2

A Logic for Categorial Grammars: Lambek's Syntactic Calculus

Summary. Our second chapter is a rather complete study of the Lambek calculus, which enables a completely logical treatment of categorial grammar.

We first present its syntax in full detail, both with sequent calculus and natural deduction, and explain the relationship between these two presentations. Then we turn our attention to the normal forms for such proofs. Normalization and its dual namely interpolation are not only pleasant mathematical properties; they also are key properties for the correspondence between Lambek grammars and more familiar phrase structure grammars; we give a detailed proof of the theorem of Pentus establishing the weak equivalence between context-free grammars and Lambek grammars.

In addition, we prove completeness for the Lambek calculus with respect to linguistically natural models: in these models categories are interpreted as subsets of a free monoid (eg. as strings of words or lexical items). Providing such a simple and natural interpretation provides another strong justification for the categorial approach.

2.1 Lambek's Syntactic Calculus and Lambek Grammars

We now turn our attention to the Lambek calculus (L) and Lambek grammars (LCG) which were introduced in the seminal paper (Lambek, 1958): we strongly recommend this paper to the reader.

The limitations of AB grammars, and the endless quest for new rules (composition, type raising, Geach laws, etc.) is a way to explain the interest of the Lambek calculus. Another is to place AB-grammar into a richer and more natural mathematical formalism.

A controversial but more interesting justification is the following: syntax is driven by resource consumption, which is neatly handled by resource conscious logics — the Lambek calculus being the first such logic. This viewpoint is not that far from Chomsky's minimalist program (Chomsky, 1995) as discussed in (Retoré and Stabler, 2004).

Lambek (categorial) grammars — or LCGs — are defined in a way very similar to AB grammars. A lexicon Lex provides each word with one or several types, constructed from the usual primitive types $\mathsf{P} = \{S, np, n, pp, \ldots\}$ — sentences, noun

R. Moot and C. Retoré: The Logic of Categorial Grammars 2012, LNCS 6850, pp. 23–63, 2012.

phrases, nouns, prepositional phrases... Types are more or less the same as the ones of AB grammars: the only difference is that Lambek types allow for a (non commutative) product or conjunction denoted by $\bullet$:

$$\mathsf{Lp} \quad ::= \quad \mathsf{P} \quad | \quad (\mathsf{Lp} \setminus \mathsf{Lp}) \quad | \quad (\mathsf{Lp} / \mathsf{Lp}) \quad | \quad (\mathsf{Lp} \bullet \mathsf{Lp})$$

When introducing AB grammars, we already explained the intuitive meaning of $A \setminus B$ and B / A: an expression is of type $A \setminus B$ (resp. B / A) when it is looking for an expression of type A on its left (resp. right) to form a compound expression of type B. An expression of type A followed by an expression B is of type $A \bullet B$, and product is related to $\setminus$ and $/$ by the following relations:

$$A \setminus (B \setminus X) = (B \bullet A) \setminus X \qquad\qquad (X / A) / B = X / (B \bullet A)$$

These relations look like currying, but beware of the order, which is required by the behavior of $\setminus$ and $/$: in the left equation both types require a sequence $a\,b$ on their left, and in the second equation both types require a sequence $b\,a$ on their right (with a, b of respective types A, B).

Recall that for AB grammars a sequence of words $w_1 \cdots w_n$, is of type u whenever there exists for each w_i a type t_i in $\mathrm{Lex}(w_i)$ such that $t_1 \cdots t_n \longrightarrow u$ with the following reduction patterns:

$$\forall u, v \in \mathsf{Lp} \qquad \begin{array}{ll} u\ (u \setminus v) \longrightarrow v & (\setminus_e) \\ (v / u)\ u \longrightarrow v & (/_e) \end{array}$$

Here the logical aspect of these rules — they look like modus ponens — will be emphasized by adding other rules, so that $\setminus$ and $/$ will really be implications (and $\bullet$ will be their associated conjunction). Accordingly $\longrightarrow$ will be written $\vdash$, and our first objective is to define this logical calculus: for the time being we only know the modus ponens of the non commutative implications $\setminus$ and $/$. Therefore we simply replace $\longrightarrow$ with $\vdash$ to obtain the following definition: a sequence of words (or terminals) $w_1 \cdots w_n$ is of type u whenever there exists for each w_i a type t_i in $\mathrm{Lex}(w_i)$ such that $t_1 \cdots t_n \vdash u$, where $\vdash$ is the deductive relation of the Lambek calculus which is defined in the next two sections. The generated language or the set of correct sentences is the set of sequences of type S.

2.2 Natural Deduction for the Lambek Calculus

To the best of our knowledge, natural deduction for Lambek has mainly been studied by van Benthem (van Benthem, 1991), one of the first papers being (van Benthem, 1987).

2.2.1 In Prawitz Style

The simplest way to define product the free Lambek calculus is probably natural deduction in a tree-like setting as shown below (we will have more to say about the requirement that the $\setminus_i$ and $/_i$ rules require at least two free hypotheses in Section 2.5).

this rule requires at least two free hyp.

$$\dfrac{\begin{array}{c}A \text{ leftmost free hyp.}\\ \ldots[A]\ldots\ldots\\ \vdots\\ B\end{array}}{A \setminus B}\ \setminus_i \text{ binding } A \qquad \dfrac{\begin{array}{cc}\Delta & \Gamma\\ \vdots & \vdots\\ A & A \setminus B\end{array}}{B}\ \setminus_e$$

this rule requires at least two free hyp.

$$\dfrac{\begin{array}{c}A \text{ rightmost free hyp.}\\ \ldots\ldots[A]\ldots\\ \vdots\\ B\end{array}}{B / A}\ /_i \text{ binding } A \qquad \dfrac{\begin{array}{cc}\Gamma & \Delta\\ \vdots & \vdots\\ B / A & A\end{array}}{B}\ /_e$$

These deductions clearly extend the derivation trees of AB grammars. AB simplification or elimination rules are two of the rules of the system, the rules $\setminus_e$ and $/_e$; the other two being the corresponding introduction rules. The fact that these rules are special cases of the rules for intuitionistic logic confirms that the fraction symbols $\setminus$ and $/$ can be viewed as implications.

It should be observed that as opposed to natural deduction for intuitionistic logic, there is no need to specify which hypothesis A is cancelled by an $/_i$ or $\setminus_i$ introduction rule. Indeed in the first case it is the leftmost free hypothesis, and in the second case it is the rightmost free hypothesis. As a consequence the formal structure of a deduction is a plain (binary/unary) tree with leaves labeled with formulae and with nodes labelled by rules: binary nodes are labelled with either $/_e$ or $\setminus_e$ and unary nodes with either $/_i$ or $\setminus_i$. Such a plain tree is enough to reconstruct the deduction, i.e. which hypothesis are free or not and which hypothesis is cancelled by which rule. This remark is the basis of the study of Tiede (2001); we can see the parse structures or proofs of a Lambek grammar as natural deduction trees, and study these trees as tree languages (Gécseg and Steinby, 1997).

Product

Lambek calculus admits a product which is related to the implications by the usual currying rules given above (or, alternatively, by the residuation rules discussed in Section 2.9.1). The product is often skipped in the natural deduction presentation of the Lambek calculus. There is no need to do so, but it is true that these rules are less natural, because of the order on hypotheses: Δ should occur in the place previously occupied by the cancelled (consecutive) A and B hypotheses of the rule.

$$\dfrac{\begin{array}{cc}\Delta & \Gamma\\ \vdots & \vdots\\ A & B\end{array}}{A \bullet B}\bullet_i \qquad\qquad \text{(no free hyp. between } A \text{ and } B\text{)}\quad \dfrac{\begin{array}{cc}\Delta & \ldots[A]_\alpha[B]_\alpha\ldots\\ \vdots & \vdots\\ A \bullet B & C\end{array}}{C}\bullet_e(\alpha) \text{ binding } A \text{ and } B$$

The main problem is that in order to apply the product elimination rule there should be no free hypothesis in between the two cancelled assumptions, A and B, and that the order of the premises after the rule is no longer the left-to-right order, but rather has the formulas Δ occurring at the place of the eliminated A and B formulas. Another problem is that, as we shall see, proof-normalization or rather the subformula property is more problematic with the product.

Also observe that there can be several consecutive free A and B hypotheses, so that a labeling (the label α in the rule above) is needed to link specific occurrences of cancelled hypotheses to the instance of the $\bullet_e$ rule: natural deductions are no longer plain trees.

Natural deduction rules of this form were first introduced by Abramsky (1993) for multiplicative linear logic, but in this commutative case the problem of the order of hypotheses disappears.

Example 2.1. As an example in Prawitz Style natural deduction, we give a proof of "Kevin talks to himself" below.

$$\dfrac{\text{Kevin}}{np}\text{Lex} \qquad \dfrac{\dfrac{\text{talks}}{(np\setminus S)/pp}\text{Lex} \qquad \dfrac{\dfrac{\text{to}}{pp/np}\text{Lex} \qquad [np]}{pp}/_e}{np\setminus S}/_e \qquad \dfrac{\text{himself}}{((np\setminus S)/np)\setminus(np\setminus S)}\text{Lex}}{S}\setminus_e \qquad \setminus_i$$

As already discussed in Example 1.1, the Lex rule used in the proof above is simply an indication that the conclusion formula F of the rule is an element of $\text{Lex}(w)$ for the premise w; that is, F is a formula that the lexicon Lex assigns to the word w.

The type for "himself" is assigned a category which selects a transitive verb to its left to produce an intransitive verb (as we will see in Example 3.2 in the next chapter, there are good semantic reasons for this type assignment).

Seen from the lexical types, the introduction of the np hypothesis is the only step which may not be immediately obvious: it is the type assigned to "himself", which, having a verb phrase $(np \setminus S)$ missing an np as its argument, introduces an np hypothesis. Section 2.6.2 will give a proof search algorithm for (product-free) natural deduction.

Exercises 2.2, 2.6, 2.8 and 2.9 at the end of this chapter will help you get familiar with finding natural deduction proofs for Lambek grammars.

2.2.2 In Gentzen Style

It is sometimes convenient to give a Gentzen style presentation of natural deduction, which specifies at each node what the free hypotheses are; this formulation is possibly clearer, in particular when formulating the rules for the product formulae. Figure 2.1 lists the rules in the calculus.

$$\frac{\Gamma \vdash A \quad \Delta \vdash A \setminus B}{\Gamma, \Delta \vdash B} \setminus_e \qquad \frac{A, \Gamma \vdash C}{\Gamma \vdash A \setminus C} \setminus_i \quad \Gamma \neq \varepsilon$$

$$\frac{\Delta \vdash B / A \quad \Gamma \vdash A}{\Delta, \Gamma \vdash B} /_e \qquad \frac{\Gamma, A \vdash C}{\Gamma \vdash C / A} /_i \quad \Gamma \neq \varepsilon$$

$$\frac{\Delta \vdash A \bullet B \quad \Gamma, A, B, \Gamma' \vdash C}{\Gamma, \Delta, \Gamma' \vdash C} \bullet_e \qquad \frac{\Delta \vdash A \quad \Gamma \vdash B}{\Delta, \Gamma \vdash A \bullet B} \bullet_i$$

$$\frac{}{A \vdash A}\ axiom$$

Fig. 2.1. Gentzen style natural deduction rules for the Lambek calculus

Nevertheless this presentation defines exactly the same logical calculus as the natural deduction rules in tree-like format given above: the proofs of the two systems are isomorphic.

A small note about the notation used in the calculus: the statements of the calculus are expressions of the form $A_1, \ldots, A_n \vdash C$ (sequents), with a comma-separated list of formulae (the *antecedent*, or the hypotheses of the statement) on the left hand side of the turnstile and a single formula on the right hand side (the *succedent* or the conclusion of the statement). Variables $\Gamma, \Delta, \ldots$ range over (possibly empty) lists of formulae which we will call *contexts*, so we can write a sequent as $\Gamma \vdash C$, or a sequent containing a formula A as $\Gamma, A, \Delta \vdash C$.

We will call the sequents above the horizontal line of a rule its *premises* and the single sequent below the horizontal line its conclusion. When given a proof, we will call the conclusion of the last rule the *end-sequent*.

Although we use sequents, this calculus is by no means a sequent calculus: there are no left rules, no cut rule, and the notion of normal proof (for having the subformula property) is completely different — as we will see in Sections 2.6 and 2.7.

Example 2.2 (Our Italian Lexicon Revisited)
Here we take up again our small example of an Italian lexicon:

Word	Type(s)
cosa	$(S/(S/np))$
guarda	(S/inf)
passare	(inf/np)
il	(np/n)
treno	n

Remember that the sentence '*Cosa guarda passare*' could not be analyzed in AB grammars, because the transitivity of / was not a rule of AB grammars. Let us show that it can be analyzed with the Lambek calculus (we use Natural Deduction in Gentzen style):

$$\dfrac{(S/(S/np)) \vdash (S/(S/np)) \qquad \dfrac{\dfrac{(S/inf) \vdash (S/inf) \qquad \dfrac{(inf/np) \vdash (inf/np) \quad np \vdash np}{(inf/np), np \vdash inf}\,/_e}{(S/inf), (inf/np), np \vdash S}\,/_e}{(S/inf), (inf/np) \vdash S/np}\,/_i}{(S/(S/np)), (S/inf), (inf/np) \vdash S}\,/_e$$

This example relies on composition for /, which is not provable in AB grammars. Composition is established by first hypothesizing an np which is then abstracted by an introduction rule: to make a comparison with Chomsky's theories (Chomsky, 1957, 1995) this hypothetical np corresponds to a trace and the introduction rule to movement.

In the Lambek calculus, we can construct sentences with object relatives such as *whom/that* having the type $(n \setminus n)/(S/np)$ — which could not be done in AB grammars, as shown in Section 1.5 — and without a need to assign $np \setminus (S/np)$ to transitive verbs. Indeed, we can derive $np \setminus (S/np)$ from the "normal" transitive verb type $(np \setminus S)/np$, since we can rearrange brackets in the Lambek calculus: $(a \setminus b)/c \vdash a \setminus (b/c)$, etc. This treatment of peripheral extraction is one of the attractive features of the Lambek calculus. Exercise 2.7 asks you to give proofs of more complicated cases of extraction.

Finally it is easily verified that one has $x \vdash (z/x) \setminus z$ and $x \vdash z/(x \setminus z)$ for all categories x and z. As we will see in the next chapter, this is interesting from a semantic viewpoint: an np (an individual) can be viewed as a $(S/np) \setminus S$ or $S/(np \setminus S)$ (a function form one-place predicates to truth values, that is the set of all the properties of this individual).

