l} s_{l+1}$$

The above inequality holds for:

$$i_n + S(t_n) - 1 < j_{l+1} + S(t_n) - 1 \text{ which is equivalent to } i_n < j_{l+1}$$

And $i_n < j_{l+1}$ is true because it is the condition for t_n to permute to the left. Finally the terms s_{l+k} precede s_{l+k+1} in (1) because

$$j_{l+k} + S(t_n) - 1 + S(s_{j_{l+k}}) - 1 < j_{l+k+1} + S(t_n) - 1$$

which is equivalent to $j_{l+k} + S(s_{j_{l+k}}) - 1 < j_{l+k+1}$ which is true by i.h..
In the case of (2), the term precedence inequalities which have to be checked are the same as (1) but with the term precedence inequality below:

$$s_l \otimes_{j_l+i_n-1} t_n \prec_{(\cdots(x \otimes_{j_1} s_1) \otimes_{j_2} \cdots) \otimes_{j_{l-1}}} s_{l+1}$$

The above precedence inequation holds since:

$$j_l + S(s_l) + S(t_n) - 1 - 1 < j_{l+1} + S(t_n) - 1$$

which is equivalent by i.h. to the true inequality:

$$j_l + S(s_l) - 1 < j_{l+1}$$

The proof is complete. □

(77) **Lemma** (*Product Terms in Normal Form*)

Consider the following two product terms in normal form:

$$\begin{array}{l} t = (\cdots((x \otimes_{i_1} t_1) \otimes_{i_2} t_2) \cdots) \otimes_{i_n} t_n \\ t' = (\cdots((x' \otimes_{i'_1} t'_1) \otimes_{i'_2} t'_2) \cdots) \otimes_{i'_m} t'_m \end{array}$$

This means that t and t' permute from left to right, that t_i $(1 \leq i \leq n)$ and t'_j $(1 \leq j \leq m)$ are in normal form, and finally, that neither t_i nor t'_j is equal to 1. Suppose that $\llbracket t \rrbracket = \llbracket t' \rrbracket$. It follows that:

$$\begin{cases} x = x' \\ n = m \\ i_k = i'_k \text{ for } 1 \leq k \leq n \end{cases}$$

Proof. We suppose without loss of generality that the feet x and x' occur linearly respectively in the terms t and t'. We have that there exist j, j' such that $\llbracket x \rrbracket = \overrightarrow{a_{S(x),j}}$ and $\llbracket x' \rrbracket = \overrightarrow{a_{S(x'),j'}}$. We ease the notation and write $\overrightarrow{a}$ instead of $\overrightarrow{a_{S(x),j}}$, $\overrightarrow{b}$ instead of $\overrightarrow{a_{S(x'),j'}}$, n instead of $S(x)$ and m instead of $S(x')$. By theorem (72) we have that for every i, k:

$$\begin{cases} \llbracket t \rrbracket = a^0 + \alpha_0 + a^{\mathbf{pos^t_x(t_i)}-1} + \llbracket t_i \rrbracket + a^{\mathbf{pos^t_x(t_i)}} + \alpha_1 + a^n \\ \llbracket t' \rrbracket = b^0 + \beta_0 + b^{\mathbf{pos^{t'}_{x'}(t'_k)}-1} + \llbracket t'_k \rrbracket + b^{\mathbf{pos^{t'}_{x'}(t'_k)}} + \beta_1 + b^m \end{cases}$$

Where $\alpha_i, \beta_i \in ((a^k_{ij})_{i,j\in\omega} \text{ and } _{0\leq k \leq i} \cup \{1\})^*$. Since neither of t_i, t'_j is equal to 1, we have that equating $\llbracket t \rrbracket = \llbracket t' \rrbracket$ satisfies the following properties:

$$n = m,\ a_i = b_i (\text{for } i = 0, \cdots, n) \text{ and that for every } i = 1, \cdots, n$$

$$\begin{array}{lcll} \mathbf{pos^t_x(t_i)} & = & \mathbf{pos^{t'}_{x'}(t'_i)} & (\star) \\ \llbracket t_i \rrbracket & = & \llbracket t'_i \rrbracket \text{ for } 1 \leq i \leq n & (\star\star) \end{array}$$

From $(\star\star)$ we get that $S(t_i) = S(t'_i)$. From $(\star)$ and the definition of $\mathbf{pos^t_x}$ we have that $i_1 = i'_1$ and in general for l such that $1 \leq l \leq n$:

$$i_l - S(t_1) - \cdots - S(t_{l-1}) + l - 1 = i_l - S(t'_1) - \cdots - S(t_{l-1'}) + l - 1$$

Since $S(t_l) = S(t'_l)$, we have that $i_l = i'_l$. We are done.
□

2.2.5 Strong Normalization for $\triangleright_{\mathbf{Eq_D}}$

(78) **Definition** (*Constant Free Linear Terms*)

We define the set $\mathbf{Lin}\text{-}T_{\Sigma_D}[X]$ to be the set of constant free linear terms. $\mathbf{Lin}\text{-}T_{\Sigma_D}[X]$ admits a definition by recursion as follows:

i) $x_{ij} \in X$ (for some $i, j \in \omega$) is a constant free linear term.

ii) If t_1, t_2 are constant free linear terms such that $Var(t_1) \cap Var(t_2) = \emptyset$, then $t_1 \oplus t_2$ is also a constant free linear term.

iii) If t_1, t_2 are constant free linear terms such that $Var(t_1) \cap Var(t_2) = \emptyset$, then $t_1 \otimes_i t_2$ $(1 \leq i \leq S(t_1))$ is also a constant free linear term.

If we have a finite set of variables $x_1, \cdots, x_n$, we define the set of constant free linear terms with variables ranging over the set $\{x_1, \cdots, x_n\}$. We denote it $\mathbf{Lin}\text{-}T_{\Sigma_D}[x_1, \cdots, x_n]$.

(79) **Lemma**
For every term $t \in \mathbf{Lin}\text{-}T_{\Sigma_D}[X]$, we have that:

$$S(t) \leq \sum_{x \in Var(t)} S(x)$$

Proof. We prove the lemma by induction on the structure of terms in $\mathbf{Lin}\text{-}T_{\Sigma_D}[X]$.

i) In the case of a variable x_{ij} $(i, j \in \omega)$, the result trivially holds, i.e. $S(x_{ij}) \leq \sum_{x \in Var(x_{ij})} S(x)$.

ii) Suppose $t = t_1 \oplus t_2$. Since $t \in \mathbf{Lin}\text{-}T_{\Sigma_D}[X]$, we have that $Var(t_1) \cap Var(t_2) = \emptyset$. Let the variables of t_1 be $\{x_1, \cdots, x_m\}$ $m \geq 1$. Similarly let the variables of t_2 be $\{x_{m+1}, \cdots, x_{m+n}\}$ with $n \geq 1$. Obviously $Var(t) = \{x_1, \cdots, x_{m+n}\}$. By induction hypothesis (i.h.) we have that:

$$S(t_1) \leq S(x_1) + \cdots + S(m) \text{ and}$$
$$S(t_2) \leq S(x_{m+1}) + \cdots + S(x_{m+n})$$

Since $S(t) = S(t_1 \oplus t_2) = S(t_1) + S(t_2)$, it follows that:

$$S(t) \leq S(x_1) + \cdots + S(m) + S(x_{m+1}) + \cdots + S(x_{m+n}) \quad = \quad \sum_{x \in Var(t)} S(x)$$

iii) Suppose $t = t_1 \otimes_i t_2$ $(1 \leq i \leq S(t_1))$. With the same notation for the variables of t_1 and t_2 as in ii), we have that by i.h.:

$$S(t) = S(t_1 \otimes_i t_2) = S(t_1) + S(t_2) - 1 \leq$$
$$S(x_1) + \cdots + S(m) + S(x_{m+1}) + \cdots + S(x_{m+n}) \quad = \quad \sum_{x \in Var(t)} S(x)$$

We are done.

□

(80) **Lemma**
For every term $t \in \mathbf{Lin}\text{-}T_{\Sigma_D}[X]$, every $\otimes_i$ term constructor (for some $i > 0$) is such that:

$$i \leq \sum_{x \in Var(t)} S(x)$$

Proof. By induction on the structure of terms of $\mathbf{Lin}\text{-}T_{\Sigma_D}[X]$.
Variables are a trivial case.
Suppose that $t = t_1 \oplus t_2$. The induction hypothesis (i.h.) holds of t_1 and t_2. Hence it holds also of t, for the term constructor is $\oplus$. Case where $t = t_1 \otimes_i t_2$. i.h. holds of both t_1 and t_2. It remains to check the result for the index i. Now, we know by the previous lemma that $S(t_1) \leq \sum_{x \in Var(t_1)} S(x)$. Hence:

$$i \leq \sum_{x \in Var(t_1)} S(x) < \sum_{x \in Var(t)} S(x)$$

We are done. □

(81) **Lemma** (*Finiteness of the Set* $\mathbf{Lin}\text{-}T_{\Sigma_D}[x_1, \cdots, x_n]$)

$\mathbf{Lin}\text{-}T_{\Sigma_D}[x_1, \cdots, x_n]$ is finite.

Proof. Let t be an arbitrary term of $\mathbf{Lin}\text{-}T_{\Sigma_D}[x_1, \cdots, x_n]$. Since the set of variables $\mathbf{Lin}\text{-}T_{\Sigma_D}[x_1, \cdots, x_n]$ is finite, we have that the height of the tree domain associated with t is at most n. For if this were not the case, since the tree associated with t is binary, then a path with length $h > n$ would force that t had more than n leaves, which is false since t has at most n leaves.
The number of term constructors could be a priori infinite since there exists an infinite number of discontinuous term constructors $\otimes_i$ with $i \in \omega$ and $i > 0$. But by the previous lemma, any t of $\mathbf{Lin}\text{-}T_{\Sigma_D}[x_1, \cdots, x_n]$ is such that i is bounded above by $M := \sum_{x \in Var(t)} S(x)$.
We have then a set of terms of bounded height and bounded number of term constructors. It is easy to see then that the set $\mathbf{Lin}\text{-}T_{\Sigma_D}[x_1, \cdots, x_n]$ is finite. This completes the proof. □

(82) **Lemma**
Let $t[(t_1 \otimes_i t_2) \otimes_j t_3]$ be an arbitrary term. Suppose there exists an **R4** redex of the form $(t_1 \otimes_i t_2) \otimes_j t_3$, i.e. we have that $t_3 \prec_{t_1} t_2$. Let $t'[(t_1 \otimes_k t_3) \otimes_l t_2]$ be such that $t[(t_1 \otimes_i t_2) \otimes_j t_3] \approx t'[(t_1 \otimes_k t_3) \otimes_l t_2]$. It follows that:

$t'[(t_1 \otimes_k t_3) \otimes_l t_2]$ cannot have an **R4** redex in $(t_1 \otimes_k t_3) \otimes_l t_2$

I.e. the following condition in $t'[(t_1 \otimes_k t_3) \otimes_l t_2]$ cannot hold:

$$t_2 \prec_{t_1} t_3$$

Proof. Without loss of generality we prove the lemma for the following term $t[(x \otimes_i y_2) \otimes_j y_3]$, where x, y_2, y_3 occur linearly in the term. Let us denote $[\![y_2]\!]$ and $[\![y_3]\!]$ by $\overrightarrow{a}$ and $\overrightarrow{b}$ respectively, i.e. $\overrightarrow{a} := [\![y_2]\!]$ and $\overrightarrow{b} := [\![y_3]\!]$.

In $t[(x \otimes_i y_2) \otimes_j y_3]$ we have by hypothesis that $y_3 \prec_x y_2$. Then it is easy to see that the image under $[\![\cdot]\!]$ of $t[(x \otimes_i y_2) \otimes_j y_3]$ is such that:

(83) $[\![t[(x \otimes_i y_2) \otimes_j y_3]]\!] = \alpha_0 + b^{S(y_3)} + \alpha_1 + a^0 + \alpha_2$

where $\alpha_0, \alpha_1, \alpha_2$ are strings of $(V_D)^*$. By the same reasoning, if $y_2 \prec_x y_3$ held in $t'[(x \otimes_k y_2) \otimes_l t_2]$ we would have:

(84) $[\![t[(x \otimes_i y_2) \otimes_j y_3]]\!] = \alpha'_0 + a^{S(y_2)} + \alpha'_1 + b^0 + \alpha'_2$

where $\alpha'_0, \alpha'_1, \alpha'_2$ are strings of $(V_D)^*$. Clearly, both (83) and (84) cannot hold at the same time. Contradiction. We are done. $\square$

(85) **Theorem** (*Strong Normalisation for* $\rhd_{\mathbf{Eq_D}}$)

The reduction relation $\rhd_{\mathbf{Eq_D}}$ is strongly normalizing, i.e. there is no infinite chains of reductions.

Proof. We prove the result by contradiction. Let us suppose that there exists an infinite sequence $(t_i)_{i \in \omega}$ such that the following holds:

$$t_i \rhd t_{i+1}, \ i \in \omega$$

There could be repetitions of terms in the sequence $(t_i)_{i \in \omega}$. Let us suppose that there does not exist such repetitions, i.e. for every $i, j \in \omega$, if $i \neq j$ then $t_i \neq t_j$. We prove that this is not possible. Suppose t_1 has n leaves with $n \geq 1$. Let $\widetilde{t_1} \in \mathbf{Lin}\text{-}T_{\Sigma_D}[x_1, \cdots, x_n]$ such that there exists a substitution σ satisfying $\sigma(\widetilde{t_1}) = t_1$. Since the $\mathbf{Eq_D}$ reductions are size-preserving there exists another term $\widetilde{t_2} \in \mathbf{Lin}\text{-}T_{\Sigma_D}[x_1, \cdots, x_n]$ such that $\sigma(\widetilde{t_2}) = t_2$. In general, for every i we obtain a sequence $(\widetilde{t_i})_{i \in \omega}$ with $\widetilde{t_i} \in \mathbf{Lin}\text{-}T_{\Sigma_D}[x_1, \cdots, x_n]$ and $\sigma(\widetilde{t_i}) = t_i$.

Now, the sequence $(\widetilde{t_i})_{i \in \omega}$ must have repetitions, say $\widetilde{t_i} = \widetilde{t_{i+l}}$ with $l > 0$, for the set $\mathbf{Lin}\text{-}T_{\Sigma_D}[x_1, \cdots, x_n]$ is finite by lemma (81). From $\widetilde{t_i} = \widetilde{t_{i+l}}$ we have that $\sigma(\widetilde{t_i}) = \sigma(\widetilde{t_{i+l}})$, whence $t_i = t_{i+l}$. Contradiction with the fact that the sequence $(t_i)_{i \in \omega}$ was supposed to have no repetitions.

It follows then that our infinite sequence $(t_i)_{i \in \omega}$ must have repetitions:

$t_i \rhd t_{i+1} \rhd \cdots \rhd t_{i+l-1} \rhd t_i$ with $i, l \in \omega$ such that $l \geq 2$ or $t_i \rhd t_i$

$(\star)$

Without loss of generality we suppose that for every k such that $1 \leq k \leq l-1$ there are no repetitions between t_i and t_{i+k}. We prove that this situation is again impossible by looking at the reduction cases:

- Suppose that in the chain of reductions $(\star)$, there is at least one **R1** reduction. Since a (continuous or discontinuous) unit is eliminated, it follows that a term constructor $\oplus$ or $\otimes_i$ (for some $i > 0$) has been removed. Hence the weight $|t|$ is decreased, where for any term s, $|s|$ is the number of connectives of s. As we already said before, the other reductions $\mathbf{R2} - \mathbf{R5}$ maintain constant the weight $|t|$. We conclude that this reduction chain cannot end with the first term of the chain t_i, since we would get $|t_i| < |t_i|$. Contradiction.

- Suppose now there is at least one **R3** reduction. We can assume there are no **R1** reductions, for in this case we could conclude that the chain of reductions $(\star)$ is impossible. Let us write t_i as:

$$t_i := t[(r_1 \oplus r_2) \otimes_i s]$$

Where r_1, r_2 and s are arbitrary terms and $i > 0$. t_i contains the **R3** redex $(r_1 \oplus r_2) \otimes_i s$. Without loss of generality we suppose that t_{i+1} is the result of reducing the $(r_1 \oplus r_2) \otimes_i s$, i.e.:

(86) $t_i \rhd t[(r_i \otimes_i s) \oplus r_2] =: t_{i+1}$

Where $1 \leq i \leq S(r_1)$.[8] All rules which can apply to (86), i.e. **R3**, **R4** and **R5** are such that in the contractum t_{i+1} no reduction is able to place s dominating $(r_1 \oplus r_2)$, and in t_i s dominates $(r_1 \oplus r_2)$. Hence, in a chain of reductions from t_{i+1}, no reduction will be such as to place s dominating both r_1 and r_2.

- Continuous associativity
We can suppose that there are no $\mathbf{R1}, \mathbf{R3}$ reductions. We define the following measure $|\cdot|_{R2}$ which is sensitive to continuous associativity:

$$\begin{array}{lcl} |0|_{R2} & = & 1 \\ |1|_{R2} & = & 1 \\ |x|_{R2} & = & 1 \text{ if } x \text{ is a variable} \\ |t_1 \oplus t_2|_{R2} & = & 2\,|t_1|_{R2} + |t_2|_{R2} \\ |t_1 \otimes_i t_2|_{R2} & = & |t_1|_{R5} + |t_2|_{R2} \end{array}$$

$|\cdot|_{R2}$ satisfies:

$$|t_1 \oplus (t_2 \oplus t_3)|_{R2} < |(t_1 \oplus t_2) \oplus t_3|_{R2}$$

$$\frac{|t_1|_{R2} < |t_1'|_{R2}}{|t_1' * t_2|_{R2} < |t_1 * t_2|_{R2}} \qquad \frac{|t_2|_{R2} < |t_2'|_{R2}}{|t_1 * t_2'|_{R2} < |t_1 * t_2|_{R2}} \quad \text{where } * \in \{\oplus\} \cup (\otimes_i)_{i>0}$$

If $t \rhd_{\mathbf{R}} t'$ where $\mathbf{R} \in \{\mathbf{R4}, \mathbf{R5}\}$
then $|t'|_{R2} = |t|_{R2}$

Now, if in $(\star)$ there is at least one **R2** redex, say $t_{i+k} \rhd_{R2} t_{i+k+1}$ with $0 < k \leq l-1$, then $|t_{i+k+1}|_{R2} < |t_{i+k}|_{R2}$ and $|t_{i+l}|_{R2} < |t_{i+k}|_{R2} \leq |t_i|_{R2}$. Hence we get the impossible $|t_i|_{R2} < |t_i|_{R2}$. Contradiction, whence there cannot exist such a chain $(\star)$ with at least one **R2** redex.

[8] The case where i is such that $S(r_1) < i \leq S(r_1) + S(r_2)$ is completely similar to the case $0 < i \leq S(r_1)$.

- We suppose now that in ($\star$), **R1-R3** do not apply. We only allow **R4-R5** reductions. Suppose we have the **R4** redex $(A \otimes_{j_k} s_k) \otimes_{j_{k+1}} s_{k+1}$ in the term:

$$t_i[(A \otimes_{j_k} s_k) \otimes_{j_{k+1}} s_{k+1}]$$

Where A is a product term. The side-condition of **R5** is the property $s_{k+1} \prec_A s_k$. Without loss of generality we assume that t_{i+1} is the **R4** contractum of t_i, i.e.:

$$t_i[(A \otimes_{j_k} s_k) \otimes_{j_{k+1}} s_{k+1}] \rhd t_i[(A \otimes_{j_{i+1}} s_{i+1}) \otimes_{j_i+S(t_{i+1})-1} s_i]$$

Now, since the reduction chain ($\star$) ends with the first term of the chain, i.e. t_i, it follows that in ($\star$) we must have the redex $(A' \otimes_{l_k} s_k) \otimes_{l_{k+1}} s_{k+1}$:

$$t_j[(A' \otimes_{l_k} s_{k+1}) \otimes_{l_{k+1}} s_k]$$

Where $i < j < i + l$ and the condition $s_k \prec_{A'} s_{k+1}$. This situation is impossible by lemma (82).

- Finally, suppose that no reductions from reduction rules **R1-R4** apply in the chain ($\star$). Therefore, in this case only reduction rule **R5** can apply. We define a measure on terms which is sensitive to discontinuous associativity:

$$\begin{array}{lcl}
|0|_{R5} & = & 1 \\
|1|_{R5} & = & 1 \\
|x|_{R5} & = & 1 \text{ if } x \text{ is a variable} \\
|t_1 \oplus t_2|_{R5} & = & |t_1|_{R5} + |t_2|_{R5} \\
|t_1 \otimes_i t_2|_{R5} & = & |t_1|_{R5} + |t_2|_{R5}, \text{ if } t_2 \text{ does not form an } \mathbf{R5} \text{ redex} \\
|(t_1 \otimes_i t_2) \otimes_j t_3|_{R5} & = & 2\,|t_1|_{R5} + |t_2 \otimes t_3|_{R5}, \text{ if } t_2 \between_{t_1} t_3
\end{array}$$

Similarly to the case of $|\cdot|_{R2}$, $|\cdot|_{R5}$ satisfies:

$$|t_1 \otimes_i (t_2 \otimes_{j-i+1} t_3)|_{R5} < |(t_1 \otimes_i t_2) \otimes_j t_3|_{R5}$$

$$\frac{|t_1|_{R5} < |t'_1|_{R5}}{|t'_1 * t_2|_{R5} < |t_1 * t_2|_{R5}} \qquad \frac{|t_2|_{R5} < |t'_2|_{R5}}{|t_1 * t'_2|_{R5} < |t_1 * t_2|_{R5}}$$

It follows that if there is at least one **R5** redex, then the measure $|\cdot|_{R5}$ is decreased. We obtain the contradictory inequality $|t_i|_{R5} < |t_i|_{R5}$.

Hence no cyclic reduction chain like ($\star$) is possible. Summing up, any chain of reductions must be terminating. This completes the proof.

□

Construction of an $\mathbf{Eq_D}$ normal form for a term t

We already know that every $T_{\mathbf{\Sigma_D}}[X]$ term has a $\mathbf{Eq_D}$ normal form[9] because strong normalization holds (theorem (85)). Here we give a particular strategy for obtaining the normal form. The construction of a normal form t^* for a given term t is done inductively. We compute the normal forms of every subterm and do some reductions if necessary. As the complexity of t is finite and the subterms we consider are strictly less complex than t, the computation of a normal form is terminating:

NF1 Render t unit-free.

NF2 If $t = 0$, then $t^* = 0$. If $t = 1$, then $t^* = 1$.

NF3 Else if $t = x_{ij}$ for a given variable $x_{ij} \in X, (i, j \in \omega)$, then $t^* = x_{ij}$.

NF4 Else if $t = s \oplus k$ and $s = s_1 \oplus s_2$, then do $R2$ reductions until either $t \rhd^* x_{ij} \oplus l$, where x_{ij} is a variable, or $t \rhd^* (l_1 \otimes_i l_2) \oplus l$, for $i > 0$. In case $t \rhd^* x_{ij} \oplus l$, $t^* = x_{ij} \oplus l^*$. In case $t \rhd^* (l_1 \otimes_i l_2) \oplus l$, $t^* = (l_1 \otimes_i l_2)^* \oplus l^*$.

NF5 Else $t = s \otimes_i k$, and s's main term constructor is $\oplus$. Then apply reduction $R3$ until $t \rhd^* (\cdots (x_{ij} \otimes_{i_1} s_1) \otimes_{i_2} s_2 \cdots) \otimes_{i_n} s_n$, for a given variable $x_{ij} \in X$. In that case, apply $R4$ until $t \rhd^* (\cdots (x_{ij} \otimes_{j_1} s_1) \otimes_{j_2} s_2 \cdots) \otimes_{j_m} s_m$, such that $m \leq S(x_{ij})$. Then $t^* = (\cdots (x_{ij} \otimes_{i_1} {s_1}^*) \otimes_{i_2} {s_2}^* \cdots) \otimes_{i_m} {s_m}^*$.

As expected, terms in normal form are unique. We need to prove some results:

(87) **Theorem** (*Equivalence theorem for $T_{\mathbf{\Sigma_D}}[X]$ Terms in Normal Form*)

Let t, s be terms in normal form. The following holds:

$$\text{If } [\![t]\!] = [\![s]\!] \text{ then } t = s$$

.

Proof. We proceed by induction on the structure of terms in normal form. More formally we want to prove that:

(88) $\forall t$ in normal form, if s is a term in normal form such that $[\![t]\!] = [\![s]\!]$, then $t = s$

- Base case:

  - $t = 0$. From lemma (65) if s is in normal form such that $[\![s]\!] = 0$ then $s = 0$. Therefore (88) holds for $t = 0$.

  - $t = 1$. By lemma (65) we get that if s is in normal form such that $[\![s]\!] = 1$ then $s = 1$. Therefore (88) holds for $t = 1$.

  - Finally, let $t = x_{ij}$ for some $i, j \in \omega$. We have that $[\![x_{ij}]\!] = \overrightarrow{a_{ij}}$. Again by lemma (65) we have that if $[\![s]\!] = \overrightarrow{a_{ij}}$ then $s = x_{ij}$.

[9]Later we see the unicity of normal forms.

- Inductive case: suppose $t = t_1 \oplus t_2$. Let s be a term in normal form such that $\llbracket s \rrbracket = \llbracket t \rrbracket$.

  Clearly, by vii) from lemma (65), since $l(\llbracket t_1 \oplus t_2 \rrbracket) \geq 2$ s cannot be 0 nor 1 nor x_{0j} (a variable of sort 0). Let now s be a variable of sort strictly greater that 0, i.e. $s = x_{ij}$ with $i > 0$. $\llbracket x_{ij} \rrbracket = \overrightarrow{a_{ij}}$. Let us see that s cannot be equal to $t_1 \oplus t_2$. If $\llbracket \overrightarrow{a_{ij}} \rrbracket \neq \llbracket t_1 \rrbracket$, either $\llbracket \overrightarrow{a_{ij}} \rrbracket$ is a proper prefix of $\llbracket t_1 \rrbracket$ or conversely. This is not possible because by lemma (65) both $\llbracket t_1 \rrbracket$ and $\llbracket x_{ij} \rrbracket$ are prefix-free. Hence, we have that $\llbracket t_1 \rrbracket = \llbracket x_{ij} \rrbracket$. We get a contradiction, for in that case $l(\llbracket x_{ij} \rrbracket) = l(\llbracket t_1 \rrbracket)$ and $l(\llbracket t_1 \rrbracket) < l(\llbracket t_1 \oplus t_2 \rrbracket)$.

  Suppose now $s = s_1 \otimes_i s_2$ for some terms s_1, s_2 and index $i > 0$. By lemma (65) s is a product term. A product term is prefix-free. By the same mentioned lemma we know that t_1 is also prefix-free. Comparing $\llbracket t_1 \rrbracket$ and $\llbracket s \rrbracket$ we have that either t_1 is a proper prefix of s or conversely. In either case we have prefix-free terms which contain a prefix. Contradiction. Hence $t_1 = s$, whence we get again a contradiction for $l(\llbracket t \rrbracket) > l(\llbracket t_1 \rrbracket) = l(\llbracket s \rrbracket)$.

  It follows then that s has to be equal to $s_1 \oplus s_2$ for some terms s_1, s_2. We know by lemma (65) that t_1, s_1 are prefix-free. Hence $\llbracket t_1 \rrbracket = \llbracket s_1 \rrbracket$. We have t_1 is a proper subterm of t, and therefore we can apply the induction hypothesis (i.h.), i.e. since t_1 and s_1 are in normal form and $\llbracket t_1 \rrbracket = \llbracket s_1 \rrbracket$ we have that $t_1 = s_1$. We have also $\llbracket t_2 \rrbracket = \llbracket s_2 \rrbracket$. We apply i.h. and get that $t_2 = s_2$. This completes the case $t = t_1 \oplus t_2$.

- Finally, suppose $s = s_1 \otimes_i s_2$ for some terms s_1, s_2 and index i. We have two product terms t and s such that $\llbracket t \rrbracket = \llbracket s \rrbracket$. We can apply lemma (77), and we get that t and s must satisfy:

$$\begin{array}{lcl} t & = & (\cdots((x_{nj} \otimes_{i_1} t_1)x \otimes_{i_2} t_2) \cdots \otimes_{i_{n-1}} t_{n-1}) \otimes_{i_n} t_n \\ s & = & (\cdots((x_{nj} \otimes_{i_1} t'_1)x \otimes_{i_2} t'_2) \cdots \otimes_{i_{n-1}} t'_{n-1}) \otimes_{i_n} t'_n \\ \llbracket t_i \rrbracket & = & \llbracket t'_i \rrbracket \text{ for } i \text{ such that } 1 \leq i \leq n \end{array}$$

  For $i = 1, \cdots, n$ we have terms in normal form t_i which are proper subterms of t and subterms s_i in normal form such that $\llbracket t_i \rrbracket = \llbracket t'_i \rrbracket$. We can apply i.h. It follows that for every $i = 1, \cdots, n$ that $t_i = t'_i$. It follows that $t = s$. This completes the last inductive case. We are done.

□

As the last results suggest, terms in $\mathbf{Eq_D}$ form are unique:

(89) **Theorem** ($\mathbf{Eq_D}$ *Normal form unicity for* $T_{\mathbf{\Sigma_D}}[X]$)

Let t^*, t^{**} be two normal forms for the term t. It follows that:

$$t^* = t^{**}$$

Proof. We have that:

$t \approx t^*$
$t \approx t^{**}$

It follows that $t^* \approx t^{**}$. By soundness, $[\![t^*]\!] = [\![t^{**}]\!]$. By the equivalence theorem $t^* = t^{**}$. We are done. $\Box$

(90) **Theorem** (*General Equivalence Theorem for* $T_{\mathbf{\Sigma_D}}[X]$)

Let t and s be arbitrary $T_{\mathbf{\Sigma_D}}[X]$ terms not necessarily in normal form. The following holds:

$$\text{If } [\![t]\!] = [\![s]\!] \text{ then } t \approx s$$

Proof. Let t^* and s^* be the normal form terms for t and s. We have $t \approx t^*$ and $s \approx s^*$. Therefore:

$$[\![t^*]\!] = [\![t]\!] = [\![s]\!] = [\![s^*]\!]$$

Hence by the general equivalence theorem (90) for $T_{\mathbf{\Sigma_D}}[X]$ terms in normal form there holds $t^* = s^*$. We have $t \approx t^* = s^* \approx s$, whence by transitivity of $\approx$, $t \approx s$. We are done. $\Box$

(91) **Theorem** (*Completeness of* $\mathbf{Eq_D}$ *and* $\mathbf{Eq_{D}}_2$ *w.r.t.* **FreeDisp**)

Let $t, s \in T_{\mathbf{\Sigma_D}}[X]$. Both $\mathbf{Eq_D}$ and $\mathbf{Eq_{D}}_2$ are complete w.r.t. **FreeDisp**:

If $\vdash_{\mathbf{Eq_D}} t \approx s$ then $\mathbf{FreeDisp} \models t \approx s$
If $\vdash_{\mathbf{Eq_{D_2}}} t \approx s$ then $\mathbf{FreeDisp} \models t \approx s$

Proof. We already have seen soundness. Suppose $\mathbf{FreeDisp} \models t \approx s$. It follows that:

$$\vdash_{\mathbf{Eq_D}} t \approx s \text{ as well as } \vdash_{\mathbf{Eq_{D_2}}} t \approx s$$

For since $\mathbf{FreeDisp} \models t \approx s$, soundness implies that $[\![t]\!] = [\![s]\!]$. By the general theorem (90) for $T_{\mathbf{\Sigma_D}}[X]$ terms we have that $t \approx s$. This completes the proof. $\Box$

2.3 Visibility for extraction in Σ_D-Terms in the theory $\mathbf{Eq_D}$

For commodity we will write x instead of x_{ij}, as well as $\overrightarrow{a}$ instead of $\overrightarrow{a_{ij}}$. In the following $t[x]$ is a Σ_D-term where x is a variable which occurs linearly in $t[x]$. The results we will prove easily extend to the general case with the concept of distinguished occurrence. We say that an x's occurrence in $t[x]$ is *wrap-free* if and only if in the normal form $t^*[x]$ of $t[x]$, x's occurrence is not a *foot*, i.e. it is not a left child of a $\otimes_i$ term constructor.

(92) **Definition** (*Wrap-Free*)

Let $t^*[x]$ be the normal form of $t[x]$. x occurs *wrap-free* in $t[x]$ iff x does not occur as a foot in $t^*[x]$.

If a term $t^*[x]$ is in normal form such that x's occurrence is not a foot, then no term $s[x]$ could be derived from $t^*[x]$ in such a way that $t^*[x] \approx s[x]$ and x is a foot in $s[x]$, i.e. not being a foot in a term in normal form is an invariant w.r.t. the application of the axioms of $\mathbf{Eq_D}$ or $\mathbf{Eq_{D2}}$.

One point which will be crucial on our road to discontinuity is the notion of visibility for extraction:

(93) **Definition** (*Visibility for Extraction*)

x is visible for extraction in $t[x]$ if and only if $t[x] \approx s \otimes_k x$ for some $k > 0$ and some term s.

(94) **Theorem** (*Visibility for Extraction*)

Let $t[x] \in T_{\mathbf{\Sigma_D}}[X]$. The following holds:

x is visible for extraction in $t[x]$ iff $[\![t[x]]\!] = [\![t']\!] \times_k [\![x]\!]$, for some $k > 0$

and $t' \in T_{\mathbf{\Sigma_D}}[X]$

Proof.

- *Only if* case. Obvious.
- *If case.* Suppose we have that:

$$[\![t[x]]\!] = [\![t']\!] \times_k [\![x]\!], \text{ for some } k > 0$$

By the general equivalence theorem (90) for $T_{\mathbf{\Sigma_D}}[X]$, it follows that:

$$t[x] \approx t' \times_k x$$

We are done.

□

In other words, visibility for extraction of a single occurrence of x in a term, means that the x occurrence can be *displaced* to the right periphery of the original term while maintaining $\mathbf{Eq_D}$-equivalence. Theorem (94) will turn out to be fundamental for the main results of the next chapter. We should point out that the proof of the theorem is not constructive for we only use the power of the general theorem of equivalence of $T_{\mathbf{\Sigma_D}}[X]$ terms (theorem (90)). In the following result, we see that *visibility for extraction* is strongly related to the concept of *wrap-free*.

(95) **Theorem** (*Visibility for Extraction and the Wrap-Free Concept*)

x is visible for extraction in $t[x]$ iff x occurs wrap-free in $t[x]$.

Proof.

- Only if direction:

  Suppose x is visible for extraction, i.e., $t[x] \approx s \otimes_k x$ for some term s and $k > 0$. Then we have:

(96) $[\![t[x]]\!] \quad = \quad [\![s \otimes_k x]\!] \quad = \quad [\![s]\!] \times_k [\![x]\!]$

- If direction:

Let $t^*[x]$ be the normal form. We reason by induction on the structural complexity of the term $t^*[x]$.

In the following cases we will use the fact that $t[x] = t^*[x]$, and therefore if x is visible for extraction in $t^*[x]$ then x is also visible for extraction in $t[x]$. The cases are as follows:

i) $t^*[x] = x$.

We put $s = 1$ and hence :

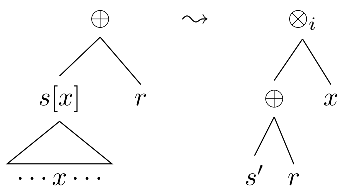
$$t^*[x] \approx 1 \otimes_1 x$$

ii) $t^*[x] = s[x] \oplus r$.

By induction hypothesis (i.h.), $s[x] \approx s' \otimes_k x$ for some $k > 0$. We have the following equational derivation:

$$\begin{array}{lll} t[x] & \approx & (s' \otimes_k x) \oplus r \\ & \approx & (1 \oplus r) \otimes_1 (s' \otimes_k x) \quad \text{by } \mathbf{SW} \\ & \approx & ((1 \oplus r) \otimes_1 s') \otimes_k x) \quad \text{by } \mathbf{Assc_d} \\ & \approx & (s' \oplus r) \otimes_k x \quad \text{by } \mathbf{SW} \end{array}$$

In tree format:

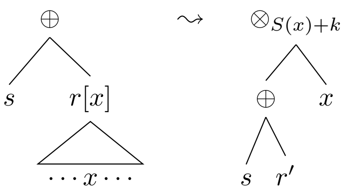


iii) $t[x] = s \oplus r[x]$

By i.h. $r[x] \approx r' \otimes_k x$ for some term r' and $k > 0$. It follows that:

$$\begin{array}{lll} t[x] & \approx & s \oplus (r' \otimes_k x) \\ & \approx & (s \oplus 1) \otimes_{S(s)+1} (r' \otimes_k x) \quad \text{by } \mathbf{SW} \\ & \approx & ((s \oplus 1) \otimes_{S(x)+1} r') \otimes_{S(s)+k} x \quad \text{by } \mathbf{Assc_d} \\ & \approx & (s \oplus r') \otimes_{S(s)+k} x \quad \text{by } \mathbf{SW} \end{array}$$

In tree format:



iv) $t[x] = s[x] \otimes_i r$ for some term $s[x]$, r and $i > 0$.

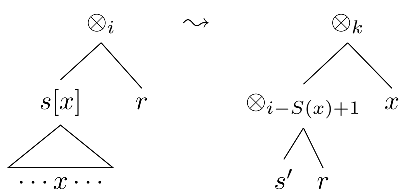
By i.h. $s[x] \approx s' \otimes_k x$ for some s' and $i > 0$. We derive the following equation:

$$t[x] \approx (s' \otimes_k x) \otimes_i r$$

Now x must permute with r in s', i.e. $x \prec_{s'} r$ or $r \prec_{s'} x$, for otherwise $x \between_{s'} r$, and then discontinuous associativity would apply with the consequence that x would be a foot, which is not possible if $t[x]$ is in normal form and x's occurrence in $t[x]$ is not a foot. Without loss of generality, let us suppose that $x \prec_{s'} r$. In that case we have:

$$t[x] \approx (s' \otimes_{i-S(x)+1} r) \otimes_k x \quad \text{by } \mathbf{MixPerm1}$$

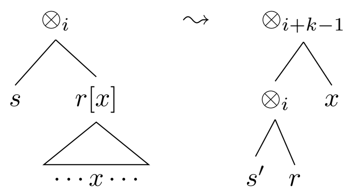
Hence x is permutated to right periphery in $t[x]$. In tree-format:



v) $t[x] = s \otimes_i r[x]$ for some terms s and $r[x]$ and $i > 0$. By i.h. $r[x] \approx r' \otimes_k x$. Then:

$$\begin{aligned} t[x] &\approx s \otimes_i (r' \otimes_k x) \\ &\approx (s \otimes_i r') \otimes_{i+k-1} x \quad \text{by } \mathbf{Assc_d} \end{aligned}$$

In tree format:



In either case the variable x can be displaced to the right periphery. This completes the proof.

□

A remark should be made. This theorem gives an *effective procedure* for deriving the term which has x's occurrence displaced at the right periphery. By applying the axioms of $\mathbf{Eq_D}$ (or of course $\mathbf{Eq_{D_2}}$) a finite number of times we get the extraction. It should be noted that all equations of $\mathbf{Eq_D}$ except continuous associativity have turn out to be necessary for the extraction. This of course raises the question of considering a weaker version of the equational theory $\mathbf{Eq_D}$ in which continuous associativity is dropped.

(97) **Corollary** (*Property for the Wrap-Free Concept*)

Let $t[x]$ be a term. The following holds:

x occurs wrap-free in $t[x]$ iff $\llbracket t[x] \rrbracket = \llbracket t' \rrbracket \times_k \llbracket x \rrbracket$, for some $k > 0$ and $t' \in T_{\mathbf{\Sigma_D}}[X]$

Proof. By theorems (94) and (95) we get the desired result. $\square$

So if x occurs wrap-free in $t[x]$ then we have:

$$t[x] \approx s \otimes_i x$$

We may ask ourselves whether the index involved in the extraction is unique. A similar question arises for the uniqueness of the term s. The answer is that effectively the index is unique and the mentioned term s is unique modulo $\approx_{\mathbf{Eq_D}}$.

(98) **Theorem** (*Uniqueness of Extractability*)

Suppose $t[x] \approx s \otimes_i x$ and $t[x] \approx s' \otimes_j x$. Then:

i) $i = j$

ii) $s \approx s'$

Proof.

- i) We proceed by contradiction. Suppose $i \neq j$. By assumption we have:

$$t[x] \approx s \otimes_i x \quad \& \quad t[x] \approx s' \otimes_j x$$

Therefore, in any displacement model $(\mathcal{A}, v)$ we have:

$$\llbracket s \otimes_i x \rrbracket_v = \llbracket s' \otimes_j x \rrbracket_v$$

Consider the following displacement model:

$$(\mathcal{A}, v),$$

where $\mathcal{A}$ is the finitely free generated monoid $\{a, 1\}^*$ and v is a valuation. As expected 1 is the prime. Let $\llbracket \cdot \rrbracket_v$ be the interpretation in $\mathcal{A}$ w.r.t. the valuation v. We put:

$$\begin{array}{lcl} v(x) & = & \underbrace{a + 1 + a + \cdots + a + 1 + a =}_{S(x) \text{ separators}} : A \\ v(y) & = & \underbrace{1 + \cdots + 1}_{S(y) \text{ separators}} \quad \text{for any variable } x \text{ of arbitrary sort such that } x \neq y \end{array}$$

It follows then that since s and s' do not contain the variable x,[10] their interpretations $\llbracket s \rrbracket_v$ and $\llbracket s' \rrbracket_v$ as strings cannot contain a's. They are equal then to the empty string or a string of 1's.

[10] Remember that the term $t[x]$ is linear and therefore x occurs in it only once.

$$\begin{aligned}
[\![S \otimes_i x]\!]_v &= \underbrace{1 + \cdots + 1}_{i-1\,\text{separators}} + A + \alpha_i + 1 + \cdots + 1 + \alpha_s \\
[\![S' \otimes_j x]\!]_v &= \underbrace{1 + \cdots + 1}_{j-1\,\text{separators}} + A + \beta_j + 1 + \cdots + 1 + \beta_s
\end{aligned}$$

Since $\mathcal{A}$ is a (finitely) free generated monoid, we cannot have $i-1 \neq j-1$ for then it would follow that a (finite) summation of 1's would be equal to a, which of course is false in a displacement model. Therefore $i-1 = j-1$ whence $i = j$.

- ii) By the first part of this theorem i), we know that the indexes of the discontinuous products in the extraction must be identical, i.e., we have now:

$$s \otimes_i x \approx s' \otimes_i x$$

We want to prove in the theory of $\mathbf{Eq_D}$ that $s \approx s'$. It must be signalled that the result for x of sort 1 is easily obtained:

$$\frac{s \otimes_i x \approx s' \otimes_i x}{s \otimes_i 1 \approx s' \otimes_i 1}\ \mathbf{Subst}\,[x/1]$$

By applying twice the equational rule **Trans** and the fact that 1 is a unit, i.e. $s \otimes_i 1 \approx s$ and $s' \otimes_i 1 \approx s'$, we get $s \approx s'$. We now prove the result for a variable x of arbitrary sort. $x \in X$ and hence there exist indices $l, k \in \omega$ such that $x = x_{lk}$. We have then that $\mathbf{S}(x_{lk}) = l$.

Since $s \otimes_i x \approx s' \otimes_i x$ we have that the sort of s and s' are equal, i.e. $S(s) = S(s') =: m$. Applying the $\lceil\!\lceil\cdot\rceil\!\rceil$ homomorphism to s and s' we have:

$$\begin{aligned}
\lceil\!\lceil s\rceil\!\rceil &= \alpha_0 + 1 + \alpha_1 + 1 + \cdots + \alpha_{i-1} + 1 + \alpha_i + \cdots + 1 + \alpha_m \\
\lceil\!\lceil s'\rceil\!\rceil &= \beta_0 + 1 + \beta_1 + 1 + \cdots + \beta_{i-1} + 1 + \beta_i + \cdots + 1 + \beta_m
\end{aligned}$$

For every i, we have that $\alpha_i, \beta_i \in (V_D)^* - \{a^0_{lk}, \cdots, a^l_{lk}\}^*$ (l is the sort of x_{lk}!). We have $\lceil\!\lceil x\rceil\!\rceil = \overrightarrow{a_{lk}}$:

$$\overrightarrow{a_{lk}} = a^0_{lk} + 1 + a^1_{lk} + \cdots + a^{l-1}_{lk} + 1 + a^l_{lk}$$

Since $s \otimes_i x \approx s' \otimes_i x$, we have that $\lceil\!\lceil s \otimes_i x\rceil\!\rceil = \lceil\!\lceil s' \otimes_i x\rceil\!\rceil$.

(99)
$$\lceil\!\lceil s \otimes_i x\rceil\!\rceil = \alpha_0 + 1 + \alpha_1 + 1 + \cdots + \alpha_{i-1} + \overrightarrow{a_{lk}} + \alpha_i + \cdots + 1 + \alpha_m$$

(100)
$$\lceil\!\lceil s' \otimes_i x\rceil\!\rceil = \beta_0 + 1 + \beta_1 + 1 + \cdots + \beta_{i-1} + \overrightarrow{a_{lk}} + \beta_i + \cdots + 1 + \beta_m$$

We now equate (99) and (100):

$$\alpha_0 + 1 + \alpha_1 + 1 + \cdots + \alpha_{i-2} + 1 + \alpha_{i-1} + \overrightarrow{a_{lk}} + \alpha_i + \cdots + 1 + \alpha_m$$
$$=$$
$$\beta_0 + 1 + \beta_1 + 1 + \cdots + \beta_{i-2} + 1 + \beta_{i-1} + \overrightarrow{a_{lk}} + \beta_i + \cdots + 1 + \beta_m$$

Since for every i, α_i and β_i do not contain as substrings elements from $\{a^0_{lk}, \cdots, a^l_{lk}\}$ we have the following equalities:

(101)
$$\begin{array}{l} \alpha_0 + 1 + \alpha_1 + 1 + \cdots + \alpha_{i-2} + 1 + \alpha_{i-1} = \\ \beta_0 + 1 + \beta_1 + 1 + \cdots + \beta_{i-2} + 1 + \beta_{i-1} \\ \\ \alpha_i + \cdots + 1 + \alpha_s = \beta_i + \cdots + 1 + \beta_s \end{array}$$

From the above equalities (101), since the α_i's and β_i's do not contain the separator 1 we infer that

$$\forall i = 0, \cdots, s,\ \alpha_i = \beta_i$$

Hence $[\![s]\!] = [\![s']\!]$. By the equivalence theorem (90) we have then:

$$s \approx s'$$

which gives our desired result for ii). This completes the proof.

□

2.4 Appendix

We have defined syntactical algebras as free monoids with a distinguished element denoted 1 called the separator. From syntactical algebras we obtain the standard displacement algebras which form the class of algebras **FreeDisp**. We have seen that all standard displacement algebras satisfy the so-called separation property (54). Here we show two examples of general displacement algebras which do not belong to the class **FreeDisp**.

- First example:

  We present a general displacement algebra which satisfies the separation property but its universe is not a free monoid and in consequence it is not defined by a syntactical algebra. Let $\mathcal{M} = \langle M, +, 0\rangle$ be an arbitrary monoid which is not free, with for example nilpotent elements.[11] We build the following new monoid we denote $\mathcal{M}^\infty$ which in fact is the least monoid containing $\mathcal{M}$ and an distinguished element which plays the role of separator in the sense of syntactical algebras. Such an element satisfies the separation property. We define $\mathcal{M}^\infty = \langle M^\infty, +, 0, 1\rangle$ as follows:

  (102) $$M^\infty \overset{def}{=} \bigcup_{i \in \omega} M^{i+1}$$

  We define the separator in $\mathcal{M}^\infty$ as follows:

[11] We recall that a nilpotent element a of a monoid is such that $a^n = 0$ for some $n > 1$.

(103) $1^{\mathcal{M}^\infty} \stackrel{def}{=} (0,0)$

We denote it 1^∞. Obviously, $0^{\mathcal{M}^\infty} = 0^{\mathcal{M}}$, which we denote 0^∞. The sort domains of sort i are defined as follows:

(104) $L_i \stackrel{def}{=} M^{i+1}, i \in \omega$

Every element of L_i is said to be of sort i. We denote $\overrightarrow{x} \in L_i$ as $\overrightarrow{x} = (x_0, \cdots, x_i)$. We can now define the operations $\oplus^{\mathcal{M}^\infty}$ and $\{\otimes_{i+1}\}_{i\in\omega}^{\mathcal{M}^\infty}$. We denote these operations simply as $+^\infty$ and $\{\times_{i+1}^\infty\}_{i\in\omega}$. Their definition is sort-polymorphic:

(105) For every $i, j \in \omega$,

$$\begin{array}{rcl} +^\infty : L_i \times L_j & \longrightarrow & L^{i+j} \\ (\overrightarrow{x}, \overrightarrow{y}) & \mapsto & (x_0, \cdots, x_i + y_0, y_1, \cdots, y_j) \end{array}$$

(106) For every $i, j \in \omega$ and $1 \leq k \leq i$,

$$\begin{array}{rcl} \times_k^\infty : L_i \times L_j & \longrightarrow & L^{i+j-1} \\ (\overrightarrow{x}, \overrightarrow{y}) & \mapsto & (x_0, \cdots, x_{k-1} + y_0, y_1, \cdots, y_j + x_k, x_{k+1}, \cdots, x_i) \end{array}$$

Now, it is clear that for every $\overrightarrow{x} \in L_i$ there exists a list of elements of L_0 $\langle x_0, \cdots, x_i\rangle$ such that:

(107) $\overrightarrow{x} = x_0 +^\infty +1^\infty +^\infty x_1 +^\infty 1^\infty +^\infty \cdots +^\infty x_{i-1} +^\infty 1^\infty + x_i$

This means that $\mathcal{M}^\infty$ satisfies the first condition of the separation property. The second condition of the separation property is also satisfied:

(108) If

$\overrightarrow{x} = x_0 +^\infty +1^\infty +^\infty x_1 +^\infty 1^\infty +^\infty \cdots +^\infty x_{i-1} +^\infty 1^\infty +^\infty x_i = \overrightarrow{y} = y_0 +^\infty +^\infty 1^\infty +^\infty y_1 +^\infty 1^\infty +^\infty \cdots +^\infty y_{i-1} +^\infty 1^\infty +^\infty y_i$

then $\forall i,\ x_i = y_i$

For such elements $\overrightarrow{x}, \overrightarrow{y} \in M^{i+1}$, and hence by definition of M^{i+1} two tuples are equal iff they are equal point-wise.

- Second example: Let us consider the following Σ_D-algebra $\mathcal{D} = \langle \mathrm{Im}([\![\cdot]\!]), +, (\times_{i+1})_{i\in\omega}, \Lambda, 1)\rangle$. The sort domains $(L_i^*)_{i\in\omega}$ of $\mathcal{D}$ are the following:

  (109) $L_i^* = [\![(T_{\Sigma_D}[X])_i]\!],\ i \in \omega$

  Since $\mathcal{D}$ is a subalgebra of the syntactical algebra generated by $V_D \cup \{1\}$, and from the fact that $[\![\cdot]\!]$ is a homomorphism of Σ_D-algebras (in fact of (standard) displacement algebras), we see that $\mathcal{D}$ is a general displacement algebra. But $\mathcal{D}$ is not a standard displacement algebra, for the first condition of the separation property is not satisfied. For example, $[\![(x_{20} \otimes_1 1) \otimes_1 1]\!] = a_{20}^0 + 1 + a_{20}^1 + 1 + a_{20}^2$, and no a_{20}^i $(i = 0, 1, 2)$ belongs to $L_0^* = [\![(T_{\Sigma_D}[X])_0]\!]$. The second condition of the separation property is trivially satisfied.

Therefore we see that **FreeDisp** $\subsetneq$ **Disp**.

Chapter 3

Proof Theory

In Chapter 2 'Model-Theoretical Foundations of Discontinuity' we did the hard work of defining a class of sorted algebras, namely the class of displacement algebras **FreeDisp**, and of formulating a sound and complete equational theory $\mathbf{Eq_D}$. This class has as we demonstrated good properties from the point of view of Type Logical Grammar, namely generalized extraction, not only peripheral extraction. This last algebraic limitation was what constituted **L** rigid with respect grammars of natural languages. Generalized extraction is the key in our type-logical agenda to define a calculus which subsumes **L** and is able to deal with discontinuous phenomena which are so wide-spread in natural language.

The road map of this Chapter is the following:

- To formulate the set of types which will be interpreted in displacement algebras.
- To formulate the categorical Discontinuous Lambek (Syntactic) Calculus, **cD**.
- To formulate for **cD** its corresponding sorted multimodal calculus **mD**.
- To export to **mD** from Chapter 2 the good formal properties which will be crucial to prove some faithful embedding translations. The details will be expounded carefully.
- To absorb à la Lambek (1958) the structural postulates in a new Gentzen sequent syntax: the Hypersequent Calculus **hD** (not to be confused with Avron's hypersequent calculus, Avron (1991)). The rise of hypersequent syntax needs the definition of new data-structures in the corresponding Gentzen calculi. We will define therefore what we call the set of configurations, now the set of hyperconfigurations and hypercontexts. As we will see the prefix *hyper* will be ubiquitous.
- To define faithful embedding translations between all the calculi defined.
- To prove the Cut Elimination theorem for **hD** using crucially the very easy hypersequent syntax which will allow us to make a proof of the Cut Elimination theorem à la Lambek.

- To extend **D** with the so-called linear logic additive connectives in order to give much more logical power to the system.
- To define very useful synthetic connectives (see 'Point Aveugle' Girard (2006)) and prove more embedding translations.

3.1 Theory of Discontinuous Lambek Calculus

The key to our treatment of discontinuity is the class **FreeDisp** of displacement algebras. We recall some definitions:

(110) **Definition** (*Syntactical Algebra*)

A *syntactical algebra* is a free algebra $(L, +, 0, 1)$ of arity $(2, 0, 0)$ such that $(L, +, 0)$ is a monoid and 1 is a prime. I.e. L is a set, $0 \in L$ and $+$ is a binary operation on L such that for all $s_1, s_2, s_3, s \in L$,

$$\begin{array}{rcll} s_1+(s_2+s_3) & = & (s_1+s_2)+s_3 & \text{associativity} \\ 0+s = & s & = s+0 & \text{identity} \end{array}$$

The distinguished constant 1 is called a *separator*.

(111) **Definition** (*Sorts*)

The *sorts* of discontinuous Lambek calculus are the naturals $0, 1, \ldots$. The sort $S(s)$ of an element s of a syntactical algebra $(L, +, 0, 1)$ is defined by the morphism of monoides S to the additive monoid of naturals defined thus:

$$\begin{array}{rcll} S(1) & = & 1 & \\ S(a) & = & 0 & \text{for a prime } a \neq 1 \\ S(s_1+s_2) & = & S(s_1) + S(s_2) & \end{array}$$

I.e. the sort of a syntactical element is simply the number of separators it contains; we require the separator 1 to be a prime and the syntactical algebra to be free in order to ensure that this induction is well-defined.

(112) **Definition** (*Sort Domains*)

Where $(L, +, 0, 1)$ is a syntactical algebra, the *sort domains* L_i of sort i of generalized discontinuous Lambek calculus are defined as follows:

$$L_i \quad = \quad \{s | S(s) = i\}, i \geq 0$$

(113) **Definition** (*Displacement Algebra*)

The *displacement algebra* defined by a syntactical algebra $(L, +, 0, 1)$ is the ω-sorted algebra with the ω-sorted signature $\Sigma_D = (\oplus, \{\otimes_{i+1}\}_{i\in\omega}, 0, 1, \omega, \Omega)$ defined in Chapter 2:

$$(\{L_i\}_{i\in\omega}, +, \{\times_{i+1}\}_{i\in\omega}, 0, 1)$$

$\mathcal{F}_i$	$::=$	$\mathcal{A}_i$	where $\mathcal{A}_i$ is the set of atomic types of sort i
$\mathcal{F}_0$	$::=$	I	Continuous unit
$\mathcal{F}_1$	$::=$	J	Discontinuous unit
$\mathcal{F}_{i+j}$	$::=$	$\mathcal{F}_i \bullet \mathcal{F}_j$	continuous product
$\mathcal{F}_j$	$::=$	$\mathcal{F}_i \backslash \mathcal{F}_{i+j}$	continuous under
$\mathcal{F}_i$	$::=$	$\mathcal{F}_{i+j} / \mathcal{F}_j$	continuous over
$\mathcal{F}_{i+j}$	$::=$	$\mathcal{F}_{i+1} \odot_k \mathcal{F}_j$	discontinuous product
$\mathcal{F}_j$	$::=$	$\mathcal{F}_{i+1} \downarrow_k \mathcal{F}_{i+j}$	discontinuous extract
$\mathcal{F}_{i+1}$	$::=$	$\mathcal{F}_{i+j} \uparrow_k \mathcal{F}_j$	discontinuous infix

Figure 3.1: The sorted types of **D**

where:

operation	is such that
$+ : L_i \times L_j \to L_{i+j}$	as in the syntactical algebra
$\times_k : L_{i+1} \times L_j \to L_{i+j}$	$\times_k(s, t)$ is the result of replacing the k-th separator in s by t

The sorted types of the discontinuous Lambek Calculus, **D**, which we will define residuating with respect to the sorted operations in (113), are defined by mutual recursion in Figure 3.1. **D** types are to be interpreted as subsets of L and satisfy what we call the *principle of well-sorted inhabitation*:

(114) **Principle of well-sorted inhabitation:**
If A is a type of sort i, $[\![A]\!] \subseteq L_i$

I.e. every syntactical inhabitant of $[\![A]\!]$ has the same sort. The connectives and their syntactical interpretations are shown in Figures 3.1 and 3.2. This syntactical interpretation is called the *standard syntactical interpretation*. Given the functionalities of the operations with respect to which the connectives are defined, the grammar defining by mutual recursion the sets $\mathcal{F}_i$ of types of sort i for each sort i on the basis of sets $\mathcal{A}_i$ of primitive types of sort i for each sort i, and the homomorphic *syntactical sort map* S sending types to their sorts, are as shown in Figure 3.3. When A is an arbitrary type, we will frequently write in latin lower-case the type in order to refer to its sort $S(A)$, i.e.:

$$a \stackrel{def}{=} S(A)$$

We observe that for any type $A \in \mathcal{F}$, the interpretation of A, i.e. $[\![A]\!]$ is contained in L_a.

The syntactical sort map is to syntax what the semantic type map is to semantics: both homomorphisms mapping syntactic types to the datatypes of the respective components of their inhabiting signs in the dimensions of language in extension: form/signifier and meaning/signified.

$$
\begin{array}{rcll}
[\![I]\!] & = & \{0\} & \text{continuous unit} \\
[\![J]\!] & = & \{1\} & \text{discontinuous unit} \\
[\![A]\!] & \subseteq & L_i \text{ for some } i \in \omega & A \in \mathcal{A}_i \\
\\
[\![A{\bullet}B]\!] & = & \{s_1{+}s_2 |\ s_1 \in [\![A]\!] \ \&\ s_2 \in [\![B]\!]\} & \text{(continuous) product} \\
[\![A\backslash C]\!] & = & \{s_2 |\ \forall s_1 \in [\![A]\!], s_1{+}s_2 \in [\![C]\!]\} & \text{under} \\
[\![C/B]\!] & = & \{s_1 |\ \forall s_2 \in [\![B]\!], s_1{+}s_2 \in [\![C]\!]\} & \text{over} \\
\\
[\![A{\odot_k}B]\!] & = & \{\times_k(s_1, s_2) |\ s_1 \in [\![A]\!] \ \&\ s_2 \in [\![B]\!]\} & k > 0 \text{ deterministic discontinuous product} \\
[\![A{\downarrow_k}C]\!] & = & \{s_2 |\ \forall s_1 \in [\![A]\!], \times_k(s_1, s_2) \in [\![C]\!]\} & k > 0 \text{ deterministic discontinuous infix} \\
[\![C{\uparrow_k}B]\!] & = & \{s_1 |\ \forall s_2 \in [\![B]\!], \times_k(s_1, s_2) \in [\![C]\!]\} & k > 0 \text{ deterministic discontinuous extract}
\end{array}
$$

Figure 3.2: Standard syntactical interpretation of **D** types

$$
\begin{array}{rcllcll}
\mathcal{F}_i & ::= & \mathcal{A}_i & S(A) & = & i & \text{for } A \in \mathcal{A}_i \\
\\
\mathcal{F}_0 & ::= & I & S(I) & = & 0 & \\
\mathcal{F}_1 & ::= & J & S(J) & = & 1 & \\
\\
\mathcal{F}_{i+j} & ::= & \mathcal{F}_i{\bullet}\mathcal{F}_j & S(A{\bullet}B) & = & S(A) + S(B) & \\
\mathcal{F}_j & ::= & \mathcal{F}_i\backslash\mathcal{F}_{i+j} & S(A\backslash C) & = & S(C) - S(A) & \\
\mathcal{F}_i & ::= & \mathcal{F}_{i+j}/\mathcal{F}_j & S(C/B) & = & S(C) - S(B) & \\
\\
\mathcal{F}_{i+j} & ::= & \mathcal{F}_{i+1}{\odot_k}\mathcal{F}_j & S(A{\odot_k}B) & = & S(A) + S(B) - 1 & 1 \le k \le i+1 \\
\mathcal{F}_j & ::= & \mathcal{F}_{i+1}{\downarrow_k}\mathcal{F}_{i+j} & S(A{\downarrow_k}C) & = & S(C) + 1 - S(A) & 1 \le k \le i+1 \\
\mathcal{F}_{i+1} & ::= & \mathcal{F}_{i+j}{\uparrow_k}\mathcal{F}_j & S(C{\uparrow_k}B) & = & S(C) + 1 - S(B) & 1 \le k \le i+1
\end{array}
$$

Figure 3.3: Sorted **D** types, and syntactical sort map for **D**

Observe also that (modulo sorting) $(\backslash, \bullet, /; \subseteq)$ and $(\downarrow_k, \odot_k, \uparrow_k; \subseteq)$ are residuated triples:

(115) $B \subseteq A\backslash C$ iff $A{\bullet}B \subseteq C$ iff $A \subseteq C/B$
$B \subseteq A{\downarrow_k}C$ iff $A{\odot_k}B \subseteq C$ iff $A \subseteq C{\uparrow_k}B$

We are now in a position to define the categorical calculus **cD** which subsumes the continuous Lambek calculus **L**. The postulates of **cD** are in bijection with the axioms of the equational theory $\mathbf{Eq_D}$ of Chapter 3. We call this set of postulates $\mathbf{Eq_D}^*$. In Figure 3.4 the categorical calculus **cD** is displayed.

In the style of Moortgat (1995), in the next section we consider the associated sorted multimodal calculus corresponding to **cD**, **mD**. There, the details of the intimately related concepts of $\mathbf{Eq_D}$ and **mD** are fully explained. We will see that the work done in Chapter 3 helps us to set up the formal properties of **mD**. As said before, we will formulate and prove important embedding translations with the goal in mind to illuminate hidden relations between **mD** and its corresponding hypersequent calculus **hD**, which is a calculus that has absorbed the structural postulates. These results are also relevant because chronologically the work of this author and Glyn Morrill and Mario Fadda presented **hD** with two equivalent distinct hypersequent sintaxes. No mention of the hidden multimodal calculus and its formal properties with respect to **FreeDisp** were made. So, this is the first time, then, that it is presented.[1]

(116) **Definition** (*arrow of* **D**)

An *arrow* of **D** is of the form $A \to B$ where A and B are types of **D** of the same sort.

We write $\vdash_{\mathbf{cD}} A \to B$ when an arrow $A \to B$ is derived in the calculus **cD**. It is not difficult to have the concept of context for subtypes of types like subterms in a term. The internal nodes (i.e. not leaves) of a context must have as type-constructors only the product connectives, i.e. $\bullet$ and $\odot_i$ $(i > 0)$. In the next section, the reader aquainted with substructural logic will notice that the concept of context for types is mimicking the concept of context for a subtructure in the antecedent of a sequent.

3.2 A Sorted Multimodal Calculus for D: mD

We consider now a sorted multimodal calculus with structural rules mimicking the axioms of the categorical calculus of **D**, **cD**, and hence the axioms of the equational theory $\mathbf{Eq_D}$ of Chapter 3. This sequent calculus is non standard in two senses. Types and structural terms are sorted. Moreover, there are two structural term constants which stand respectively for the continuous unit and discontinuous unit. Structural term constructors are of two kinds: $\circ$ (which stands for term concatenation) and $\circ_i$ (which stands for term wrapping at the

[1] It might be interesting for the reader to know that the first version accepted (but not the one that was finally published) in the Journal of Logic, Language and Information of the paper *The Displacement Calculus*, contained in part material on the hidden multimodal calculus **mD**. In one conference (Conference Joachim Lambek: Mathematics, Logic and Language, Chieti 2011) and one workshop (Second Workshop on Types, Logic, and Grammar (Barcelona, 2007)) this material was presented with more details.

Axiom

$A \to A$ for every $A \in \mathcal{F}$

Residuation laws

$B \to A\backslash C$ iff $A \bullet B \to C$ iff $A \to C/B$

$B \to A{\downarrow_k}C$ iff $A \odot_k B \to C$ iff $A \to C{\uparrow_k}B$

Postulates

$A \bullet (B \bullet C) \leftrightarrow (A \bullet B) \bullet C$	Continuous associativity
$(A \odot_i B) \odot_j C \leftrightarrow (A \odot_j C) \odot_{i+S(B)-1} B$, if $j < i$	Mixed permutation (Case 1)
$(A \odot_i B) \odot_j C \leftrightarrow A \odot_i (B \odot_{j+1-i} C)$, if $i \leq j < i + S(B)$	Discontinuous associativity
$(A \odot_i B) \odot_j C \leftrightarrow (A \odot_{j+1-S(B)} C) \odot_i B$, if $i + S(B) \leq j$	Mixed permutation (Case 2)
$(A \bullet J) \odot_{S(A)+1} B \leftrightarrow A \bullet B$	Split-Wrap (Case 1)
$(J \bullet B) \odot_1 A \leftrightarrow A \bullet B$	Split-Wrap (Case 2)
$I \bullet A \leftrightarrow A \leftrightarrow A \bullet I$	Continuous unit
$J \odot_1 A \leftrightarrow A \leftrightarrow A \odot_i J$, $1 \leq i \leq S(A)$	Discontinuous unit

Transitivity

$A \to C$ if $A \to B$ and $B \to C$

Figure 3.4: Categorical calculus **cD**

i-th position, $i \in \omega^+$). Again, as in the case of sorted types, structural terms are defined by mutual recursion and the *sort map* is computed in a similar way (see (118)).

$X[Y]$ denotes a structural term with a distinguished position occupied by the structural term Y. If A, X are respectively a type and a structural term, then a and x denote their sorts. We will enter into more detail in the following lines. We are interested in the cardinality of the set $\mathcal{F}$ of types of **D** and their structure. Consider the following lemma:

(117) **Lemma**
The set of types $\mathcal{F}$ is denumerable iff the set of atomic types is denumerable.

$$\begin{array}{rcl} \mathcal{F} & = & \bigcup_{i\in\omega} \mathcal{F}_i \\ \mathcal{F}_i & = & (A_{ij})_{j\in\omega} \end{array}$$

Proof. The proof can be carried out by coding in a finite alphabet the set of types $\mathcal{F}$. Of course, it is crucial that the set of sorted atomic types forms a denumerable set. In the next Chapter, this coding is explicitely worked out. □

Let $\mathbf{StructTerm_D}[\mathcal{F}]$ be the ω-sorted algebra over the signature $\Sigma_D = (\{\circ\} \cup (\circ_{i+1})_{i\in\omega}, \mathbb{I}, \mathbb{J})$. The sort functionality of Σ_D is:

$$((i, j \to i + j)_{i,j\in\omega}, (i + 1, j \to i + j)_{i,j\in\omega}, 0, 1)$$

Observe that the operations $\circ$ and $\circ_i$'s (with $i > 0$) are sort polymorphic. In the following, we will abbreviate $\mathbf{StructTerm_D}[\mathcal{F}]$ by **StructTerm**. The set of *structural terms* can be defined in BNF notation as follows:

$$(118)\quad \begin{array}{rcl} \mathbf{StructTerm}_0 & ::= & \mathbb{I} \\ \mathbf{StructTerm}_1 & ::= & \mathbb{J} \\ \mathbf{StructTerm}_i & ::= & \mathcal{F}_i \\ \mathbf{StructTerm}_{i+j} & ::= & \mathbf{StructTerm}_i \circ \mathbf{StructTerm}_j \\ \mathbf{StructTerm}_{i+j} & ::= & \mathbf{StructTerm}_{i+1} \circ_k \mathbf{StructTerm}_j \end{array}$$

It is clear that the sort of $\mathbf{StructTerm}_i$ is such that:

$$S(\mathbf{StructTerm}_i) = i$$

As we remarked above we realize that **StructTerm** looks like an ω-sorted term algebra. This intuition is correct for we can put **StructTerm** in bijection with the ω-sorted term algebra $T_{\Sigma_D}[X]$ of Chapter 3.

Let us consider the following bijective mapping f from the set of variables X of Chapter 3 into the set of types $\mathcal{F}$.[2]

$$\begin{array}{rccc} f: & X & \longrightarrow & \mathcal{F} \\ & x_{ij} & \mapsto & A_{ij} \end{array}$$

This bijection is such that (for $i, j \in \omega$):

[2]The existence of this bijection is not difficult to see given that the set of atomic variables and atomic types are denumerable.

$$S(x_{ij}) = S(f(x_{ij})) = S(A_{ij}) = i$$

So, for every $i \in \omega$, the sets $(x_{ij})_{j\in\omega}$ and $(A_{ij})_{j\in\omega}$ are respectively the set of sorted variables of sort i and the set of types of sort i. f extends recursively to f^* as follows:

(119)

$$\begin{array}{llll} f^*: & T_{\Sigma_D}[X] & \longrightarrow & \mathbf{StructTerm} \\ & 0 & \mapsto & \mathbb{I} \\ & 1 & \mapsto & \mathbb{J} \\ & x_{ij} & \mapsto & A_{ij} \\ & t \oplus s & \mapsto & f(t) \circ f(s) \\ & t \otimes_i s & \mapsto & f(t) \circ_i f(s) \end{array}$$

Since f is bijective and f extends recursively to f^*, it is easy to prove by induction on the structure of **StructTerm** that f^* is bijective. Notice that in fact f^* is a sorted Σ_D-isomorphism.

3.2.1 The Multimodal Calculus mD

The multimodal calculus **mD** is shown in Figures 3.5 and 3.6.

Like in the case of the equational theory $\mathbf{Eq_D}$ of Chapter 3, we need to define some important relations between structural terms. These directly mimick the *precedence* and *wrap* relations between terms in a product context. We overload the symbols used in Chapter 3:

(120) **Definition** (*Wrapping and Permutable Terms*)

Given the term $(T_1 \circ_i T_2) \circ_j T_3$, we say that:

(P1) $T_2 \prec_{T_1} T_3$ iff $i + t_2 - 1 < j$.

(P2) $T_3 \prec_{T_1} T_2$ iff $j < i$.

(O) $T_2 \between_{T_1} T_3$ iff $i \leq j \leq i + t_2 - 1$.

Observe that in a term like $(T_1 \circ_i T_2) \circ_j T_3$, if $(P1)$ or $(P2)$ hold, (O) does not apply. Conversely, if (O) is applicable, neither $(P1)$ nor $(P2)$ hold. If $T_2 \prec_{T_1} T_3$ (respectively $T_3 \prec_{T_1} T_2$), we say that T_2 and T_3 (respectively T_3 and T_2) *permute* in T_1. Otherwise, if (O) holds, we say that T_2 *wraps* T_3 in T_1.