2.3 Sequent Calculus

Figure 2.2 shows the rules of the Lambek calculus in Sequent Calculus, as given in the original paper (Lambek, 1958). Although it also handles expressions $A_1, \ldots, A_n \vdash C$,

$$\frac{\Gamma, B, \Gamma' \vdash C \quad \Delta \vdash A}{\Gamma, \Delta, A \setminus B, \Gamma' \vdash C} \setminus_h \qquad \frac{A, \Gamma \vdash C}{\Gamma \vdash A \setminus C} \setminus_i \quad \Gamma \neq \varepsilon$$

$$\frac{\Gamma, B, \Gamma' \vdash C \quad \Delta \vdash A}{\Gamma, B / A, \Delta, \Gamma' \vdash C} /_h \qquad \frac{\Gamma, A \vdash C}{\Gamma \vdash C / A} /_i \quad \Gamma \neq \varepsilon$$

$$\frac{\Gamma, A, B, \Gamma' \vdash C}{\Gamma, A \bullet B, \Gamma' \vdash C} \bullet_h \qquad \frac{\Delta \vdash A \quad \Gamma \vdash B}{\Delta, \Gamma \vdash A \bullet B} \bullet_i$$

$$\frac{\Gamma \vdash A \quad \Delta_1, A, \Delta_2 \vdash B}{\Delta_1, \Gamma, \Delta_2 \vdash B} \mathit{cut} \qquad \frac{}{A \vdash A} \mathit{axiom}$$

Fig. 2.2. Sequent calculus rule for the Lambek calculus

let us insist that it is different from Natural Deduction in sequent style given above: for instance the modus ponens or elimination rules of AB grammars are not rules of this system (they are just derivable) and the notion of a normal proof is rather different (though (Girard et al, 1988) rightly remark there is a "moral equivalence" between the two).

A note on the names of the rules: we will use rule name $\setminus_h$, $/_h$ and $\bullet_h$ (for $\setminus$, $/$ and $\bullet$ used as a *hypothesis*) and $\setminus_i$, $/_i$, $\bullet_i$ (for the *introduction* rules for $\setminus$, $/$ and $\bullet$) instead of the frequently used $L\setminus$, $L/$ and $L\bullet$ (for the connectives on the *left*-hand side of the turnstile) and $R\setminus$, $R/$ and $R\bullet$ (for the connectives on the *right*-hand side of the turnstile). Our notation emphasizes that fact that the $\setminus_i$ rule is shared between the sequent calculus and natural deduction, and that the difference between sequent calculus and natural deduction is whether we add the $\setminus_e$, $/_e$ and $\bullet_e$ rules — as we do for natural deduction — or the $\setminus_h$, $/_h$ and $\bullet_h$ rules — as we do for the sequent calculus.

For a sequent calculus rule (or an instantiation of a rule as we find in an actual proof) it is normal to talk about the *main* formula of the rules for the logical connectives: the main formula is the formula with the connective introduced by the rule as its main connective; so the main formula of the $\bullet_h$ and $\bullet_i$ rule is the formula $A \bullet B$. We will call its direct subformulae, that is the formulae A and B, the *active* formulae of the rule (or the rule application).

Here is an obvious proposition (Exercise 2.3 at the end of this chapter asks you to prove this yourself).

Proposition 2.3. *Every axiom $A \vdash A$ can be derived from axioms $p \vdash p$, with p being a primitive type (and the proof does not use the cut rule).*

Definition 2.4 (Polarity). *The polarity of an occurrence of a propositional variable p in a formula is defined as usual:*

- *p is positive in p*
- *if p is positive in A, then*
  - *p is positive in* $X \bullet A$, $A \bullet X$, $X \setminus A$, A / X
  - *p is negative in* $A \setminus X$, X / A
- *if p is negative in A, then*
  - *p is negative in* $X \bullet A$, $A \bullet X$, $X \setminus A$, A / X
  - *p is positive in* $A \setminus X$, X / A

The polarity of an occurrence of a propositional variable p in a sequent $\Gamma \vdash C$ *is:*

- *if p is in C, the polarity of p in C*
- *if p is in a formula G of* Γ*, the opposite of the polarity of p in G.*

Example 2.5. Polarity is an important notion, which will return in Chapter 6. In a sequent

$$a / b, (a / b) \setminus (d / c) \vdash (d / c)$$

a occurs positively in a / b but negatively in $((a / b) \setminus (d / c)$

If a proof only uses atomic axioms (this is always possible, as said above) ie. $p \vdash p$ with p a primitive type, then one can follow these two occurrences of p, one being negative and the other positive and none of the rules changes the polarity of an occurrence of a primitive type. The two occurrences of p either lead to a cut formula (the formula which disappears from the conclusion of the cut rule) or to the end-sequent. Now observe that the cut rule cancels a formula in positive position (on the right) with the same formula in negative position (on the left), so that the same number of positive and negative occurrences of p disappear. Consequently:

Proposition 2.6. *Each propositional variable has exactly the same number of positive and negative occurrences in a provable sequent.*

Some authors call Proposition 2.6 the *count check* or *count invariant* (van Benthem, 1986; Moortgat, 1988; Roorda, 1991) and it can be used to eliminate sequents which fail to satisfy the condition of Proposition 2.6 at little computational cost.

Example 2.7. Here is an example of a proof in sequent calculus, corresponding to the analysis of '*Cosa guarda passare*' already given in natural deduction format in Example 2.2. It is somewhat less natural, but has other advantages, like an easier subformula property.

$$\dfrac{S \vdash S \quad \dfrac{\dfrac{\dfrac{S \vdash S \quad inf \vdash inf}{S / inf, inf \vdash S}\,/_h \quad np \vdash np}{S / inf, inf / np, np \vdash S}\,/_h}{S / inf, inf / np \vdash S / np}\,/_i}{(S / (S / np)), S / inf, inf / np \vdash S}\,/_h$$

2.4 Equivalence of Sequent Calculus and Natural Deduction

As we will see, this equivalence is absolutely clear as far as provability is concerned. In fact there is a correspondence for proofs as well, but it is not a straightforward isomorphism (Girard et al, 1988).

As the introduction rules are common to both formalisms, we just need to mimic elimination rules $\diamond_e$ in sequent calculus and left rules $\diamond_h$ in natural deduction (for $\diamond \in \{\backslash, /, \bullet\}$), and by induction on the height of the proofs the equivalence of both formalisms follows. This section is an easy adaptation of the results in (Girard et al, 1988) for intuitionistic logic.

2.4.1 From Natural Deduction to Sequent Calculus

It is possible to do "better" than the translation we provide here; indeed, when the natural deduction is normal, one can manage to obtain a cut-free proof, and this better translation is implicitly used when one uses proof nets for λ-calculus see e.g. (Girard, 1987; de Groote and Retoré, 1996)

Replace:	with:
$\dfrac{\Delta \vdash A \quad \Gamma \vdash A \backslash B}{\Delta, \Gamma \vdash B}\,\backslash_e$	$\dfrac{\Gamma \vdash A \backslash B \qquad \dfrac{\Delta \vdash A \qquad \dfrac{}{B \vdash B}\,ax}{\Delta, A \backslash B \vdash B}\,\backslash_h}{\Delta, \Gamma \vdash B}\,cut$
$\dfrac{\Gamma \vdash B / A \quad \Delta \vdash A}{\Gamma, \Delta \vdash B}\,/_e$	$\dfrac{\Gamma \vdash B / A \qquad \dfrac{\Delta \vdash A \qquad \dfrac{}{B \vdash B}\,ax}{B / A, \Delta \vdash B}\,/_h}{\Gamma, \Delta \vdash B}\,cut$
$\dfrac{\Gamma \vdash A \bullet B \quad \Delta, A, B, \Theta \vdash C}{\Delta, \Gamma, \Theta \vdash C}\,\bullet_e$	$\dfrac{\Gamma \vdash A \bullet B \qquad \dfrac{\Delta, A, B, \Theta \vdash C}{\Delta, A \bullet B, \Theta \vdash C}\,\bullet_h}{\Delta, \Gamma, \Theta \vdash C}\,cut$

2.4.2 From Sequent Calculus to Natural Deduction

By induction on the height of a sequent calculus proof, let us see that it can be turned into a natural deduction. As above, we will not exhibit a translation from cut free proofs to normal deductions, although it is possible.

- If the proof consists in an axiom, its translation is obvious.
- If the proof ends with an introduction rule, $\backslash_i$, $/_i$ or $\bullet_i$ by induction hypothesis we have a deduction of the premise(s) and as these rules also exist in natural deduction and the translation is obvious.
- If the proof ends with an $\backslash_h$ rule:

$$\dfrac{\begin{array}{cc}\vdots\,\gamma & \vdots\,\delta \\ \Gamma,B,\Gamma' \vdash C & \Delta \vdash A\end{array}}{\Gamma,\Delta,A\backslash B,\Gamma' \vdash C}\backslash_h$$

then by induction hypothesis we have two natural deduction proofs, γ^* of $\Gamma,B,\Gamma' \vdash C$ and δ^* of $\Delta \vdash A$ and a translation of the whole proof is:

$$\begin{array}{ccccc} & & \Delta & & \\ & & \vdots\,\delta^* & & \\ & & A \qquad A\backslash B & & \\ & & \dfrac{}{B}\backslash_e & & \\ \Gamma & & & & \Gamma' \\ & & \vdots\,\gamma^* & & \\ & & C & & \end{array}$$

- If the proof ends with $/_h$ we proceed symmetrically.
- If the proof ends with $\bullet_h$:

$$\dfrac{\begin{array}{c}\vdots\,\gamma \\ \Gamma,A,B,\Gamma' \vdash C\end{array}}{\Gamma,A\bullet B,\Gamma' \vdash C}\bullet_h$$

by induction hypothesis we have a proof γ^* of $\Gamma,A,B,\Gamma' \vdash C$ and a translation is the following:

$$\dfrac{\begin{array}{ccccc} \Gamma & A & B & \Gamma' \\ & & \vdots\,\gamma^* & \\ A\bullet B & & C & \end{array}}{C}\bullet_e$$

- If the proof ends with a cut:

$$\dfrac{\begin{array}{cc}\vdots\,\gamma & \vdots\,\delta \\ \Gamma \vdash X & \Delta,X,\Delta' \vdash C\end{array}}{C}\,cut$$

by induction hypothesis we have two natural deductions $\gamma^*$of $\Gamma \vdash X$ and δ^* of $\Delta,X,\Delta' \vdash C$ and a translation is:

$$\begin{array}{ccccc} & & \Gamma & & \\ & & \vdots\,\gamma^* & & \\ \Delta & & X & & \Delta' \\ & & \vdots\,\delta^* & & \\ & & C & & \end{array}$$

2.5 The Empty Sequence

In the formulation of the introduction rules, we have required that the antecedent contains at least two formulae: therefore the antecedent is never empty after an application of an introduction rule. By case inspection we see that this guarantees that the antecedent of a sequent (the sequence on the left of $\vdash$) never is empty in a proof.

This is justified by the intended meaning of the connectives. Indeed by assigning the type $A \setminus B$ to a word or an expression e, we mean that an expression a of type A is *required* before e to obtain an expression ae of type B. This would fail without the "no empty sequence" requirement.

To explain this, let L1 be the calculus L without this restriction. Indeed, assume A is a tautology of L1, i.e. $\vdash_{L1} A$ (*); now let Γ be a sequence of type $A \setminus B$, that is $\Gamma \vdash_{L1} A \setminus B$ (**). Then from (*) and (**) we can infer by $\setminus_e$ the sequent $\Gamma \vdash_{L1} B$ without any sequence preceding Γ. This can actually happen in natural language; indeed some expression, including all modifiers do have such a tautology type, like $X \setminus X$.

For instance, a natural type for English adjectives is n / n and thus *very* gets the type $(n / n) / (n / n)$: when applied to an adjective on its right, one obtains an adjective phrase. Without the exclusion of the empty sequence, one is able to analyze in L1 the expression *"a very book"* as a noun phrase: indeed the adjective following *very* can be provided by the empty sequence, since n / n is derivable in L1. Let us give the proof in L1 using Prawitz-style natural deduction (the rules Lex, with premise w and conclusion formula A, are axioms indicating that A is in $\mathrm{Lex}(w)$):

$$\dfrac{\dfrac{\mathrm{a}}{np / n}\,\mathrm{Lex} \qquad \dfrac{\dfrac{\dfrac{\mathrm{very}}{(n / n) / (n / n)}\,\mathrm{Lex} \qquad \dfrac{[n]_\alpha}{n / n}\,/_i{-}\alpha}{n / n}\,/_e \qquad \dfrac{\mathrm{book}}{n}\,\mathrm{Lex}}{n}\,/_e}{np}\,/_e$$

One may wonder why such a requirement was not needed for AB grammars. As AB grammars contain only elimination rules, no hypotheses are cancelled during a derivation, and since there are hypotheses at the beginning of every sub-analysis (the types of the words in the analyzed sequence) there always is at least one hypothesis.

2.6 Normalization of Natural Deduction

This section is also an easy adaptation of similar results presented in (Girard et al, 1988). Throughout this section, we will mostly be concerned with the product free case, a brief discussion about why normalization is more complicated in the presence of the product is found in Section 2.6.3.

2.6.1 Normalization for the Product-Free Lambek Calculus

A natural deduction is said to be normal whenever it does not contain an introduction rule followed by an elimination rule. There are two such possible configurations:

$$\dfrac{\dfrac{\dfrac{\ldots\ldots[A]_\alpha\ldots}{\begin{array}{c}\vdots\\ \delta'\\ \vdots\\ B\end{array}}}{B/A}/_i\alpha \qquad \begin{array}{c}\Delta\\ \vdots\\ \delta\\ \vdots\\ A\end{array}}{B}/_e \qquad\qquad \dfrac{\begin{array}{c}\Delta\\ \vdots\\ \delta\\ \vdots\\ A\end{array} \qquad \dfrac{\dfrac{\ldots[A]_\alpha\ldots\ldots}{\begin{array}{c}\vdots\\ \delta'\\ \vdots\\ B\end{array}}}{A\backslash B}\backslash_i\alpha}{B}\backslash_e$$

Remember that we call implication formulas B/A and $A\backslash B$ *functors* and the formulas A their *arguments*.