Let us see now the structural rules. We define the following relation between structural terms $\sim$:

(121) $T \sim S$ iff S is the result of applying one structural rule to T

$\sim^*$ is defined to be the reflexive, symmetric and transitive closure of $\sim$. There exists a faithful embedding translation between **cD** and **mD**:

$$\begin{array}{ccc} \tau : \mathbf{cD} = (\mathcal{F}, \mathcal{F}, \rightarrow) & \longrightarrow & \mathbf{mD} = (\mathbf{StructTerm}, \mathcal{F}, \rightarrow) \\ A \rightarrow B & \mapsto & A \rightarrow B \end{array}$$

Let us see some simple useful results on **cD**. If we have the provable arrows $A \rightarrow B$ and $C \rightarrow D$ in **cD** then we have $\mathbf{cD} \vdash A \bullet C \rightarrow B \bullet D$. This was

$$A \to A \ \mathrm{Id} \qquad \frac{S \to A \qquad T[A] \to B}{T[S] \to B}\,Cut$$

$$\frac{T[\mathbb{I}] \to A}{T[I] \to A}\,IL \qquad \frac{}{\mathbb{I} \Rightarrow I}\,IR$$

$$\frac{T[\mathbb{J}] \to A}{T[J] \to A}\,JL \qquad \frac{}{\mathbb{J} \Rightarrow J}\,JR$$

$$\frac{X \to A \qquad Y[B] \to C}{Y[X \circ A\backslash B] \to C}\,\backslash L \qquad \frac{A \circ X \to B}{X \to A\backslash B}\,\backslash R$$

$$\frac{X \to A \qquad Y[B] \to C}{Y[B/A \circ X] \to C}\,/L \qquad \frac{X \circ A \to B}{X \to B/A}\,/R$$

$$\frac{X \to A \qquad Y[B] \to C}{Y[B \uparrow_i A \circ_i X] \to C}\uparrow_i L \qquad \frac{X \circ_i A \to B}{X \to B \uparrow_i A}\uparrow_i R$$

$$\frac{X \to A \qquad Y[B] \to C}{Y[X \circ_i A \downarrow_i B] \to C}\downarrow_i L \qquad \frac{A \circ_i X \to B}{X \to A \downarrow_i B}\downarrow_i R$$

$$\frac{X[A \circ B] \to C}{X[A \bullet B] \to C}\bullet L \qquad \frac{X \to A \qquad Y \to B}{X \circ Y \to A \bullet B}\bullet R$$

$$\frac{X[A \circ_i B] \to C}{X[A \odot_i B] \to C}\odot_i L \qquad \frac{X \to A \qquad Y \to B}{X \circ_i Y \to A \odot_i B}\odot_i R$$

Figure 3.5: Sorted multimodal calculus **mD**, Part I

Structural rules for units

- Continuous unit:

$$\frac{T[X] \to A}{T[\mathbb{I}\circ X] \to A} \qquad \frac{T[\mathbb{I}\circ X] \to A}{T[X] \to A} \qquad \frac{T[X] \to A}{T[X\circ \mathbb{I}] \to A} \qquad \frac{T[X\circ \mathbb{I}] \to A}{T[X] \to A}$$

- Discontinuous unit:

$$\frac{T[X] \to A}{T[\mathbb{J}\circ_1 X] \to A} \qquad \frac{T[\mathbb{J}\circ_1 X] \to A}{T[X] \to A} \qquad \frac{T[X] \to A}{T[X\circ_i \mathbb{J}] \to A} \qquad \frac{T[X\circ_i \mathbb{J}] \to A}{T[X] \to A}$$

Continuous associativity

$$\frac{X[(T_1\circ T_2)\circ T_3] \to D}{X[T_1\circ (T_2\circ T_3)] \to D}\,Assc_c \qquad \frac{X[T_1\circ (T_2\circ T_3)] \to D}{X[(T_1\circ T_2)\circ T_3] \to D}\,Assc_c$$

Split-wrap

$$\frac{T_1[T_2\circ T_3] \to D}{T_1[(\mathbb{J}\circ T_3)\circ_1 T_2] \to D}\,SW \qquad \frac{T_1[(\mathbb{J}\circ T_3)\circ_1 T_2] \to D}{T_1[T_2\circ T_3] \to D}\,SW$$

$$\frac{T_1[T_2\circ T_3] \to D}{T_1[(T_2\circ \mathbb{J})\circ_{t_2+1} T_3] \to D}\,SW \qquad \frac{T_1[(T_2\circ \mathbb{J})\circ_{t_2+1} T_3] \to D}{T_1[T_2\circ T_3] \to D}\,SW$$

Discontinuous associativity $T_2 \between_{T_1} T_3$

$$\frac{S[T_1\circ_i (T_2\circ_j T_3)] \to C}{S[(T_1\circ_i T_2)\circ_{i+j-1} T_3)] \to C}\,\mathbf{Assc_d 1} \qquad \frac{S[(T_1\circ_i T_2)\circ_j T_3] \to C}{S[T_1\circ_i (T_2\circ_{j-i+1} T_3)] \to C}\,\mathbf{Assc_d 2}$$

Mixed permutation 1 case $T_2 \prec_{T_1} T_3$

$$\frac{S[(T_1\circ_i T_2)\circ_j T_3] \to C}{S[(T_1\circ_{j-S(T_2)+1} T_3)\circ_i T_2] \to C}\,\mathbf{MixPerm1} \qquad \frac{S[(T_1\circ_i T_3)\circ_j T_2] \to C}{S[(T_1\circ_j T_2)\circ_{i+S(T_2)-1} z] \to C}\,\mathbf{MixPerm1}$$

Mixed permutation 2 case $T_3 \prec_{T_1} T_2$

$$\frac{S[(T_1\circ_i T_2)\circ_j T_3] \to C}{S[(T_1\circ_j T_3)\circ_{i+S(T_3)-1} T_2] \to C}\,\mathbf{MixPerm2} \qquad \frac{S[(T_1\circ_i T_3)\circ_j T_2] \to C}{S[(T_1\circ_{j-S(T_3)+1} T_2)\circ_i T_3] \to C}\,\mathbf{MixPerm2}$$

Figure 3.6: Sorted multimodal calculus **mD**, Part II

already proved by Lambek (1958) for unsorted types. For sorted types the proof is exactly the same. In the case of the discontinuous products $\odot_i$ ($0 < i \in \omega$) the same result holds , i.e. $A \to B$ and $C \to D$ are provable in **cD** then $\mathbf{cD} \vdash A \odot_i c \to B \odot_i D$. Again, the same reasoning works for this case.

We need the concept of type occurrence in a type (and later we will need the concept of structural term occurrence in another structural term) and therefore the concept of type context. Let us extend the set of types as follows:

(122) $\mathcal{F}_c \stackrel{def}{=} \mathcal{F} \cup (*^i_f)_{i \in \omega}$

The set $(*^i_f)_{i \in \omega}$ can be considered as an extra set of sorted types (the subindex f makes reference to the formulas or types). We call $*^i_f$ *type holes.* We say that an $A \in \mathcal{F}_c$ is a *product type context* iff either A is a simple type of $\mathcal{F}$ or A contains a single (or linear) occurrence of a type hole $*^i_f$ such that the $*^i_f$'s occurrence is not dominated in its subformula tree by an implicative connective like $\backslash, /$ or $\downarrow_i, \uparrow_i$ ($0 < i \in \omega$). Let A be a type context with type hole $*^b_f$ and B ($b = S(B)$, i.e. $*^b_f$ and B have the same sort) another type context (possibly a normal type). We define $A[B]$ as follows:

(123) $A[B] \stackrel{def}{=} \sigma_{B/*^b_f}(A)$

Where $\sigma_{B/*^b_f}$ denotes the type substitution map which replaces $*^b_f$ by B. Notice we have the interesting identity:

(124) $A[*^b_f] = A$

(125) **Remark**
The condition that a type hole's occurrence must not be dominated by an implicative connective in its subformula tree rules out type contexts such as for example $A = B \odot_2 (C/(E \bullet *^1_f))$, whereas a type context such as for example $B \odot_2 ((C\backslash D) \bullet *^1_f)$ is a correct product type context.

We need a simple lemma:

(126) **Lemma**
Let C be a product type context with a type hole $*^a_f$. Suppose that $\mathbf{cD} \vdash A \to B$, where A and B are types.[3] Then:

$$\mathbf{cD} \vdash C[A] \to C[B]$$

Proof. By induction on the structure of C:

1. The case $C = *^a_f$ is trivial.
2. Suppose without loss of generality that $C = C_1 \odot_k C_2[*^a_f]$ for some $0 < k \in \omega$. By induction hypothesis we have that:

$$\mathbf{cD} \vdash C_2[A] \to C_2[B]$$

[3]If A and B were in $\mathcal{F}_c$ the result would go through also. But for the purposes of this lemma it is not really necessary.

By the monotonicity properties of **cD** we have the following inference:

$$\frac{C_1 \to C_1 \qquad C_2[A] \to C_2[B]}{C_1 \odot_k C_2[A] \to C_1 \odot_k C_2[B]} \textbf{ Monotonicity}$$

Hence the lemma is true for discontinuous products. For continuous products it is completely similar. This completes the proof.

□

The work we have done on type contexts must also be done for structural terms. Let us extend their set **StructTerm** as follows:

(127) $\textbf{StructTerm}_c \overset{def}{=} \textbf{StructTerm} \cup (*^i_{st})_{i\in\omega}$

The set $\textbf{StructTerm}_c$ for structural terms mimics the set $\mathcal{F}_c$. The set of *structural holes*[4] can be considered to be an extra set of structural terms. A structural context T is therefore defined as a structural term with variables in $\mathcal{F} \cup (*^i_{st})_{i\in\omega}$ such that every occurrence of a structural hole is linear. Consider T with structural hole $*^i_{st}$. Let S be another structural context of sort $s = S(S)$. We define $T[S]$ as follows:

(128) $T[S] \overset{def}{=} \sigma_{S/*^s_{st}}(T)$

As in the case of type contexts we have the following interesting identity:

$$T[*^i_{st}] = T$$

Where $\sigma_{S/*^s_{st}}$ denotes the structural term substitution map which replaces $*^s_{st}$ by S. We now define what we call the *type equivalent for a structural term*:

(129) **Definition** (*Type Equivalent of a Structural Term*)

Given a structural term T we define the type-equivalent $T^\bullet$ of T by induction on the structure of T:

- If $T = \mathbb{I}$ then $T^\bullet = I$.
- If $T = \mathbb{J}$ then $T^\bullet = J$.
- If $T = A \in \mathcal{F}$ then $T^\bullet = A$.
- If $T = S_1 \circ S_2$ then $T^\bullet = S_1^\bullet \bullet S_2^\bullet$.
- If $T = S_1 \circ_i S_2$ then $T^\bullet = S_1^\bullet \odot_i S_2^\bullet$.

The type equivalent can also be defined for structural contexts. We have to add to the above conditions the following one:

(130) If $S = *^i_{st}$ then $S^\bullet = *^i_f$

The following lemma will be useful for theorem (133):

(131) **Lemma**
Let T be a structural context and S be a structural term.[5] Then:

[4] the subscript *st* stands for *structural terms.*
[5] S could be also a structural context.

(132) $(T[S])^\bullet = T^\bullet[S^\bullet]$

Proof. By induction on the structure of structural terms:

1. Suppose $T = *^s_{st}$ $(s = S(S))$. We have that $T^\bullet = *^s_f$. It follows that $T^\bullet[S^\bullet] = S^\bullet$.

2. Suppose now T is complex, say:

$$T = T_1 \circ_i T_2[*^s_{st}]$$

$T^\bullet = T_1^\bullet \odot_i T_2^\bullet[*^s_f]$, so $T^\bullet[S^\bullet] = T_1^\bullet \odot_i T_2^\bullet[S^\bullet]$. On the other hand, $(T[S])^\bullet = T_1^\bullet \odot_i (T_2[S])^\bullet$. By induction hypothesis $(T_2[S])^\bullet = T_2^\bullet[S^\bullet]$. Hence:

$$(T[S])^\bullet = T^\bullet[S^\bullet]$$

Other cases are completely similar. This completes the proof.

□

With the formal machinery we have defined we can prove the following theorem:

(133) **Theorem** (*Embedding Translation between* **cD** *and* **mD**)

Let $A, B \in \mathcal{F}$ and $T \in$ **StructTerm**. The following holds:

i) If $\vdash_{\mathbf{cD}} A \to B$ then $\vdash_{\mathbf{mD}} A \to B$

ii) If $\vdash_{\mathbf{mD}} T \to A$ then $\vdash_{\mathbf{cD}} T^\bullet \to B$

Proof.

i) Axioms are obvious as well as Cut. In **mD** we have the following fact:

$$\vdash_{\mathbf{mD}} A \odot_i B \to C \text{ iff } \vdash_{\mathbf{mD}} A \circ_i B \to C$$

The *if* case corresponds to the $\odot_i$ left rule. The *only if* is justified by Cut with $A \circ_i B \to A \odot_i B$.

Let us see the residuation laws of **cD**:[6]

$$A \odot_i B \to C \text{ iff } A \to C \uparrow_i B$$

[*only if*] case. We have:

$$\dfrac{\dfrac{A \circ_i B \to A \odot_i B \qquad A \odot_i B \to C}{A \circ_i B \to C}\,Cut}{A \to C \uparrow_i B}\,\uparrow_i R$$

[*if*] case. We have:

[6] We consider the case of discontinuous connectives. Continuous connectives follow an analogous proof.

$$\frac{A \to C\uparrow_i B \qquad C\uparrow_i B\circ_i B \to C}{A\circ_i B \to C}\, Cut$$

Whence $A \odot_i B \to C$. The case of $\downarrow_i$ is completely similar.

Non-logical axioms of **cD**. Consider for example the case of Split-Wrap:

$$(A \bullet J)\odot_{a+1} \leftrightarrow A \bullet B$$

(134)
$$\cfrac{\cfrac{\cfrac{\cfrac{A\circ B \to A \bullet B}{(A\circ\mathbb{J})\circ_{a+1}B \to A \bullet B}\,\mathbf{SW}}{(A\circ J)\circ_{a+1}B \to A \bullet B}\,JL}{(A \bullet J)\circ_{a+1}B \to A \bullet B}\,\bullet L}{(A \bullet J)\odot_{a+1} B \to A \bullet B}\,\odot_{a+1}L$$

Other structural rules have similar reasoning.

ii) We prove by induction on the length of the derivations of **mD** that if $\vdash_{\mathbf{mD}} T \to A$ then $\vdash_{\mathbf{cD}} T^\bullet \to A$. We will use extensively two lemmas, lemma (131) on type equivalents and lemma (126) on type contexts.

- Atomic axioms are obvious.
- Suppose that the last rule is $\uparrow_i L$:

$$\frac{T \to A \qquad S[B] \to C}{S[B\uparrow_i A\circ_i S] \to C}\uparrow_i L$$

  By induction hypothesis (i.h.), $T^\bullet \to A$ and $S^\bullet[B] \to C$ are provable in **cD**. We have that $\vdash_{\mathbf{cD}} B\uparrow_i A \odot_i T^\bullet \to B\uparrow_i A \odot A \to B$. By transitivity $\vdash_{\mathbf{cD}} B\uparrow_i A \odot_i T^\bullet \to B$.

$$S^\bullet[B\uparrow_i A \odot_i T^*] \to S^\bullet[B] \text{ and } S^\bullet[B] \to C$$

  Again, by transitivity $S^\bullet[B\uparrow_i A \odot_i T^\bullet \to C$. And we know that $(S[B\uparrow_i A \odot_i T^\bullet)^\bullet = S^\bullet[B\uparrow_i A \odot_i T^*]$. This proves the case of $\uparrow_i L$.
- Suppose that the last rule is $\uparrow_i R$. We have by i.h. that $(T\circ_i A)^\bullet \to B$. We know that $(T\circ_i A)^\bullet = T^\bullet \odot_i A$. By residuation in **cD** we infer that $T^\bullet \to B\uparrow_i A$. This proves the case of $\uparrow_i R$. The case of $\downarrow_i L$ and $\downarrow_i R$ are almost identical.
- The cases of $\odot_i L$ and $\odot_i R$ are completely similar in reasoning to the cases of the two rules for $\uparrow_i$.
- Suppose now that the last rule is for example discontinuous associativity, i.e.:

$$\frac{T[A\circ_i(B\circ_j C)] \to C}{T[(A\circ_i B)\circ_{i+j-1}C] \to C}\,\mathbf{Assc_d1}$$

  By i.h. $(T[A\circ_i(B\circ_j C)])^\bullet \to C$ is provable in **cD**. We have that $(T[A\circ_i(B\circ_j C)])^\bullet = T^\bullet[A \odot_i (B \odot_j C)]$. In **cD** $\vdash_{\mathbf{cD}} (A \odot_i B) \odot_{i+j-1} C \to A \odot_i (B \odot_j C)$. Hence the following arrow is provable:

$$T^\bullet[(A \odot_i B) \odot_{i+j-1} C] \to T^\bullet[A \odot_i (B \odot_j C)]$$

By i.h. $(T[A \odot_i (B \odot_j C)])^\bullet \to D$ and we know that $(T[A \odot_i (B \odot_j C)])^\bullet = T^\bullet[A \odot_i (B \odot_j C)]$ and $(T[(A \odot_i B) \odot_{i+j-1} C])^\bullet = T^\bullet[(A \odot_i B) \odot_{i+j-1} C]$. By transitivity we have therefore the following provable arrow:

$$(T[(A \odot_i B) \odot_{i+j-1} C])^\bullet \to D$$

Which is what we wanted to prove. Other non-logical axioms of **cD** follow completely similar reasonings.

– Suppose the last rule is Cut:

$$\frac{T \to A \qquad S[A] \to C}{S[T] \to C}\,Cut$$

By i.h. :

$$T^\bullet \to A \qquad (S[A])^\bullet \to C$$

We know that $(S[A])^\bullet = S^\bullet[A] \to C$ and that $(S[T])^\bullet = S^\bullet[T^\bullet]$. We have that

$$S^\bullet[T^\bullet] \to S^\bullet[A] \to C$$

By Transitivity, the following arrow is provable:

$$S^\bullet[T^\bullet] \to C$$

This completes the proof.

□

3.3 Absorbing the Structural Rules: the Rise of Hypersequent Syntax

If one wants to absorb the structural rules of a Gentzen sequent system in a substructural logic, one has to discover a convenient data structure for the antecedent and the succedent of sequents. Here we propose two equivalent data structures. They will both be called *hypersequent syntax*, a term which must not be confused with Avron's hypersequents (Avron (1991)). The reason for using the prefix *hyper* in the term *sequent* is that the data-structures which are proposed are quite nonstandard.

3.3.1 Hypersequent Syntax I: the String-Based Version

Let us define what we call the set of types segments:

(135) **Definition** (*Type Segments*)

In hypersequent calculus we define the *types segments* $\mathcal{SF}_k$ of sort k:

$$\begin{array}{lll} \mathcal{SF}_0 & ::= & A \quad \text{for } A \in \mathcal{F}_0 \\ \mathcal{SF}_a & ::= & \sqrt[i]{A} \quad \text{for } A \in \mathcal{F}_a \text{ and } 0 \leq i \leq a = S(A) \end{array}$$

A notational convention. When A is an arbitrary type, we will frequently write in lowercase the type in order to refer to its sort $S(A)$, i.e.:

$$a \stackrel{def}{=} S(A)$$

Types segments of sort 0 are types. But, types segments of sort greater than 0 are no longer types. Strings of types segments can form meaningful logical material like the set of hyperconfigurations, which we now define. The *hyperconfigurations* $\mathcal{O}$ are defined unambiguously by mutual recursion as follows, where Λ is the empty string:

$$\begin{array}{lll} \mathcal{O} & ::= & \Lambda \\ \mathcal{O} & ::= & A, \mathcal{O} \text{ for } S(A) = 0 \\ \mathcal{O} & ::= & [], \mathcal{O} \\ \mathcal{O} & ::= & \sqrt[0]{A}, \mathcal{O}, \sqrt[1]{A}, \ldots, \sqrt[a-1]{A}, \mathcal{O}, \sqrt[a]{A}, \mathcal{O} \\ & & \text{for } a = S(A) > 0 \end{array}$$

By the vectorial notation $\overrightarrow{A}$ we mean a particular hyperconfiguration for every type A which we will use many times throughout this Chapter:

$$\overrightarrow{A} \stackrel{def}{=} \begin{cases} A \text{ if } S(A) = 0 \\ \sqrt[0]{A}, [], \sqrt[1]{A}, \ldots, \sqrt[a-1]{A}, [], \sqrt[a]{A} \quad \text{if } a = S(A) > 0 \end{cases}$$

Notice that this notation is nonstandard in the field of substructural logics. The intended meaning of $\sqrt[i]{A}$ (for a type A with sort $a = S(A) > 0$ and $0 \leq i \leq a$) is that the i-th element α_i of a string $\alpha_0 + 1 + \alpha_1 + \ldots + \alpha_{a-1} + 1 + \alpha_a$ inhabiting $[\![A]\!]$. It is important to signal that a single occurrence of a types segment does not have the status of a type. Only the so-called vectors $\overrightarrow{A}$ and *generalized wrappings* (to be defined later) constitute the correct hyperconfigurations and subhyperconfigurations. Moreover, note that not every substring of a hyperconfiguration is a (well-formed) hyperconfiguration because as well as containing all the segments of discontinuous types, these segments must be separated by correct hyperconfigurations. For example, where C is supposed to be of sort 0 if A is a type of sort 2 then the following expression is incorrect:

(136) $\sqrt[0]{A}, C \sqrt[1]{A} \notin \mathcal{O}$

We define the *components* of a hyperconfiguration as its maximal substrings not containing the separator $[]$. Components can be incorrect hyperconfigurations. (136) exemplifies an incorrect hyperconfiguration whereas the following hyperconfiguration is correct (here A and C denote arbitrary types of sort respectively 2 and 0):

$$\underbrace{\sqrt[0]{A}, C\sqrt[1]{A}}_{\text{first component}}, [], \underbrace{\sqrt[2]{A}, I, C/C}_{\text{second component}} \in \mathcal{O}$$

The set of the string-based hyperconfigurations will be denoted when necessary as $\mathcal{O}_s$ or simply $\mathcal{O}$.

3.3.2 Hypersequent Syntax II: the Tree-Based Version

The main point of this subsection is to give an alternative hypersequent syntax which is tree-based. It is interesting to note that this kind of hypersequent syntax allowed[7] Morrill (2011a) to implement the system CatLog which has all the power of **D** and its extensions studied in this thesis. Let $\mathcal{N}$ and $\mathcal{L}$ be the following sets:

(137)
$$\begin{array}{lcl} \mathcal{N} & \stackrel{def}{=} & \{\cdot\} \cup \bigcup_{i\in\omega} \mathcal{F}_{i+1} \\ \mathcal{L} & \stackrel{def}{=} & \{\Lambda\} \cup \{[]\} \cup \mathcal{F}_0 \end{array}$$

Sets $\mathcal{N}$ and $\mathcal{L}$ are called respectively, the set of *internal nodes* and the set of *leaves*. We now define a set of trees which we call *hypertrees*. Let $\mathcal{T}$ be the set of hypertrees generated by:

i) The set **Init** of initial hypertrees of Figure 3.7.

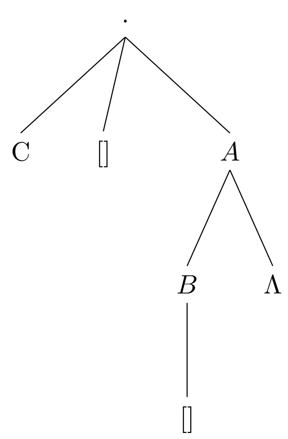
ii) The hypertree generating functions of Figure 3.8.

The partial functions $\mathbf{Conc_1}$, $\mathbf{Conc_2}$ and $\mathbf{Init_i}$ have disjoint domains. In fact, $\mathcal{T}$ is the least set satisfying the equation:

(138) $\mathcal{T} \stackrel{def}{=} \mathbf{Init} \cup \mathbf{Conc_1}(\mathcal{T},\mathcal{T}) \cup \mathbf{Conc_2}(\mathcal{T},\mathcal{T}) \cup \mathbf{Int_i}(\mathcal{T},\mathcal{T})$

Let us give an example:

(139) a. Consider the hypertree: where $A \in \mathcal{F}_2, B \in \mathcal{F}_1$ and $C \in \mathcal{F}_0$:



We now give the linearized version $\mathcal{O}_t$ corresponding to $\mathcal{T}$:

[7]Glyn Morrill p.c.

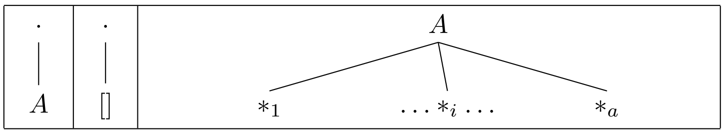


Figure 3.7: Initial hypertrees with $*_i \in \{\Lambda, []\}$

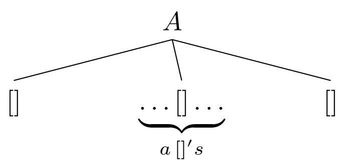
(140)
$$\begin{array}{lcl}\mathcal{O}_t & ::= & \Lambda \\ \mathcal{O}_t & ::= & A, \mathcal{O}_t \\ \mathcal{O}_t & ::= & [], \mathcal{O}_t \\ \mathcal{O}_t & ::= & A\{\underbrace{\mathcal{O}_t : \ldots : \mathcal{O}_t}_{a\ \mathcal{O}_t\text{'s}}\}, \mathcal{O}_t \text{ for } a = S(A) > 0\end{array}$$

It is easy to see that $\mathcal{O}_t$ and $\mathcal{T}$ are in bijective correspondance. Types of sort greater than 0 appear in the hypertrees as internal nodes. The intuition is that types of sort greater than 0 appear in our multimodal calculus **mD** as leaves which wrap some structural terms. In order to maintain the information that these types are wrapping is the fact of occurring as internal nodes in the hypertrees. The children of a given type A with $a = S(A) > 0$ correspond to the material wrapped in the multimodal calculus. This intuition justifies that we call *hyperleaves* the internal nodes labelled by types (of sort greater than 0). Therefore, the interpretation in a syntactical algebra is the following:

$$\begin{array}{l}[\![A\{\Delta_1 : \cdots : \Delta_a\}]\!] \stackrel{def}{=} \\ \{\alpha_0 + \delta_1 + \alpha_1 + \cdots + \alpha_{a-1} + \delta_a + \alpha_a : \\ \alpha_0 + 1 + \alpha_1 + \cdots + \alpha_{a-1} + 1 + \alpha_a \in [\![A]\!] \text{ and } \delta_i \in [\![\Delta_i]\!],\ 1 \leq i \leq a\}\end{array}$$

The vectorial notation for types, used in the string-based hypersequent syntax, corresponds in the tree-based case to:

(141) $\overrightarrow{A} \stackrel{def}{=}$



Both string and tree based hypersequent syntaxes are very similar as we shall see later. By recursion on the structure of $\mathcal{O}_s$ we see that $\mathcal{O}_s$ and $\mathcal{O}_t$ are in bijective correspondance:

(142)
$$\begin{array}{rcl}\mathcal{O}_s & \stackrel{h}{\longrightarrow} & \mathcal{O}_t \\ \Lambda & \mapsto & \Lambda \\ A, \Delta & \mapsto & A, h(\Delta) \\ [], \Delta & \mapsto & [], h(\Delta) \\ \sqrt[0]{A}, \Delta_1, \sqrt[1]{A}, \cdots, \sqrt[a-1]{A}, \Delta_a, \sqrt[a]{A}, \Delta_{a+1} & \mapsto & A\{h(\Delta_1) : \cdots : h(\Delta_a)\}, h(\Delta_{a+1}) \\ & & \text{A is such that } S(A) > 0 \text{ and } \Delta_i \in \mathcal{O}_s, \\ & & i = 1, \cdots, a+1\end{array}$$

The set of tree-based hyperconfigurations will be denoted when necessary as $\mathcal{O}_t$ or simply $\mathcal{O}$.

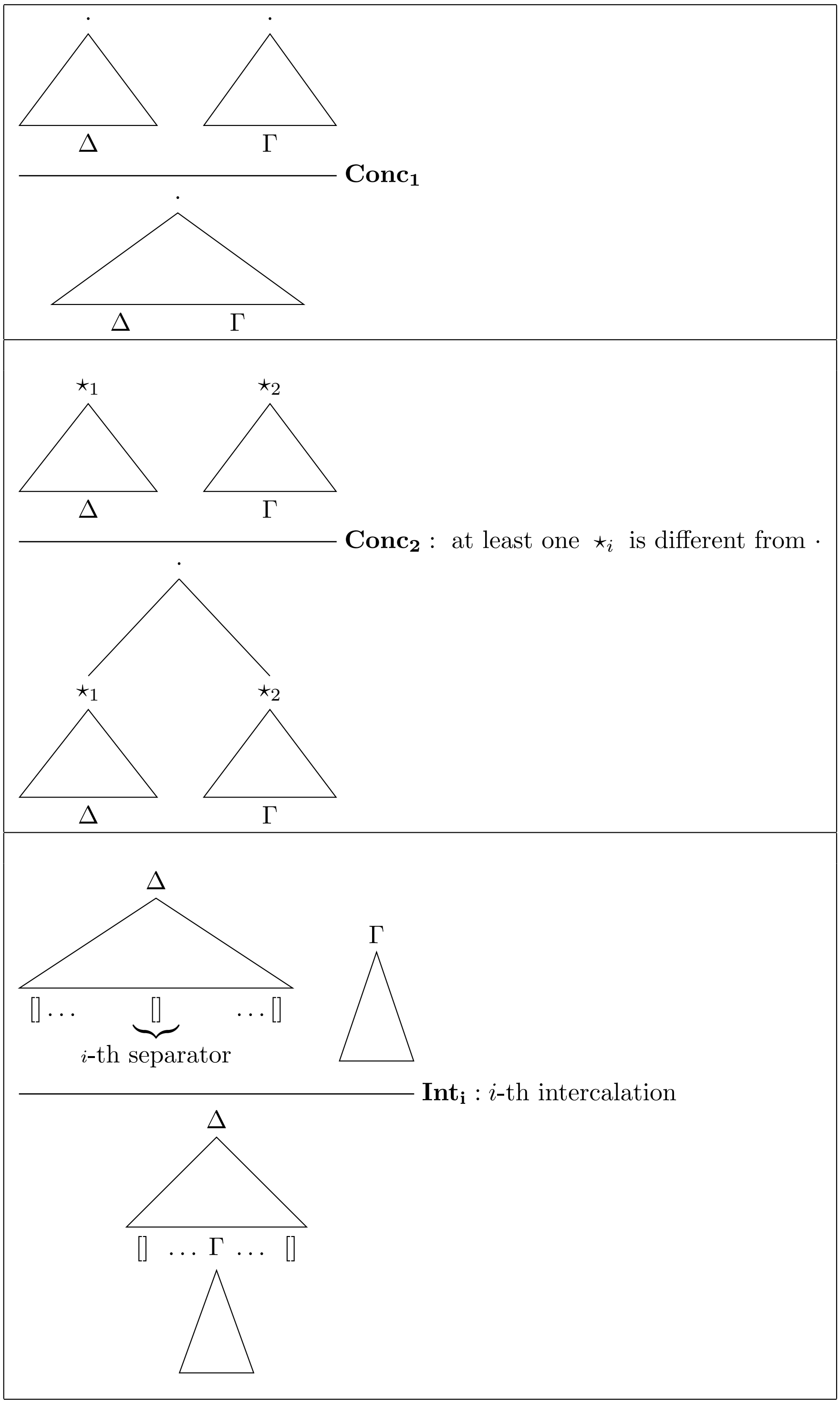


Figure 3.8: Hypertree generating functions

3.4 Hypersequent Calculus: the Intermediate Formulation

Some useful operations on hyperconfigurations

In this subsection we present some syntactic machinery in order to manipulate easily hyperconfigurations in both hypersequent syntaxes. The results we present will be worked out with $\mathcal{O}_s$ and hold of also in $\mathcal{O}_t$ with completely similar reasonings. In the following, we will write $\mathcal{O}$ instead of $\mathcal{O}_s$. A convention is to write in latin lower-case the sort of a hyperconfiguration, i.e.:

$$d \stackrel{def}{=} S(\Delta)$$

We now define three operations on hyperconfigurations and show that the set of hyperconfigurations is closed under them. We use the string-based hypersequent syntax. Later we will see that these results hold also in the case of the tree-based hypersequent syntax.

(143) **Definition** (*Concatenation of hyperconfigurations*)

$$\begin{array}{lll} (,): & \mathcal{O} \times \mathcal{O} & \longrightarrow (\mathcal{SF} \cup \{[]\})^* \\ & (\Delta, \Gamma) & \mapsto \Delta, \Gamma \end{array}$$

(144) **Definition** (*Intercalation of a Hyperconfiguration at the* i*-th Occurrence of a Separator in a Hyperconfiguration*)

$$\begin{array}{lll} |_i: & \mathcal{O} \times \mathcal{O} & \longrightarrow (\mathcal{SF} \cup \{[]\})^* \\ & (\Delta, \Gamma) & \mapsto \Delta|_i\Gamma \end{array}$$

(145) **Definition** (*Generalized Wrapping*)

$$\begin{array}{lll} \otimes: & \mathcal{O} \times \mathbf{List}(\mathcal{O}) & \longrightarrow (\mathcal{SF} \cup \{[]\})^* \\ & (\Delta, \langle \Gamma_1, \cdots, \Gamma_d \rangle) & \mapsto \Delta \otimes \langle \Gamma_1, \cdots, \Gamma_d \rangle \end{array}$$

Where $\mathbf{List}(\mathcal{O})$ is the set of finite lists of hyperconfigurations and $1 \leq b \leq d$ and $d = S(\Delta)$. In the case that $\Delta = \overrightarrow{A}$ we see that:

$$\begin{cases} \overrightarrow{A} \otimes \langle \Delta_1, \cdots, \Delta_a \rangle = \sqrt[0]{A}, \Delta_1, \sqrt[1]{A}, \cdots, \Delta_{a-1}, \sqrt[a-1]{A}, \Delta_a, \sqrt[a]{A} \\ \text{for the string-based hypersequent syntax} \\ \\ \overrightarrow{A} \otimes \langle \Delta_1, \cdots, \Delta_a \rangle = A\{\Delta_1 : \cdots : \Delta_a\} \\ \text{for the tree-based hypersequent syntax} \end{cases}$$

In the following lines we prove some lemmas which tell us that the set of hyperconfigurations is closed under the operations of intercalation and generalized wrapping.

(146) **Remark**
In these lemmas we use the fact that if $\mathcal{O} \Rightarrow \Delta$ then we have also $\mathcal{O} \Rightarrow \Delta, \mathcal{O}$, for the BNF production rules are all right-branching and a leftmost derivation has the form $\mathcal{O} \Rightarrow \alpha, \mathcal{O}$, where α may contain the (unique) non-terminal $\mathcal{O}$.

(147) **Lemma**
Let Δ and Γ be arbitrary hyperconfigurations. We have that:

$$\Delta, \Gamma \in \mathcal{O}$$

Proof. If $\Delta = \Lambda$ then trivially $\Delta, \Gamma \in \mathcal{O}$. Suppose then that $\Delta \neq \Lambda$. We know by the remark above that if $\mathcal{O} \Rightarrow \Delta$ then $\mathcal{O} \Rightarrow \Delta, \mathcal{O}$:

$$\mathcal{O} \Rightarrow \Delta, \mathcal{O}$$

Since $\mathcal{O} \Rightarrow \Gamma$, we derive then from $\mathcal{O} \Rightarrow \Delta, \mathcal{O}$ the following:

$$\mathcal{O} \Rightarrow \Delta, \mathcal{O} \Rightarrow \Delta, \Gamma$$

This proves that $\Delta, \Gamma \in \mathcal{O}$. □

(148) **Lemma**
Let $\Delta \in \mathcal{O}$ such that $d = S(\Delta) > 0$. Let $\Gamma \in \mathcal{O}$. It follows that:

$$\Delta|_i\Gamma \in \mathcal{O}$$

Proof. By hypothesis we have that:

$$\mathcal{O} \Rightarrow \Delta \text{ and } \mathcal{O} \Rightarrow \Gamma$$

The leftmost derivation of Δ until the i-th separator appears in the derivation as follows:

$$\mathcal{O} \Rightarrow \alpha, \mathcal{O}, \beta \overset{\mathcal{O} \to [], \mathcal{O}}{\Rightarrow} \alpha, \underbrace{[]}_{i\text{-th } []}, \mathcal{O}, \beta \Rightarrow \alpha, [], \gamma, \tau = \Delta$$

Where we have applied the BNF production $\mathcal{O} \to [], \mathcal{O}$. Here, α is terminal, β is such that it may contain occurrences of $\mathcal{O}$ and $\mathcal{O}, \beta \Rightarrow \gamma, \tau$, where γ and τ are terminal. We know that if $\mathcal{O} \Rightarrow \Gamma$ then $\mathcal{O} \Rightarrow \Gamma, \mathcal{O}$. From the derivation $\mathcal{O} \Rightarrow \alpha, \underline{\mathcal{O}}, \beta$ and replacing the underlined $\mathcal{O}$ by $\mathcal{O} \Rightarrow \Gamma, \mathcal{O}$:

$$\mathcal{O} \Rightarrow \alpha, \mathcal{O}, \beta \overset{\mathcal{O} \Rightarrow \Gamma, \mathcal{O}}{\Rightarrow} \alpha, \Gamma, \mathcal{O}, \beta \overset{\mathcal{O}, \beta \overset{*}{\Rightarrow} \gamma, \tau}{\Rightarrow} \alpha, \Gamma, \gamma, \tau = \Delta|_i\Gamma$$

This completes the proof. □

(149) **Lemma**
Let $\Delta \in \mathcal{O}$ such that $d = S(\Delta) > 0$. Consider hyperconfigurations Δ_i with $1 \leq i \leq d$. It follows that:

$$\Delta \otimes \langle \Delta_1, \cdots, \Delta_d \rangle \in \mathcal{O}$$

Proof. By induction on $d \geq 1$.
Base case: d=1. This corresponds to the following:

$$\Delta \otimes \langle \Delta_1 \rangle$$

Here generalized wrapping corresponds to intercalation:

$$\Delta \otimes \langle \Delta_1 \rangle = \Delta|_1 \Delta_1 \in \mathcal{O}$$

By the previous lemma, we know that hyperconfigurations are closed under intercalation. The base case is proved.

Inductive step: given Δ_i with $1 \leq i \leq d+1$, we want to prove that:

$$\Delta \otimes \langle \Delta_1, \ldots, \Delta_{d+1} \rangle \in \mathcal{O}$$

By induction hypothesis:

$$\Delta \otimes \langle \Delta_1, \ldots, \Delta_d \rangle \in \mathcal{O}$$

Let j be the index of the $d+1$-th separator of Δ. We have that:

$$\Delta \otimes \langle \Delta_1, \ldots, \Delta_{d+1} \rangle = (\Delta \otimes \langle \Delta_1, \ldots, \Delta_d \rangle)|_j \Delta_{d+1} \in \mathcal{O}$$

By the previous lemma, since hyperconfigurations are closed under intercalation and $\Delta \otimes \langle \Delta_1, \ldots, \Delta_d \rangle \in \mathcal{O}$, we have that:

$$\Delta \otimes \langle \Delta_1, \ldots, \Delta_{d+1} \rangle \in \mathcal{O}$$

This completes the proof. □

The proofs of the closedness of operations we have presented before works also for the tree-based hypersequent syntaxes. Given the fact that $\mathcal{O}_s$ and $\mathcal{O}_t$ are in bijective correspondance and they are closed by the operations $+$ (concatenation) and $|_i$ (i-th wrapping or intercalation) ($0 < i \in \omega$) we see that the mapping $\mathcal{O}_s \xrightarrow{h} \mathcal{O}_t$ is an isomorphism of displacement algebras.

Hypercontexts for Hypersequent Syntax

We consider a set of sorted holes $(*^i_j)_{i,j\in\omega}$, which are important objects for defining the concept of hypercontext for hyperconfigurations. Sorted holes can be thought of as an extra set of types. Like types holes have sort:

$$S(*^j_i) \overset{def}{=} i, \text{ for } i, j \in \omega$$

We will give two kinds of hypercontexts, both necessary for the formulation of a calculus which absorbs the structural rule of **mD**:

- Continuous hypercontexts
- Abstract hypercontexts

(150) **Definition** (*Continuous Hypercontexts*)

Hypercontexts are defined by unambiguous mutual recursion.

$$\begin{array}{lll}
\mathcal{C} & ::= & \Lambda \\
\mathcal{C} & ::= & A, \mathcal{C} \text{ for } S(A) = 0 \\
\mathcal{C} & ::= & *^j, \mathcal{C} \text{ for } j \in \omega \\
\mathcal{C} & ::= & [\,], \mathcal{C} \\
\mathcal{C} & ::= & \sqrt[0]{A}, \mathcal{C}, \sqrt[1]{A}, \ldots, \sqrt[a-1]{A}, \mathcal{C}, \sqrt[a]{A}, \mathcal{C} \\
 & & \text{for } A \in \mathcal{F},\ a = S(A) > 0
\end{array}$$

Observe that the set $\mathcal{O}$ of hyperconfigurations is properly contained in the set $\mathcal{C}$ of hypercontexts. We will make an assumption on the holes of an arbitrary hypercontext: given a hypercontext Δ with n holes, every hole has a single occurrence. Given n hypercontexts Γ_i $(i = 1, \cdots, n)$ we represent the operation of simultaneous replacement of every hole by hypercontexts Γ_i as follows:

(151)
$$\Delta(\Gamma_1, \cdots, \Gamma_n)$$

That the operation of simultaneous replacement of hyperholes is closed in the set of hypercontexts can be justified in a completely analogous way to the one of intercalation of hyperconfigurations at a separator.

Because of this similarity of separators and hyperholes, the reader is advised:

(152) **Remark**
It is fundamental to keep in mind that the separator [] and holes are distinct objects. Usually hypercontexts Γ are hyperconfigurations. The [] is an object level expression of hypersequent syntax and of course helps to refer to the context, but at an object level. Holes have the purpose of making easier to work with and read with more commodity the rules of hypersequent calculus. Without these mathematical devices, reading and formulating hypersequent syntax would be very painful.

The results of concatenation, i-th intercalation and generalized wrapping on hyperconfigurations can be easily extended to continuous hypercontexts. Note that if Δ is a continuous hypercontext with an occurrence of a hyperhole $*_i^j$ then $\Delta = \Delta(*_i^j)$.

3.4.1 The Intermediate Hypersequent Calculus

We present now the intermediate hypersequent calculus **hD** which has two notable features: 1) there are only logical rules (there is no presence of structural rules), i.e. as claimed, the structural postulates of **mD** are absorbed, and 2) the notation of **hD** subsumes both string-based and tree-based hypersequent syntax. The formulation of the logical rules of **hD** are easily motivated by the interpretation in an arbitrary displacement algebra $\mathcal{A}$.

Consider an arbitrary type B of sort $S(B) > 0$, arbitrary hyperconfigurations Δ_i with $1 \leq i \leq S(B)$ and an arbitrary valuation v on types. We will write $[\![\cdot]\!]_v$ as $[\![\cdot]\!]$. We have that:

$$\begin{array}{l} [\![B \otimes \langle \Delta_1, \cdots, \Delta_b \rangle]\!] = \\ \{b^0 + d_1 + b^1 \cdots + b^{S(b)-1} + d_{S(B)} + b^{S(B)} : \\ b^0 + 1 + b^1 + \cdots + b^{S(B)-1} + 1 + b^{S(B)} \in [\![B]\!] \text{ and } d_i \in [\![\Delta_i]\!],\ 1 \leq i \leq S(B)\} \end{array}$$

In the following Γ, Γ_i, Θ and Θ_i will denote arbitrary hyperconfigurations or hypercontexts.

3.4.2 Axioms

$$\frac{}{\overrightarrow{A} \Rightarrow A}\, id$$

Continuous connective rules

a) / rules:
Consider the following syntactically interpreted hyperconfiguration:

$$[\![\overrightarrow{B} \otimes \langle \Delta_1, \cdots, \Delta_b \rangle]\!]$$

And the following set-theoretical inclusion:

$$[\![\Gamma]\!] \subseteq [\![A]\!]$$

Now since $[\![\overrightarrow{B/A}, \Gamma]\!] \subseteq [\![\overrightarrow{B/A}, A]\!] \subseteq [\![B]\!]$, we have that

$$[\![(\overrightarrow{B/A}, \Gamma) \otimes \langle \Delta_1, \cdots, \Delta_b \rangle]\!] \subseteq [\![\overrightarrow{B} \otimes \langle \Delta_1, \cdots, \Delta_b \rangle]\!]$$

Extending to a hypercontext Δ for $\overrightarrow{B} \otimes \langle \Delta_1, \cdots, \Delta_b \rangle$ then:

$$\frac{\text{If } [\![\Delta(\overrightarrow{B} \otimes \langle \Delta_1, \cdots, \Delta_b \rangle)]\!] \subseteq [\![C]\!] \qquad \text{and} \qquad [\![\Gamma]\!] \subseteq [\![A]\!]}{\text{Then } [\![\Delta((\overrightarrow{B/A}, \Gamma) \otimes \langle \Delta_1, \cdots, \Delta_b \rangle)]\!] \subseteq [\![C]\!]}$$

This corresponds to the **hD** left rule for the connective /. The right rule for this continuous implication has no surprises. Hence the following rules obtain soundly:

$$\frac{\Gamma \Rightarrow A \qquad \Delta(\overrightarrow{B} \otimes \langle \Delta_1, \cdots, \Delta_b \rangle) \Rightarrow C}{\Delta((\overrightarrow{B/A}, \Gamma) \otimes \langle \Delta_1, \cdots, \Delta_b \rangle) \Rightarrow C} /L \qquad \frac{\Delta, \overrightarrow{A} \Rightarrow B}{\Delta \Rightarrow B/A} /R$$

b) $\backslash$ The rules for this connective are completely similar to the previous case. Hence we have the rules:

$$\frac{\Gamma \Rightarrow A \qquad \Delta(\overrightarrow{B} \otimes \langle \Delta_1, \cdots, \Delta_b \rangle) \Rightarrow C}{\Delta((\Gamma, \overrightarrow{A\backslash B}) \otimes \langle \Delta_1, \cdots, \Delta_b \rangle) \Rightarrow C} \backslash L \qquad \frac{\overrightarrow{A}, \Delta \Rightarrow B}{\Delta \Rightarrow A\backslash B} \backslash R$$

c) $\bullet$ rules:

$$\frac{\Delta((\overrightarrow{A}, \overrightarrow{B}) \otimes \langle \Delta_1, \cdots, \Delta_{a+b} \rangle) \Rightarrow C}{\Delta(\overrightarrow{A \bullet B} \otimes \langle \Delta_1, \cdots, \Delta_{a+b} \rangle) \Rightarrow C} \bullet L \qquad \frac{\Delta \Rightarrow A \qquad \Gamma \Rightarrow B}{\Delta, \Gamma \Rightarrow A \bullet B} \bullet R$$

Discontinuous connective rules

a) $\downarrow_i$ rules ($i > 0$):

$$\frac{\Gamma \Rightarrow A \qquad \Delta(\overrightarrow{B} \otimes \langle \Delta_1, \cdots, \Delta_b \rangle) \Rightarrow C}{\Delta((\Gamma|_i \overrightarrow{A\downarrow_i B}) \otimes \langle \Delta_1, \cdots, \Delta_b \rangle) \Rightarrow C} \uparrow_i L \qquad \frac{\overrightarrow{A}|_i \Delta \Rightarrow B}{\Delta \Rightarrow A\downarrow_i B} \downarrow_i R$$

b) $\uparrow_i$ rules ($i > 0$):

$$\frac{\Gamma \Rightarrow A \qquad \Delta(\overrightarrow{B} \otimes \langle \Delta_1, \cdots, \Delta_b \rangle) \Rightarrow C}{\Delta((\overrightarrow{B\uparrow_i A}|_i \Gamma) \otimes \langle \Delta_1, \cdots, \Delta_b \rangle) \Rightarrow C} \uparrow_i L \qquad \frac{\Delta|_i \overrightarrow{A} \Rightarrow B}{\Delta \Rightarrow B\uparrow_i A} \uparrow_i R$$

c) $\odot_i$ rules ($i > 0$):

$$\frac{\Delta((\overrightarrow{A}|_i \overrightarrow{B}) \otimes \langle \Gamma_1, \cdots, \Gamma_{a+b-1} \rangle) \Rightarrow C}{\Delta(\overrightarrow{A \odot_i B} \otimes \langle \Delta_1, \cdots, \Delta_{a+b-1} \rangle) \Rightarrow C} \odot_i L \qquad \frac{\Delta \Rightarrow A \qquad \Gamma \Rightarrow B}{\Delta|_i \Gamma \Rightarrow A \odot_i B} \odot_i R$$

Units

a) Continuous unit rules:

$$\frac{\Delta(\Lambda) \Rightarrow A}{\Delta(I) \Rightarrow A} IL \qquad \frac{}{\Lambda \Rightarrow I} IR$$

b) Discontinuous unit rules:

$$\frac{\Delta([] \otimes \langle \Gamma \rangle) \Rightarrow A}{\Delta(J \otimes \langle \Gamma \rangle) \Rightarrow A} JL \qquad \frac{}{[] \Rightarrow J} JR$$

Cut rule

$$\frac{\Delta \Rightarrow A \qquad \Gamma(\overrightarrow{A} \otimes \langle \Gamma_1, \ldots, \Gamma_a \rangle) \Rightarrow B}{\Gamma(\Delta \otimes \langle \Gamma_1, \ldots, \Gamma_a \rangle) \Rightarrow B} Cut$$

We summarize the rules in Figure 3.9.

We consider the following embedding translation from **mD** to **hD**:

$$\begin{array}{rcl} (\cdot)^\sharp : \mathbf{mD} = (\mathcal{F}, \mathbf{StructTerm}, \rightarrow) & \longrightarrow & \mathbf{hD} = (\mathcal{F}, \mathcal{O}, \Rightarrow) \\ T \rightarrow A & \mapsto & (T)^\sharp \Rightarrow (A)^\sharp \end{array}$$

$(\cdot)^\sharp$ is such that:

$A^\sharp = \sqrt[0]{A}, [], \sqrt[1]{A}, \cdots, \sqrt[a-1]{A}, [], \sqrt[a]{A}$ if A is a type of at least sort 1 or
$A^\sharp = A\{[] : \cdots : []\}$ if we use the tree-based hypersequent syntax
$A^\sharp = A$ if A is of sort 0
$(T_1 \circ T_2)^\sharp = T_1^\sharp, T_2^\sharp$
$(T_1 \circ_i T_2)^\sharp = T_1^\sharp|_i T_2^\sharp$
$\mathbb{I}^\sharp = \Lambda$
$\mathbb{J}^\sharp = []$

Collapsing the structural rules

Let us see how the structural rules are absorbed in **hD**. The proof uses the string-based hypersequent syntax, but the tree-based one has a very similar proof. We show here that structural postulates of **mD** collapse into the same textual form when they are mapped through $(\cdot)^\sharp$. Later we will show that:

$$\frac{}{\overrightarrow{A} \Rightarrow A}\, id \qquad \frac{\Delta \Rightarrow A \qquad \Gamma(\overrightarrow{A} \otimes \langle \Gamma_1, \ldots, \Gamma_a \rangle) \Rightarrow B}{\Gamma(\Delta \otimes \langle \Gamma_1, \ldots, \Gamma_a \rangle) \Rightarrow B}\, Cut$$

$$\frac{\Gamma \Rightarrow A \qquad \Delta(\overrightarrow{B} \otimes \langle \Delta_1, \cdots, \Delta_b \rangle) \Rightarrow C}{\Delta((\overrightarrow{B/A}, \Gamma) \otimes \langle \Delta_1, \cdots, \Delta_b \rangle) \Rightarrow C}\, /L \qquad \frac{\Delta, \overrightarrow{A} \Rightarrow B}{\Delta \Rightarrow B/A}\, /R$$

$$\frac{\Gamma \Rightarrow A \qquad \Delta(\overrightarrow{B} \otimes \langle \Delta_1, \cdots, \Delta_b \rangle) \Rightarrow C}{\Delta((\Gamma, \overrightarrow{A\backslash B}) \otimes \langle \Delta_1, \cdots, \Delta_b \rangle) \Rightarrow C}\, \backslash L \qquad \frac{\overrightarrow{A}, \Delta \Rightarrow B}{\Delta \Rightarrow A\backslash B}\, \backslash R$$

$$\frac{\Delta((\overrightarrow{A}, \overrightarrow{B}) \otimes \langle \Delta_1, \cdots, \Delta_{a+b} \rangle) \Rightarrow C}{\Delta(\overrightarrow{A \bullet B} \otimes \langle \Delta_1, \cdots, \Delta_{a+b} \rangle) \Rightarrow C}\, \bullet L \qquad \frac{\Delta \Rightarrow A \qquad \Gamma \Rightarrow B}{\Delta, \Gamma \Rightarrow A \bullet B}\, \bullet R$$

$$\frac{\Gamma \Rightarrow A \qquad \Delta(\overrightarrow{B} \otimes \langle \Delta_1, \cdots, \Delta_b \rangle) \Rightarrow C}{\Delta((\overrightarrow{B\uparrow_i A}|_i \Gamma) \otimes \langle \Delta_1, \cdots, \Delta_b \rangle) \Rightarrow C}\, \uparrow_i L \qquad \frac{\Delta|_i \overrightarrow{A} \Rightarrow B}{\Delta \Rightarrow B\uparrow_i A}\, \uparrow_i R$$

$$\frac{\Gamma \Rightarrow A \qquad \Delta(\overrightarrow{B} \otimes \langle \Delta_1, \cdots, \Delta_b \rangle) \Rightarrow C}{\Delta((\Gamma|_i \overrightarrow{A\downarrow_i B}) \otimes \langle \Delta_1, \cdots, \Delta_b \rangle) \Rightarrow C}\, \uparrow_i L \qquad \frac{\overrightarrow{A}|_i \Delta \Rightarrow B}{\Delta \Rightarrow A\downarrow_i B}\, \downarrow_i R$$

$$\frac{\Delta((\overrightarrow{A}|_i \overrightarrow{B}) \otimes \langle \Gamma_1, \cdots, \Gamma_{a+b-1} \rangle) \Rightarrow C}{\Delta(\overrightarrow{A \odot_i B} \otimes \langle \Delta_1, \cdots, \Delta_{a+b-1} \rangle) \Rightarrow C}\, \odot_i L \qquad \frac{\Delta \Rightarrow A \qquad \Gamma \Rightarrow B}{\Delta|_i \Gamma \Rightarrow A \odot_i B}\, \odot_i R$$

$$\frac{\Delta \Rightarrow A \qquad \Gamma(\overrightarrow{A} \otimes \langle \Gamma_1, \ldots, \Gamma_a \rangle) \Rightarrow B}{\Gamma(\Delta \otimes \langle \Gamma_1, \ldots, \Gamma_a \rangle) \Rightarrow B}\, Cut$$

Figure 3.9: Intermediate hypersequent calculus **hD**

$$\text{If } T{\sim}^* S \text{ then } T^\sharp = S^\sharp$$

Moreover will see that for every $A, B, C \in \mathcal{F}$ the following hypersequents are provable in **hD**:

(153) • **Continuous associativity**

$$\overrightarrow{A \bullet (B \bullet C)} \Rightarrow (A \bullet B) \bullet C \text{ and } \overrightarrow{(A \bullet B) \bullet C} \Rightarrow A \bullet (B \bullet C)$$

- **Mixed associativity** If we have that $B \between_A C$:

$$\overrightarrow{A \odot_i (B \odot_j C)} \Rightarrow (A \odot_i B) \odot_{i+j-1} C \text{ and } \overrightarrow{(A \odot_i B) \odot_{i+j-1} C} \Rightarrow A \odot_i (B \odot_j C)$$

- **Mixed permutation** If we have that $B \prec_A C$:

$$\overrightarrow{(A \odot_i B) \odot_j C} \Rightarrow (A \odot_{j-b+1} C) \odot_i C \text{ and } \overrightarrow{(A \odot_{j-b+1} C) \odot_i C} \Rightarrow (A \odot_i B) \odot_j C)$$

If we have that $C \prec_A B$:

$$\overrightarrow{(A \odot_i B) \odot_j C} \Rightarrow (A \odot_j C) \odot_{i+c-1} C \overrightarrow{(A \odot_j C) \odot_{i+c-1} C} \Rightarrow (A \odot_i B) \odot_j C)$$

- **Split wrap**:

$$\overrightarrow{A \bullet B} \Rightarrow (A \bullet J) \odot_{a+1} B \text{ and } \overrightarrow{(A \bullet J) \odot_{a+1} B} \Rightarrow A \bullet B$$

  and:
$$\overrightarrow{(J \bullet B) \odot_1 A} \Rightarrow A \bullet B \text{ and } \overrightarrow{A \bullet B} \Rightarrow (J \bullet B) \odot_1 A$$

- **Continuous unit and discontinuous unit**:

$$\overrightarrow{A \bullet I} \Rightarrow A \text{ and } \overrightarrow{A} \Rightarrow A \bullet I \text{ and } \overrightarrow{I \bullet A} \Rightarrow A \text{ and } \overrightarrow{A} \Rightarrow I \bullet A$$

  and:

$$\overrightarrow{A \odot_i J} \Rightarrow A \text{ and } \overrightarrow{A} \Rightarrow A \odot_i J \text{ and } \overrightarrow{J \odot_1 A} \Rightarrow A \text{ and } \overrightarrow{A} \Rightarrow J \odot_1 A$$

That **hD** absorbs the rules is proved in the following theorem:

(154) **Theorem** (**hD** *Absorption of* $\mathbf{Eq_D}^*$ *Structural Rules*)

For any $T, S \in$ **StructTerm**, if $T{\sim}^*S$ then $(T)^\sharp = (S)^\sharp$.

Proof. We define a useful notation for vectorial types which will help us to prove the theorem. Where A is an arbitrary type of sort greater than 0:

(155) $$\overrightarrow{A}_i^j = \begin{cases} \sqrt[i]{A}, \text{ if } i = j \\ \overrightarrow{A}_i^{j-1}, [], \sqrt[j]{A}, \text{ if } j - i > 0 \end{cases}$$

Note that $\overrightarrow{A} = \overrightarrow{A}_0^a$. Now, consider arbitrary types A, B and C. As usual we denote their sorts respectively by a, b and c. We have then:

- Continuous associativity:

$$\begin{cases} ((A \circ B) \circ C)^\sharp & = & (\overrightarrow{A}, \overrightarrow{B}), \overrightarrow{C} = \overrightarrow{A}, \overrightarrow{B}, \overrightarrow{C} \\ (A \circ (B) \circ C))^\sharp & = & \overrightarrow{A}, (\overrightarrow{B}, \overrightarrow{C}) = \overrightarrow{A}, \overrightarrow{B}, \overrightarrow{C} \end{cases}$$

- Discontinuous associativity: Suppose that $B \between_A C$

  We have that:

$$\overrightarrow{B}|_j \overrightarrow{C} = \overrightarrow{B}_0^{i-1}, \overrightarrow{C}, \overrightarrow{B}_i^b$$
$$\overrightarrow{A}|_i (\overrightarrow{B}|_j \overrightarrow{C}) = \overrightarrow{A}_0^{i-1}, \overrightarrow{B}_0^{j-1}, \overrightarrow{C}, \overrightarrow{B}_j^b, \overrightarrow{A}_i^a$$

  On the other hand, we have that:

$$\overrightarrow{A}|_i \overrightarrow{B} = \overrightarrow{A}_0^{i-1}, \overrightarrow{B}, \overrightarrow{A}_i^a = \overrightarrow{A}_0^{i-1}, \overrightarrow{B}_0^{j-1}, \underbrace{[]}_{(i+j-1)\text{-th } []}, \overrightarrow{B}_j^b, \overrightarrow{A}_i^a$$

  It follows that:

$$(\overrightarrow{A}|_i \overrightarrow{B})|_{i+j-1} \overrightarrow{C} = \overrightarrow{A}_0^{i-1}, \overrightarrow{B}_0^{j-1}, \overrightarrow{C}, \overrightarrow{B}_j^b, \overrightarrow{A}_i^a$$

Summarizing:

$$\begin{cases} (A\circ_i(B\circ_j C))^\sharp & = & \overrightarrow{A}_0^{i-1}, \overrightarrow{B}_0^{j-1}, \overrightarrow{C}, \overrightarrow{B}_j^b, \overrightarrow{A}_i^a \\ ((A\circ_i B)\circ_{i+j-1} C)^\sharp & = & \overrightarrow{A}_0^{i-1}, \overrightarrow{B}_0^{j-1}, \overrightarrow{C}, \overrightarrow{B}_j^b, \overrightarrow{A}_i^a \end{cases}$$

Hence:

$$(A\circ_i(B\circ_j C))^\sharp = ((A\circ_i B)\circ_{i+j-1} C)^\sharp$$

For the case $(A\circ_i B)\circ_k C$, if one puts $k = i + j - 1$ one gets $j = k - i + 1$. Therefore, changing indices: we have that:

$$((A\circ_i B)\circ_j C)^\sharp = (A\circ_i(B\circ_{j-i+1} C))^\sharp$$

This ends the case of discontinuous associativity.

- Mixed permutation:

There are two cases: $B \prec_A C$ or $C \prec_A B$. We consider only the first case, i.e. $B \prec_A C$. The other case is analogous. Let us see $((A\circ_i B)\circ_j C)^\sharp$:

$$\overrightarrow{A}|_i\overrightarrow{B} = \overrightarrow{A}_0^{i-1}, \overrightarrow{B}, \overrightarrow{A}_i^{k-1}, \underbrace{[]}_{j\text{-th }[]}, \overrightarrow{A}_k^a$$

We have therefore:

$$j = k - 1 + b \text{ iff } k = j - b + 1$$

$$((A\circ_i B)\circ_j C)^\sharp = \overrightarrow{A}_0^{i-1}, \overrightarrow{B}, \overrightarrow{A}_i^{k-1}, \overrightarrow{C}, \overrightarrow{A}_k^a$$

Hence:

$$(\overrightarrow{A}\circ_{j-b+1}\overrightarrow{C})^\sharp = \overrightarrow{A}_0^{i-1}, [], \overrightarrow{A}_i^{k-1}, \overrightarrow{C}, \overrightarrow{A}_k^a$$

It follows that:

$$((A\circ_{j-b+1} C)\circ_i B)^\sharp = \overrightarrow{A}_0^{i-1}, \overrightarrow{B}, \overrightarrow{A}_i^{k-1}, \overrightarrow{C}, \overrightarrow{A}_k^a$$

Summarizing:

$$\begin{cases} ((A\circ_i B)\circ_j C)^\sharp = \overrightarrow{A}_0^{i-1}, \overrightarrow{B}, \overrightarrow{A}_i^{k-1}, \overrightarrow{C}, \overrightarrow{A}_k^a \\ ((A\circ_{j-b+1} C)\circ_i B)^\sharp = \overrightarrow{A}_0^{i-1}, \overrightarrow{B}, \overrightarrow{A}_i^{k-1}, \overrightarrow{C}, \overrightarrow{A}_k^a \end{cases}$$

Hence

$$((A\circ_i B)\circ_j C)^\sharp = ((A\circ_{j-b+1} C)\circ_i B)^\sharp$$

Putting $i = j - b + 1$ we have that $j = i + b - 1$. Hence:

$$((A \circ_i C) \circ_j B)^\sharp = ((A \circ_j C) \circ_{i+b-1} B)^\sharp$$

This ends the case of mixed permutation.

- Split-wrap:
  We have:

$$\begin{array}{lcl} ((A \circ \mathbb{J}) \circ_{a+1} B)^\sharp & = & (\overrightarrow{A}, [])|_{a+1} \overrightarrow{B} = \overrightarrow{A}, \overrightarrow{B} \\ ((\mathbb{J} \circ B) \circ_1 A)^\sharp & = & ([], \overrightarrow{B})|_1 \overrightarrow{A} = \overrightarrow{A}, \overrightarrow{B} \end{array}$$

Hence:

$$\begin{array}{lcl} ((A \circ \mathbb{J}) \circ_{a+1} B)^\sharp & = & (A \circ B)^\sharp \\ & \text{and} & \\ ((\mathbb{J} \circ B) \circ_1 A)^\sharp & = & (A \circ B)^\sharp \end{array}$$

This ends the case of split-wrap.

- Units:

$$\begin{array}{lclcl} (\mathbb{I} \circ A)^\sharp & = & \overrightarrow{A} & = & (A \circ \mathbb{I})^\sharp \\ (\mathbb{J} \circ_1 A)^\sharp & = & ([]|_1 \overrightarrow{A}) & = & \overrightarrow{A} = \overrightarrow{A}|_i[] = (A \circ_i \mathbb{J})^\sharp \end{array}$$

We recall that types play the role of variables of structural terms. Now, we have seen that structural rules for arbitrary type variables collapse into the same textual form. This result generalizes to arbitrary structural terms by simply using type substitution.

More concretely, we have proved that: if $T \sim S$ (i.e. S is the result of applying a single structural rule to T) then $T^\sharp = S^\sharp$. Suppose we have $T \sim^* S$ (we omit the trivial case $T \sim^* T$). We have then a chain:

$$T := T_1 \sim T_2 \sim \cdots \sim T_{i-1} \sim T_i =: S \text{ for } i \geq 2$$

Applying $(\cdot)^\sharp$ to each $T_k \sim T_{k+1}$ $(1 \leq k \leq i-1)$ we have proved that:

$$(T_k)^\sharp = (T_{k+1})^\sharp$$

We have therefore a chain of identities:

$$(T)^\sharp = (T_1)^\sharp = (T_2)^\sharp = \ldots = (T_i)^\sharp = (S)^\sharp$$

This completes the proof.

□

(156) **Remark**
Note that we have also proved that $\mathcal{O}_s$ is a displacement algebra. Analogously, we have then that $\mathcal{O}_t$ is a general displacement algebra. In fact, we have that the bijective mapping $\mathcal{O}_s \xrightarrow{h} \mathcal{O}_t$ is an isomorphism of general displacement algebras.

We will now prove the associativity theorems of **hD** displayed in (153). Other theorems corresponding to the structural postulates of **mD** have similar proofs.

- Continuous associativity is obvious as in the Lambek calculus. The only difference is that types are sorted and in our notation the antecedent of hypersequents have the vectorial notation.

- Discontinuous associativity: we suppose that $B \between_A C$. The following hypersequents are provable:

$$\overrightarrow{(A \odot_i B) \odot_{i+j-1} C} \Rightarrow A \odot_i (B \odot_j C)$$

  And:

$$\overrightarrow{A \odot_i (B \odot_j C)} \Rightarrow (A \odot_i B) \odot_{i+j-1} C$$

  By the previous lemma the identity $\overrightarrow{A}|_i(\overrightarrow{B}|_j\overrightarrow{C}) = (\overrightarrow{A}|_i\overrightarrow{B})|_{i+j-1}\overrightarrow{C}$ holds. We have the two following hypersequent derivations:

$$\dfrac{\dfrac{\dfrac{\dfrac{\overrightarrow{A} \Rightarrow A \qquad \dfrac{\overrightarrow{B} \Rightarrow B \qquad \overrightarrow{C} \Rightarrow C}{\overrightarrow{B}|_j\overrightarrow{C} \Rightarrow B \odot_j C}\odot_j R}{\overrightarrow{A}|_i(\overrightarrow{B}|_j\overrightarrow{C}) = (\overrightarrow{A}|_i\overrightarrow{B})|_{i+j-1}\overrightarrow{C} \Rightarrow A \odot_i (B \odot_j C)}\odot_i R}{(\overrightarrow{A}|_i\overrightarrow{B})|_{i+j-1}\overrightarrow{C} \Rightarrow A \odot_i (B \odot_j C)}}{\overrightarrow{A \odot_i B}|_{i+j-1}\overrightarrow{C} \Rightarrow A \odot_i (B \odot_j C)}\odot_i L}{\overrightarrow{(A \odot_i B) \odot_{i+j-1} C} \Rightarrow A \odot_i (B \odot_j C)}\odot_{i+j-1} L$$

  and

$$\dfrac{\dfrac{\dfrac{\dfrac{\overrightarrow{A} \Rightarrow A \qquad \overrightarrow{B} \Rightarrow B}{\overrightarrow{A}|_i\overrightarrow{B} \Rightarrow (A \odot_i B)}\odot_i R \qquad \overrightarrow{C} \Rightarrow C}{(\overrightarrow{A}|_i\overrightarrow{B})|_{i+j-1}\overrightarrow{C} = \overrightarrow{A}|_i(\overrightarrow{B}|_j\overrightarrow{C}) \Rightarrow (A \odot_i B) \odot_{i+j-1} C}\odot_{i+j-1} R}{\overrightarrow{A}|_i(\overrightarrow{B \odot_j C}) \Rightarrow (A \odot_i B) \odot_{i+j-1} C}\odot_j L}{\overrightarrow{A \odot_i (B \odot_j C)} \Rightarrow (A \odot_i B) \odot_{i+j-1} C}\odot_i L$$

3.5 Hypersequent Calculus: the abstract metanotation

In this section we present the abstract hypersequent calculus which we will use in the direct syntactic algorithmic proof of the Cut elimination theorem. As in the case of continuous hypercontexts, we will use the set of hyperholes $(*_i^j)_{i,j\in\omega}$. In order to see tree-based hyperconfigurations at work, we define abstract hypercontexts in terms of $\mathcal{O}_t$. Of course, in the case of $\mathcal{O}_s$ the results are completely analogous. The set of abstract hypercontexts will be denoted $\mathcal{C}$, although if one wanted to use a particular hypersequent syntax, the notations $\mathcal{C}_s$ and $\mathcal{C}_t$ are completely sound.

(157) **Definition** (*Abstract Hypercontexts*)

$$\begin{array}{lll}
\mathcal{C} & ::= & \Lambda \\
\mathcal{C} & ::= & A, \mathcal{C} \text{ for } S(A) = 0 \\
\mathcal{C} & ::= & *_0^j, \mathcal{C} \text{ for } j \in \omega \\
\mathcal{C} & ::= & [], \mathcal{C} \\
\mathcal{C} & ::= & A\{\underbrace{\mathcal{C}, \ldots, \mathcal{C}}_{a \text{ times}}\}, \mathcal{C} \text{ for } A \in \mathcal{F},\ a = S(A) > 0 \\
\mathcal{C} & ::= & *_i^j\{\underbrace{\mathcal{C}, \ldots, \mathcal{C}}_{i \text{ times}}\}, \mathcal{C}, \text{ where } i > 0
\end{array}$$

If we consider the set of hyperholes $(*_i^j)_{i,j \in \omega,\, i>0}$ as an extra set of types it is easy to see (mimicking the proof for hyperconfigurations) that $\mathcal{C}$ is closed under concatenation, intercalation, generalized wrapping and substitution of $*_i^j$ by a hypercontext. Like in the case of continuous hypercontexts we adopt the convention that every hyperhole has a linear occurrence. Notice here that hyperholes occur in a very different way from the case of continuous hyperholes, because they only appear either as hyperholes of sort 0 or hyperholes of sort greater than 0 which are wrapping material. The last BNF definition of (157) is the key to define the abstract formulation of hypersequent calculus. This gives a final version of hypersequent syntax that is very close (in its formulation) to the one of the sequent calculus for **L**. Finally, as claimed, this abstract formulation of hypersequent syntax for **hD** will enable us to make a syntactic proof of the Cut Elimination theorem following the same strategy which Lambek (1958) used to prove Cut elimination for his calculus **L**.