Now, whenever such a configuration appears, it can be reduced as follows:

1. find the hypothesis A which has been cancelled in the proof δ' of B under some hypotheses including A
2. replace this hypothesis with the proof δ of A

So the configurations above reduce to:

$$\begin{array}{c}\Delta\\ \vdots\\ \delta\\ \vdots\\ \ldots A\ldots\\ \vdots\\ \delta'\\ \vdots\\ B\end{array} \qquad\qquad \begin{array}{c}\Delta\\ \vdots\\ \delta\\ \vdots\\ \ldots A\ldots\\ \vdots\\ \delta'\\ \vdots\\ B\end{array}$$

Proposition 2.8. *Natural deduction for L without product enjoys strong normalization, that is there are no infinite reduction sequences.*

Proof. Observe that the size of the proof decreases in each reduction step. □

The proof of strong normalization is so simple because each introduction rule binds exactly one formula A and therefore we never copy nor delete the proof δ, ie. contraction and weakening are not valid in the Lambek calculus. In intuitionistic logic — where the introduction rule for the implication can discharge any number of hypotheses of the formula A — strong normalization is valid as well, but the proof is a bit more delicate (see Girard et al, 1988, Chapters 4 and 6 for details).

Proposition 2.9. *Normalization is a locally confluent process. In other words, if a proof d reduces, in one step, to two different proofs e and f then there exists a proof g such the both e and f reduce, in some number of steps, to g.*

Proof. If a proof d contains two redexes, they correspond to two elimination rules e' and e'' between sub-proofs corresponding to a functor f' applied to an argument a' and to a functor f'' applied to an argument a''. One of the following case applies:

- e'' is in a'
- e'' is in f'

- e' is in a''
- e' is in f''
- e' and e'' can not be compared.

Assume we reduce e'. The redex e'' which is not reduced possesses a unique trace $\bar{e}''$ in the reduced proof d'. Symmetrically if we reduce e'' the redex e' which is not reduced possesses a unique trace $\bar{e}$" in d''. If in d' we reduce $\bar{e}''$ we obtain a proof d''' but if in d'' we reduce $\bar{e}'$ we also obtain d'''. □

We will now show, by an easy induction on the proofs, that whenever a natural deduction is normal (that is without such configuration) each formula is a subformula of a free hypothesis or of the conclusion. More precisely. In order to establish this, let us introduce the notion of principal branch.

Let us call a *principal branch* leading to F a sequence $H_0, \ldots, H_n = F$ of formulae of a natural deduction tree such that:

- H_0 is a free hypothesis
- H_i is the principal premise — the one carrying the eliminated symbol — of an elimination rule whose conclusion is H_{i+1}
- H_n is F

Proposition 2.10. *Let d be a normal natural deduction (without product), then:*

1. *if d ends with an elimination then there is a principal branch leading to its conclusion*
2. *each formula in d is the subformula of a free hypothesis or of the conclusion*

Proof. By induction on d.

[axiom] If d is an axiom, (1) and (2) hold.

[$\backslash_i$ introduction] (1) holds by vacuity. Assume d is made out of d' by the introduction $\backslash_i$ rule: by induction hypothesis each formula in d' is a subformula of A, Γ (the free hypotheses under which B is proved) or a subformula of B; so it is true that each formula in d is a subformula of $\Gamma, A \setminus B$, since A and B are subformulae of $A \setminus B$.

[$\backslash_e$ elimination] Assume d is an elimination rule $\backslash_e$ applied to:

- d' with conclusion A and free hypotheses Γ
- d'' with conclusion $A \setminus B$ and free hypotheses Δ

(1) Since d is normal the last rule of d'' is an elimination: indeed, if it were an introduction rule then it would be a $\backslash_i$ introduction making a redex with the final elimination in d. As d'' ends with an elimination, by induction hypothesis, there is a principal branch leading from H_0 in Δ to $A \setminus B$, so d contains a principal branch leading to its conclusion B.

(2) By induction hypothesis

- all formulae in d' are subformula of A or Γ (the free hypotheses under which A is proved)
- all formulae in d'' are subformulae of $\Delta, A \setminus B$.

Because of the principal branch of d' leading to $A \setminus B$, the conclusion $A \setminus B$ of d' is a subformula of some H_0 in Δ. Thus each formulae in d is a subformula of Γ, Δ hence of Γ, Δ, B

[$/_i$ introduction] as $\setminus_i$ introduction.

[$/_e$ elimination] as $\setminus_e$ elimination □

Here is a proposition of (Cohen, 1967) that we shall use to prove that every context-free grammar is weakly equivalent to a Lambek grammar.

The order $o(A)$ of a formula A is the number of alternating implications, defined formally as follows.

Definition 2.11. *The order of a formula A is defined as follows.*

$$\begin{aligned} o(p) &= 0 \qquad \textit{when } p \textit{ is an atomic type} \\ o(A \setminus B) &= \max(o(A)+1, o(B)) \\ o(B / A) &= \max(o(A)+1, o(B)) \end{aligned}$$

Thus, the order of $(np \setminus S) / np$ is 1, but the order of $S / (np \setminus S)$ is two.

Proposition 2.12. *A provable sequent $A_1, \ldots, A_n \vdash p$ of the product free Lambek calculus with $o(A_i) \leq 1$ and p a primitive type (and therefore of order zero) is provable with $\setminus_e$ and $/_e$ only — in other words AB derivations and L derivations coincide when types are of order at most one.*

Proof. If $A_1, \ldots, A_n \vdash p$ is provable, then it has a normal proof. We claim that this normal proof must contain only $\setminus_e$ and $/_e$. We proceed by contradiction, so we assume that the normal deduction contains an introduction rule, and so there is a lowest introduction rule — one without any introduction rule below.

Let us consider an arbitrary lowest introduction I.

- If the chosen lowest introduction I is an $\setminus_i$ introduction leading from y to $b \setminus y$. This introduction cannot be the last rule, because the conclusion is a primitive type p. So this rule is followed by a an elimination rule E, and there are three possibilities:
  - If $b \setminus y$ is the principal premise of the elimination rule E, then the rule E is an $\setminus_e$ elimination rule other premise b; we then have a redex I, E and this conflicts with the deduction being normal.
  - If $b \setminus y$ is not the principal premise of the elimination rule E, then E is either an $\setminus_e$ elimination rule with principal premise being $(b \setminus y) \setminus z$ or an $/_e$ elimination rule with principal premise $z / (b \setminus y)$. In both cases the principal premise is of order at least two. This conflicts with d enjoying the subformula property which is forced by d being normal (previous Proposition 2.10).
- If the chosen lowest introduction I is an $/_i$ rule, the argument is symmetrical.

Therefore there is no lowest introduction, hence no introduction at all. □

2.6.2 Decidability of Natural Deduction

We can use the principal branch property and Proposition 2.10 to show that natural deduction in the product free Lambek calculus is decidable (we will prove the classic result of decidability for the Lambek calculus *with* product using cut elimination for the sequent calculus in Section 2.8). In order to do so, we first need to introduce some notation to facilitate talking about left and right arguments of a formula.

Given a formula C, and a sequence of length p of pairs consisting of a letter ε_i (where $\varepsilon_i \in \{l, r\}$) and a formula G_i we denote by

$$C[(\varepsilon_1, G_1), \ldots, (\varepsilon_p, G_p)]$$

the formula defined as follows:

if $p = 0$ $C[] = C$
if $\varepsilon_i = l$ $C[(\varepsilon_1, G_1), \ldots, (\varepsilon_p, G_p)] = G_p \setminus C[(\varepsilon_1, G_1), \ldots, (\varepsilon_{p-1}, G_{p-1})]$
if $\varepsilon_i = r$ $C[(\varepsilon_1, G_1), \ldots, (\varepsilon_p, G_p)] = C[(\varepsilon_1, G_1), \ldots, (\varepsilon_{p-1}, G_{p-1})] / G_p$

Figure 2.3 shows an example of a formula with its arguments in list-of-pairs form and its step-by-step conversion to the corresponding formula in standard Lambek calculus notation: note how G_1 is the most deeply embedded argument of C and therefore an element of the first pair on the list.

$$\begin{gathered}
\mathbf{C[}(r, G_1), (l, G_2), (r, G_3), (l, G_4)\mathbf{]} = \\
G_4 \setminus C[(r, G_1), (l, G_2), (r, G_3)] = \\
G_4 \setminus (C[(r, G_1), (l, G_2)] / G_3) = \\
G_4 \setminus (G_2 \setminus C[(r, G_1)]) / G_3) = \\
G_4 \setminus (G_2 \setminus (C[] / G_1)) / G_3) = \\
G_4 \setminus (G_2 \setminus (C / G_1)) / G_3)
\end{gathered}$$

Fig. 2.3. Example of $C[(\varepsilon_1, G_1), \ldots, (\varepsilon_p, G_p)]$

Similarly, assume we are given a proof d of $\Delta \vdash C[(\varepsilon_1, G_1), (\varepsilon_2, G_2), \ldots, (\varepsilon_p, G_p)]$ — in what follows, d will just be an axiom $C[\ldots] \vdash C[\ldots]$ — and n proofs d_i of $\Gamma_i \vdash G_i$. Let us call $l_1, l_2, ..., l_l$ (resp. $r_1, r_2, ..., r_{p-l}$) the subsequence of indices (hence it's an increasing sequence) such that $\varepsilon_i = l$ (resp. $\varepsilon_i = r$).

We write $d[(\varepsilon_1, d_1), \ldots, (\varepsilon_p, d_p)]$ for the following proof of

$$\Gamma_{l_1}, \Gamma_{l_2}, \ldots, \Gamma_{l_l}, \Delta, \Gamma_{r_{p-l}}, \Gamma_{r_{p-l-1}}, \ldots, \Gamma_{r_1} \vdash C$$

if $p = 0$ $d[] = d$
if $\varepsilon_i = l$ $d[(\varepsilon_1, d_1), \ldots, (\varepsilon_i, d_i)] = \setminus_e(d_i, d[(\varepsilon_1, d_1), \ldots, (\varepsilon_{i-1}, d_{i-1})])$.
if $\varepsilon_i = r$ $d[(\varepsilon_1, d_1), \ldots, (\varepsilon_i, d_i)] = /_e(d[(\varepsilon_1, d_1), \ldots, (\varepsilon_{i-1}, d_{i-1})], d_i)$.

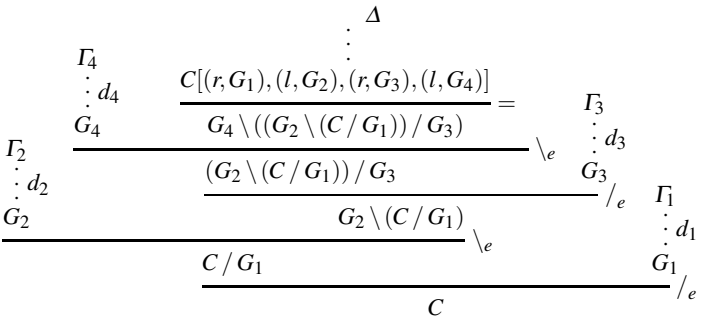

Fig. 2.4. Example: $d[(r,d_1),(l,d_2),(r,d_3),(l,d_4)]$ proving $\Gamma_2,\Gamma_4,\Delta,\Gamma_3,\Gamma_1$

Corollary 2.13. *Whenever a proof d of*

$$H_1,\ldots,H_n \vdash C$$

is normal and the last rule of the proof is an elimination rule, there exists an H_{i_0} which is equal to $C[(\varepsilon_1,G_1),\ldots,(\varepsilon_p,G_p)]$ and subproofs d_i of $\Gamma_i \vdash G_i$ such that d is $d[(\varepsilon_1,d_1),\ldots,(\varepsilon_p,d_p)]$.

Proof. Immediate from Proposition 2.10 □

Consequently, with the above notations l_i and r_j, $H_1,\ldots,H_{i_0-1} = \Gamma_{l_1},\Gamma_{l_2},\ldots,\Gamma_{l_l}$ and $H_{i_0+1},\ldots,H_n = \Gamma_{r_{p-l}},\Gamma_{r_{p-l-1}},\ldots,\Gamma_{r_1}$.

Let us prove, by induction on the number of connectives and atoms that $\Gamma \vdash C$ is provable in the Lambek calculus is a decidable question. Because of normalization, if there exists a proof, then there exists a normal proof.

If the (normal) proof ends with an introduction rule, C must be B/A (or $A\backslash B$) and proving $\Gamma \vdash B/A$ is equivalent to prove $\Gamma,A \vdash B$ (resp. $A,\Gamma \vdash B$) which is, by induction hypothesis, a decidable question.

Otherwise, the proof ends with an elimination rule. Because of Corollary 2.13 proving $\Gamma \vdash C$ is equivalent to prove a finite number of smaller sequents $\Gamma_i \vdash G_i$ which are obtained as follows: for all hypotheses H_{i_0}in Γ which are of the form $C[(\varepsilon_1,G_1),\ldots,(\varepsilon_p,G_p)]$, with l times $\varepsilon_i = l$ and r times $\varepsilon_i = r$, and for all partitions of the context $H_1,\ldots,H_{i_0-1}$ into l consecutive parts $\Gamma_{l_1},\Gamma_{l_2},\ldots,\Gamma_{l_l}$, and for all partitions of the context $H_{i_0+1},\ldots,H_n$ into r consecutive parts $\Gamma_{r_{p-l}},\Gamma_{r_{p-l-1}},\ldots,\Gamma_{r_1}$ consider the sequent $\Gamma_i \vdash G_i$, which has lesser atoms and connectives and therefore allows us to apply the induction hypothesis.

The above method also provides a decision procedure. The procedure, though simple, is actually rather effective — the only non-deterministic parts are the selection of a formula $C[\ldots]$ corresponding to the conclusion C and partitioning the context for the subproofs. We will improve upon it only in Chapter 6.

The natural deduction decision procedure discussed in this section is also rather close to so-called *normal form* sequent proofs, which are sequent proofs with some procedural restrictions as proposed by König (1989); Hepple (1990); Hendriks (1993) (compare the figure on page 210 of Hendriks, 1993, with our proof search algorithm above), though the correctness of natural deduction proof search is, in our opinion, quite a bit easier to prove using standard proof-theoretic notions such as principal branches. Hepple (1990), in the final section of his article, is the only one to make some brief remarks about a possible connection to natural deduction theorem proving.

2.6.3 Normalization and Lambek Calculus with Product

We have to introduce commutative conversions for the product, otherwise is is possible that a normal proof does not satisfy the subformula property:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{A \vdash A \qquad B \vdash B}{A, B \vdash A \bullet B}\bullet_i \qquad D \vdash D}{A, B, D \vdash (A \bullet B) \bullet D}\bullet_i}{A, B \vdash (A \bullet B) \bullet D / D}/_i \qquad A \bullet B \vdash A \bullet B}{A \bullet B \vdash (A \bullet B) \bullet D / D}\bullet_e \qquad D \vdash D}{A \bullet B, D \vdash (A \bullet B) \bullet D}/_e$$

Let us mention that this can be achieved by adding some "commutative conversions" which basically amount to putting product elimination rules as high as possible (just after the cancelled hypotheses A and B have met), and then rearranging the sub-trees made of product elimination rules with a kind of associativity so that the eliminated product never is the conclusion of another product elimination. Proving this result in full detail is a rather lengthy and technical exercise and is not (in our opinion) very insightful: this kind of result can also be deduced, though indirectly, from the correspondence with the sequent calculus.

2.7 Cut-Elimination for the Sequent Calculus

Cut elimination is the process under which a proof is turned into a proof of the same sequent *without any cut rule* — in other words, the cut rule is redundant.

Cut elimination is one of the fundamental properties of classical and intuitionistic logic, first proved by Gentzen (1934), see e.g. (Girard et al, 1988) for a modern proof and discussion. For L, it was originally proved in (Lambek, 1958).

Cut elimination has an important consequence, which we state before proving cut elimination:

Proposition 2.14. *In a cut-free proof of $A_1,\ldots,A_n \vdash A_{n+1}$ every formula of every sequent is a subformula of some formula A_i ($1 \leq i \leq n+1$).*

Proof. By case inspection it is easily observed that every rule of the sequent calculus except the cut rule satisfies the property that every formula in its premise sequent(s) is a subformula of some formula in its conclusion sequent. □

We give a syntactic proof of cut elimination (while models could be used as well): it is lengthy, tedious and without surprises, but one has to see this kind of proof at least once.

We begin by defining two notions on which we will base the induction. The *degree* of a cut and the *depth* of a cut.

Definition 2.15. *The* degree *of a formula A, written $d(A)$ (or simply d if the formula is clear from the context), is the depth of its subformula tree.*

$$\begin{aligned} d(p) &= 1 \qquad \textit{when } p \textit{ is a primitive type} \\ d(A \bullet B) &= \max(d(A), d(B)) + 1 \\ d(A \setminus B) &= \max(d(A), d(B)) + 1 \\ d(B / A) &= \max(d(A), d(B)) + 1 \end{aligned}$$

The degree of a cut *is the degree of the cut formula which disappears after application of the rule.*

Definition 2.16. *Let A by a formula which is eliminated by the application of a cut rule c. Let left(A) be the rule which introduces the formula occurrence A on the subproof which ends in the left premise of the cut rule and let right(A) be the rule which introduces the formula occurrence A in the subproof which ends at the right premise of the cut rule. The* depth *of a cut formula A, written $r(A)$ (or r if the cut formula is clear from the context), is the number of rules between left(A) and c plus the number of rules between right(A) and c.*

We show how to remove a single cut of smallest depth, reducing the total number of cut rules by one at each iteration until the proof is cut-free.

We select one of the cut rules with smallest depth and remove it as follows. We proceed by induction on (d,r) with $(d,r) < (d',r')$ if $d < d'$ or $d = d' \wedge r < r'$ where r is the depth of the cut rule, and d the maximal degree of a cut, assumed to be 0 when there is no cut.

$$\dfrac{\dfrac{\vdots\,\gamma}{\Gamma \vdash X}\,R^a \qquad \dfrac{\vdots\,\delta}{\Delta, X, \Delta' \vdash C}\,R^f}{\Delta, \Gamma, \Delta' \vdash C}\,cut\ d$$

Notice that because the last rule is a cut of smallest depth, neither R^a nor R^f is a cut rule and neither γ nor δ contain cut rules.

Before exploring all possible values for R^a and R^f, we will first give an overview of the general classes of the reductions; each pair of rules will fall into at least one of the following classes.

1. One of R^a or R^f is an axiom: both the cut and the axiom are suppressed.
2. R^a does not create the cut-formula, — so $R^a \neq \bullet_i, \backslash_i, /_i$. In this case it is possible to apply R^a after the cut. We can apply the induction hypothesis to the proof(s) with R^a and the cut rule reversed — since the depth r of the cut rule is less than the depth of the cut rule before reduction — and turn it into a cut-free proof.
3. If R^f does not create the cut formula, we proceed symmetrically.
4. If both R^a and R^f create the cut formula, then this cut of degree d is replaced by two cuts of strictly smaller degree. Hence, the maximal degree of a cut is strictly smaller (as the last rule was the only cut) and by induction hypothesis we are done.

We only describe the cases for $\backslash$ because the ones for $/$ are strictly symmetrical.

1 R^a or R^f is an axiom The final cut can be suppressed.

$$\dfrac{X \vdash X \qquad \begin{array}{c}\vdots\,\delta\\ \Gamma, X, \Delta \vdash C\end{array}}{\Gamma, X, \Delta \vdash C}\,cut \qquad \text{reduces to} \qquad \begin{array}{c}\vdots\,\delta\\ \Gamma, X, \Delta \vdash C\end{array}$$

2 R^a does not create X, the cut formula		
R^a	Before reduction	After reduction
$\bullet_h$	$\dfrac{\dfrac{\begin{array}{c}\vdots\,\gamma\\ \Gamma, A, B, \Gamma' \vdash X\end{array}}{\Gamma, A \bullet B, \Gamma' \vdash X}\bullet_h \qquad \begin{array}{c}\vdots\,\delta\\ \Delta, X, \Delta' \vdash C\end{array}}{\Delta, \Gamma, A \bullet B, \Gamma', \Delta' \vdash C}\,cut\ d$	$\dfrac{\dfrac{\begin{array}{c}\vdots\,\gamma\\ \Gamma, A, B, \Gamma' \vdash X\end{array} \qquad \begin{array}{c}\vdots\,\delta\\ \Delta, X, \Delta' \vdash C\end{array}}{\Delta, \Gamma, A, B, \Gamma', \Delta' \vdash C}\,cut\ d}{\Delta, \Gamma, A \bullet B, \Gamma', \Delta' \vdash C}\bullet_h$
$\backslash_h$	$\dfrac{\dfrac{\begin{array}{c}\vdots\,\delta\\ \Delta, B, \Delta'' \vdash X\end{array} \qquad \begin{array}{c}\vdots\,\delta'\\ \Delta' \vdash A\end{array}}{\Delta, \Delta', A \backslash B, \Delta'' \vdash X}\backslash_h \qquad \begin{array}{c}\vdots\,\gamma\\ \Gamma, X, \Gamma' \vdash C\end{array}}{\Gamma, \Delta, \Delta', A \backslash B, \Delta'', \Gamma' \vdash C}\,cut\ d$	$\dfrac{\begin{array}{c}\vdots\,\delta\\ \Delta, B, \Delta'' \vdash X\end{array} \qquad \dfrac{\begin{array}{c}\vdots\,\gamma\\ \Gamma, X, \Gamma' \vdash C\end{array} \qquad \begin{array}{c}\vdots\,\delta'\\ \Delta' \vdash A\end{array}}{\Gamma, \Delta, B, \Delta'', \Gamma' \vdash C}\,cut\ d}{\Gamma, \Delta, \Delta', A \backslash B, \Delta'', \Gamma' \vdash C}\backslash_h$

3 R^f **does not create** X**, the cut formula**		
R^f	Before reduction	After reduction
$\bullet_h$	$\dfrac{\Delta \vdash X \quad \dfrac{\Gamma, X, \Gamma', A, B, \Gamma'' \vdash C}{\Gamma, X, \Gamma', A \bullet B, \Gamma'' \vdash C}\bullet_h}{\Gamma, \Delta, \Gamma', A \bullet B, \Gamma'' \vdash C}\, cut\ d$	$\dfrac{\Delta \vdash X \quad \dfrac{\Gamma, X, \Gamma', A, B, \Gamma'' \vdash C}{\Gamma, \Delta, \Gamma', A, B, \Gamma'' \vdash C}\, cut\ d}{\Gamma, \Delta, \Gamma', A \bullet B, \Gamma'' \vdash C}\bullet_h$
$\bullet_h$	$\dfrac{\Delta \vdash X \quad \dfrac{\Gamma, A, B, \Gamma', X, \Gamma'' \vdash C}{\Gamma, A \bullet B, \Gamma', X, \Gamma'' \vdash C}\bullet_h}{\Gamma, A \bullet B, \Gamma', \Delta\Gamma'' \vdash C}\, cut\ d$	$\dfrac{\Delta \vdash X \quad \dfrac{\Gamma, X, \Gamma', A, B, \Gamma'' \vdash C}{\Gamma, \Delta, \Gamma', A, B, \Gamma'' \vdash C}\, cut\ d}{\Gamma, \Delta, \Gamma', A \bullet B, \Gamma'' \vdash C}\bullet_h$
$\backslash_h$	$\dfrac{\Delta \vdash X \quad \dfrac{\Gamma, B, \Gamma''' \vdash C \quad \Gamma', X, \Gamma'' \vdash A}{\Gamma, \Gamma', X, \Gamma'', A \backslash B, \Gamma''' \vdash C}\backslash_h}{\Gamma, \Gamma', \Delta, \Gamma'', A \backslash B, \Gamma''' \vdash C}\, cut\ d$	$\dfrac{\dfrac{\Gamma, B, \Gamma''' \vdash C \quad \dfrac{\Delta \vdash X \quad \Gamma', X, \Gamma'' \vdash A}{\Gamma', \Delta, \Gamma'' \vdash A}\, cut d}{\Gamma, \Gamma', \Delta, \Gamma'', A \backslash B, \Gamma''' \vdash C}\backslash_h$
$\backslash_h$	$\dfrac{\Delta \vdash X \quad \dfrac{\Gamma, B, \Gamma', X, \Gamma'' \vdash C \quad \Theta \vdash A}{\Gamma, \Theta, A \backslash B, \Gamma', X, \Gamma'' \vdash C}\backslash_h}{\Gamma, \Theta, A \backslash B, \Gamma', \Delta, \Gamma'' \vdash C}\, cut\ d$	$\dfrac{\dfrac{\Delta \vdash X \quad \dfrac{\Gamma, B, \Gamma', X, \Gamma'' \vdash C}{\Gamma, B, \Gamma', \Delta, \Gamma'' \vdash C}\, cut\ d \quad \Theta \vdash A}{\Gamma, \Theta, A \backslash B, \Gamma', \Delta, \Gamma'' \vdash C}\backslash_h$
$\bullet_i$	$\dfrac{\Delta \vdash X \quad \dfrac{\Gamma, X, \Gamma' \vdash A \quad \Theta \vdash B}{\Gamma, X, \Gamma', \Theta \vdash A \bullet B}\bullet_i}{\Gamma, \Delta, \Gamma', \Theta \vdash A \bullet B}\, cut\ d$	$\dfrac{\dfrac{\Delta \vdash X \quad \Gamma, X, \Gamma' \vdash A}{\Gamma, \Delta, \Gamma' \vdash A}\, cut\ d \quad \Theta \vdash B}{\Gamma, \Delta, \Gamma', \Theta \vdash A \bullet B}\bullet_i$
$\bullet_i$	$\dfrac{\Delta \vdash X \quad \dfrac{\Gamma \vdash A \quad \Theta, X, \Theta' \vdash B}{\Gamma, \Theta, X, \Theta' \vdash A \bullet B}\bullet_i}{\Gamma, \Theta, \Delta, \Theta' \vdash A \bullet B}\, cut\ d$	$\dfrac{\Gamma \vdash A \quad \dfrac{\Delta \vdash X \quad \Theta, X, \Theta' \vdash B}{\Theta, \Delta, \Theta' \vdash B}\, cut\ d}{\Gamma, \Theta, \Delta, \Theta' \vdash A \bullet B}\bullet_i$
$\backslash_i$	$\dfrac{\Delta \vdash X \quad \dfrac{A, \Gamma, X, \Gamma' \vdash B}{\Gamma, X, \Gamma' \vdash A \backslash B}\backslash_i}{\Gamma, \Delta, \Gamma' \vdash A \backslash B}\, cut\ d$	$\dfrac{\Delta \vdash X \quad \dfrac{A, \Gamma, X, \Gamma' \vdash B}{A, \Gamma, \Delta, \Gamma' \vdash B}\, cut\ d}{\Gamma, \Delta, \Gamma' \vdash A \backslash B}\backslash_i$

4 Both R^a and R^f create the cut-formula	
Before reduction	After reduction
$\bullet$ $\dfrac{\dfrac{\dfrac{\vdots\,\delta}{\Delta\vdash U}\quad\dfrac{\vdots\,\theta}{\Theta\vdash V}}{\Delta,\Theta\vdash U\bullet V}\bullet_i\quad\dfrac{\vdots\,\gamma}{\Gamma,U,V,\Gamma'\vdash C}}{\Gamma,\Delta,\Theta,\Gamma'\vdash C}\,cut\ d$	$\dfrac{\dfrac{\vdots\,\delta}{\Delta\vdash U}\quad\dfrac{\dfrac{\vdots\,\theta}{\Theta\vdash V}\quad\dfrac{\vdots\,\gamma}{\Gamma,U,V,\Gamma'\vdash C}}{\Gamma,U,\Theta,\Gamma'\vdash C}\,cut<d}{\Gamma,\Delta,\Theta,\Gamma'\vdash C}\,cut<d$
$\backslash$ $\dfrac{\dfrac{\dfrac{\vdots\,\delta}{U,\Delta\vdash V}}{\Delta\vdash U\backslash V}\backslash_i\quad\dfrac{\vdots\,\gamma}{\Gamma,V,\Gamma'\vdash C}\quad\dfrac{\vdots\,\theta}{\Theta\vdash U}\backslash_h}{\Gamma,\Theta,\Delta,\Gamma'\vdash C}\,cut\ d$	$\dfrac{\dfrac{\vdots\,\theta}{\Theta\vdash U}\quad\dfrac{\vdots\,\delta}{U,\Delta\vdash V}}{\Theta,\Delta\vdash V}\,cut<d\quad\dfrac{\vdots\,\gamma}{\Gamma,V,\Gamma'\vdash C}}{\Gamma,\Theta,\Delta,\Gamma'\vdash C}\,cut<d$

To be fully complete one should check that whenever the original proof contains no sequent with an empty antecedent, so does the cut free proof we inductively defined.

Now let us summarize what we have proved in this section:

Theorem 2.17. *Every proof of a given sequent $\Gamma\vdash C$ can be turned into a cut free proof of the same sequent — all formulae in the cut-free proof being subformulae of the sequent $\Gamma\vdash C$.*

When we compare normalization for natural deduction to cut elimination for the sequent calculus, we saw in Section 2.6.3 that natural deduction required commutative conversions only for the product formulas whereas the "commutative conversions" of the sequent calculus, which are items 2 and 3 of the case analysis for cut elimination above, make up a rather large part of the proof.

2.8 Decidability

One may wonder why we wanted to have normal or cut free proofs since the computational process of cut elimination or normalization is of little interest for categorial grammars.