From the intermediate hypersequent syntax to the abstract hypersequent syntax

Let us consider a hyperconfiguration $\Gamma = \Delta(\overrightarrow{A} \otimes \langle \Delta_1, \cdots, \Delta_a \rangle)$ where $a = S(A)$. Δ is a hypercontext with an occurrence of a continuous sorted hyperhole, say $*_i^j$ where $i = S(\overrightarrow{A} \otimes \langle \Delta_1, \cdots, \Delta_a \rangle)$. The notation with parentheses signals that the distinguished occurrence is continuous. Let $*_a^k$ be a hyperhole such that $S(*_a^k) = a$. Let us replace the occurrence of $\overrightarrow{A}$ in Γ by our hyperhole. We get:

$$\Delta(*_a^k \otimes \langle \Delta_1, \cdots, \Delta_a \rangle) \qquad (\star)$$

As a convention we write the expression in $(\star)$ as follows:

(158) $\Delta\langle *_a^k \rangle$

The angle brackets in (158) indicate that any substitution of $*_a^k$ by an arbitrary hyperconfiguration Θ of sort a will wrap (i.e. $\otimes$) the material $\langle \Delta_1, \cdots, \Delta_a \rangle$. For example, we can replace $*_a^k$ by $\overrightarrow{A}$:

$$\Delta\langle \overrightarrow{A} \rangle$$

$\Delta\langle \overrightarrow{A} \rangle$ is identical to $\Delta(\overrightarrow{A} \otimes \langle \Delta_1, \cdots, \Delta_a \rangle)$. Notice that we have the identity $\Delta = \Delta\langle *_a^k \rangle$. Now, $\Delta\langle *_a^k \rangle$ is quite different from $\Delta(*_i^k)$. In $\Delta\langle \overrightarrow{A} \rangle$ there is the hidden presence of the material $\langle \Delta_1, \cdots, \Delta_a \rangle$, while in $\Delta(*_i^j)$ this is not the case. Let us remark that the use of angle brackets has as purpose to discern when we are dealing with a continuous hypercontext or an abstract hypercontext. Finally, as useful *abus de langage*, the distinguished occurrence of a hyperconfiguration of sort 0 is also signalled with the angle brackets notation. In this sense abstract hypercontexts subsume continuous hypercontexts.

We are now in a position to formulate the abstract hypersequent calculus:

- Continuous connectives:

$$\frac{\Gamma \Rightarrow A \qquad \Delta(\overrightarrow{B} \otimes \langle \Delta_1, \cdots, \Delta_b \rangle) \Rightarrow C}{\Delta((\overrightarrow{B/A}, \Gamma) \otimes \langle \Delta_1, \cdots, \Delta_b \rangle) \Rightarrow C}\,/L \qquad \frac{\Delta, \overrightarrow{A} \Rightarrow B}{\Delta \Rightarrow B/A}\,/R$$

$$\rightsquigarrow$$

$$\frac{\Gamma \Rightarrow A \qquad \Delta\langle \overrightarrow{B} \rangle \Rightarrow C}{\Delta\langle \overrightarrow{B/A}, \Gamma \rangle \Rightarrow C}\,/L \qquad \frac{\Delta, \overrightarrow{A} \Rightarrow B}{\Delta \Rightarrow B/A}\,/R$$

The case of $\backslash$ is completely similar to the one of $/$. The continuous product abstract formulation is as follows:

$$\frac{\Delta((\overrightarrow{A}, \overrightarrow{B}) \otimes \langle \Delta_1, \cdots, \Delta_{a+b} \rangle) \Rightarrow C}{\Delta(\overrightarrow{A \bullet B} \otimes \langle \Delta_1, \cdots, \Delta_{a+b} \rangle) \Rightarrow C}\,\bullet L \qquad \frac{\Delta \Rightarrow A \qquad \Gamma \Rightarrow B}{\Delta, \Gamma \Rightarrow A \bullet B}\,\bullet R$$

$$\rightsquigarrow$$

$$\frac{\Delta\langle \overrightarrow{A}, \overrightarrow{B} \rangle \Rightarrow C}{\Delta\langle \overrightarrow{A \bullet B} \rangle \Rightarrow C}\,\bullet L \qquad \frac{\Delta \Rightarrow A \qquad \Gamma \Rightarrow B}{\Delta, \Gamma \Rightarrow A \bullet B}\,\bullet R$$

- Discontinuous connectives:

$$\frac{\Gamma \Rightarrow A \qquad \Delta(\overrightarrow{B} \otimes \langle \Delta_1, \cdots, \Delta_b \rangle) \Rightarrow C}{\Delta((\overrightarrow{B\uparrow_i A}|_i \Gamma) \otimes \langle \Delta_1, \cdots, \Delta_b \rangle) \Rightarrow C}\,\uparrow_i L \qquad \frac{\Delta|_i \overrightarrow{A} \Rightarrow B}{\Delta \Rightarrow B\uparrow_i A}\,\uparrow_i R$$

$$\rightsquigarrow$$

$$\frac{\Gamma \Rightarrow A \qquad \Delta\langle \overrightarrow{B} \rangle \Rightarrow C}{\Delta\langle \overrightarrow{B\uparrow_i A}|_i \Gamma \rangle \Rightarrow C}\,\uparrow_i L \qquad \frac{\Delta|_i \overrightarrow{A} \Rightarrow B}{\Delta \Rightarrow B\uparrow_i A}\,\uparrow_i R$$

The case of $\downarrow_i$ is very similar to the case of $\uparrow_i$. Let us formulate the rules for the discontinuous product $\odot_i$:

$$\frac{\Delta((\overrightarrow{A}|_i \overrightarrow{B}) \otimes \langle \Gamma_1, \cdots, \Gamma_{a+b-1} \rangle) \Rightarrow C}{\Delta(\overrightarrow{A \odot_i B} \otimes \langle \Delta_1, \cdots, \Delta_{a+b-1} \rangle) \Rightarrow C}\,\odot_i L \qquad \frac{\Delta \Rightarrow A \qquad \Gamma \Rightarrow B}{\Delta|_i \Gamma \Rightarrow A \odot_i B}\,\odot_i R$$

$$\rightsquigarrow$$

$$\frac{\Delta\langle \overrightarrow{A}|_i \overrightarrow{B} \rangle \Rightarrow C}{\Delta\langle \overrightarrow{A \odot_i B} \rangle \Rightarrow C}\,\odot_i L \qquad \frac{\Delta \Rightarrow A \qquad \Gamma \Rightarrow B}{\Delta|_i \Gamma \Rightarrow A \odot_i B}\,\odot_i R$$

- Units. Continuous unit:

$$\frac{\Delta(\Lambda) \Rightarrow A}{\Delta(I) \Rightarrow A} IL \qquad \frac{}{\Lambda \Rightarrow I} IR$$

$$\rightsquigarrow$$

$$\frac{\Delta\langle\Lambda\rangle \Rightarrow A}{\Delta\langle I\rangle \Rightarrow A} IL \qquad \frac{}{\Lambda \Rightarrow I} IR$$

The discontinuous unit abstract formulation is as follows:

$$\frac{\Delta([] \otimes \langle\Gamma\rangle) \Rightarrow A}{\Delta(J \otimes \langle\Gamma\rangle) \Rightarrow A} JL \qquad \frac{}{[] \Rightarrow J} JR$$

$$\rightsquigarrow$$

$$\frac{\Delta\langle[]\rangle \Rightarrow A}{\Delta\langle\overrightarrow{J}\rangle \Rightarrow A} JL \qquad \frac{}{[] \Rightarrow J} JR$$

- Finally the Cut rule:

$$\frac{\Delta \Rightarrow A \qquad \Gamma(\overrightarrow{A} \otimes \langle\Gamma_1, \dots, \Gamma_a\rangle) \Rightarrow B}{\Gamma(\Delta \otimes \langle\Gamma_1, \dots, \Gamma_a\rangle) \Rightarrow B} Cut$$

$$\rightsquigarrow$$

$$\frac{\Delta \Rightarrow A \qquad \Gamma\langle\overrightarrow{A}\rangle \Rightarrow B}{\Gamma\langle\Delta\rangle \Rightarrow B} Cut$$

3.6 The Faithful Embedding Translation Theorem between mD and hD

We recall from section 3.4.1 the mapping from **mD** to **hD**:

$$\begin{array}{rcl} (\cdot)^\sharp : \mathbf{mD} = (\mathbf{StructTerm}, \mathcal{F}, \rightarrow) & \longrightarrow & \mathbf{hD} = (\mathcal{O}, \mathcal{F}, \Rightarrow) \\ T \rightarrow A & \mapsto & (T)^\sharp \Rightarrow (A)^\sharp \end{array}$$

Consider the following two sets:

$$\begin{array}{rcl} V_D \cup \{1\} & = & (a_{ij}^k)_{i,j,k\in\omega} \text{ with } _{k\leq i} \cup \{1\} \\ \mathcal{SF} \cup \{[]\} & = & (\sqrt[k]{A_{ij}})_{i,j,k\in\omega} \text{ with } _{k\leq i} \cup \{[]\} \end{array}$$

We want to prove that **mD** and **hD** are in fact equivalent. One has structural rules while the other one does not have.

$$\begin{array}{rcl} g : V_D \cup \{1\} & \longrightarrow & \mathcal{SF} \cup \{[]\} \\ a_{ij}^k & \mapsto & \sqrt[k]{A_{ij}} \\ 1 & \mapsto & [] \end{array}$$

Let us define the following two displacement algebras freely generated by the previous sets $V_D \cup \{1\}$ and $\mathcal{SF} \cup \{[]\}$:

$$\begin{array}{rcl}\mathcal{D}_1 & = & \langle\langle V_D \cup \{1\}\rangle, +, (\times_i)_{i>0}, \Lambda, 1\rangle \\ \mathcal{D}_2 & = & \langle\langle \mathcal{SF} \cup \{[]\}\rangle, (\cdot,\cdot), (|_i)_{i>0},, \Lambda, []\rangle\end{array}$$

g induces a morphism of displacement algebras g^*:

$$\begin{array}{ccc} g^* : \mathcal{D}_1 & \longrightarrow & \mathcal{D}_2 \\ \Lambda & \mapsto & \Lambda \\ 1 & \mapsto & [] \\ a_{ij}^k & \mapsto & \sqrt[k]{A_{ij}} \\ r + s & \mapsto & g^*(r), g^*(s) \\ r \times_i s & \mapsto & g^*(r)|_i g^*(s) \end{array}$$

We now formulate a lemma which clarifies the relations between these displacement algebras and the four mappings $\llbracket\cdot\rrbracket, (\cdot)^\sharp, f^*$ and g^*.

(159) **Lemma**
The following diagram commutes:

$$\begin{array}{ccc} T_{\Sigma_D}[X] & \xrightarrow{f^*} & \textbf{StructTerm} \\ \llbracket\cdot\rrbracket \downarrow & & \downarrow (\cdot)^\sharp \\ \mathcal{D}_1 & \xrightarrow[g^*]{} & \mathcal{D}_2 \end{array}$$

Proof. We must prove that given an arbitrary $T_{\Sigma_D}[X]$ term r we have that:

$$g^*(\llbracket r\rrbracket) = (f^*(r))^\sharp$$

We proceed by induction on the structure of the term t:

- Constants:

$$g^*(\llbracket t\rrbracket) = g^*(0) = \Lambda = (f^*(r))^\sharp = (\mathbb{I})^\sharp \quad \text{if } t = 0$$

$$g^*(\llbracket t\rrbracket) = g^*(1) = [] = (f^*(r))^\sharp = (\mathbb{J})^\sharp \quad \text{if } t = 1$$

- Variables:

$$g^*(\llbracket x_{ij}\rrbracket) = g^*(\overrightarrow{a_{ij}}) = \overrightarrow{A_{ij}}$$

$$(f^*(x_{ij}))^\sharp = (A_{ij})^\sharp = \overrightarrow{A_{ij}}$$

It follows that $g^*(\llbracket x_{ij}\rrbracket) = (f^*(x_{ij}))^\sharp$.

- If $t = r \oplus s$:

$g^*([\![t]\!]) = g^*([\![r \oplus s]\!]) = g^*([\![r]\!] + [\![s]\!]) = g^*([\![r]\!]), g^*([\![s]\!])$

$(f^*(r \oplus s))^\sharp = (f^*(r) \circ f^*(s))^\sharp = (f^*(r))^\sharp, (f^*(r))^\sharp$

By induction hypothesis (i.h.) we have that:

$$g^*([\![r]\!]) = (f^*(r))^\sharp$$
$$g^*([\![s]\!]) = (f^*(s))^\sharp$$

Hence we have that $g^*([\![r \oplus s]\!]) = (f^*(r \oplus s))^\sharp$.

- If $t = r \otimes_i s$:

$g^*([\![r \otimes_i s]\!]) = g^*([\![r]\!] \times_i [\![s]\!]) = g^*([\![r]\!])|_i g^*([\![r]\!])$

$(f^*(r \otimes_i s))^\sharp = (f^*(r) \circ_i f^*(s))^\sharp = (f^*(r))^\sharp|_i (f^*(r))^\sharp$

By i.h. we have that:

$$g^*([\![r]\!]) = (f^*(r))^\sharp$$
$$g^*([\![s]\!]) = (f^*(s))^\sharp$$

Hence we have that $g^*([\![r \otimes_i s]\!]) = (f^*(r \otimes_i s))^\sharp$. This completes the proof.

□

It is easy to see that for any $r, s \in T_{\Sigma_D}[X]$ we have that:

(160)
$$\mathbf{Eq_D} \vdash r \approx s \text{ iff } \mathbf{Eq_D}^* \vdash f^*(r) \sim^* f^*(s)$$

We see then that as expected, $\mathbf{Eq_D}$ and $\mathbf{Eq_D}^*$ are very close. We can therefore benefit from the results on $\mathbf{Eq_D}$ in Chapter 2. The concept of normal form defined in Chapter 2 and the corresponding associated results can be *exported* to the equational theory $\mathbf{Eq_D}^*$. The following theorem is a witness of what we have exposed:

(161) **Theorem** (*Equivalence Theorem for* **StructTerm**)

Let R and S be arbitrary structural terms. The following holds:

$$R \sim^* S \text{ iff } (R)^\sharp = (S)^\sharp$$

Proof. Let $r, s \in T_{\Sigma_D}[X]$ such that $r := f^{*-1}(R)$ and $s := f^{*-1}(S)$. We have that:

$$\begin{array}{lll} (R)^\sharp = (S)^\sharp & \text{iff} & [\![r]\!] = [\![s]\!] \ (\star) \\ & \text{iff} & \mathbf{Eq_D} \vdash r \approx s \text{ (Equivalence theorem for } T_{\Sigma_D}[X]) \\ & \text{iff} & \mathbf{Eq_D}^* \vdash R \sim^* S \end{array}$$

Now, $(\star)$ holds for we have that by lemma (159):

$$\begin{array}{rcccl} g^*(\llbracket r \rrbracket) & = & (f^*(r))^\sharp & = & (f^*({f^*}^{-1}(R)))^\sharp = (R)^\sharp \\ g^*(\llbracket s \rrbracket) & = & (f^*(s))^\sharp & = & (f^*({f^*}^{-1}(S)))^\sharp = (S)^\sharp \end{array}$$

Since g^* is bijective we have that:

$$(R)^\sharp = g^*(\llbracket r \rrbracket) = g^*(\llbracket s \rrbracket) = (S)^\sharp \text{ iff } \llbracket r \rrbracket = \llbracket s \rrbracket$$

Hence $(\star)$ is proved. This completes the proof. □

(162) **Lemma** ($(\cdot)^\sharp$ *is an Epimorphism*)

For every $\Delta \in \mathcal{O}$ there exists a structural term[8] T_Δ such that:

$$(T_\Delta)^\sharp = \Delta$$

Proof. This can be proved by induction on the structure of hyperconfigurations, say the tree-based hyperconfigurations. We define recursively T_Δ such that $(T_\Delta)^\sharp = \Delta$:

- Case $\Delta = \Lambda$ (the empty tree): $T_\Delta = \mathbb{I}$.
- Case where $\Delta = A, \Gamma$: $T_\Delta = A \circ T_\Gamma$, where by induction hypothesis (i.h.) $(T_\Gamma)^\sharp = \Gamma$.
- Case where $\Delta = [], \Gamma$: $T_\Delta = \mathbb{J} \circ T_\Gamma$, where by i.h. $(T_\Gamma)^\sharp = \Gamma$.
- Case $\Delta = A\{\Delta_1 : \cdots : \Delta_a\}, \Delta_{a+1}$. By i.h. we have:

$$(T_{\Delta_i})^\sharp = \Delta_i \text{ for } 1 \leq i \leq a+1$$

$$T_\Delta = (A \circ_1 T_{\Delta_1}) \circ T_{\Delta_2} \text{ if } a = 1$$
$$T_\Delta = (\cdots((A \circ_1 T_{\Delta_1}) \circ_{1+d_1} T_{\Delta_2}) \cdots) \circ_{1+d_1+\cdots+d_{a-1}} T_{\Delta_{a+1}} \text{ if } a > 1$$

□

By induction on the structure of **StructTerm**, we have the following intuitive result on the relationship between structural contexts and hypercontexts:

(163) $(T[S])^\sharp = T^\sharp \langle S^\sharp \rangle$

These two technical results we have seen above are necessary for the proof of the faithful embedding translation $(\cdot)^\sharp$ of theorem (165). We prove now an important theorem which is crucial for the mentioned theorem (165). This theorem could be proved via the commutative diagram of lemma (159), but we think it is informative to prove it directly with the help of the equivalence theorem for $T_{\mathbf{\Sigma_D}}[X]$ (theorem (161)).

[8]In fact there exists an infinite set of such structural terms.

(164) **Theorem** (*Visibility for Extraction in* **StructTerm**)

Let $T[A]$ be a structural term with a linear occurrence of type A. Suppose that:

$$(T[A])^\sharp = \Delta|_i\overrightarrow{A}$$

where $\Delta \in \mathcal{O}$[9] and $A \in \mathcal{F}$. Then A is visible for extraction in $T[A]$, i.e. there exist a structural term T' and an index i such that:

$$T \sim^* T' \circ_i A$$

Proof. Let T_Δ be a structural term such that $(T_\Delta)^\sharp = \Delta$. This is possible by lemma (162). We have $(T_\Delta \circ_i A)^\sharp = \Delta|_i\overrightarrow{A}$. We have then $(T_\Delta \circ_i A)^\sharp = (T[A])^\sharp$. By the equivalence theorem (161) it follows that $T[A] \sim^* T_\Delta \circ_i A$. Put $T' := T_\Delta$. We are done. □

This result easily extends to the non-linear A's occurrence case with the help of the notion of *distinguished* occurrence of a term. This theorem will be crucial for the proof of the $(\cdot)^\sharp$ embedding theorem (165).

(165) **Theorem** (*Faithfulness of* $(\cdot)^\sharp$ *Embedding Translation Theorem*)

Let A, X and Δ be respectively a type, a structural term and a hyperconfiguration. The following statements hold:

i) If $\vdash_{\mathbf{mD}} X \rightarrow A$ then $\vdash_{\mathbf{hD}} (X)^\sharp \Rightarrow A$

ii) For any X such that $(X)^\sharp = \Delta$, if $\vdash_{\mathbf{hD}} \Delta \Rightarrow A$ then $\vdash_{\mathbf{mD}} X \rightarrow A$

Proof.

i) Logical rules in **mD** translate without any problem to **hD**. We need recall only that if X and Y are structural terms then $(X \circ Y)^\sharp = (X)^\sharp, (Y)^\sharp$ and $(X \circ_i Y)^\sharp = (X)^\sharp|_i(Y)^\sharp$. Structural rules in **mD** collapse in the same textual form as theorem (154) proves. Finally, the Cut rule has no surprise. This proves i).

ii) This part of the theorem becomes easy if we use the following four facts:

- Lemma (162) which states that for any hyperconfiguration Δ there is a structural term T_Δ such that $(T_\Delta)^\sharp = \Delta$.
- The fact (163) we stated before which gives the relationship between structural terms and hypercontexts $(T[A])^\sharp = T^\sharp\langle\overrightarrow{A}\rangle$.
- Theorem (161).
- Theorem (164).

[9] As usual, Δ could be a string-based or a tree-based hyperconfiguration.

The proof is by induction on the length of **hD** derivations. The three first facts prove the induction of all the rules but the right rule of the connectives $\uparrow_i$. Suppose the last rule of a **hD** derivation is $\uparrow_i R$:

$$\frac{\Delta|_i\overrightarrow{A} \Rightarrow B}{\Delta \Rightarrow B\uparrow_i A}\uparrow_i R$$

Let $T[A]$ be such that $(T[A])^\sharp = \Delta|_i\overrightarrow{A}$. We know by induction hypothesis that $\vdash_{\mathbf{mD}} T[A] \Rightarrow B$. By the last fact of above, i.e. theorem (164) of visibility of extraction, since $(T[A])^\sharp = \Delta|_i\overrightarrow{A}$, we know there exist T' and i such that $T[A]{\sim}^* T'\circ_i A$. It follows that in **mD**:

$$\frac{\dfrac{T[A] \to B}{\vdots \quad \textbf{Sequence of structural rules}}}{\dfrac{T'\circ_i A \to B}{T' \to B\uparrow_i A}\uparrow_i R}$$

Hence, $\vdash_{\mathbf{mD}} T' \to B\uparrow_i A$. And T' is in fact T_Δ, and therefore $(T')^\sharp = \Delta$. Moreover, for any S such that $(S)^\sharp{\sim}^* T'$, we have that applying a finite number of structural rules we obtain the **mD** provable sequent $S \to B\uparrow_i A$, and of course $(S)^\sharp = \Delta$. This completes the proof of ii).

□

We define now what we call the *type equivalent* of a hyperconfiguration, which will be very useful for the next chapter:

(166) **Definition** (*Type Equivalent of a Hyperconfiguration*)

The *type equivalent* $\Delta^\bullet$ of an arbitrary hyperconfiguration Δ is defined by recursion on the structure of hyperconfigurations as follows:

- $\Lambda^\bullet \overset{def}{=} I$.
- Given a type A of sort 0 and a hyperconfiguration Γ, if $\Gamma \neq \Lambda$ then $(A,\Gamma)^\bullet \overset{def}{=} A \bullet \Gamma^\bullet$, otherwise $(A,\Gamma)^\bullet \overset{def}{=} A$.
- Given a hyperconfiguration Γ, if $\Gamma \neq \Lambda$ then $([],\Gamma)^\bullet \overset{def}{=} J \bullet \Gamma^\bullet$, otherwise $([],\Gamma)^\bullet \overset{def}{=} J$.
- Suppose $\Gamma_1, \cdots, \Gamma_{a+1}$ are hyperconfigurations. Let $i_1 < i_2 < \cdots < i_k$ (with $1 \leq i_j \leq a$) be such that $\Gamma_{i_k} \neq \Lambda$. If $\Gamma_{a+1} \neq \Lambda$ then $(A\{\Gamma_1 : \cdot : \Gamma_a\},\Gamma_{a+1})^\bullet \overset{def}{=} ((\cdots(A \odot_{i_1} \Gamma_{i_1}^\bullet)\cdots) \odot_{1+i_1+\cdots+i_{k_1}} \Gamma_{i_k}^\bullet) \bullet \Gamma_{a+1}^\bullet$. If $\Gamma_{a+1} = \Lambda$ then $(A\{\Gamma_1 : \cdot : \Gamma_a\},\Gamma_{a+1})^\bullet \overset{def}{=} ((\cdots(A \odot_{i_1} \Gamma_{i_1}^\bullet)\cdots) \odot_{1+i_1+\cdots+i_{k_1}} \Gamma_{i_k}^\bullet$. If $\Gamma_1 = \cdots = \Gamma_a = \Lambda$ and $\Gamma_{a+1} \neq \Lambda$ then $(A\{\Gamma_1 : \cdot : \Gamma_a\},\Gamma_{a+1})^\bullet \overset{def}{=} A \bullet \Gamma_{a+1}^\bullet$. Finally, if all Γ_i are equal to Λ then $(A\{\Gamma_1 : \cdot : \Gamma_a\},\Gamma_{a+1})^\bullet \overset{def}{=} A$.

3.7 Extension of D with the Additive Conjunction and Disjunction of Linear Logic

We consider the extension of **D** to a more powerful calculus from the logical and linguistic point of view: the linear logic additive conjunction and disjunction (Girard (1987)). The extended system is called **DA**. The set of types $\mathcal{F}$ of **DA** is extended as follows:

(167) $\mathcal{F}_i ::= \mathcal{F}_i \& \mathcal{F}_i \mid \mathcal{F}_i \oplus \mathcal{F}_i$

$$[\![A\&B]\!]_v = \{s : s \in [\![A]\!]_v \text{ and } s \in [\![B]\!]_v\}$$

$$[\![A\oplus B]\!]_v = \{s : s \in [\![A]\!]_v \text{ or } s \in [\![B]\!]_v\}$$

Figure 3.10: Standard interpretation of linear logic additive connectives in **DA**

Their interpretation in a standard displacement algebra is showed in Figure 3.10. The new type type-constructors adhere to the principle of well-sorted inhabitation (114) as is clear from the BNF definition (167). Figures 3.11, 3.12 and 3.13 display the rules of the new connectives added to **D**.

The notations for the categorical, multimodal and hypersequent calculi are respectively **cDA**, **mDA** and **hDA**. The theorems (133) and (165) which state the faithfulness of the embedding translations from the categorical calculus **cD** to the multimodal calculus **mD** and $(\cdot)^\sharp$, which is an embedding translation from **mD** to **hD**, can be extended straightforwardly to **DA**.

3.8 Synthetic Connectives

We define the following synthetic (see Girard (2006)) connectives. These connectives are classified in two groups: the deterministic and the nondeterministic synthetic connectives.

$$\frac{A \to C}{A\&B \to C}\,\&L_1 \qquad \frac{B \to C}{A\&B \to C}\,\&L_2$$

$$\frac{A \to B \qquad A \to C}{A \to B\&C}\,\&R$$

$$\frac{A \to C \qquad B \to C}{A\oplus B \to C}\,\oplus L$$

$$\frac{A \to C}{A \to B\oplus C}\,\oplus R_1 \qquad \frac{A \to B}{A \to B\oplus C}\,\oplus R_2$$

Figure 3.11: Categorical calculus **cDA**: the additive connectives

$$\frac{\Gamma[A] \Rightarrow C}{\Gamma[A\&B] \Rightarrow C}\&L_1 \qquad \frac{\Gamma[B] \Rightarrow C}{\Gamma[A\&B] \Rightarrow C}\&L_2$$

$$\frac{\Gamma \Rightarrow A \qquad \Gamma \Rightarrow B}{\Gamma \Rightarrow A\&B}\&R$$

$$\frac{\Gamma[A] \Rightarrow C \qquad \Gamma[B] \Rightarrow C}{\Gamma[A\oplus B] \Rightarrow C}\oplus L$$

$$\frac{\Gamma \Rightarrow A}{\Gamma \Rightarrow A\oplus B}\oplus R_1 \qquad \frac{\Gamma \Rightarrow B}{\Gamma \Rightarrow A\oplus B}\oplus R_2$$

Figure 3.12: Multimodal **mDA**: the additive connectives

$$\frac{\Gamma\langle\overrightarrow{A}\rangle \Rightarrow C}{\Gamma\langle\overrightarrow{A\&B}\rangle \Rightarrow C}\&L_1 \qquad \frac{\Gamma\langle\overrightarrow{B}\rangle \Rightarrow C}{\Gamma\langle\overrightarrow{A\&B}\rangle \Rightarrow C}\&L_2$$

$$\frac{\Gamma \Rightarrow A \qquad \Gamma \Rightarrow B}{\Gamma \Rightarrow A\&B}\&R$$

$$\frac{\Gamma\langle\overrightarrow{A}\rangle \Rightarrow C \qquad \Gamma\langle\overrightarrow{B}\rangle \Rightarrow C}{\Gamma\langle\overrightarrow{A\oplus B}\rangle \Rightarrow C}\oplus L$$

$$\frac{\Gamma \Rightarrow A}{\Gamma \Rightarrow A\oplus B}\oplus R_1 \qquad \frac{\Gamma \Rightarrow B}{\Gamma \Rightarrow A\oplus B}\oplus R_2$$

Figure 3.13: Hypersequent **hDA**: the additive connectives

$$
\begin{array}{llll}
\triangleleft A & \stackrel{def}{=} & A \bullet J & \text{left injection} \\
\triangleleft^{-1} A & \stackrel{def}{=} & A / J & \text{left projection} \\
\triangleright A & \stackrel{def}{=} & J \bullet A & \text{right injection} \\
\triangleright^{-1} A & \stackrel{def}{=} & J \backslash A & \text{right injection} \\
\hat{}^{i} A & \stackrel{def}{=} & A \odot_i I & i\text{-th bridge} \\
\check{}^{i} A & \stackrel{def}{=} & A \uparrow_i I & i\text{-th split}
\end{array}
$$

(168)
$$
\begin{array}{llll}
A \circledcirc B & \stackrel{def}{=} & \bigoplus_{i=1,\cdots,a} A \odot_i B & \text{nondeterministic product} \\
B \Uparrow A & \stackrel{def}{=} & B \uparrow_1 A \,\&\, \cdots \,\&\, B \uparrow_{b-a+1} A & \text{nondeterministic extract} \\
A \Downarrow B & \stackrel{def}{=} & A \downarrow_1 B \,\&\, \cdots \,\&\, A \downarrow_{b-a+1} B & \text{nondeterministic infix} \\
\\
\hat{} A & \stackrel{def}{=} & \bigoplus_{i=1,\cdots,a} A \odot_i I & \text{nondeterministic bridge} \\
\check{} B & \stackrel{def}{=} & B \uparrow_1 I \,\&\, \cdots \,\&\, B \uparrow_{b-a+1} I & \text{nondeterministic split}
\end{array}
$$

From a logical point of view, synthetic connectives abbreviate derivations mainly in sequent systems. They form new connectives with left and right sequent rules. Using a linear logic slogan, synthetic connectives help to eliminate some *bureaucracy* in cut-free proofs and in the (syntactic) Cut elimination algorithms. But interestingly, from the linguistic side these defined connectives will turn out to be very useful (see Chapter 6).

We add these connectives to the categorical calculus as logical equivalences (see Figure 3.14). By these equivalences it is readily seen that the unary synthetic connectives form residuated pairs, and the binary synthetic connectives form residuated triples. We call **DA** extended with the synthetic connectives (168) **DADND**. **DND** designates the calculus with the synthetic connectives but without the additive connectives. Later we will see an interesting faithful embedding translation between **hDND** and **hDADND**

$$
\begin{array}{lll}
\triangleleft A & \leftrightarrow & A \bullet J \\
\triangleleft^{-1} A & \leftrightarrow & A / J \\
\triangleright A & \leftrightarrow & J \bullet A \\
\triangleright^{-1} A & \leftrightarrow & J \backslash A \\
\hat{}^{i} A & \leftrightarrow & A \odot_i I \\
\check{}^{i} A & \leftrightarrow & A \uparrow_i I \\
\\
A \circledcirc B & \leftrightarrow & \bigoplus_{i=1,\cdots,a} A \odot_i B \\
B \Uparrow A & \leftrightarrow & B \uparrow_1 A \,\&\, \cdots \,\&\, B \uparrow_{b-a+1} A \\
A \Downarrow B & \leftrightarrow & A \downarrow_1 B \,\&\, \cdots \,\&\, A \downarrow_{b-a+1} B
\end{array}
$$

Figure 3.14: Deterministic and nondeterministic Synthetic Connectives

3.8.1 Deterministic Synthetic Rules

We show in Figure 3.15 the hypersequent calculus rules for the deterministic synthetic connectives.

$$\frac{\Gamma\langle\overrightarrow{A}\rangle \Rightarrow B}{\Gamma\langle\overrightarrow{\triangleleft^{-1}A}, []\rangle \Rightarrow B}\triangleleft^{-1}L \qquad \frac{\Gamma, [] \Rightarrow A}{\Gamma \Rightarrow \triangleleft^{-1}A}\triangleleft^{-1}R$$

$$\frac{\Gamma\langle\overrightarrow{A}, []\rangle \Rightarrow B}{\Gamma\langle\overrightarrow{\triangleleft A}, []\rangle \Rightarrow B}\triangleleft L \qquad \frac{\Gamma \Rightarrow A}{\Gamma, [] \Rightarrow \triangleleft A}\triangleleft R$$

$$\frac{\Gamma\langle\overrightarrow{A}\rangle \Rightarrow B}{\Gamma\langle [], \overrightarrow{\triangleright^{-1}A}\rangle \Rightarrow B}\triangleright^{-1}L \qquad \frac{[], \Gamma \Rightarrow A}{\Gamma \Rightarrow \triangleright^{-1}A}\triangleright^{-1}R$$

$$\frac{\Gamma\langle [], \overrightarrow{A}\rangle \Rightarrow B}{\Gamma\langle\overrightarrow{\triangleright A}, []\rangle \Rightarrow B}\triangleright L \qquad \frac{\Gamma \Rightarrow A}{[], \Gamma \Rightarrow \triangleright A}\triangleright R$$

$$\frac{\Delta\langle\overrightarrow{B}\rangle \Rightarrow C}{\Delta\langle\overrightarrow{\check{}^{i}B}|_i\Lambda\rangle \Rightarrow C}\check{}^{i}L \qquad \frac{\Delta|_i\Lambda \Rightarrow BB}{\Delta \Rightarrow \check{}^{i}B}\check{}R$$

$$\frac{\Delta\langle\overrightarrow{B}|_i\Lambda\rangle \Rightarrow C}{\Delta\langle\overrightarrow{\hat{}^{i}B}\rangle \Rightarrow \hat{}C}\hat{}^{i}L \qquad \frac{\Delta \Rightarrow B}{\Delta|_i\Lambda \Rightarrow \hat{}B}\hat{}R$$

Figure 3.15: Hypersequent rules for deterministic synthetic connectives

(169) **Lemma** (*Elementary Syntactic Results for Deterministic Connectives in* **hD**)

The following hypersequents are provable in **hD**:

i) $\vdash_{\mathbf{hD}} \overrightarrow{\triangleleft A} \Rightarrow A \bullet J$

ii) $\vdash_{\mathbf{hD}} \overrightarrow{A \bullet J} \rightarrow \triangleleft A$

iii) $\vdash_{\mathbf{hD}} \overrightarrow{\triangleleft^{-1} A} \Rightarrow A/J$

iv) $\vdash_{\mathbf{hD}} \overrightarrow{A/J} \Rightarrow \triangleleft^{-1} A$

v) $\vdash_{\mathbf{hD}} \overrightarrow{\triangleright^{-1} A} \Rightarrow J\backslash A$

vi) $\vdash_{\mathbf{hD}} \overrightarrow{J\backslash A} \Rightarrow \triangleright^{-1} A$

vii) $\vdash_{\mathbf{hD}} \overrightarrow{\hat{}^{i} A} \rightarrow A \odot_i I$

viii) $\vdash_{\mathbf{hD}} \overrightarrow{A \odot_i I} \rightarrow \hat{}^{i} A$

ix) $\vdash_{\mathbf{hD}} \overrightarrow{\check{}^{i} A} \Rightarrow A{\uparrow_i}I$

x) $\vdash_{\mathbf{hD}} \overrightarrow{A{\uparrow_i}I} \Rightarrow \check{}^{i} A$

Proof.

i) $$\cfrac{\cfrac{\overrightarrow{A} \Rightarrow A \qquad [] \Rightarrow J}{\overrightarrow{A}, [] \Rightarrow A \bullet J}\bullet R}{\overrightarrow{\triangleleft A} \Rightarrow A \bullet J}\triangleleft L$$

ii) $$\cfrac{\cfrac{\cfrac{\overrightarrow{A} \Rightarrow A}{\overrightarrow{A}, [] \Rightarrow \triangleleft A}\triangleleft R}{\overrightarrow{A}, \overrightarrow{J} \Rightarrow \triangleleft A}JL}{\overrightarrow{A \bullet J} \Rightarrow \triangleleft A}\bullet L$$

$$\frac{\Delta\langle\overrightarrow{A}|_1\overrightarrow{B}\rangle \Rightarrow C \qquad \cdots \qquad \Delta\langle\overrightarrow{A}|_a\overrightarrow{B}\rangle \Rightarrow C}{\Delta\langle\overrightarrow{A \circledcirc B}\rangle \Rightarrow C}\circledcirc L \qquad \frac{\Delta \Rightarrow A \qquad \Gamma \Rightarrow B}{\Delta|_i\Gamma A \circledcirc B}\circledcirc R$$

$$\frac{\Delta \Rightarrow A \qquad \Gamma\langle\overrightarrow{B}\rangle \Rightarrow C}{\Gamma\langle\overrightarrow{B\Uparrow A}|_i\Gamma\rangle \Rightarrow C}\Uparrow L \qquad \frac{\Delta|_1\overrightarrow{A} \Rightarrow B \qquad \cdots \qquad \Delta|_d\overrightarrow{A} \Rightarrow B}{\Delta \Rightarrow B\Uparrow A}\Uparrow R$$

$$\frac{\Delta \Rightarrow A \qquad \Gamma\langle\overrightarrow{B}\rangle \Rightarrow C}{\Gamma\langle\Gamma|_i\overrightarrow{A\Downarrow B}\rangle \Rightarrow C}\Downarrow L \qquad \frac{\overrightarrow{A}|_1\Delta \Rightarrow B \qquad \cdots \qquad \overrightarrow{A}|_a\Delta \Rightarrow B}{\Delta \Rightarrow A\Downarrow B}\Downarrow R$$

$$\frac{\Delta\langle\overrightarrow{B}\rangle \Rightarrow C}{\Delta\langle\overrightarrow{\check{}B}|_i\Lambda\rangle \Rightarrow C}\check{}L \qquad \frac{\Delta|_1\Lambda \Rightarrow B \qquad \cdots \qquad \Delta|_d\Lambda \Rightarrow B}{\Delta \Rightarrow \check{}B}\check{}R$$

$$\frac{\Delta\langle\overrightarrow{B}|_1\Lambda\rangle \Rightarrow C \qquad \cdots \qquad \Delta\langle\overrightarrow{B}|_b\Lambda\rangle \Rightarrow C}{\Delta\langle\overrightarrow{\hat{}B}\rangle \Rightarrow \hat{}C}\hat{}L \qquad \frac{\Delta \Rightarrow B}{\Delta|_i\Lambda \Rightarrow \hat{}B}\hat{}R$$

Figure 3.16: Hypersequent calculus rules for nondeterministic synthetic connectives

iii)
$$\dfrac{\dfrac{\dfrac{\overrightarrow{A} \Rightarrow A}{\overrightarrow{\triangleleft^{-1}A}, [] \Rightarrow A}\triangleleft^{-1}L}{\overrightarrow{\triangleleft^{-1}A}, J \Rightarrow A}JL}{\overrightarrow{\triangleleft^{-1}A} \Rightarrow A/J}/R$$

iv)
$$\dfrac{\dfrac{\overrightarrow{A} \Rightarrow A \qquad [] \Rightarrow J}{\overrightarrow{A/J}, [] \Rightarrow A}/L}{\overrightarrow{A/J} \Rightarrow \triangleleft^{-1}A}\triangleleft^{-1}R$$

v) and vi) are a direct mirror image of iii) and iv).

vii)
$$\dfrac{\dfrac{\overrightarrow{A} \Rightarrow \qquad \Lambda \Rightarrow I}{\overrightarrow{A}|_i\Lambda \Rightarrow A \odot_i I}\odot_i R}{\overrightarrow{\hat{}^iA} \Rightarrow A \odot_i I}\hat{}^iL$$

viii)
$$\dfrac{\dfrac{\dfrac{\overrightarrow{A} \Rightarrow A}{\overrightarrow{A}|_i\Lambda \Rightarrow A\hat{}^iA}\hat{}^iR}{\overrightarrow{A}|_i\Lambda \Rightarrow \hat{}^iA}IL}{\overrightarrow{A \odot_i I} \Rightarrow \hat{}^iA}\odot_i L$$

ix)
$$\dfrac{\dfrac{\dfrac{\overrightarrow{A} \Rightarrow A}{\overrightarrow{\check{}^iA}|_i\Lambda \Rightarrow A}\check{}^i}{\overrightarrow{\check{}^iA}|_iI \Rightarrow A}IL}{\overrightarrow{\check{}^iA} \Rightarrow A\uparrow_i I}\uparrow_i R$$

x)
$$\dfrac{\dfrac{\Lambda \Rightarrow I \qquad \overrightarrow{A} \Rightarrow A}{\overrightarrow{A\uparrow_i I}|_i\Lambda \Rightarrow A}\uparrow_i L}{\overrightarrow{A\uparrow_i I} \Rightarrow \check{}^iA}\check{}^iR$$

□

3.8.2 Nondeterministic Synthetic Rules

We show in Figure 3.16 the hypersequent calculus rules for the nondeterministic synthetic connectives. Let us see at some elementary syntactic results w.r.t. the nondeterministic connectives.

(170) **Lemma** (*Elementary Syntactic Results for Nondeterministic Connectives*)

The following hypersequents are provable in **hDADND**:

i) $\vdash_{\mathbf{hDAND}} \overrightarrow{A \circledcirc B} \Rightarrow \bigoplus_{i=1,\cdots,a} A \odot_i B$

ii) $\vdash_{\mathbf{hDAND}} \overrightarrow{\bigoplus_{i=1,\cdots,a} A \odot_i B} \Rightarrow A \circledcirc B$

iii) $\vdash_{\mathbf{hDAND}} \overrightarrow{B\Uparrow A} \Rightarrow B\uparrow_1 A \,\&\, \cdots \,\&\, B\uparrow_{b-a+1} A$

iv $\vdash_{\mathbf{hDAND}} \overrightarrow{B\uparrow_1 A \,\&\, \cdots \,\&\, B\uparrow_{b-a+1} A} \Rightarrow B\Uparrow A$

v) $\vdash_{\mathbf{hDAND}} \overrightarrow{A\Downarrow B} \Rightarrow A\downarrow_1 B \,\&\, \cdots \,\&\, A\downarrow_{b-a+1} B$

vi) $\vdash_{\mathbf{hDAND}} \overrightarrow{A\downarrow_1 B \,\&\, \cdots \,\&\, A\downarrow_{b-a+1} B} \Rightarrow A\Downarrow B$

vii) $\vdash_{\mathbf{hDAND}} \overrightarrow{\hat{}A} \Rightarrow \bigoplus_{i=1,\cdots,a} A \odot_i I$

viii) $\vdash_{\mathbf{hDAND}} \overrightarrow{\bigoplus_{i=1,\cdots,a} A \odot_i I} \Rightarrow \hat{}A$

ix) $\vdash_{\mathbf{hDAND}} \overrightarrow{\check{}A} \Rightarrow A\uparrow_1 I \,\&\, \cdots \,\&\, A\uparrow_{a+1} I$

x) $\vdash_{\mathbf{hDAND}} \overrightarrow{A\uparrow_1 I \,\&\, \cdots \,\&\, A\uparrow_{b+1} I} \Rightarrow \check{}A$

Proof.

i) For every $k = 1, \cdots, a$ we have that:

$$\overrightarrow{A}|_k\overrightarrow{B} \Rightarrow \bigoplus_{i=1,\cdots,a} A \odot_i B$$

$$\dfrac{\overrightarrow{A}|_1\overrightarrow{B} \Rightarrow \bigoplus_{i=1,\cdots,a} A \odot_i B \qquad \cdots \qquad \overrightarrow{A}|_a\overrightarrow{B} \Rightarrow \bigoplus_{i=1,\cdots,a} A \odot_i B}{\overrightarrow{A \circledcirc B} \Rightarrow \bigoplus_{i=1,\cdots,a} A \odot_i B}\circledcirc L$$

ii) For every $k = 1, \cdots, a$ we have by the $\odot_k$ right rule that:

$$\overrightarrow{A}|_k\overrightarrow{B} \Rightarrow A \circledcirc B$$

By applying $a-1$ times the $\oplus$ left rule we obtain:

$$\dfrac{\vdots}{\bigoplus_{i=1,\cdots,a} A \odot_i B \Rightarrow A \circledcirc B}\oplus L$$

iii) For every $k = 1, \cdots, b-a+1$, by application of the $\Uparrow$ left rule we have that:

$$\overrightarrow{B\Uparrow A}|_k\overrightarrow{A} \Rightarrow B$$

For every $k = 1, \cdots, b-a+1$, by application of the $\uparrow_k$ right rule we have that:

$$\overrightarrow{B\Uparrow A} \Rightarrow B\uparrow_k A$$

By $b-a$ applications of the & right rule:

$$\frac{}{\overrightarrow{B\Uparrow A} \Rightarrow B\uparrow_1 A \& \cdots \& B\uparrow_{b-a+1} A}\,\&R$$

iv) Let i be any index ranging in $\{1, \cdots, b-a+1\}$. We have by application of the $\uparrow_i$ left rule:

$$\overrightarrow{B\uparrow_i A}|_i \overrightarrow{A} \Rightarrow B$$

Appplying the & left rule we obtain:

$$\overrightarrow{(\cdots(B\uparrow_1 A \& B\uparrow_2 A)\& \cdots)\& B\uparrow_i A} \Rightarrow B$$

Appplying one more time the & left rule we obtain:

$$\overrightarrow{B\uparrow_1 A \& \cdots \& B\uparrow_{b-a+1}}|_i \overrightarrow{A} \Rightarrow \overrightarrow{B}$$

Summarizing, we can apply the $\Uparrow$ right rule:

$$\frac{\overrightarrow{B\uparrow_1 A \& \cdots \& B\uparrow_{b-a+1} A}|_1 \overrightarrow{A} \Rightarrow \overrightarrow{B} \qquad \cdots \qquad \overrightarrow{B\uparrow_1 A \& \cdots \& B\uparrow_{b-a+1} A}|_{b-a+1} \overrightarrow{A} \Rightarrow \overrightarrow{B}}{\overrightarrow{B\uparrow_1 A \& \cdots \& B\uparrow_{b-a+1} A} \Rightarrow \overrightarrow{B\Uparrow A}}\,\Uparrow R$$

The remaining cases are completely similar to i), ii), iii) and iv).

□

3.8.3 An Embedding Translation from DND into DA

We consider an interesting embedding from **DND** into **DA**, more concretely $\tau : (\mathcal{O}_{\mathbf{DND}}, \mathcal{F}_{\mathbf{DND}}, \Rightarrow) \longrightarrow (\mathcal{O}_{\mathbf{DA}}, \mathcal{F}_{\mathbf{DA}} \Rightarrow)$:

$\tau(\hat{\ }A) = \tau(A)\odot I$
$\tau(\check{\ }A) = \tau(A) \uparrow I$
$\tau(\hat{\ }^i A) = \tau(A)\odot_i I$
$\tau(\check{\ }^i A) = \tau(A) \uparrow_i I$
$\tau(\triangleleft A) = \tau(A) \bullet J$
$\tau(\triangleleft^{-1} A) = \tau(A)/J$
$\tau(\triangleright A) = J \bullet \tau(A)$
$\tau(\triangleright^{-1} A) = J\backslash\tau(A)$
$\tau(A \circledcirc B) = \bigoplus_{i=1,\cdots,a} A \odot_i B$
$\tau(B\Uparrow A) = \tau(B)\uparrow_1\tau(A)\& \cdots \&\tau(B)\uparrow_{b-a+1}\tau(A)$
$\tau(B\Downarrow A) = \tau(A)\downarrow_1\tau(B)\& \cdots \&\tau(A)\downarrow_{b-a+1}\tau(B)$
$\tau(A \star B) = \tau(A) \star \tau(B)$ for $\star \in \{\backslash, /, \bullet\} \cup (\downarrow_{i+1})_{i\in\omega} \cup (\uparrow_{i+1})_{i\in\omega} \cup (\odot_{i+1})_{i\in\omega}$

τ extends to a hypersequent $\Delta \Rightarrow A$, where A is a type and Δ a hyperconfiguaration, by mapping Δ under τ recursively as follows:

$$\begin{array}{l}\tau(\Lambda) = \Lambda \\ \tau(A, \Gamma) = \tau(A), \tau(\Gamma) \text{ where } S(A) = 0 \\ \tau([], \Gamma) = [], \tau(\Gamma) \\ \tau(A\{\Gamma_1 : \cdots : \Gamma_a\}, \Gamma_{a+1}) = \tau(A)\{\tau(\Gamma_1) : \cdots : \tau(\Gamma_a)\}, \tau(\Gamma_{a+1})\end{array}$$

In the following lines we assume that the Cut elimination theorem for **hDADND** holds (see 176).

(171) **Lemma**
Let A be a **DND** type. It follows that:

i)

$$\vdash_{\mathbf{DADND}} \overrightarrow{A} \Rightarrow \tau(A)$$
and
$$\vdash_{\mathbf{DADND}} \overrightarrow{\tau(A)} \Rightarrow A$$

ii)

$$\vdash_{\mathbf{DADND}} \underbrace{[], \cdots, []}_{b} \Rightarrow \tau(B) \text{ iff } \vdash_{\mathbf{DADND}} \underbrace{[], \cdots, []}_{b} \Rightarrow B$$

Where if $b = 0$ then $\underbrace{[], \cdots, []}_{b}$ is the empty string Λ.

Proof. In this proof the hypersequent derivations take place in **DADND**.

i) We proceed by induction on the structure of **DND** types. In the case of atomic cases there is nothing to prove.

- Suppose we have the type $\check{}^i A$. By lemma (169) we have that:

$$\overrightarrow{\check{}^i A} \Rightarrow A\uparrow_i I$$

By induction hypothesis (i.h.), $\overrightarrow{A} \Rightarrow \tau(A)$. By the tonicity properties of $\uparrow_i$ $\overrightarrow{A\uparrow_i I} \Rightarrow \tau(A)\uparrow_i I$. It follows that:

$$\frac{\overrightarrow{\check{}^i A} \Rightarrow A\uparrow_i I \qquad \overrightarrow{A\uparrow_i I} \Rightarrow \tau(A)\uparrow_i I}{\overrightarrow{\check{}^i A} \Rightarrow \tau(A)\uparrow_i I = \tau(\check{}^i A)}\, Cut$$

Conversely: by i.h. $\overrightarrow{\tau(A)} \Rightarrow A$. We know by lemma (169) that $\overrightarrow{A \odot_i B} \Rightarrow \check{}^i A$. Therefore we have the following derivation:

$$\frac{\dfrac{\overrightarrow{\tau(A)} \Rightarrow A \qquad \Lambda \Rightarrow I}{\overrightarrow{\tau(A)}|_i\Lambda \Rightarrow A \odot_i I}\, \odot_i R \qquad \overrightarrow{A} \odot_i I \Rightarrow \check{}^i A}{\overrightarrow{\tau(A)}|_i\Lambda \Rightarrow \check{}^i A}\, Cut$$

By application of the rules IL and $\odot_i L$ to the end-hypersequent and the fact that $\overrightarrow{\tau(\check{}^i A)} = \overrightarrow{\tau(A) \odot_i I}$, we obtain $\overrightarrow{\tau(\check{}^i A)} \Rightarrow \check{}^i A$.

- $\circledcirc$ case. We want to see:

$$A \circledcirc B \Rightarrow \overrightarrow{\tau(A \circledcirc B)}$$

By lemma (170) we have $\overrightarrow{A \circledcirc B} \Rightarrow \bigoplus_{i=1,\cdots,a} A \odot_i B$. By i.h. $\overrightarrow{A \odot_i B} \Rightarrow \tau(A \odot_i B)$. By monotonicity of $\oplus$ and Cut, the following holds:

$$\overrightarrow{A \circledcirc B} \Rightarrow \bigoplus_{i=1,\cdots,a} \tau(A \odot_i B) = \tau(A \circledcirc B)$$

Conversely, by i.h. $\overrightarrow{\tau(A \odot_i B)} \Rightarrow A \odot_i B$. By monotonicity of $\oplus$, we have that $\overrightarrow{A \circledcirc B} = \overrightarrow{\bigoplus_{i=1,\cdots,a} \tau(A \odot_i B)} \Rightarrow \bigoplus_{i=1,\cdots,a} A \odot_i B$. By lemma (170) $\overrightarrow{\bigoplus_{i=1,\cdots,a} A \odot_i B} \Rightarrow A \circledcirc B$.

The remaining connectives have analogous proofs.

ii) We prove a case, e.g. the $\circledcirc$ one. If $\Rightarrow A \circledcirc B$ is provable we have that:

$$\dfrac{\Rightarrow A \circledcirc B \qquad \overrightarrow{A \circledcirc B} \Rightarrow \tau(A \circledcirc B),\ \text{by lemma i)}}{\Rightarrow \tau(A \circledcirc B)}\ Cut$$

And if $\Rightarrow \tau(A \circledcirc B)$ is provable we have that:

$$\dfrac{\Rightarrow \tau(A \circledcirc B) \qquad \overrightarrow{\tau(A \circledcirc B)} \Rightarrow A \circledcirc B,\ \text{by lemma i)}}{\Rightarrow A \circledcirc B}\ Cut$$

This completes the proof. $\square$

(172) **Theorem** (*Faithful Embedding Translation from* **hDND** *into* **hDA**)

Let $\Delta \Rightarrow A$ be a **DND** hypersequent. Then we have that:

$$\vdash_{\mathbf{DND}} \Delta \Rightarrow A \text{ iff } \vdash_{\mathbf{DADND}} \tau(\Delta) \Rightarrow \tau(A)$$

Proof. Let $\Delta\langle\overrightarrow{A_1}; \cdots; \overrightarrow{A_n}\rangle \Rightarrow B$ be a **DND** hypersequent with exactly n type occurrences. In the following we write $\vdash$ for $\vdash_{\mathbf{DADND}}$. When necessary, we will write $\vdash_{\mathbf{DND}}$. We prove by induction on n that:

$$\Delta\langle\tau(\overrightarrow{A_1}); \cdots; \tau(\overrightarrow{A_n})\rangle \Rightarrow \tau(B) \text{ iff } \Delta\langle\overrightarrow{A_1}; \cdots; \overrightarrow{A_n}\rangle \Rightarrow B$$

If $n = 0$ we have by the previous lemma that:

(173) $\underbrace{[], \cdots, []}_{b} \Rightarrow \tau(B)$ iff $\underbrace{[], \cdots, []}_{b} \Rightarrow B$

Suppose $n \geq 1$. By the previous lemma we know that for every $A \in \mathcal{F}_{\mathbf{DND}}$, $\vdash \overrightarrow{A} \Rightarrow \tau(A)$ and $\overrightarrow{\tau(A)} \Rightarrow A$. We have the following two derivations:

(174)
$$\cfrac{\cfrac{\overrightarrow{A_n} \Rightarrow \tau(A_n) \qquad \Delta\langle\tau(\overrightarrow{A_1});\cdots;\tau(\overrightarrow{A_n})\rangle \Rightarrow \tau(B)}{\Delta\langle\tau(\overrightarrow{A_1});\cdots;\overrightarrow{A_n}\rangle \Rightarrow \tau(B)}\,Cut \qquad \overrightarrow{\tau(B)} \Rightarrow B}{\Delta\langle\tau(\overrightarrow{A_1});\cdots;\overrightarrow{A_n}\rangle \Rightarrow B}\,Cut$$

(175)
$$\cfrac{\cfrac{\overrightarrow{\tau(A_n)} \Rightarrow A_n \qquad \Delta\langle\tau(\overrightarrow{A_1});\cdots;\overrightarrow{A_n}\rangle \Rightarrow B}{\Delta\langle\tau(\overrightarrow{A_1});\cdots;\tau(\overrightarrow{A_n})\rangle \Rightarrow B}\,Cut \qquad \overrightarrow{B} \Rightarrow \tau(B)}{\Delta\langle\tau(\overrightarrow{A_1});\cdots;\tau(\overrightarrow{A_n})\rangle \Rightarrow \tau(B)}\,Cut$$

Hence we have:

$$\Delta\langle\tau(\overrightarrow{A_1});\cdots;\tau(\overrightarrow{A_n})\rangle \Rightarrow \tau(B) \text{ iff } \Delta\langle\tau(\overrightarrow{A_1});\cdots;\overrightarrow{A_n}\rangle \Rightarrow B$$

Doing the same as in (174) and (175) with the remaining type occurrences $A_1, \cdots, A_{n-1}$, we get eventually:

$$\Delta\langle\tau(\overrightarrow{A_1});\cdots;\tau(\overrightarrow{A_n})\rangle \Rightarrow \tau(B) \text{ iff } \Delta\langle\overrightarrow{A_1};\cdots;\overrightarrow{A_n}\rangle \Rightarrow B$$

Now, since Cut elimination holds of **DADND** (see section 3.9), for **DND** we have that for an arbitrary **DND** hyperconfiguration Δ and for an arbitrary **DND** type A:

$$\vdash_{\mathbf{DND}} \Delta \Rightarrow A \text{ iff } \vdash_{\mathbf{DADND}} \tau(\Delta) \Rightarrow \tau(A)$$

This completes the proof. □

It is time to formulate the Cut elimination theorem for **D**. In fact we prove it for all the extensions, i.e. we prove Cut elimination for **DADND**.

(176) **Theorem** (*Cut Elimination Theorem*)

hDADND enjoys Cut elimination.

Proof. See Section 3.9. □

(177) **Corollary** (*Subformula property*)

In **hDADND** every provable hypersequent contains a proof containing only its subformulas.

Proof. Every rule except Cut has the property that all the types in the premises are either in the conclusion or are the immediate subtypes of the active formula. And we know that Cut in **hDADND** is eliminable. We are done. □

(178) **Corollary** (*Decidability of* **hDADND**)

In **hDADND**, it is decidable whether a hypersequent is provable.

Proof. By backward-chaining in the finite Cut-free hypersequent search space. □

We give the semantic type T map for discontinuous Lambek calculus in Figure 3.17. T sends **hDADND** hypersequent derivations into intuitionistic proofs in the same way as for the Lambek calculus. Therefore, the so-called

Curry-Howard categorial type logical semantics comes for free. Since the Cut-free search space of **hDADND** is finite, it follows that the number of derivations is finite, whence the so-called finite reading property holds. We have proved the following theorem:

(179) **Theorem** (*Finite Reading Property for* **hDADND**)

In **hDADND**, every hypersequent has a finite number of *semantic readings.*

By the subformula property for **hDADND**, it follows that the following fragments enjoy Cut elimination, the subformula property, decidability and the finite reading property:

- **hDA**
- **hDA**
- **hDND**

$$
\begin{array}{lcl}
T(I) & = & \top \\
T(J) & = & \top \\
T(A\backslash C) & = & T(A) \to T(C) \\
T(C/B) & = & T(B) \to T(C) \\
T(A \bullet B) & = & T(A)\&T(B) \\
T(A{\downarrow_k}C) & = & T(A) \to T(C) \\
T(C{\uparrow_k}B) & = & T(B) \to T(C) \\
T(A \odot_k B) & = & T(A)\&T(B) \\
T(A\&B) & = & T(A)\&T(B) \\
T(A \oplus B) & = & T(A) \vee T(B) \\
T(A{\Downarrow}C) & = & T(A) \to T(C) \\
T(C{\Uparrow}B) & = & T(B) \to T(C) \\
T(A \circledcirc B) & = & T(A)\&T(B) \\
T(A{\downarrow_k}C) & = & T(A) \to T(C) \\
T(C{\uparrow_k}B) & = & T(B) \to T(C) \\
T(*A) & = & T(A) \text{ for } * \in \{\hat{}^k, \check{}^k, \hat{}, \check{}, \triangleleft, \triangleleft^{-1}, \triangleright, \triangleright^{-1}\}
\end{array}
$$

Figure 3.17: Semantic type map for **DADND**

3.9 A Direct Proof of Cut Elimination for hD and its Extensions

Cut Elimination Steps for D

Lambek (1958) proved Cut elimination for the Lambek calculus **L**. Cut elimination states that every theorem can be proved without the use of Cut. Lambek's proof is simpler than that of Gentzen for standard logic due to the absence of structural rules. It consists of defining a notion of degree of Cut instances and showing how Cuts in a proof can be succesively replaced by Cuts of lower degree until they are removed altogether. Thus Lambek's proof provides an algorithm for transforming proofs into Cut-free counterparts. The Cut-elimination theorem has as corollaries the subformula property and decidability. Here we prove Cut-elimination for the displacement calculus **D** in its hypersequent presentation, i.e. **hD**. Like **L**, **hD** contains no structural rules (structural properties are built into the sequent calculus notation) and the Cut-elimination is proved following the same strategy as for **L**.

We define the *weight* $|A|$ of a type A as the number of connectives occurrences (including units) that it contains. The weight $|\Gamma|$ of a hyperconfiguration is the sum of the weights of the types that occur in it, that is, it is defined by unambiguous recursion[10] as follows:

(180)
$$\begin{array}{lcl} |\Lambda| & = & 0 \\ |A, \Delta| & = & |A| + |\Delta| \\ |[], \Delta| & = & |\Delta| \text{ i.e. } |[]| = 0 \\ |A\{\Gamma_1 : \cdots : \Gamma_{i+1}\}, \Delta| & = & |A| + \sum_{j=1}^{i+1} |\Gamma_j| + |\Delta| \end{array}$$

It is not difficult to prove that if Δ, Γ are arbitrary hyperconfiguration then concatenation and intercalation of hyperconfigurations satisfy:

$$|\Gamma, \Theta| \;=\; |\Gamma| + |\Theta| \text{ and } |\Gamma\,|_i\,\Theta| \;=\; |\Gamma| + |\Theta|$$

The weight of a hypercontext is defined similarly with a hole having weight zero.

Consider the Cut rule:

(181)
$$\frac{\Gamma \Rightarrow A \qquad \Delta\langle \overrightarrow{A} \rangle \Rightarrow B}{\Delta\langle \Gamma \rangle \Rightarrow B}\, Cut \quad (\star)$$

We define the *degree* $d(\star)$ of an instance $\star$ of the Cut rule as follows:

(182) $d(\star) = |\Gamma| + |\Delta| + |A| + |B|$

We call the type A in (181) the *Cut formula*. Consider a proof which is not Cut-free. Then there is some Cut-instance above which there are no Cuts. We will show that this Cut can either be removed or replaced by one or two Cuts of lower degree. The following three cases are exhaustive:

(183)
- A premise of the Cut is the identity axiom: then the conclusion is identical to the other premise and the Cut as a whole can be removed.
- Both the premises are conclusions of logical rules and it is not the case that the Cut formula is the active formula of both premises: then we apply *permutation conversion* cases.
- Both the premises are conclusions of logical rules and the Cut formula is the active formula of both premises: then we apply *principal Cut* cases.

There are several cases to consider. We give representative examples.

- Permutation conversion cases

[10] In order that $|\cdot|$ be well defined, we require the recursive definition to be unambiguous.

The active formula in the left premise of the Cut rule is not the Cut formula

- The rule applying at the left premise of the Cut rule is $\odot_i L$:

$$\dfrac{\dfrac{\Delta\langle\overrightarrow{B}|_i\overrightarrow{C}\rangle \Rightarrow A}{\Delta\langle\overrightarrow{B\odot_i C}\rangle \Rightarrow A}\odot_i L \qquad \Gamma\langle\overrightarrow{A}\rangle \Rightarrow D}{\Gamma\langle\Delta\langle\overrightarrow{B\odot_i C}\rangle\rangle \Rightarrow D}\,Cut$$

$$\rightsquigarrow$$

$$\dfrac{\dfrac{\Delta\langle\overrightarrow{B}|_i\overrightarrow{C}\rangle \Rightarrow A \qquad \Gamma\langle\overrightarrow{A}\rangle \Rightarrow D}{\Gamma\langle\Delta\langle\overrightarrow{B}|_i\overrightarrow{C}\rangle\rangle \Rightarrow A}\,Cut}{\Gamma\langle\Delta\langle\overrightarrow{B\odot_i C}\rangle\rangle \Rightarrow D}\odot_i L$$

- The rule applying at the left premise of the Cut rule is $\uparrow_i L$:

$$\dfrac{\dfrac{\Gamma\langle\overrightarrow{C}\rangle \Rightarrow A \qquad \Delta \Rightarrow B}{\Gamma\langle\overrightarrow{C\uparrow_i B}|_i\Delta\rangle \Rightarrow A}\uparrow_i L \qquad \Theta\langle\overrightarrow{A}\rangle \Rightarrow D}{\Theta\langle\Gamma\langle\overrightarrow{C\uparrow_i B}|_i\Delta\rangle\rangle \Rightarrow D}\,Cut$$

$$\rightsquigarrow$$

$$\dfrac{\dfrac{\Gamma\langle\overrightarrow{C}\rangle \Rightarrow A \qquad \Theta\langle\overrightarrow{A}\rangle \Rightarrow D}{\Theta\langle\Gamma\langle\overrightarrow{C}\rangle\rangle \Rightarrow D}\,Cut \qquad \Delta \Rightarrow B}{\Theta\langle\Gamma\langle\overrightarrow{C\uparrow_i B}|_i\Delta\rangle\rangle \Rightarrow D}\uparrow_i L$$

- The rule applying at the left premise of the Cut rule is JL:

$$\dfrac{\dfrac{\Gamma\langle[]\rangle \Rightarrow A}{\Gamma\langle\overrightarrow{J}\rangle \Rightarrow A}\,JL \qquad \Delta\langle\overrightarrow{A}\rangle \Rightarrow B}{\Delta\langle\Gamma\langle\overrightarrow{J}\rangle\rangle \Rightarrow B}\,Cut$$

$$\rightsquigarrow$$

$$\dfrac{\dfrac{\Gamma\langle[]\rangle \Rightarrow A \qquad \Delta\langle\overrightarrow{A}\rangle \Rightarrow B}{\Delta\langle\Gamma\langle[]\rangle\rangle \Rightarrow A}\,Cut}{\Delta\langle\Gamma\langle\overrightarrow{J}\rangle\rangle \Rightarrow B}\,JL$$

- The rule applying at the left premise of the Cut rule is $\lhd^{-1}L$:

$$\dfrac{\dfrac{\Delta\langle\overrightarrow{B}\rangle \Rightarrow A}{\Delta\langle\overrightarrow{\lhd^{-1}}, []\rangle \Rightarrow A}\lhd^{-1}L \qquad \Gamma\langle\overrightarrow{A}\rangle \Rightarrow C}{\Gamma\langle\Delta\langle\overrightarrow{\lhd^{-1}B}, []\rangle\rangle \Rightarrow C}\,Cut$$

$$\rightsquigarrow$$

$$\dfrac{\dfrac{\Delta\langle\overrightarrow{B}\rangle \Rightarrow A \qquad \Gamma\langle\overrightarrow{A}\rangle \Rightarrow C}{\Gamma\langle\Delta\langle\overrightarrow{B}\rangle\rangle C}\,Cut}{\Gamma\langle\Delta\langle\overrightarrow{\lhd^{-1}B}, []\rangle\rangle \Rightarrow C}\lhd^{-1}L$$

- The rule applying at the left premise of the Cut rule is $\triangleleft L$:

$$\dfrac{\dfrac{\Delta\langle\overrightarrow{\triangleleft B}, []\rangle \Rightarrow A}{\Delta\langle\overrightarrow{\triangleleft B}\rangle \Rightarrow A}\triangleleft L \qquad \Gamma\langle\overrightarrow{A}\rangle \Rightarrow C}{\Gamma\langle\Delta\langle\overrightarrow{\triangleleft B}\rangle\rangle \Rightarrow C}\,Cut$$

$$\rightsquigarrow$$

$$\dfrac{\dfrac{\Delta\langle\overrightarrow{\triangleleft B}, []\rangle \Rightarrow A \qquad \Gamma\langle\overrightarrow{A}\rangle \Rightarrow C}{\Gamma\langle\Delta\langle\overrightarrow{B}, []\rangle\rangle \Rightarrow C}\,Cut}{\Gamma\langle\Delta\langle\overrightarrow{\triangleleft B}\rangle\rangle \Rightarrow C}\triangleleft L$$

- The rule applying at the left premise of the Cut rule is $\check{}^{\,i}L$:

$$\dfrac{\dfrac{\Delta\langle\overrightarrow{B}\rangle \Rightarrow A}{\Delta\langle\overrightarrow{\check{}^{\,i}B}|_i\Lambda\rangle \Rightarrow A}\check{}^{\,i}L \qquad \Gamma\langle\overrightarrow{A}\rangle \Rightarrow C}{\Gamma\langle\Delta\langle\overrightarrow{\check{}^{\,i}B}|_i\Lambda\rangle\rangle \Rightarrow C}\,Cut$$

$$\rightsquigarrow$$

$$\dfrac{\dfrac{\Delta\langle\overrightarrow{B}\rangle \Rightarrow A \qquad \Gamma\langle\overrightarrow{A}\rangle \Rightarrow C}{\Gamma\langle\Delta\langle\overrightarrow{B}\rangle\rangle \Rightarrow C}\,Cut}{\Gamma\langle\Delta\langle\overrightarrow{\check{}^{\,i}B}|_i\Lambda\rangle\rangle \Rightarrow C}\check{}^{\,i}L$$

- The rule applying at the left premise of the Cut rule is $\hat{}^{\,i}L$:

$$\dfrac{\dfrac{\Delta\langle\overrightarrow{B}|_i\Lambda\rangle \Rightarrow A}{\Delta\langle\overrightarrow{\hat{}^{\,k}B}\rangle \Rightarrow A}\hat{}^{\,i}L \qquad \Gamma\langle\overrightarrow{A}\rangle \Rightarrow C}{\Gamma\langle\Delta\langle\overrightarrow{\hat{}^{\,i}B}\rangle\rangle \Rightarrow C}\,Cut$$

$$\rightsquigarrow$$

$$\dfrac{\dfrac{\Delta\langle\overrightarrow{B}|_i\Lambda\rangle \Rightarrow A \qquad \Gamma\langle\overrightarrow{A}\rangle \Rightarrow C}{\Gamma\langle\Delta\langle\overrightarrow{B}|_i\Lambda\rangle\rangle \Rightarrow C}\,Cut}{\Gamma\langle\Delta\langle\overrightarrow{\hat{}^{\,i}B}\rangle\rangle \Rightarrow C}\hat{}^{\,i}L$$

The active formula in the right premise of the Cut rule is not the Cut formula

- The rule applying at the right premise of the Cut rule is $\uparrow_i L$:

$$\dfrac{\Delta \Rightarrow A \qquad \dfrac{\Gamma\langle\overrightarrow{A};\overrightarrow{C}\rangle \Rightarrow D \qquad \Theta \Rightarrow B}{\Gamma\langle\overrightarrow{A};\overrightarrow{C\uparrow_i B}|_i\Theta\rangle \Rightarrow D}\uparrow_i L}{\Gamma\langle\Delta;\overrightarrow{C\uparrow_i B}|_i\Theta\rangle \Rightarrow D}\,Cut$$

$$\rightsquigarrow$$

$$\dfrac{\dfrac{\Delta \Rightarrow A \qquad \Gamma\langle\overrightarrow{A};\overrightarrow{C}\rangle \Rightarrow D}{\Gamma\langle\Delta;\overrightarrow{C}\rangle \Rightarrow D}\,Cut \qquad \Theta \Rightarrow B}{\Gamma\langle\Delta;\overrightarrow{C\uparrow_i B}|_i\Theta\rangle \Rightarrow D}\uparrow_i L$$

- The rule applying at the right premise of the Cut rule is $\uparrow_i R$:

$$\dfrac{\Delta \Rightarrow A \qquad \dfrac{\Gamma\langle\overrightarrow{A}\rangle|_i\overrightarrow{B} \Rightarrow C}{\Gamma\langle\overrightarrow{A}\rangle \Rightarrow C\uparrow_i B}\uparrow_i R}{\Gamma\langle\Delta\rangle \Rightarrow C\uparrow_i B} Cut$$

$$\rightsquigarrow$$

$$\dfrac{\dfrac{\Delta \Rightarrow A \qquad \Gamma\langle\overrightarrow{A}\rangle|_i\overrightarrow{B} \Rightarrow C}{\Gamma\langle\Delta\rangle|_i\overrightarrow{B} \Rightarrow C} Cut}{\Gamma\langle\Delta\rangle \Rightarrow C\uparrow_i B}\uparrow_i R$$

- The rule applying at the right premise of the Cut rule is $\odot_i L$:

$$\dfrac{\Delta \Rightarrow A \qquad \dfrac{\Gamma\langle\overrightarrow{A};\overrightarrow{B}|_i\overrightarrow{C}\rangle \Rightarrow D}{\Gamma\langle\overrightarrow{A};\overrightarrow{B\odot_i C}\rangle \Rightarrow D}\odot_i L}{\Gamma\langle\Delta;\overrightarrow{B\odot_i C}\rangle \Rightarrow D} Cut$$

$$\rightsquigarrow$$

$$\dfrac{\dfrac{\Delta \Rightarrow A \qquad \Gamma\langle\overrightarrow{A};\overrightarrow{B}|_i\overrightarrow{C}\rangle \Rightarrow D}{\Gamma\langle\Delta;\overrightarrow{B}|_i\overrightarrow{C}\rangle \Rightarrow D} Cut}{\Gamma\langle\Delta;\overrightarrow{B\odot_i C}\rangle \Rightarrow D}\odot_i L$$

- The rule applying at the right premise of the Cut rule is $\odot_i R$:

$$\dfrac{\Delta \Rightarrow A \qquad \dfrac{\Gamma\langle\overrightarrow{A}\rangle \Rightarrow B \qquad \Theta \Rightarrow C}{\Gamma\langle\overrightarrow{A}\rangle|_i\Theta \Rightarrow B\odot_i C}\odot_i R}{\Gamma\langle\Delta\rangle|_i\Theta \Rightarrow B\odot_i C} Cut$$

$$\rightsquigarrow$$

$$\dfrac{\dfrac{\Delta \Rightarrow A \qquad \Gamma\langle\overrightarrow{A}\rangle \Rightarrow B}{\Gamma\langle\Delta\rangle \Rightarrow B} Cut \qquad \Theta \Rightarrow C}{\Gamma\langle\Delta\rangle|_i\Theta \Rightarrow B\odot_i C}\odot_i R$$

- The rule applying at the right premise of the Cut rule is $\triangleleft^{-1}R$:

$$\dfrac{\Delta \Rightarrow A \qquad \dfrac{\Gamma\langle\overrightarrow{A}\rangle, [] \Rightarrow C}{\Gamma\langle\overrightarrow{A}\rangle \Rightarrow \triangleleft^{-1}C}\triangleleft^{-1}R}{\Gamma\langle\Delta\rangle \Rightarrow \triangleleft^{-1}C} Cut$$

$$\rightsquigarrow$$

$$\dfrac{\dfrac{\Delta \Rightarrow A \qquad \Gamma\langle\overrightarrow{A}\rangle, [] \Rightarrow C}{\Gamma\langle\Delta\rangle, [] \Rightarrow C} Cut}{\Gamma\langle\Delta\rangle \Rightarrow \triangleleft^{-1}C}\triangleleft^{-1}R$$

- The rule applying at the right premise of the Cut rule is $\triangleleft R$:

$$\dfrac{\Delta \Rightarrow A \qquad \dfrac{\Gamma\langle\overrightarrow{A}\rangle \Rightarrow C}{\Gamma\langle\overrightarrow{A}\rangle, [] \Rightarrow \triangleleft C}\,\triangleleft R}{\Gamma\langle\Delta\rangle \Rightarrow \triangleleft C}\,Cut$$

$$\rightsquigarrow$$

$$\dfrac{\dfrac{\Delta \Rightarrow A \qquad \Gamma\langle\overrightarrow{A}\rangle \Rightarrow C}{\Gamma\langle\Delta\rangle \Rightarrow C}\,Cut}{\Gamma\langle\Delta\rangle, [] \Rightarrow \triangleleft C}\,\triangleleft R$$

- The rule applying at the right premise of the Cut rule is $\check{}^{i} R$:

$$\dfrac{\Delta \Rightarrow A \qquad \dfrac{\Gamma\langle\overrightarrow{A}\rangle|_i\Lambda \Rightarrow C}{\Gamma\langle\overrightarrow{A}\rangle \Rightarrow \check{}^{i} C}\,\check{}^{i} R}{\Gamma\langle\Delta\rangle \Rightarrow \check{}^{i} C}\,Cut$$

$$\rightsquigarrow$$

$$\dfrac{\dfrac{\Delta \Rightarrow A \qquad \Gamma\langle\overrightarrow{A}|_i\Lambda\rangle \Rightarrow C}{\Gamma\langle\Delta|_i\Lambda\rangle \Rightarrow C}\,Cut}{\Gamma\langle\Delta\rangle \Rightarrow \check{}^{i} C}\,\check{}^{i} R$$

- The rule applying at the right premise of the Cut rule is $\hat{}^{i} R$:

$$\dfrac{\Delta \Rightarrow A \qquad \dfrac{\Gamma\langle\overrightarrow{A}\rangle \Rightarrow C}{\Gamma\langle\overrightarrow{A}\rangle|_i\Lambda \Rightarrow \hat{}^{i} C}\,\hat{}^{i} R}{\Gamma\langle\Delta\rangle|_i\Lambda \Rightarrow \hat{}^{i} C}\,Cut$$

$$\rightsquigarrow$$

$$\dfrac{\dfrac{\Delta \Rightarrow A \qquad \Gamma\langle\overrightarrow{A}\rangle \Rightarrow C}{\Gamma\langle\Delta\rangle \Rightarrow C}\,Cut}{\Gamma\langle\Delta\rangle|_i\Lambda \Rightarrow \hat{}^{i} C}\,\check{}^{i} R$$

- Principal Cut cases

- The rules applying at the left and right premises of the Cut rule are respectively $\odot_i R$ and $\odot_i L$:

$$\dfrac{\dfrac{\Delta \Rightarrow A \qquad \Gamma \Rightarrow B}{\Delta|_i\Gamma \Rightarrow A\odot_i B}\,\odot_i R \qquad \dfrac{\Theta\langle\overrightarrow{A}|_i\overrightarrow{B}\rangle \Rightarrow C}{\Theta\langle\overrightarrow{A\odot_i B}\rangle \Rightarrow C}\,\odot_i L}{\Theta\langle\Delta|_i\Gamma\rangle \Rightarrow C}\,Cut$$

$$\rightsquigarrow$$

$$\dfrac{\Gamma \Rightarrow B \qquad \dfrac{\Delta \Rightarrow A \qquad \Theta\langle\overrightarrow{A}|_i\overrightarrow{B}\rangle \Rightarrow C}{\Theta\langle\Delta|_i\overrightarrow{B}\rangle \Rightarrow C}\,Cut}{\Theta\langle\Delta|_i\Gamma\rangle \Rightarrow C}\,Cut$$

- The rules applying at the left and right premises of the Cut rule are respectively $\uparrow_i R$ and $\uparrow_i L$:

$$\dfrac{\dfrac{\Delta|_i\overrightarrow{A} \Rightarrow B}{\Delta \Rightarrow B{\uparrow_i}A}\uparrow_i R \qquad \dfrac{\Gamma \Rightarrow A \qquad \Theta\langle\overrightarrow{B}\rangle \Rightarrow C}{\Theta\langle B{\uparrow_i}A|_i\Gamma\rangle \Rightarrow C}\uparrow_i L}{\Theta\langle\Delta|_i\Gamma\rangle \Rightarrow C}\,Cut$$

$$\rightsquigarrow$$

$$\dfrac{\Delta \Rightarrow A \qquad \dfrac{\Delta|_i\overrightarrow{A} \Rightarrow B \qquad \Theta\langle\overrightarrow{B}\rangle \Rightarrow C}{\Theta\langle\overrightarrow{A}|_i\Gamma\rangle \Rightarrow C}\,Cut}{\Theta\langle\Delta|_i\Gamma\rangle \Rightarrow C}\,Cut$$

- The rules applying at the left and right premises of the Cut rule are respectively IR and IL:

$$\dfrac{\dfrac{}{\Lambda \Rightarrow I}IR \qquad \dfrac{\Delta\langle\Lambda\rangle \Rightarrow A}{\Delta\langle I\rangle \Rightarrow A}IL}{\Delta\langle\Lambda\rangle \Rightarrow A}\,Cut$$

$$\rightsquigarrow$$

$$\Delta\langle\Lambda\rangle \Rightarrow A$$

- The rules applying at the left and right premises of the Cut rule are respectively JR and JL:

$$\dfrac{\dfrac{}{[] \Rightarrow J}JR \qquad \dfrac{\Delta\langle[]\rangle \Rightarrow A}{\Delta\langle\overrightarrow{J}\rangle \Rightarrow A}JL}{\Delta\langle[]\rangle \Rightarrow A}\,Cut$$

$$\rightsquigarrow$$

$$\Delta\langle[]\rangle \Rightarrow A$$

- The rules applying at the left and right premises of the Cut rule are respectively $\triangleleft^{-1}R$ and $\triangleleft^{-1}L$:

$$\dfrac{\dfrac{\Delta, [] \Rightarrow A}{\Delta \Rightarrow \triangleleft^{-1}A}\triangleleft^{-1}R \qquad \dfrac{\Gamma\langle\overrightarrow{A}\rangle \Rightarrow C}{\Gamma\langle\overrightarrow{A}, []\rangle \Rightarrow C}\triangleleft^{-1}L}{\Gamma\langle\Delta, []\rangle \Rightarrow B}\,Cut$$

$$\rightsquigarrow$$

$$\dfrac{\Delta, [] \Rightarrow A \qquad \Gamma\langle\overrightarrow{A}\rangle \Rightarrow C}{\Gamma\langle\Delta, []\rangle \Rightarrow C}\,Cut$$

- The rules applying at the left and right premises of the Cut rule are respectively $\triangleleft R$ and $\triangleleft L$:

$$\dfrac{\dfrac{\Delta \Rightarrow A}{\Delta, [] \Rightarrow \triangleleft A}\,\triangleleft R \qquad \dfrac{\Gamma\langle \overrightarrow{A}, []\rangle \Rightarrow C}{\Gamma\langle \overrightarrow{\triangleleft A}\rangle \Rightarrow C}\,\triangleleft L}{\Gamma\langle \Delta, []\rangle \Rightarrow B}\,Cut$$

$$\rightsquigarrow$$

$$\dfrac{\Delta \Rightarrow A \qquad \Gamma\langle \overrightarrow{A}, []\rangle \Rightarrow C}{\Gamma\langle \Delta, []\rangle \Rightarrow C}\,Cut$$

- The rules applying at the left and right premises of the Cut rule are respectively $\check{}^{i}R$ and $\check{}^{i}L$:

$$\dfrac{\dfrac{\Delta|_i\Lambda \Rightarrow A}{\Delta \Rightarrow \check{}^{i}A}\,\check{}^{i}R \qquad \dfrac{\Gamma\langle \overrightarrow{A}\rangle \Rightarrow C}{\Gamma\langle \overrightarrow{\check{}^{i}A}|_i\Lambda\rangle \Rightarrow C}\,\check{}^{i}L}{\Gamma\langle \Delta|_i\Lambda\rangle \Rightarrow C}\,Cut$$

$$\rightsquigarrow$$

$$\dfrac{\Delta|_i\Lambda \Rightarrow A \qquad \Gamma\langle \overrightarrow{A}\rangle \Rightarrow C}{\Gamma\langle \Delta|_i\Lambda\rangle \Rightarrow C}\,Cut$$

- The rules applying at the left and right premises of the Cut rule are respectively $\hat{}^{i}R$ and $\hat{}^{i}L$:

$$\dfrac{\dfrac{\Delta \Rightarrow A}{\Delta|_i\Lambda \Rightarrow \hat{}^{i}A}\,\hat{}^{i}R \qquad \dfrac{\Gamma\langle \overrightarrow{A}|_i\Lambda\rangle \Rightarrow C}{\Gamma\langle \overrightarrow{\hat{}^{i}A}\rangle \Rightarrow C}\,\hat{}^{i}L}{\Gamma\langle \Delta|_i\Lambda\rangle \Rightarrow C}\,Cut$$

$$\rightsquigarrow$$

$$\dfrac{\Delta|_i\Lambda \Rightarrow A \qquad \Gamma\langle \overrightarrow{A}|_i\Lambda\rangle \Rightarrow C}{\Gamma\langle \Delta|_i\Lambda\rangle \Rightarrow C}\,Cut$$

Let us see the nondeterministic cases:

Principal Cuts

- The rules applying at the left and right premises of the Cut rule are respectively $\circledcirc R$ and $\circledcirc L$:

$$\dfrac{\dfrac{\Delta \to A \qquad \Gamma \to B}{\Delta|_i\Gamma \to A \circledcirc B}\,\circledcirc R \qquad \dfrac{\Theta\langle \overrightarrow{A}|_1\overrightarrow{B}\rangle \to C \quad \cdots \quad \Theta\langle \overrightarrow{A}|_a\overrightarrow{B}\rangle \to C}{\Theta\langle \overrightarrow{A \circledcirc B}\rangle \to C}\,\circledcirc L}{\Theta\langle \Delta|_i\Gamma\rangle \to C}\,Cut$$

$$\rightsquigarrow$$

$$\dfrac{\dfrac{\Delta \to A \qquad \Theta\langle A|_iB\rangle \to C}{\Theta\langle \Delta|_iB\rangle \to C}\,Cut \qquad \Gamma \to B}{\Theta\langle \Delta|_i\Gamma\rangle \to C}\,Cut$$

- The rules applying at the left and right premises of the Cut rule are respectively $\Uparrow R$ and $\Uparrow L$:

$$\frac{\dfrac{\Delta|_1\overrightarrow{A} \to B \qquad \cdots \qquad \Delta|_x\overrightarrow{A} \to B}{\Delta \to B{\Uparrow}A}\Uparrow L \qquad \dfrac{\Gamma \to A \qquad \Theta\langle\overrightarrow{B}\rangle \to C}{\Theta\langle\overrightarrow{B{\uparrow}A}|_i\Gamma\rangle \to C}\Uparrow L}{\Theta\langle\Delta|_i\Gamma\rangle \to C}\,Cut$$

$$\leadsto$$

$$\frac{\dfrac{\Delta|_i\overrightarrow{A} \to B \qquad \Theta\langle\overrightarrow{B}\rangle \to B}{\Theta\langle\Delta|_i\overrightarrow{A}\rangle \to C}\,Cut \qquad \Gamma \to A}{\Theta\langle\Delta|_i\Gamma\rangle \to C}\,Cut$$

- The rules applying at the left and right premises of the Cut rule are respectively $\Downarrow R$ $\Downarrow L$:

$$\frac{\dfrac{\overrightarrow{A}|_1\Gamma \to B \qquad \cdots \qquad \overrightarrow{A}|_a\Gamma \to B}{\Gamma \to A{\Downarrow}B}\Downarrow R \qquad \dfrac{X \to A \qquad \Theta\langle\overrightarrow{B}\rangle \to C}{\Theta\langle X|_i\overrightarrow{A{\Downarrow}B}\rangle \to C}\Downarrow L}{\Theta\langle X|_i\Gamma\rangle \to C}\,Cut$$

$$\leadsto$$

$$\frac{\dfrac{\overrightarrow{A}|_i\Gamma \to B \qquad \Theta\langle B\rangle \to C}{\Theta\langle\overrightarrow{A}|_i\Gamma\rangle \to C}\,Cut \qquad X \to A}{\Theta\langle X|_i\Gamma\rangle \to C}\,Cut$$

Permutation Conversions

- The rule applying at the left premise of the Cut rule is $\circledcirc L$:

$$\frac{\dfrac{\Delta\langle\overrightarrow{B}|_1\overrightarrow{C}\rangle \to A \qquad \cdots \qquad \Delta\langle\overrightarrow{B}|_b\overrightarrow{C}\rangle \to A}{\Delta\langle\overrightarrow{B \circledcirc C}\rangle \to A}\circledcirc L \qquad \Gamma\langle\overrightarrow{A}\rangle \to D}{\Gamma\langle\Delta\langle\overrightarrow{B \circledcirc C}\rangle\rangle \to D}\,Cut$$

$$\leadsto$$

$$\frac{\dfrac{\Delta\langle\overrightarrow{B}|_1\overrightarrow{C}\rangle \to A \qquad \Gamma\langle\overrightarrow{A}\rangle \to D}{\Gamma\langle\Delta\langle\overrightarrow{B}|_1\overrightarrow{C}\rangle\rangle \to D}\,Cut \qquad \cdots \qquad \dfrac{\Delta\langle\overrightarrow{B}|_b\overrightarrow{C}\rangle \to A \qquad \Gamma\langle\overrightarrow{A}\rangle \to D}{\Gamma\langle\Delta\langle\overrightarrow{B}|_b\overrightarrow{C}\rangle\rangle \to D}\,Cut}{\Gamma\langle\Delta\langle\overrightarrow{B \circledcirc C}\rangle\rangle \to D}\circledcirc L$$

- The rule applying at the left premise of the Cut rule is $\Downarrow L$:

$$\frac{\dfrac{\Delta \to B \qquad \Gamma\langle\overrightarrow{C}\rangle \to A}{\Gamma\langle\Delta|_i\overrightarrow{B{\Downarrow}C}\rangle \to A}\Downarrow L \qquad Z\langle\overrightarrow{A}\rangle \to D}{Z\langle\Gamma\langle\Delta|_i\overrightarrow{B{\Downarrow}C}\rangle\rangle \to D}\,Cut$$

$$\leadsto$$

$$\frac{\Delta \to B \qquad \dfrac{\Gamma\langle\overrightarrow{C}\rangle \to A \qquad Z\langle\overrightarrow{A}\rangle \to D}{Z\langle\Gamma\langle\overrightarrow{C}\rangle\rangle \to D}\,Cut}{Z\langle\Gamma\langle\Delta|_i\overrightarrow{B{\Downarrow}C}\rangle\rangle \to D}\Downarrow L$$

- The rule applying at the left premise of the Cut rule is $\Uparrow L$:

$$\dfrac{\Delta \to \overrightarrow{A} \qquad \dfrac{\Gamma\langle\overrightarrow{A}\rangle|_1\overrightarrow{B} \to C \qquad \cdots \qquad \Gamma\langle\overrightarrow{A}\rangle|_{S(\Gamma\langle\overrightarrow{A}\rangle)}\overrightarrow{B} \to C}{\Gamma\langle\overrightarrow{A}\rangle \to C\Uparrow B}\Uparrow R}{\Gamma\langle\Delta\rangle \to C\uparrow B}\,Cut$$

$$\rightsquigarrow$$

$$\dfrac{\dfrac{\Delta \to A \qquad \Gamma\langle\overrightarrow{A}\rangle|_1\overrightarrow{B} \to C}{\Gamma\langle\Delta\rangle|_1\overrightarrow{B} \to C}\,Cut \qquad \cdots \qquad \dfrac{\Delta \to A \qquad \Gamma\langle\overrightarrow{A}\rangle|_{S(\Gamma\langle\overrightarrow{A}\rangle)}\overrightarrow{B} \to C}{\Gamma\langle\Delta\rangle|_c B \to C}\,Cut}{\Gamma\langle\Delta\rangle \to C\uparrow B}\uparrow R$$

W.r.t. Cut elimination including the additive conjunction and disjunction, the proof is standard.

Chapter 4

Syntactical Interpretation

In this chapter we study what we call *syntactical interpretation* of the displacement calculi which we defined and studied from a proof theoretical point of view in Chapter 3. Why the term syntactical interpretation? The logic mainstream would say semantical interpretion or simply, semantics, for in fact our calculi are (noncommutative) substructural logics which are homomorphic with intuitionistic multiplicative additive linear logic with units (**MALL**). But we are doing type-logical grammar, and this involves *signs* (in the de Saussure tradition), which at least incorporate a semantical dimension, a syntactical/prosodic dimension and finally a sign combinatorial dimension, which in our framework is simply the use of (syntactic) types. Our logics constitute an instance of the so-called intuitionistic *type logics* which are used as a main means to parse natural language sentences and assign them what we call *semantic form*, or *logical form* if one prefers the generative terminological tradition. Semantics is for us the derivation of a theorem in our type logics, which thanks to the so-called Curry-Howard[1] homomorphism, is a term of the (extended[2]) simply typed lambda calculus. Adding logical constants to the simply typed lambda calculus give us the way to get (typed) higher-order logic and give a meaning to sentences of fragments of natural language. So, it is clear that we need another term to refer to what logicians call semantics, and Glyn Morrill with some collaboration of this author, decided to use the term syntactical interpretation.

In this chapter we address the model-theoretical interpretation of displacement calculi. We see different syntactical interpretations which all have in common that they are close to what linguists call *language models*. For many authors, language models are sets of string or trees. As has been clear in this thesis, for the author, language models are sets of strings. The syntactical interpretations we define preserve the closeness to objects which are meaningful for linguists.

More technically, we will see the soundness and completeness of four syntactical interpretations. This chapter has as eventual goal to give finally completeness results for the logics of discontinuity (in our words displacement logics) which have been in the (type logical) air for many years. The results of this

[1]In fact, in type logical grammar this important isomorphism/homomorphism is related to other famous scholars like Jim Lambek, Johan Van Benthem, and some more authors.

[2]We consider type logics which contain many connectives over and above the traditional implication.

chapter appear for the first time in a manuscript, i.e., in this thesis.

4.1 On Residuated Displacement Algebras and the Standard Syntactical Interpretation

We already presented displacement algebras in Chapters 2 and 3. We recall some definitions. A syntactical algebra is a free monoid with a distinguished element called the prime, in notation, $\mathcal{S} = \langle M, +, 0, 1\rangle$. A prime in a freely generated monoid is simply an element of the set of generators. Here then, the prime implies the choice of one element of the set of generators. In a syntactical algebra we can define a map called the sort map which we recall here:

$S(a) = 0$ for a prime a different from 1
$S(1) = 1$
$S(w_1 + w_2) = S(w_1) + S(w_2)$

From this definition, we see that what the sort map does is to count the number of primes composing a string of the syntactical algebra. From the syntactical algebra $\mathcal{S}$, we can induce a standard displacement algebra which is in fact an ω-indexed algebra:

$$\mathcal{A} = \langle \{L_i\}_{i\in\omega}, +, \{\times_k\}_{k>0}, 0, 1\rangle$$

Where the ω-indexed $\{L_i\}_{i\in\omega}$ are the sequence or collection of what we have called sort domains, which are defined as follows:

$$L_i \;=\; \{w : w \in M \text{ and } S(w) = i\}$$

A valuation is a mapping $v : \mathcal{A} \longrightarrow \bigcup 2^{L_i}$, where:

$$\mathcal{A} = \bigcup_{i\in\omega} \mathcal{A}_i$$

Where $\mathcal{A}_i$ ($i \in \omega$) are the sets of atomic types of sort i. Given the valuation v, we define the map $[\![\cdot]\!]^{\mathcal{A}}_v$ which (syntactically) interprets types. A notational convention: the map $[\![\cdot]\!]^{\mathcal{A}}_v$ will be simply denoted $[\![\cdot]\!]_v$, or even $[\![\cdot]\!]$. Let us see the recursive definition of $[\![\cdot]\!]^{\mathcal{A}}_v$ for the deterministic connectives (including the synthetic connectives).

$[\![I]\!]$	=	$\{0\}$	continuous unit
$[\![J]\!]$	=	$\{1\}$	discontinous unit
$[\![\triangleright A]\!]$	=	$\{1{+}s \mid s \in [\![A]\!]\}$	right injection
$[\![\triangleright^{-1} B]\!]$	=	$\{s \mid 1{+}s \in [\![B]\!]\}$	right projection
$[\![\triangleleft A]\!]$	=	$\{s{+}1 \mid s \in [\![A]\!]\}$	left injection
$[\![\triangleleft^{-1} B]\!]$	=	$\{s \mid s{+}1 \in [\![B]\!]\}$	left projection
$[\![\hat{}^{\,i} A]\!]$	=	$\{s_1{+}s_2 \mid s_1{+}1{+}s_2 \in [\![A]\!] \text{ and } i = S(s_1)+1\}$	bridge
$[\![\check{}^{\,i} B]\!]$	=	$\{s_1{+}1{+}s_2 \mid s_1 + s_2 \in [\![B]\!] \text{ and } i = S(s_1)+1\}$	split
$[\![A{\bullet}B]\!]$	=	$\{s_1{+}s_2 \mid s_1 \in [\![A]\!]\ \&\ s_2 \in [\![B]\!]\}$	(continuous) product
$[\![A\backslash C]\!]$	=	$\{s_2 \mid \forall s_1 \in [\![A]\!], s_1{+}s_2 \in [\![C]\!]\}$	under
$[\![C/B]\!]$	=	$\{s_1 \mid \forall s_2 \in [\![B]\!], s_1{+}s_2 \in [\![C]\!]\}$	over
$[\![A{\odot_i}B]\!]$	=	$\{s_1{+}s_2{+}s_3 \mid s_1{+}1{+}s_3 \in [\![A]\!]\ \&\ s_2 \in [\![B]\!] \text{ and } i = S(s_1)+1\}$	discontinuous product
$[\![A{\downarrow_i}C]\!]$	=	$\{s_2 \mid \forall s_1{+}1{+}s_3 \in [\![A]\!], s_1{+}s_2{+}s_3 \in [\![C]\!] \text{ and } i = S(s_1)+1\}$	infix
$[\![C{\uparrow_i}B]\!]$	=	$\{s_1{+}1{+}s_3 \mid \forall s_2 \in [\![B]\!], s_1{+}s_2{+}s_3 \in [\![C]\!] \text{ and } i = S(s_1)+1\}$	extract

We give now the interpretation of the (linear logic) additive connectives and the nondeterministic connectives:

$$
\begin{array}{rcll}
[\![A\&B]\!] & = & [\![A]\!] \cap [\![B]\!] & \text{additive conjunction} \\
[\![A \oplus B]\!] & = & [\![A]\!] \cup [\![B]\!] & \text{additive disjunction} \\
[\![C{\Uparrow}B]\!] & = & \bigcap_{i=1}^{S(C)-S(B)+1} [\![C{\uparrow}_i B]\!] & \text{nondeterministic extract} \\
[\![B{\Downarrow}C]\!] & = & \bigcap_{i=1}^{S(C)-S(B)+1} [\![B{\downarrow}_i C]\!] & \text{nondeterministic infix} \\
[\![A \circledcirc B]\!] & = & \bigcup_{i=1}^{S(A)} [\![A \odot_i B]\!] & \text{nondeterministic wrapping} \\
[\![\check{} B]\!] & = & \bigcap_{i=1}^{S(\check{}^{\,i} B)} [\![\check{}^{\,i} B]\!] & \text{nondeterministic split} \\
[\![\hat{} A]\!] & = & \bigcup_{i=1}^{S(A)} [\![\hat{}^{\,i} AB]\!] & \text{nondeterministic wrapping}
\end{array}
$$

This syntactical interpretation is what we have called in this thesis the *standard syntactical interpretation*. Later, we will see, as we said before, a generalization of syntactical algebras, the so-called general displacement algebras (in Chapter 2 we already introduced general displacement algebras). In fact, looking at the definition of the equational theory of general displacement algebras, we can abstract notions as sort, prime and freeness. This will be addressed at the end of the chapter.

4.2 Some Interesting Completeness results for fragments of D

Let us consider the following alphabet $\Sigma_D = \{p, \$, \backslash, /, \downarrow, \uparrow, \Uparrow, \Downarrow, \triangleleft^{-1}, \triangleright^{-1}, \langle, \rangle, [,], \surd\}$. We define the following sets:

(184)
$$
\begin{array}{rcl}
\Sigma_D^{*,0} & = & (\Sigma_D - \{\$\})^* \\
\Sigma_D^{*,n+1} & = & \Sigma_D^{*,n} \cdot \$ \cdot \Sigma_D^{*,0} \text{ for } n \geq 0
\end{array}
$$

The sets $\Sigma_D^{*,i}$ $(i \in \omega)$ can be then represented as follows:

$$\Sigma_D^{*,i} = \underbrace{{\Sigma_D}^{*,0} \cdot \$ \cdot {\Sigma_D}^{*,0} \cdot \ldots \cdot \$ \cdot {\Sigma_D}^{*,0}}_{i\ \$\text{'s}}$$

Our main goal is to code the set of types, segmented types and hyperconfigurations with the symbols of the finite set Σ_D:

$$\Sigma_D^* = \bigcup_{i \in \omega} \Sigma_D^{*,i}$$

We build now the sets $\widetilde{\mathcal{F}}$, $\widetilde{SF}$ and $\widetilde{\mathcal{O}}$. In Chapter 3 we saw that the set of atomics types of $\mathbf{D}$ can be represented as follows:

$$\begin{array}{rcl} \mathcal{A}_i & = & (p^i_j)_{j\in\omega} \\ \mathcal{A} & = & \bigcup_{i\in\omega} \mathcal{A}_i \end{array}$$

We restrict the set of types to what we call the *implicative* fragment, which we denote $\mathcal{F}[\rightarrow]$:

$$\mathcal{F}[\rightarrow] \stackrel{def}{=} \mathcal{F}[\backslash, /, \Downarrow, \Uparrow, \{\downarrow_i\}_{i>0}, \{\uparrow_i\}_{i>0}, \triangleleft^{-1}, \triangleright^{-1}]$$

The term *implicative* follows from the following facts. We consider the continuous and discontinuous connectives (deterministic or nondeterministic) which have a model-theoretical definition in terms of an implication, e.g.:

$$[\![B\uparrow_i A]\!] = \{t : \forall a[a \in [\![A]\!] \rightarrow t \times_i a \in [\![B]\!]]\}$$

Implicative unary connectives are those which in Chapter 3 were proved to be definable in terms of continuous or discontinous implications. We recall these definable unary connectives:

$$\begin{array}{rcl} \triangleleft^{-1}A & = & A/J \\ \triangleright^{-1}A & = & J\backslash A \\ \check{}^{\,i}A & = & A\uparrow_i I \\ \check{}A & = & A\Uparrow I \end{array}$$

We can code the set of types with the symbols of Σ_D. We define a mapping ρ at the level of atomic types and then recursively for the remaining (implicative) types:

$$\begin{array}{llll} \rho: & \mathcal{F} & \longrightarrow & \Sigma_D^* \\ & p^i_j \in \mathcal{A} & \mapsto & \langle 0^i\, p\, 0^j\rangle \\ & B/A & \mapsto & \langle \rho(B)/\rho(A)\rangle \\ & A\backslash B & \mapsto & \langle \rho(A)\backslash\rho(B)\rangle \\ & B\uparrow_k A & \mapsto & \langle \rho(B)\langle\uparrow 0^k\rangle\rho(A)\rangle \\ & A\downarrow_k B & \mapsto & \langle \rho(A)\langle\downarrow 0^k\rangle\rho(B)\rangle \\ & B\Uparrow A & \mapsto & \langle \rho(B)\Uparrow\rho(A)\rangle \\ & (A\Downarrow B) & \mapsto & \langle \rho(A)\Downarrow\rho(B)\rangle \\ & \triangleleft^{-1}A & \mapsto & \langle \triangleleft^{-1}\rho(A)\rangle \\ & \triangleright^{-1}A & \mapsto & \langle \triangleright^{-1}\rho(A)\rangle \end{array}$$

Notice we do not include the unary connectives $(\check{}^{\,i+1})_{i\in\omega}$ and $\check{}$.[3] Segmented types can be coded as well. The coding mapping is again called ρ:

$$\begin{array}{llll} \rho: & \mathcal{SF} & \longrightarrow & \Sigma_D^* \\ & \sqrt[i]{A} & \mapsto & [0^i\sqrt{\rho(A)}] \end{array}$$

Finally we code the hyperconfigurations. For the sake of commodity for the reader, we show again the (unambiguous) recursive definition of hyperconfigurations:

[3]The reason will be apparent later in a remark after the truth lemma (193).

$$\begin{array}{lcl}
\mathcal{O} & ::= & \Lambda \\
\mathcal{O} & ::= & A, \mathcal{O} \text{ for } S(A) = 0 \\
\mathcal{O} & ::= & [], \mathcal{O} \\
\mathcal{O} & ::= & \sqrt[0]{A}, \mathcal{O}, \sqrt[1]{A}, \ldots, \sqrt[S(A)-1]{A}, \mathcal{O}, \sqrt[S(A)]{A}, \mathcal{O} \\
 & & \text{for } S(A) > 0
\end{array}$$

The coding of $\widetilde{\mathcal{O}}$ is therefore defined recursively as follows:

$$\begin{array}{lcl}
\widetilde{\mathcal{O}} & ::= & \Lambda \\
\widetilde{\mathcal{O}} & ::= & \rho(A), \widetilde{\mathcal{O}} \text{ for } S(A) = 0 \\
\widetilde{\mathcal{O}} & ::= & \$, \widetilde{\mathcal{O}} \\
\widetilde{\mathcal{O}} & ::= & \rho(\sqrt[0]{A}), \widetilde{\mathcal{O}}, \rho(\sqrt[1]{A}), \ldots, \rho(\sqrt[S(A)-1]{A}), \widetilde{\mathcal{O}}, \rho(\sqrt[S(A)]{A}), \widetilde{\mathcal{O}} \\
 & & \text{for } S(A) > 0
\end{array}$$

The ρ map codes three types of sets. It is readily seen that this overloaded mapping is injective:

$$(185) \quad \begin{array}{lcl}
\rho : \mathcal{F} & \longrightarrow & \Sigma_D^* \\
\rho : \mathcal{SF} & \longrightarrow & \Sigma_D^* \\
\rho : \mathcal{O} & \longrightarrow & \Sigma_D^*
\end{array}$$

The images of ρ are denoted respectively $\widetilde{\mathcal{F}}$, $\widetilde{\mathcal{SF}}$ and $\widetilde{\mathcal{O}}$. Because of the fact that ρ is injective, $\widetilde{\mathcal{F}}$, $\widetilde{\mathcal{SF}}$ and $\widetilde{\mathcal{O}}$ are in bijection respectively with $\mathcal{F}$, $\mathcal{SF}$ and $\mathcal{O}$. Moreover, the ρ mapping from $\mathcal{O}$ into $\widetilde{\mathcal{O}}$ is an isomorphism of displacement algebras. Later, we will see that both $\mathcal{O}$ and $\widetilde{\mathcal{A}}$ constitute a generalization of the concept of displacement algebras. Without terminological surprise, these displacement algebras will be called *general displacement algebras*, which as we shall see, will become crucial to get different sound and complete syntactical interpretations of the different displacement calculi.

4.2.1 A canonical model for the implicative fragment $\mathcal{F}[\to]$

We are now in a position to define a standard displacement algebra which is moreover finitely generated:

(186)

$$\mathcal{A}_{\mathrm{S}} = \langle \{\Sigma_D^{*,i}\}_{i\in\omega}, (\cdot,\cdot), \{|_i\}_{i>0}, \Lambda, \$\rangle \ (*)$$

The prime is here denoted \$ (which corresponds in fact to the so-called separator of the set of hyperconfigurations). The operations $(\cdot,\cdot)$ (which stands for concatenation) and $\{|_{i+1}\}_{i\in\omega}$ (which stand for $i+1$-th intercalation) mimic the operations defined on the set of hyperconfigurations. From $\mathcal{A}$ we will build a canonical model which will give us a completeness result in the standard semantics because the displacement algebra $\mathcal{A}$ is standard, i.e. it is induced by a syntactical algebra, in this case $\langle \Sigma_D^*, (,), \$, \Lambda\rangle$. The following set-theoretical equality holds:

$$(187) \quad L_i = \underbrace{\Sigma_D^{*,0} \cdot 1 \cdot \Sigma_D^{*,0} \cdot \ldots \cdot \Sigma_D^{*,0} \cdot 1 \cdots \Sigma_D^{*,0}}_{i \text{ 1's}}$$

(188) **Remark**
Notice that the Σ_D-algebra $\mathcal{C} = \langle \{\widetilde{\mathcal{O}}_i\}_{i\in\omega}, (\cdot,\cdot), \{|_i\}_{i>0}, \Lambda, \$\rangle$ is a (general) displacement algebra which is not standard for the first condition of the separation property ((54) defined in Chapter 2) is not satisfied, and we know that the separation property holds of **FreeDisp**. In fact we have that $\mathcal{C}$ is a subalgebra of $\mathcal{A}$, and their sort domains are in a relation of proper inclusion:

$$\widetilde{\mathcal{O}_i} \subsetneq L_i,\ i \in \omega$$

We need some technical results in order to prove the truth lemma which will be crucial for the construction of the canonical model. Given a $t[\$] \in \Sigma_D^*$ and $\gamma \in \widetilde{\mathcal{O}}$, we say that γ *well-occurs* in $t[\gamma] \in \Sigma_D^*$ iff $t[\$] \in \widetilde{\mathcal{O}}$.

(189) **Lemma** (*Results on Substrings of the Image of the Mapping* ρ)

The following properties hold:

i) Let A be an arbitrary type of sort 0. Let $t[\$] = \alpha, \$, \beta \in \Sigma_D^*$. We have that $\rho(A, A/A)$ well-occurs in $t[\rho(A, A/A)]$.

ii) Let A be an arbitrary type of sort strictly greater than 0. We have then that $\rho(\overrightarrow{A})$ well-occurs in $t[\rho(\overrightarrow{A})]$.

Proof. We proceed for both cases i) and ii) by induction on the structure of derivations of $\widetilde{\mathcal{O}}$ (by the unambiguous BNF grammar generating $\widetilde{\mathcal{O}}$):

i) - Suppose $t[\rho(A, A/A)] = \rho(B), \alpha$ where $S(B) = 0$ and $\alpha \in \widetilde{\mathcal{O}}$. We have that $\rho(A, A/A) = \rho(A), \rho(A/A)$. Clearly $\rho(A), \rho(A/A)$ cannot be a proper infix $\rho(B)$ because in that case $\rho(B)$ would contain two adjacent types (of sort 0) coded (by ρ), which is not possible. By the same token, $\rho(A), \rho(A/A)$ could not be a proper prefix or suffix of $\rho(B)$. By the balancing (of parenthesis) properties, a proper suffix of $\rho(B)$ cannot be a prefix of $\rho(A), \rho(A/A)$. Hence, $\rho(A), \rho(A/A)$ must occur in α. By induction hypothesis (i.h.), it follows that $\rho(A), \rho(A/A)$ well-occurs in α. Summing up, $\rho(A), \rho(A/A)$ well-occurs in $\rho(B), \alpha$.

- Suppose $t[\rho(A, A/A)] = \$, \alpha$ with $\alpha \in \widetilde{\mathcal{O}}$. Necessarily, $\rho(A, A/A)$ well-occurs in α because \$ does not appear in $\rho(A, A/A)$.

- Suppose that:

$$t[\rho(A, A/A)] = \rho(\sqrt[0]{B}), \alpha_1, \rho(\sqrt[1]{B}), \alpha_2 \cdots, \rho(\sqrt[S(B)-1]{B}), \alpha_{S(B)}, \rho(\sqrt[S(B)]{B}), \beta$$

where $\alpha_i, \beta \in \widetilde{\mathcal{O}} (i = 1, \cdots, S(B))$. $\rho(A), \rho(A/A)$ cannot be a prefix or suffix of $\rho(\sqrt[i]{B})$ $(i = 1, \cdots, S(B))$ for in that case $\rho(A), \rho(A/A)$ would contain the symbols [or]. $\rho(A), \rho(A/A)$ cannot be an infix of a type segment for it would contain two coded adjacent types (of sort 0). We are done.

ii) Suppose $S(A) > 0$.

- Suppose that $t[\rho(\overrightarrow{A})] = \rho(B), \alpha$, where $S(B) = 0$ and $\alpha \in \widetilde{\mathcal{O}}$: $\rho(\overrightarrow{A})$ can be neither a proper infix of $\rho(B)$ nor a proper suffix. $\rho(\overrightarrow{A})$ cannot be equal to $\rho(B)$. Consequently, if $\rho(\overrightarrow{A})$ occurs in t it must occur in α and hence, by i.h., it well-occurs in α.
- Suppose that $t[\rho(\overrightarrow{A})] = \$, \alpha$, where $\alpha \in \widetilde{\mathcal{O}}$. In this case, $\rho(\overrightarrow{A})$ forcefully occurs in α. Hence, by i.h., $\rho(\overrightarrow{A})$ well-occurs in t.
- Suppose that $t[\rho(\overrightarrow{A})]$ has the form:

$$\sqrt[0]{\rho(B)}, \alpha_1, \sqrt[1]{\rho(B)}, \alpha_2, \cdots, \sqrt[S(B)-1]{\rho(B)}, \alpha_{S(B)}, \sqrt[S(B)]{\rho(B)}, \beta$$

where $\alpha_i, \beta \in \widetilde{\mathcal{O}}$. We have that $\rho(\overrightarrow{A})$ cannot occur as a proper infix of a type segment, in this case in $\sqrt[i]{\rho(B)}$, for in that case t would not belong to $\widetilde{\mathcal{O}}$. And $\rho(\overrightarrow{A})$ cannot occur as a proper prefix or suffix of a type segment, in this case in $\sqrt[i]{\rho(B)}$, for in that case t it would not belong to $\widetilde{\mathcal{O}}$. $\rho(\overrightarrow{A})$ could be equal to (but not contain as a proper prefix):

$$\sqrt[0]{\rho(B)}, \alpha_1, \sqrt[1]{\rho(B)}, \cdots, \sqrt[S(B)-1]{\rho(B)}, \alpha_{S(B)-1}, \sqrt[S(B)]{\rho(B)}$$

In that case, necessarily $A = B$ and $\alpha_i = []$ $(i = 1, \cdots, S(B))$. Otherwise, $\rho(\overrightarrow{A})$ can only occur in α_i $(i = 1, \cdots, S(B))$ or β. Hence, by i.h. $\rho(\overrightarrow{A})$ would well-occur in $t[\rho(\overrightarrow{A})]$.

□

(190) **Remark**

Notice that for example the string $\langle \$ \rangle \in \Sigma_D^{*,1}$ does not belong to $\widetilde{\mathcal{O}}$. But if we replace \$ by p we obtain $\langle p \rangle \in \widetilde{\mathcal{O}}$. By i) of the previous lemma, if \$ in $\langle \$ \rangle$ is substituted by $\rho(A), \rho(A/A)$ for an arbitrary type A of sort 0 then the result is a string of $\widetilde{\mathcal{O}}$. This will be crucial for the truth lemma (193). On the other hand, if the result of replacing the \$ of a string $t[\$] \in \Sigma_D^*$ by the empty string is a $\widetilde{\mathcal{O}}$ string, $t[\$]$ is not necessarily a $\widetilde{\mathcal{O}}$. Consider for example $\langle \$p \rangle$.

We construct the canonical power-set displacement algebra over the standard displacement algebra $\mathcal{A}_\mathrm{S}$:

$$\mathcal{CA}_\mathrm{S} = \langle \{2^{\Sigma_D^{*,i}}\}_{i\in\omega}, \circ, \{\circ_{i+1}\}_{i\in\omega}, \{\Lambda\}, \{\$\} \rangle$$

Here the operators $\circ$, and $\{\circ_i\}_{i>0}$ are induded by the operations of the displacement algebra. $\mathcal{CA}_\mathrm{S}$ has moreover induced residuated analogues of the discontinuous connectives $\{\bullet, \backslash, /, (\odot_{i+1})_{i\in\omega}, (\downarrow_{i+1})_{i\in\omega}, (\uparrow_{i+1})_{i\in\omega}\}$. The partial order associated with $\mathcal{CA}_\mathrm{S}$ is the inclusion $\subseteq$.
Let us consider the following valuation v on the set of atomic types $\mathcal{A}$:

$$(191) \quad \begin{array}{lll} v : \mathcal{A} & \longrightarrow & \bigcup_{i\in\omega} 2^{L_i} \\ p_j^i & \mapsto & v(p_j^i) = \{\delta : \delta \in \widetilde{\mathcal{O}} \text{ and } \vdash_{\mathbf{hD}} \rho^{-1}(\delta) \Rightarrow p_j^i\} \end{array}$$

We consider the canonical model $\mathcal{CM}_\mathrm{S} = (\mathcal{CA}_\mathrm{S}, v)$. Let us define the following operator:

$$(192)\quad \begin{array}{llll} \vdash^{-1}: & \mathcal{F} & \longrightarrow & \bigcup_{i\in\omega} 2^{\Sigma_D^{*,i}} \\ & A & \mapsto & \vdash^{-1}(A) \stackrel{def}{=} \{\delta : \delta \in \widetilde{\mathcal{O}} \text{ and } \vdash_{\mathbf{hD}} \rho^{-1}(\delta) \Rightarrow A\} \end{array}$$

(193) **Lemma** (*Truth Lemma*)

- For every type $A, B \in \mathcal{F}[\rightarrow]$ and for every connective $* \in \{\backslash, /, \Downarrow, \Uparrow, \{\downarrow_{i+1}\}_{i\in\omega}, \{\uparrow_{i+1}\}_{i\in\omega}\}$:

$$[\![B * A]\!]_v^{\mathcal{CM}\text{s}} \quad = \quad \vdash^{-1}(B * A)$$

- For every type $A, B \in \mathcal{F}[\rightarrow]$ and for every connective $* \in \{\triangleleft^{-1}, \triangleright^{-1}\}$:

$$[\![*A]\!]_v^{\mathcal{CM}\text{s}} \quad = \quad \vdash^{-1}(*A)$$

Proof. By induction on the structure of the types generated from the implicative fragment. For the set $\mathcal{A}$ of atomic types we are done by the fact that in the valuation in the canonical model we have by definition (see (191)):

$$v(p_j^i) \quad = \quad \vdash^{-1}(p_j^i)$$

In the following we write $\vdash \Delta \Rightarrow A$ instead of $\vdash_{\mathbf{hD}} \Delta \Rightarrow A$ and $[\![\cdot]\!]$ instead of $[\![\cdot]\!]_v^{\mathcal{CM}\text{s}}$.

- Binary connectives:
  - $/$: We want to prove:

$$[\![B/A]\!] \quad = \quad \vdash^{-1}(B/A)$$

[$\supseteq$]: let $\delta \in \vdash^{-1}(B/A)$. We have therefore that $\delta \in \widetilde{\mathcal{O}}$ and $\rho^{-1}(\delta) \in \vdash^{-1}(B/A)$. In particular:

$$\rho^{-1}(\delta), \overrightarrow{A} \Rightarrow B$$

By induction hypothesis (i.h.) $[\![A]\!] = \vdash^{-1}(A)$. Hence for every $\gamma_A \in [\![A]\!]$ we have that $\vdash \rho^{-1}(\gamma_A) \Rightarrow A$. Consider the following derivation:

$$\dfrac{\dfrac{\rho^{-1}(\delta) \Rightarrow B/A \qquad \rho^{-1}(\gamma_A) \Rightarrow A}{\rho^{-1}(\delta), \rho^{-1}(\gamma_A) \Rightarrow B/A\bullet A}\bullet R \qquad B/A \bullet A \Rightarrow B}{\rho^{-1}(\delta), \rho^{-1}(\gamma_A) \Rightarrow B}\,Cut$$

Therefore $\rho(\rho^{-1}(\delta), \rho^{-1}(\gamma_A)) = \rho(\rho^{-1}(\delta)), \rho(\rho^{-1}(\gamma_A)) = \delta, \gamma_A$ with $\delta, \gamma_A \in \vdash^{-1}(B)$ and since by i.h. $\vdash^{-1}(B) = [\![B]\!]$, we have that $\delta, \gamma_A \in [\![B]\!]$. We are done.

[$\subseteq$]: Let $\delta \in [\![B/A]\!]$. $\rho(\overrightarrow{A}) \in \vdash^{-1}(A)$, for $\rho(\overrightarrow{A}) \in \widetilde{\mathcal{O}}$ and:

$$\vdash \rho^{-1}(\rho(\overrightarrow{A})) = \overrightarrow{A} \Rightarrow A$$

Since by i.h. $\vdash^{-1}(A) = [\![A]\!]$ we have that $\rho(\overrightarrow{A}) \in [\![A]\!]$. By assumption $\delta \in [\![B/A]\!]$. It follows that:

$$\delta, \rho(\overrightarrow{A}) \in [\![B]\!]$$

By i.h. $\vdash^{-1}(B) = [\![B]\!]$, and hence:

$$\rho^{-1}(\delta, \rho(\overrightarrow{A})) \in \widetilde{\mathcal{O}} \text{ and } \rho^{-1}(\delta, \rho(\overrightarrow{A})) \Rightarrow B$$

By lemma (189) $\delta \in \widetilde{\mathcal{O}}$. Hence $\rho^{-1}(\delta, \rho(\overrightarrow{A})) = \rho^{-1}(\delta), \overrightarrow{A} \in \widetilde{\mathcal{O}}$. Summing up:

$$\vdash \rho^{-1}(\delta), \overrightarrow{A} \Rightarrow B$$

By application of the right rule of / we have:

$$\frac{\rho^{-1}(\delta), \overrightarrow{A} \Rightarrow B}{\rho^{-1}(\delta) \Rightarrow B/A}\,/R$$

We have therefore that $\delta \ \vdash^{-1}(B/A)$.

– $\uparrow_i$: Here i ranges over $\{1, \cdots, S(B) - S(A) + 1\}$ We want to prove:

$$[\![C{\uparrow_i}B]\!] \quad = \quad \vdash^{-1}(C{\uparrow_i}B)$$

[⊇]: Let $\delta \in \vdash^{-1}(C{\uparrow_i}B)$. Call Δ to $\rho^{-1}(\delta)$. By assumption:

$$\delta \in \widetilde{\mathcal{O}} \quad \text{and} \quad \vdash \rho^{-1}(\delta) = \Delta \Rightarrow C{\uparrow_i}B$$

We have to see that for any $\gamma_B \in [\![B]\!]$ $\delta|_i\gamma_B \in [\![C]\!]$. By i.h. we have:

$$\begin{array}{lcl} [\![B]\!] & = & \vdash^{-1}(B) \\ \gamma_B \in \widetilde{\mathcal{O}} & \text{and} & \vdash \rho^{-1}(\gamma_B) \Rightarrow B \end{array}$$

$$\frac{\dfrac{\rho^{-1}(\delta) \Rightarrow C{\uparrow_i}B \qquad \rho^{-1}(\gamma_B) \Rightarrow B}{\rho^{-1}(\delta)|_i\rho^{-1}(\gamma_B) \Rightarrow (C{\uparrow_i}B){\odot_i}B}\,\odot_i \qquad (C{\uparrow_i}B){\odot_i}B \Rightarrow B}{\rho^{-1}(\delta)|_i\rho^{-1}(\gamma_B) \Rightarrow C}\,Cut$$

By i.h. $[\![C]\!] = \vdash^{-1}(C)$. Since $\delta|_i\gamma_B \in \vdash^{-1}(C)$, it follows that $\delta|_i\gamma_B \in [\![C]\!]$ which proves the inclusion.

[⊆]: Let $\delta \in [\![C{\uparrow_i}B]\!]$. By lemma (189), if $S(B) > 0$, from the fact that $\delta|_i\rho(\overrightarrow{A}) \in \widetilde{\mathcal{O}}$, then $\delta \in \widetilde{\mathcal{O}}$. If $S(B) = 0$, from the fact that $\delta|_i\rho(A, A/A) \in \widetilde{\mathcal{O}}$ then $\delta \in \widetilde{\mathcal{O}}$. In either case we have that $\delta \in \widetilde{\mathcal{O}}$. By i.h. $[\![B]\!] = \vdash^{-1}(B)$. Hence $\rho(\overrightarrow{B}) \in \vdash^{-1}(B)$. By definition of $[\![C{\uparrow_i}B]\!]$:

$$\delta|_i\, \rho(\overrightarrow{B}) \in [\![C]\!]$$

By i.h. $\Delta|_i\overrightarrow{B} \Rightarrow C$. Hence:

$$\frac{\Delta|_i\overrightarrow{B} \Rightarrow C}{\Delta \Rightarrow C\uparrow_i B}\uparrow_i R$$

It follows then $\delta \in \vdash^{-1}(C\uparrow_i B)$. This proves the inclusion $[\subseteq]$.

– Nondeterministic $\Uparrow$. We want to prove:

$$[\![C\Uparrow B]\!] = \vdash^{-1}(C\Uparrow B)$$

$[\supseteq]$: Let $\delta \in \vdash^{-1}(C\Uparrow B)$

$$\frac{\dfrac{\rho^{-1}(\delta) \Rightarrow C\Uparrow B \qquad \gamma_B \Rightarrow B}{\rho^{-1}(\delta)|_i\gamma_B \Rightarrow (C\Uparrow B)\odot_i B}\odot R \qquad \dfrac{\dfrac{\overrightarrow{C} \Rightarrow C \qquad \overrightarrow{B} \Rightarrow B}{\overrightarrow{(C\Uparrow B)}|_i\overrightarrow{B} \Rightarrow C}\Uparrow L}{\overrightarrow{(C\Uparrow B)\odot_i B} \Rightarrow C}\odot L}{\rho^{-1}(\delta)|_i\gamma_B \Rightarrow C \text{ for every } i \in \{1, \cdots, S(C) - S(B) + 1\}}Cut$$

By i.h. :

$$[\![C]\!] = \vdash^{-1}(C)$$

Hence for every $i \in \{1, \cdots, S(C) - S(B) + 1\}$:

$$\rho(\rho^{-1}(\delta)|_i\rho^{-1}(\gamma_B)) = \delta|_i\gamma_B \in [\![C]\!]$$

The inclusion $[\supseteq]$ has been proved.

$[\subseteq]$: For every $i \in 1, \cdots, S(C) - S(B) + 1$:

$$\delta|_i\gamma_B \in [\![C]\!]$$

By a reasoning similar to the case $[\subseteq]$ for $\uparrow_i$ we have that:

$$\delta \in \widetilde{\mathcal{O}} \text{ and } \rho^{-1}(\delta|_i\rho(\overrightarrow{B})) \Rightarrow C$$

Since $\rho^{-1}(\delta|_i\rho(\overrightarrow{B})) = \rho^{-1}(\delta)|_i\overrightarrow{B}$ we have that:

$$\text{For every } i\ \rho^{-1}(\delta)|_i\overrightarrow{B} \Rightarrow C$$

and hence:

$$\frac{\rho^{-1}(\delta)|_1\overrightarrow{B} \Rightarrow C \qquad \cdots \qquad \rho^{-1}(\delta)|_{S(C)-S(B)+1}\overrightarrow{B} \Rightarrow C}{\rho^{-1}(\delta) \Rightarrow C}\Uparrow R$$

Therefore $\delta \in \vdash^{-1}(C)$.

- Unary connectives:

– $\triangleleft^{-1}$:

[$\supseteq$]: Let $\delta \in \vdash^{-1}(\triangleleft^{-1}A)$. It follows that:

$$\dfrac{\dfrac{\rho^{-1}(\delta) \Rightarrow \triangleleft^{-1}A}{[], \rho^{-1}(\delta) \Rightarrow \triangleleft\triangleleft^{-1}A}\triangleleft^{-1}R \qquad \overrightarrow{\triangleleft\triangleleft^{-1}A} \Rightarrow A}{[], \rho^{-1}(\delta) \Rightarrow A} Cut$$

We have $\rho([], \rho^{-1}(\delta)) = \$ \cdot \rho(\rho^{-1}(\delta) = \$ \cdot \delta$. Since by i.h. $\vdash^{-1}(A) = [\![A]\!]$, $\$ \cdot \delta \in [\![A]\!]$. We are done.

[$\subseteq$]: Consider $\delta \in [\![\triangleleft^{-1}A]\!]$. By definition:

$$\$ \cdot \delta \in [\![A]\!]$$

By i.h. $\vdash^{-1}(A) = [\![A]\!]$. Hence:

$$\rho(\$ \cdot \delta) \in \widetilde{\mathcal{O}} \text{ and } \vdash \rho^{-1}(\rho(\$ \cdot \delta)) \Rightarrow A$$

Now, again by lemma (189) we have that $\delta \in \widetilde{\mathcal{O}}$. Therefore, $\rho^{-1}(\$ \cdot \delta) = [], \delta$ and it follows that:

$$\vdash [], \delta \Rightarrow A$$

By right application of $\triangleleft^{-1}$ we obtain:

$$\dfrac{\vdash [], \delta \Rightarrow A}{\vdash [], \delta \Rightarrow \triangleleft^{-1}A}\triangleleft^{-1}R$$

We are done.

The other implicative connectives follow similar reasonings.
□

(194) **Remark**
We could have tried to include in what we have called implicative connectives the unary $\check{}^{i}$ and $\check{}$. But the result would not go through in the case of an underlying finitely generated syntactical algebra. For in the standard interpretation $t[\$] := \langle \$p \rangle$ is such that $t[\Lambda] = \rho(p_0^0) \in \widetilde{\mathcal{O}}$, but $t[\$] \notin \widetilde{\mathcal{O}}$. Nevertheless, if we considered a syntactical algebra generated by the infinite alphabet $\Sigma_D = \mathcal{F}[\rightarrow] \cup \mathcal{SF}[\rightarrow] \cup \{[]\}$ then the truth lemma for the connectives $\check{}^{i}$ and $\check{}$ would work.

(195) **Lemma**
For any $\Delta \in \mathcal{O}$, $\rho(\Delta) \in [\![\Delta]\!]$.

Proof. By induction on the complexity of $\mathcal{O}$. □

(196) **Theorem** (*Completeness of the Restricted Implicative fragment of* **D** *w.r.t. Displacement Algebras induced by Finitely Generated Syntactical Algebras*)

D is complete w.r.t. the class of powerset residuated displacement algebras over finitely finitely generated syntactical algebras in the so-called restricted implicative fragment:

$$\mathbf{D}[\backslash, /, \Downarrow, \Uparrow, \{\downarrow_i\}_{i>0}, \{\uparrow_i\}_{i>0}, \triangleleft^{-1}, \triangleright^{-1}]$$

Proof. Let $\mathcal{S} = \Delta \Rightarrow A$ be a hypersequent valid in every powerset displacement model over a finitely generated free syntactical algebra. Here, $[\![]\!]_v^{\mathcal{CM}_S}$ will be simply denoted $[\![\cdot]\!]$. In particular, $\mathcal{S}$ holds of the canonical model defined above:

$$\mathcal{CM}_S \models \Delta \Rightarrow A$$

By the previous lemma, we have that $\rho(\Delta) \in [\![\Delta]\!]$. Since $[\![\Delta]\!] \subseteq [\![A]\!]$, $\rho(\Delta) \in [\![A]\!]$. By the truth lemma:

$$\rho(\Delta) \in \vdash^{-1}(A)(\star)$$

We have then that $\rho^{-1}(\rho(\Delta)) = \Delta$ and $\vdash \Delta \Rightarrow A$. So we are done. □

We present now an extended implicative completeness result where we add the connectives $\{\check{}^{i+1}\}_{i\in\omega}$ and $\check{}$. But this result is weaker than theorem (196) in the sense that we no longer interpret types in *finitely* generated syntactical algebras.

(197) **Theorem** (*Completeness of the Implicative fragment of* **D** *w.r.t. Displacement Algebras induced by Syntactical Algebras*)

D is complete w.r.t. the class of powerset residuated displacement algebras over syntactical algebras in the so-called implicative fragment:

$$\mathbf{D}[\backslash, /, \Downarrow, \Uparrow, \{\downarrow_i\}_{i>0}, \{\uparrow_i\}_{i>0}, \triangleleft^{-1}, \triangleright^{-1}, (\check{}^{i+1})_{i\in\omega}, \check{}\,]$$

Proof. The proof is similar to the one of theorem (196). But the proof of this theorem is easier because we are considering the class of syntactical algebras[4], a fact which allows us to build a simpler canonical model based on the canonical displacement algebra:

$$\mathcal{CA}_S = \langle \{2^{\Sigma_D^{*,i}}\}_{i\in\omega}, \circ, \{\circ_{i+1}\}_{i\in\omega}, \{\Lambda\}, \{\$\}\rangle$$

Where $\Sigma_D^* \stackrel{def}{=} (\mathcal{F}_0 \cup \mathcal{SF} \cup \{[]\})^*$. Of course, it is not finitely generated. Here Σ_D^* is a set graduated by the collection of $\Sigma_D^{*,i}$ which are simply the subsets Σ_D^* whose strings contain i []'s. Given an arbitrary type A, the operator is defined in such a way that $\vdash^{-1}(A) \stackrel{def}{=} \{\Delta : \Delta \in \mathcal{O} \text{ and } \vdash \Delta \Rightarrow A\}$.

We have now only to check the connectives $(\check{}^{i+1})_{i\in\omega}$ and nondeterministic $\check{}$:

- Deterministic $\check{}^{k}$:

  [⊇]: Let $\delta \in \vdash^{-1}(\check{}^{k} A)$. It follows that:

[4]Which properly contains the class of finitely generated syntactical algebras.

$$\frac{\delta \Rightarrow \check{}_k(A)}{\Delta|_k\Lambda \Rightarrow \hat{}_k\check{}_k(A)}\ \hat{}_k R$$

Since $\hat{}_k\check{}_k(A) \Rightarrow A$ it follows that:

$$\frac{\Delta|_k\Lambda \Rightarrow \hat{}_k\check{}_k(A) \qquad \overrightarrow{\hat{}_k\check{}_k A} \Rightarrow A}{\Delta|_k\Lambda \Rightarrow A}\ Cut$$

It follows that $\Delta|_k\Lambda \in \vdash^{-1}(A)$. Hence, by i.h. $\Delta|_k\Lambda \in [\![A]\!]$. We are done.

[⊆]:
Let $\Delta \in [\![\check{}_k A]\!]$. It follows that $\Delta|_k\Lambda \in [\![A]\!]$. By i.h. $\Delta|\Lambda \Rightarrow A$. Hence:

$$\frac{\Delta|_k\Lambda \Rightarrow A}{\Delta \Rightarrow \check{}_k A}\ \check{}_k R$$

It follows that $\Delta \in \vdash^{-1}\check{}_k(A)$. This proves the inclusion.

- Nondeterministic $\check{}$:

[⊇]:

$$\frac{\dfrac{\Delta \Rightarrow \check{} A}{\Delta|_i\Lambda \Rightarrow \hat{}\check{} A}\ \hat{}_i \qquad \dfrac{\dfrac{\overrightarrow{A} \Rightarrow A}{\overrightarrow{\check{} A}|_i\Lambda \Rightarrow A}\ \check{} L}{\overrightarrow{\hat{}_i\check{} A} \Rightarrow A}\ \hat{}_i}{\Delta|_i\Lambda \Rightarrow A \text{ for every i} \in \{1, \cdots, S(A)+1\}}$$

By i.h. for every i, $\Delta|_i\Lambda \in [\![A]\!]$. The inclusion is proved.

[⊆]: by a reasoning similar to the case [⊆] for the connectives $\check{}_i$ we have that:

$$\Delta|_i\Lambda \Rightarrow A$$

Therefore:

$$\text{For every } i \in \{1, \cdots, S(A)+1\}\ \Delta|_i\Lambda \Rightarrow A$$

Hence:

$$\frac{\Delta|_1\Lambda \Rightarrow A \qquad \cdots \qquad \Delta|_{S(A)+1}\Lambda \Rightarrow A}{\Delta \Rightarrow \check{} A}\ \check{} R$$

We have therefore that $\Delta \in \vdash^{-1}(\check{} A)$. We are done.

□

4.2.2 A source of incompleteness: some remarks w.r.t. units and their interpretation in standard semantics

Units may be the source of incompleteness with the more standard semantics, i.e. powerset frames over full displacement algebras. Hence incompleteness may appear in the displacement calculus. We present two examples, one from a subsystem of the displacement calculus, $\mathbf{L} + \{I\}$, i.e. the Lambek Calculus with unit. This calculus is enriched with the usual left/right rules for the unit of sequent calculus. $\mathbf{L} + \{I\}$ corresponds to 0-$\mathbf{D}$ necessarily restricted to the fragment with types ranging over $\mathcal{F}[\bullet, /, \backslash, I]$.

As we said, interpreting the calculus in free monoids we immediately get incompleteness in 0-$\mathbf{D}$:

$$\models I/A, B, A \Rightarrow B \text{ for A,B atomic types } (\star)$$

We have $[\![I/A, B, A]\!] = \begin{cases} \emptyset & \text{if } [\![A]\!] \neq \{\lambda\} \\ [\![B]\!] & \text{if } [\![A]\!] = \{\lambda\} \end{cases}$. Suppose $[\![A]\!] \neq \{\lambda\}$. If $[\![A]\!] = \emptyset$, then the antecedent of $(\star)$ is empty. If $[\![A]\!] \neq \emptyset$ but $(\neq \{\lambda\})$, then $[\![I/A]\!] = \emptyset$ and hence the antecedent of $(\star)$ is again the empty set. Now, if $[\![A]\!] = \{\lambda\}$, then $[\![A]\!] = [\![I/A]\!] = \lambda$, whence the antecedent of $(\star)$ is $[\![B]\!]$. In either case, $[\![I/A, B, A]\!] \subseteq [\![B]\!]$, and hence $\models I/A, B, A \Rightarrow B$. This sequent is however not derivable in $\mathbf{L} + \{I\}$ as a simple inspection in the finite Cut-free search space shows.

In 1-$\mathbf{D}$ we have the following derivation:

$$\cfrac{\cfrac{I \Rightarrow I}{J\{I\} \Rightarrow I}\,JL}{J\{[]\} \Rightarrow I{\uparrow}I}\,{\uparrow}R$$

However the converse hypersequent is underivable in 1-$\mathbf{D}$ but is valid in the standard powerset frame semantics:

$$\models I{\uparrow}I\{[]\} \Rightarrow J$$

For we have:

$$[\![I{\uparrow}I]\!] = \{1\} = [\![J]\!]$$

If we want to maintain units in our syntactic calculus, we should use units in a rather safer way. This is exactly what we do in the next section. The interpretation is nonstandard but almost standard.

4.3 Towards Full Completeness for D

We present a syntactical interpretation which is very close to what we called standard semantics. The idea of this syntactical interpretation originated in a Fadda and Morrill paper (Fadda and Morrill (2005)) in which there was considered the concept of preordered monoid in order to give a sound and complete syntactical interpretation for the Lambek Calculus with brackets **Lb**.

Here we extend the idea to the concept of *preordered displacement algebras* and *preordered nondeterministic displacement algebras*. In this way, we will be able to give for the first time two completeness results for the displacement calculus including all the connectives, i.e., units, deterministic products and all the nondeterministic connectives. As in the standard case, we are given a syntactical algebra $\mathcal{M} = \langle M, +, 0, 1\rangle$, i.e., we preserve freenees. From $\mathcal{M}$ we induce what we call a general displacement algebra $\mathcal{A} = \langle \{L_i\}_{i\in\omega}, +, \{\times_k\}_{k>0}, 0, 1\rangle$:

(198)
$$\forall i \in \omega,\ L_i \subseteq \underbrace{M_0 \cdot 1 \cdot M_0 \cdots M_0 \cdot 1 \cdot M_0}_{i\ 1\text{'s}}$$

M_0 denotes the subset of M with 0 primes, and $L_0 \subseteq M_0$. In general displacement algebras, the inclusions of sort domains in (198) crucially may be proper. As we saw before, units are a source of incompleteness in standard semantics for displacement calculus, and even for the plain Lambek calculus. Moving to general models will be one of the keys to avoid incompleteness. Another difference with standard semantics, which will be very useful, is to incorporate into our displacement algebra $\mathcal{A}$ a preorder $\leq$ compatible with the operations $+$ and $\{\times_{i+1}\}_{i\in\omega}$.

(199) **Definition** (*Preorder Compatibility*)

Let $\mathcal{A} = \langle \{L_i\}_{i\in\omega}, +, \{\times_{k+1}\}_{k\in\omega}, 0, 1\rangle$ be a general displacement algebra. We say that a binary relation $\leq$ on $\mathcal{A}$ is a preorder compatible with the operations iff:

- For all $x, y \in L_i$ and $t, z \in L_j$ with $i, j \in \omega$:

$$\frac{x \leq y \qquad t \leq z}{x + t \leq y + z}\ \mathbf{Comp_1}$$

- For all $x, y \in L_i$ with $i > 0$ and $t, z \in L_j$ with $j \in \omega$:

$$\frac{x \leq y \qquad t \leq z}{x \times_k t \leq y \times_k z}\ \mathbf{Comp_2}$$

The interpretation of types in this new nonstandard semantics will be made in the set of subsets of L_i ($i \in \omega$) which are well-sorted and downward-closed (d.c.):

(200)

$A \subseteq L_i$ with $i \in \omega$ is d.c.
iff
for every x, if $x \leq a$ and $a \in A$ then $x \in A$

The set of d.c. sets of $\mathcal{A}$ is denoted $\bigcup_{i\in\omega} 2^{L_i}_{\leq}$.

Let us now define powerset frame models over general preordered displacement algebras. Let $(\mathcal{A}; \leq)$ be a general preordered displacement algebra and let v be a mapping of atomic types to $\bigcup_{i\in\omega} 2^{L_i}_{\leq}$. The new interpretation of types is as follows:

(201) **Definition** (*Interpretation of Types in Powerset Frame Models over Preordered Displacement Algebras*)

$$
\begin{array}{rcl}
[\![A]\!]_v & \stackrel{def}{=} & v(A) \in 2^{L_{S(A)}}_{\leq} \text{ for A atomic type} \\
[\![I]\!]_v & \stackrel{def}{=} & \{x : x \leq 0\} \\
[\![J]\!]_v & \stackrel{def}{=} & \{y : y \leq 1\} \\
[\![A{\bullet}B]\!]_v & \stackrel{def}{=} & \{c : \exists a \in [\![A]\!]_v \, \exists b \in [\![B]\!]_v,\ c \leq a + b\} \\
[\![B/A]\!]_v & \stackrel{def}{=} & \{t : \forall a \in [\![A]\!]_v,\ t + a \in [\![B]\!]_v\} \\
[\![A\backslash B]\!]_v & \stackrel{def}{=} & \{t : \forall a \in [\![A]\!]_v,\ a + t \in [\![B]\!]_v\} \\
[\![A\odot_i B]\!]_v & \stackrel{def}{=} & \{c : \exists a \in [\![A]\!]_v \, \exists b \in [\![B]\!]_v,\ c \leq a \times_i b\} \text{ with } 1 \leq i \leq S(A) \\
[\![B{\uparrow_i}A]\!] & \stackrel{def}{=} & \{t : \forall a \in [\![A]\!]_v,\ t \times_i a \in [\![B]\!]_v\} \text{ with } 1 \leq i \leq S(B) - S(A) + 1\} \\
[\![A{\downarrow_i}B]\!]_v & \stackrel{def}{=} & \{t : \forall a \in [\![A]\!]_v,\ a \times_i t \in [\![B]\!]_v\} \text{ with } 1 \leq i \leq S(A) \\
[\![\triangleleft^{-1} A]\!] & \stackrel{def}{=} & \{t : \forall x \leq 1, t + x \in [\![A]\!]_v\} \\
[\![\triangleright^{-1} A]\!] & \stackrel{def}{=} & \{t : \forall x \leq 1, x + t \in [\![A]\!]_v\} \\
[\![A \circledcirc B]\!]_v & \stackrel{def}{=} & \{c : \exists i \text{ with } 1 \leq i \leq S(A), \exists a \in [\![A]\!]_v \, \exists b \in [\![B]\!]_v,\ c \leq a \times_i b\} \\
[\![B{\Uparrow}A]\!]_v & \stackrel{def}{=} & \{t : \forall a \in [\![A]\!]_v, \forall i \text{ with } 1 \leq i \leq S(B) - S(A) + 1,\ t \times_i a \in [\![B]\!]_v\} \\
[\![A{\Downarrow}B]\!]_v & \stackrel{def}{=} & \{t : \forall a \in [\![A]\!]_v, \forall i \text{ with } 1 \leq i \leq S(A),\ t \times_i a \in [\![B]\!]_v\} \\
[\![\check{}^{\,i} A]\!]_v & \stackrel{def}{=} & \{t : \forall a \leq 0,\ t \times_i a \in [\![A]\!]_v\} \\
[\![\hat{}^{\,i} A]\!]_v & \stackrel{def}{=} & \{t : \exists a \in [\![A]\!]_v \, \exists \gamma \leq 0, \text{ such that } t \leq a \times_i \gamma\} \\
[\![\check{}A]\!]_v & \stackrel{def}{=} & \{t : \forall a \leq 0, \forall i \text{ with } 1 \leq S(A) + 1,\ t \times_i a \in [\![A]\!]_v\} \\
[\![\hat{}A]\!]_v & \stackrel{def}{=} & \{t : \exists i \text{ with } 1 \leq i \leq S(A), \exists a \in [\![A]\!]_v,\ \exists \gamma \leq 0 \text{ such that } t \leq a \times_i \gamma\}
\end{array}
$$

(202) **Lemma**
Let $\mathcal{M} = (\mathcal{A}, [\![\cdot]\!]_v)$ be a powerset frame model over a general preordered displacement algebra $\mathcal{A}$. We have that the interpretation of every type is d.c.:

$$\text{For every type } A \in \mathcal{F}_i, i \in \omega,\ [\![A]\!] \in 2^{L_i}_{\leq}$$

Proof. By induction on the complexity of types.