What is nevertheless very interesting about such a result is that instead of looking for any proof when we want, for instance to parse and analyze a sentence, we can restrict our search space to these canonical proofs, either normal deductions or cut-free proofs. As we have seen, cut elimination (or normalization for natural deduction) entails the subformula property and this makes it quite easy to show that the calculus is decidable:

Proposition 2.18. *There is an algorithm which decides whether a sequent is derivable in L.*

Proof. Assume we want to prove a sequent. Since the cut rule is not needed, we have finitely many rules to try, each of these rules requiring us to prove one or two smaller sequents which, since they contain only subformulae of the sequent we want to prove, are also in finite number. □

Though sequent proof search is decidable, it is rather less efficient than the natural deduction algorithm we presented for the product-free calculus in Section 2.6.2. In particular, there can be many sequent calculus proofs which correspond to a single natural deduction proof and these different sequent calculus proofs differ only for 'bureaucratic' reasons. So though both normal natural deduction proofs and cut-free sequent proofs are canonical proofs in their respective calculi, there are many more of these canonical proofs in the sequent calculus than there are in natural deduction.

This problem of sequent calculus proof search is often called spurious ambiguity. On the other hand, the product formulas are unproblematic for cut-free sequent proof search. We will return to these issues in Chapter 6, were we will introduce proof nets for the Lambek calculus and show how they both handle product formulas unproblematically and make the problem of spurious ambiguity disappear.

2.9 Models for the Lambek Calculus and Completeness

We now turn our attention towards models for the Lambek calculus. As we have seen that as far as provability is concerned, cut-free sequent calculus, sequent calculus and natural deduction are equivalent, we are going to use the most adequate formalism to establish properties of models with respect to the deductive system.

These models have been first investigated in (Buszkowski, 1982) and our presentation follows (Buszkowski, 1997).

As we have said Lambek calculus prohibits the empty sequence, and we will present models for L with this restriction. Let us nevertheless say that all these results can be adapted by adding a unit to residuated semi-groups and to semi-groups — replacing the word "semi-group" with the word "monoid".

2.9.1 Residuated Semi-groups

Let us call a *residuated semi-group*, a structure $(M, \circ, \backslash\backslash, /\!/, \sqsubset)$ where

- M is a set.
- $\circ$ is an associative composition over M — $(M, \circ)$ is a semi-group.
- $\backslash\backslash$ and $/\!/$ are binary composition laws on M.
- $\sqsubset$ is an order on M.

which satisfies the following property:

(RSG) The following order relations are equivalent:

$$\begin{aligned} a &\sqsubset (c /\!/ b) \\ (a \circ b) &\sqsubset c \\ b &\sqsubset (a \backslash\backslash c) \end{aligned}$$

Proposition 2.19. *In a residuated semi-group $(M,\circ,\backslash\backslash,//,\sqsubset)$, for all $a,b,x,y \in M$ one has:*

1. $a \sqsubset b \quad \Rightarrow \quad (a\circ x) \sqsubset (b\circ x)$
2. $a \sqsubset b \quad \Rightarrow \quad (x\circ a) \sqsubset (x\circ b)$
3. $\begin{pmatrix} a \sqsubset b \\ \textit{and} \\ x \sqsubset y \end{pmatrix} \quad \Rightarrow \quad (a\circ x) \sqsubset (b\circ y)$

In other words, a residuated semi-group is in particular an ordered semi-group.

Proof. (1) From $(b\circ x) \sqsubset (b\circ x)$ ($\sqsubset$ is an order) (RSG) yields $b \sqsubset ((b\circ x)\,//\,x)$; if we assume $a \sqsubset b$ by transitivity of $\sqsubset$ we have $a \sqsubset ((b\circ x)\,//\,x)$ which by (RSG) yields $(a\circ x) \sqsubset (b\circ x)$.
(2) From $(x\circ b) \sqsubset (x\circ b)$ ($\sqsubset$ is an order) (RSG) yields $b \sqsubset (x \backslash\backslash (x\circ b))$; if we assume $a \sqsubset b$ by transitivity of $\sqsubset$ we have $a \sqsubset (x \backslash\backslash (x\circ b))$ which by (RSG) yields $(x\circ a) \sqsubset (x\circ b)$.
(3) The assumption $a \sqsubset b$ yields $(a\circ x) \sqsubset (b\circ x)$ (*) by (1). The assumption $x \sqsubset y$ yields $(b\circ x) \sqsubset (b\circ y)$ (**) by (2). By transitivity of $\sqsubset$, (*) and (**) yields $(a\circ x) \sqsubset (b\circ y)$. □

Given a residuated semi-group, an *interpretation* $[.]$ is a map from primitive types to elements in M, which extends to types and sequences of types in the obvious way:

$$[A,B] = [A]\circ[B] \qquad [A \setminus B] = [A] \backslash\backslash [B]$$
$$[A \bullet B] = [A]\circ[B] \qquad [B / A] = [B] // [A]$$

A sequent $\Gamma \vdash C$ is said to be *valid* in a residuated semi-group whenever $[\Gamma] \sqsubset [C]$.

2.9.2 The Free Group Model

A particular case of residuated semi-group is the free group over primitive types. It will be especially important in Section 2.11. The free group interpretation for L is

- a particular residuated semi-group where
  - $(M,\cdot)$ is the free group over the propositional variables,
  - $a \backslash\backslash b$ is $a^{-1}b$
  - $b // a$ is ba^{-1}
  - $a \sqsubset b$ is $a = b$ (the discrete order)

  One easily observes that the three equalities

  $$ab = c \qquad a = cb^{-1} \qquad b = a^{-1}c$$

  are equivalent — so (RSG) holds.
- a standard interpretation defined by $[p] = p$

Because of the soundness of L w.r.t. residuated semi-groups (next proposition) whenever a sequent $\Gamma \vdash C$ is provable one has $[\Gamma] = [C]$ in the free group. The free group model is of course not complete: indeed it interprets $\vdash$ by a symmetrical relation ($=$) while $\vdash$ is not symmetrical: $n \vdash s / (n \setminus s)$ is provable but not $s / (n \setminus s) \vdash n$.

2.9.3 L Is Sound and Complete w.r.t. Residuated Semi-groups

Proposition 2.20. *A provable sequent is valid in every residuated semi-group, for every interpretation of the primitive types.*

Proof. We proceed by induction on the natural deduction proof.

- If the proof consists in an axiom $X \vdash X$ then the result is true: $[X] \sqsubset [X]$ whatever the semi-group or the interpretation is.
- If the last rule is the introduction rule $\backslash_i$:

$$\frac{A, \Gamma \vdash C}{\Gamma \vdash A \backslash C} \backslash_i \quad \Gamma \neq \varepsilon$$

by induction hypothesis we have $[A] \circ [\Gamma] \sqsubset [C]$, thus, by (RSG) we have $[\Gamma] \sqsubset ([A] \backslash\backslash [C])$, so the sequent $\Gamma \vdash A \backslash C$ is valid as well.

If the last rule is the introduction rule $/_i$ we proceed as for $\backslash_i$.

If the last rule is the elimination rule $\backslash_e$:

$$\frac{\Gamma \vdash A \quad \Delta \vdash A \backslash B}{\Gamma, \Delta \vdash B} \backslash_e$$

then by induction hypothesis we know that $[\Gamma] \sqsubset [A]$, and using Proposition 2.19 we can conclude $[\Gamma] \circ [\Delta] \sqsubset [A] \circ [\Delta]$ (1); we also have $[\Delta] \sqsubset [A] \backslash\backslash [B]$ — hence by (RSG) $([A] \circ [\Delta]) \sqsubset [B]$ (2). Therefore from (1) and (2) we obtain,

$$[\Gamma, \Delta] \quad = \quad [\Gamma] \circ [\Delta] \quad \sqsubset (1) \quad [A] \circ [\Delta] \quad \sqsubset (2) \quad [B]$$

- If the last rule is the elimination rule $/_e$ we proceed as for $/_i$.
- If the last rule is the product elimination rule $\bullet_e$

$$\frac{\Gamma \vdash A \bullet B \quad \Delta, A, B, \Delta' \vdash C}{\Delta, \Gamma, \Delta' \vdash C} \bullet_e$$

By induction hypothesis we know that $[\Gamma] \sqsubset [A \bullet B] = [A] \circ [B]$, and, using Proposition 2.19 we obtain $[\Delta] \circ [\Gamma] \circ [\Delta'] \sqsubset [\Delta] \circ [A] \circ [B] \circ [\Delta']$. We also know that $[\Delta, A, B, \Delta'] = [\Delta] \circ [A] \circ [B] \circ [\Delta'] \sqsubset [C]$. We therefore have

$$[\Delta, \Gamma, \Delta'] = [\Delta] \circ [\Gamma] \circ [\Delta'] \sqsubset [\Delta] \circ [A] \circ [B] \circ [\Delta'] \sqsubset [C]$$

- If the last rule is the product introduction rule $\bullet_i$ by induction hypothesis we know that $[\Delta] \sqsubset [A]$ and that $[\Delta'] \sqsubset [B]$; consequently

$$[\Delta, \Delta'] = [\Delta] \circ [\Delta'] \sqsubset [A] \circ [B] = [A \bullet B] \qquad \square$$

Proposition 2.21. *A sequent which is valid in every residuated semi-group is derivable.*

Proof. Let F be the set of formulae and let $M = F/\vdash\!\dashv$ be the quotient of formulae by the equivalence relation $\vdash\!\dashv$; this relation $\vdash\!\dashv$ is defined by $A \vdash\!\dashv B$ whenever $A \vdash B$ and $B \vdash A$; it is symmetrical, it is reflexive by the axiom rule and the cut rule ensures it is transitive.

Is is easily observed that $\backslash$, $/$, $\bullet$ and $\vdash$ can be defined over equivalence classes, that is: whenever $A \vdash\!\dashv A'$ and $B \vdash\!\dashv B'$ one has $(A \diamond B) \vdash\!\dashv (A' \diamond B')$, for $\diamond \in \{\backslash, /, \bullet\}$. So let us define $\circ, \backslash\backslash, /\!/$ as the corresponding operations over equivalence classes of $\vdash\!\dashv$: $A^{\vdash\!\dashv} \circ B^{\vdash\!\dashv} = (A \bullet B)^{\vdash\!\dashv}$, $A^{\vdash\!\dashv} \backslash\backslash B^{\vdash\!\dashv} = (A \backslash B)^{\vdash\!\dashv}$ and $B^{\vdash\!\dashv} /\!/ A^{\vdash\!\dashv} = (B/A)^{\vdash\!\dashv}$ Finally let $\sqsubset$ be $\vdash$ which can also be defined for equivalence classes: if $A \vdash\!\dashv A'$ and $B \vdash\!\dashv B'$ then $A \vdash B$ is equivalent to $A' \vdash B'$.

The property (RSG) is satisfied i.e. $(A^{\vdash\!\dashv} \circ B^{\vdash\!\dashv}) \sqsubset C^{\vdash\!\dashv}$ is equivalent to $A^{\vdash\!\dashv} \sqsubset (C^{\vdash\!\dashv} /\!/ B^{\vdash\!\dashv})$ and to $A^{\vdash\!\dashv} \sqsubset (B^{\vdash\!\dashv} \backslash\backslash C^{\vdash\!\dashv})$. Indeed $A \vdash A$ and $B \vdash B$ lead to $A, B \vdash A \bullet B$; thus from $A \bullet B \vdash C$ one obtains $A, B \vdash C$ which yields $A \vdash C/B$ by $/_i$ and $B \vdash A \backslash C$ by $\backslash_i$; from $B \vdash A \backslash C$ (resp. $A \vdash B/C$) using $A \vdash A$ (resp. $B \vdash B$) one obtains $A, B \vdash C$ by $\backslash_e$ (resp. by $/_e$) and $A \bullet B \backslash C$ by $\bullet_h$.

Now let us consider the interpretation $[p] = p^{\vdash\!\dashv}$ for every primitive type. Then for every formula $[A] = A^{\vdash\!\dashv}$.

To say that a sequent $H_1, \ldots, H_n \vdash A$ is valid in this model under this interpretation is to say that $[H_1, \ldots, H_n] \sqsubset [A]$. Therefore $H_1 \bullet \cdots \bullet H_n \vdash A$ is provable which entails that $H_1, \ldots, H_n \vdash A$ is provable as well — indeed from $H_1 \bullet \cdots \bullet H_n \vdash A$ one obtains $\vdash H_1 \bullet \cdots \bullet H_n \backslash A$ (*); then by n rules $\bullet_i$ on the axioms $H_i \vdash H_i$ one obtains $H_1, \ldots, H_n \vdash H_1 \bullet \cdots \bullet H_n$ (**) and an application of $\bullet_e$ to (*) and (**) yields $H_1, \ldots, H_n \vdash A$. □

2.9.4 L Is Sound and Complete w.r.t. (Free) Semi-group Models

A more interesting class of models is provided by semi-groups. Indeed, the interpretation of a category should be the set of the words and expressions of this category, shouldn't it?

So, given a semi-group $(W, .)$ that is a set W endowed with an associative composition "." one can define a residuated semi-group as follows:

- $M = 2^W$
- $A \circ B = \{ab \mid a \in A \text{ and } b \in B\}$
- $A \backslash\backslash B = \{z \mid \forall a \in A \quad az \in B\}$
- $B /\!/ A = \{z \mid \forall a \in A \quad za \in B\}$
- $A \sqsubset B$ whenever $A \subset B$ (as sets).

It is easily seen that this structure really is a residuated semi-group:

- $\circ$ is associative:

$$(A \circ B) \circ C = \{abc \mid a \in A \text{ and } b \in B \text{ and } c \in C\} = A \circ (B \circ C)$$

- $\subset$ is an order on 2^W
  (RSG) The following statements are clearly equivalent:

$$\begin{array}{l}(A \circ B) \subset C \ : \forall a \in A \ \forall b \in B \quad ab \in C \\ A \subset (C /\!\!/ B) : \forall a \in A \quad a \in (C /\!\!/ B) \\ B \subset (A \backslash\!\!\backslash C) : \forall b \in B \quad b \in (A \backslash\!\!\backslash C)\end{array}$$

The free semi-group models are particularly interesting, since there are no equations between sequences of words. The following proposition may be understood as stating that L is the logic of free semi-groups:

Proposition 2.22. *Product free L is complete over free semi-group models.*

Proof. Take as semi-group the finite non empty sequences of formulae F^+, endowed with concatenation $(A_1,\ldots,A_n)\cdot(B_1,\ldots,B_p) = A_1,\ldots,A_n,B_1,\ldots,B_p$.

For a primitive type p define $[p]$ by $\{\Gamma \mid \Gamma \vdash p\}$.