- Base cases
  i) Atomic types: d.c. by definition.
  ii) Units: d.c. by definition.
- Inductive step: we show some representative examples.
  i) Products and product-like unary synthetic connectives: d.c. by definition. Let us see them in some detail. In powerset frames over preordered displacement algebras continuous product is defined as follows:

$$[\![A{\bullet}B]\!]_v \stackrel{def}{=} \{x : \exists a \in [\![A]\!]_v,\ \exists b \in [\![B]\!]_v \text{ such that } x \leq a + b\}$$

Suppose $\alpha \leq \gamma$ such that $\gamma \in [\![A{\bullet}B]\!]_v$. We know that that there exist $a \in [\![A]\!]_v$ and $b \in [\![B]\!]_v$ such that $\gamma \leq a + b$. We have by transitivity of $\leq$:

$$\frac{\alpha \leq \gamma \qquad \gamma \leq a + b}{\alpha \leq a + b}\,\textbf{Trans}$$

Hence $\alpha \in [\![A \bullet B]\!]_v$.
Nondeterministic product: suppose that $y \in [\![A \circledcirc B]\!]$ and $x \leq y$. It follows that for some i, $a \in [\![A]\!]$ and $b \in [\![B]\!]$ we have that $y \leq a \times_i b$. By transitivity of $\leq$, $x \leq a \times_i b$. Hence, there exists $i \in \{1, \cdots, S(A)\}$, $a \in [\![A]\!]$ and $b \in [\![B]\!]$ such that $x \leq a \times_i b$, whence $x \in [\![A \circledcirc B]\!]$.
Let us see a product-like synthetic connective $\hat{}^k$:

$$[\![\hat{}^k A]\!]_v \stackrel{def}{=} \{t : \exists a \in [\![A]\!]_v\, \exists \gamma \leq 0, \text{ such that } t \leq a \times_k \gamma\}$$

Following a reasoning similar to the previous one, the nondeterministic (synthetic) unary connective $\hat{}$ is also d.c.

ii) Implicative connectives:
Consider the continuous connective $/$. We want to see that $[\![B/A]\!]$ is d.c. Let $\delta \in [\![B/A]\!]$. Suppose we are given x such that $x \leq \delta$. Let $a \in [\![A]\!]$. We have:

$$\frac{x \leq \delta \qquad a \leq a}{x + a \leq \delta + a}\,\mathbf{Comp_1}$$

$\delta + a \in [\![B]\!]$. By induction hypothesis (i.h.) $[\![B]\!]$ is d.c. Whence $x + a \in [\![B]\!]$. Hence, we have that for every $a \in [\![A]\!]$, $x + a \in [\![B]\!]$. It follows that $x \in [\![B/A]\!]$.
Let us see the case of a discontinuous connective, say $\downarrow_i$ for some $i > 0$. Let $\delta \in [\![B]\!]$. Let a be an arbitrary element of $[\![A]\!]$. We have:

$$\frac{x \leq \delta \qquad a \leq a}{a \times_i x \leq a \times_i \delta}\,\mathbf{Comp_2}$$

$a \times_i \delta \in [\![A]\!]$. By i.h. $[\![A]\!]$ is d.c. Hence by transitivity $a \times_i x \in [\![B]\!]$, whence $x \in [\![A {\downarrow_i} B]\!]$.
Finally, let us see nondeterministic extraction $\Uparrow$. Let $\delta \in [\![B \Uparrow A]\!]$ and x be such that $x \leq \delta$. Let a be an arbitrary element of $[\![A]\!]$. For every $i = 1, \cdots, S(B) - S(A) + 1$, we have that:

$$\frac{x \leq \delta \qquad a \leq a}{x \times_i a \leq \delta \times_i a}\,\mathbf{Comp_2}$$

For every $i = 1, \cdots, S(B) - S(A) + 1$ $\delta \times_i a \in [\![B]\!]$. By i.h. $[\![B]\!]$ for every $i = 1, \cdots, S(B) - S(A) + 1$, $x \times_i a \in [\![B]\!]$, whence $x \in [\![B \Uparrow A]\!]$.

□

The new semantics we have defined is sound, i.e., every derivable hypersequent is valid. Let us call $\mathbb{PDA}$ the class of general preordered displacement algebras. We see the result for the categorical calculus of the full fragment including the synthetic connectives defined in Chapter 3. Since the categorical calculus and the hypersequent calculus are equivalent, the result extends to the hypersequent calculus:

(203) **Theorem** (*Soundness of Full* **DND** *w.r.t. Powerset Frame Models over General PDA's*)

Let $A \to B$ be a provable categorical arrow in **cDND**. It follows that:

(204)
$$\mathbf{PDA} \models A \to B$$

Proof. We proceed by induction on the length of derivations in the categorical calculus **cDND**. The property of compatibility of the operations in a preordered displacement algebra and the fact that type are interpreted in d.c. sets will turn out to be crucial in the proof.

- Axiom:

$$A \to A$$

for any type $A \in \mathcal{F}$. This case is trivial.

- The laws of residuation for the different connectives are sound. Let us see this with an example. The remaining connectives have completely similar reasonings. Consider the nondeterministic residuated triple $(\circledcirc, \Downarrow, \Uparrow)$: Nondeterministic residuation:

$$\begin{array}{lll} A \circledcirc B \to C & \text{iff} & A \to C \Uparrow B \\ & \text{iff} & B \to A \Downarrow C \end{array}$$

Let us see the first *only if*. Suppose we have $[\![A \circledcirc B]\!] \subseteq [\![C]\!]$. Let $a \in [\![A]\!]$. Let $b \in [\![A]\!]$ and i be such that $0 < i \leq S(A)$. Hence by induction hypothesis (i.h.), $a \times_i b \in [\![A \circledcirc B]\!] \subseteq [\![C]\!]$. It follows then that $a \in [\![C \Uparrow B]\!]$.

Let us see the first *if*. By i.h., we have $[\![A]\!] \subseteq [\![C \Uparrow B]\!]$. Let $a \in [\![A]\!]$, $b \in [\![B]\!]$ and i be such that $0 < i \leq S(A)$. Since $a \in [\![C \Uparrow B]\!]$ we have $a \times_i b \in [\![C]\!]$. Let γ be such that $\gamma \leq a \times_i b$. Since the types and arrows are interpreted in sets which are d.c. it follows that $\gamma \in [\![A \circledcirc B]\!] \subseteq [\![C]\!]$. We are done. The second *only if* and *if* is completely similar.

- Structural rules are sound.

  A remark on notation. We adopt the following convention:

  x_A or x_{BC} mean respectively that $x_A \in [\![A]\!]$ and $x_{BC} \in [\![B * C]\!]$ where $*$ is an operation which is clear from the context.

  - From the axioms of **cD**, let us see for example the case of discontinuous associativity. The other axioms have analogous proofs. Consider the following:

$$A \odot_i (B \odot_j C) \leftrightarrow (A \odot_i B) \odot_{i+j-1} C \text{ with } B \between_A C$$

We want to see that:

$$[\![A \odot_i (B \odot_j C)]\!] = [\![(A \odot_i B) \odot_{i+j-1} C]\!] \text{ with } B \between_A C$$

Let us see the case [⊆]. Let $\delta \in [\![A \times_i (B \times_j C)]\!]$. It follows that by our nonstandard definition of products (in particular the discontinuous products) we have the following:

$$\delta \leq x_A \times_i (x_{BC})$$

where x_A is an arbitrary element of $[\![A]\!]$ and x_{BC} is such that $x_{BC} \leq x_B \times_j x_C$ for arbitrary elements x_B and x_C. In the underlying syntactical algebra $\langle M, +, 0, 1\rangle$ the following holds:

$$x_A \times_i (x_B \times_j x_C) = (x_A \times_i x_B) \times_{i+j-1} x_C$$

Since $\delta \leq x_A \times_i x_{BC} \leq x_A \times_i (x_B \times_j x_C)$, it follows that:

$$\delta \leq (x_A \times_i x_B) \times_{i+j-1} x_C$$

whence $\delta \in [\![(A \odot_i B) \odot_{i+j-1} C]\!]$, because as we have proved, for any type D, $[\![D]\!]$ is d.c., in particular $[\![(A \odot_i B) \odot_{i+j-1} C]\!]$. We are done.

The other inclusion [⊇] is completely similar to the case of [⊆].

- Nondeterministic rule:

We consider the two nondeterministic rules $ND1$ and $ND2$:

$$\frac{A \odot_1 B \to C \qquad \cdots \qquad A \odot_a B \to C}{A \circledcirc B \to C} ND1$$

$$\frac{A \circledcirc B \to C}{A \odot_i B \to C \text{ for } i = 1, \cdots, a} ND2$$

Let us prove $ND1$. Let γ be such that there exist $i \in \{1, \cdots, a\}$, $x_A \in [\![A]\!]$, $x_B \in [\![B]\!]$ and $\gamma \leq a \times_i b$. The premises of $ND1$ give us the arrow $A \odot_i B \to C$. It is clear that $\gamma \in [\![A \odot_i B]\!]$. By assumption, $[\![A \odot_i B]\!] \subseteq [\![C]\!]$. Hence $\gamma \in [\![C]\!]$. Consider now rule $ND2$. Suppose we have γ and i, x_A, x_B such that $\gamma \leq x_A \times_i x_B$. By assumption we have that $[\![A \circledcirc B]\!] \subseteq [\![C]\!]$. By definition of the syntactical interpretation of $A \circledcirc B$, it follows that $\gamma \in [\![A \circledcirc B]\!]$. We have then that $\gamma \in [\![C]\!]$, whence $[\![A \odot_i B]\!] \subseteq C$.

- Units: we have that the following rules for units are sound:

$$I \bullet A \leftrightarrow A \leftrightarrow A \bullet I$$
$$J \odot_1 A \leftrightarrow A \leftrightarrow A \odot_i J \text{ for } 1 \leq i \leq S(A)$$

Let us see the case of discontinous units. We show an example:

$$[\![J \odot_1 A]\!] = [\![A]\!] = [\![A \odot_i J]\!] \text{ for } 1 \leq i \leq S(A)$$

Consider the second equality. Let $\delta \in [\![A]\!]$. Let x_J be an arbitrary element of $[\![J]\!]$, i.e., an element x such that $x \leq 1$. Since we are interpreting types in a preordered displacement algebra, we have that:

$$\delta = \delta \times_i 1 \in [\![A]\!]$$

Since $\delta = \delta \times_i 1$, $\delta \leq \delta \times_i 1 \in [\![A \odot_i J]\!]$. Hence we have seen the inclusion $[\![A]\!] \subseteq [\![A \odot_i J]\!]$. Let us see the other inclusion, i.e., $[\![A \odot_i J]\!] \subseteq [\![A]\!]$. Consider an arbitrary element x_{AJ} such that $x_{AJ} \in [\![A \odot_i J]\!]$. It follows that:

$$x_{AJ} \leq x_A \times_i x_j \text{ with } x_j \leq 1 \text{ and } x_A \times_i x_j \leq x_A \times_i 1 \leq x_A$$

It follows that $x_{AJ} \leq x_A$ with $x_A \in [\![A]\!]$. Since $[\![A]\!]$ is d.c., we have that $x_{AJ} \in [\![A]\!]$, whence the inclusion $[\![A \odot_i J]\!] \subseteq [\![A]\!]$ holds.

The first equality $[\![J \odot_1 A]\!] = [\![A]\!]$ follows a similar reasoning in which the fact that $x_A = x_A \times_i 1$ holds in preordered displacement algebra.

Finally, the proof of soundness in the case of the continuous unit is completely similar to the discontinous one where in the reasonings $\times_i$ and 1 must be replaced respectively by $+$ and 0.

- The Cut rule for **cDND**:

$$\frac{A \to B \qquad B \to C}{A \to C}\, Cut$$

This case holds trivially by the transitivity of the inclusion.

This completes the proof.

□

In the following subsection we see how this new nonstandard semantics works falsifying the hypersequents which are underivable but valid w.r.t. the standard semantics. After this, we will see two results of completeness: one for the full fragment of **D** without the nondetermistic connectives and the second one for the full fragment including the nondeterministics rules. In the second completeness result, we will slightly modify the definition of preordered displacement algebra and the interpretation of types.

4.3.1 The new semantics at work: falsifying some underivable hypersequents

(205) **Lemma**

There exist powerset frame models over full preordered displacement algebras which falsify the following hypersequents:

i) $I/A, B, A \Rightarrow B$

ii) $I{\uparrow}I\{[]\} \Rightarrow J$

Proof.

i) $I/A, B, A \Rightarrow B$:

Ler us build a powerset model over a preordered displacement algebra (PDA) which falsifies the above sequent. Let $(M, +, 0, 1)$ be the syntactical algebra freely generated by the set $\Sigma = \{1, a, b\}$ and consider its associated standard displacement algebra. We will build a preorder based on a mapping on strings of M. Let w be a string of M. Let us consider the following *factorization of w around the a's* :

$$w = \alpha_0 + a^{i_1} + \alpha_1 + \cdots + a^{i_{n-1}} + \alpha_{n-1} + a^{i_n} + \alpha_n$$

Where the α_i's ($i = 1, \cdots, n-1$ if $n > 0$) are nonempty strings without any occurrence of a. α_0 and α_n are possibly empty strings with no occurrence of a. If $n = 0$ then we put $w = \alpha_0$. In sigma notation we would have:

$$w = \alpha_0 + \sum_{k=1}^{n}(a^{i_k} + \alpha_k)$$

With this factorization around the a's we define a mapping, which we call the *reducing mapping*, which erases every occurrence of a^3 of a given string. Before defining formally this mapping, which we denotate as $[\cdot]$, let us see an example:

$$[b + a^2 + b^2 + a^5 + b^2 + a^9 + b + a^7] = b + a^2 + b^2 + a^2 + b^3 + a$$

More formally:

$$\begin{array}{lcl} [\cdot] : M & \longrightarrow & M \\ w = \alpha_0 + \sum_{k=1}^{n}(a^{i_k} + \alpha_k) & \mapsto & [w] = \alpha_0 + \sum_{k=1}^{n}(a^{i_k \,\mathbf{mod}\, 3} + \alpha_k) \end{array}$$

Where $i\,\mathbf{mod}\,3$ with $i \in \omega$ is the standard arithmetical operation i *modulo* 3.

Using the mapping $[\cdot]$ allows us to define the following relation on M:

$$\boxed{\text{For every } r, s, \in M\ r \leq s \text{ iff } [r] = [s]}$$

We have to check that $\leq$ is a preorder compatible with the operations:

– **Preorder**:
Reflexifity of $\leq$: Given $m \in M$, we have by the functionality of $[\cdot]$ that $[m] = [m]$ whence $m \leq m$. Transitivity comes from the transitivity of the equality:

$$\frac{r \leq s \qquad s \leq t}{r \leq t}\,Trans$$

The premises of $Trans$ can be expressed in terms of $[\cdot]$:

$$[r] = [s] \text{ and } [s] = [t]$$

Therefore $[r] = [t]$ which is equivalent by definition to $r \preceq t$. A remark: in fact $\preceq$ is an equivalence relation because symmetry holds:

$$m \preceq n \text{ iff } [m] = [n] \text{ iff } n \preceq m$$

– **Compatibility**: We have the following fact:
Given $m, n \in M$ we have that:
(206)
$$[[m] + [n]] = [m + n]$$
And in general, we have that:
(207)
$$[[m_1] + \cdots + [m_n]] = [m_1 + \cdots + m_n]$$
(206) and (207) are due to the following fact of elementary arithmetic:[5]

$$(a + b) \,\mathbf{mod}\, p = (a \,\mathbf{mod}\, p + b \,\mathbf{mod}\, p) \,\mathbf{mod}\, p$$

And in general, we have:

$$(a_1 + \cdots + a_n) \,\mathbf{mod}\, p = (a_1 \,\mathbf{mod}\, p + \cdots + a_n \,\mathbf{mod}\, p) \,\mathbf{mod}\, p$$

Where a, b and $a_i \in \omega$ for $i = 1, \cdots, n$ and p denotes an arbitrary natural number. Let us prove now that $\preceq$ is compatible with $+$. We want to see that:

$$\frac{x \preceq y \qquad t \preceq z}{x + t \preceq y + z}$$

From $x \preceq y$ and $t \preceq z$ we have that:

$$[x] = [y] \text{ and } [t] = [z]$$

It follows that:

$$[x] + [t] = [y] + [z]$$

Applying $[\cdot]$ to the last equality we obtain:

$$[[x] + [t]] = [[y] + [z]]$$

By (206) the following holds:

$$[x + t] = [[x] + [t]] = [[y] + [z]] = [y + z]$$

Hence $[x + t] = [y + z]$ whence $x + t \preceq y + z$.

[5]The proofs of these arithmetical facts are from to the so-called theorem of arithmetical division: for every $a, b \in \omega$ there exist $q, r \in \omega$ such that:

$$a = b \cdot q + r \text{ such that } 0 \leq r < b$$

Moreover one can easily prove that q and r are unique.

Let us see now the compatibility of $\leq$ with the $\times_i$ $(i > 0)$ operations:

$$\frac{x \leq y \qquad t \leq z}{x \times_i t \leq y \times_i z}$$

Let us split x and y around the i-th occurrence of 1:

$$x = x_1 + 1 + x_2 \text{ and } y = y_1 + 1 + y_2$$

It is readily seen that:

$$[x_1 + 1 + x_2] = [x_1] + 1 + [x_2]$$
$$[y_1 + 1 + y_2] = [y_1] + 1 + [y_2]$$

From $x \leq y$ we have that:

$$[x_1] + 1 + [x_2] = [x_1 + 1 + x_2] = [y_1 + 1 + y_2] = [y_1] + 1 + [y_2]$$

From $t \leq z$ we have that $[t] = [z]$. Hence:

$$[x_1] + [t] + [x_2] = [y_1] + [z] + [y_2]$$

Applying $[\cdot]$ to the last equation:

$$[[x_1] + [t] + [x_2]] = [[y_1] + [z] + [y_2]]$$

It follows that:

$$[x_1 + t + x_2] = [y_1 + z + y_2]$$

From the last equation:

$$[x \times_i t] = [x_1 + t + x_2] = [y_1 + z + y_2] = [y \times_i z]$$

Hence:

$$[x \times_i t] \leq [y \times_i z]$$

We have seen therefore that $\leq$ is a preorder compatible with the operations of the displacement algebra. Notice that in fact what we have proved is that $\leq$ is a congruence, for we have seen that $\leq$ is symmetric whence it is an equivalence relation. Let us define the following powerset frame model over our preordered displacement algebra:

$$\begin{array}{rcl} v(A) & = & \{x : x \leq a\} \\ v(B) & = & \{x : x \leq b\} \end{array}$$

By definition of a preordered displacement model:

$$\begin{array}{rcl} [\![I]\!] & = & \{x : x \leq 0\} \\ [\![J]\!] & = & \{y : y \leq 1\} \end{array}$$

$$[\![I/A]\!]_v \quad = \quad \{x : \forall a \in [\![A]\!]\, x + a \leq 0\}$$

The last interpretation of the type I/A can be rewritten as:

$$[\![I/A]\!]_v \quad = \quad \{x : \forall a \in [\![A]\!],\, [x + a] = [0]\}$$

Now, $a^2 \in [\![I/A]\!]_v$ for $[a^2 + a] = [a^3] = [0]$. We have that $a \in [\![A]\!]_v$ and $b \in [\![B]\!]_v$. It follows that:

$$a^2 + b + a \in [\![I/A, B, A]\!]$$

But $a^2 + b + a \not\leq b$ for $[a^2 + b + a] = a^2 + b + a \neq b = [b]$, whence $a^2 + b + a \notin [\![B]\!]_v$.

It follows that:

$$\not\models I/A, B, A \Rightarrow B$$

ii) Let us falsify the hypersequent:

$$I{\uparrow}I\{[]\} \Rightarrow J$$

With the techniques from i) we define a preordered displacement algebra freely generated by $\{a, 1\}$. We build a similar preorder to the one from i), which is compatible with the operations as follows:

$$m \leq n \text{ iff } [m] = [n]$$

Where the function $[\cdot]$ is defined similarly to the one from i) with the following difference:

$$[a^2] = 0$$

The interpretation of units is of course

$$\begin{array}{lcl} [\![I]\!] & = & \{x : x \leq 0\} \\ [\![J]\!] & = & \{y : y \leq 1\} \end{array}$$

Let us see the interpretation of $[\![I{\uparrow}I]\!]$

$$[\![I{\uparrow}I]\!] \quad = \quad \{x_1 + 1 + x_2 : \forall y \in [\![I]\!],\ x_1 + y + x_2 \in [\![I]\!]\}$$

We consider therefore the set:

$$\{x_1 + 1 + x_2 : \forall y \leq 0,\ x_1 + y + x_2 \leq 0\}$$

We observe that $a + 1 + a \in [\![I{\uparrow}I]\!]$ for:

$$(a + 1 + a) \times_1 0 = a^2 \leq 0$$

$a^2 \leq 0$ because $[a^2] = 0$. Now we have that:

$$a + 1 + a \not\leq 1$$

Because $a + 1 + a = [a + 1 + a] \neq 1$. Hence:

$$\not\models I{\uparrow}I\{[]\} \Rightarrow J$$

□

4.3.2 Completeness I

In this subsection and the following ones we prove three completeness results for **D**. The first one covers the implicative fragment and the deterministic products and units. In this completeness result there is room for what we have called implicative synthetic connectives including the nondetermistic ones, but nondeterministic products are not included.

The completeness result is very close to the one of the implicative fragment. We are given as in the implicative fragment a syntactical algebra freely generated by a finite set of generators we call Σ_D:

$$\Sigma_D = \{p, q, r, 0, \$, \backslash, /, \bullet, \downarrow, \uparrow, \odot, \Uparrow, \Downarrow, (,), [,], \check{}\ \}$$

The coding map ρ defined in Section 4.2.1 must be extended to the following cases (the cases already covered in section 4.2.1 are omitted):

$$\begin{array}{llll} \rho: & \mathcal{F} & \longrightarrow & \Sigma_D^* \\ & I & \mapsto & \langle q \rangle \\ & J & \mapsto & \langle r \rangle \\ & A \bullet B & \mapsto & \langle \rho(A) \bullet \rho(B) \rangle \\ & A \odot_i B & \mapsto & \langle \rho(A) \langle \odot\, 0^i \rangle \rho(B) \rangle \end{array}$$

The ρ map is extended as in section 4.2.1 to the sets $\widetilde{\mathcal{SF}}$ and $\widetilde{\mathcal{O}}$. We can then define in a similar way the canonical displacement algebra, where it must be understood that now all the deterministic and implicative nondeterministic connectives are covered:

(208)

$$\mathcal{A} = \langle \{\widetilde{\mathcal{O}}_i\}_{i \in \omega}, (\cdot, \cdot), \{|_i\}_{i>0}, \Lambda, \$ \rangle$$

We define now the relation $\leq_\vdash$ on $\mathcal{A}$ as follows:

$$\leq_\vdash \stackrel{def}{=} \{(\gamma, \delta) : \gamma, \delta \in \widetilde{\mathcal{O}} \text{ and } \mathbf{hD} \vdash \rho^{-1}(\gamma) \Rightarrow (\rho^{-1}(\delta))^\bullet\}$$

Where the map $(\cdot)^\bullet$ was defined in Chapter 3.

(209) **Lemma**

$\leq_\vdash$ is a prorder compatible with the operations of the canonical algebra $\mathcal{A}$.

Proof.

- $\leq_\vdash$ is preorder:

  In Chapter 3 we saw that for every $\Delta \in \mathcal{O}$ and any type A the following two results hold:

  i) $\mathbf{hD} \vdash \Delta \Rightarrow \Delta^\bullet$
  ii) If $\mathbf{hD} \vdash \Delta \Rightarrow A$ then $\mathbf{hD} \vdash \Delta^\bullet \Rightarrow A$

  Consider three hyperconfigurations $\Delta = \rho^{-1}(\delta), \Gamma = \rho^{-1}(\gamma)$ and $\Theta = \rho^{-1}(\tau)$ where δ, γ and τ are arbitrary elements of $\widetilde{\mathcal{O}}$. It follows that the following properties hold:

  – $\leq_\vdash$ is reflexive. For $\delta \leq_\vdash \delta$ because $\Delta \Rightarrow \Delta^\bullet$.
  – $\leq_\vdash$ is transitive:

  $$\frac{\delta \leq_\vdash \gamma \qquad \gamma \leq_\vdash \tau}{\delta \leq_\vdash \tau}$$

  For we have:

  $$\dfrac{\Delta \Rightarrow \Gamma^\bullet \qquad \dfrac{\dfrac{\Gamma \Rightarrow \Theta^\bullet}{\vdots}}{\overrightarrow{\Gamma^\bullet} \Rightarrow \Theta^\bullet}}{\Delta \Rightarrow \Theta^\bullet}\, Cut$$

- $\leq_\vdash$ is compatible with the operations: Consider now four arbitrary hyperconfigurations $\Delta = \rho^{-1}(\delta), \Gamma = \rho^{-1}(\gamma)$, $\Theta = \rho^{-1}(\tau)$ and $\Sigma = \rho^{-1}(\sigma)$. There are two cases to consider:

  – Compatibility with concatenation $(\cdot, \cdot)$:

  $$\frac{\delta \leq_\vdash \gamma \qquad \tau \leq_\vdash \sigma}{\delta, \tau \leq_\vdash \gamma, \sigma}$$

  We have to prove:

  $$\Delta, \Theta \Rightarrow (\Gamma, \Sigma)^\bullet$$

  We can derive the following:

  (210)
  $$\frac{\Delta \Rightarrow \Gamma^\bullet \qquad \Theta \Rightarrow \Sigma^\bullet}{\Delta, \Theta \Rightarrow \Gamma^\bullet \bullet \Sigma^\bullet}$$

  Suppose we have $\overrightarrow{\Gamma^\bullet \bullet \Sigma^\bullet} \Rightarrow (\Gamma, \Sigma)^\bullet$. From this and (210) and by applying Cut we obtain:

  $$\Delta, \Theta \Rightarrow (\Gamma, \Sigma)^\bullet$$

  Let us prove now that $\overrightarrow{\Gamma^\bullet \bullet \Sigma^\bullet} \Rightarrow (\Gamma, \Sigma)^\bullet$ is derivable:

$$\dfrac{\dfrac{\dfrac{\Gamma, \Sigma \Rightarrow (\Gamma, \Sigma)^\bullet}{\vdots}}{\Gamma^\bullet, \Sigma^\bullet \Rightarrow (\Gamma, \Sigma)^\bullet}}{\overrightarrow{\Gamma^\bullet \bullet \Sigma^\bullet} \Rightarrow (\Gamma, \Sigma)^\bullet}\ \bullet L$$

– Compatibility with wrapping:

$$\dfrac{\delta \leq_\vdash \gamma \qquad \tau \leq_\vdash \sigma}{\delta|_i\tau \leq_\vdash \gamma|_i\sigma}$$

We have to prove:

$$\Delta|_i\Theta \Rightarrow (\Gamma|_i\Sigma)^\bullet$$

We can derive the following:

$$\dfrac{\dfrac{\dfrac{\Gamma|_i\Sigma \Rightarrow (\Gamma|_i\Sigma)^\bullet}{\vdots}}{\Gamma|_i\overrightarrow{\Sigma^\bullet} \Rightarrow (\Gamma|_i\Sigma)^\bullet}}{\Gamma \Rightarrow (\Gamma|_i\Sigma)^\bullet\uparrow_i\Sigma^\bullet}\ \uparrow_i R$$

From the last hypersequent of the above derivation we derive:

$$\dfrac{\dfrac{\Gamma \Rightarrow (\Gamma|_i\Sigma)^\bullet\uparrow_i\Sigma^\bullet}{\vdots}}{\overrightarrow{\Gamma^\bullet} \Rightarrow (\Gamma|_i\Sigma)^\bullet\uparrow_i\Sigma^\bullet}$$

We have therefore:

$$\dfrac{\overrightarrow{\Gamma^\bullet} \Rightarrow (\Gamma|_i\Sigma)^\bullet\uparrow_i\Sigma^\bullet \qquad \overrightarrow{\Sigma^\bullet} \Rightarrow \Sigma^\bullet}{\overrightarrow{\Gamma^\bullet}|_i\overrightarrow{\Sigma^\bullet} \Rightarrow ((\Gamma|_i\Sigma)^\bullet\uparrow_i\Sigma^\bullet) \odot_i \Sigma^\bullet}\ \odot_i R$$

Since:

$$((\Gamma|_i\Sigma)^\bullet\uparrow_i\Sigma^\bullet) \odot_i \Sigma^\bullet \Rightarrow (\Gamma|_i\Sigma)^\bullet$$

Applying Cut we have:

(211)

$$\dfrac{\overrightarrow{\Gamma^\bullet}|_i\overrightarrow{\Sigma^\bullet} \Rightarrow ((\Gamma|_i\Sigma)^\bullet\uparrow_i\Sigma^\bullet) \odot_i \Sigma^\bullet \qquad \overrightarrow{((\Gamma|_i\Sigma)^\bullet\uparrow_i\Sigma^\bullet) \odot_i \Sigma^\bullet} \Rightarrow (\Gamma|_i\Sigma)^\bullet}{\overrightarrow{\Gamma^\bullet}|_i\overrightarrow{\Sigma^\bullet} \Rightarrow (\Gamma|_i\Sigma)^\bullet}$$

It follows by the application of the left rule of $\odot_i$:

$$\dfrac{\overrightarrow{\Gamma^\bullet}|_i\overrightarrow{\Sigma^\bullet} \Rightarrow (\Gamma|_i\Sigma)^\bullet}{\overrightarrow{\Gamma^\bullet \odot_i \Sigma^\bullet} \Rightarrow (\Gamma|_i\Sigma)^\bullet}\ \odot_i L$$

We have:

$$\frac{\Delta \Rightarrow \Gamma^{\bullet} \qquad \Theta \Rightarrow \Sigma^{\bullet}}{\Delta|_i\Gamma \Rightarrow \Gamma^{\bullet} \odot_i \Sigma^{\bullet}} \odot_i R$$

From the end hypersequent of (211) we have by application of Cut

$$\frac{\Delta|_i\Gamma \Rightarrow \Gamma^{\bullet} \odot_i \Sigma^{\bullet} \qquad \overrightarrow{\Gamma^{\bullet} \odot_i \Sigma^{\bullet}} \Rightarrow (\Gamma|_i\Sigma)^{\bullet}}{\Delta|_i\Gamma \Rightarrow (\Gamma|_i\Sigma)^{\bullet}} Cut$$

We are done.

□

In section 4.2 we introduced the $\vdash^{-1}$ operator. We slightly modify its definition:

$$\begin{array}{llll} \vdash^{-1}: & \mathcal{F} & \longrightarrow & \bigcup_{i\in\omega} 2^{\widetilde{\mathcal{O}}_i} \\ & A & \mapsto & \vdash^{-1}(A) \stackrel{def}{=} \{\delta : \delta \in \widetilde{\mathcal{O}} \text{ and } \vdash_{\mathbf{hD}} \rho^{-1}(\delta) \Rightarrow A\} \end{array}$$

(212) **Lemma** (*Truth Lemma for Deterministic* **D** *and Nondeterministic Implicative* **D**)

For every type $A \in \mathcal{F}[\bullet, \backslash, /, \{\odot_{i+1}\}_{i\in\omega}, \{\uparrow_i\}_{i>0}, \{\downarrow_i\}_{i>0}, \Downarrow, \Uparrow, \{\check{}^{\,i}{}_{i+1}\}_{i\in\omega}, \check{}\,, I, J]$ we have:

$$[\![A]\!]_v = \vdash^{-1}(A)$$

Where $[\![\cdot]\!]_v$ is the valuation in the canonical model we have been considering through the different proofs of the completeness theorems.

Proof. We already covered the proofs for the so-called implicative fragment of **D**. The proofs work also with the new syntactical interpretation. Let us see the result for the product connectives $\bullet$ and $\{\odot_{i+1}\}_{i\in\omega}$ and for units. We proceed by induction on the structures of types:

- Units. By definition of models based on preordered displacement algebras, we have to define the interpretations of units in terms of the preorder relation:

$$\begin{array}{lll} [\![I]\!] & = & \{\gamma : \gamma \leq 0\} \\ [\![J]\!] & = & \{\gamma : \gamma \leq 1\} \end{array}$$

In the canonical model we have:

$$(213) \quad \begin{array}{lll} [\![I]\!] & = & \{\gamma : \gamma \in \widetilde{\mathcal{O}} \text{ and } \vdash_{\mathbf{hD}} \rho^{-1}(\gamma) \Rightarrow I\} \\ [\![J]\!] & = & \{\gamma : \gamma \in \widetilde{\mathcal{O}} \text{ and } \vdash_{\mathbf{hD}} \overrightarrow{\rho^{-1}(\gamma)} \Rightarrow J\} \end{array}$$

It is clear by the definition given in (213) that $[\![I]\!] = \vdash^{-1}(I)$ and $[\![J]\!] = \vdash^{-1}(J)$.

- Continuous product. We want to prove that:

$$[\![A \bullet B]\!]_v = \vdash^{-1} (A \bullet B)$$

The proof follows the one of Fadda and Morrill (2005), in which a preordered monoidal interpretation was given to the (continuous) Lambek calculus with brackets.[6]

- Discontinuous products. We want to prove for every $i > 0$ that:

$$[\![A \odot_i B]\!]_v = \vdash^{-1} (A \odot_i B)$$

  - [$\subseteq$]: Let $\delta \in [\![A \odot_i B]\!]_v$. There exist $\delta_A \in [\![A]\!]_v$ and $\delta_B \in [\![B]\!]_v$ such that:

    (214) $\delta \leq_{\vdash} \delta_A|_i\delta_B$

    Let $\Delta_A := \rho^{-1}(\delta_A)$, $\Delta_B := \rho^{-1}(\delta_B)$ and $\Delta := \rho^{-1}(\Delta)$. From (214) and the fact that $\rho^{-1}(\delta_A|_i\delta_B) = \rho^{-1}(\delta_A)|_i\rho^{-1}(\delta_B) = \Delta_A|_i\Delta_B$, it follows that:

$$\Delta \Rightarrow (\Delta_A|_i\Delta_B)^{\bullet}$$

    By induction hypothesis (i.h.), we have that $[\![A]\!]_v = \vdash^{-1} (A)$ and $[\![B]\!]_v = \vdash^{-1} (B)$. Hence:

$$\begin{array}{l} \Delta_A \Rightarrow A \\ \Delta_B \Rightarrow B \end{array}$$

    It follows that

$$\frac{\Delta_A \Rightarrow A \qquad \Delta_B \Rightarrow B}{\Delta_A|_i\Delta_B \Rightarrow A \odot_i B} \odot_i R$$

    We derive now the following

$$\frac{\dfrac{\Delta_A|_i\Delta_B \Rightarrow A \odot_i B}{\vdots}}{\overrightarrow{(\Delta_A|_i\Delta_B)^{\bullet}} \Rightarrow A \odot_i B}$$

    We have then:

$$\frac{\Delta \Rightarrow (\Delta_A|_i\Delta_B)^{\bullet} \qquad \overrightarrow{(\Delta_A|_i\Delta_B)^{\bullet}} \Rightarrow A \odot_i B}{\Delta \Rightarrow A \odot_i B} Cut$$

    Hence, $\delta = \rho^{-1}(\Delta) \in \vdash^{-1} (A \odot_i B)$.

[6] In (*op. cit*) the problem of discontinuity was not addressed.

– [$\supseteq$]: Let $\delta \in \vdash^{-1} (A \odot_i B)$. We have therefore ($\Delta := \rho^{-1}(\delta)$):

$$\Delta \Rightarrow A \odot_i B$$

By i.h., $[\![A]\!]_v = \vdash^{-1} (A)$ and $[\![B]\!]_v = \vdash^{-1} (B)$. Hence:

$$\overrightarrow{A} \Rightarrow A \text{ and } \overrightarrow{B} \Rightarrow B$$

We have that: $\delta \leq_\vdash \rho^{-1}(\overrightarrow{A})|_i \rho^{-1}(\overrightarrow{B})$ which is equivalent to:

$$\Delta \Rightarrow (\overrightarrow{A}|_i \overrightarrow{B})^\bullet$$

I.e.:

$$\Delta \Rightarrow A \odot_i B$$

Hence, $\delta \in [\![A \odot_i B]\!]_v$. We are done.

□

4.3.3 Completeness II: Accomodating the Nondeterministic Connectives

In this subsection we have to extend the notion of syntactical algebra with a new operation we call nondeterministic product, i.e., we have $\mathcal{M} = \langle M, +, \times, 0, 1\rangle$. As in the case of the syntactical interpretations which we have studied so far, we preserve the freeness of the monoid $(M, +, 0, 1)$. This new extended syntactical algebra will be called a nondeterministic syntactical algebra. From $\mathcal{M}$ we induce a nondeterministic general displacement algebra. The sort map of an element of a nondeterministic syntactical algebra must be extended to the case of the new operation $\times$:

$$\begin{array}{lll} S(0) & = & 0 \\ S(1) & = & 1 \\ S(a+b) & = & S(a) + S(b) \\ S(a \times b) & = & S(a) + S(b) - 1 \end{array}$$

We define the following ω-indexed family of subsets of $\mathcal{M}$:

$$M_i = \{a : a \in M \text{ such that } S(i) = i\}$$

The induced nondeterministic general displacement algebra has as in the case of (deterministic) general displacement algebras the following feature characterizing the sort domains:

(215)

$$\text{For every } i \geq 0,\ L_i \subseteq M_i$$

As we already saw in the result of completeness for full deterministic **D**, it is crucial that the inclusion (215) may be proper. We have then the nondeterministic general displacement algebra:

$$\mathcal{A} = \langle \{L_i\}_{i\in\omega}, +, \{\times_i\}_{i>0}, \times, 0, 1\rangle$$

Notice that the operation $\times$ is sort polymorphic of sort functionality $(i+1, j) \to i+j$ (with $i, j > 0$).

As in the result of completeness I, we define a preorder $\leq$ in the algebra which has to be compatible with the operations of the algebra. In completeness I we required the compatiblity of $\leq$ with the operations $+$, $\{\times\}_{i>0}$. Now, we require in addition compatibility with the operation $\times$, i.e.:

$$\frac{a \leq b \qquad c \leq d}{a \times c \leq b \times d}\,\mathbf{Comp_3}$$

The preordered algebra $(\mathcal{A}; \leq)$ must satisfy the following rules:

(216)
$$\frac{a \times_1 b \leq c \qquad \cdots \qquad a \times_{S(a)} b \leq c}{a \times b \leq c}\,(nd_1) \quad \frac{a \times b \leq c}{a \times_i b \leq c}\,(nd_2)$$

4.3.4 A Modification of the Nonstandard Syntactical Interpretation

Accomodating the deterministic product-like connectives has been done in the class of general preordered displacement algebras. The syntactical interpretation of types $[\![\cdot]\!]_v$ has been carried out in the set of well-sorted and downward-closed (d.c.) subsets of a given general preordered displacement algebra. We have now added to the general preordered displacement algebras the axioms connecting the operations $\times_i$ $(i > 0)$ and the nondeterministic operation $\times$. Here syntactical interpretation of types $[\![\cdot]\!]_v$ will be slightly modified. Interpretation in a nondeterministic general preordered displacement algebra (we refer to this class $\mathbb{NDA}$) $(\mathcal{A}; \leq)$ will be done in d.c. subsets B with a top element. More concretely, every type $B \in \mathcal{F}_{\mathbf{DND}}$ is such that in a powerset frame model over a **NDA** algebra:

- $[\![B]\!]_v$ is d.c., i.e.:

$$\text{If } \delta \leq a \text{ and } a \in [\![B]\!]_v \text{ then } \delta \in [\![B]\!]_v \textit{ (DC)}$$

- $[\![B]\!]_v$ has a top element $\top_{[\![B]\!]_v}$, i.e.:

$$\text{For every } a \in [\![B]\!]_v, \exists\, \top_{[\![B]\!]_v} \in [\![B]\!]_v \text{ such that } a \leq \top_{[\![B]\!]_v}\ (TE)$$

Powerset frame models over $\mathbb{NDA}$ algebras adhering to *(DC)* and *(TE)* will be called powerset frame models over nondeterministic displacement algebras, in notation, $\mathbb{PNDA}$. Valididy of a categorical arrow $A \to B$ or a hypersequent $\Delta \Rightarrow A$ will be denoted $\mathbb{PNDA} \models A \to B$ or $\mathbb{PNDA} \models \Delta \Rightarrow A$. We note that downward-closedness of interpreted types (DC) is easily provable for every nondeterministic displacement algebra. It is the condition *(TE)* which has to be imposed. If one shows downward-closedness by induction on the structure of types the case of the nondeterministic product $\circledcirc$ is trivial:

(217) $[\![A \circledcirc B]\!]_v \overset{def}{=} \{c :\ \text{there exist } a \in [\![A]\!]_v, b \in [\![B]\!] \text{ such that } c \leq a \times b\}$

(218) **Theorem** (*Soundness of Full* **D** *w.r.t. Nondeterministic Powerset Frame Models*)

Let $A \to B$ be a provable categorical arrow in **cD**. It follows that:

$$\mathbb{PNDA} \models A \to B$$

Proof. The soundness of the deterministic structural rules and residuation contains no surprises. The proof given in the deterministic case goes through. We have to check then the soundness of the nondeterministic rule which we recall now:

$$\frac{A \odot_1 B \to C \qquad \dots \qquad A \odot_{S(A)} B \to C}{A \circledcirc B \to C}\ \mathbf{ND_1}$$

$$\frac{A \circledcirc B \to C}{A \odot_i B \to C}\ \mathbf{ND_2} \text{ with } 1 \leq i \leq S(A)$$

Let $\langle \mathcal{A}; \leq \rangle$ be an $\mathbb{NDA}$ algebra. Let us see the soundness of the rule $\mathbf{ND_1}$. Suppose $\gamma \in [\![A \circledcirc B]\!]_v$. This means that $\gamma \leq a \times b$ for $a \in [\![A]\!]_v$ and $b \in [\![A]\!]_v$. We want to see that $\gamma \in [\![C]\!]_v$. By hypothesis $[\![C]\!]_v$ has a top element. Let us denote it $\top_{[\![C]\!]_v}$.
Now, let $a \in [\![A]\!]_v$ and $b \in [\![B]\!]_v$. Since by hypothesis we have for every $i = 1, \cdots, S(A)$ that $[\![A \odot_i B]\!]_v \subseteq [\![C]\!]_v$ it follows that

$$\begin{aligned} &a \times_1 b \subseteq [\![C]\!]_v \\ &\vdots \\ &a \times_{S(A)} b \subseteq [\![C]\!]_v \end{aligned}$$

$[\![C]\!]_v$ has a top element $\top_{[\![C]\!]_v}$ (condition (TE) of interpreted $\mathbb{NDA}$ algebras). Therefore:

$$\begin{aligned} &a \times_1 b \leq \top_{[\![C]\!]_v} \\ &\vdots \\ &a \times_{S(A)} b \leq \top_{[\![C]\!]_v} \end{aligned}$$

By the property (nd_1) of $\mathbb{NDA}$ algebras, we have that:

$$\frac{a \times_1 b \leq \top_{[\![C]\!]_v} \qquad \cdots \qquad a \times_1 b \leq \top_{[\![C]\!]_v}}{a \times b \leq \top_{[\![C]\!]_v}}\ (nd_1)$$

Since we have:

$$\gamma \leq a \times b \leq \top_{[\![C]\!]_v} \in [\![C]\!]_v$$

and $[\![C]\!]_v$ is d.c., it follows that:

$$\gamma \in [\![C]\!]_v$$

This proves the nondeterministic rule $\mathbf{ND_1}$.

Let us consider now the other rule, $\mathbf{ND_2}$. Suppose we have that:

$$[\![A \circledcirc B]\!]_v \subseteq [\![C]\!]_v$$

As before, let $\top_{[\![C]\!]_v}$ be a top element of $[\![C]\!]_v$. Let $a \in [\![A]\!]_v$ and $b \in [\![B]\!]_v$. We have that:

$$a \times b \leq \top_{[\![C]\!]_v}$$

Hence, by the property (nd_2) of $\mathbb{NDA}$ algebras, for any $i = 1, \cdots, S(A)$ it is the case that:

$$a \times_i b \leq \top_{[\![C]\!]_v} \in [\![C]\!]_v$$

Now, let $\gamma \leq a \times_i b$ for a given i. Since $[\![C]\!]_v$ is d.c., it follows that:

$$\gamma \in [\![C]\!]_v$$

This proves $\mathbf{ND_2}$.

Let us see now the soundness of residuation for the nondeterministic triple $(\circledcirc, \Downarrow, \Uparrow)$:

$$A \circledcirc B \to C \text{ iff } A \to C \Uparrow B \text{ iff } B \to A \Downarrow C$$

Let us consider the first iff ($\Uparrow$ case). Let $a \in [\![A]\!]_v$. For every $b \in [\![B]\!]_v$, since by hypothesis $[\![A \circledcirc B]\!]_v \subseteq [\![C]\!]_v$, we have that:

$$a \times b \subseteq [\![C]\!]_v$$

We have then $[\![A]\!]_v \subseteq [\![C \Uparrow B]\!]_v$. This proves the *only if*.

Conversely, let $a \in [\![A]\!]_v$ and $b \in [\![B]\!]_v$. By hypothesis, $a \in [\![C \Uparrow B]\!]_v$. Hence, by definition of $\Uparrow$:

$$a \times b \in [\![C]\!]_v$$

Now, let $\gamma \leq a \times b$. By definition of the interpretation of $\circledcirc$:

$$\text{If } \gamma \leq a \times b \in [\![A \circledcirc B]\!]_v \text{ then } \gamma \in [\![A \circledcirc B]\!]_v$$

We have then the inclusion $[\![A \circledcirc B]\!]_v \subseteq [\![C]\!]_v$. This proves the *if* part. We are done.

□

In the definition of $\widetilde{\mathcal{F}}$, $\widetilde{\mathcal{SF}}$ and $\widetilde{\mathcal{O}}$ which we defined in Completeness I, the result of completeness did not cover the nondeterministic $\circledcirc$. We repeat a proof by the construction of a canonical model, but this time we can cover the nondetermistic product. The canonical displacement algebra is now the following:

(219) $\mathcal{A} = \langle \{\widetilde{\mathcal{O}}_i\}_{i \in \omega}, (\cdot, \cdot), \{|_i\}_{i>0}, \times, \Lambda, \$ \rangle$

Where $\times$ is defined as follows:

$$\gamma \times \delta \stackrel{def}{=} \rho(\overrightarrow{(\rho^{-1}(\gamma))^\bullet \circledcirc (\rho^{-1}(\delta))^\bullet}) \quad (*)$$

Here $(\cdot)^\bullet$ is the type equivalent map defined in Chapter 3. To put $(*)$ in a simpler form, let $\Delta := \rho^{-1}(\delta)$ and $\Gamma := \rho^{-1}(\gamma)$. Then $\gamma \times \delta$ is:

$$\gamma \times \delta = \rho(\overrightarrow{\Gamma^\bullet \circledcirc \Delta^\bullet})$$

We define now the relation $\leq_\vdash$ on $\mathcal{A}$ as follows:

$$\leq_\vdash \stackrel{def}{=} \{(\gamma, \delta) : \gamma, \delta \in \widetilde{\mathcal{O}} \text{ and } \mathbf{hDND} \vdash \rho^{-1}(\gamma) \Rightarrow (\rho^{-1}(\delta))^\bullet\}$$

(220) **Lemma**

$\leq_\vdash$ is a preorder compatible with the operations of the canonical algebra $\mathcal{A}$.

Proof. In the section of Completeness I, we already saw almost all that we have to show here. There only remains the case of the compatibility of $\times$ with $\leq_\vdash$. As in the previous lemma on the compatibility of the preorder with the operations, we give some useful notation, namely $\Delta = \rho^{-1}(\delta)$, $\Gamma = \rho^{-1}(\gamma)$, $\Theta = \rho^{-1}(\tau)$ and $\Sigma = \rho^{-1}(\sigma)$, where δ, γ, τ and σ belong to $\widetilde{\mathcal{O}}$. Consider the following situation:

$$\frac{\delta \leq_\vdash \gamma \qquad \tau \leq_\vdash \sigma}{\delta \times \tau \leq_\vdash \gamma \times \sigma}$$

By hypothesis we have that:

$$\Delta \Rightarrow \overrightarrow{\Gamma^\bullet} \text{ and } \Theta \Rightarrow \overrightarrow{\Sigma^\bullet}$$

For every $i = 1, \cdots, S(\Delta)$:

$$\frac{\Delta \Rightarrow \overrightarrow{\Gamma^\bullet} \qquad \Theta \Rightarrow \overrightarrow{\Sigma^\bullet}}{\Delta|_i\Theta \Rightarrow \overrightarrow{\Gamma^\bullet \circledcirc \Sigma^\bullet}} \circledcirc R$$

By a similar reasoning to the one of lemma (209) (which refers to the preorder compatibility in the case without nondetermistic product-like connectives) we have that from:

$$\Delta|_i\Theta \Rightarrow \overrightarrow{\Gamma^\bullet \circledcirc \Sigma^\bullet}$$

we get:

$$\overrightarrow{\Delta^\bullet}|_i\overrightarrow{\Theta^\bullet} \Rightarrow \overrightarrow{\Gamma^\bullet \circledcirc \Sigma^\bullet}$$

We can then apply the $\circledcirc$ left rule:

$$\frac{\overrightarrow{\Delta^\bullet}|_1\overrightarrow{\Theta^\bullet} \Rightarrow \overrightarrow{\Gamma^\bullet \circledcirc \Sigma^\bullet} \qquad \cdots \qquad \overrightarrow{\Delta^\bullet}|_{S(\Delta)}\overrightarrow{\Theta^\bullet} \Rightarrow \overrightarrow{\Gamma^\bullet \circledcirc \Sigma^\bullet}}{\overrightarrow{\Delta^\bullet \circledcirc \Theta^\bullet} \Rightarrow \overrightarrow{\Gamma^\bullet \circledcirc \Sigma^\bullet}} \circledcirc R$$

Hence we proved that:

$$\delta \times \tau \leq_\vdash \gamma \times \sigma$$

□

(221) **Lemma**
The canonical displacement algebra:

$$\mathcal{A} = \langle \{\widetilde{\mathcal{O}}_i\}_{i\in\omega}, (\cdot,\cdot), \{|_i\}_{i>0}, \times, \Lambda, \$\rangle$$

is effectively a $\mathbb{NDA}$ algebra.

Proof. Let $\Delta = \rho^{-1}(\delta)$, $\Sigma = \rho^{-1}(\sigma)$ and $\Gamma = \rho^{-1}(\gamma)$. Let us see (nd_1):

$$\frac{\delta \times_1 \sigma \leq_{\vdash} \gamma \qquad \cdots \qquad \delta \times_{S(\Delta)} \sigma \leq_{\vdash} \gamma}{\delta \times \sigma \leq_{\vdash} \gamma}$$

Let us suppose that:

$$\Delta|_1\Sigma \Rightarrow \Gamma^{\bullet} \cdots \Delta|_{S(\Delta)}\Sigma \Rightarrow \Gamma^{\bullet}$$

After several steps we can get:

$$\overrightarrow{\Delta^{\bullet}}|_1\overrightarrow{\Sigma^{\bullet}} \Rightarrow \overrightarrow{\Gamma^{\bullet}} \cdots \overrightarrow{\Delta^{\bullet}}|_{S(\Delta)}\overrightarrow{\Sigma^{\bullet}} \Rightarrow \Gamma^{\bullet}$$

We can now apply the $\circledcirc$ left rule:

$$\overrightarrow{\Delta^{\bullet} \circledcirc \Sigma^{\bullet}} \Rightarrow \overrightarrow{\Gamma^{\bullet}}$$

This proves (nd_1).

Let us see (nd_2):

$$\frac{\delta \times \sigma \leq_{\vdash} \gamma}{\delta \times_i \sigma \leq_{\vdash} \gamma}$$

We have that:

$$\overrightarrow{\Delta^{\bullet} \circledcirc \Sigma^{\bullet}} \Rightarrow \Gamma^{\bullet}$$

By Cut:

$$\overrightarrow{\Delta^{\bullet} \odot_i \Sigma^{\bullet}} \Rightarrow \overrightarrow{\Delta^{\bullet} \circledcirc \Sigma^{\bullet}}$$

By Cut:

$$\overrightarrow{\Delta^{\bullet} \odot_i \Sigma^{\bullet}} \Rightarrow \Gamma^{\bullet}$$

This proves (nd_2).
□

Like in Completeness I, we need again the $\vdash^{-1}$ operator:

$$\begin{array}{llll} \vdash^{-1}: & \mathcal{F} & \longrightarrow & \bigcup_{i\in\omega} 2^{\widetilde{\mathcal{O}}_i} \\ & A & \mapsto & \vdash^{-1}(A) \stackrel{def}{=} \{\delta : \delta \in \widetilde{\mathcal{O}} \text{ and } \vdash_{\mathbf{hDND}} \rho^{-1}(\delta) \Rightarrow A\} \end{array}$$

We need now a truth lemma:

(222) **Lemma** (*Truth Lemma for* **cDND**)

For every type A we have:

$$[\![A]\!]_v \ = \vdash^{-1}(A)$$

Where $[\![\cdot]\!]_v$ is the valuation in the canonical model we have been considering throughout the different proofs of the completeness theorems.

Proof. We already covered all the deterministic fragment of **D**. All the proofs of the truth lemma presented for both the so-called implicative fragment and the remaining deterministic product-like connectives are correct for the new nonstandard syntactical interpretation we are considering. Let us see the equality (222) for the nondeterministic product-like connectives.

Let us prove that the following equality holds:

$$[\![A \circledcirc B]\!] \quad = \quad \vdash^{-1}(A \circledcirc B)$$

- $[\subseteq]$:

  Let $\delta \in \widetilde{\mathcal{O}}$ be such that $\delta \leq \alpha \times \beta$ with $\alpha \in [\![A]\!]$ and $\beta \in [\![B]\!]$. For commodity of the reading of the text let $\Delta_A := \rho^{-1}(\alpha)$ and $\Delta_B := \rho^{-1}(\beta)$. By induction hypothesis (i.h.):

  $$\alpha \ \in \vdash^{-1}(A) \text{ and } \beta \ \in \vdash^{-1}(B)$$

  Hence:

  $$\Delta_A \Rightarrow A \text{ and } \Delta_B \Rightarrow B$$

  It follows that:

  $$\overrightarrow{\Delta_A^\bullet} \Rightarrow A \text{ and } \overrightarrow{\Delta_B^\bullet} \Rightarrow B$$

  By application of the left rule for $\circledcirc$:

  $$\frac{\overrightarrow{\Delta_A^\bullet}|_1\overrightarrow{\Delta_B^\bullet} \Rightarrow A \circledcirc B \qquad \cdots \qquad \overrightarrow{\Delta_A^\bullet}|_{S(A)}\overrightarrow{\Delta_B^\bullet} \Rightarrow A \circledcirc B}{\overrightarrow{\Delta_A^\bullet \circledcirc \Delta_B^\bullet} \Rightarrow A \circledcirc B}\circledcirc L$$

  Since $\delta \leq_\vdash \alpha \times \beta$ it follows that:

  $$\Delta \Rightarrow \Delta_A^\bullet \circledcirc \Delta_B^\bullet$$

  By applying Cut we obtain:

  $$\frac{\Delta \Rightarrow \Delta_A^\bullet \circledcirc \Delta_B^\bullet \qquad \overrightarrow{\Delta_A^\bullet \circledcirc \Delta_B^\bullet} \Rightarrow A \circledcirc B}{\Delta \Rightarrow A \circledcirc B}\,Cut$$

Hence:

$$\delta = \rho^{-1}(\Delta) \in \vdash^{-1} (A \circledcirc B)$$

This proves the inclusion.

- [$\supseteq$]: Let $\delta \in \vdash^{-1} (A \circledcirc B)$. Let $\Delta := \rho^{-1}(\delta)$. We have by hypothesis:

$$\Delta \Rightarrow A \circledcirc B$$

Now, $\rho(\overrightarrow{A}) \in [\![A]\!]$ and $\rho(\overrightarrow{B}) \in [\![B]\!]$ for by i.h.:

$$[\![A]\!]_v = \vdash^{-1} (A) \text{ and } [\![B]\!]_v = \vdash^{-1} (B)$$

$$\rho(\overrightarrow{A}) \times \rho(\overrightarrow{B}) = \rho(\overrightarrow{\overrightarrow{A}^\bullet \circledcirc \overrightarrow{B}^\bullet}) = \rho(\overrightarrow{A \circledcirc B})$$

We have:

$$\delta \leq_\vdash \rho^{-1}(\overrightarrow{A}) \times \rho^{-1}(\overrightarrow{B})$$

Hence $\delta \in [\![A \circledcirc B]\!]_v$. We are done.

□

(223) **Lemma**
Consider the canonical displacement model $\mathcal{M} = (\mathcal{A}; [\![\cdot]\!]_v)$: we have that $\mathcal{M}$ is a $\mathbb{PDND}$ model.

Proof. We have already seen that $\mathcal{A}$ is a $\mathbb{DND}$ algebra. It remains to see that for every interpreted type $[\![A]\!]_v$ has a top element.

$$\text{For every } A \in \mathcal{F}, \rho(A) \text{ is a top element of } [\![A]\!]_v \ (*)$$

$(*)$ holds because for every A, $[\![A]\!]_v = \vdash^{-1} (A)$. For every $\delta \in [\![A]\!]_v$, we have that $\mathbf{DND} \vdash \rho^{-1}(\delta) \Rightarrow \overrightarrow{\rho^{-1}(\rho(A))} = \overrightarrow{A}$. This proves then that $\mathcal{M}$ is a $\mathbb{PDNA}$ model. □

(224) **Theorem** (*Completeness for Full* **DND**)

DND is complete for the $\mathbb{PDND}$ syntactical interpretation.

Proof. Suppose that:

$$\mathbb{PDND} \models \Delta \Rightarrow \overrightarrow{A}$$

It follows that $\Delta \Rightarrow \overrightarrow{A}$ holds of the canonical model $\mathcal{M} = (\mathcal{A}; v)$. As we have already proved before we have in the canonical model that:

$$\rho(\Delta) \in [\![\Delta]\!]_v$$

Hence we have that $\rho(\Delta) \in [\![A]\!]_v$, which is equivalent (by the truth lemma) to:

$$\rho(\Delta) \in \vdash^{-1} (A)$$

Therefore $\mathbf{DND} \vdash \Delta = \rho^{-1}(\rho(\Delta)) \Rightarrow \overrightarrow{A}$. i.e.:

$$\mathbf{hDND} \vdash \Delta \Rightarrow \overrightarrow{A}$$

This proves the theorem.

□

4.3.5 Completeness III: giving DND a syntactical interpretation using DA

In this subsection, we propose a syntactical interpretation of **DND** with the aid of the displacement calculus with additive disjunction and conjunction, i.e., **DA**. The other connectives have the same interpretation as in full **D**. We suppose that we are given a syntactical algebra $\mathcal{M} = \langle M, +, \{\times_{i+1}\}_{i\in\omega}, 0, 1\rangle$. We add the following operator $\sqcup$ of sort functionality $(i, i) \to i$ for every $i \in \omega$. Notice that $\sqcup$ does not modify the sort of the arguments which are moreover required to have the same sort. So, if one writes $a \leq b \sqcup c$, it must be understood that a, b and c have the same sort, where the sort map S is defined as usual. Where $\alpha = \underbrace{a_0 + 1 + a_1 + \cdots + a_{n-1} + 1 + a_n}_{n\ 1's}$, S is defined as follows:

$$S(a_0 + 1 + a_1 + \cdots + a_{n-1} + 1 + a_n) \quad \stackrel{def}{=} \quad n$$

We have then the extended syntactical algebra:

$$\mathcal{M} = \langle M, +, \{\times_i\}_{i>0}, \sqcup, 0, 1\rangle \ (\mathbf{DAA})$$

For every $\mathcal{M} \in \mathbf{DAA}$, it is required that the following holds:

(225)

$$\frac{a \leq c \qquad b \leq c}{a \sqcup b \leq c} \qquad \frac{a \leq b}{a \leq b \sqcup c} \text{ For any } c$$

$$\frac{a \leq b}{a \leq c \sqcup b} \text{ For any } c$$

For every $i \in \omega$ we define as usual the sort domains:

$$L_i \subseteq M_i = \{\alpha \in M : S(\alpha) = i\}$$

The interpretation of the connectives we have already seen in the induced deterministic powerset frame remains the same. We now extend the syntactical interpretation of full deterministic **D** to the additive connectives. There are no nondeterministic connectives (neither implicative-like nor product-like). The interpretation of the additive connectives we give is the following which does not follow the linear logic (Girard (1987)) *phase semantics* definition:

$$\begin{array}{lcl} [\![A \,\&\, B]\!]_v & \stackrel{def}{=} & [\![A]\!]_v \cap [\![B]\!]_v \\ [\![A \oplus B]\!]_v & \stackrel{def}{=} & \{\gamma : \exists\, a \in [\![A]\!]_v \exists b \in [\![B]\!]_v \text{ such that } \gamma \leq a \sqcup b\} \end{array}$$

The induced powerset frame model for a given valuation v is constrained as follows:

(226)

$$\text{For every } A \in \mathcal{F}_i \ (i \in \omega),\ \forall a, b \in [\![A]\!]_v,\ a \sqcup b \in [\![A]\!]_v$$

This class of powerset frame models over general preordered displacement algebras with additives will be denoted $\mathbb{PDAA}$. We now see the soundness of **DA** w.r.t. the proposed syntactical interpretation.

(227) **Theorem** (*Soundness of* **DA** *w.r.t.* $\mathbb{PDAA}$)

For any categorical derivable arrow $A \to B \in$ **Arrows**(**DA**), there holds that:

$$\mathbb{PDAA} \models A \to B$$

Proof. Structural postulates of **cD** hold in the syntactical algebra with the preordered product as we have already seen in the theorem of soundness of full deterministictic **D**. Therefore, we have only to check the soundness for the additive connectives. The proof goes by induction on the length of derivations of **cDA**.

- Additive conjunction &:

  Let us see the left rule for &:

$$\frac{A \to C}{A \& B \to C}\ \& L$$

  We have that $[\![A\&B]\!]_v \stackrel{def}{=} [\![A]\!]_v \cap [\![B]\!]_v$ and since $[\![A]\!]_v \subseteq [\![C]\!]_v$ and $[\![A\&B]\!]_v \subseteq [\![A]\!]_v$, we obtain $[\![A\&B]\!]_v \subseteq [\![C]\!]_v$.

  Let us see now the right rule for &:

$$\frac{A \to B \qquad A \to C}{A \to B\&C}\ \& R$$

  Suppose we have:

$$[\![A]\!]_v \subseteq [\![B]\!]_v \qquad [\![A]\!]_v \subseteq [\![C]\!]_v$$

  Hence, $[\![A]\!]_v \subseteq [\![A]\!]_v \cap [\![B]\!]_v \stackrel{def}{=} [\![A\&B]\!]_v$.

- Additive disjunction $\oplus$:

  Let us see the $\oplus$ left rule:

$$\frac{A \to C \qquad B \to C}{A \oplus B \to C}\ \oplus L$$

  Let $a \in [\![A]\!]_v$ and $b \in [\![B]\!]_v$. Both a and b belong to $[\![C]\!]_v$. Since $[\![C]\!]_v$ is closed by joins $\sqcup$, it follows that:

$$a \sqcup b \in [\![C]\!]_v$$

Hence, by the definition of $[\![A \oplus B]\!]_v$ and the fact that interpreted types are downward-closed, for any $c \leq a \sqcup b$ there holds that $c \in [\![C]\!]_v$. Hence:

$$[\![A \oplus B]\!]_v \subseteq [\![C]\!]_v$$

We have to check the soundness of the $\oplus$ right rule:

$$\frac{A \to B}{A \to B \oplus C} \oplus R$$

Let $a \in [\![A]\!]_v$. Let $c \in [\![C]\!]_v$. We have that:

$$a \leq a \sqcup c$$

Since by hypothesis $a \in [\![B]\!]_v$, we have that $a \sqcup c \in [\![B \oplus C]\!]_v$, and hence, given the fact that $a \leq a \sqcup c$, by downward-closedness of $[\![B \oplus C]\!]_v$ $a \in [\![B \oplus C]\!]_v$. We are done.

□

We proceed now to prove completeness. Consider the following canonical algebra:

(228)

$$\mathcal{A} = \langle \{\widetilde{\mathcal{O}}_i\}_{i \in \omega}, (\cdot, \cdot), \{|_i\}_{i>0}, \sqcup, \Lambda, \$\rangle$$

$\sqcup$ is defined as follows:

$$\delta \sqcup \gamma \overset{def}{=} \rho(\Delta^\bullet \oplus \Gamma^\bullet)$$

Where $\delta = \rho(\Delta)$ and $\gamma = \rho(\gamma)$.

(229) **Lemma** (*Preorder Compatibility*)

$\leq_\vdash$ is compatible with the operations.

Proof. We have to check that for every $\delta, \gamma, \tau, \sigma \in \widetilde{\mathcal{O}}$:

$$\frac{\delta \leq_\vdash \gamma \qquad \tau \leq_\vdash \sigma}{\delta \sqcup \tau \leq_\vdash \gamma \sqcup \sigma}$$

We define $\Delta = \rho^{-1}(\delta)$, $\Gamma = \rho^{-1}(\gamma)$, $\Theta = \rho^{-1}(\tau)$, $\Sigma = \rho^{-1}(\sigma)$. We have that (we omit some explanations which have already been seen in similar lemmas):

$$\frac{\Delta \Rightarrow \Gamma^\bullet \qquad \Theta \Rightarrow \Sigma^\bullet}{\delta \sqcup \tau \leq_\vdash \gamma \sqcup \sigma}$$

We have that:

$$\frac{\Delta \Rightarrow \Gamma^\bullet}{\Delta \Rightarrow \Gamma^\bullet \oplus \Sigma^\bullet} \oplus R$$

Similarly:

$$\frac{\Theta \Rightarrow \Sigma^\bullet}{\Theta \Rightarrow \Gamma^\bullet \oplus \Sigma^\bullet} \oplus R$$

We can derive the following:

$$\Delta^\bullet \Rightarrow \Gamma^\bullet \oplus \Sigma^\bullet$$
$$\Theta^\bullet \Rightarrow \Gamma^\bullet \oplus \Sigma^\bullet$$

Applying the $\oplus$ left rule:

$$\frac{\Delta^\bullet \Rightarrow \Gamma^\bullet \oplus \Sigma^\bullet \qquad \Theta^\bullet \Rightarrow \Gamma^\bullet \oplus \Sigma^\bullet}{\overrightarrow{\Delta^\bullet \oplus \Theta^\bullet} \Rightarrow \overrightarrow{\Gamma^\bullet \oplus \Sigma^\bullet}}$$

We have then proved that

$$\delta \sqcup \tau \leq_\vdash \gamma \sqcup \sigma$$

□

Like in Completeness II, we need again the $\vdash^{-1}$ operator which we slightly modify:

$$\begin{array}{llll} \vdash^{-1}: & \mathcal{F}_{\mathbf{DA}} & \longrightarrow & \bigcup_{i\in\omega} 2^{\widetilde{\mathcal{O}}_i} \\ & A & \mapsto & \vdash^{-1}(A) \stackrel{def}{=} \{\delta : \delta \in \widetilde{\mathcal{O}} \text{ and } \vdash_{\mathbf{hDA}} \rho^{-1}(\delta) \Rightarrow A\} \end{array}$$

(230) **Lemma** (*Truth Lemma for* **DA**)

For every $A \in \mathcal{F}_{\mathbf{DA}}$:

$$[\![A]\!]_v = \vdash^{-1}(A)$$

Proof. The proof is similar to the cases of the previous truth lemmas. We have only to check the connectives & and $\oplus$.

- &: We have to show the following equality:

$$[\![A\&B]\!]_v = \vdash^{-1}(A\&B)$$

[⊆]:

Let $\delta \in [\![A\&B]\!]_v$. We have then that:

$$\delta \in [\![A]\!]_v \text{ and } \delta \in [\![B]\!]_v$$

Let $\Delta = \rho^{-1}(\delta)$. By induction hypothesis (i.h.):

$$\Delta \Rightarrow \overrightarrow{A} \text{ and } \Delta \Rightarrow \overrightarrow{B}$$

Applying the & right rule:

$$\Delta \Rightarrow \overrightarrow{A\&B}$$

Hence, $\delta \in \vdash^{-1} (A\&B)$.

[$\supseteq$]:
Let $\delta \in \vdash^{-1} (A\&B)$. Hence:

$$\Delta \Rightarrow A\&B$$

Since & is an invertible connective[7] we have that:

$$\begin{array}{c}\Delta \Rightarrow A \\ \Delta \Rightarrow B\end{array}$$

By i.h. $\delta \in [\![A]\!]_v$ and $\delta \in [\![B]\!]_v$. Hence:

$$\delta \in [\![A\&B]\!]_v$$

- $\oplus$: We have to show the following equality:

$$[\![A \oplus B]\!]_v = \vdash^{-1} (A \oplus B)$$

[$\subseteq$]: Let $\gamma \in [\![A \oplus B]\!]_v$. By definition of the interpretation of the connective $\oplus$:

$$\gamma \leq_{\vdash} \delta_A \sqcup \delta_B$$

Now, $\delta_A \sqcup \delta_B = \rho(\Delta_A^\bullet \oplus \Delta_A^\bullet)$. We have that:

$$\begin{array}{c}\Delta_A^\bullet \Rightarrow A \\ \Delta_B^\bullet \Rightarrow B\end{array}$$

By two applications of the $\oplus$ right rule we get:

$$\begin{array}{c}\Delta_A^\bullet \Rightarrow A \oplus B \\ \Delta_B^\bullet \Rightarrow A \oplus B\end{array}$$

By application of the $\oplus$ left rule we have that:

$$\overrightarrow{\Delta_A^\bullet \oplus \Delta_B^\bullet} \Rightarrow \overrightarrow{A \oplus B}$$

Hence, $\delta_A \sqcup \delta_B \in \vdash^{-1} (A \oplus B)$.

[$\supseteq$]: Let $\delta \in \vdash^{-1} (A \oplus B)$. It follows that:

$$\Delta \Rightarrow \overrightarrow{A \oplus B}$$

[7]We can prove this by Cut and the following facts:

$$\begin{array}{c}A\&B \Rightarrow A \\ A\&B \Rightarrow B\end{array}$$

Where $\rho^{-1}(A) \sqcup \rho^{-1}(B) = \rho(A \oplus B) \in [\![A \oplus B]\!]_v$. Hence, for any $\delta \leq_\vdash \rho(A \oplus B)$ there holds:

$$\delta \in [\![A \oplus B]\!]_v$$

We are done.
□

(231) **Theorem** (*Completeness for Full* **DA**)

DA is complete for the $\mathbb{PDAA}$ syntactical interpretation.

Proof. Suppose that:

$$\mathbb{PDAA} \models \Delta \Rightarrow \overrightarrow{A}$$

It follows that $\Delta \Rightarrow \overrightarrow{A}$ holds of the canonical model $\mathcal{M} = (\mathcal{A}; v)$. As we have already proved before we have in the canonical model that:

$$\rho(\Delta) \in [\![\Delta]\!]_v$$

Hence we have that $\rho(\Delta) \in [\![A]\!]_v$, which is equivalent (by the truth lemma) to:

$$\rho(\Delta) \in \vdash^{-1}(A)$$

Therefore $\mathbf{DA} \vdash \Delta = \rho^{-1}(\rho(\Delta)) \Rightarrow \overrightarrow{A}$, i.e.:

$$\mathbf{hDA} \vdash \Delta \Rightarrow \overrightarrow{A}$$

This proves the theorem.
□

4.3.6 Using $\mathbb{PDAA}$ syntactical interpretation for the interpretation of DND

Let $\Delta \in \mathcal{O}_{\mathbf{DND}}$ and $A \in \mathcal{F}_{\mathbf{DND}}$. In Chapter 3 we saw the following lemma which we recall:

$$\mathbf{DND} \vdash \Delta \Rightarrow \overrightarrow{A} \text{ iff } \mathbf{DA} \vdash \tau(\Delta) \Rightarrow \overrightarrow{\tau(A)}(\star)$$

where τ is the faithfull embedding translation between **DND** and **DA**. The faithful embedding translation of Chapter 3 is the following:

$$\begin{array}{rcl}
\tau(A) & \stackrel{def}{=} & A \text{ if } A \text{ is atomic} \\
\tau(A \circledcirc B) & \stackrel{def}{=} & \bigoplus_{i=1,\cdots,S(A)} \tau(A) \odot_i \tau(B) \\
\tau(B \Uparrow A) & \stackrel{def}{=} & (\tau(B) \uparrow_1 \tau(A)) \& \cdots \& (\tau(B) \uparrow_n \tau(B)) \text{ where } n = S(B) - S(A) + 1 \\
\tau(A \Downarrow B) & \stackrel{def}{=} & (\tau(A) \downarrow_1 \tau(B)) \& \cdots \& (\tau(A) \downarrow_n \tau(B)) \text{ where } n = S(B) - S(A) + 1 \\
\tau(A * B) & \stackrel{def}{=} & \tau(A) * \tau(B) \text{ for other binary connectives}
\end{array}$$

(232) **Definition** (**DND** *Interpretation with* **DAA** *Algebras*)

Given a $\mathbb{DAA}$ model $\mathcal{M} = (\mathcal{A}, v)$, we can interpret via the faithful τ $\mathcal{F}_{\mathbf{DND}}$ types:

$$\begin{array}{rcl} [\![A \oplus B]\!]_v & \stackrel{def}{=} & [\![\bigoplus_{i=1,\cdots,S(A)} \tau(A) \odot_i \tau(B)]\!]_v \\ [\![B \Uparrow A]\!]_v & \stackrel{def}{=} & [\![(\tau(B) \uparrow_1 \tau(A)) \,\&\, \cdots \,\&\, (\tau(B) \uparrow_n \tau(B))]\!]_v \\ [\![B \Uparrow A]\!]_v & \stackrel{def}{=} & [\![(\tau(A) \downarrow_1 \tau(B)) \,\&\, \cdots \,\&\, (\tau(A) \downarrow_n \tau(B))]\!]_v \end{array}$$

(233) **Theorem** (*Completeness of* **DND** *w.r.t.* $\mathbb{DAA}$ *algebras*)

Let $\Delta \in \mathcal{O}_{\mathbf{DND}}$ and $A \in \mathcal{F}_{\mathbf{DND}}$; we have that:

$$\mathbf{DND} \vdash \Delta \Rightarrow \overrightarrow{A} \text{ iff } \mathbb{PDAA} \models \tau(\Delta) \Rightarrow \overrightarrow{\tau(A)}$$

Proof. We have that:

$$\begin{array}{ll} \mathbf{DND} \vdash \Delta \Rightarrow \overrightarrow{A} & \text{iff} \\ \mathbf{DA} \vdash \tau(\Delta) \Rightarrow \overrightarrow{\tau(A)} & \text{iff} \\ \mathbb{PDAA} \models \tau(\Delta) \Rightarrow \overrightarrow{\tau(A)} & \end{array}$$

Hence, following the chain of iff's:

$$\mathbf{DND} \vdash \Delta \Rightarrow \overrightarrow{A} \text{ iff } \mathbb{PDAA} \models \tau(\Delta) \Rightarrow \overrightarrow{\tau(A)}$$

This completes the proof. □

Chapter 5

On the generative capacity of D-grammars

5.1 On Discontinuous Lambek grammars

Digo para no mucho
En fin cansar.
Luis Valentín (Breviario de Extinción)

This chapter is based on Morrill and Valentín (2010d). Let Σ be a (finite) alphabet containing a distinguished element 1 which plays the role of a *prime* (see Chapters 2 and 3 for the definition of a prime). As 1 is a prime with respect to Σ, strings belonging to Σ^+ decompose uniquely around 1. We define the set of *admissible* words which can inhabit a syntactic type as follows:

(234) $\textbf{AdmissWords} \overset{def}{=} \Sigma^+ - \{\underbrace{1 + \cdots + 1}_{n} : n > 0\}$

We see then the empty string as well as strings containing only primes 1 cannot inhabit types. This will be justified by the fact that the lexical empty string assignment or the assignment of a string formed by only primes constitute a source of undecidability. By contrast, as we shall see, the problem of language recognition in the class of displacement grammars with lexicons with types inhabited by elements of **AdmissWords** is decidable.

A lexical assignment comprises a type A and a string $\alpha \in \textbf{AdmissWords}$ of sort $S(A)$. A *lexicon* is a finite set of lexical assignments. More formally:

(235) **Definition** (*Lexicon*)

A *lexicon* **Lex** is a finite subrelation of $\textbf{AdmissWords} \times \mathcal{F}$ such that for every $(\alpha, A) \in \textbf{Lex}$ the sorts of the string α and the type A are the same, i.e.:

$$S(\alpha) = S(A)$$

Typically, an element of **Lex** will be denoted $w : A$. Lexicons will be displayed as follows:

$$\begin{array}{lll} \mathtt{w}_1 & : & A_1^{\mathtt{w}_1}, \cdots, A_{n_{\mathtt{w}_1}}^{\mathtt{w}_1} \\ \mathtt{w}_2 & : & A_1^{\mathtt{w}_2}, \cdots, A_{n_{\mathtt{w}_2}}^{\mathtt{w}_2} \\ & \vdots & \\ \mathtt{w}_\mathtt{k} & : & A_1^{\mathtt{w}_\mathtt{k}}, \cdots, A_{n_{\mathtt{w}_\mathtt{k}}}^{\mathtt{w}_\mathtt{k}} \end{array}$$

A lexical entry can be read in two ways: A raw $\mathtt{w}_\mathtt{k}{:}A_1^{\mathtt{w}_\mathtt{k}}, \cdots, A_{n_{\mathtt{w}_\mathtt{k}}}^{\mathtt{w}_\mathtt{k}}$ means that the word w_k is assigned $n_{\mathtt{w}_\mathtt{k}}$ types. Another view is to say that the types $A_1^{\mathtt{w}_\mathtt{k}}, \cdots, A_{n_{\mathtt{w}_\mathtt{k}}}^{\mathtt{w}_\mathtt{k}}$ are all inhabited by the word $\mathtt{w}_\mathtt{k}$.

We recall now from chapter 3 the definitions of *level* of a type and *atomicity* of a type.

(236) **Definition** (*Level of a Type*)

A type $A \in \mathcal{F}$ is said to be of *level* l iff l is the maximum sort of the (finite) set of subterms of A, i.e.:

$$\text{level}(A) \stackrel{def}{=} \max\{S(B) : B \text{ is a subtype of } A\}$$

We illustrate this definition with some examples. Let A, B be atomic types of sort 0. Consider the type $C := (((((B{\uparrow_1}A){\uparrow_1}A){\uparrow_1}A) \odot_1 A) \odot_1 A) \odot_1 A$. Clearly $S(C) = 0$. But there are subtypes of C such that their sort is greater than 0, for instance:

$$((B{\uparrow_1}A){\uparrow_1}A){\uparrow_1}A$$

This type has sort 3. In this case, a little inspection at the (finite) algebra of subtypes of C gives that the level of C is 3, i.e. $level(C) = 3$.
Consider now the type $B{\uparrow_1}A$. Clearly it has sort 1 and its level is 1 ($B{\uparrow_1}A$ is a subtype of $B{\uparrow_1}A$!).

The *atomicity* of a type A gives the maximum sort of all the atomic types which are subtypes of A:

(237) **Definition** (*Atomicity of a Type*)

Let $A \in \mathcal{F}$. The atomicity of A is defined as follows:

$$atomicity(A) \stackrel{def}{=} \max\{S(B) : B \text{ is atomic and is a subtype of } A\}$$

Let us look again at the type C used above to illustrate the level of a type. A look at $C := (((((B{\uparrow_1}A){\uparrow_1}A){\uparrow_1}A) \odot_1 A) \odot_1 A) \odot_1 A$ shows that all the atomic types of C are of sort 0. Hence $atomicity(C) = 0$. In the following we extend these three notions to lexicons.

Given a lexicon **Lex**, it will be called a sort-k lexicon iff k equals the upper sort of the type assignments, i.e.:

(238) **Definition** (*Sort of a Lexicon*)

Let **Lex** be a lexicon. The sort $S(\mathbf{Lex})$ is defined as:

$$k = \max_{(w:A)\in\mathbf{Lex}} S(A)$$

(239) **Definition** (*Level of a Lexicon*)

A discontinuous Lambek lexicon **Lex** is said to be of level l if the maximum of all the levels of the types of **Lex** is equal to l:

$$level(\mathbf{Lex}) \stackrel{def}{=} \max\{level(A) : \text{ where } A \text{ is a type of } \mathbf{Lex}\}$$

Finally we look at the atomicity of a lexicon:

(240) **Definition** (*Atomicity of a Lexicon*)

Let **Lex** be a lexicon. The *atomicity* of **Lex** is defined as follows:

$$atomicity(\mathbf{Lex}) \stackrel{def}{=} \max\{atomicity(A) : \text{ where } A \text{ is a type of } \mathbf{Lex}\}$$

(241) **Example**
Let us consider the following lexicons $\mathbf{Lex}_1$, $\mathbf{Lex}_2$ and $\mathbf{Lex}_3$. Here n, s are atomic types of sort 0 and r an atomic type of sort 1:

$$\mathbf{Lex}_1 = \begin{cases} \texttt{everyone} : (s\uparrow n)\downarrow s \\ \texttt{love} : (n\backslash s)/n \\ \texttt{Mary} : n \end{cases}$$

$$\mathbf{Lex}_2 = \begin{cases} \texttt{a}+\texttt{1}+\texttt{b}+\texttt{1} : (s\uparrow_1 n)\uparrow_2 n \\ \texttt{c} : s/s \\ \texttt{d} : n \\ \texttt{e} : n \end{cases}$$

$$\mathbf{Lex}_3 = \begin{cases} \texttt{a}+\texttt{1}+\texttt{b} : (r\uparrow_2 n)\odot_1 n \\ \texttt{c} : r\downarrow s \end{cases}$$

$\mathbf{Lex}_1$ is a sort 0 lexicon. $\mathbf{Lex}_2$ and $\mathbf{Lex}_3$ are respectively a sort 2 lexicon and a sort 1 lexicon. $\mathbf{Lex}_1$ and $\mathbf{Lex}_2$ are 0-atomic lexicons because there is no type assignment which uses atomic types of sort greater than 0, and $\mathbf{Lex}_3$ is a lexicon of atomicity 1. Finally $\mathbf{Lex_1}$ is a lexicon of level 1, and lexicons $\mathbf{Lex_2}$ and $\mathbf{Lex_3}$ are lexicons of level 2.

What is a discontinuous Lambek grammar? As in the case of Lambek grammars, a discontinuous Lambek grammar is represented by two objects, one lexicon and a distinguised type. More concretely:

(242) **Definition** (*Discontinuous Lambek Grammars*)

A discontinuous Lambek grammar is a pair $\mathcal{G} = (\mathbf{Lex}, S)$, i.e., a pair consisting of a discontinuous Lambek lexicon and a distinguished type.

We will refer to discontinuous Lambek grammars as **D**-grammars.

(243) **Definition** (*k-Discontinuous Lambek Grammars*)

A discontinuous Lambek grammar $\mathcal{G} = (\mathbf{Lex}, S)$ is said to be of level k, in notation a k-**D**-grammar, if and only if **Lex** is of level k.

We will usually write k-grammar instead of k-**D**-grammar.

We now turn to **D**-grammars and the languages they generate. We define a *labelling* map σ of a hyperconfiguration Δ as a mapping sending each type occurrence A in Δ to a string of sort sA.

(244)
$$\begin{array}{rccl} \sigma : & \Delta & \longrightarrow & \Sigma^+ \\ & A & \mapsto & \sigma(A) \text{ such that } \sigma(A) \in \mathbf{AdmissWords} \\ & [] & \mapsto & 1 \end{array}$$

A *labelled hyperconfiguration* Δ^σ comprises a hyperconfiguration Δ and a labelling σ of Δ. We define the *yield* of a labelled hyperconfiguration Δ^σ as follows:

(245)
$$\begin{array}{l} yield(\Lambda^\sigma) = \Lambda \\ yield([]^\sigma) = 1 \\ yield((\Delta, \Gamma)^\sigma) = yield(\Delta^\sigma) + yield(\Gamma^\sigma) \\ yield(A^\sigma) = \sigma(A) \text{ for } A \text{ of sort } 0 \\ yield((A\{\Delta_1 : \cdots : \Delta_{sA}\})^\sigma) = \\ a_1 + yield(\Delta_1^\sigma) + a_2 + yield(\Delta_2^\sigma) + \cdots + \\ yield(\Delta_{S(A)-1}^\sigma) + a_{s(A)-1} + yield(\Delta_{s(A)}^\sigma) + a_{s(A)} \end{array}$$

where in the last line of the definition A is of sort greater than 0 and $\sigma(A)$ is $a_1 + 1 + a_2 + \cdots + a_{s(A)-1} + 1 + a_{s(A)}$.

A labelling σ of a hyperconfiguration Δ is *compatible* with a lexicon **Lex** if and only if $\sigma(A)$: $A \in \mathbf{Lex}$ for every A in Δ. The language $L(\mathbf{Lex}, A)$ generated from lexicon **Lex** for type A is defined as follows:

(246) $L(\mathbf{Lex}, A) = \{yield(\Delta^\sigma)|$ such that $\Delta \Rightarrow A$ is a theorem of **D** and σ is compatible with $\mathbf{Lex}\}$

(247) **Theorem** (*Recognition of* **D**-*Grammars*)

The problem of recognition in the class of **D**-grammars is decidable. Given a **D**-grammar $(\mathbf{Lex}, A)$ and a word $w \in \mathbf{AdmissWords}$:

(248) The problem $w \in^? L(\mathbf{Lex}, A)$ is decidable.

Proof. We are given $(\mathbf{Lex}, A)$ and $w \in \mathbf{AdmissWords}$. Consider the following set X:

$$X = \{(\Delta, \sigma) : \Delta \in \mathcal{O},\ \sigma \text{ is compatible with } \mathbf{Lex} \text{ and } yield(\Delta, \sigma) = w\}$$

(249) **Claim** X is finite.

Let us prove claim (249). By way of contradiction suppose X is infinite. Let us suppose that the set of underlying hyperconfigurations is infinite. In this case then we would have an infinite sequence of underlying hyperconfigurations $(\Delta_n)_{n\in\omega}$, such that their number of types occurrences would be unbounded, because the number of different types occurring in Δ is finite.[1] In that case, $length(yield(\Delta_n, \sigma))$ would be also unbounded because any occurrence in Δ_n would contribute with an element of **AdmissWords** which has at least one element of $\Sigma - \{1\}$. We have proved that the set of underlying hyperconfigurations of X is finite. Let us see that the number of possible different labelling maps of X is also finite. This is simply due to the fact that **Lex** is finite and hence the number of compatible labelling maps is finite.

Now as theoremhood in **D** is decidable and X is a finite set, we have then that the problem of recognition is decidable since it reduces to a finite number of tests of theoremhood. □

A Prolog parser/theorem-prover for the calculus of displacement has been implemented. It operates by Cut-free backward-chaining hypersequent proof search (see Morrill (2011a)).

5.1.1 A Source of Undecidability

Let us consider the question of lexicons with empty string assignments. We will work in this reduction with the unidirectional Lambek calculus without product, i.e., $\mathbf{L}_{\{/\}}$.[2] Buszkowski (Buszkowski (1982)) proves that the languages recognized by the class of $\mathbf{L}_{\{/\}}$ grammars with a finite set of non-logical axioms are the class of recursively enumerable languages. Let $\mathbf{F_{Ax}}$ be a finite set of non-logical axioms of the form $A \Rightarrow B$ with $A, B \in \mathcal{F}_/$.

Let $\mathbf{Lex_{F_{Ax}}}$ be a lexicon of $\mathbf{L}_{\{/\}}+\mathbf{F_{Ax}}$. We consider a lexicon in $\mathbf{L}_\epsilon$, $\mathbf{Lex}_\epsilon$, as follows:

$$\mathbf{Lex}_\epsilon \stackrel{def}{=} \mathbf{Lex_{F_{Ax}}} \cup \{\epsilon : B/A : \text{ such that } A \Rightarrow B \in \mathbf{F_{Ax}}\}$$

We present a reduction map τ from $\mathbf{L}_{\{/\}}+\mathbf{F_{Ax}}$ derivations into $\mathbf{Lex}_\epsilon$ derivations. Every proof derivation of a sequent in $\mathbf{L}^{Ax}_{\{/\}}$ is mapped into a proof derivation of a sequent in $\mathbf{L}_\epsilon$ without any trouble in the case of left/right logical rules for the / connective (in fact they are identically mapped). The only interesting case is when we have an instance of the Cut rule with one premise belonging to the finite set of non-logical axioms $\mathbf{F_{Ax}}$. We have:

(250)
$$\vdash_{\mathbf{L}^{Ax}_{\{/\}}} \dfrac{\dfrac{}{\alpha : A \Rightarrow \alpha : B}\mathbf{F_{Ax}} \qquad \Delta(b : B) \Rightarrow \beta[b] : C}{\Delta(\alpha : A) \Rightarrow \beta[\alpha] : C}\,Cut$$
iff
$$\vdash_{\mathbf{L}_\epsilon} \dfrac{\alpha : A \Rightarrow \alpha : A \qquad \tilde{\Delta}(b : B) \Rightarrow \beta[b] : C}{\tilde{\Delta}(\epsilon : B/A, \alpha : A) \Rightarrow \beta[\epsilon + \alpha] = \beta[\alpha] : C}\,/L$$

[1] If we disposed of an infinite number $(A_n)_{n\in\omega}$ of types then we would have an infinite number of different hyperconfigurations, say $\Delta_n := A_n, n \geq 0$.

[2] We could of course consider the other slash, i.e., \. The same result on undecidability would hold.

Here, type are labelled with syntactical for. Now, the correspondance (250) is such that the use of non-logical axioms is substituted by empty string lexical assignments coding the non-logical axioms. In (250), a proof-derivation $\mathcal{D}$ of a sequent $\Delta \Rightarrow \beta : C$ is mapped into a proof derivation $\tau(\mathcal{D})$ of a sequent $\tilde{\Delta} \Rightarrow \beta : C$. This correspondance crucially assigns the same syntactical term β to the succedent type C. It follows then that a labelling map σ of Δ is compatible with $\mathbf{Lex}_{\mathbf{F_{Ax}}}$ if and only if the labelling map $\sigma \cup \{(B/A, \epsilon) :$ for assignments $\epsilon : B/A \in \mathbf{Lex}_\epsilon\}$ is compatible with $\mathbf{Lex}_\epsilon$. Hence for a given $w \in \Sigma^+$, $w \in L(\mathbf{Lex}_{\mathbf{F_{Ax}}}, S)$ if and only if $w \in L(\mathbf{Lex}_\epsilon, S)$. It follows then that the problem of recognition of languages defined by Lambek grammar lexicons with empty string assignments is undecidable, i.e.:

(251) **Theorem** (*Undecidability of Empty String Lexical Assignments*)

Let $\mathbf{Lex}_\epsilon$ be a Lambek lexicon with possibly empty string lexical assignments. The following problem is undecidable:

$$w \overset{?}{\in} L(\mathbf{Lex}_\epsilon, S)$$

As Morrill (Morrill (2011b)) points it out the remarkable (and desirable) finite reading property of type-logical grammars is lost in presence of lexicons with empty string lexical assignments.[3] Of course, the reasoning in the case of the Lambek calculus works also in the case of lexical assignments of the type $1 : A$. This gives us the following corollary:

(252) **Corollary** (*Undecidability of Empty String or* 1 *Lexical Assignments*)

Let **Lex** be a displacement lexicon with possibly empty string or 1 lexical assignments. The following problem is undecidable:

$$w \overset{?}{\in} L(\mathbf{Lex}, S)$$

As a conclusion, we could say that the disposal of empty string lexical assignments is comparable with the use of lexical rules which as we know in many cases leads the systems to undecidability.

5.2 On the Generative Capacity of D-grammars

In this section we take a look at some properties of the (weak) generative capacity of **D**-grammars.

5.2.1 Some non-context free D-languages

In this section we abbreviate $\downarrow_1$, $\odot_1$ and $\uparrow_1$ as $\downarrow$, $\odot$ and $\uparrow$; only discontinuities with a single separator are considered (i.e. we consider 1-grammars).
The non-context free language $\{a^n b^n c^n |\ n > 0\}$ is generated by the following assignments where $s(A) = s(B) = s(C) = 1$ and the distinguished type is $A \odot I$.

[3]Morrill (*op.cit.*) in fact notices the loss of the finite reading property with assignments of the type $1 : A$ where 1 is the prime.

(253) b: $J\backslash B, J\backslash(A{\downarrow}B)$
c: $B\backslash C$
a: A/C

The assignment b: $J\backslash B$ generates $1{+}b$: B. Then combination with the assignment for c generates $1{+}b{+}c$: C and combination of this with the assignment for a gives $a{+}1{+}b{+}c$: A. Wrapping this around the product unit gives $a{+}b{+}c$: $A{\odot}I$; alternatively b: $J\backslash(A{\downarrow}B)$ which gives $1{+}b$: $A{\downarrow}B$ can infix to form $a{+}1{+}b{+}b{+}c$: B which combines with c and a again, and so on.

The non-context free copy language $\{ww|\ w \in \{a,b\}^+\}$ is generated by the following assignments where $s(A) = s(B) = 0$ and $s(S) = 1$ and the distinguished type is $S{\odot}I$.

(254) a: $J\backslash(A\backslash S), J\backslash(S{\downarrow}(A\backslash S)), A$
b: $J\backslash(B\backslash S), J\backslash(S{\downarrow}(B\backslash S)), B$

The two lexicons of these examples are of sort 0, level 1, atomicity 1, and the distingushed type is a complex type.

Let G be a rewrite grammar containing productions of the form $A \to a$ and $B \to cD \mid Dc$. Replacing the former by a: A and the latter by c: $(D{\uparrow}I){\downarrow}B$ gives a displacement grammar which generates the permutation closure of $L(G)$. It follows that there is a displacement grammar for every language Mix_n of strings with equal numbers of symbols $a_1, \ldots, a_n$. In particular, the non-context free language $Mix = \{w \in \{a,b,c\}|\ |w|_a = |w|_b = |w|_c > 0\}$ is generated by the following assignments:

(255) a: $A, (S{\uparrow}I){\downarrow}A$
b: $(A{\uparrow}I){\downarrow}B$
c: $(B{\uparrow}I){\downarrow}S$

Here $s(A) = s(B) = s(C) = 0$ and the distinguished type is S. Appendix B contains a sample derivation of this **D**-grammar for Mix. Interestingly, this non-context free language has a lexicon of sort 0, level 0, atomicity 0, and the distinguished type is atomic. In the next subsection we generalize the results for the languages Mix and Mix_n.

5.2.2 Lower bounds on the recognizing power of D-grammars

D-grammars and the permutation closure of Context-Free grammars

In this section we prove that **D**-grammars recognize the permutation closures of the context-free languages.

This result is obtained using a restricted fragment of the calculus. We define the set $T = \{A|\ A \text{ is an atomic type}\} \cup \{(A{\uparrow}I){\downarrow}B|\ A \text{ and } B \text{ are atomic types}\}$. A *T-hypersequent* is a hypersequent such that the types of the antecedent belong to T and the succedent is an atomic type.

As we saw in Chapter 3 the ˇ operation can be used to *simulate* types of the form $A{\uparrow}I$:

$$\check{}A \stackrel{def}{=} A{\uparrow}I$$

In the following lines we will use both notations.

(256) **Lemma** (*Rearrangement Lemma*)

Let $\Delta \Rightarrow S$ be a provable T-hypersequent with at least one occurrence of an implicative type (i.e. of the form $\check{}R{\downarrow}S$ for some atomic R, S) in the antecedent. Then, where $\mathcal{D}$ is a derivation of $\Delta \Rightarrow S$, $\mathcal{D}$ can be rearranged into a new derivation $\mathcal{D}^\star$ of $\Delta \Rightarrow S$ such that the last rule of $\mathcal{D}^\star$ has an axiom $S \Rightarrow S$ as the right premise of the $\downarrow$ left rule, i.e.:

$$\mathcal{D} \vdash \frac{\vdots}{\Delta \Rightarrow S}{\downarrow}L \quad \rightsquigarrow \quad \mathcal{D}^\star \vdash \frac{\overset{\vdots}{\Gamma([]) \Rightarrow \check{}R} \qquad S \Rightarrow S}{\Delta \Rightarrow S}{\downarrow}L$$

where $\Delta = \Gamma(\check{}R{\downarrow}S)$ for some atomic type R.

Proof.

Let us prove the result by induction on the implicative weight $|\cdot|$ of the antecedents of the provable T-hypersequent. Here $|\cdot|$ is defined for types as the number of implicative connective occurrences (in this case, the infix connective $\downarrow$), i.e.:

$$\begin{array}{rcl} |A| & = & 0 \text{ for some atomic } A \\ |\check{}A{\downarrow}B| & = & 1 \text{ for atomic } A, B \end{array}$$

In the case of the antecedents of T-hypersequents the weight of the types occurring will be 0 or 1. $|\cdot|$ is extended to antecedents as follows:

$$\begin{array}{lll} |A, \Delta| & =_{def} & |A| + |\Delta| \\ |\Lambda| & = & 0 \end{array}$$

- Base case: suppose the provable T-hypersequent $\Delta \Rightarrow A$ has the least implicative weight, i.e., 1. We have then two possibilities:

$$\frac{R, [] \Rightarrow \check{}R \qquad S \Rightarrow S}{R, \check{}R{\downarrow}S \Rightarrow S}{\downarrow}L$$

$$\frac{[], R \Rightarrow \check{}R \qquad S \Rightarrow S}{\check{}R{\downarrow}S, R \Rightarrow S}{\downarrow}L$$

- Case where $|\Delta| = n + 1$ with $n > 0$. Let us suppose that the derivation of the T-hypersequent is as follows:

$$\frac{\Gamma([]) \Rightarrow \check{}P \qquad \Delta(Q, \check{}R{\downarrow}S) \Rightarrow S}{\Delta(\Gamma(\check{}P{\downarrow}Q), \check{}R{\downarrow}S) \Rightarrow S}{\downarrow}L$$

Clearly $|\Delta(Q, \check{}R{\downarrow}S)| < n + 1 = |\Delta(\Gamma(\check{}P{\downarrow}Q), \check{}R{\downarrow}S)|$ because there is no occurrence of $\check{}P{\downarrow}Q$ in $\Delta(Q, \check{}R{\downarrow}S)$ and hence its implicative weight is necessarily less than the one of $\Delta(\Gamma(\check{}P{\downarrow}Q), \check{}R{\downarrow}S)$. Hence we can apply the induction hypotheses to the hypersequent $\Delta(Q, \check{}R{\downarrow}S) \Rightarrow S$ obtaining the following derivation:

$$\dfrac{\Gamma([]) \Rightarrow \check{}P \qquad \dfrac{\Delta(Q,[]) \Rightarrow \check{}R \qquad S \Rightarrow S}{\Delta(Q, \check{}R{\downarrow}S) \Rightarrow S}\,{\downarrow}L}{\Delta(\Gamma(\check{}P{\downarrow}Q), \check{}R{\downarrow}S) \Rightarrow S}\,{\downarrow}L$$

We now rearrange the above derivation as follows:

$$\dfrac{\dfrac{\Gamma([]) \Rightarrow \check{}P \qquad \Delta(Q;[]) \Rightarrow \check{}R}{\Delta(\Gamma(\check{}P{\downarrow}Q),[]) \Rightarrow \check{}R}\,{\downarrow}L \qquad S \Rightarrow S}{\Delta(\Gamma(\check{}P{\downarrow}Q); \check{}R{\downarrow}S) \Rightarrow S}\,{\downarrow}L$$

This completes the proof.

□

(257) **Lemma** (*Fronting lemma*)

Let $\Delta(A) \Rightarrow S$ be a provable T-hypersequent with a distinguished occurrence of type A. Then:

$$\vdash A, \Delta(\Lambda) \Rightarrow S$$

Proof. We proceed by induction on the length of hypersequents. We shall write $A_1, \cdots, A_j, \cdots, A_n \Rightarrow S$ for $\Delta(A)$ where we consider A_j as the distinguished occurrence we want to be displaced to the left of the antecedent. By the previous lemma, $A_1, \cdots, A_j, \cdots, A_n \Rightarrow S$ has a derivation with last rule:[4]

$$\dfrac{A_1, \cdots, A_{i-1}, [], A_{i+1} \cdots, A_j, \cdots, A_n \Rightarrow R{\uparrow}I \qquad S \Rightarrow S}{A_1, \cdots, A_{i-1}, (R{\uparrow}I){\downarrow}S, A_{i+1} \cdots, A_j, \cdots, A_n \Rightarrow S}\,{\downarrow}L$$

Two cases are considered:

• Case $A_j \neq (R{\uparrow}I){\downarrow}S$.

We have $(R{\uparrow}I){\odot}I \Rightarrow R$ is provable. By applying the Cut rule to the left premise of the last rule, we derive:

$$A_1, \cdots, A_{i-1}, \Lambda, A_{i+1} \cdots, A_j, \cdots, A_n \Rightarrow R$$

Hence by induction hypothesis:

$$\vdash A_j, A_1, \cdots, A_{i-1}, \Lambda, A_{i+1} \cdots, \Lambda, \cdots, A_n \Rightarrow R$$

We apply now the $\uparrow$ right rule after the introduction of the unit I:

$$\dfrac{A_j, A_1, \cdots, A_{i-1}, I, A_{i+1} \cdots, \Lambda, \cdots, A_n \Rightarrow R}{A_j, A_1, \cdots, A_{i-1}, [], A_{i+1} \cdots, \Lambda, \cdots, A_n \Rightarrow R{\uparrow}I}\,{\uparrow}R$$

By the $\downarrow$ left rule:

$$\dfrac{A_j, A_1, \cdots, A_{i-1}, [], A_{i+1} \cdots, \Lambda, \cdots, A_n \Rightarrow R{\uparrow}I \qquad S \Rightarrow S}{A_j, A_1, \cdots, A_{i-1}, (R{\uparrow}I){\downarrow}S, A_{i+1} \cdots, \Lambda, \cdots, A_n \Rightarrow S{\uparrow}I}\,{\downarrow}L$$

[4] Without loss of generality we write A_j to the right of $(R{\uparrow}I){\downarrow}S$.

In this case, we have proved the fronting lemma.
• Case $A_j = (R{\uparrow}I){\downarrow}S$.
As before we have the following provable hypersequent:

$$A_1, \cdots, A_{i-1}, \Lambda, A_{i+1}, \cdots, A_n \Rightarrow R$$

By the right $\uparrow$ rule after the introduction of I, we derive:

$$[], A_1, \cdots, A_{i-1}, \Lambda, A_{i+1} \cdots, A_j, \cdots, A_n \Rightarrow R{\uparrow}I$$

By the left $\downarrow$ rule:

$$\frac{[], A_1, \cdots, A_{i-1}, \Lambda, A_{i+1}, \cdots, A_n \Rightarrow R{\uparrow}I \qquad S \Rightarrow S}{(R{\uparrow}I){\downarrow}S, A_1, \cdots, A_{i-1}, \Lambda, A_{i+1}, \cdots, A_n \Rightarrow S}\downarrow L$$

We have proved the fronting lemma in case $A_j = (R{\uparrow}I){\downarrow}S$. In both cases then, the lemma is proved. □

We now show how the permutation closure of any regular language (excluding the empty string) can be recognized by a **D**-grammar. Let $G = (N, \Sigma, P, S)$ be a regular grammar in normal form. Suppose G is right-linear. We define a **D**-grammar comprising a lexicon $\mathbf{Lex_G}$ with atomic types the nonterminals N of G. The vocabulary of $\mathbf{Lex_G}$ is $\Sigma \cup \{1\}$. For every production of the form $A \to c$ with A nonterminal and $c \in \Sigma$, we stipulate that c: $A \in \mathbf{Lex_G}$. And for every production of the form $B \to cA$ (with $A, B \in N$ and $c \in \Sigma$), we stipulate c: $(A{\uparrow}I){\downarrow}B \in \mathbf{Lex_G}$. We want to prove that the language recognized by $\mathbf{Lex_G}$ with distinguished symbol S is the permutation closure of the language generated by G: $L(\mathbf{Lex_G}, S) = Perm(L(G, S))$. The following lemmas prove the equation.

(258) **Lemma**
$L(G, S) \subseteq L(\mathbf{Lex_G}, S)$.

Proof. The proof of this lemma proceeds by a simple induction on the length of the derivations of G. The base case is obvious. For the inductive case, suppose we have the derivation whose rewritten string is $a_1 \cdots a_n A$ such that $A \to cB \in P$. Then by induction hypothesis $a_1 + \cdots + a_n \in L(G, B) \subseteq L(\mathbf{Lex_G}, B)$. Hence there exists a labelled hyperconfiguration Δ^σ whose types belong to the types of $\mathbf{L_G}$, $\vdash \Delta \Rightarrow B$ and the yield of Δ^σ is $a_1 + \cdots + a_n$; after the introduction of the unit:

$$\frac{\dfrac{I, \Delta \Rightarrow B}{[], \Delta \Rightarrow B{\uparrow}I}\uparrow R \qquad A \Rightarrow A}{(B{\uparrow}I){\downarrow}A, \Delta \Rightarrow A}\downarrow L$$

Now, c: $(B{\uparrow}I){\downarrow}A \in \mathbf{Lex_G}$. Hence $c + a_1 + \cdots + a_n \in L(\mathbf{Lex_G}, A)$. □

(259) **Lemma**
$Perm(L(G, S)) \subseteq L(\mathbf{Lex_G}, S)$

Proof. Let $\Delta \Rightarrow S$ be a provable hypersequent with a compatible labelling such that the yield of Δ is $w \in L(G, S)$ and the types occurring in Δ belong to the set of types of $\mathbf{Lex_G}$:

$$a_1 : A_1, \cdots, a_n : A_n \Rightarrow S,\ w = a_1 + \cdots + a_n$$

By the fronting lemma, any type A_i can be fronted, i.e.: $\vdash a_i : A_i, a_1 : A_1, \cdots, a_{i-1} : A_{i-1}, \Lambda, a_{i+1} : A_{i+1}, \cdots, a_n : A_n \Rightarrow S$. By repeating this process via the fronting lemma, any permutation of the initial w can be obtained. □

(260) **Lemma**
$L(\mathbf{Lex_G}, S) \subseteq Perm(L(G, S))$

Proof. We prove that for every atomic type $A \in N$:

$$L(\mathbf{Lex_G}, A) \subseteq Perm(L(G, A))$$

This entails in particular $L(\mathbf{Lex_G}, S) \subseteq Perm(L(G, S))$ where $S \in N$ is the distinguished nonterminal symbol. The proof goes by induction on the length of the antecedent of hypersequents $\Delta \Rightarrow A$ such that the types occurring in Δ belong to the types occurring in $\mathbf{Lex_G}$.

Case where the length of the hypersequent is 1. Three hypersequents are possible:

$$\begin{cases} A \Rightarrow A \\ \Lambda \Rightarrow I \\ [] \Rightarrow J \end{cases}$$

Only the first hypersequent, the axiom case, corresponds to a $\mathbf{Lex_G}$-hypersequent. Here A is a type of **Lex**. By definition of $\mathbf{Lex_G}$ every $a : A$ is such that $A \rightarrow a$ corresponds to a production $A \rightarrow a$ where $A \in N$. The result holds trivially.

Case where the length is greater than 1: suppose we have a derivation of the hypersequent $\Delta \Rightarrow A$, where the types occurring in Δ belong to the lexicon $\mathbf{Lex_G}$. Let $n + 1$ be the length of Δ. Clearly $\Delta \Rightarrow A$ is a T-hypersequent. By the rearrangement lemma, the derivation of $\Delta \Rightarrow A$ can be modified in such a way that the last rule of the derivation is a left $\downarrow$ rule, and the right premise of the rule is an axiom:

$$\frac{\Delta([]) \Rightarrow \check{}B \qquad A \Rightarrow A}{\Delta(\check{}B{\downarrow}A) \Rightarrow A}\downarrow L$$

By application of the $\hat{}$ right rule we have a $\mathbf{Lex_G}$-hypersequent whose length is n:

$$\Delta(\Lambda) \Rightarrow \hat{}\check{}B$$

As $\hat{}\check{}B \Rightarrow B$, by Cut $\Delta(\Lambda) \Rightarrow B$ is derivable. Since the types of $\Delta(\Lambda)$ belong to the types of $\mathbf{Lex_G}$ and the length of $\Delta(\Lambda)$ is n we can apply the induction hypothesis, and then we have that $L(\mathbf{Lex_G}, B) \subseteq Perm(L(G, B))$. Now, every $w \in L(\mathbf{Lex_G}, B)$ is the permutation of some $\widetilde{w} \in L(G, B)$. If we apply the rule $A \rightarrow cB$ (for some $c \in \Sigma$) corresponding to the lexical type assignment c: $\check{}B{\downarrow}A$ we get $c + \widetilde{w} \in L(G, A)$. Hence, if we insert c (by the infixation of $\check{}B{\downarrow}A$) in w we get a permutation of $c + \widetilde{w}$. □

(261) **Theorem** (*Permutation Closure of Regular Languages by* **D**-*Grammars*)

For every regular grammar G we have $L(\mathbf{Lex_G}, S) = Perm(L(G, S))$

(262) **Corollary** (*Permutation Closure of Context-Free Languages by* **D**-*Grammars*)

For every context-free language L, the permutation closure of L $Perm(L)$ is recognized by a **D**-grammar.

Proof. By an argument invoking properties of semi-linear sets,[5] we know that any permutation closure of a context-free language is equal to the permutation closure of some regular language. This reduces the proof of this corollary to the class of regular languages. The previous theorem proves it. □

5.2.3 On Head-Grammars and D-grammars

Let Σ be a fixed alphabet. Elements of Σ will be denoted with latin letters and words belonging to Σ^* will be denoted with greek alphabet letters. In this section strings of sort 1 will be displayed as pairs in order to meet the standard notation of formal language literature. Pairs of strings will be denoted as (α, β) where $\alpha, \beta \in \Sigma^*$. Consider the following linear functions on pairs of strings:

Function	Infix notation	functionality
wrap₁	$\odot_1$	$1 \times 1 \longrightarrow 1$
wrap₂	$\odot_2$	$1 \times 0 \longrightarrow 0$
conc$_>$	$\circ_>$	$1 \times 1 \longrightarrow 1$
conc$_<$	$\circ_<$	$1 \times 1 \longrightarrow 1$
conc₁	$\circ_1$	$1 \times 0 \longrightarrow 1$
conc₂	$\circ_2$	$0 \times 1 \longrightarrow 1$
conc₃	$\circ_3$	$0 \times 0 \longrightarrow 0$
bridge	$\hat{}$	$0 \longrightarrow 1$

These functions are defined as follows:

$$
\begin{array}{llll}
\mathbf{wrap_1}: & L_1 \times L_1 & \longrightarrow & L_1 \\
 & ((x,y),(z,t)) & \mapsto & (xz, ty) \\
\mathbf{wrap_2}: & L_1 \times L_0 & \longrightarrow & L_0 \\
 & ((x,y),z) & \mapsto & xzy \\
\mathbf{conc}_>: & L_1 \times L_1 & \longrightarrow & L_1 \\
 & ((x,y),(z,t)) & \mapsto & (xyz, t) \\
\mathbf{conc}_<: & L_1 \times L_1 & \longrightarrow & L_1 \\
 & ((x,y),(z,t)) & \mapsto & (x, yzt) \\
\mathbf{conc_1}: & L_1 \times L_0 & \longrightarrow & L_1 \\
 & ((x,y),z) & \mapsto & (x, yz) \\
\mathbf{conc_2}: & L_0 \times L_1 & \longrightarrow & L_1 \\
 & (x,(y,z)) & \mapsto & (xy, z) \\
\mathbf{conc_3}: & L_0 \times L_0 & \longrightarrow & L_0 \\
 & (x,y) & \mapsto & x\,y \\
\mathbf{bridge}: & L_1 & \longrightarrow & L_0 \\
 & (x,y) & \mapsto & xy
\end{array}
$$

K Vijay-Shanker (1986) worked with the so-called modified Head-Grammars which use the three linear functions **wrap₁**, **conc**$_>$ and **conc**$_<$.
We can define the following sort polymorphic functions:

[5]See van Benthem (1991).

(263)
$$\begin{array}{lcl}\mathbf{conc} & \stackrel{def}{=} & \mathbf{conc_1} \uplus \mathbf{conc_2} \uplus \mathbf{conc_3} \\ \mathbf{wrap} & \stackrel{def}{=} & \mathbf{wrap_1} \uplus \mathbf{wrap_2}\end{array}$$

The functions from (264) can be encoded with the functions **wrap** and **bridge** as follows

(264)
$$\begin{array}{lcl}\mathbf{conc}_{>}(u,v) & = & \mathbf{conc}(\mathbf{bridge}(u),v) \\ \mathbf{conc}_{<}(u,v) & = & \mathbf{conc}(u,\mathbf{bridge}(v))\end{array}$$

5.2.4 From Lexicalized Head-Grammars to D-grammars

We define a class of **D**-grammars of level 1, which are close to head-grammars.

(265) **Definition** (*Lexicalized Head-Grammars*)

Let $G = (V, N, P, S)$ be a modified head-grammar. We say that G is lexicalized if and only if all the production rules have one of the following types:

- $X \to Y \odot (\alpha,\, \beta)$
- $X \to (\alpha,\, \beta) \odot Y$
- $X \to \hat{}Y \circ (\alpha,\, \beta)$
- $X \to \alpha \circ Y$
- $X \to (\alpha,\, \beta) \circ \hat{}Y$
- $X \to Y \circ \alpha$

We will say that a lexicalized extended Head-Grammar is *simply* lexicalised if and only if α occurrences are a single element a of the alphabet and in the case of pairs of strings $(\alpha,\, \beta)$ have the form (a, ϵ) or (ϵ, a) where $a \in \Sigma$.

Let G be a lexicalized head-grammar. We present a procedure to transform G into a simply lexicalized head-grammar. We supose that $\alpha = a_1 \cdots a_n$ and $\beta = b_1 \cdots b_m$, where $a_i, b_j \in \Sigma$. In case $n = 0$ or $m = 0$ then α or β are both the empty string ϵ.

(266)

- For a production rule $X \to Y \odot (\alpha,\, \beta)$, substitute it by the following productions:

$$\begin{array}{rcl} X & \to & Y \odot (a_1, \epsilon) \odot X_{\alpha,\beta} \\ X_{\alpha,\beta} & \to & X_{\alpha,\beta} \odot (a_2, \epsilon) \\ & & \vdots \\ X_{\alpha,\beta} & \to & X_{\alpha,\beta} \odot (a_m, \epsilon) \\ X_{\alpha,\beta} & \to & X_{\alpha,\beta} \odot (\epsilon, b_m) \\ X_{\alpha,\beta} & \to & X_{\alpha,\beta} \odot (\epsilon, b_{m-1}) \\ & & \vdots \\ X_{\alpha,\beta} & \to & X_{\alpha,\beta} \odot (\epsilon, b_1) \end{array}$$

- For a production rule $X \to (\alpha,\, \beta) \circ \hat{}Y$, substitute it by the following productions:

$$
\begin{array}{rcl}
X & \to & ((a_1, \epsilon) \odot X_{\alpha,\beta}) \circ \hat{\ } Y \\
X_{\alpha,\beta} & \to & (a_2, \epsilon) \odot X_{\alpha,\beta} \\
 & & \vdots \\
X_{\alpha,\beta} & \to & (a_m, \epsilon) \odot X_{\alpha,\beta} \\
X_{\alpha,\beta} & \to & X_{\alpha,\beta} \odot (a_m, \epsilon) \\
X_{\alpha,\beta} & \to & X_{\alpha,\beta} \odot (\epsilon, b_m) \\
X_{\alpha,\beta} & \to & X_{\alpha,\beta} \odot (\epsilon, b_{m-1}) \\
 & & \vdots \\
X_{\alpha,\beta} & \to & X_{\alpha,\beta} \odot (\epsilon, b_1)
\end{array}
$$

For the remaining rules of (264) a similar construction is carried out. If we apply this process to all the productions of G we obtain another one G^* which is weakly equivalent to G. The construction of G^* is such that it is a simply lexicalized head-grammar.

(267) **Theorem** (*Languages Recognized by Lexicalized Head-Grammar*)

Let $\mathbf{G} = (V, N, P, S)$ be a lexicalized head-grammar. There exists a 1-**D** sort 0 lexicon which recognizes the language $L(\mathbf{G}, S)$.

Proof. By the procedure (266) we know there exists a simply lexicalized head-grammar $\mathbf{G}^*$ weakly equivalent to $\mathbf{G}$. From $\mathbf{G}^*$ we define a displacement lexicon $\mathbf{Lex}_{\mathbf{G}^*}$ as follows:

- If $N \to (a, \epsilon)$ is a production of $\mathbf{G}^*$ then $a : \triangleleft^{-1}(N) \in \mathbf{Lex}_{\mathbf{G}^*}$
- If $X \to ((a_1, \epsilon) \odot X_{\alpha,\beta}) \circ \hat{\ } Y$ is a production of $\mathbf{G}^*$ then $a_1 : \triangleleft^{-1}((\hat{\ } Y \backslash X) \uparrow_1 X_{\alpha,\beta}) \in \mathbf{Lex}_{\mathbf{G}^*}$

Other cases from (266) are similar to the ones above.
It is readily seen by induction on the length of the derivations that:

$$L(\mathbf{G}, S) = L(\mathbf{G}^*, S) = L(\mathbf{Lex}_{\mathbf{G}^*}, S)$$

□

Appendix A. Computer-generated output for Dutch verb raising and cross-serial dependencies

The outputs have been generated under a preliminary version of CatLog (Morrill (2011a)).

boeken : N : *books*
cecilia : N : c
de : N/CN : ι
jan : N : j
helpen : $J\backslash((N\backslash Si)\downarrow(N\backslash(N\backslash Si)))$: $\lambda A \lambda B \lambda C \lambda D((help\ D)\ (B\ C))$
henk : N : h
kan : $(N\backslash Si)\downarrow(N\backslash S)$: $\lambda A \lambda B((isable\ B)\ (A\ B))$
kunnen : $J\backslash((N\backslash Si)\downarrow(N\backslash Si))$: $\lambda A \lambda B \lambda C((beable\ C)\ (B\ C))$
las : $N\backslash(N\backslash S)$: *reads*
lezen : $J\backslash(N\backslash(N\backslash Si))$: $\lambda A\, read$
nijlpaarden : CN : *hippos*
voeren : $J\backslash(N\backslash(N\backslash Si))$: $\lambda A\, feed$
wil : $(N\backslash Si)\downarrow(N\backslash S)$: $\lambda A \lambda B((wants\ B)\ (A\ B))$
zag : $(N\backslash Si)\downarrow(N\backslash(N\backslash S))$: $\lambda A \lambda B \lambda C((saw\ C)\ (A\ B))$

(1) **jan+boeken+las** : S

$N : j,\ N : books,\ N\backslash(N\backslash S) : reads \ \Rightarrow\ S$

$$\dfrac{\dfrac{}{N \Rightarrow N} \qquad \dfrac{\dfrac{}{N \Rightarrow N} \quad \dfrac{}{S \Rightarrow S}}{N, N\backslash S \Rightarrow S}\backslash L}{N, N, N\backslash(N\backslash S) \Rightarrow S}\backslash L$$

$((reads\ books)\ j)$

(2) **jan+boeken+kan+lezen** : S

$N : j, N : books, (N\backslash Si)\downarrow(N\backslash S) : \lambda A\lambda B((isable\ B)\ (A\ B)), J\backslash(N\backslash(N\backslash Si)) : \lambda A read \Rightarrow S$

$$\dfrac{\dfrac{\dfrac{\dfrac{}{[\,] \Rightarrow J}JR \qquad \dfrac{\dfrac{}{N \Rightarrow N} \quad \dfrac{\dfrac{}{N \Rightarrow N} \quad \dfrac{}{Si\{[\,]\} \Rightarrow Si}}{N, N\backslash Si\{[\,]\} \Rightarrow Si}\backslash L}{N, N, N\backslash(N\backslash Si)\{[\,]\} \Rightarrow Si}\backslash L}{N, N, [\,], J\backslash(N\backslash(N\backslash Si)) \Rightarrow Si}\backslash L}{N, [\,], J\backslash(N\backslash(N\backslash Si)) \Rightarrow N\backslash Si}\backslash R \qquad \dfrac{\dfrac{}{N \Rightarrow N} \quad \dfrac{}{S \Rightarrow S}}{N, N\backslash S \Rightarrow S}\backslash L}{N, N, (N\backslash Si)\downarrow(N\backslash S), J\backslash(N\backslash(N\backslash Si)) \Rightarrow S}\downarrow L$$

$((isable\ j)\ ((read\ books)\ j))$

(3) **jan+boeken+wil+kunnen+lezen** : S

$N : j, N : books, (N\backslash Si)\downarrow(N\backslash S) : \lambda A\lambda B((wants\ B)\ (A\ B)), J\backslash((N\backslash Si)\downarrow(N\backslash Si)) : \lambda A\lambda B\lambda C((beable\ C)\ (B\ C)), J\backslash(N\backslash(N\backslash Si)) : \lambda A read \Rightarrow S$

$$\dfrac{\dfrac{\dfrac{}{[\,]\Rightarrow J}JR \quad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{[\,]\Rightarrow J}JR \quad \dfrac{\dfrac{}{N\Rightarrow N} \quad \dfrac{\dfrac{}{N\Rightarrow N} \quad \dfrac{}{Si\{[\,]\}\Rightarrow Si}}{N, N\backslash Si\{[\,]\}\Rightarrow Si}\backslash L}{N, N, N\backslash(N\backslash Si)\{[\,]\}\Rightarrow Si}\backslash L}{N, N, [\,], J\backslash(N\backslash(N\backslash Si))\Rightarrow Si}\backslash L}{N, [\,], J\backslash(N\backslash(N\backslash Si))\Rightarrow N\backslash Si}\backslash R \quad \dfrac{\dfrac{}{N\Rightarrow N} \quad \dfrac{}{Si\{[\,]\}\Rightarrow Si}}{N, N\backslash Si\{[\,]\}\Rightarrow Si}\backslash L}{N, N, (N\backslash Si)\downarrow(N\backslash Si)\{[\,]\}, J\backslash(N\backslash(N\backslash Si))\Rightarrow Si}\downarrow L}{N, N, [\,], J\backslash((N\backslash Si)\downarrow(N\backslash Si)), J\backslash(N\backslash(N\backslash Si))\Rightarrow Si}\backslash L}{N, [\,], J\backslash((N\backslash Si)\downarrow(N\backslash Si)), J\backslash(N\backslash(N\backslash Si))\Rightarrow N\backslash Si}\backslash R \quad \dfrac{\dfrac{}{N\Rightarrow N} \quad \dfrac{}{S\Rightarrow S}}{N, N\backslash S\Rightarrow S}}{N, N, (N\backslash Si)\downarrow(N\backslash S), J\backslash((N\backslash Si)\downarrow(N\backslash Si)), J\backslash(N\backslash(N\backslash Si))\Rightarrow S}\downarrow L$$

$((wants\ j)\ ((beable\ j)\ ((read\ books)\ j)))$

(4) **jan+cecilia+henk+de+nijlpaarden+zag+helpen+voeren** : S

$N : j,\ N : c,\ N : h,\ N/CN : \iota,\ CN : hippos,\ (N\backslash Si)\downarrow(N\backslash(N\backslash S)) : \lambda A\lambda B\lambda C((saw\ C)\ (A\ B)),\ J\backslash((N\backslash Si)\downarrow(N\backslash(N\backslash Si))) : \lambda A\lambda B\lambda C\lambda D((help\ D)\ (B\ C)),\ J\backslash(N\backslash(N\backslash Si)) : \lambda A feed\ \Rightarrow\ S$

$$\dfrac{\dfrac{\dfrac{\dfrac{}{CN\Rightarrow CN} \quad \dfrac{\dfrac{}{[\,]\Rightarrow J}JR \quad \dfrac{\dfrac{\dfrac{\dfrac{}{[\,]\Rightarrow J}JR \quad \dfrac{\dfrac{}{N\Rightarrow N} \quad \dfrac{\dfrac{}{N\Rightarrow N} \quad \dfrac{}{Si\{[\,]\}\Rightarrow Si}}{N, N\backslash Si\{[\,]\}\Rightarrow Si}\backslash L}{N, N, N\backslash(N\backslash Si)\{[\,]\}\Rightarrow Si}\backslash L}{N, N, [\,], J\backslash(N\backslash(N\backslash Si))\Rightarrow Si}\backslash L}{N, [\,], J\backslash(N\backslash(N\backslash Si))\Rightarrow N\backslash Si}\backslash R \quad \dfrac{\dfrac{}{N\Rightarrow N} \quad \dfrac{\dfrac{}{N\Rightarrow N} \quad \dfrac{}{Si\{[\,]\}\Rightarrow Si}}{N, N\backslash Si\{[\,]\}\Rightarrow Si}\backslash L}{N, N, N\backslash(N\backslash Si)\{[\,]\}\Rightarrow Si}\backslash L}{N, N, N, (N\backslash Si)\downarrow(N\backslash(N\backslash Si))\{[\,]\}, J\backslash(N\backslash(N\backslash Si))\Rightarrow Si}\downarrow L}{N, N, N, [\,], J\backslash((N\backslash Si)\downarrow(N\backslash(N\backslash Si))), J\backslash(N\backslash(N\backslash Si))\Rightarrow Si}\backslash L}{N, N, N/CN, CN, [\,], J\backslash((N\backslash Si)\downarrow(N\backslash(N\backslash Si))), J\backslash(N\backslash(N\backslash Si))\Rightarrow Si}/L}{N, N/CN, CN, [\,], J\backslash((N\backslash Si)\downarrow(N\backslash(N\backslash Si))), J\backslash(N\backslash(N\backslash Si))\Rightarrow N\backslash Si}\backslash R \quad \dfrac{\dfrac{}{N\Rightarrow N} \quad \dfrac{\dfrac{}{N\Rightarrow N} \quad \dfrac{}{S\Rightarrow S}}{N, N\backslash S\Rightarrow S}}{N, N, N\backslash(N\backslash S)\Rightarrow S}\backslash L}{N, N, N, N/CN, CN, (N\backslash Si)\downarrow(N\backslash(N\backslash S)), J\backslash((N\backslash Si)\downarrow(N\backslash(N\backslash Si))), J\backslash(N\backslash(N\backslash Si))\Rightarrow S}\downarrow L$$

$((saw\ j)\ ((help\ c)\ ((feed\ (\iota\ hippos))\ h)))$

Appendix B. Computer generated derivation of *accbab* in *Mix*

$$\frac{}{a \Rightarrow a} \qquad \frac{a \Rightarrow a}{a, I \Rightarrow a}\, IL \qquad \frac{a, I \Rightarrow a}{a, I, I \Rightarrow a}\, IL \qquad \frac{a, I, I \Rightarrow a}{a, I, I, I \Rightarrow a}\, IL$$

$$\frac{a, I, I, I \Rightarrow a}{a, I, I, [\,] \Rightarrow a{\uparrow}I}\, {\uparrow}R \qquad \frac{}{b \Rightarrow b}$$

$$\frac{a, I, I, [\,] \Rightarrow a{\uparrow}I \qquad b \Rightarrow b}{a, I, I, (a{\uparrow}I){\downarrow}b \Rightarrow b}\, {\downarrow}L \qquad \frac{a, I, I, (a{\uparrow}I){\downarrow}b \Rightarrow b}{a, I, [\,], (a{\uparrow}I){\downarrow}b \Rightarrow b{\uparrow}I}\, {\uparrow}R$$

$$\frac{}{S \Rightarrow S} \qquad \frac{S \Rightarrow S}{S, I \Rightarrow S}\, IL \qquad \frac{S, I \Rightarrow S}{S, [\,] \Rightarrow S{\uparrow}I}\, {\uparrow}R$$

$$\frac{}{a \Rightarrow a} \qquad \frac{a \Rightarrow a}{a, I \Rightarrow a}\, IL \qquad \frac{a, I \Rightarrow a}{a, [\,] \Rightarrow a{\uparrow}I}\, {\uparrow}R \qquad \frac{}{b \Rightarrow b} \qquad \frac{a, [\,] \Rightarrow a{\uparrow}I \qquad b \Rightarrow b}{a, (a{\uparrow}I){\downarrow}b \Rightarrow b}\, {\downarrow}L$$

$$\frac{S, [\,] \Rightarrow S{\uparrow}I \qquad a, (a{\uparrow}I){\downarrow}b \Rightarrow b}{S, (S{\uparrow}I){\downarrow}a, (a{\uparrow}I){\downarrow}b \Rightarrow b}\, {\downarrow}L$$

$$\frac{a, I, [\,], (a{\uparrow}I){\downarrow}b \Rightarrow b{\uparrow}I \qquad S, (S{\uparrow}I){\downarrow}a, (a{\uparrow}I){\downarrow}b \Rightarrow b}{a, I, (b{\uparrow}I){\downarrow}S, (a{\uparrow}I){\downarrow}b, (S{\uparrow}I){\downarrow}a, (a{\uparrow}I){\downarrow}b \Rightarrow b}\, {\downarrow}L$$

$$\frac{a, I, (b{\uparrow}I){\downarrow}S, (a{\uparrow}I){\downarrow}b, (S{\uparrow}I){\downarrow}a, (a{\uparrow}I){\downarrow}b \Rightarrow b}{a, [\,], (b{\uparrow}I){\downarrow}S, (a{\uparrow}I){\downarrow}b, (S{\uparrow}I){\downarrow}a, (a{\uparrow}I){\downarrow}b \Rightarrow b{\uparrow}I}\, {\uparrow}R \qquad \frac{}{S \Rightarrow S}$$

$$\frac{a, [\,], (b{\uparrow}I){\downarrow}S, (a{\uparrow}I){\downarrow}b, (S{\uparrow}I){\downarrow}a, (a{\uparrow}I){\downarrow}b \Rightarrow b{\uparrow}I \qquad S \Rightarrow S}{a, (b{\uparrow}I){\downarrow}S, (b{\uparrow}I){\downarrow}S, (a{\uparrow}I){\downarrow}b, (S{\uparrow}I){\downarrow}a, (a{\uparrow}I){\downarrow}b \Rightarrow S}\, {\downarrow}L$$

Chapter 6

Linguistic Applications

This chapter addresses linguistic issues which are discontinuous in nature. The discontinuous Lambek calculus **D** and its extensions, which were studied in previous chapters, can be seen now at work.[1] The slogan *grammaticality = provability* is fully applied by working linguistic examples in hypersequent calculus and in labelled natural deduction. The sources of these examples can be found in Morrill, Valentín, and Fadda (2011), Morrill and Valentín (2010a) and in Morrill and Valentín (2011).

There are several linguistic phenomena we account for, such as discontinuous idioms, quantification, VP ellipsis, medial-extraction, pied-piping, appositive relativization, parentheticals, gapping, comparative subdeletion, (a first analysis of) English reflexivization, dative alternation and particle shift.

Moreover, we analyze in some detail binding theory. English binding theory was studied in Morrill and Valentín (2010c) and Morrill and Valentín (2011). In these two last references there were considered several extensions[2] of **D**, which are presented in detail in this thesis. With respect to Catalan (and in fact in some other Romance languages), reflexive binding theory is also analyzed.[3] We show and formally account for some remarkable puzzling linguistic differences which exist between the so-called reflexive clitic and the reflexive syntactic anaphor (cf. Alsina (1996)). Finally, Catalan reflexive clitic climbing is addressed.

A relevant feature of all the linguistic analyses which are carried out here, is that no structural rule is used.

6.1 Linguistic Applications of $\mathbf{D_b}$

By Basic discontinuous Lambek calculus $\mathbf{D_b}$ we mean **D** with level 1, atomicity 0, and in which the displacement algebra algebra is restricted to just $+ : L_0 \times L_0 \to L_0$ and $\times : L_1 \times L_0 \to L_0$. Therefore the only discontinuous connectives it contains are sort non-polymorphic $\downarrow$, $\odot$ and $\uparrow$. In this section we list accounts of linguistic phenomena falling within the scope of this minimal discontinuity

[1] This chapter is based on Morrill, Valentín, and Fadda (2011).

[2] The extensions we refer to are different from the extensions considered in this thesis, in this case **DADND**.

[3] This work on Catalan binding theory is new, and consequently it does not appear in the works cited.

calculus. Notice that $\mathbf{D_b}$ is a strictly proper subcalculus of 1-**D** with atomicity 0.

Discontinuous Idioms

Idioms are complex expressions which have a meaning not compositionally attributable to the meanings of their parts (e.g. *red herring*). In grammar delivering logical semantics, they must be listed in the lexicon, because there is no other place from which their meaning can come. In discontinuous idioms, the idiomatic material is interpolated by non-idiomatic dependents, for example:

(268) Mary *gave* John/the man/... *the cold shoulder*.

Let there be the following lexical assignment:

(269) $\textbf{gave}+1+\textbf{the}+\textbf{cold}+\textbf{shoulder} : (\mathrm{N}\backslash\mathrm{S})\uparrow\mathrm{N} : \mathit{shun}$

Then our example is derived as follows in the hypersequent calculus and the labelled natural deduction calculus respectively:

(270)
$$\dfrac{\mathrm{N}\Rightarrow\mathrm{N} \qquad \dfrac{\mathrm{N}\Rightarrow\mathrm{N} \qquad \mathrm{S}\Rightarrow\mathrm{S}}{\mathrm{N},\mathrm{N}\backslash\mathrm{S}\Rightarrow\mathrm{S}}\backslash L}{\mathrm{N},(\mathrm{N}\backslash\mathrm{S})\uparrow\mathrm{N}\{\mathrm{N}\}\Rightarrow\mathrm{S}}\uparrow L$$

(271)
$$\dfrac{\dfrac{\text{Mary}}{\textbf{Mary} : N : m} \qquad \dfrac{\dfrac{\text{gave ... the cold shoulder}}{\textbf{gave}+1+\textbf{the}+\textbf{cold}+\textbf{shoulder} : (\mathrm{N}\backslash\mathrm{S})\uparrow\mathrm{N} : \mathit{shun}} \qquad \dfrac{\text{John}}{\textbf{John} : N : j}}{\textbf{gave}+\textbf{John}+\textbf{the}+\textbf{cold}+\textbf{shoulder} : \mathrm{N}\backslash\mathrm{S} : (\mathit{shun}\ j)}E\uparrow}{\textbf{Mary}+\textbf{gave}+\textbf{John}+\textbf{the}+\textbf{cold}+\textbf{shoulder} : \mathrm{S} : (\mathit{shun}\ j\ m)}E\backslash$$

Quantification

Quantification is a classical instance of discontinuity, i.e. syntactic-semantic mismatch. Quantifier phrases occupy nominal positions syntactically but take sentential scope semantically, for example:

(272) a. John gave every book to Mary.

b. $\forall x[(\mathit{book}\ x) \to (\mathit{give}\ m\ x\ j)]$

We treat quantification by type assignments such as the following:

(273) $\textbf{every} : ((\mathrm{S}\uparrow\mathrm{N})\downarrow\mathrm{S})/\mathrm{CN} : \lambda x\lambda y\forall z[(x\ z) \to (y\ z)]$

Such a composite of extraction and infixation to treat quantification was suggested in Moortgat (1991), but he did not have a calculus ensuring that the extraction and infixation points would be one and the same.

An example like (272a) is derived (with the right semantics) as follows, where PTV abbreviates (N\S)/(N•PP) or ((N\S)/PP)/N:

(274)
$$\frac{\mathrm{CN} \Rightarrow \mathrm{CN} \qquad \dfrac{\dfrac{\mathrm{N, PTV, N, PP} \Rightarrow \mathrm{S}}{\mathrm{N, PTV, [], PP} \Rightarrow \mathrm{S}{\uparrow}\mathrm{N}}\uparrow R \qquad \mathrm{S} \Rightarrow \mathrm{S}}{\mathrm{N, PTV, (S{\uparrow}N){\downarrow}S, PP} \Rightarrow \mathrm{S}}\downarrow L}{\mathrm{N, PTV, ((S{\uparrow}N){\downarrow}S)/CN, CN, PP} \Rightarrow \mathrm{S}}/L$$

Montague (1973) presumably takes its title from its treatment of quantifiers and it is interesting to compare our treatment with his rule of term-insertion S14. Ignoring for the moment pronoun-binding aspects, S14 replaces by a noun phrase a variable in a nominal position in a sentence and semantically applies the noun phrase to the lambda abstraction of the sentence meaning over that of the nominal position. Our analysis splits such a step into two parts: conditionalization of the sentence over the nominal, semantically interpreted by functional abstraction over the nominal meaning, and infixing of the quantifier phrase into the conditionalized sentence, semantically interpreted by functional application of the infix to the circumfix.

Like that of Montague, our account allows quantifier phrases to take scope at the level of any embedding sentence, a feature which must eventually be constrained. However this successfully characterises the de re/specific and de dicto/nonspecific ambiguity of (275).

(275) Mary thinks someone left.

The de dicto reading, where the propositional attitude verb has wider scope than the existential quantifier (Mary does not necessarily have a particular person in mind), is generated by:

(276)
$$\frac{\dfrac{\dfrac{\mathrm{N, N\backslash S} \Rightarrow \mathrm{S}}{\mathrm{[], N\backslash S} \Rightarrow \mathrm{S{\uparrow}N}}\uparrow R \qquad \mathrm{S} \Rightarrow \mathrm{S}}{\mathrm{(S{\uparrow}N){\downarrow}S, N\backslash S} \Rightarrow \mathrm{S}}\downarrow L \qquad \mathrm{N, N\backslash S} \Rightarrow \mathrm{S}}{\mathrm{N, (N\backslash S)/S, (S{\uparrow}N){\downarrow}S, N\backslash S} \Rightarrow \mathrm{S}}/L$$

The de re reading, where the existential quantifier has wider scope than the propositional attitude verb (Mary has a particular person in mind), is generated by:

(277)
$$\frac{\dfrac{\mathrm{N, (N\backslash S)/S, N, N\backslash S} \Rightarrow \mathrm{S}}{\mathrm{N, (N\backslash S)/S, [], N\backslash S} \Rightarrow \mathrm{S{\uparrow}N}}\uparrow R \qquad \mathrm{S} \Rightarrow \mathrm{S}}{\mathrm{N, (N\backslash S)/S, (S{\uparrow}N){\downarrow}S, N\backslash S} \Rightarrow \mathrm{S}}\downarrow L$$

Also like the account of Montague, ours allows multiple quantifiers to scope in any order, another feature which must eventually be constrained (for example, *each* appears to always take wider scope). But this successfully characterizes the classical example of ambiguity:

(278) Everyone loves someone.

On the (dominant) subject wide scope reading, different people love, in general, different people (as in when we all love our respective mothers). On the (subordinate) object wide scope reading, different people love the same person (as in when we all love one and the same queen). The subject wide scope (∀∃) reading is generated by:

$$(279)\quad \dfrac{\dfrac{\dfrac{\dfrac{\mathrm{N}, (\mathrm{N}\backslash \mathrm{S})/\mathrm{N}, \mathrm{N} \Rightarrow \mathrm{S}}{\mathrm{N}, (\mathrm{N}\backslash \mathrm{S})/\mathrm{N}, [] \Rightarrow \mathrm{S}{\uparrow}\mathrm{N}}{\uparrow}R \quad \mathrm{S} \Rightarrow \mathrm{S}}{\mathrm{N}, (\mathrm{N}\backslash \mathrm{S})/\mathrm{N}, (\mathrm{S}{\uparrow}\mathrm{N}){\downarrow}\mathrm{S} \Rightarrow \mathrm{S}}{\downarrow}L}{[], (\mathrm{N}\backslash \mathrm{S})/\mathrm{N}, (\mathrm{S}{\uparrow}\mathrm{N}){\downarrow}\mathrm{S} \Rightarrow \mathrm{S}{\uparrow}\mathrm{N}}{\uparrow}R \quad \mathrm{S} \Rightarrow \mathrm{S}}{(\mathrm{S}{\uparrow}\mathrm{N}){\downarrow}\mathrm{S}, (\mathrm{N}\backslash \mathrm{S})/\mathrm{N}, (\mathrm{S}{\uparrow}\mathrm{N}){\downarrow}\mathrm{S} \Rightarrow \mathrm{S}}{\downarrow}L$$

The object wide scope ($\exists\forall$) reading is generated by:

$$(280)\quad \dfrac{\dfrac{\dfrac{\dfrac{\mathrm{N}, (\mathrm{N}\backslash \mathrm{S})/\mathrm{N}, \mathrm{N} \Rightarrow \mathrm{S}}{[], (\mathrm{N}\backslash \mathrm{S})/\mathrm{N}, \mathrm{N} \Rightarrow \mathrm{S}{\uparrow}\mathrm{N}}{\uparrow}R \quad \mathrm{S} \Rightarrow \mathrm{S}}{(\mathrm{S}{\uparrow}\mathrm{N}){\downarrow}\mathrm{S}, (\mathrm{N}\backslash \mathrm{S})/\mathrm{N}, \mathrm{N} \Rightarrow \mathrm{S}}{\downarrow}L}{(\mathrm{S}{\uparrow}\mathrm{N}){\downarrow}\mathrm{S}, (\mathrm{N}\backslash \mathrm{S})/\mathrm{N}, [] \Rightarrow \mathrm{S}{\uparrow}\mathrm{N}}{\uparrow}R \quad \mathrm{S} \Rightarrow \mathrm{S}}{(\mathrm{S}{\uparrow}\mathrm{N}){\downarrow}\mathrm{S}, (\mathrm{N}\backslash \mathrm{S})/\mathrm{N}, (\mathrm{S}{\uparrow}\mathrm{N}){\downarrow}\mathrm{S} \Rightarrow \mathrm{S}}{\downarrow}L$$

(The sooner processed, i.e. the nearer the root of the sequent proof, the wider the scope of the quantifier). Note that even assuming nondeterministic wrapping, in our account multiple quantifiers cannot get tangled up and bind each others' positions because the types driving the derivation ensure that the quantifier separator positions are only opened up and closed off one at a time, so that the only positions ever available are the unique correct ones.

For an account of the preference for $\forall\exists$ scope over $\exists\forall$ scope (i.e. left-to-right quantifier scope preference), see Morrill (2000b), which defines a complexity metric on analyses expressed as proof nets (Girard (1987)) motivated by the incrementality of processing. A range of other performance phenomena are also accounted for there in the same way.