Let us first verify that for every formula F, the set of finite sequences $[F]$ defined inductively from the $[p]$'s by the definition of $\backslash\!\!\backslash$ and $/\!\!/$ is precisely $Ctx(F) = \{\Delta \mid \Delta \vdash F\}$. We proceed by induction on F. Is F if some primitive type, it is true by definition. Now assume that $[G] = Ctx(G)$ and $[H] = Ctx(H)$ and let us verify that $[G \setminus H] = Ctx(G \setminus H)$ — the case for H / G being symmetrical.

$Ctx(G \setminus H) \subset [G \setminus H]$ Let Δ be a sequence such that $\Delta \in Ctx(G \setminus H)$ that is $\Delta \vdash G \setminus H$ (1) and let us see that for every $\Theta \in [G]$ we have $\Theta, \Delta \in [H]$ — which entails $\Delta \in [G \setminus H]$. By induction hypothesis we have $Ctx(G) = [G]$ so $\Theta \vdash G$ (2). From (1) and (2) we obtain $\Theta, \Delta \vdash H$, so $\Theta, \Delta \in Ctx(H)$. Since by induction hypothesis $Ctx(H) = [H]$ we have $\Theta, \Delta \in [H]$. As this holds for every Θ we have $\Delta \in [G \setminus H]$.

$[G \setminus H] \subset Ctx(G \setminus H)$ Let Δ be a sequence such that $\Delta \in [G \setminus H]$. Let us show that $\Delta \vdash G \setminus H$. Since $G \vdash G$ we have $G \in Ctx(G)$ and by induction hypothesis $G \in [G]$. By the definition of $[G \setminus H]$ we thus have $G, \Delta \in [H]$ and, since by induction hypothesis we have $[H] = Ctx(H)$ we obtain $G, \Delta \vdash H$. Now, by the $\setminus_i$ introduction rule we obtain $\Delta \vdash G \setminus H$, that is $\Delta \in Ctx(G \setminus H)$.

If a sequent $A_1,\ldots,A_n \vdash C$ is valid in this model under this interpretation, what does it mean? We have $[A_1] \circ \cdots \circ [A_n] \subset [C]$ and as $A_i \in [A_i]$ we have $A_1,\ldots,A_n \in [C]$ that is $A_1,\ldots,A_n \vdash C$. □

Next follows a very difficult result due to Pentus (Pentus, 1993a), that we state without giving the proof.

Proposition 2.23. *L with product is also complete w.r.t. free semi-groups models.*

2.10 Interpolation

This section presents the interpolation theorem for the Lambek calculus, which appeared in the thesis of Roorda (Roorda, 1991).

Interpolation is somehow the converse of cut elimination. The interest of cut free proofs is that they obey the subformula property. The usual interest of interpolation, say for classical or intuitionistic logic is to be able to factor equal sub-proofs in a given proof. In the Lambek calculus where contraction is prohibited, nothing like this can happen. So the interest is very different, let us explain it in a few words.

Assume we are able to formulate the calculus with a set of axioms, and only the cut rule: viewing $\vdash$ as $\longrightarrow$ (in the opposite direction) the calculus is nothing but a set of context-free production rules — the cut rule is the substitution rule often left implicit in phrase structure grammars.

Indeed a production rule $X \longrightarrow X_1 \cdots X_n$ corresponds to an axiom $X_1, \ldots, X_n \vdash X$ and the cut rule simply states that if we have been able to derive

$$W \longrightarrow V_1 \cdots V_k\, T\, U_1 \cdots U_l$$
$$T \longrightarrow Z_1 \cdots Z_j$$

then we are able to derive

$$W \longrightarrow V_1 \cdots V_k\, Z_1 \cdots Z_j\, U_1 \cdots U_l.$$

Now observe that for a given Lambek grammar because of cut elimination we know that the types appearing in any syntactic analysis are all subformulae of the conclusion sequent: indeed a syntactic analysis is a proof of $t_1, \ldots, t_n \vdash S$ with all t_i in the lexicon. Can we derive any syntactic analysis from a *finite* number of provable sequents by means of the cut rule only? As we shall see in the next section, this is possible and consequently Lambek grammars are weakly equivalent to context-free grammars.

Given a formula or a sequence of formulae Δ and a primitive type p we denote by $\rho_p(\Delta)$ *the number of occurrences of* p *in* Δ.

Proposition 2.24. *Let $\Gamma, \Delta, \Theta \vdash C$ be a provable sequent in L, with $\Delta \neq \varepsilon$. There exists an interpolant of Δ that is a formula I such that:*

1. $\Delta \vdash I$
2. $\Gamma, I, \Theta \vdash C$
3. $\rho_p(I) \leq \rho_p(\Delta)$ *for every primitive type* p
4. $\rho_p(I) \leq \rho_p(\Gamma, \Theta, C)$ *for every primitive type* p

Proof. We proceed by induction on the size of a cut free proof of $\Gamma, \Delta, \Theta \vdash C$ — there are many cases in this proof, according to the nature of the last rule, and to the respective position of the created formula and Δ.

$$\boxed{\text{axiom } X \vdash X}$$

If the proof is an axiom, then Δ is a formula X and $I = X$ obviously works:

1. $X \vdash X$
2. $X \vdash X$
3. $\rho_p(X) = \rho_p(X)$
4. $\rho_p(X) = \rho_p(\varepsilon, \varepsilon, X)$

$$\frac{\Pi \vdash X \quad \Phi \vdash Y}{\Pi, \Phi \vdash X \bullet Y} \bullet_i$$

- $\Pi = \Pi', \Delta, \Pi''$ — so $\Gamma = \Pi'$ and $\Theta = \Pi'', \Phi$.
  By induction hypothesis we have an interpolant I for Δ in $\Pi', \Delta, \Pi'' \vdash X$, let us see it is an interpolant for Δ in $\Pi', \Delta, \Pi'', \Phi \vdash X \bullet Y$.
  1. We already have $\Delta \vdash I$
  2. From $\Pi', I, \Pi'' \vdash X$ and $\Phi \vdash Y$, we have $\Pi', I, \Pi'', \Phi \vdash X \bullet Y$.
  3. We already have $\rho_p(I) \leq \rho_p(\Delta)$.
  4. From $\rho_p(I) \leq \rho_p(\Pi', \Pi'', X)$ we obtain $\rho_p(I) \leq \rho_p(\Pi', \Pi'', \Phi, X, Y)$.
- $\Phi = \Phi', \Delta, \Phi''$ — so $\Gamma = \Pi, \Phi'$ and $\Theta = \Phi''$.
  Symmetrical to the previous case.
- $\Pi = \Pi', \Delta'$ $\Phi = \Delta'', \Phi''$ and $\Delta = \Delta', \Delta''$ — so $\Gamma = \Pi'$ and $\Theta = \Phi''$.
  By induction hypothesis we have an interpolant I' for Δ' in $\Pi', \Delta' \vdash X$ and an interpolant I'' for Δ'' in $\Delta'', \Phi'' \vdash X$. Then $I = I' \bullet I''$ is an interpolant for $\Delta = \Delta', \Delta''$ in $\Pi', \Delta', \Delta'', \Phi'' \vdash X \bullet Y$.
  1. From $\Delta' \vdash I'$ and $\Delta'' \vdash I''$ we obtain $\Delta', \Delta'' \vdash X \bullet Y$ by $\bullet_i$.
  2. From $\Pi', I' \vdash X$ and $I'', \Phi'' \vdash Y$ we have $\Pi', I', I'', \Phi'' \vdash X \bullet Y$ by $\bullet_i$ and finally $\Pi', I' \bullet I'', \Phi'' \vdash X \bullet Y$ by $\bullet_h$
  3. From $\rho_p(I') \leq \rho_p(\Pi', X)$ and $\rho_p(I'') \leq \rho_p(\Phi'', Y)$ we get $\rho_p(I' \bullet I'') = \rho_p(I') + \rho_p(I'') \leq \rho_p(\Pi', X) + \rho_p(\Phi'', Y) = \rho_p(\Pi', \Phi'', X, Y) = \rho_p(\Pi', \Phi'', X \bullet Y)$.
  4. From $\rho_p(I') \leq \rho_p(\Delta')$ and $\rho_p(I'') \leq \rho_p(\Delta'')$ we get $\rho_p(I' \bullet I'') = \rho_p(I') + \rho_p(I'') \leq \rho_p(\Delta', \Delta'') = \rho_p(\Delta)$.

$$\frac{\Pi, X, Y, \Phi \vdash C}{\Pi, X \bullet Y, \Phi \vdash C} \bullet_h$$

Let Δ' be defined as follows: if Δ contains $X \bullet Y$ then $\Delta' = \Delta[X \bullet Y := X, Y]$, otherwise $\Delta' = \Delta$. Let I be an interpolant for Δ' in $\Pi, X, Y, \Phi \vdash C$. Then I is itself an interpolant for Δ in $\Pi, X \bullet Y, \Phi \vdash C$.

1. From $\Delta' \vdash I$ we have $\Delta \vdash I$ (possibly using $\bullet_h$).
2. From $\Pi, X \bullet Y, \Phi[\Delta' := I] \vdash C$ we get $\Pi, X \bullet Y, \Phi[\Delta := I] \vdash C$.
3. From $\rho_p(\Delta) = \rho_p(\Delta')$ we obtain $\rho_p(I) \leq \rho_p(\Delta)$
4. Since $\rho_p((\Pi, X \bullet Y, \Phi)[\Delta' := \varepsilon], C) = \rho_p((\Pi, X \bullet Y, \Phi)[\Delta := \varepsilon], C)$ we have $\rho_p(I) \leq \rho_p((\Pi, X \bullet Y, \Phi)[\Delta := \varepsilon], C)$.

$$\frac{X, \Gamma, \Delta, \Theta \vdash Y}{\Gamma, \Delta, \Theta \vdash X \setminus Y} \setminus_i$$

By induction hypothesis we have an interpolant I for Δ in $A, \Gamma, \Delta, \Theta \vdash B$. It is an interpolant for Δ in $\Gamma, \Delta, \Theta \vdash X \setminus Y$ as well.

1. We already have $\Delta \vdash I$.
2. From $X, \Gamma, I, \Theta \vdash Y$ we obtain $\Gamma, I, \Theta \vdash X \setminus Y$ by $\setminus_i$.

3. We already have $\rho_p(I) \leq \rho_p(\Delta)$.
4. We have: $\rho_p(I) \leq \rho_p(X, \Gamma, \Theta, Y) = \rho_p(\Gamma, \Theta, X \setminus Y)$.

$$\boxed{\dfrac{\Pi \vdash X \quad \Phi, Y, \Psi \vdash C}{\Phi, \Pi, X \setminus Y, \Psi \vdash C} \setminus_h}$$

- Δ is included into Π Let I be an interpolant for Δ in the premise containing it. Then I is an interpolant for Δ in $\Phi, \Pi, X \setminus Y, \Psi \vdash C$.
  1. We already have $\Delta \vdash I$
  2. From $\Pi[\Delta := I] \vdash X$ and $\Phi, Y, \Psi \vdash C$, by $\setminus_h$ we obtain $\Phi, \Pi[\Delta := I], X \setminus Y, \Psi \vdash C$
  3. We already have $\rho_p(I) \leq \rho_p(\Delta)$.
  4. From $\rho_p(I) \leq \rho_p(\Pi[\Delta := \varepsilon], X)$ we have $\rho_p(I) \leq \rho_p(\Phi, \Pi[\Delta := \varepsilon], X \setminus Y, \Psi, C)$
- Δ is included in Φ (resp. Ψ) Let I be an interpolant for Δ in the premise containing it. Then I is an interpolant for Δ in $\Phi, \Pi, X \setminus Y, \Psi \vdash C$.
  1. We already have $\Delta \vdash I$
  2. From $\Phi[\Delta := I], Y, \Psi \vdash C$ (resp. $\Phi, Y, \Psi[\Delta := I] \vdash C$) and $\Pi \vdash X$, by $\setminus_h$ we obtain $\Phi[\Delta := I], \Pi, X \setminus Y, \Psi \vdash C$ (resp. $\Phi, \Pi, X \setminus Y, \Psi[\Delta := I] \vdash C$)
  3. We already have $\rho_p(I) \leq \rho_p(\Delta)$.
  4. From $\rho_p(I) \leq \rho_p(\Phi[\Delta := \varepsilon], Y, \Psi, C)$ (resp. $\rho_p(I) \leq \rho_p(\Phi, Y, \Psi[\Delta := \varepsilon], C)$) we have $\rho_p(I) \leq \rho_p(\Phi[\Delta := \varepsilon], \Pi, X \setminus Y, \Psi, C)$ (resp. $\rho_p(I) \leq \rho_p(\Phi, \Pi, X \setminus Y, \Psi[\Delta := \varepsilon], C)$).
- $\Delta = \Delta', \Delta''$ and $\Phi = \Phi', \Delta'$ and $\Pi = \Delta'', \Pi''$.
  Let I' be an interpolant for Δ' in $\Phi', \Delta', Y, \Psi \vdash C$, and let I'' be an interpolant for Δ'' in $\Delta'', \Pi'' \vdash X$. Then $I = I' \bullet I''$ is an interpolant for Δ', Δ'' in $\Phi', \Delta', \Delta'', \Pi'', X \setminus Y, \Psi \vdash C$.
  1. From $\Delta' \vdash I'$ and $\Delta'' \vdash I''$ we have $\Delta', \Delta'' \vdash I' \bullet I''$ by $\bullet_i$.
  2. From $I'', \Pi'' \vdash X$ and $\Phi', I', Y, \Psi \vdash C$ we have $\Phi', I', I'', X \setminus Y, \Psi \vdash C$ by $\setminus_h$ and $\Phi', I' \bullet I'', X \setminus Y, \Psi \vdash C$ by $\bullet_i$.
  3. We have $\rho_p(I' \bullet I'') = \rho_p(I') + \rho_p(I'') \leq \rho_p(\Delta') + \rho_p(\Delta'') = \rho_p(\Delta)$.
  4. We have $\rho_p(I' \bullet I'') = \rho_p(I') + \rho_p(I'') \leq \rho_p(\Phi', Y, \Psi, C) + \rho_p(\Pi'', X) = \rho_p(\Phi', \Pi'', X \setminus Y, \Psi, C)$.
- $\Delta = \Phi'', \Pi, X \setminus Y, \Psi'$ with $\Phi = \Phi', \Phi''$ and $\Psi = \Psi', \Psi''$.
  Let I be an interpolant for Φ'', Y, Ψ' in $\Phi', \Phi'', Y, \Psi', \Psi'' \vdash C$. Then I is itself interpolant for $\Phi'', \Pi, X \setminus Y, \Psi'$ in $\Phi', \Phi'', \Pi, X \setminus Y, \Psi', \Phi'' \vdash C$.
  1. From $\Phi'', Y, \Psi' \vdash I$ and $\Pi \vdash X$ we have $\Phi'', \Pi, X \setminus Y, \Psi' \vdash I$ by $\setminus_h$.
  2. We already have $\Phi', I, \Psi'' \vdash C$.
  3. We already have $\rho_p(I) \leq \rho_p(\Phi', \Psi'', C)$
  4. We have $\rho_p(I) \leq \rho_p(\Phi'', Y, \Psi') \leq \rho_p(\Phi'', \Pi, X \setminus Y, \Psi')$.
- $\Delta = \Pi'', X \setminus Y, \Psi'$ with $\Pi = \Pi', \Pi''$ and $\Psi = \Psi', \Psi''$.
  Let I' be and interpolant for Π' in $\Pi', \Pi'' \vdash X$ and let I'' be an interpolant for Y, Ψ' in $\Phi, Y, \Psi', \Psi'' \vdash C$. Then $I' \setminus I''$ is an interpolant for $\Delta = \Pi'', X \setminus Y, \Psi'$ in $\Phi, \Pi', \Pi'', X \setminus Y, \Psi', \Psi'' \vdash C$.

1. From $I', \Pi'' \vdash X$ and $Y, \Psi' \vdash I''$ we have $I', \Pi'', X \setminus Y, \Psi' \vdash I''$ by $\setminus_h$ and $\Pi'', X \setminus Y, \Psi' \vdash I' \setminus I''$ by $\setminus_i$.
2. From $\Phi, I'', \Psi'' \vdash C$ and $\Pi' \vdash I'$ we have $\Phi, \Pi', I' \setminus I'', \Psi'' \vdash C$.
3. We have $\rho_p(I' \setminus I'') \leq \rho_p(\Pi'', X) + \rho_p(Y, \Psi') = \rho_p(\Pi'', X \setminus Y, \Psi')$
4. We have $\rho_p(I' \setminus I'') \leq \rho_p(\Pi') + \rho_p(\Phi, \Psi'', C) = \rho_p(\Phi, \Pi', \Psi'', C)$

This ends the proof because $/_i$ and $/_e$ are symmetrical to $\setminus_i$ and $\setminus_e$. □

2.11 Lambek Grammars and Context-Free Grammars

At the beginning of this section we shall see that context-free grammars translate into weakly equivalent Lambek grammars (Cohen, 1967): this is non trivial but unsurprising, and this section is in fact devoted to prove the converse, known as the Chomsky conjecture, stated in 1963 (Chomsky, 1963, p. 413) and proved by Pentus (Pentus, 1993b, 1997): *Languages generated by Lambek grammars are context-free languages*. This result was already suggested in the previous section on interpolation: if we are able to derive all sequents corresponding to syntactic analyses from a *finite* set of sequents by the cut rule only, then Lambek grammars are context-free.

Let us define the *size* $|A|$ of a formula A by its number of primitive types. We are going to show that given an integer m there exists a *finite* set $AX(m)$ of provable sequents such that all provable sequent containing only formulae of size smaller than m are derivable from sequents in $AX(m)$ by means of the cut rule only. This easily entails that Lambek grammars are context-free. Note that the restriction on the size of formulas is essential, since Zielonka (1981, 1989) shows that the Lambek calculus in general (without this size restriction) does *not* permit a formulation consisting of a finite number of sequents AX and the cut rule.

Even though Lambek grammars generate only context-free languages, they have a number of pleasant properties — in addition to their logical foundations — which make them interesting objects of study:

- they are lexicalized,
- they offer a pleasant interface with semantics (as we will see in Chapter 3),
- they permit an elegant treatment of peripheral extraction (as we have seen in Example 2.2, for example). To the best of our knowledge, among the grammar formalisms which generate context-free languages, only the Lambek calculus (and some closely related formalisms) have such a simple and elegant treatment of peripheral extraction.
- finally, let us say that while the derivation trees of a context-free grammars constitute a regular tree language (Thatcher, 1967; Gécseg and Steinby, 1997) the derivation trees (natural deduction trees) of a Lambek grammar can constitute a tree language which is not regular (Tiede, 2001): Kanazawa and Salvati (2009) show that the natural deduction trees of a Lambek grammar correspond to tree languages generated by hyperedge replacement grammars (Engelfriet, 1997; Engelfriet and Maneth, 2000). In other words, if we are interested in *trees* rather than strings, Lambek grammars are more expressive than context-free grammars.

There are basically two ingredients for the Pentus proof that Lambek grammars are context-free. One is interpolation and we already explained its relevance to this question. The other is a property of the free group to be applied to the free group model of Section 2.9.2 on page 45. This property is needed to find, in a sequent where all formulae have sizes lower than m, two (or more) consecutive formulae whose interpolant also has a size less than m — this is of course to be used for the final induction.

We mainly follow (Pentus, 1993b), and borrow a few things from (Pentus, 1997; Buszkowski, 1997).

2.11.1 From Context-Free Grammars to Lambek Grammars

It is natural to think that every AB grammar corresponds to a Lambek grammar, because the Lambek calculus includes the AB elimination rules and is therefore at least as expressive as AB grammars. In fact this result — although not as difficult as the result we will prove in the remainder of this section, where we will show how to translate Lambek grammars to context-free grammars — is not completely trivial: not all theorems of L are also theorems of AB, so we need to be careful about using an AB grammar as a Lambek grammar.

By Proposition 1.2 from Chapter 1, we know that any AB grammar is weakly equivalent to an AB grammar containing only types of order at most 1. Now, by Proposition 2.12 a sequent $A_1,\ldots,A_n \vdash S$ with $o(A_i) \leq 1$ is provable with AB elimination rules if and only if it is provable in L. Consequently the language generated by an AB grammar with types of order at most 1 coincides with the language generated by the Lambek grammar with the same lexicon.

Using the weak equivalence between AB grammars and context-free grammars (Propositions 1.11 and 1.10) we have the result of (Cohen, 1967):

Proposition 2.25. *Every ε-free context-free grammar is weakly equivalent to a Lambek grammar.*

2.11.2 A Property of the Free Group

Let w be an element of the free group; then $\|w\|$ stands for the length of the reduced word corresponding to w — e.g. $\|cb^{-1}a^{-1}abc\| = 2$.

This lemma, which is needed for a refinement of interpolation, only concerns the free group. It had actually been proved before in (Nivat, 1971) and was reproved in (Autebert et al, 1984).

Proposition 2.26. *The two following properties of the free group hold:*

1. *Let u,v,w be elements of the free group; if $\|u\| < \|uv\|$ and $\|uv\| \geq \|uvw\|$ then $\|vw\| \leq \max(\|v\|,\|w\|)$.*

2. *Let u_i $i = 1,\ldots,n+1$ be elements of the free group with $u_1 \cdot \cdots \cdot u_{n+1} = 1$. Then there exists $k \leq n$ such that*

$$\|u_k u_{k+1}\| \leq \max(\|u_k\|, \|u_{k+1}\|)$$

Proof. The first part is actually a lemma for the second part.

Proof of 1 We proceed by reductio ad absurdum, so we assume that

a. $\|u\| < \|uv\|$
b. $\|uv\| \geq \|uvw\|$
c. $\|vw\| > \|v\|$
d. $\|vw\| > \|w\|$

There exists three reduced words x, y, z such that

- $u = xy^{-1}$ $v = yz$ $uv = xz$
- xy^{-1} yz xz are reduced.

From (a) we have $\|x\| + \|y\| < \|x\| + \|z\|$ so $\|y\| < \|z\|$ and therefore $\|y\| < \frac{1}{2}\|v\|$ (*).

Similarly there exists three reduced words x', y', z' such that

- $v = x'y'$ $w = y'^{-1}z'$ $vw = x'z'$
- $x'y'$ $y'^{-1}z'$ $x'z'$ are reduced.

From (c) we have $\|y'\| + \|z'\| < \|x'\| + \|z'\|$ so $\|y'\| < \|x'\|$ and therefore $\|y'\| < \frac{1}{2}\|v\|$ (**)

From $v = yz = x'y'$ with $\|y\| < \frac{1}{2}\|v\|$ (*) and $\|y'\| < \frac{1}{2}\|v\|$ (**), there exists a non empty a such that

- $z = ay'$ $x' = ya$ $v = yay'$
- ay' ya yay' are reduced

So we have $uvw = xy^{-1}yay'y'^{-1}z' = xaz'$ — as xa and az' are reduced, xaz' is reduced as well. From (b) we have

$$\|uvw\| = \|xaz'\| \leq \|xay'\| = \|xz\| = \|uv\|$$

and therefore $\|z'\| \leq \|y'\|$.

Since from d we have $\|x'z'\| > \|x'y'\|$ so $\|z'\| > \|y'\|$, there is a contradiction.

Proof of 2 Let k be the first index such that $\|u_1 \cdots u_k\| \geq \|u_1 \cdots u_k u_{k+1}\|$.

If $k = 1$ $\|u_1\| \geq \|u_1 u_2\|$ then $\max(\|u_1\|, \|u_2\|) \geq \|u_1\| \geq \|u_1 u_2\|$.

Otherwise, let

$$u = u_1 \cdots u_{k-1} \qquad v = u_k \qquad w = u_{k+1}$$

we have

$$\|u\| = \|u_1 \cdots u_{k-1}\| < \|uv\| = \|u_1 \cdots u_{k-1}u_k\|$$

and

$$\|uv\| = \|u_1 \cdots u_{k-1}u_k\| \geq \|uvw\| = \|u_1 \cdots u_k u_{k+1}\|$$

so applying the first part (1) of this proposition we obtain

$$\|u_k u_{k+1}\| \leq \max(\|u_k\|, \|u_{k+1}\|)$$ □

2.11.3 Interpolation for Thin Sequents

A sequent $\Gamma \vdash C$ is said to be *thin* whenever it is provable and for all p, $\rho_p(\Gamma, C)$ is at most 2 — where $\rho_p(\Theta)$ is the number of occurrences of a primitive type p in Θ. Notice that by Proposition 2.6 which says that a provable sequent contains as many positive and negative occurrences of a primitive type, $\rho_p(\Gamma, A)$ is either 0 or 2.

Here is a proposition which is very representative of multiplicative calculi, in which formulas can be neither contracted or weakened:

Proposition 2.27. *Each provable sequent may be obtained from a thin sequent by substituting primitive types with primitive types.*

Proof. Given a cut free proof d with only primitive axioms of a sequent $\Gamma \vdash C$, number the axioms and replace each axiom $p \vdash p$ by $p_i \vdash p_i$ where i is the number of the axiom, and also replace all the traces of this occurrence of p in the proof with p_i. Clearly the result is itself a proof of a sequent $\Gamma' \vdash C'$, which contains exactly two or zero occurrences of each primitive type, and which gives back $\Gamma \vdash C$ when each p_i is substituted with p. □

Proposition 2.28. *Let $\Gamma, \Delta, \Theta \vdash C$ be a thin sequent. Then there exists a formula B such that:*

1. *$\Delta \vdash B$ is thin*
2. *$\Gamma, B, \Theta \vdash C$ is thin*
3. *$|B| = \|[\Delta]\|$ — the number of primitive types in B is the size of the interpretation of Δ in the free group (see Section 2.9.2 on page 45).*

Proof. p stands for any primitive type,

Let B be an interpolant of Δ which exists by Theorem 2.24. We then have:

a. $\Delta \vdash B$ is provable
b. $\Gamma, B, \Theta \vdash C$ is provable
c. $\rho_p(B) \leq \min(\rho_p(\Gamma, \Theta, C), \rho_p(\Delta))$

Let Us First Prove 1. As the sequent $\Gamma, \Delta, \Theta \vdash C$ is thin,

$$\rho_p(\Gamma, \Delta, \Theta, C) = \rho_p(\Gamma, \Theta, C) + \rho_p(\Delta)$$

is either 0 or 2; so by c $\rho_p(B)$ is either 0 or 1, and we have

$$\rho_p(\Delta, B) = \rho_p(\Delta) + \rho_p(B) \leq \rho_p(\Gamma, \Delta, \Theta, C) + \rho_p(B) \leq 2 + 1$$

Since $\Delta \vdash B$ is provable (a), $\rho_p(\Delta, B)$ is even, and thus $\rho_p(\Delta, B) \leq 2$. So, being provable, $\Delta \vdash B$ is thin.

Now Let Us Prove 2. Similarly,

$$\rho_p(\Gamma, B, \Theta, C) = \rho_p(\Gamma, \Theta, C) + \rho_p(B) \leq \rho_p(\Gamma, \Delta, \Theta, C) + \rho_p(B) \leq 2 + 1$$

Since $\Gamma, B, \Theta \vdash C$ is provable (b) $\rho_p(\Gamma, B, \Theta, C)$ is even, so $\rho_p(\Gamma, B, \Theta, C) \leq 2$ So, being provable, $\Gamma, B, \Theta \vdash C$ is thin.

Finally Let Us Prove 3

- if p does not occur in Δ then p does neither occur in $[\Delta]$ nor in B, by c.
- if p occurs once in Δ then it occurs once in $[\Delta]$ too — it cannot cancel with another occurrence of p; as $\Delta \vdash B$ is thin it also occurs once in B — it occurs twice in $\Delta \vdash B$ and once in Δ so it occurs once in B.
- if p occurs twice in Δ then it does not occur in Γ, Θ, C; therefore it does not occur in B by (c). The soundness of the interpretation in the free group entails $[\Gamma, \Delta, \Theta] = [C]$ that is $[\Delta] = [\Gamma]^{-1}[C][\Theta]^{-1}$ As p does not occur in Γ, Θ, C, there is no occurrence of p in $[\Gamma]^{-1}[C][\Theta]^{-1}$ and therefore no occurrence of p in $[\Delta]$

So for every primitive type, and whatever its number of occurrences in Δ is, there are exactly as many occurrences of p in B and in $[\Delta]$, so the number of primitive types in B and in $[\Delta]$ are equal: $|B| = \|[\Delta]\|$. □

Proposition 2.29. *Let $A_1, \ldots, A_n \vdash A_{n+1}$ be a thin sequent with $|A_i| \leq m$; then either:*

- *there exists an index k and a type B with $|B| \leq m$ such that the following sequents are thin:*
  $A_1, \ldots, A_{k-1}, B, A_{k+2}, \ldots, A_n \vdash A_{n+1}$
  $A_k, A_{k+1} \vdash B$
- *there exists a type B with $|B| \leq m$ such that the following sequents are thin:*
  $B, A_n \vdash A_{n+1}$
  $A_1, \ldots, A_{n-1} \vdash B$

Proof. Let $u_i = [A_i]$ for $1 \leq i \leq n$ and $u_{n+1} = [C]^{-1}$. Interpreting the provability in the free group we obtain: $u_1 \cdots u_n u_{n+1} = 1$ By Lemma 2.26 there exists an index $k \leq n$ for which $\|u_k u_{k+1}\| \leq \max(\|u_k\|, \|u_{k+1}\|) \leq m$.