6.1.1 Linguistic Applications of 1-D

By **1-D** we mean **D** with level 1. Connectives are now sort-polymorphic, but only one single separator is found in its interpretation in a displacement algebra. The atomicity of **1-D** equals the level of **1-D**, i.e. it is equal to 1. 1-**D** allows several linguistic applications. Let us see them.

VP Ellipsis

VP ellipsis refers to a class of constructions in which a form of *do* (perhaps suffixed by *too*) takes its interpretation from a preceding verb phrase, for example:

(281) a. John slept before Mary did.

b. John slept and Mary did too.

Let there be the following lexical type assignment to the auxiliary, where VP abbreviates N\S:

(282) **did** : $((\mathrm{VP}{\uparrow}\mathrm{VP})/\mathrm{VP})\backslash(\mathrm{VP}{\uparrow}\mathrm{VP}) : \lambda x\lambda y(x\ y\ y)$

Then an example such as (281a) is derived as follows:

(283)
$$\dfrac{\dfrac{\dfrac{\mathrm{VP}, (\mathrm{VP}\backslash\mathrm{VP})/\mathrm{S}, \mathrm{N}, \mathrm{VP} \Rightarrow \mathrm{VP}}{[\,], (\mathrm{VP}\backslash\mathrm{VP})/\mathrm{S}, \mathrm{N}, \mathrm{VP} \Rightarrow \mathrm{VP}{\uparrow}\mathrm{VP}}\uparrow R}{[\,], (\mathrm{VP}\backslash\mathrm{VP})/\mathrm{S}, \mathrm{N} \Rightarrow (\mathrm{VP}{\uparrow}\mathrm{VP})/\mathrm{VP}}/R \qquad \dfrac{\mathrm{VP} \Rightarrow \mathrm{VP} \qquad \mathrm{N}, \mathrm{VP} \Rightarrow \mathrm{S}}{\mathrm{N}, \mathrm{VP}{\uparrow}\mathrm{VP}\{\mathrm{VP}\}, \Rightarrow \mathrm{S}}\uparrow L}{\mathrm{N}, \mathrm{VP}, (\mathrm{VP}\backslash\mathrm{VP})/\mathrm{S}, \mathrm{N}, ((\mathrm{VP}{\uparrow}\mathrm{VP})/\mathrm{VP})\backslash(\mathrm{VP}{\uparrow}\mathrm{VP}) \Rightarrow \mathrm{S}}\backslash L$$

VP ellipsis can also occur intersententially, so an account must eventually be set up at the level of discourse.

Medial Extraction

Extraction in which the gap is not at the periphery such as

(284) dog that Mary saw today

can be modelled as follows:

(285) **that** : $(\mathrm{CN}\backslash\mathrm{CN})/\hat{}(\mathrm{S}{\uparrow}\mathrm{N}) : \lambda x\lambda y\lambda z[(x\ z) \wedge (y\ z)]$

The example (284) is derived thus in the hypersequent calculus:

(286)
$$\dfrac{\dfrac{\dfrac{\mathrm{N}, (\mathrm{N}\backslash\mathrm{S})/\mathrm{N}, \mathrm{N}, (\mathrm{N}\backslash\mathrm{S})\backslash(\mathrm{N}\backslash\mathrm{S}) \Rightarrow \mathrm{S}}{\mathrm{N}, (\mathrm{N}\backslash\mathrm{S})/\mathrm{N}, [\,], (\mathrm{N}\backslash\mathrm{S})\backslash(\mathrm{N}\backslash\mathrm{S}) \Rightarrow \mathrm{S}{\uparrow}\mathrm{N}}\uparrow R}{\mathrm{N}, (\mathrm{N}\backslash\mathrm{S})/\mathrm{N}, (\mathrm{N}\backslash\mathrm{S})\backslash(\mathrm{N}\backslash\mathrm{S}) \Rightarrow \hat{}(\mathrm{S}{\uparrow}\mathrm{N})}\hat{}R \qquad \mathrm{CN}, \mathrm{CN}\backslash\mathrm{CN} \Rightarrow \mathrm{CN}}{\mathrm{CN}, (\mathrm{CN}\backslash\mathrm{CN})/\hat{}(\mathrm{S}{\uparrow}\mathrm{N}), \mathrm{N}, (\mathrm{N}\backslash\mathrm{S})/\mathrm{N}, (\mathrm{N}\backslash\mathrm{S})\backslash(\mathrm{N}\backslash\mathrm{S}) \Rightarrow \mathrm{CN}}/L$$

The derivation in labelled natural deduction is as shown in Figure 6.1.

Pied-Piping

Pied-piping is the embedding of a filler such as a relative pronoun within accompanying material from the extraction site:

(287) scene the painting of which by Cezanne John sold for \$10,000,000

The depth of embedding is unbounded:

(288) thesis the height of the lettering on the first line of the second page of the third chapter of ... of which is 0.5cm

Pied-piping can be treated by assignment as follows (cf. Morrill (1994) (ch. 4); Morrill (1995)):

(289) **which** : $(\mathrm{N}{\uparrow}\mathrm{N}){\downarrow}((\mathrm{CN}\backslash\mathrm{CN})/\hat{}(\mathrm{S}{\uparrow}\mathrm{N})) : \lambda x\lambda y\lambda z\lambda w[(z\ w) \wedge (y\ (x\ w))]$

Then (287) is derived as shown in Figure 6.2, where PTV abbreviates $(\mathrm{N}\backslash\mathrm{S})/(\mathrm{N}\bullet\mathrm{PP})$. Note that (289) can also generate relativisation in which there is no pied-piping by deriving an empty pied-piping context as $\mathrm{N}{\uparrow}\mathrm{N}$ ($[\,] \Rightarrow \mathrm{N}{\uparrow}\mathrm{N}$ is a theorem): once the assignment (289) is included, that of *medial extraction* is no longer required: the assignment (285) is derivable from, and so subsumed by, (289). The reader should notice that all these derivations of pied-piping are allowed because **D** is defined model-theoretically in displacement algebras which are based in monoids. Hence the usual restriction of non-empty antecedent in left/right slashes rules of the Lambek calculus as formulated in Lambek (1958) are no longer required.

$$\dfrac{\text{saw}}{\textbf{saw}-\mathit{see}\colon (\mathrm{N}\backslash \mathrm{S})/\mathrm{N}} \qquad \dfrac{}{a-x\colon \mathrm{N}}\; i$$

$$\dfrac{\textbf{saw}-\mathit{see}\colon (\mathrm{N}\backslash \mathrm{S})/\mathrm{N} \qquad a-x\colon \mathrm{N}}{\textbf{saw}+a-(\mathit{see}\ x)\colon \mathrm{N}\backslash \mathrm{S}}\; E/ \qquad \dfrac{\text{today}}{\textbf{today}-\mathit{today}\colon (\mathrm{N}\backslash \mathrm{S})\backslash(\mathrm{N}\backslash \mathrm{S})}$$

$$\dfrac{\text{Mary}}{\textbf{Mary}-m\colon \mathrm{N}} \qquad \dfrac{\textbf{saw}+a-(\mathit{see}\ x)\colon \mathrm{N}\backslash \mathrm{S} \qquad \textbf{today}-\mathit{today}\colon (\mathrm{N}\backslash \mathrm{S})\backslash(\mathrm{N}\backslash \mathrm{S})}{\textbf{saw}+a+\textbf{today}-(\mathit{today}\ (\mathit{see}\ x))\colon \mathrm{N}\backslash \mathrm{S}}$$

$$\dfrac{\textbf{Mary}-m\colon \mathrm{N} \qquad \textbf{saw}+a+\textbf{today}-(\mathit{today}\ (\mathit{see}\ x))\colon \mathrm{N}\backslash \mathrm{S}}{\textbf{Mary}+\textbf{saw}+a+\textbf{today}-(\mathit{today}\ (\mathit{see}\ x)\ m)\colon \mathrm{S}}\; E\backslash$$

$$\dfrac{\textbf{Mary}+\textbf{saw}+a+\textbf{today}-(\mathit{today}\ (\mathit{see}\ x)\ m)\colon \mathrm{S}}{\textbf{Mary}+\textbf{saw}+1+\textbf{today}-\lambda x(\mathit{today}\ (\mathit{see}\ x)\ m)\colon \mathrm{S}{\uparrow}\mathrm{N}}\; I{\uparrow}^{i}$$

$$\dfrac{\textbf{Mary}+\textbf{saw}+1+\textbf{today}-\lambda x(\mathit{today}\ (\mathit{see}\ x)\ m)\colon \mathrm{S}{\uparrow}\mathrm{N}}{\textbf{Mary}+\textbf{saw}+\textbf{today}-\lambda x(\mathit{today}\ (\mathit{see}\ x)\ m)\colon \hat{}(\mathrm{S}{\uparrow}\mathrm{N})}\; I\hat{}$$

$$\dfrac{\text{that}}{\textbf{that}-\lambda x\lambda y\lambda z[(x\ z)\wedge(y\ z)]\colon (\mathrm{CN}\backslash \mathrm{CN})/\hat{}(\mathrm{S}{\uparrow}\mathrm{N})}$$

$$\dfrac{\textbf{that}-\lambda x\lambda y\lambda z[(x\ z)\wedge(y\ z)]\colon (\mathrm{CN}\backslash \mathrm{CN})/\hat{}(\mathrm{S}{\uparrow}\mathrm{N}) \qquad \textbf{Mary}+\textbf{saw}+\textbf{today}-\lambda x(\mathit{today}\ (\mathit{see}\ x)\ m)\colon \hat{}(\mathrm{S}{\uparrow}\mathrm{N})}{\textbf{that}+\textbf{Mary}+\textbf{saw}+\textbf{today}-\lambda y\lambda z[(\mathit{today}\ (\mathit{see}\ z)\ m)\wedge(y\ z)]\colon \mathrm{CN}\backslash \mathrm{CN}}\; E/$$

$$\dfrac{\text{dog}}{\textbf{dog}-\mathit{dog}\colon \mathrm{CN}}$$

$$\dfrac{\textbf{dog}-\mathit{dog}\colon \mathrm{CN} \qquad \textbf{that}+\textbf{Mary}+\textbf{saw}+\textbf{today}-\lambda y\lambda z[(\mathit{today}\ (\mathit{see}\ z)\ m)\wedge(y\ z)]\colon \mathrm{CN}\backslash \mathrm{CN}}{\textbf{dog}+\textbf{that}+\textbf{Mary}+\textbf{saw}+\textbf{today}-\lambda z[(\mathit{today}\ (\mathit{see}\ z)\ m)\wedge(\mathit{dog}\ z)]\colon \mathrm{CN}}\; E\backslash$$

Figure 6.1: Labelled natural deduction derivation of medial extraction (284)

$$\dfrac{\dfrac{\mathrm{N/CN, CN, N, CN\backslash CN} \Rightarrow \mathrm{N}}{\mathrm{N/CN, CN/PP, PP/N, [\,], CN\backslash CN} \Rightarrow \mathrm{N{\uparrow}N}}\uparrow R \qquad \dfrac{\dfrac{\dfrac{\mathrm{N, PTV, N, PP} \Rightarrow \mathrm{S}}{\mathrm{N, PTV, [\,], PP} \Rightarrow \mathrm{S{\uparrow}N}}\uparrow R}{\mathrm{N, PTV, PP} \Rightarrow\, \hat{}(\mathrm{S{\uparrow}N})}\hat{}R \qquad \mathrm{CN, CN\backslash CN} \Rightarrow \mathrm{CN}}{\mathrm{CN, (CN\backslash CN)/}\hat{}(\mathrm{S{\uparrow}N}), \mathrm{N, PTV, PP} \Rightarrow \mathrm{CN}}/L}{\mathrm{CN, N/CN, CN/PP, PP/N, (N{\uparrow}N){\downarrow}((CN\backslash CN)/}\hat{}(\mathrm{S{\uparrow}N})), \mathrm{N, PTV, PP} \Rightarrow \mathrm{CN}}\downarrow L$$

Figure 6.2: Hypersequent derivation of pied-piping (287)

Appositive Relativisation

Apposative ('nonrestrictive') relativisation is relativisation in which the relative clause forms a lowered intonational phrase marked off by commas in writing, and modifies a noun phrase:

(290) John, who jogs, sneezed.

Semantically, the predication of the body of the appositive relative clause to the noun phrase modified is conjoined with the semantics of the embedding sentence in which the noun phrase is (also) understood. This discontinuity can be treated by the following assignment:

(291) **which** : $(\mathrm{N}\backslash((\mathrm{S}{\uparrow}\mathrm{N}){\downarrow}\mathrm{S}))/\hat{}(\mathrm{S}{\uparrow}\mathrm{N}) : \lambda x\lambda y\lambda z[(x\ y) \wedge (z\ y)]$

Our example (290) is derived as follows:

(292)
$$\dfrac{\dfrac{\dfrac{\mathrm{N}, \mathrm{N}\backslash\mathrm{S} \Rightarrow \mathrm{S}}{[\,], \mathrm{N}\backslash\mathrm{S} \Rightarrow \mathrm{S}{\uparrow}\mathrm{N}}\,{\uparrow}R}{\mathrm{N}\backslash\mathrm{S} \Rightarrow \hat{}(\mathrm{S}{\uparrow}\mathrm{N})}\,\hat{}R \qquad \dfrac{N \Rightarrow N \qquad \dfrac{\dfrac{\mathrm{N}, \mathrm{N}\backslash\mathrm{S} \Rightarrow \mathrm{S}}{[\,], \mathrm{N}\backslash\mathrm{S} \Rightarrow \mathrm{S}{\uparrow}\mathrm{N}}\,{\uparrow}R \qquad \mathrm{S} \Rightarrow \mathrm{S}}{(\mathrm{S}{\uparrow}\mathrm{N}){\downarrow}\mathrm{S}, \mathrm{N}\backslash\mathrm{S} \Rightarrow \mathrm{S}}\,{\downarrow}L}{\mathrm{N}, \mathrm{N}\backslash((\mathrm{S}{\uparrow}\mathrm{N}){\downarrow}\mathrm{S}), \mathrm{N}\backslash\mathrm{S} \Rightarrow \mathrm{S}}\,\backslash L}{\mathrm{N}, (\mathrm{N}\backslash((\mathrm{S}{\uparrow}\mathrm{N}){\downarrow}\mathrm{S}))/\hat{}(\mathrm{S}{\uparrow}\mathrm{N}), \mathrm{N}\backslash\mathrm{S}, \mathrm{N}\backslash\mathrm{S} \Rightarrow \mathrm{S}}\,/L$$

Parentheticals

Parentheticals are adsentential modifiers such as *fortunately* which, to a very rough first approximation, can appear anywhere in the sentence they modify:[4]

(293) a. Fortunately, John has perseverance.

b. John, fortunately, has perseverance.

c. John has, fortunately, perseverance.

d. John has perseverance, fortunately.

Such a distribution is captured by the following type assignment, as in Morrill and Merenciano (1996):

(294) **fortunately** : $\check{}\mathrm{S}{\downarrow}\mathrm{S}$: *fortunately*

For example, (293c) is derived as follows in the hypersequent calculus:

(295)
$$\dfrac{\dfrac{\mathrm{N}, (\mathrm{N}\backslash\mathrm{S})/\mathrm{N}, \mathrm{N} \Rightarrow \mathrm{S}}{\mathrm{N}, (\mathrm{N}\backslash\mathrm{S})/\mathrm{N}, [\,], \mathrm{N} \Rightarrow \check{}\mathrm{S}}\,\check{}R \qquad \mathrm{S} \Rightarrow \mathrm{S}}{\mathrm{N}, (\mathrm{N}\backslash\mathrm{S})/\mathrm{N}, \check{}S{\downarrow}S, \mathrm{N} \Rightarrow \mathrm{S}}\,{\downarrow}L$$

In labelled natural deduction, example (293c) is derived as shown in Figure 6.3.

[4] Of course, parentheticals cannot really occur *any*where, e.g. **The, fortunately, man left.* In the end there will have to be some kinds of domains which they cannot penetrate.

$$\dfrac{\text{John}}{\textbf{John}-j\text{: N}} \qquad \dfrac{\text{has}}{\textbf{has}-\mathit{have}\text{: (N}\backslash\text{S)/N}} \qquad \dfrac{\text{perseverance}}{\textbf{perseverance}-\mathit{perseverance}\text{: N}}$$

$$\dfrac{\textbf{has}-\mathit{have}\text{: (N}\backslash\text{S)/N} \qquad \textbf{perseverance}-\mathit{perseverance}\text{: N}}{\textbf{has+perseverance}-(\mathit{have\ perseverance})\text{: N}\backslash\text{S}}\ E/$$

$$\dfrac{\textbf{John}-j\text{: N} \qquad \textbf{has+perseverance}-(\mathit{have\ perseverance})\text{: N}\backslash\text{S}}{\textbf{John+has+perseverance}-(\mathit{have\ perseverance\ j})\text{: S}}\ E\backslash$$

$$\dfrac{\textbf{John+has+perseverance}-(\mathit{have\ perseverance\ j})\text{: S}}{\textbf{John+has+1+perseverance}-(\mathit{have\ perseverance\ j})\text{: }\hat{}\text{S}}\ I\hat{}$$

$$\dfrac{\text{fortunately}}{\textbf{fortunately}-\mathit{fortunately}\text{: }\hat{}\text{S}{\downarrow}\text{S}}$$

$$\dfrac{\textbf{John+has+1+perseverance}-(\mathit{have\ perseverance\ j})\text{: }\hat{}\text{S} \qquad \textbf{fortunately}-\mathit{fortunately}\text{: }\hat{}\text{S}{\downarrow}\text{S}}{\textbf{John+has+fortunately+perseverance}-(\mathit{fortunately}\ (\mathit{have\ perseverance\ j}))\text{: S}}\ E{\downarrow}$$

Figure 6.3: Labelled natural deduction derivation of parenthesization (293c)

$$
\dfrac{
\dfrac{\dfrac{N, TV, N \Rightarrow S}{N, [\,], N \Rightarrow S{\uparrow}TV}\,{\uparrow}R}{N, N \Rightarrow \hat{}(S{\uparrow}TV)}\,\hat{}R
\qquad
\dfrac{\dfrac{N, TV, N \Rightarrow S}{N, [\,], N \Rightarrow S{\uparrow}TV}\,{\uparrow}R \qquad \dfrac{TV \Rightarrow TV \qquad S \Rightarrow S}{S{\uparrow}TV\{TV\} \Rightarrow S}\,{\uparrow}L}{N, TV, N, (S{\uparrow}TV)\backslash(S{\uparrow}TV) \Rightarrow S}\,\backslash L
}{N, TV, N, ((S{\uparrow}TV)\backslash(S{\uparrow}TV))/\hat{}(S{\uparrow}TV), N, N \Rightarrow S}\,/L
$$

Figure 6.4: Hypersequent derivation of gapping (296)

Gapping

Gapping is a coordinate construction in which, in English in the simplest case, a verb missing medially in the second conjunct shares its interpretation with one present in the first conjunct:

(296) John studies logic, and Charles, phonetics.

Hendriks (1995) proposed a like-type coordination assignment for gapping which we adapt as follows, where TV abbreviates (N\S)/N.

(297) **and** : $((S{\uparrow}TV)\backslash(S{\uparrow}TV))/\hat{}(S{\uparrow}TV) : \lambda x \lambda y \lambda z[(y\ z) \wedge (x\ z)]$

That the coordination is (almost) like-type is attractive, since it narrows the distance between gapping and constituent coordination (cf. Steedman (1990)). The example (296) is derived as shown in Figure 6.4.

Comparative Subdeletion

Comparative subdeletion refers to comparisons in which the *than*-clause is missing a determiner:

(298) John ate more donuts than Mary bought bagels.

Type-logical analyses were given in Hendriks (1995), see also Morrill and Merenciano (1996). Here we assign separate types to the two comparative elements:

$$\dfrac{\dfrac{\mathrm{N, TV, Q, CN} \Rightarrow \mathrm{S}}{\mathrm{N, TV, [], CN} \Rightarrow \mathrm{S{\uparrow}Q}}\uparrow R \qquad \dfrac{\dfrac{\dfrac{\mathrm{CP/S, N, TV, Q, CN} \Rightarrow \mathrm{CP}}{\mathrm{CP/S, N, TV, [], CN} \Rightarrow \mathrm{CP{\uparrow}Q}}\uparrow R}{\mathrm{CP/S, N, TV, CN} \Rightarrow \hat{}(\mathrm{CP{\uparrow}Q})}\hat{}R \qquad \mathrm{S} \Rightarrow \mathrm{S}}{\mathrm{S/\hat{}(CP{\uparrow}Q), CP/S, N, TV, CN} \Rightarrow \mathrm{S}}/L}{\mathrm{N, TV, (S{\uparrow}Q){\downarrow}(S/\hat{}(CP{\uparrow}Q)), CN, CP/S, N, TV, CN} \Rightarrow \mathrm{S}}\downarrow L$$

Figure 6.5: Hypersequent derivation of comparative subdeletion (298)

(299) **more** :$(\mathrm{S}{\uparrow}(((\mathrm{S}{\uparrow}\mathrm{N}){\downarrow}\mathrm{S})/\mathrm{CN})){\downarrow}(\mathrm{S}/(\mathrm{CP}{\uparrow}\hat{}(((\mathrm{S}{\uparrow}\mathrm{N}){\downarrow}\mathrm{S})/\mathrm{CN})))$:
$\lambda x\lambda y[|\lambda z(x\ \lambda p\lambda q[(p\ z) \wedge (q\ z)])| > |\lambda z(y\ \lambda p\lambda q[(p\ z) \wedge (q\ z)])|]$

(300) **than** :CP/S: λxx

Then (298) is derived as shown in Figure 6.5, where Q abbreviates $((\mathrm{S}{\uparrow}\mathrm{N}){\downarrow}\mathrm{S})/\mathrm{CN}$ and TV abbreviates $(\mathrm{N}\backslash\mathrm{S})/\mathrm{N}$.

Cross-Serial Dependencies

Chomsky (1957) informally argued that even if natural languages were context-free, context-free grammar could never give a scientifically satisfactory characterisation of even English. Although some have defended context-freeness (Gazdar, Klein, Pullum, and Sag (1985)), Huybregts (1976), Huybregts (1985) argued that Dutch is not context-free and Shieber (1985) formally proved that Swiss German is not context-free.[5] The relevant feature of both languages is semantic cross-serial dependency in subordinate clauses, and the formal (i.e. string set) proof is enabled by the morphological case-marking of dependents by verbs in Swiss German (but not Dutch). Cross-serial dependency in Swiss German is illustrated by the following examples:

(301) a. . . . das mer em Hans es huus hälfed aastriiche

. . . das	mer	em Hans	es huus	hälfed	aastriiche
that	we	Hans-dat	the house-acc	helped	paint

“that we helped Hans paint the house”

b.

. . . das	mer	d’chind	em Hans	es huus	lönd	hälfe	aastriiche
that	we	the children-acc	Hans-dat	the house-acc	let	help	paint

“that we let the children help Hans paint the house”

Calcagno (1995) provides an analysis of cross-serial dependencies which is a close precedent to ours, but in terms of categorial head-wrapping of headed strings. In that account, all expressions are of the same datatype (headed string) and there is no sorting. Here we present a similar account, but using the same sorted discontinuity calculus also motivated by our other linguistic applications.

Dutch subordinate clauses are verb final:

(302)

(. . . dat)	Jan	boeken	las		
(. . . that)	J.	books	read		
CP/S	N	N	$\mathrm{N}\backslash(\mathrm{N}\backslash\mathrm{S})$	$\Rightarrow$	CP

‘(. . . that) Jan read books’

[5]See also Culy (1985)Culy (1985) for the non-context-freeness of Bambara morphology.

Modals and control verbs, so-called verb raising triggers, appear in a clause-final verb cluster but in the English word order relative to one another:

(303)

(… dat)	Jan	boeken	kan	lezen		
(… that)	J.	books	is able	read		
CP/S	N	N	$(\mathrm{N}\backslash\mathrm{Si}){\downarrow}(\mathrm{N}\backslash\mathrm{S})$	$\triangleright^{-1}(\mathrm{N}\backslash(\mathrm{N}\backslash\mathrm{Si}))$	$\Rightarrow$	CP

'(… that) Jan is able to read books'

(304)

(… dat)	Jan	boeken	wil	kunnen
(… that)	J.	books	wants	be able
CP/S	N	N	$(\mathrm{N}\backslash\mathrm{Si}){\downarrow}(\mathrm{N}\backslash\mathrm{S})$	$\triangleright^{-1}((\mathrm{N}\backslash\mathrm{Si}){\downarrow}(\mathrm{N}\backslash\mathrm{Si}))$

lezen		
read		
$\triangleright^{-1}(\mathrm{N}\backslash(\mathrm{N}\backslash\mathrm{Si}))$	$\Rightarrow$	CP

'(… that) Jan wants to be able to read books'

The basic idea of our analysis Morrill (2000a), Morrill, Fadda, and Valentín (2007) is to mark the left edge of the subordinate clause verb cluster with a separator, and to have successive verb-raising triggers infixing at this point and inserting another separator to their own left (if they are infinitive) or closing off the point of discontinuity (if they are finite). The labelled natural deduction derivation of the subordinate clause verb phrase in (304) is given in Figure 6.6.

However, caution needs to be taken in relation to the interaction of verb-raising with our account of quantification:

(305) a.

…	dat	Jan	alles	las
	that	Jan	everything	read

"that Jan read everything"

b.

…	dat	Jan	alles	kan	lezen
	that	Jan	everything	is-able	read

"that Jan can read everything"

Such a quantifier phrase presumably at least sometimes needs to be allowed to take scope which is intermediate with respect to the clause-final verbs, i.e. within clauses that still contain the verb-raising separator, and the process of quantification introduces its own separator in addition. The two positions of discontinuity must not be confused.

We thus propose to allow a new additional separator for the verb cluster left edge. Thus Si is of sort 1_v where 1_v is the new verb cluster left edge separator, etc. All the machinery is just duplicated for the additional separator. The ease of this iteration (which can apparently be repeated indefinitely) would appear to be a virtue of the separator approach. But for the purposes of illustration here we use the same notation as always.

To generate alternative quantifier scopings of examples like (305b) we require a quantifier assignment for quantification in Si of sort 1, in addition to the standard one for quantification in S(fin) of sort 0 for (305a); we would want to collapse these into a sort-polymorphic type:

(306) **alles** $-$ $\lambda x \forall y[(\mathit{thing}\ y) \rightarrow (x\ y)]$
$:=$ $(\mathrm{S}\alpha{\uparrow}\mathrm{N}){\downarrow}\mathrm{S}\alpha$

$$
\dfrac{
  \dfrac{
    \dfrac{\text{boeken}}{\textbf{boeken}-\textit{books}\colon \mathrm{N}}
    \quad
    \dfrac{\dfrac{\text{lezen}}{\textbf{lezen}-\textit{read}\colon \triangleright^{-1}\mathrm{N}\backslash(\mathrm{N}\backslash\mathrm{Sinf})}}{1+\textbf{lezen}-\textit{read}\colon \mathrm{N}\backslash(\mathrm{N}\backslash\mathrm{Sinf})}\,E\triangleright^{-1}
  }{\textbf{boeken}+1+\textbf{lezen}-(\textit{read books})\colon \mathrm{N}\backslash\mathrm{Sinf}}\,E\backslash
  \quad
  \dfrac{\dfrac{\text{kunnen}}{\textbf{kunnen}-\textit{be-able}\colon \triangleright^{-1}(\mathrm{N}\backslash\mathrm{Sinf})\downarrow(\mathrm{N}\backslash\mathrm{Sinf})}}{1+\textbf{kunnen}-\textit{be-able}\colon (\mathrm{N}\backslash\mathrm{Sinf})\downarrow(\mathrm{N}\backslash\mathrm{Sinf})}\,E\triangleright^{-1}
}{
  \dfrac{\textbf{boeken}+1+\textbf{kunnen}+\textbf{lezen}-(\textit{be-able}\ (\textit{read books}))\colon \mathrm{N}\backslash\mathrm{Sinf}\quad \dfrac{\text{wil}}{\textbf{wil}-\textit{want}\colon (\mathrm{N}\backslash\mathrm{Sinf})\downarrow(\mathrm{N}\backslash\mathrm{S})}}{\textbf{boeken}+\textbf{wil}+\textbf{kunnen}+\textbf{lezen}-(\textit{want}\ (\textit{be-able}\ (\textit{read books})))\colon \mathrm{N}\backslash\mathrm{Sinf}}\,E\downarrow
}\,E\downarrow
$$

Figure 6.6: Labelled natural deduction derivation of Dutch verb-raising (304)

When the infinitival complement verbs also take objects, cross-serial dependencies are generated:[6]

(307)

(... dat)	Jan	Cecilia$_1$	Henk$_2$	de	nijlpaarden$_3$
(... that)	J.	C.	H.	the	hippos
CP/S	N	N	N	N/CN	CN

zag$_1$	helpen$_2$	voeren$_3$		
saw	help	feed		
$(N\backslash Si)\downarrow(N\backslash(N\backslash S))$	$\triangleright^{-1}((N\backslash Si)\downarrow(N\backslash(N\backslash Si)))$	$\triangleright^{-1}(N\backslash(N\backslash Si))$	$\Rightarrow$	CP

'(... that) Jan saw$_1$ Cecilia$_1$ help$_2$ Henk$_2$ feed$_3$ the hippos$_3$'

These are generated by assignments with the verbs seeking objects cross-serially to the far left after infixing at the separator-marked left edge of the verb cluster.

Main clause yes/no interrogative word order, V1, is derived from subordinate clause word order by fronting the finite verb. We therefore propose a lexical rule mapping (subordinate clause) finite verb types V to $Q/\hat{}(S{\uparrow}V)$, cf. Hepple (1990).

(308)

Wil	Jan	boeken	lezen?		
wants	J.	books	read		
$Q/\hat{}(S{\uparrow}((N\backslash Si)\downarrow(N\backslash S)))$	N	N	$\triangleright^{-1}(N\backslash(N\backslash Si))$	$\Rightarrow$	Q

'Does Jan want to read books?'

Main clause declarative word order, V2, is further derived from V1 by fronting a major constituent. We propose to achieve this by allowing complex distinguished types (Morrill and Gavarró (1992)).

(309)

Jan	wil	boeken	lezen.		
J.	wants	books	read		
N	$Q/\hat{}(S{\uparrow}((N\backslash Si)\downarrow(N\backslash S)))$	N	$\triangleright^{-1}(N\backslash(N\backslash Si))$	$\Rightarrow$	$N\bullet\hat{}(Q{\uparrow}N)$

'Jan wants to read books.'

A hypersequent calculus derivation of the main clause *Jan wil boeken lezen* is given in Figure 6.7, where VP abbreviates $N\backslash S$ and VPi abbreviates $N\backslash Si$. A hypersequent calculus derivation of the main clause *Marie zegt dat Jan Cecilia Henk de nijlpaarden zag helpen voeren* ('Marie says that Jan saw Cecilia help Henk feed the hippos'), with subordinate clause cross-serial dependencies, is given in Figure 6.8.

6.1.2 Linguistic Applications of Deterministic 2-D

Here we consider deterministic discontinuity allowing two separators. The level of 2-**D** is of course 2 and atomicity also 2. Here nevertheless, linguistic applications have only at most atomicity 1.

Reflexivization

Reflexive pronouns occupy nominal positions and take their interpretation from an antecedent noun phrase. This antecedent is usually clause-local (Principle A). The antecedent can be a subject as in (310a) or an object as in (310b):

[6] 'An increasing load in processing makes such multiple embeddings increasingly unacceptable', Steedman (1985), fn. 29, p.546.

$$\dfrac{\mathrm{N} \Rightarrow \mathrm{N} \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\mathrm{N} \Rightarrow \mathrm{N} \qquad \dfrac{\sqrt[0]{\mathrm{VPi}}, [], \sqrt[1]{\mathrm{VPi}} \Rightarrow \sqrt[0]{\mathrm{VPi}}, [], \sqrt[1]{\mathrm{VPi}} \qquad \mathrm{N}, \mathrm{VP} \Rightarrow \mathrm{S}}{\mathrm{N}, \sqrt[0]{\mathrm{VPi}}, \mathrm{VPi}{\downarrow}\mathrm{VP}, \sqrt[1]{\mathrm{VPi}} \Rightarrow \mathrm{S}}\downarrow L}{\mathrm{N}, \mathrm{N}, \sqrt[0]{\mathrm{N}\backslash\mathrm{VPi}}, \mathrm{VPi}{\downarrow}\mathrm{VP}, \sqrt[1]{\mathrm{N}\backslash\mathrm{VPi}} \Rightarrow \mathrm{S}}\backslash L}{\mathrm{N}, \mathrm{N}, \mathrm{VPi}{\downarrow}\mathrm{VP}, \triangleright^{-1}(\mathrm{N}\backslash\mathrm{VPi}) \Rightarrow \mathrm{S}}\triangleright^{-1}L}{\mathrm{N}, \mathrm{N}, [], \triangleright^{-1}(\mathrm{N}\backslash\mathrm{VPi}) \Rightarrow \sqrt[0]{\mathrm{S}{\uparrow}(\mathrm{VPi}{\downarrow}\mathrm{VP})}, [], \sqrt[1]{\mathrm{S}{\uparrow}(\mathrm{VPi}{\downarrow}\mathrm{VP})}}\uparrow R}{\mathrm{N}, \mathrm{N}, \triangleright^{-1}(\mathrm{N}\backslash\mathrm{VPi}) \Rightarrow \hat{}(\mathrm{S}{\uparrow}(\mathrm{VPi}{\downarrow}\mathrm{VP}))}\hat{}R \qquad \mathrm{Q} \Rightarrow \mathrm{Q}}{\mathrm{Q}/\hat{}(\mathrm{S}{\uparrow}(\mathrm{VPi}{\downarrow}\mathrm{VP})), \mathrm{N}, \mathrm{N}, \triangleright^{-1}(\mathrm{N}\backslash\mathrm{VPi}) \Rightarrow \mathrm{Q}}/L}{\mathrm{Q}/\hat{}(\mathrm{S}{\uparrow}(\mathrm{VPi}{\downarrow}\mathrm{VP})), [], \mathrm{N}, \triangleright^{-1}(\mathrm{N}\backslash\mathrm{VPi}) \Rightarrow \sqrt[0]{\mathrm{Q}{\uparrow}\mathrm{N}}, [], \sqrt[1]{\mathrm{Q}{\uparrow}\mathrm{N}}}\uparrow R}{\mathrm{Q}/\hat{}(\mathrm{S}{\uparrow}(\mathrm{VPi}{\downarrow}\mathrm{VP})), \mathrm{N}, \triangleright^{-1}(\mathrm{N}\backslash\mathrm{VPi}) \Rightarrow \hat{}(\mathrm{Q}{\uparrow}\mathrm{N})}\hat{}R}{\mathrm{N}, \mathrm{Q}/\hat{}(\mathrm{S}{\uparrow}(\mathrm{VPi}{\downarrow}\mathrm{VP})), \mathrm{N}, \triangleright^{-1}(\mathrm{N}\backslash\mathrm{VPi}) \Rightarrow \mathrm{N}{\bullet}\hat{}(\mathrm{Q}{\uparrow}\mathrm{N})}\bullet R$$

Figure 6.7: Hypersequent derivation of (309): *Jan wil boeken lezen.*

(310) a. John$_i$ sent himself$_i$ flowers.

b. Dorothy bet [the straw man]$_i$ half of himself$_i$ that she would reach Emerald City first.

In e.g. Norwegian, subject-oriented and object-oriented reflexives have distinct forms; in English they are the same, but we treat them apart. For the subject case we assume the following assignment, cf. Moortgat (1991):

(311) **himself** : $((\mathrm{N}\backslash\mathrm{S}){\uparrow}\mathrm{N}){\downarrow}(\mathrm{N}\backslash\mathrm{S}) : \lambda x \lambda y(x\ y\ y)$

Then (310a) is derived as follows:

(312) $$\dfrac{\dfrac{(\mathrm{N}\backslash\mathrm{S})/(\mathrm{N}{\bullet}\mathrm{N}), \mathrm{N}, \mathrm{N} \Rightarrow \mathrm{N}\backslash\mathrm{S}}{(\mathrm{N}\backslash\mathrm{S})/(\mathrm{N}{\bullet}\mathrm{N}), [], \mathrm{N} \Rightarrow (\mathrm{N}\backslash\mathrm{S}){\uparrow}\mathrm{N}}\uparrow R \qquad \mathrm{N}, \mathrm{N}\backslash\mathrm{S} \Rightarrow \mathrm{S}}{\mathrm{N}, (\mathrm{N}\backslash\mathrm{S})/(\mathrm{N}{\bullet}\mathrm{N}), ((\mathrm{N}\backslash\mathrm{S}){\uparrow}\mathrm{N}){\downarrow}(\mathrm{N}\backslash\mathrm{S}), \mathrm{N} \Rightarrow \mathrm{S}}\downarrow L$$

We leave it to the reader to check that this account of reflexivisation interacts correctly with our account of quantification to create binding of a reflexive by a quantified antecedent, as in:

(313) [Every man]$_i$ loves himself$_i$.

On its own however, this account overgenerates, allowing long-distance reflexivisation in English:

(314) *John$_i$ thinks Mary loves himself$_i$.

In Morrill (1990) and Morrill (1994, chap. 4), it is claimed that once TLG is modalised so as to deliver intensional semantics, the locality of reflexivisation can be captured by a modal type assignment suitably restricting the antecedent to lie within the local intensional/temporal domain, i.e. the same tensed clause as the reflexive.

In English, an object-oriented reflexive must be preceded by its antecedent:

$$\frac{\mathrm{N} \Rightarrow \mathrm{N} \qquad \mathrm{N}, \mathrm{VP/CP}, \mathrm{CP/S}, \mathrm{N}, \mathrm{VP} \Rightarrow \mathrm{S}}{\mathrm{N}, \mathrm{VP/CP}, \mathrm{CP/S}, \mathrm{N}, \mathrm{N}, \mathrm{N}\backslash\mathrm{VP} \Rightarrow \mathrm{S}}\ \backslash L \quad (1)$$

$$\frac{\sqrt[0]{\mathrm{VPi}}, [\,], \sqrt[1]{\mathrm{VPi}} \Rightarrow \sqrt[0]{\mathrm{VPi}}, [\,], \sqrt[1]{\mathrm{VPi}} \qquad (1)}{\mathrm{N}, \mathrm{VP/CP}, \mathrm{CP/S}, \mathrm{N}, \mathrm{N}, \sqrt[0]{\mathrm{VPi}}, \mathrm{VPi}{\downarrow}(\mathrm{N}\backslash\mathrm{VP}), \sqrt[1]{\mathrm{VPi}} \Rightarrow \mathrm{S}}\ {\downarrow}L \quad (2)$$

$$\frac{\mathrm{N} \Rightarrow \mathrm{N} \qquad (2)}{\mathrm{N}, \mathrm{VP/CP}, \mathrm{CP/S}, \mathrm{N}, \mathrm{N}, \mathrm{N}, \sqrt[0]{\mathrm{N}\backslash\mathrm{VPi}}, \mathrm{VPi}{\downarrow}(\mathrm{N}\backslash\mathrm{VP}), \sqrt[1]{\mathrm{N}\backslash\mathrm{VPi}} \Rightarrow \mathrm{S}}\ \backslash L \quad (3)$$

$$\frac{\sqrt[0]{\mathrm{VPi}}, [\,], \sqrt[1]{\mathrm{VPi}} \Rightarrow \sqrt[0]{\mathrm{VPi}}, [\,], \sqrt[1]{\mathrm{VPi}} \qquad (3)}{\mathrm{N}, \mathrm{VP/CP}, \mathrm{CP/S}, \mathrm{N}, \mathrm{N}, \mathrm{N}, \sqrt[0]{\mathrm{VPi}}, \sqrt[0]{\mathrm{VPi}{\downarrow}(\mathrm{N}\backslash\mathrm{VPi})}, \mathrm{VPi}{\downarrow}(\mathrm{N}\backslash\mathrm{VP}), \sqrt[1]{\mathrm{VPi}{\downarrow}(\mathrm{N}\backslash\mathrm{VPi})}, \sqrt[1]{\mathrm{VPi}} \Rightarrow \mathrm{S}}\ {\downarrow}L \quad (4)$$

$$\frac{(4)}{\mathrm{N}, \mathrm{VP/CP}, \mathrm{CP/S}, \mathrm{N}, \mathrm{N}, \mathrm{N}, \sqrt[0]{\mathrm{VPi}}, \mathrm{VPi}{\downarrow}(\mathrm{N}\backslash\mathrm{VP}), \triangleright^{-1}(\mathrm{VPi}{\downarrow}(\mathrm{N}\backslash\mathrm{VPi})), \sqrt[1]{\mathrm{VPi}} \Rightarrow \mathrm{S}}\ \triangleright^{-1}L \quad (5)$$

$$\frac{\mathrm{N} \Rightarrow \mathrm{N} \qquad (5)}{\mathrm{N}, \mathrm{VP/CP}, \mathrm{CP/S}, \mathrm{N}, \mathrm{N}, \mathrm{N}, \mathrm{N}, \sqrt[0]{\mathrm{N}\backslash\mathrm{VPi}}, \mathrm{VPi}{\downarrow}(\mathrm{N}\backslash\mathrm{VP}), \triangleright^{-1}(\mathrm{VPi}{\downarrow}(\mathrm{N}\backslash\mathrm{VPi})), \sqrt[1]{\mathrm{N}\backslash\mathrm{VPi}} \Rightarrow \mathrm{S}}\ \backslash L \quad (6)$$

$$\frac{(6)}{\mathrm{N}, \mathrm{VP/CP}, \mathrm{CP/S}, \mathrm{N}, \mathrm{N}, \mathrm{N}, \mathrm{N}, \mathrm{VPi}{\downarrow}(\mathrm{N}\backslash\mathrm{VP}), \triangleright^{-1}(\mathrm{VPi}{\downarrow}(\mathrm{N}\backslash\mathrm{VPi})), \triangleright^{-1}(\mathrm{N}\backslash\mathrm{VPi}) \Rightarrow \mathrm{S}}\ \triangleright^{-1}L \quad (7)$$

$$\frac{(7)}{\mathrm{N}, [\,], \mathrm{CP/S}, \mathrm{N}, \mathrm{N}, \mathrm{N}, \mathrm{N}, \mathrm{VPi}{\downarrow}(\mathrm{N}\backslash\mathrm{VP}), \triangleright^{-1}(\mathrm{VPi}{\downarrow}(\mathrm{N}\backslash\mathrm{VPi})), \triangleright^{-1}(\mathrm{N}\backslash\mathrm{VPi}) \Rightarrow \sqrt[0]{\mathrm{S}{\uparrow}(\mathrm{VP/CP})}, [\,], \sqrt[1]{\mathrm{S}{\uparrow}(\mathrm{VP/CP})}}\ {\uparrow}R \quad (8)$$

$$\frac{(8)}{\mathrm{N}, \mathrm{CP/S}, \mathrm{N}, \mathrm{N}, \mathrm{N}, \mathrm{N}, \mathrm{VPi}{\downarrow}(\mathrm{N}\backslash\mathrm{VP}), \triangleright^{-1}(\mathrm{VPi}{\downarrow}(\mathrm{N}\backslash\mathrm{VPi})), \triangleright^{-1}(\mathrm{N}\backslash\mathrm{VPi}) \Rightarrow \hat{}(\mathrm{S}{\uparrow}(\mathrm{VP/CP}))}\ \hat{}R \quad (9)$$

$$\frac{(9) \qquad \mathrm{Q} \Rightarrow \mathrm{Q}}{\mathrm{Q}/\hat{}(\mathrm{S}{\uparrow}(\mathrm{VP/CP})), \mathrm{N}, \mathrm{CP/S}, \mathrm{N}, \mathrm{N}, \mathrm{N}, \mathrm{N}, \mathrm{VPi}{\downarrow}(\mathrm{N}\backslash\mathrm{VP}), \triangleright^{-1}(\mathrm{VPi}{\downarrow}(\mathrm{N}\backslash\mathrm{VPi})), \triangleright^{-1}(\mathrm{N}\backslash\mathrm{VPi}) \Rightarrow \mathrm{Q}}\ /L \quad (10)$$

$$\frac{(10)}{\mathrm{Q}/\hat{}(\mathrm{S}{\uparrow}(\mathrm{VP/CP})), [\,], \mathrm{CP/S}, \mathrm{N}, \mathrm{N}, \mathrm{N}, \mathrm{N}, \mathrm{VPi}{\downarrow}(\mathrm{N}\backslash\mathrm{VP}), \triangleright^{-1}(\mathrm{VPi}{\downarrow}(\mathrm{N}\backslash\mathrm{VPi})), \triangleright^{-1}(\mathrm{N}\backslash\mathrm{VPi}) \Rightarrow \sqrt[0]{\mathrm{Q}{\uparrow}\mathrm{N}}, [\,], \sqrt[1]{\mathrm{Q}{\uparrow}\mathrm{N}}}\ {\uparrow}R \quad (11)$$

$$\frac{(11)}{\mathrm{Q}/\hat{}(\mathrm{S}{\uparrow}(\mathrm{VP/CP})), \mathrm{CP/S}, \mathrm{N}, \mathrm{N}, \mathrm{N}, \mathrm{N}, \mathrm{VPi}{\downarrow}(\mathrm{N}\backslash\mathrm{VP}), \triangleright^{-1}(\mathrm{VPi}{\downarrow}(\mathrm{N}\backslash\mathrm{VPi})), \triangleright^{-1}(\mathrm{N}\backslash\mathrm{VPi}) \Rightarrow \hat{}(\mathrm{Q}{\uparrow}\mathrm{N})}\ \hat{}R \quad (12)$$

$$\frac{\mathrm{N} \Rightarrow \mathrm{N} \qquad (12)}{\mathrm{N}, \mathrm{Q}/\hat{}(\mathrm{S}{\uparrow}(\mathrm{VP/CP})), \mathrm{CP/S}, \mathrm{N}, \mathrm{N}, \mathrm{N}, \mathrm{N}, \mathrm{VPi}{\downarrow}(\mathrm{N}\backslash\mathrm{VP}), \triangleright^{-1}(\mathrm{VPi}{\downarrow}(\mathrm{N}\backslash\mathrm{VPi})), \triangleright^{-1}(\mathrm{N}\backslash\mathrm{VPi}) \Rightarrow \mathrm{N}\bullet\hat{}(\mathrm{Q}{\uparrow}\mathrm{N})}\ \bullet R \quad (13)$$

Figure 6.8: Hypersequent derivation of *Marie zegt dat Jan Cecilia Henk de nijlpaarden zag helpen voeren*

$$\dfrac{\dfrac{\dfrac{\text{(VP/CP)/(N}\bullet\text{N), N, N/N, N, CP} \Rightarrow \text{VP}}{\text{(VP/CP)/(N}\bullet\text{N), [], N/N, N, CP} \Rightarrow \text{VP}\uparrow\text{N}}\uparrow R}{\text{(VP/CP)/(N}\bullet\text{N), [], N/N, [], CP} \Rightarrow \text{(VP}\uparrow\text{N)}\uparrow_2\text{N}}\uparrow_2 R \quad \dfrac{\text{N} \Rightarrow \text{N} \qquad \text{VP} \Rightarrow \text{VP}}{\text{VP}\uparrow\text{N}\{\text{N}\} \Rightarrow \text{VP}}\uparrow L}{\text{(VP/CP)/(N}\bullet\text{N), N, N/N, ((VP}\uparrow\text{N)}\uparrow_2\text{N)}\downarrow_2\text{(VP}\uparrow\text{N), CP} \Rightarrow \text{VP}}\downarrow_2 L$$

Figure 6.9: Hypersequent derivation of VP medial object-oriented reflexivization (310b)

(315) a. Mary talked to John$_i$ about himself$_i$.

b. *Mary talked about himself$_i$ to John$_i$

Such a feature can be captured using second-position deterministic wrapping (VP abbreviates N\S):

(316) **himself** : $((\text{VP}\uparrow\text{N})\uparrow_2\text{N})\downarrow_2(\text{VP}\uparrow\text{N}) : \lambda x\lambda y(x\ y\ y)$

Then (310b) is derived as shown in Figure 6.9.

6.1.3 Nondeterminism

Until now we have used the implicative and product connectives of **D** jointly with deterministic synthetic connectives. In this subsection we take a look at examples using nondeterministic connectives (unary and binary). Here we consider nondeterministic discontinuity allowing two separators. The level is of course 2 and the examples' atomicity is at most 1. For the commodity of the reader, we recall the definition of the nondeterministic synthetic connectives with their standard syntactical interpretation leaving aside the definition of these connectives with the linear logic additive connectives conjunction and disjunction & and $\oplus$. We recall the sort polymorphic relations $+(s_1, s_2, s_3)$ and $\times(s_1, s_2, s_3)$. $+(s_1, s_2, s_3)$ holds iff $s_3 = s_1{+}s_2$ or $s_3 = s_2{+}s_1$. $\times(s_1, s_2, s_3)$ holds iff either $s_3 = s_1 \times_1 s_2$ or $s_3 = s_1 \times_2 s_2$ or $\ldots$ or $s_3 = s_1 \times_{S(s_1)} s_2$. Denotation $[\![\,\cdot\,]\!]_v$ for a given displacement algebra and a valuation v will be simply written $[\![\,\cdot\,]\!]$.

nondeterministic division
$[\![\frac{B}{A}]\!]\overset{def}{=}\{s : \forall s' \in [\![A]\!], s_3,\ +(s, s', s_3) \rightarrow s_3 \in [\![B]\!]\}$

nondeterministic product
$[\![A \otimes B]\!]\overset{def}{=}\{s_3 : \exists s_1 \in [\![A]\!], s_2 \in [\![B]\!]\},\ +(s_1, s_2, s_3)\}$

nondeterministic infix
$[\![A {\Downarrow} C]\!]\overset{def}{=}\{s_2 : \forall s_1 \in [\![A]\!], s_3,\ \times(s_1, s_2, s_3) \rightarrow s_3 \in [\![C]\!]\}$

(317) nondeterministic extract
$[\![C {\Uparrow} B]\!]\overset{def}{=}\{s_1 : \forall s_2 \in [\![B]\!], s_3,\ \times(s_1, s_2, s_3) \rightarrow s_3 \in [\![C]\!]\}$

nondeterministic discontinuous product
$[\![A \circledcirc B]\!]\overset{def}{=}\{s_3 : \exists s_1 \in [\![A]\!], s_2 \in [\![B]\!],\ \times(s_1, s_2, s_3)\}$

nondeterministic bridge
$[\![\hat{\ } A]\!]\overset{def}{=}\{s_3 : \exists s_1 \in [\![A]\!],\ \times(s_1, \epsilon, s_3)\}$

nondeterministic split
$[\![\check{\ } A]\!]\overset{def}{=}\{s_1 : \times(s_1, \epsilon, s_3) \rightarrow s_3 \in [\![A]\!]\}$

Subject postposing

A functor of type $\frac{B}{A}$ can concatenate with its argument either to the left or to the right to form B. For example in Catalan:

(318) a. Barcelona creix.
'Barcelona expands/grows.'

b. Creix Barcelona.
'Expands/grows barcelona.'

This generalization may be captured by assigning a verb phrase such as *creix* the type $\frac{\mathrm{B}}{\mathrm{A}}$.
Another equivalent formulation of $\frac{B}{A}$ can be simulated (with the aid of deterministic and nondeterministic unary synthetic connectives) by:

$$\frac{B}{A} \Rightarrow \triangleleft^{-1}\triangleright^{-1}(\check{\ }(S{\uparrow}_1 N))$$
$$\triangleleft^{-1}\triangleright^{-1}(\check{\ }(S{\uparrow}_1 N)) \Rightarrow \frac{B}{A}$$

Notice that the level of $\frac{B}{A}$ is 1, whereas its equivalent type in **DADND** is 2, for the subtype $\check{\ }(S{\uparrow}N)$ has level 2.
Let us see some examples involving the equivalent type $\triangleleft^{-1}\triangleright^{-1}(\check{\ }(S{\uparrow}_1 N))$. The derivations in labelled natural deduction of (318) are the following:

$$\cfrac{\cfrac{\cfrac{\cfrac{\mathbf{creix} : \triangleleft^{-1}\triangleright^{-1}(\breve{}(S{\uparrow_1}N))}{\mathbf{1} + \mathbf{creix} : \triangleright^{-1}(\breve{}(S{\uparrow_1}N))}\triangleleft^{-1}E}{\mathbf{1} + \mathbf{creix} + \mathbf{1} : \breve{}(S{\uparrow_1}N)}\triangleright^{-1}E}{\mathbf{1} + \mathbf{creix} : S{\uparrow_1}N}\breve{}E \qquad \mathbf{Barcelona} : N}{\mathbf{Barcelona} + \mathbf{creix} : N}{\uparrow_1}E$$

$$\cfrac{\cfrac{\cfrac{\cfrac{\mathbf{creix} : \triangleleft^{-1}\triangleright^{-1}(\breve{}(S{\uparrow_1}N))}{\mathbf{1} + \mathbf{creix} : \triangleright^{-1}(\breve{}(S{\uparrow_1}N))}\triangleleft^{-1}E}{\mathbf{1} + \mathbf{creix} + \mathbf{1} : \breve{}(S{\uparrow_1}N)}\triangleright^{-1}E}{\mathbf{creix} + \mathbf{1} : S{\uparrow_1}N}\breve{}E \qquad \mathbf{Barcelona} : N}{\mathbf{creix} + \mathbf{Barcelona} : N}{\uparrow_1}E$$

The derivation of (318b) *Creix Barcelona* in hypersequent calculus is the following:

$$\cfrac{\cfrac{\cfrac{\cfrac{N \Rightarrow N \qquad S \Rightarrow S}{S{\uparrow_1}N\{N\} \Rightarrow S}{\uparrow_1}L}{\breve{}(S{\uparrow_1}N))\{\Lambda : N\} \Rightarrow S}\breve{}L}{\triangleright^{-1}((\breve{}(S{\uparrow_1}N))\{N\} \Rightarrow S}\triangleright^{-1}L}{\triangleleft^{-1}\triangleright^{-1}(\breve{}(S{\uparrow_1}N)), N \Rightarrow S}\triangleleft^{-1}L$$

(319) **Remark**
Some scholars (Vallduví, chapter 4 in Solà, Lloret, Mascaró, and Pérez (1996)) claim that Catalan (and in fact also Iberian Spanish) is a VO language with no constraint on the subject's position, and therefore the typing for a Catalan (Iberian Spanish) transitive verb could be:

$$\mathrm{TV} \stackrel{def}{=} (\frac{\mathrm{S}}{\mathrm{N}})/\mathrm{N}$$

6.1.4 Beyond 2-D?

A question remains as to the levels which the lexicons of natural languages need. Is it necessary to go beyond 2-**D**, to say 3-**D**? This is clearly an open problem. Interestingly, in some linguistic registers, spanish transitive verbs admit orders, SVO, VOS, VSO and OVS.[7] In **D** we can give a typing for such orders (although in fact we *generate* more orders which are very unlikely), and in that case the level would be 4. Consider:

(320) $\mathrm{TV} \stackrel{def}{=} \triangleleft^{-1}\triangleleft^{-1}\triangleright^{-1}\triangleright^{-1}(\breve{}(\breve{}(\mathrm{S}{\uparrow}\mathrm{N}){\Uparrow}\mathrm{N}))$

[7]This is my intuition as an Iberian Spanish native speaker.

Using labelled natural deduction, it is relatively straightforward to derive the $3! = 6$ possible orders which the typing (320) allows to accept. Let us see how the order VSO could be derived:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\mathbf{V} : \triangleleft^{-1}\triangleleft^{-1}\triangleright^{-1}\triangleright^{-1}(\check{}(\check{}(S{\uparrow}N){\Uparrow}N))}{\vdots}\,\triangleleft^{-1}E}{\mathbf{1{+}1{+}V{+}1{+}1} : \check{}(\check{}((S{\uparrow}N){\Uparrow}N))}\,\check{}E}{\mathbf{1{+}V{+}1{+}1} : \check{}((S{\uparrow}N){\Uparrow}N)}\,\check{}E}{\mathbf{V{+}1{+}1} : (S{\uparrow}N){\Uparrow}N}\,\check{}E}{\mathbf{V{+}S{+}1} : (S{\uparrow}N)}\,{\Uparrow}E}{\mathbf{V{+}S{+}O} : S}\,{\uparrow}E$$

6.2 More Logical Machinery for further Binding Theory

In this section $\mathbf{D}$ is extended with the following connectives:

- The Jäger connective $|$ (Jäger (2005)).
- The additive conjunction and disjunction of linear logic (Girard (1987)).
- The S4 modality.
- The difference operator $-$ based on the concept of *negation as failure*, familiar in the field of logic programming.

The Lambek calculus is free of structural rules but anaphora involves duplication of antecedent semantics. Jäger (2005) extends the Lambek calculus with limited contraction to provide an account of anaphora with syntactic duplication. Here we employ a very slight variant of this in the context of the displacement calculus. Limited contraction is for a binary type-constructor $|$ such that $B|A$ signifies an expression of type B containing a free anaphor of type A (cf. Jacobson (1999), who writes B^A). We extend the types of the displacement calculus as follows:

(321) $\mathcal{F}_{i+j} ::= \mathcal{F}_{i+j}|\mathcal{F}_j$

We assume rules as follows, where the semicolon separates disjoint hyperoccurrences which may be consistently in any order left-to-right:[8]

(322) $$\dfrac{\Gamma \Rightarrow A \qquad \Delta\langle\overrightarrow{A};\overrightarrow{B}\rangle \Rightarrow D}{\Delta\langle\Gamma;\overrightarrow{B|A}\rangle \Rightarrow D}\,|L \qquad \dfrac{\Gamma\langle\overrightarrow{B_0};\ldots;\overrightarrow{B_n}\rangle \Rightarrow D}{\Gamma\langle\overrightarrow{B_0|A};\ldots;\overrightarrow{B_n|A}\rangle \Rightarrow D|A}\,|R$$

We call $\mathbf{DC}$ the extension of $\mathbf{D}$ with this version of limited contraction. $\mathbf{DA}$, is the extension of $\mathbf{D}$ with the additive connectives of linear logic (Girard (1987)). In the sorting and hypersequent regime of the discontinuous Lambek calculus these are as follows:

[8] Jäger (2005) has only $|L$ (limited contraction) with the antecedent preceding the anaphor, giving rise to backward anaphora only; our variant allows also forward anaphora (cataphora).

(323) $\mathcal{F}_i ::= \mathcal{F}_i \& \mathcal{F}_i \mid \mathcal{F}_i \oplus \mathcal{F}_i$

(324) $$\frac{\Gamma\langle\overrightarrow{A}\rangle \Rightarrow C}{\Gamma\langle\overrightarrow{A\&B}\rangle \Rightarrow C}\&L_1 \qquad \frac{\Gamma\langle\overrightarrow{B}\rangle \Rightarrow C}{\Gamma\langle\overrightarrow{A\&B}\rangle \Rightarrow C}\&L_2$$

$$\frac{\Gamma \Rightarrow A \qquad \Gamma \Rightarrow B}{\Gamma \Rightarrow A\&B}\&R$$

$$\frac{\Gamma\langle\overrightarrow{A}\rangle \Rightarrow C \qquad \Gamma\langle\overrightarrow{B}\rangle \Rightarrow C}{\Gamma\langle\overrightarrow{A \oplus B}\rangle \Rightarrow C}\oplus L$$

$$\frac{\Gamma \Rightarrow A}{\Gamma \Rightarrow A \oplus B}\oplus L_1 \qquad \frac{\Gamma \Rightarrow B}{\Gamma \Rightarrow A \oplus B}\oplus L_2$$

We call **D**$\Box$ the extension of **D** with S4 modality. In the sorting and hypersequent regime of the **D** this is as follows, where $\Box\Gamma$ signifies a configuration all the types of which have $\Box$ as the main connective.

(325) $\mathcal{F}_i ::= \Box\mathcal{F}_i$

(326) $$\frac{\Gamma\langle\overrightarrow{A}\rangle \Rightarrow B}{\Gamma\langle\overrightarrow{\Box A}\rangle \Rightarrow B}\Box L \qquad \frac{\Box\Gamma \Rightarrow A}{\Box\Gamma \Rightarrow \Box A}\Box R$$

We call **DCA**$\Box$ the extension of **D** with limited contraction, additives and S4 modality. The Cut-elimination for this calculus is proved in Morrill and Valentín (2011) (to appear).

Finally, we add the difference operator $-$. The definition of syntactic types and the semantic type map T sending syntactic types to semantic types is as shown in Figure 6.10 for the calculus **DCA**$\Box$ with succedent difference. We call this calculus **DCA**$\Box-$. The definition distinguishes types with antecedent polarity (superscript $^\bullet$) and succedent polarity (superscript ^{o}); where p is a polarity, $\overline{p}$ is the opposite polarity. The rule for $-$ is the following:

(327) $$\frac{\Gamma \Rightarrow A : \phi \qquad \not\vdash \Gamma \Rightarrow B : \_}{\Gamma \Rightarrow A - B : \phi}-R$$

The sort map for the difference operator satisfies the following:[9]

$$S(A - B) = S(A) = S(B)$$

This rule is nonstandard since the right premise has a non provable hypersequent. We have therefore a rule which combines provability with disprovability. Moreover, this rule has a side effect. If we have a provable **DCA**$\Box-$ hypersequent $\Gamma \Rightarrow A - B$, then the last rule must necessarily be the difference operator rule. And as stated in Figure 6.10, if A and B are types (with the same sort!), then $A - B$ must occur with succedent polarity.

Adding the difference operator brings our categorial logic into the realms of non-monotonic reasoning where the transitivity of the consequence relation

[9] In rule (327) the two premises have the same antecedent Γ. This justifies that $S(A) = S(B)$, and of course that $S(A - B) = S(A) = S(B)$.

$$\begin{array}{rclrcl}
\mathcal{F}^p_j & ::= & \mathcal{F}^{\overline{p}}_i\backslash\mathcal{F}^p_{i+j} & T(A\backslash C) & = & T(A)\to T(C)\\
\mathcal{F}^p_i & ::= & \mathcal{F}^p_{i+j}/\mathcal{F}^{\overline{p}}_j & T(C/B) & = & T(B)\to T(C)\\
\mathcal{F}^p_{i+j} & ::= & \mathcal{F}^p_i\bullet\mathcal{F}^p_j & T(A\bullet B) & = & T(A)\&T(B)\\
\mathcal{F}^p_0 & ::= & I & T(I) & = & \top\\
\mathcal{F}^p_j & ::= & \mathcal{F}^{\overline{p}}_{i+1}\downarrow_k\mathcal{F}^p_{i+j} & T(A\downarrow_k C) & = & T(A)\to T(C)\\
\mathcal{F}^p_{i+1} & ::= & \mathcal{F}^p_{i+j}\uparrow_k\mathcal{F}^{\overline{p}}_j & T(C\uparrow_k B) & = & T(B)\to T(C)\\
\mathcal{F}^p_{i+j} & ::= & \mathcal{F}^p_{i+1}\odot_k\mathcal{F}^p_j & T(A\odot_k B) & = & T(A)\&T(B)\\
\mathcal{F}^p_1 & ::= & J & T(J) & = & \top\\
\mathcal{F}^p_{i+j} & ::= & \mathcal{F}^p_{i+j}|\mathcal{F}^{\overline{p}}_j & T(B|A) & = & T(A)\to T(B)\\
\mathcal{F}^p_i & ::= & \mathcal{F}^p_i\&\mathcal{F}^p_i & T(A\&B) & = & T(A)\&T(B)\\
\mathcal{F}^p_i & ::= & \mathcal{F}^p_i\oplus\mathcal{F}^p_i & T(A\oplus B) & = & T(A)+T(B)\\
\mathcal{F}^p_i & ::= & \Box\mathcal{F}^p_i & T(\Box A) & = & \mathbf{L}T(A)\\
\mathcal{F}_i{}^o & ::= & \mathcal{F}_i{}^o-\mathcal{F}_i{}^o & T(A-B) & = & T(A)
\end{array}$$

Figure 6.10: Connectives and type map

must be dropped. The other connectives used in this chapter, the additive connectives, the limited contraction | and the S4 modality, enjoy Cut-elimination (see the paper Morrill and Valentín (2011)). But in the presence of negation as failure, the Cut rule must be considered not only no longer eliminable, but inadmissible as Morrill and Valentín (*op. cit.*) show. However, the subformula property holds of all the connectives used here: the hypersequent presentation is such that for every rule, the formula occurrences in the premises are always subformulas of those in the conclusion. Given this state of affairs, the Cut-free backward chaining sequent proof search space is finite and hence the categorial logic $\mathbf{DCA}\Box-$ is decidable.

The treatment of this connective will crucially assign a Curry-Howard term to all the derivations of hypersequents in $\mathbf{DCA}\Box-$, as Morrill and Valentín (*op. cit.*) prove. In the next section in which we present the semantics; Figures 6.14 and 6.15 display the $\mathbf{DCA}\Box-$ calculus labelled with Curry-Howard terms.

6.3 Semantics

The set $\mathcal{T}$ of *semantic types* is defined on the basis of a set δ of *basic semantic types* as follows:

(328) $\mathcal{T} ::= \delta \mid \top \mid \mathcal{T}{+}\mathcal{T} \mid \mathcal{T}\&\mathcal{T} \mid \mathcal{T}{\to}\mathcal{T} \mid \mathbf{L}\mathcal{T}$

A *semantic frame* comprises a non-empty set W of worlds and a family $\{D_\tau\}_{\tau\in\mathcal{T}}$ of non-empty *semantic type domains* such that:

(329)
$$\begin{array}{rclll}
D_\top & = & \{0\} & & \text{singleton set}\\
D_{\tau_1+\tau_2} & = & D_{\tau_2}\uplus D_{\tau_1} & (\{1\}\times D_{\tau_1})\cup(\{2\}\times D_{\tau_2}) & \text{disjoint union}\\
D_{\tau_1\&\tau_2} & = & D_{\tau_1}\times D_{\tau_2} & \{\langle m_1,m_2\rangle|\ m_1\in D_{\tau_1}\ \&\ m_2\in D_{\tau_2}\} & \text{Cartesian product}\\
D_{\tau_1\to\tau_2} & = & D_{\tau_2}^{D_{\tau_1}} & \text{the set of all functions from } D_{\tau_1} \text{ to } D_{\tau_2} & \text{functional exponentiation}\\
D_{\mathbf{L}\tau} & = & D_\tau^W & \text{the set of all functions from } W \text{ to } D_\tau & \text{functional exponentiation}
\end{array}$$

The sets Φ_τ of *terms* of type τ for each type τ are defined on the basis of sets C_τ of constants of type τ and denumerably infinite sets V_τ of variables of type τ for each type τ as shown in Figure 6.11.

$$
\begin{array}{rcll}
\Phi_\tau & ::= & C_\tau & \\
\Phi_\tau & ::= & V_\tau & \\
\Phi_\top & ::= & 0 & \\
\Phi_\tau & ::= & \Phi_{\tau_1+\tau_2} \to V_{\tau_1}.\Phi_\tau;\ V_{\tau_2}.\Phi_\tau & \text{case statement} \\
\Phi_{\tau+\tau'} & ::= & \iota_1\Phi_\tau & \text{first injection} \\
\Phi_{\tau'+\tau} & ::= & \iota_2\Phi_\tau & \text{second injection} \\
\Phi_\tau & ::= & \pi_1\Phi_{\tau\&\tau'} & \text{first projection} \\
\Phi_\tau & ::= & \pi_2\Phi_{\tau'\&\tau} & \text{second projection} \\
\Phi_{\tau\&\tau'} & ::= & (\Phi_\tau, \Phi_{\tau'}) & \text{ordered pair formation} \\
\Phi_\tau & ::= & (\Phi_{\tau'\to\tau}\ \Phi_{\tau'}) & \text{functional application} \\
\Phi_{\tau\to\tau'} & ::= & \lambda V_\tau \Phi_{\tau'} & \text{functional abstraction} \\
\Phi_\tau & ::= & {}^\vee\Phi_{\mathbf{L}\tau} & \text{extensionalization} \\
\Phi_{\mathbf{L}\tau} & ::= & {}^\wedge\Phi_\tau & \text{intensionalization}
\end{array}
$$

Figure 6.11: Syntax of terms for semantic representation

$$
\begin{array}{rcl}
[a]^{g,i} & = & f(a) \text{ for constant } a \in C_\tau \\
[x]^{g,i} & = & g(x) \text{ for variable } x \in V_\tau \\
[0]^{g,i} & = & 0 \\
[\phi \to x.\psi; y.\chi]^{g,i} & = & \begin{cases} [\psi]^{g[x:=m],i} & \text{if } [\phi]^{g,i} = \langle 1, m\rangle \\ [\chi]^{g[y:=m],i} & \text{if } [\phi]^{g,i} = \langle 2, m\rangle \end{cases} \\
[\iota_1\phi]^{g,i} & = & \langle 1, [\phi]^{g,i}\rangle \\
[\iota_2\phi]^{g,i} & = & \langle 2, [\phi]^{g,i}\rangle \\
[\pi_1\phi]^{g,i} & = & \mathbf{fst}([\phi]^{g,i}) \\
[\pi_2\phi]^{g,i} & = & \mathbf{snd}([\phi]^{g,i}) \\
[(\phi, \psi)]^{g,i} & = & \langle [\phi]^{g,i}, [\psi]^{g,i}\rangle \\
[(\phi\ \psi)]^{g,i} & = & [\phi]^{g,i}([\psi]^{g,i}) \\
[\lambda x\phi]^{g,i} & = & m \mapsto [\phi]^{g[x:=m],i} \\
[{}^\vee\phi]^{g,i} & = & [\phi]^{g,i}(i) \\
[{}^\wedge\phi]^{g,i} & = & j \mapsto [\phi]^{g,j}
\end{array}
$$

Figure 6.12: Semantics of terms for semantic representation

Given a semantic frame, a *valuation* f is a function mapping each constant of type τ into an element of D_τ, and an *assignment* g is a function mapping each variable of type τ into an element of D_τ. Where g is such, the update $g[x := m]$ is $(g - \{(x, g(x))\}) \cup \{(x, m)\}$. Relative to a valuation, an assignment g and a world $i \in W$, each term ϕ of type τ receives an interpretation $[\phi]^{g,i} \in D_\tau$ as shown in Figure 6.12.

An occurrence of a variable x in a term is called *free* if and only if it does not fall within any part of the term of the form $x.\cdot$ or $\lambda x\cdot$; otherwise it is *bound* (by the closest $x.$ or λx within the scope of which it falls). The result $\phi[\psi_1/x_1, \ldots, \psi_n/x_n]$ of substituting terms $\psi_1, \ldots, \psi_n$ (of types $\tau_1, \ldots, \tau_n$) for variables $x_1, \ldots, x_n$ (of types $\tau_1, \ldots, \tau_n$) in a term ϕ is the result of simultaneously replacing by $\psi_1, \ldots, \psi_n$ every free occurrence of $x_1, \ldots, x_n$ respectively in ϕ. We say that ψ is *free for x in ϕ* if and only if no variable occurrence in ψ becomes bound in $\phi[\psi/x]$ (i.e. if and only if there is no "accidental capture"). We say that a term is *modally closed* if and only if every occurrence of $^\vee$ occurs

$$\begin{array}{rcl}
\phi \to x.\psi; y.\chi &=& \phi \to z.(\psi[z/x]); y.\chi \\
\multicolumn{3}{l}{\text{if } z \text{ is not free in } \psi \text{ and is free for } x \text{ in } \psi} \\
\phi \to x.\psi; y.\chi &=& \phi \to x.\psi; z.(\chi[z/y]) \\
\multicolumn{3}{l}{\text{if } z \text{ is not free in } \chi \text{ and is free for } y \text{ in } \chi} \\
\lambda x\phi &=& \lambda y(\phi[y/x]) \\
\multicolumn{3}{l}{\text{if } y \text{ is not free in } \phi \text{ and is free for } x \text{ in } \phi} \\
&& \textit{α-conversion}
\end{array}$$

$$\begin{array}{rcl}
\iota_1\phi \to y.\psi; z.\chi &=& \psi[\phi/y] \\
\multicolumn{3}{l}{\text{if } \phi \text{ is free for } y \text{ in } \psi \text{ and modally free for } y \text{ in } \psi} \\
\iota_2\phi \to y.\psi; z.\chi &=& \chi[\phi/z] \\
\multicolumn{3}{l}{\text{if } \phi \text{ is free for } z \text{ in } \chi \text{ and modally free for } z \text{ in } \chi} \\
\pi_1(\phi, \psi) &=& \phi \\
\pi_2(\phi, \psi) &=& \psi \\
(\lambda x\phi\ \psi) &=& \phi[\psi/x] \\
\multicolumn{3}{l}{\text{if } \psi \text{ is free for } x \text{ in } \phi \text{, and modally free for } x \text{ in } \phi} \\
{}^{\vee\wedge}\phi &=& \phi \\
&& \textit{β-conversion}
\end{array}$$

$$\begin{array}{rcl}
(\pi_1\phi, \pi_2\phi) &=& \phi \\
\lambda x(\phi\ x) &=& \phi \\
\multicolumn{3}{l}{\text{if } x \text{ is not free in } \phi} \\
{}^{\wedge\vee}\phi &=& \phi \\
\multicolumn{3}{l}{\text{if } \phi \text{ is modally closed}} \\
&& \textit{η-conversion}
\end{array}$$

Figure 6.13: Semantic conversion laws

within the scope of an $^\wedge$. A modally closed term is denotationally invariant across worlds. We say that a term ψ is *modally free for x in ϕ* if and only if either ψ is modally closed, or no free occurrence of x in ϕ is within the scope of an $^\wedge$. The laws of conversion in Figure 6.13 obtain; for the sake of brevity we omit the so-called commuting conversions for the case statement.

6.3.1 More on Reflexives

Reflexive pronouns such as *himself/herself/itself* can take subject antecedents or object antecedents.

Subject-oriented reflexivization like

(330) John_i buys himself_i coffee.

is generated by assignment as follows, where here and throughout *VP* abbreviates $N\backslash S$, and as remarked earlier we allow ourselves to omit the subscript 1 for $\downarrow_1$ and $\uparrow_1$:

(331) **himself/herself/itself** : $\Box(((N\backslash S)\uparrow N)\downarrow(N\backslash S)) : {}^\wedge\lambda x\lambda y(x\ y\ y)$

Consider:

(332) a. *John_i believes Mary likes himself_i.

$$\frac{}{\overrightarrow{A}:x \Rightarrow A:x}\,id$$

$$\frac{\Gamma \Rightarrow A:\phi \qquad \Delta\langle\overrightarrow{C}:z\rangle \Rightarrow D:\omega}{\Delta\langle\Gamma,\overrightarrow{A\backslash C}:y\rangle \Rightarrow D:\omega[(y\ \phi)/z]}\,\backslash L \qquad \frac{\overrightarrow{A}:x,\Gamma \Rightarrow C:\chi}{\Gamma \Rightarrow A\backslash C:\lambda x\chi}\,\backslash R$$

$$\frac{\Gamma \Rightarrow B:\psi \qquad \Delta\langle\overrightarrow{C}:z\rangle \Rightarrow D:\omega}{\Delta\langle\overrightarrow{C/B}:x,\Gamma\rangle \Rightarrow D:\omega[(x\ \psi)/z]}\,/L \qquad \frac{\Gamma,\overrightarrow{B}:y \Rightarrow C:\chi}{\Gamma \Rightarrow C/B:\lambda y\chi}\,/R$$

$$\frac{\Delta\langle\overrightarrow{A}:x,\overrightarrow{B}:y\rangle \Rightarrow D:\omega}{\Delta\langle\overrightarrow{A\bullet B}:z\rangle \Rightarrow D:\omega[\pi_1 z/x,\pi_2 z/y]}\,\bullet L \qquad \frac{\Gamma_1 \Rightarrow A:\phi \qquad \Gamma_2 \Rightarrow B:\psi}{\Gamma_1,\Gamma_2 \Rightarrow A\bullet B:(\phi,\psi)}\,\bullet R$$

$$\frac{\Gamma \Rightarrow A:\phi \qquad \Delta\langle\overrightarrow{C}:z\rangle \Rightarrow D:\omega}{\Delta\langle\Gamma|_k\overrightarrow{A{\downarrow_k}C}:y\rangle \Rightarrow D:\omega[(y\ \phi)/z]}\,{\downarrow_k}L \qquad \frac{\overrightarrow{A}:x|_k\Gamma \Rightarrow C:\chi}{\Gamma \Rightarrow A{\downarrow_k}C:\lambda x\chi}\,{\downarrow_k}R$$

$$\frac{\Delta\langle\overrightarrow{A}:x,\overrightarrow{B}:y\rangle \Rightarrow D:\omega}{\Delta\langle\overrightarrow{A\odot_i B}:z\rangle \Rightarrow D:\omega[\pi_1 z/x,\pi_2 z/y]} \qquad \frac{\Gamma_1 \Rightarrow A:\phi \qquad \Gamma_2 \Rightarrow B:\psi}{\Gamma_1|_i\Gamma_2 \Rightarrow A\odot_i B:(\phi,\psi)}$$

$$\frac{\Gamma \Rightarrow B:\psi \qquad \Delta\langle\overrightarrow{C}:z\rangle \Rightarrow D:\omega}{\Delta\langle\overrightarrow{C{\uparrow_k}B}:x|_k\Gamma\rangle \Rightarrow D:\omega[(x\ \psi)/z]}\,{\uparrow_k}L \qquad \frac{\Gamma|_k\overrightarrow{B}:y \Rightarrow C:\chi}{\Gamma \Rightarrow C{\uparrow_k}B:\lambda y\chi}\,{\uparrow_k}R$$

$$\frac{\Gamma \Rightarrow A:\phi \qquad \Delta\langle A:x;B:y\rangle \Rightarrow D:\omega}{\Delta\langle\Gamma;B|A:z\rangle \Rightarrow D:\omega[\phi/x,(z\ \phi)/y]}\,|L$$

$$\frac{\Gamma\langle B_0:y_0;\ldots;B_n:y_n\rangle \Rightarrow D:\omega}{\Gamma\langle B_0|A:z_0;\ldots;B_n|A:z_n\rangle \Rightarrow D|A:\lambda x(\omega[(z_0\ x)/y_0,\ldots,(z_n\ x)/y_n])}\,|R$$

$$\frac{\Gamma\langle\overrightarrow{A}:x\rangle \Rightarrow B:\psi}{\Gamma\langle\overrightarrow{\Box A}:z\rangle \Rightarrow B:\psi[{}^{\vee}z/x]}\,\Box L \qquad \frac{\Box\Gamma \Rightarrow A:\phi}{\Box\Gamma \Rightarrow \Box A:{}^{\wedge}\phi}\,\Box R$$

$$\frac{\Gamma \Rightarrow A:\phi \qquad \not\vdash \Gamma \Rightarrow B:\_}{\Gamma \Rightarrow A-B:\phi}\,-R$$

Figure 6.14: Semantically labelled hypersequent calculus for extended **D**, Part I

$$\frac{\Gamma\langle\overrightarrow{A}:x\rangle \Rightarrow C:\varphi[x]}{\Gamma\langle\overrightarrow{A\&B}:z\rangle \Rightarrow C:\varphi[\pi_1 z]}\&L1 \qquad \frac{\Gamma\langle\overrightarrow{B}:y\rangle \Rightarrow C:\varphi[y]}{\Gamma\langle\overrightarrow{A\&B}:z\rangle \Rightarrow C:\varphi[\pi_2 z]}\&L2$$

$$\frac{\Gamma \Rightarrow A:\Phi \qquad \Gamma \Rightarrow B:\Psi}{\Gamma \Rightarrow A\&B:(\Phi,\Psi)}\&R$$

$$\frac{\Gamma\langle\overrightarrow{A}:x\rangle \Rightarrow B:\varphi_1 \qquad \Gamma\langle\overrightarrow{B}:y\rangle \Rightarrow B:\varphi_2}{\Gamma\langle\overrightarrow{A\oplus B}:z\rangle \Rightarrow C:(z\rightarrow x\cdot\varphi_1;y\cdot\varphi_2)}\oplus L$$

$$\frac{\Gamma \Rightarrow A:\Phi}{\Gamma \Rightarrow A\oplus B:\iota_1\Phi}\oplus R1 \qquad \frac{\Gamma \Rightarrow B:\Psi}{\Gamma \Rightarrow A\oplus B:\iota_2\Psi}\oplus R2$$

Figure 6.15: Semantically labelled sequent calculus for extended **D**, Part II

Crucially, **believes** is assigned:

$$\textbf{believes} : \Box((N\backslash S)/\Box S)$$

The subtype $\Box S$ in succedent polarity projects therefore a local domain ruling out long-distance reflexivization (cf. Principle A of Chomsky (1981)).

For object-oriented reflexivization such as

(333) John talked to Mary$_i$ about herself$_i$.

We assume assignment:

(334) **himself/herself/itself** : $\Box(((\mathrm{VP}{\uparrow}\mathrm{N}){\uparrow}_2\mathrm{N}){\downarrow}_2(\mathrm{VP}{\uparrow}\mathrm{N}))$: $^\wedge\lambda x\lambda y(x\ y\ y)$

This embodies a precedence condition on object-oriented reflexivization:

(335) *Mary revealed himself$_i$ to John$_i$.

The fact that the antecedent hypothetical subtype is not modalized prevents a clause non-local antecedent:

(336) *Mary notified the fact that John$_i$ won to himself$_i$.

6.3.2 Personal pronouns

We distinguish "external anaphora" in which the antecedent is intrasentential but outside the clause of the pronoun, or intersentential or extralinguistic, and "internal anaphora" in which the antecedent is within the clause of the pronoun or within a clause subordinate to that clause.

We assign to the nominative personal pronouns *he/she* as follows:

(337) **he/she** : $\Box((\Box S|N)/\Box\mathrm{VP})$: $^\wedge\lambda x\lambda y^\wedge(^\vee x\ y)$

This captures that nominative pronouns only appear in subject positions, and that they permit no internal anaphora (cf. Principle C of Chomsky (1981)):

(338) a. *He$_i$ likes John$_i$.
b. *He$_i$ believes John$_i$ flies.
c. *He$_i$ believes Mary likes John$_i$.

To the both nominative and accusative personal pronoun *it* we assign for external anaphora thus:

(339) **it** : $\Box(\Box(S{\uparrow}N){\downarrow}(\Box S|N)) : {}^\wedge\lambda x\lambda y^\wedge({}^\vee x\ y)$

This allows it to appear in both nominative and accusative positions.
To the accusative pronouns *him/her* we assign for external anaphora:

(340) **him/her** : $\Box(\Box((S{\uparrow}N) - (J{\bullet}\mathrm{VP})){\downarrow}(\Box S|N)) : {}^\wedge\lambda x\lambda y^\wedge({}^\vee x\ y)$

This represents that case in English is configurational and that the default case is accusative: the use of the difference operator (i.e. negation as failure) allows the accusative pronouns to appear anywhere except in subject position. For example, **John$_i$ thinks him$_i$ runs* blocks because 1+**runs**, although it is of type $(\Box)(S{\uparrow}N)$, is also of type $J{\bullet}\mathrm{VP}$.

Finally, for internal anaphora we assign thus to the accusative personal pronouns *him/her/it*:

(341) **him/her/it** : $\Box((((S{\uparrow}N){\uparrow}\Box N){-}(J{\bullet}(\mathrm{VP}{\uparrow}N))){\downarrow}_2(S{\uparrow}\Box N)) : {}^\wedge\lambda x\lambda y(x\ y\ {}^\vee y)$

A similar device as before limits the accusative pronouns to only non-subject positions. That the antecedent hypothetical subtype is modalized allows a non-clause-local internal antecedent (by contrast with the reflexivization (336)):

(342) The fact that Mary employed John$_i$ surprised him$_i$.

The type embodies a precedence constraint on internal anaphora:

(343)*Mary revealed him$_i$ to John$_i$.

And the negation ensures that the pronoun cannot take as antecedent the subject of its own clause (cf. Principle B of Chomsky (1981)):

(344)*John$_i$ likes him$_i$.

The exemplification for Binding English using the machinery we have presented can be found in Morrill and Valentín (*op. cit.*).

6.3.3 Some Binding Theory for Romance Languages:

In this subsection we review some interesting linguistic phenomena which can be observed in Romance languages. The main linguistic focus will be Catalan reflexives. The main reference used in this sketch of Romance reflexive binding theory is Alsina (1996), namely his book *The Role of Argument Structure in Grammar: Evidence from Romance.*

Alsina distinguishes in Catalan (and in fact in other Romance languages) two types of reflexive, which have remarkable syntactic and semantic differences (in the literal translations of linguistic examples we will denote the reflexive clitic as RF):

- The *reflexive clitic se.*[10]

[10] Phonological constraints may alter the pronunciation of this word and its spelling as the following examples show:

- The *reflexive syntactic anaphor a si mateix/a si mateixa.*[11]

Alsina adopts a theory which focuses on the argument structure and the mapping from argument structure to syntactic functions (such as subject, direct object, oblique, and so on). He is able to account in a modified Lexical Functional Grammar framework (LFG, see Bresnan (1982)) for several puzzling linguistic phenomena. One important feature of Alsina's work is his rejection of a strongly supported principle in the generative community: the so-called UTAH ("Uniformity of Theta Assignment Hypothesis", see Baker (1988)). UTAH basically consists of the claim that theta-roles are in a bijective mapping with syntactic functions. By allowing a many to one mapping, he is able to solve several problematic linguistic issues for generative-based theories.

Here we give a type-logical solution to what we call Alsina's puzzle on reflexives. Our account is formal contrary to Alsina's which is rather informal, but of course we do not attempt to explain in our type-logical approach the great variety of linguistic problems which Alsina is able to solve. We also formally analyze reflexive clitic climbing, which is an nondeterministic linguistic phenomenon because in a verbal complex apparently there is freedom for the site of insertion of the reflexive clitic, although with some constraints we will detail later. Our explanation of the linguistic cases makes essential use of the categorial apparatus developed in the previous section, namely **DCA**$\Box-$ with moreover atomic subtyping. Nevertheless, the necessary projection of domains for reflexives with the modal machinery is excluded.

Alsina's puzzle on reflexive clitic binding in Catalan

Catalan has a constraint on the reflexive syntactic anaphor (and in fact in all nominative/accusative clitic pronouns). This is the clitic doubling condition, i.e. in our case, the reflexive syntactic anaphor must cooccur with the reflexive clitic as the following contrast shows:

(346) a. La Maria *es* defensa *a si mateixa.*
The Maria RF defends herself
'Maria defends herself.'

b. *La Maria defensa *a si mateixa.*
The Maria defends herself

It is interesting to remark that the property of cooccurrence of both reflexive clitic and reflexive syntactic anaphor does not hold of all Romance languages as the case Italian (*op. cit.* Alsina p.243) shows:

(345) a) Es vol defensar.
RF wants defend
'He wants to defend himself.'

b) Vol defensar-se.

a. Wants defend RF

b. 'He wants to defend himself.'

In this account of the reflexive clitic we will skip over these details.

[11]The reflexive syntactic anaphor varies with respect to the gender of the antecedent.

(347) a. Ugo ha difeso se stesso.
Ugo has defended himself
'Ugo has defended himself.'

b. Ugo si è defeso.
Ugo RF is defended
'Ugo has defended himself.'

But it is important to point out that a simple reflexive sentence has a priori the optionality of cooccurring with a reflexive syntactic anaphor with no apparent semantic differences. Consider the following contrast:

(348) a. El Pere es defensa a si mateix.
The Pere RF defends himself
'Pere defends himself.'

b. El Pere es defensa.
The Pere RF defend
'Pere defends himself.'

Sentences a) and b) from (348) are synonymous. They share the same semantic form: (*defend peter peter*). But only in simple sentences there seems that both options are equivalent. Reflexive sentences with elliptical verb phrases show important semantic differences which hide deep syntactic differences. In a nutshell, Alsina's main claim is that the reflexive clitic without the cooccurrence of the reflexive syntactic anaphor has an effect of what he calls A-structure binding which essentially consists of giving the same index to both proto-agentive (P-A) and proto-patient (P-P) theta roles in the argument structure. This has the consequence that at the level of functional structure, or f-structure, there is only one syntactic function, i.e. the subject.[12] The object function is not present, but the external argument which has subject function has the property that both P-A and P-P are mapped onto it. In other words, a sentence like (348) b) is an intransitive sentence. We say that the reflexive clitic intransitivizes the sentence, i.e. there is a valence reduction effect.

But the case of the reflexive syntactic anaphor has another story. As we already said, it must cooccur with the reflexive clitic (as if the reflexive clitic were an agreement marker). Now the f-structure associated to a sentence like (348) a) has two syntactic functions, namely a subject and an object. They have according to the semantic theory Alsina adopts the same referential indexes, but crucially there is no A-structure binding at the level of argument structure. Despite the differences at the level of f-structures (and in fact at the level of c-structures), the sentences a) and b) from (348) are synonymous. But with more complex sentences, in particular with elliptical material, we can expect semantic differences. The following sentences from Alsina (*op. cit.* p. 246 and 247) show these differences:

[12] In fact by the mapping principles Alsina assumes, by what he call the subject condition, there has to be a subject function.

(349) 1) La Gertrudis es defensa a si mateixa millor que la Maria.

a. 'Gertrudis$_i$ defends herself$_i$ better than Maria (defends Gertrudis).'

b. 'Gertrudis$_i$ defends herself$_i$ better than (Gertrudis defends) Maria.'

c. 'Gertrudis$_i$ defends herself$_i$ better than Maria$_j$ (defends herself$_j$).'

(350) 2) La Gertrudis es defensa millor que la Maria.

a. '*Gertrudis$_i$ defends herself$_i$ better than Maria (defends Gertrudis).'

b. '*Gertrudis$_i$ defends herself$_i$ better than (Gertrudis defends) Maria.'

c. 'Gertrudis$_i$ defends herself$_i$ better than Maria$_j$ (defends herself$_j$).'

Readings a), b) and c) in sentence (349) are possible whereas in a similar sentence like (350) with only the reflexive clitic, only reading c) is possible. In the literature, readings a) and b) are called strict readings, and the c) reading, the sloppy reading. These differences in readings between sentences (349) and (350) constitute what we call Alsina's puzzle on reflexives.

We proceed now to account for Alsina's puzzle and clitic climbing phenomenon in our type-logical framework. In our account of Alsina's puzzle we assume for simplicity that there is no clitic climbing. We omit also the definite article which precedes proper names in Catalan. Later we show how to give adequate lexical entries for both clitic climbing and Alsina's puzzle.
Let us consider the following lexical entries:[13]

- **a+si+mateixa** : $SELF : \lambda P \lambda w.(P\ \ w\ \ w)$. Here $SELF$ abbreviates:[14]

$$SELF \stackrel{def}{=} ((\mathrm{N}\backslash \mathrm{S_r})\uparrow \mathrm{N}))\downarrow(\mathrm{N}\backslash \mathrm{S_r})$$

- **se** : CLSELF : λV.V, where CLSELF abbreviates $\mathrm{TV_r}/\mathrm{TV}$ and TV_r and TV abbreviate respectively $(\mathrm{N}\backslash \mathrm{S_r})/\mathrm{N}$ and $(\mathrm{N}\backslash \mathrm{S})/\mathrm{N}$. This lexical entry is necessary for accounting for clitic doubling with the syntactic reflexive **a+si+mateixa/a+si+mateix**.

- **mq** : COMPAR : $\lambda \mathrm{P} \lambda \mathrm{z}.\mathrm{BT}[(\pi_1 \mathrm{P}\ \ \pi_2 \mathrm{P}), (\pi_1 \mathrm{P}\ \ \mathrm{z})]$, where the semantic constant BT stands for *better than* and the syntactic constant **mq** abbreviates **millor+que** (better than). Here, the type abbreviation COMPAR stands for:

$$\mathrm{COMPAR} \stackrel{def}{=} ((\mathrm{S_r}\uparrow \mathrm{N}) \odot \mathrm{N})\backslash(\mathrm{S_r}/\mathrm{N}))$$

- **se** : SE : $\lambda \mathrm{V} \lambda \mathrm{w}.(\mathrm{V}\ \ \mathrm{w}\ \ \mathrm{w})$, where the type abbreviation SE stands for:

$$\mathrm{SE} \stackrel{def}{=} (\mathrm{N}\backslash \mathrm{S})/\mathrm{TV}$$

[13]In order to be consistent with the previous section, the types we are proposing should be intensionalized. It is only for commodity of the reader that we have avoided intensionalize the types in order to project syntactic domains for binding.

[14]The gender marking of the reflexive syntactic anaphor in Catalan is ignored.

Notice how SE has the role of intransitivizing, in absence of the syntactic reflexive, the transitive verb as Alsina's theory predicts. There is a valence reduction effect and consequently only one gramatical function is possible, namely the subject function, which in our framework (without atomic subtyping for case) corresponds to the subtype $N\backslash S$ of the type SE.

Observe the subtyping $\mathrm{S_r}$ which ensures that reflexive syntactic anaphor is infixing in a reflexive clitized transitive verb phrase as it is necessary in Catalan. It is then necessary to postulate the atomic axiom:

$$S_r \Rightarrow S$$

In Figure 6.16 we have the labelled ND hypersequent derivation of the sloppy reading of sentence (349), and in Figures 6.17 and 6.18 we find respectively the labelled ND and hypersequent derivations of the strict readings of sentence (349). These lexical entries we have proposed account for all the Alsina's predictions of what we have called Alsina's puzzle. If one inspects carefully the derivations in ND or hypersequent format, we realize that in the absence of the reflexive syntactic anaphor the sentence behaves as an intransitive sentence and that the reflexive clitic has a role of intransitivizer or reducer of the valence of the transitive verb, whereas in the sentences with reflexive clitic doubling, this has a role similar to an agreement marker. In the reflexive clitic doubling sentence the two possible strict readings depend on which argument of the transitive verb is extracted.

$$\dfrac{}{\textbf{es} : CLSELF : \lambda V.V}\,Lex \qquad \dfrac{}{\textbf{defensa} : TV : defend}\,Lex$$

$$\textbf{es+defensa} : TV_r : defend\ \ (/E) \qquad \textbf{a} : N : x_a$$

$$\textbf{es+defensa+a} : N\backslash S_r : (defend\ \ x_a)\ \ (/E)$$

$$\textbf{es+defensa+1} : (N\backslash S_r)\uparrow N : \lambda x_a.(defend\ \ x_a)\ \ (\uparrow I) \qquad \dfrac{}{\textbf{a+si+mateixa} : SELF : \lambda V.\lambda w.(V\ \ w\ \ w)}\,Lex$$

$$\textbf{es+defensa+a+si+mateixa} : (N\backslash S_r) : \lambda w.(defend\ \ w\ \ w)\ \ (\downarrow E) \qquad \textbf{b} : N : x_b$$

$$\textbf{b+es+defensa+a+si+mateixa} : S_r : (defend\ \ x_b\ \ x_b)\ \ (\backslash E)$$

$$\textbf{1+es+defensa+a+si+mateixa} : S_r\uparrow N : (\lambda x_b.defend\ \ x_b\ \ x_b)\ \ (\uparrow I) \qquad \dfrac{}{\textbf{gertrudis:N:gertrudis}}\,Lex$$

$$\textbf{gertrudis+es+defensa+a+si+mateixa} : (S_r\uparrow N)\odot N : (\lambda x_b.(defend\ \ x_b\ \ x_b), gertrudis)\ \ (\odot I)$$

$$\dfrac{}{\textbf{mq} : CMPAR : \lambda P.\lambda z.BT[(\lambda x_b.(defend\ \ x_b\ \ x_b)\ \ gertrudis, (\lambda x_b.(defend\ \ x_b\ \ x_b)\ \ gertrudis)]}\,Lex$$

$$\textbf{gertrudis+es+defensa+a+si+mateixa+mq} : S_r/N : \lambda z.BT[(defend\ \ gertrudis\ \ gertrudis), (defend\ \ z\ \ z)]\ (\backslash E)$$

$$\dfrac{}{\textbf{maria} : N : maria}\,N$$

$$\textbf{gertrudis+es+defensa+a+si+mateixa+mq+maria} : S_r :$$
$$BT[(defend\ \ gertrudis\ \ gertrudis), (defend\ \ maria\ \ maria)]\ (/E) \qquad S_r \Rightarrow S$$

$$\textbf{gertrudis+es+defensa+a+si+mateixa+mq+maria} : S :$$
$$BT[(defend\ \ gertrudis\ \ gertrudis), (defend\ \ maria\ \ maria)]\ (Cut)$$

Figure 6.16: Labelled ND derivation of the sloppy reading of sentence (349c)

$$\mathbf{a} : N : x_a \qquad \frac{}{\mathbf{defensa} : \mathrm{TV} : \mathrm{defend}}\ Lex \qquad \frac{}{\mathbf{g} : N : g}\ Lex \qquad \mathbf{b} : N : x_b \qquad \frac{}{\mathbf{es} : \mathrm{CLSELF} : \lambda \mathrm{V.V}}\ Lex$$

$\mathbf{es{+}defensa} : (N\backslash S_r)/N : defend\ \ (/E)$
$\mathbf{es{+}defensa{+}a} : N\backslash S_r : (defend\ \ x_a)\ \ (/E)$
$\mathbf{b{+}es{+}defensa{+}a} : S_r : (defend\ \ x_a\ \ x_b)\ \ (\backslash E)$

$\mathbf{1{+}es{+}defensa{+}a} : S_r{\uparrow}N : \lambda x_b(defend\ \ x_a\ \ x_b)\ \ {\uparrow}I \qquad \mathbf{t} : n : x_t$

$\mathbf{t{+}es{+}defensa{+}a} : (S_r{\uparrow}N) \odot N :< \lambda x_b.(defend\ \ x_a\ \ x_b), x_t >\ \ (\odot I)$

$$\frac{}{\mathbf{mq} : ((S_r{\uparrow}N) \odot N)\backslash(S_r/N) : \lambda P \lambda z.BT[(\pi_1 P\ \ \pi_2 P), (\pi_1 P\ \ z)]}\ Lex$$

$\mathbf{t{+}es{+}defensa{+}a{+}mq} : S_r/N : \lambda z.BT[((\lambda x_b.defend\ \ x_a\ \ x_b)\ \ x_t), (\lambda x_b.(defend\ \ x_a\ \ x_b)\ \ z)] =_\beta$
$\lambda z.BT[(defend\ \ x_a\ \ x_t), (defend\ \ x_a\ \ z))]\ (\backslash E) \qquad \frac{}{\mathbf{maria} : N : maria}\ Lex$

$\mathbf{t{+}es{+}defensa{+}a{+}mq{+}maria} : S_r : BT[((defend\ \ x_a\ \ x_t), (defend\ \ x_a\ \ maria))]\ (/E)$

$\mathbf{es{+}defensa{+}t{+}mq{+}maria} : N\backslash S_r : \lambda x_t.BT[((defend\ \ x_a\ \ x_t), (defend\ \ x_a\ \ maria))](\ \backslash I)$

$\mathbf{es{+}defensa{+}1{+}mq{+}maria} : (N\backslash S_r){\uparrow}N : \lambda x_a \lambda x_t.BT[(defend\ \ x_a\ \ x_t), (defend\ \ x_a\ \ maria))]\ ({\uparrow}I)$

Let $\Phi := \lambda x_a \lambda x_t.BT[(defend\ \ x_a\ \ x_t), (defend\ \ x_a\ \ maria))]$

$$\frac{}{\mathbf{a{+}si{+}mateixa} : ((N\backslash S_r){\uparrow}N){\downarrow}(N\backslash S_r) : \lambda P.w(P\ \ w\ \ w)}\ Lex$$

$\mathbf{es{+}defensa{+}a{+}si{+}mateixa{+}mq{+}maria} : N\backslash S_r : (\lambda P \lambda w(P\ \ w\ \ w)\ \ \Phi)\ =_\beta \lambda w.(\Phi\ \ w\ \ w)) =_\beta$
$\lambda w.BT[((defend\ \ w\ \ w), (defend\ \ w\ \ maria))]\ ({\downarrow}E)$

$$\frac{}{\mathbf{gertrudis} : N : gertrudis}\ Lex$$

$\mathbf{gertrudis{+}es{+}defensa{+}a{+}si{+}mateixa{+}mq{+}maria} : S_r :$
$BT[((defend\ \ gertrudis\ \ gertrudis), (defend\ \ gertrudis\ \ maria))]\ \ (\backslash E)$

$S_r \Rightarrow S$
$\mathbf{gertrudis{+}es{+}defensa{+}a{+}si{+}mateixa{+}mq{+}maria} : S :$
$BT[((defend\ \ gertrudis\ \ gertrudis), (defend\ \ gertrudis\ \ maria))]\ (Cut)$

Figure 6.17: Labelled ND derivation of the first strict reading of sentence (349a)

$$\mathbf{a}: N : x_a \qquad \frac{}{\mathbf{defensa}: \mathrm{TV} : \mathrm{defend}}Lex \qquad \frac{}{\mathbf{g}: N : g}Lex \qquad \mathbf{b}: N : x_b \qquad \frac{}{\mathbf{es}: \mathrm{CLSELF} : \lambda V.V}Lex$$

$\mathbf{es+defensa}: (N\backslash S_r)/N : defend$
$\mathbf{es+defensa+a}: N\backslash S_r : (defend\ \ x_a)\ \ (/E)$
$\mathbf{b+es+defensa+a}: S_r : (defend\ \ x_a\ \ x_b)\ \ (\backslash E)$

$\mathbf{b+es+defensa+1}: S_r{\uparrow}N : \lambda x_a(defend\ \ x_a\ \ x_b)\ \ (\backslash E) \qquad \mathbf{t}: n : x_t$

$\mathbf{b+es+defensa+t}: (S_r{\uparrow}N)\odot N : (\lambda x_a.(defend\ \ x_a)\ \ x_b), x_t)\ \ (\odot I)$

$$\frac{}{\mathbf{mq}: ((S_r{\uparrow}N)\odot N)\backslash(S_r/N) : \lambda P\lambda z.BT[(\pi_1 P\ \ \pi_2 P), (\pi_1 P\ \ z)]}Lex$$

$\mathbf{b+es+defensa+t+mq}: S_r/N : \lambda z.BT[((\lambda x_a.defend\ \ x_a\ \ x_b)\ \ t)), \lambda x_a.(defend\ \ x_a\ \ x_b)\ \ z)]\ (\backslash E)$

$\mathbf{b+es+defensa+t+mq}: S_r/N : \lambda z.BT[((defend\ \ x_t\ \ x_b))), (defend\ \ z\ \ x_b))] \qquad \frac{}{\mathbf{m}: N : maria}Lex$

$\mathbf{b+es+defensa+t+mq+maria}: S_r : BT[((defend\ \ x_t\ \ x_b))), (defend\ \ maria\ \ x_b))](\ \mathrm{E}\ /)$

$\mathbf{es+defensa+t+mq+maria}: N\backslash S_r : \lambda x_b.BT[((defend\ \ t\ \ x_b), (defend\ \ maria\ \ x_b))](\ \mathrm{I}\ \backslash)$

$\mathbf{es+defensa+1+mq+maria}: (N\backslash S_r){\uparrow}N : \lambda x_t\lambda x_b.BT[(defend\ \ x_t\ \ x_b), (defend\ \ maria\ \ x_b)]\ ({\uparrow}I)$

Let $\Phi := \lambda x_t\lambda x_b.BT[(defend\ \ x_t\ \ x_b), (defend\ \ maria\ \ x_b)]$

$$\frac{}{\mathbf{a+si+mateixa}: ((N\backslash S_r){\uparrow}N){\downarrow}(N\backslash S_r) : \lambda P\lambda w(P\ \ w\ \ w)}Lex$$

$\mathbf{es+defensa+a+si+mateixa+mq+maria}: N\backslash S_r : (\lambda P\lambda w(P\ \ w\ \ w)\ \ \Phi) =_\beta \lambda w(\Phi\ \ w\ \ w)(\downarrow E)$

$$\frac{}{\mathbf{gertrudis}: N : gertrudis}Lex$$

$\mathbf{gertrudis+es+defensa+a+si+mateixa+mq+maria}: S_r : (\lambda w(\Phi\ \ w\ \ w)\ \ gertrudis) =_\beta$
$(\Phi\ \ gertrudis\ \ gertrudis)) =_\beta BT[(defend\ \ gertrudis\ \ gertrudis), (defend\ \ maria\ \ gertrudis)]\ (\backslash E)$

$S_r \Rightarrow S$

$\mathbf{gertrudis+es+defensa+a+si+mateixa+mq+maria}: S :$
$BT[(defend\ \ gertrudis\ \ gertrudis), (defend\ \ maria\ \ gertrudis)]\ (Cut)$

Figure 6.18: Labelled ND derivation of the strict reading of sentence (349b)

$$\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{[\,], TV_r, N \Rightarrow S_r{\uparrow}N \qquad TV \Rightarrow TV}{[\,], CLSELF, TV, N \Rightarrow S_r{\uparrow}N}/L \qquad S_r/N, N \Rightarrow S_r}{N.CLSELF, TV, N \Rightarrow (S{\uparrow}N)\odot N}\odot R \qquad S_r/N, N \Rightarrow S}{N, CLSELF, TV, N, COMPAR, N \Rightarrow S_r}/L}{CLSELF, TV, N, COMPAR, N \Rightarrow N\backslash S_r}\backslash R}{CLSELF, TV, [\,], COMPAR, N \Rightarrow (N\backslash S_r){\uparrow}N}{\uparrow}R \qquad \cfrac{N \Rightarrow N \qquad S_r \Rightarrow S}{N, N\backslash S_r \Rightarrow S}\backslash L}{N, CLSELF, TV, SELF, COMPAR, N \Rightarrow S}{\downarrow}L$$

Figure 6.19: Hypersequent derivation of the strict reading of sentence (349a)

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{N, TV_r, [] \Rightarrow S_r{\uparrow}N \qquad TV \Rightarrow TV}{N, CLSELF, TV, [] \Rightarrow S_r{\uparrow}N}\,/L \qquad S_r/N, N \Rightarrow S_r}{N.CLSELF, TV, N \Rightarrow (S{\uparrow}N) \odot N}\,\odot R \qquad S_r/N, N \Rightarrow S}{N, CLSELF, TV, N, COMPAR, N \Rightarrow S_r}\,/L}{CLSELF, TV, N, COMPAR, N \Rightarrow N\backslash S_r}\,\backslash R}{CLSELF, TV, [], COMPAR, N \Rightarrow (N\backslash S_r){\uparrow}N}\,{\uparrow}R \qquad \dfrac{N \Rightarrow N \qquad S_r \Rightarrow S}{N, N\backslash S_r \Rightarrow S}\,\backslash L}{N, CLSELF, TV, SELF, COMPAR, N \Rightarrow S}\,{\downarrow}L$$

Figure 6.20: Hypersequent derivation of the strict reading of sentence (349b)

$$\dfrac{\dfrac{\dfrac{\dfrac{TV \Rightarrow TV \qquad \dfrac{TV_r, [] \Rightarrow S_r{\uparrow}N \qquad N, N\backslash S_r \Rightarrow S_r}{N, TV_r, SELF \Rightarrow S_r}\,{\downarrow}L}{N, TV_r/TV, TV, SELF \Rightarrow S_r}\,/L}{[], TV_r/TV, TV, SELF \Rightarrow S_r{\uparrow}N}\,{\uparrow}R \qquad N \Rightarrow N}{N, TV_r/TV, TV, SELF \Rightarrow (S_r{\uparrow}N) \odot N}\,\odot R \qquad \dfrac{N \Rightarrow N \qquad S_r \Rightarrow S}{S_r/N, N \Rightarrow S}\,/L}{N, CLSELF, TV, SELF, COMPAR, N \Rightarrow S}\,\backslash L$$

Figure 6.21: Hypersequent derivation of the sloppy reading of sentence (349c)

$$\dfrac{}{\textbf{es} : CLSELF : \lambda V.V}\ Lex \qquad \dfrac{}{\textbf{defensa} : TV : defend}\ Lex$$

$\textbf{es+defensa} : TV_r : defend\ \ (/E) \qquad \textbf{a} : N : x_a$

$\textbf{es+defensa+a} : N\backslash S_r : (defend\ \ x_a)\ \ (/E)$

$\textbf{es+defensa+1} : (N\backslash S_r)\uparrow N : \lambda x_a.(defend\ \ x_a)\ \ (\uparrow I) \qquad \textbf{a+si+mateixa} : SELF : \lambda V.\lambda w.(V\ \ w\ \ w)$

$\textbf{es+defensa+a+si+mateixa} : (N\backslash S_r) : \lambda w.(defend\ \ w\ \ w)\ \ (\downarrow E) \qquad \textbf{b} : N : x_b$

$\textbf{b+es+defensa+a+si+mateixa} : S_r : (defend\ \ x_b\ \ x_b)\ \ (\backslash E)$

$\textbf{1+es+defensa+a+si+mateixa} : S_r\uparrow N : (\lambda x_b.defend\ \ x_b\ \ x_b)\ \ (\uparrow I) \qquad \dfrac{}{\textbf{gertrudis:N:gertrudis}}\ Lex$

$\textbf{gertrudis+es+defensa+a+si+mateixa} : (S_r\uparrow N) \odot N :< \lambda x_b.defend\ \ x_b\ \ x_b), gertrudis >\ \ (\odot I)$

$$\dfrac{}{\textbf{mq} : CMPAR : \lambda P.\lambda z.BT[(\lambda x_b.(defend\ \ x_b\ \ x_b)\ \ gertrudis, (\lambda x_b.(defend\ \ x_b\ \ x_b)\ \ gertrudis)]}\ Lex$$

$\textbf{gertrudis+es+defensa+a+si+mateixa+mq} : S_r/N : \lambda z.BT[(defend\ \ gertrudis\ \ gertrudis), (defend\ \ z\ \ z)]\ (\backslash E)$

$$\dfrac{}{\textbf{maria} : N : maria}\ N$$

$\textbf{gertrudis+es+defensa+a+si+mateixa+mq+maria} : S_r :$
$BT[(defend\ \ gertrudis\ \ gertrudis), (defend\ \ maria\ \ maria)]\ (/E) \qquad S_r \Rightarrow S$

$\textbf{gertrudis+es+defensa+a+si+mateixa+mq+maria} : S :$
$BT[(defend\ \ gertrudis\ \ gertrudis), (defend\ \ maria\ \ maria)]\ (Cut)$

Figure 6.22: Labelled ND derivation of the sloppy reading of sentence (350) a.

Labelled natural Deduction has the attractive feature of inserting the lexical semantics through the derivation. Hypersequent calculus does not have this feature. We can nevertheless give the derivational semantics of a hypersequent derivation, and then substitute lexical semantics.[15] For example, the hypersequent derivation (349a) is the following:

$$(x_{\text{COMPAR}}(\lambda x_1.(x_{\text{SELF}}\ \ x_{\text{TV}}\ \ x_1), x_g)\ \ x_m)$$

Where the free variables of the above expression correspond to the lexical entries:

(351)

$$\begin{array}{lll} x_{\text{COMPAR}} & \rightsquigarrow & \lambda P \lambda z.BT[(\pi_1 P\ \ \pi_2 P), (\pi_1 P\ \ z)] \\ x_{\text{SELF}} & \rightsquigarrow & \lambda P \lambda w.(P\ \ w\ \ w) \\ x_g & \rightsquigarrow & gertudis \\ x_m & \rightsquigarrow & maria \end{array}$$

If in the above expression we replace every free variable by its corresponding lexical entry we get after β-reduction:

$$\begin{array}{l}(x_{\text{COMPAR}} < \lambda x_1.(x_{\text{SELF}}\ \ x_{\text{TV}}\ \ x_1), x_g >\ \ x_m) \rightsquigarrow \\ BT[(defend\ \ gertrudis\ \ gertrudis), (defend\ \ maria\ \ maria)]\end{array}$$

The hypersequent derivation corresponding to the strict reading (349) c) has the following derivational semantics:

$$(x_{\text{SELF}}\ \ (\lambda x_t.\lambda x_{subj}.(x_{COMPAR}\ \ (\lambda x_{obj}.(x_{TV}\ \ x_{obj}\ \ x_{subj}), x_t)\ \ x_m))\ \ x_g)$$

Substituting lexical semantics from (351) and β-reducing we get:

$$\begin{array}{l}(x_{\text{SELF}}\ \ (\lambda x_t.\lambda x_{subj}.(x_{COMPAR}\ \ < \lambda x_{obj}.(x_{TV}\ \ x_{obj}\ \ x_{subj}), x_t >\ \ x_m))\ \ x_g) \rightsquigarrow \\ BT[(defend\ \ gertrudis\ \ gertrudis), (defend\ \ maria\ \ gertrudis)]\end{array}$$

Hypersequent derivational semantics and lexical semantics substitution corresponding to the first strict reading (349) b) gives the expected semantics seen in the ND derivation (6.17).

Notice how the insertion of the reflexive syntactic anaphor type precedes the insertion of the comparative type in the ND and hypersequent derivations of the sloppy reading of sentence (349 c)), whereas in the corresponding strict readings of sentence (349) in both ND and hypersequent calculus derivations the comparative type insertion precedes the reflexive syntactic anaphor type insertion.

Accounting for reflexive clitic climbing in Catalan

When we treated Alsina's puzzle on reflexives in Catalan, we had to satisfy the Catalan constraint on reflexive syntactic anaphors which requires the cooccurrence of the reflexive clitic, (see Alsina *op.cit.*), i.e. *a si mateix/mateixa* without

[15]In the sentences with the reflexive syntactic anaphor we have avoided to include the semantic contribution of *CLSELF* in the derivational semantics, given the fact that *CLSELF* is a syntactic type whose function is to signal a kind of agreement between the reflexive syntactic anaphor and the reflexive clitic, and that the lexical semantics of *CLSELF* is the identity function.

the occurrence of the reflexive clitic renders the sentence ungramatical.

The type *CLSELF* used before was too much simplistic, for in particular it does not account for the reflexive clitic climbing property which holds of Catalan and several other Romance languages (see Bosque and Demonte (1999)). We recall the type definition of *CLSELF*:

$$CLSELF := TV_r/TV$$

Consider the following grammatical sentences. The type *CLSELF* for the lexical entry of the reflexive clitic only accepts as grammatical sentences b) and e):

(352) a. La Maria vol defensar-se.
The Maria wants defend RF
'Maria wants to defend herself.'

b. La Maria es vol defensar.
The Maria RF wants defend
'Maria wants to defend herself.'

c. La Maria vol poder defensar-se.
The Maria wants be_able defend RF
'Maria wants to be able to defend herself.'

d. La Maria vol poder-se defensar.
The Maria wants be_able RF defend
'Maria wants to be able to defend herself.'

e. La Maria es vol poder defensar.
The Maria RF wants be_able defend
'Maria wants to be able to defend herself.'

Moreover clitic climbing in Catalan has an important constraint: the reflexive clitic cannot occur between a finite verb and a non-finite verb, or occur after a single transitive verb, as the following contrast shows:[16]

(354) a. *La Maria defensa's.
The Maria defends RF

b. La Maria vol defensar-se.
The Maria wants defend RF
'Maria wants to defend herself.'

[16] It is interesting to remark that Iberian Spanish clitic climbing is very similar to the case of Catalan but the constraint we are referring to does not hold. Hence, it is possible to say:

(353) a. María defiéndese.
María defends RF
'María defends herself.'

b. María quiérese defender.
María wants RF defend
'Maria wants to defend herself.'

c. *La Maria vol es defensar.
The Maria wants RF defend

d. La Maria es vol defensar
. Maria RF wants defend
'Maria wants to defend herself.'

We propose now lexical entries (for clitic climbing) which use the machinery of **DCA**$\Box$-. We will use some basic type abbreviations:

- TV and $\mathrm{TV_r}$, as we already saw, respectively stand for $(N\backslash S)/N$ and $(N\backslash S_r)/N$.
- $\mathrm{TV_-}$ is an abbreviation for infinite transitive verb:

$$\mathrm{TV_-} \stackrel{def}{=} (\mathrm{S_-/N})/\mathrm{N}$$

- MV stands for a tensed modal verb:

$$\mathrm{MV} \stackrel{def}{=} (\mathrm{N\backslash S})/(\mathrm{S_-/N})$$

We give reflexive types for reflexive clitic double constructions: a) for a single reflexive clitic transitive verb, and b) for a reflexive complex transitive verb:

$$\mathrm{CLSELF_1} \stackrel{def}{=} (\mathrm{J} \bullet \mathrm{TV} - \check{}((\mathrm{TV}{\uparrow}\mathrm{MV}) \odot \mathrm{MV})){\downarrow}\mathrm{TV_r}$$
for a single transitive verb
$$\mathrm{CLSELF_2} \stackrel{def}{=} (\check{}((\mathrm{TV}{\uparrow}\mathrm{MV}) \odot \mathrm{MV})) - \mathrm{MV} \bullet \mathrm{J} \bullet \mathrm{TV_-}){\downarrow}\mathrm{TV_r}$$
for a complex transitive verb

In the case of a reflexive verb construction without clitic doubling, the following types are proposed. Here the type abbreviation used is SE because as we showed before there are important semantic differences between a reflexive construction without clitic doubling and a reflexive construction with clitic doubling:

$$\mathrm{SE_1} \stackrel{def}{=} (\mathrm{J} \bullet \mathrm{TV} - \check{}((\mathrm{TV}{\uparrow}\mathrm{MV}) \odot \mathrm{MV})){\downarrow}(\mathrm{N\backslash S})$$
for a single transitive verb
$$\mathrm{SE_2} \stackrel{def}{=} (\check{}((\mathrm{TV}{\uparrow}\mathrm{MV}) \odot \mathrm{MV})) - \mathrm{MV} \bullet \mathrm{J} \bullet \mathrm{TV_-}){\downarrow}(\mathrm{N\backslash S})$$
for a complex transitive verb

Useful type abbreviations for the *landing site* of the reflexive clitic in the types we have proposed above are (CLINS is an abbreviation of CLitical INSertion):

$$\mathrm{CLINS_1} \stackrel{def}{=} \mathrm{J} \bullet \mathrm{TV} - \check{}((\mathrm{TV}{\uparrow}\mathrm{MV}) \odot \mathrm{MV})$$
for a single transitive verb
$$\mathrm{CLINS_2} \stackrel{def}{=} \check{}((\mathrm{TV}{\uparrow}\mathrm{MV}) \odot \mathrm{MV}) - \mathrm{MV} \bullet \mathrm{J} \bullet \mathrm{TV_-}$$
for a complex transitive verb

Finally, we can give only one type which does the work of $\mathrm{CLSELF_1}$, $\mathrm{CLSELF_2}$, $\mathrm{SE_1}$ and $\mathrm{SE_2}$:

(355)

$$\mathrm{RFCL} \stackrel{def}{=} (\mathrm{CLINS_1} \oplus \mathrm{CLINS_2}) \downarrow (\mathrm{TV_r}\ \ \&\ \ \mathrm{N \backslash S})$$

The lexical semantics assigned to RFCL is:

$$\lambda Q \cdot Q \to [x]\langle x, \lambda w.(x\ \ w\ \ w)\rangle; [y]\langle(\pi_1 y\ \ \pi_2 y), \lambda w.((\pi_1 y\ \ \pi_2 y)\ \ w\ \ w)\rangle$$

We have to point out that relative pronouns such as who/which must have the subtype $\hat{}(S{\uparrow}N)$ modalized, i.e.:

$$\textbf{who/which} : \Box(\mathrm{CN \backslash CN})/\Box(\hat{}(\mathrm{S{\uparrow}N}))$$

In this way we successfully block reflexive clitic insertion in phrases as the following:

(356) a. **saw + the + man + who + ate** : TV

Some examples of reflexive clitic climbing

Consider the following sentences:

(357) a) La Gertrudis es defensa.
The Gertrudis RF defends
'Gertrudis defends herself.'

b) La Gertrudis es defensa a si mateixa.
The Gertrudis RF defends herself
'Gertrudis defends herself.'

c. *La Gertrudis defensa's.
The Gertrudis defends RF

d. La Gertrudis es vol defensar.
The Gertrudis RF wants defend
'Gertrudis wants to defend herself.'

e. *La Gertrudis vol-se defensar.
The Gertrudis wants RF defend

f. La Gertrudis vol defensar-se.
The Gertrudis wants defend RF. 'Gertrudis wants to defend herself.'

We give a **DCA**□- account for examples of (357):

- Hypersequent corrresponding to a):

$$\vdash \mathrm{N, RFCL, TV \Rightarrow S}$$

Proof:

$$\dfrac{\dfrac{[], TV \Rightarrow CLINS_1}{[], TV \Rightarrow CLINS_1 \oplus CLINS_2}\oplus R \qquad \dfrac{N, N\backslash S \Rightarrow S}{N, TV_r \& (N\backslash S) \Rightarrow S}\&L}{\mathrm{N, RFCL, TV \Rightarrow S}}\downarrow L$$

And $\vdash [], TV \Rightarrow CLINS_1$ for:

$$\vdash [], TV \Rightarrow J \bullet TV \text{ and}$$
$$\not\vdash [], TV \Rightarrow \check{}((TV{\uparrow}MV) \odot MV)$$

- Hypersequent corresponding to b):

$$\vdash \mathrm{N, RFCL, TV, SELF} \Rightarrow \mathrm{S}$$

Proof:

$$\dfrac{\dfrac{[], TV \Rightarrow CLINS_1}{[], TV \Rightarrow CLINS_1 \oplus CLINS_2}\oplus R \qquad \dfrac{\dfrac{N, TV_r, [] \Rightarrow (N\backslash S_r){\uparrow}N \qquad S_r \Rightarrow S}{N, TV_r, SELF \Rightarrow S}{\downarrow}L}{N, TV_r\&(N\backslash S), SELF \Rightarrow S}\&L}{\mathrm{N, RFCL, TV, SELF} \Rightarrow \mathrm{S}}{\downarrow}L$$

- Hypersequent corresponding to c):

$$\not\vdash N, TV, RFCL$$

Disproof:

$$\dfrac{\dfrac{\not\vdash TV, [] \Rightarrow CLINS_1 \qquad \not\vdash TV, [] \Rightarrow CLINS_2}{\not\vdash TV, [] \Rightarrow CLINS_1 \oplus CLINS_2}\oplus R \qquad \dfrac{N, N\backslash S \Rightarrow S}{N, TV_r\&(N\backslash S) \Rightarrow S}\&L}{\not\vdash \mathrm{N, TV, RFCL} \Rightarrow \mathrm{S}}{\downarrow}L$$

I.e. the last rule ${\downarrow}L$ cannot be applied, and the hypersequent corresponding to c) is not provable.

- Hypersequent corresponding to d):

$$\vdash N, RFCL, MV, TV_- \Rightarrow S$$

This hypersequent is provable for:

$$\vdash [], MV, TV_- \Rightarrow CLINS_2$$

Which is provable because:

$$\vdash [], MV, TV_- \Rightarrow \check{}((TV{\uparrow}MV) \odot MV) \text{ and}$$
$$\not\vdash [], MV, TV_- \Rightarrow MV \bullet J \bullet TV_-$$

- Hypersequent corresponding to e):

$$\not\vdash N, MV, RFCL, TV_- \Rightarrow S$$

For $\not\vdash MV, [], TV_- \Rightarrow CLINS_1$ and $\vdash MV, [], TV_- \Rightarrow \check{}((TV{\uparrow}MV) \odot MV)$ but:

$$\vdash MV[\,], TV_{-} \Rightarrow MV \bullet J \bullet TV_{-}$$

Whence $\nvdash MV, [\,], TV_{-} \Rightarrow CLINS_1 \oplus CLINS_2$.

f) Hypersequent corresponding to f):

$$\vdash N, MV, TV_{-}, RFCL, SELF \Rightarrow S$$

On the one hand we have that:

$$\frac{\vdash MV, TV_{-}, [\,] \Rightarrow CLINS_2}{\vdash MV, TV_{-}, [\,] \Rightarrow CLINS_1 \oplus CLINS_2} \oplus R$$

On the other hand:

$$\frac{\vdash N, TV_r, SELF \Rightarrow S}{\vdash N, (TV_r \& (N\backslash S)), SELF \Rightarrow S} \& L$$

Hence, applying the $\downarrow$ left rule we get:

$$\vdash N, MV, TV_{-}, RFCL, SELF \Rightarrow S$$

Chapter 7

Conclusions

For several years, the main goal of this author has been to solve open problems in the so-called wrapping approach (cf. Chapter 1 of this thesis), jointly with co-workers Glyn Morrill and Mario Fadda. In the wrapping approach of the 90's the attention was always focused on the algebraic side. The calculi which were proposed were labelled deductive systems with syntactic[1] and semantic annotation. The sequent systems were not ordered in the sense that the algebraic labelling was necessary to have all the information of the derivation of a sequent. Morrill (1997) gave a first formulation of a sequence logic (or ordered logic) which constitutes for this author a germ of the hypersequent syntax (in its segmented formulation; see Chapter 3) with an unlimited number of separators which appeared in Morrill, Fadda, and Valentín (2007). In that work the so-called hypersequent syntax was formally formulated with a clear sound syntactical interpretation. In the following papers to which this author contributed, several variations were proposed trying to find the *final* (hyper)sequent calculus.[2]

Nevertheless, in the wrapping approach of the 90's, there was no exploration of several important logical and computational aspects. Cut elimination theorems and completeness results were not given. Moreover, results on expressivity, generative capacity and decidability were also absent. But this is certainly not a criticism against this period. In the 90s Morrill and co-workers aspired to find the right and if possible definitive calculus which could satisfy the problem of discontinuity in natural languages. During these years of active and exciting research, the tools they disposed of were the classes of algebras they were proposing as well as the labelled deductive systems (in natural deduction form or in Gentzen sequent style)

It follows then that this thesis has aspired to give solutions to all these problems that simply were not explored:

- The underlying (sorted universal) algebraic machinery of the discontinuous Lambek calculus (see Chapter 2).

- A detailed proof theory with a syntactic proof of the Cut elimination

[1] In the 90's the term *prosodic algebra* was used instead of the term *syntactical algebra (the latter term was the one chosen in several papers after the year 2000 and in this thesis).*

[2] It was in Morrill, Valentín, and Fadda (2011) that the tree-based hypersequent syntax was proposed.

theorem for **D** and the extensions which have been proposed in Chapter 3.

- Several soundness and completeness results.
- A first approach to the expressive power of the discontinuous Lambek calculus.
- Finally, in Chapter 6 there have been exposed the theory at work with natural language examples.

The hypersequent calculus as said before was formally formulated for the first time in Morrill, Fadda, and Valentín (2007). The sequence logic, although non-standard (see Chapter 3), has no structural rules. But in this work we have demonstrated the intimate relation between the sorted multimodal calculus **mD** and the hypersequent calculus **hD**. In other words the hidden structural postulates which are absorbed in the hypersequent calculus have been discovered. We think that this is a remarkable finding.

7.1 Future Work and Open Problems

Is this thesis a kind of ending of the wrapping approach? We can say that we have tried to conclude it. We have formulated and proved several results. But as could be expected, many questions have arisen:

- Is it possible to extend **D** with the Moortgat's residuated unary modalities (or bracket modalities in Morrill's approach)? We already know that an absorbed syntax for the S4 modality can extend **D**.
- What happens with the sorting regime if brackets are added to **D**?
- Is Fadda and Morrill (2005)'s syntactical interpretation for bracket modalities compatible with sorts? Can we invent a new sorting system in order to reach compatibility with the syntactical interpretation of brackets?
- Is it possible to extend hypersequent syntax with another binary modes of composition. For example, we think of the morphology composition mode opposed to word level composition mode. This is particularly relevant in some Amerindian languages (e.g. Navajo) which are called polysynthetic, in which the morphological mode of composition is extremely rich and complex (see Baker (1996)).
- Following the previous item, can we give **D** structural rules? What would happen with the *syntax* of the hypersequent syntax and the sorting regime?
- Is a Pentus like theorem for **D**-grammars possible? We are sceptical. As we have proved, the class of the permutation closure of context-free languages is is recognized by **D**-grammars. For the time being, the Lambek calculus with permutation, i.e. **LP** does not have a Pentus like proof. And undoubtably, **LP** is simpler than **D**.

Bibliography

Ajdukiewicz, K. (1935). Die syntaktische konnexität. *Studia Philosophica 1*, 1–27. Translated in S. McCall, editor, 1967, *Polish Logic: 1920–1939*, Oxford University Press, Oxford, 207–231.

Alsina, A. (1996). *The Role of Argument Structure in Grammar. Evidence from Romance*. Leland Stanford Junior University: CSLI Lecture Notes.

Anoun, H. and A. Lecomte (2006). Linear grammars with labels. In *Proceedings of the Conference Formal Grammar*, Stanford.

Avron, A. (1991). Hypersequents, Logical Consequence and Intermediate Logic form Concurrency. *Annals of Mathematics and Artificial Intelligence 4*, 225–248.

Bach, E. (1981). Discontinuous constituents in generalized categorial grammars. In V. Burke and J. Pustejovsky (Eds.), *Proceedings of the 11th Annual Meeting of the North Eastern Linguistics Society, New York*, pp. 1–12. Amherst, Massachussets: GLSA Publications, Department of Linguistics, University of Massachussets at Amherst.

Bach, E. (1984). Some Generalizations of Categorial Grammars. In F. Landman and F. Veltman (Eds.), *Varieties of Formal Semantics: Proceedings of the Fourth Amsterdam Colloquium*, pp. 1–23. Dordrecht: Foris. Reprinted in Walter J. Savitch, Emmon Bach, William Marsh and Gila Safran-Naveh, editors, 1987, *The Formal Complexity of Natural Language*, D. Reidel, Dordrecht, 251–279.

Baker, M. (1988). *Incorporation: A Theory of Grammatical Function Changing.* Chicago: University of Chicago Press.

Baker, M. (1996). *The polysynthesis parameter.* New York: Oxford University Press.

Bar-Hillel, Y. (1953). A quasi-arithmetical notation for syntactic description. *Language 29*, 47–58.

Barker, C. (2002). Continuations and the nature of quantification. *Natural Language Semantics 10*, 211–242.

Bosque, I. and V. Demonte (1999). *Gramática descriptiva de la lengua española.* Espasa.

Bresnan, J. (1982). *The Mental Representation of Grammatical Relations.* MIT Press.

Bresnan, J. (2001). *Lexical Functional Syntax.* Basil Blackwell.

Buszkowski, W. (1982). Compatibility of categorial grammar with an associated category system. *Zeitschrift für mathematische Logik und Grundlagen der Mathematik 28*, 539–548.

Calcagno, M. (1995). A Sign-Based Extension to the Lambek Calculus for Discontinuous Constituency. *Bulletin of the IGPL 3*(4), 555–578.

Chomsky, N. (1957). *Syntactic Structures.* The Hague: Mouton.

Chomsky, N. (1981). *Lectures on Government and Binding: The Pisa Lectures.* Dordrecht: Foris Publications.

Culy, C. (1985). The Complexity of the Vocabulary of Bambara. *Linguistics and Philosophy 8*, 345—351. Reprinted in Walter J. Savitch, Emmon Bach, William Marsh and Gila Safran-Naveh, editors, 1987, *The Formal Complexity of Natural Language*, D. Reidel, Dordrecht, 349–357.

Curien, P. and H. Herbelin (2000). Duality of computation. In *Proceedings of the Fith AGM SIGPLAN*, Montreal.

De Groote, P. (2001). Towards abstract categorial grammars. In *ACL*, pp. 148–155.

Fadda, M. and G. Morrill (2005). The Lambek Calculus with Brackets. In C. Casadio, P. Scott, and R. Seely (Eds.), *Language and Grammar: Studies in Mathematical Linguistics and Natural Language*, Number 168 in CSLI Lecture Notes, pp. 113–128. Stanford: CSLI Publications.

Gazdar, G., E. Klein, G. Pullum, and I. Sag (1985). *Generalized Phrase Structure Grammar.* Oxford: Basil Blackwell.

Girard, J.-Y. (1987). Linear logic. *Theoretical Computer Science 50*, 1–102.

Girard, J.-Y. (2006). *Le Point Aveugle I*, Volume 1. Paris: Hermann éditeurs.

Hendriks, P. (1995). *Comparatives and Categorial Grammar.* Ph. D. thesis, Rijksuniversiteit Groningen, Groningen.

Hepple, M. (1990). *The Grammar and Processing of Order and Dependency.* Ph. D. thesis, University of Edinburgh.

Herbelin, H. (2005). *Au coeur de la dualité.* Ph. D. thesis, Université Paris 11.

Huybregts, M. (1976). Overlapping dependencies in Dutch. *Utrecht Working Papers in Linguistics 1*, 24–65.

Huybregts, R. (1985). The weak inadequacy of Context-Free Phrase Structure Grammars. In G. J. de Haan, M. Trommelen, and W. Zonneveld (Eds.), *Van Periferie naar Kern*, pp. 81–99. Dordrecht: Foris Publications.

Jacobson, P. (1999). Towards a variable-free semantics. *Linguistics and Philosophy 22*(2), 117–184.

Jäger, G. (2005). *Anaphora and Type Logical Grammar*, Volume 24 of *Trends in Logic – Studia Logica Library.* Dordrecht: Springer.

K Vijay-Shanker, D J Weir, A. K. J. (1986). Tree adjoining and head wrapping. In G. Bosch and Lang (Eds.), *Logic, Language, and Computation: 7th International Tbilisi Symposium, Revised Selected Papers*, Number 5422 in Springer Lecture Notes in Artificial Intelligence, Berlin, pp. 272–286. Springer.

Lalement, R. (1990). *Logique, réduction, résolution.* Masson. Études et recherches en informatique.

Lambek, J. (1958). The mathematics of sentence structure. *American Mathematical Monthly 65*, 154–170. Reprinted in Buszkowski, W., W. Marciszewski, and J. van Benthem, editors, 1988, *Categorial Grammar*, Linguistic & Literary Studies in Eastern Europe volume 25, John Benjamins, Amsterdam, 153–172.

Lambek, J. (1961). On the calculus of syntactic types. In R. Jakobson (Ed.), *Structure of Language and its Mathematical Aspects, Proceedings of the Symposia in Applied Mathematics XII*, pp. 166–178. Providence, Rhode Island: American Mathematical Society.

Lambek, J. (1988). Categorial and Categorical Grammars. In R. T. Oehrle, E. Bach, and D. Wheeler (Eds.), *Categorial Grammars and Natural Language Structures*, Volume 32 of *Studies in Linguistics and Philosophy*, pp. 297–317. Dordrecht: D. Reidel.

Lecomte, A. (2005). Categorial grammar for minimalism. In C. Casadio, P. Scott, and R. Seely (Eds.), *Language and Grammar: Studies in Mathematical Linguistics and Natural Language*, Number 168 in CSLI Lecture Notes, pp. 163–188. Stanford: CSLI Publications.

Lecomte, A. (2011). *Meaning, Logic and Ludics.* London: Imperial College Press.

Montague, R. (1970). Universal grammar. *Theoria 36*, 373–398. Reprinted in R.H. Thomason, editor, 1974, *Formal Philosophy: Selected Papers of Richard Montague*, Yale University Press, New Haven, 222–246.

Montague, R. (1973). The Proper Treatment of Quantification in Ordinary English. In J. Hintikka, J. Moravcsik, and P. Suppes (Eds.), *Approaches to Natural Language: Proceedings of the 1970 Stanford Workshop on Grammar and Semantics*, pp. 189–224. Dordrecht: D. Reidel. Reprinted in R.H. Thomason, editor, 1974, *Formal Philosophy: Selected Papers of Richard Montague*, Yale University Press, New Haven, 247–270.

Moortgat, M. (1988). *Categorial Investigations: Logical and Linguistic Aspects of the Lambek Calculus.* Foris, Dordrecht. PhD thesis, Universiteit van Amsterdam.

Moortgat, M. (1991). Generalized Quantification and Discontinuous Type Constructors. *Manuscript, Universiteit Utrecht*. Published in H. Bunt, editor, *Discontinuous Constituency*, De Gruyter, Berlin, 1996.

Moortgat, M. (1995). Multimodal linguistic inference. *Journal of Logic, Language and Information 5*, 349–385. Also in *Bulletin of the IGPL*, 3(2,3):371–401, 1995.

Moortgat, M. (1996). In situ binding: a modal analysis. In P. Dekker and M. Stokhof (Eds.), *Proceedings of the 10th Amsterdam Colloquium*, Amsterdam, pp. 235–240. ILLC, Universiteit van Amsterdam.

Moortgat, M. (1997). Categorial Type Logics. In J. van Benthem and A. ter Meulen (Eds.), *Handbook of Logic and Language*, pp. 93–177. Amsterdam and Cambridge, Massachusetts: Elsevier Science B.V. and The MIT Press.

Moortgat, M. (2009). Symmetric categorial grammar. *Journal of Philosophical Logic 38*, 681–710.

Morrill, G. (1990). Intensionality and Boundedness. *Linguistics and Philosophy 13*(6), 699–726.

Morrill, G. (1995). Discontinuity in Categorial Grammar. *Linguistics and Philosophy 18*(2), 175–219.

Morrill, G. (1997). Proof Syntax of Discontinuity. In P. Dekker, M. Stokhof, and Y. Venema (Eds.), *Proceedings of the 11th Amsterdam Colloquium*, Universiteit van Amsterdam, pp. 235–240. Institute for Logic, Language and Computation, ILLC.

Morrill, G. (2000a). Dutch Word Order and Binding. *Report de Recerca LSI-00-59-R, Departament de Llenguatges i Sistemes Informàtics, Universitat Politècnica de Catalunya*.

Morrill, G. (2000b). Incremental Processing and Acceptability. *Computational Linguistics 26*(3), 319–338.

Morrill, G. (2002). Towards Generalised Discontinuity. In G. Jäger, P. Monachesi, G. Penn, and S. Wintner (Eds.), *Proceedings of the 7th Conference on Formal Grammar*, Trento, pp. 103–111. ESSLLI.

Morrill, G. (2011a). Logic Programming of the Displacement Calculus. In S. Pogodalla and J.-P. Prost (Eds.), *Logical Aspects of Computational Linguistics*, Number 6736 in Lecture Notes in Artificial Intelligence, Berlin, pp. 175–189. Springer.

Morrill, G., M. Fadda, and O. Valentín (2007). Nondeterministic Discontinuous Lambek Calculus. In J. Geertzen, E. Thijsse, H. Bunt, and A. Schiffrin (Eds.), *Proceedings of the Seventh International Workshop on Computational Semantics, IWCS-7*, pp. 129–141. Tilburg University.

Morrill, G. and A. Gavarró (1992). Catalan Clitics. In A. Lecomte (Ed.), *Word Order in Categorial Grammar / L'Ordre des mots dans les grammaires catégorielles*, pp. 211–232. Clermont-Ferrand: Édicions Adosa.

Morrill, G. and J.-M. Merenciano (1996). Generalising discontinuity. *traitement automatique des langues 37*(2), 119–143.

Morrill, G. and M. T. Solias (1993). Tuples, Discontinuity and Gapping in Categorial Grammar. In *Proceedings of the European Chapter of the Association for Computational Linguistics, EACL93*, Utrecht, pp. 287–297.

Morrill, G. and O. Valentín (2010a). Displacement Calculus. *Linguistic Analysis 36*(1-4), 167–192.

Morrill, G. and O. Valentín (2010b). Generalized discontinuity. In *Proceedings of formal grammar 2010*, Copenhagen. To appear.

Morrill, G. and O. Valentín (2010c). On Anaphora and the Binding Principles in Categorial Grammar. In A. Dawar and R. de Queiroz (Eds.), *Proceedings of the 17th International Workshop on Logic, Language, Information and Computation, WoLLIC 2010, Brasília*, Number LNAI 6188 in Lecture Notes in Artificial Intelligence, Berlin, pp. 176–190. Springer.

Morrill, G. and O. Valentín (2010d). On calculus of displacement. In *Proceedings of TAG+Related Formalisms 2010*, Yale.

Morrill, G. and O. Valentín (2011). Displacement Logic for Anaphora. *Journal of Computer and System Sciences*. To appear.

Morrill, G., O. Valentín, and M. Fadda (2009). Dutch grammar and processing: A case study in tlg. In G. Bosch and Lang (Eds.), *Logic, Language, and Computation: 7th International Tbilisi Symposium, Revised Selected Papers*, Number 5422 in Springer Lecture Notes in Artificial Intelligence, Berlin, pp. 272–286. Springer.

Morrill, G., O. Valentín, and M. Fadda (2011). The Displacement Calculus. *Journal of Logic, Language and Information 20*(1), 1–48. Doi 10.1007/s10849-010-9129-2.

Morrill, G. V. (1994). *Type Logical Grammar: Categorial Logic of Signs*. Dordrecht: Kluwer Academic Press.

Morrill, G. V. (2011b). *Categorial Grammar: Logical Syntax, Semantics, and Processing*. Oxford: Oxford University Press.

Pentus, M. (1992). Lambek grammars are context-free. Technical report, Dept. Math. Logic, Steklov Math. Institute, Moskow. Also published as ILLC Report, University of Amsterdam, 1993, and in *Proceedings Eighth Annual IEEE Symposium on Logic in Computer Science*, Montreal, 1993.

Pentus, M. (1993). Lambek calculus is L-complete. *ILLC Report, University of Amsterdam*. Shortened version published as Language completeness of the Lambek calculus, *Proceedings of the Ninth Annual IEEE Symposium on Logic in Computer Science*, Paris, pages 487–496, 1994.

Pollard, C. (1984). *Generalized Phrase Structure Grammars, Head Grammars and Natural Language*. Ph. D. thesis, Stanford University.

Pollard, C. (2007). Convergent grammars. Technical report, Ohio State University.

Shieber, S. (1985). Evidence Against the Context-Freeness of Natural Language. *Linguistics and Philosophy 8*, 333–343. Reprinted in Walter J. Savitch, Emmon Bach, William Marsh and Gila Safran-Naveh, editors, 1987, *The Formal Complexity of Natural Language*, D. Reidel, Dordrecht, 320–334.

Solà, J., M. R. Lloret, J. Mascaró, and M. Pérez (1996). *Gramàtica del català contemporani*. Barcelona: Empúries.

Solias Arís, M. T. (1992). *Gramáticas Categoriales, Coordinación Generalizada y Elisión*. Ph. D. thesis, Universidad Autónoma de Madrid. Revised version published as *Gramática categorial: Modelos y aplicaciones*, Editorial Sintesis, Madrid, 1996.

Steedman, M. (1985). Dependency and Coordination in the Grammar of Dutch and English. *Language 61*, 523–568.

Steedman, M. (1990). Gapping as Constituent Coordination. *Linguistics and Philosophy 13*(2), 207–263.

Valentín, O. (2006). 1-Discontinuous Lambek Calculus: Type Logical Grammar and discontinuity in natural language. *DEA dissertation, Universitat Autònoma de Barcelona*. http://seneca.uab.es/ggt/tesis.htm.

van Benthem, J. (1983). The Semantics of Variety in Categorial Grammar. *Report 83–29*. Published in Buszkowski, W., W. Marciszewski, and J. van Benthem, editors, 1988, *Categorial Grammar*, Linguistic & Literary Studies in Eastern Europe Volume 25, John Benjamins, Amsterdam, 37–55.

van Benthem, J. (1991). *Language in action: Categories Lambdas, and Dynamic Logic*, Volume 130 of *Studies in Logic and the Foundations of Mathematics*. Amsterdam: North-Holland.

Versmissen, K. (1991). Discontinuous Type Constructors in Categorial Grammar. *MSc. dissertation, Universiteit Utrecht*.