- If $k < n$, we apply Proposition 2.28 with
  $\Delta = A_k, A_{k+1}$,
  $\Gamma = A_1, \ldots, A_{k-1}$
  $\Theta = A_{k+2}, \ldots, A_n$.
  So the sequents
  $A_1, \ldots, A_{k-1}, B, A_{k+2}, \ldots, A_n \vdash A_{n+1}$
  $A_k, A_{k+1} \vdash B$
  are thin, and

$$|B| = \|[A_k, A_{k+1}]\| = \|u_k u_{k+1}\| \leq m$$

- If $k = n$, we apply Proposition 2.28 with
  $\Gamma = \varepsilon$,
  $\Delta = A_1, \ldots, A_{n-1}$
  $\Theta = A_n$.
  So the sequents $A_1, \ldots, A_n \vdash B$ and $B, A_n \vdash B$ are thin.
  Since $[A_1, \ldots, A_{n-1}, A_n] = [C]$
  we have $|B| = \|[A_1, \ldots, A_{n-1}]\| = \|[C][A_n]^{-1}\|$

therefore

$$|B| = \|[C][A_n]^{-1}\| = \|([A_n][C]^{-1})^{-1}\| = \|(u_n u_{n+1})^{-1}\| = \|u_n u_{n+1}\| \leq m \qquad \square$$

2.11.4 From Lambek Grammars to Context-Free Grammars

Proposition 2.30. *If a sequent $A_1,\ldots,A_n \vdash A_{n+1}$ with each $|A_i| \leq m$ is provable in L, then it is provable from provable sequents $U,V \vdash X$ or $U \vdash X$ with $|U|,|V|,|X| \leq m$ by means of the cut rule only.*

Proof. We proceed by induction on n. If $n \leq 2$ then there is nothing to prove. Otherwise, let $A'_1,\ldots,A'_n \vdash A'_{n+1}$ be a corresponding thin sequent obtained as in Proposition 2.27 — using a different primitive type for each axiom in the proof of $A_1,\ldots,A_n \vdash A_{n+1}$. Thus there exists a substitution σ replacing primitive types with primitive types and preserving provability such that $\sigma(A') = A$.

As the substitution replaces primitive types with primitive types, we also have $|A'_i| \leq m$. By Proposition 2.29 there exists a formula B' with $|B'| \leq m$ such that either:

- $A'_1,\ldots,A'_{k-1},B',A'_{k+2},\ldots,A'_n \vdash A'_{n+1}$
  $A'_k,A'_{k+2} \vdash B'$

  are thin, and therefore provable.

  Let $B = \sigma(B')$, so B has at most m primitive types as well; applying the substitution we obtain two provable sequents

  $A_1,\ldots,A_{k-1},B,A_{k+2},\ldots,A_n \vdash A_{n+1}$
  $A_k,A_{k+1} \vdash B$.

  By induction hypothesis

  $$A_1,\ldots,A_{k-1},B,A_{k+2},\ldots,A_n \vdash A_{n+1} \qquad (*)$$

  is provable from provable sequents $U,V \vdash X$ or $U \vdash X$ with $|U|,|V|,|X| \leq m$ by means of the cut rule only.

  Notice that $A_k,A_{k+1} \vdash B \quad (**)$ is of the form $U,V \vdash X$ with $|U|,|V|,|X| \leq m$. A cut rule between the proof of $(*)$ and $(**)$ yields a proof of

  $$A_1,\ldots,A_n \vdash A_{n+1}$$

  from provable sequents $U,V \vdash X$ or $U \vdash X$ with $|U|,|V|,|X| \leq m$ by means of the cut rule only.
- $B',A'_n \vdash A'_{n+1}$ and $A_1,\ldots,A_{n-1} \vdash B$ are thin and therefore provable.
  Let $B = \sigma(B')$, so $|B| \leq m$; applying the substitution we obtain two provable sequents
  $B,A_n \vdash A_{n+1}$
  $A_1,\ldots,A_{n-1} \vdash B$.

By induction hypothesis

$$A_1,\ldots,A_{n-1},B \vdash A_{n+1} \qquad (+)$$

is provable from provable sequents $U,V \vdash X$ or $U \vdash X$ with U,V,X having at most m primitive types by means of the cut rule only.

Notice that $B,A_n \vdash A_{n+1}$ $(++)$ is of the form $U,V \vdash X$ with $|U|,|V|,|X| \leq m$. A cut rule between the proof of $(+)$ and $(++)$ yields a proof of

$$A_1,\ldots,A_n \vdash A_{n+1}$$

from provable sequents $U,V \vdash X$ or $U \vdash X$ with $|U|,|V|,|X| \leq m$ by means of the cut rule only. □

Theorem 2.31. *Let* Lex *be the lexicon of a Lambek grammar G_L, and let and let m the maximal number of primitive types in a formula of the lexicon. Then the language $L(G_L)$ generated by G_L is the same as the language $L(G_C)$ generated by the following context-free grammar G_C:*

- *Terminals: terminals (words) of G_L*
- *Non-Terminals: all formulae A with $|A| \leq m$*
- *Start symbol S, the one of G_L*
- *$X \longrightarrow a$ whenever $X \in \mathrm{Lex}(a)$*
- *$X \longrightarrow A$ whenever $A \vdash X$ is provable in L*
- *$X \longrightarrow A\,B$ whenever $A,B \vdash X$ is provable in L*

Observe that the rules are in finite number, because there are finitely many sequents $U,V \vdash X$ or $U \vdash X$ when U,V,X contains at most m primitive types — hence there are only finitely many provable such sequents.

Proof. Assume $a_1 \cdots a_n \in L(G_C)$. Hence there exist types $X_i \in \mathrm{Lex}(a_i)$ such that $S \longrightarrow X_1 \cdots X_n$. The derivation in the CFG G_C can be turned into a derivation in L using only the cut rule (reversing $\longrightarrow$ into $\vdash$), therefore $a_1 \cdots a_n \in L(G_L)$.

Assume now that $a_1 \cdots a_n \in L(G_L)$. Hence there exist types $X_i \in \mathrm{Lex}(a_i)$ such that $X_1,\ldots,X_n \vdash S$. By Proposition 2.30 such a sequent is provable by means of the sequents corresponding to production rules, and of the cut rule only. By induction on the size of the cut-only proof, it is easily seen that the proof corresponds to a derivation in the CFG G_C. If the proof is reduced to a proper axiom, than this axiom is itself a production rule. If the last rule is a cut, say between $\Gamma,B,\Theta \vdash C$ and $\Delta \vdash B$, then by induction hypothesis we have $B \longrightarrow \Delta$ and $C \longrightarrow \Gamma\,B\,\Theta$ hence $C \longrightarrow \Gamma\,\Delta\,\Theta$. Thus, if $a_1 \cdots a_n \in L(G_L)$, we have $S \longrightarrow X_1 \cdots X_n$ with $X_i \in \mathrm{Lex}(A_i)$; as $X_i \in \mathrm{Lex}(a_i)$ we have $S \longrightarrow a_1 \cdots a_n$. □

2.12 Concluding Remarks

This concludes our chapter on the Lambek calculus. We have given proofs of many of the classic results (cut elimination, soundness and completeness, equivalence of

Lambek grammars and context-free grammars). In the following chapters, we discuss some variants of the Lambek calculus.

Since complexity is not one of the major themes of the current book, we have decided not to include a recent proof of NP-completeness of the Lambek calculus (Pentus, 2006) in this chapter, although Section 4.6.1 gives a very brief overview of the known complexity results for the Lambek calculus and some of it variants.

In the next chapter we will see another aspect of the Lambek calculus, which is its direct link with natural language semantics in the style of Montague.

Exercises for Chapter 2

Exercise 2.1. Give sequent calculus derivations for each of the following sequents.

1. $(A \setminus B) / A \vdash A \setminus (B / A)$
2. $A / B \vdash (A / C) / (B / C)$
3. $(A \bullet B) \setminus C \vdash B \setminus (A \setminus C)$
4. $B \setminus (A \setminus C) \vdash (A \bullet B) \setminus C$
5. $(B / A) \setminus B \vdash ((A \setminus B) / A) \setminus (A \setminus B)$

Exercise 2.2. Give natural deduction derivations for each of the sequents of the previous exercise.

Exercise 2.3. Prove Proposition 2.3 on page 29.

Exercise 2.4. Definition 2.11 on page 36 defines the order of formulae. Calculate the order of the following formulae.

1. np / n
2. $((np \setminus S) / pp) / np$
3. $(S / np) \setminus S$
4. $((np \setminus S) / np) \setminus (np \setminus S)$
5. $((n / n) / (n / n)) / ((n / n) / (n / n))$

Exercise 2.5. Prove Proposition 2.6 on page 30.

Exercise 2.6. Using the following lexicon, find two different normal derivations in natural deduction for "Someone loves everyone" and "Someone is_missing"

Word	Type(s)
someone	$(S / (np \setminus S))$
everyone	$((S / np) \setminus S)$
loves	$((np \setminus S) / np)$
is_missing	$((S / (np \setminus S)) \setminus S)$
gave	$(((np \setminus S) / pp) / np)$
a_book	np
to	pp / np

Exercise 2.7. Following Carpenter (1996, Section 6.3), look at the following lexicon.

Word	Type(s)
kid	n
who	$((n \setminus n) / (S / np))$
Kelly	np
Terry	np
Robin	np
likes	$((np \setminus S) / np)$
believes	$((np \setminus S) / S)$
knows	$((np \setminus S) / S)$

Give natural deduction derivations showing that the lexicon above allows us to show that all of the following expressions are of type n.

(2.1) kid who Kelly likes.
(2.2) kid who Kelly believes Terry likes.
(2.3) kid who Kelly believes Terry knows Robin likes.

Exercise 2.8. Extend the lexicon from Exercise 2.6 with types for "every" and "a", as well as for "child" and "toy" in such a way that "every child loves a toy" obtains exactly two natural deduction derivations.

Exercise 2.9. Using the lexicon from Exercise 2.6, give two natural deduction derivations for "Someone gave a_book to everyone" .

Exercise 2.10. Section 2.6.2 gives a decision procedure for natural deduction without product. Use this decision procedure to find all proofs for the sequent of Example 2.1 on page 26. How many proofs are there? What is their relation to the natural deduction proof shown in Example 2.1?

References

Abramsky, S.: Computational interpretations of linear logic. Theoretical Computer Science 111, 3–57 (1993)

Autebert, J.M., Boasson, L., Sénizergues, G.: Langages de parenthèses, langages n.t.s. et homomorphismes inverses. RAIRO Informatique Théorique 18(4), 327–344 (1984)

van Benthem, J.: Categorial grammar. In: Essays in Logical Semantics, ch. 7, pp. 123–150. Reidel, Dordrecht (1986)

van Benthem, J.: Categorial grammars and lambda calculus. In: Skordev, D. (ed.) Mathematical Logic and its Applications. Plenum Press (1987)

van Benthem, J.: Language in Action: Categories, Lambdas and Dynamic Logic. Sudies in logic and the foundation of mathematics, vol. 130. North-Holland, Amsterdam (1991)

Buszkowski, W.: Compatibility of a categorial grammar with an asssociated category system. Zeitschrift für Matematische Logik und Grundlagen der Mathematik 28, 229–238 (1982)

Buszkowski, W.: Mathematical linguistics and proof theory. In: van Benthem, J., ter Meulen, A. (eds.) Handbook of Logic and Language, ch. 12, pp. 683–736. North-Holland Elsevier, Amsterdam (1997)

Carpenter, B.: Lectures on Type-Logical Semantics. MIT Press, Cambridge (1996)

Chomsky, N.: Syntactic structures. Janua linguarum. Mouton, The Hague (1957)

Chomsky, N.: Formal properties of grammars. In: Handbook of Mathematical Psychology, vol. 2, pp. 323–418. Wiley, New-York (1963)

Chomsky, N.: The minimalist program. MIT Press, Cambridge (1995)

Cohen, J.M.: The equivalence of two concepts of categorial grammars. Information and Control 10, 475–484 (1967)

Engelfriet, J.: Context-free graph grammars. In: Rosenberg, G., Salomaa, A. (eds.) Handbook of Formal Languages 3: Beyond Words, pp. 125–213. Springer, New York (1997)

Engelfriet, J., Maneth, S.: Tree Languages Generated by Context-Free Graph Grammars. In: Ehrig, H., Engels, G., Kreowski, H.-J., Rozenberg, G. (eds.) TAGT 1998. LNCS, vol. 1764, pp. 15–29. Springer, Heidelberg (2000)

Gécseg, F., Steinby, M.: Tree languages. In: Rozenberg, G., Salomaa, A. (eds.) Handbook of Formal Languages, vol. 3, ch. 1. Springer, Berlin (1997)

Gentzen, G.: Untersuchungen über das logische Schließen. Mathematische Zeitschrift 39, 176–210, 405–431 (1934)

Girard, J.Y.: Linear logic. Theoretical Computer Science 50(1), 1–102 (1987)

Girard, J.Y., Lafont, Y., Taylor, P.: Proofs and Types. Cambridge Tracts in Theoretical Computer Science, vol. 7. Cambridge University Press (1988)

de Groote, P., Retoré, C.: Semantic readings of proof nets. In: Kruijff, G.J., Morrill, G., Oehrle, D. (eds.) Formal Grammar, pp. 57–70. FoLLI, Prague (1996)

Hendriks, H.: Studied flexibility: Categories and types in syntax and semantics. PhD thesis, University of Amsterdam, ILLC Dissertation Series (1993)

Hepple, M.: Normal form theorem proving for the Lambek calculus. In: Proceedings of COLING 1990, Helsinki, pp. 173–178 (1990)

Kanazawa, M., Salvati, S.: On the derivations of lambek grammars (2009) (unpublished manuscript)

König, E.: Parsing as natural deduction. In: Proceedings of the Annual Meeting of the Association for Computational Linguistics, pp. 272–297 (1989)

Lambek, J.: The mathematics of sentence structure. American Mathematical Monthly, 154–170 (1958)

Moortgat, M.: Categorial Investigations: Logical and Linguistic Aspects of The Lambek Calculus. Foris, Dordrecht (1988)
Nivat, M.: Congruences de thue et t-langages. Studia Scientiarum Mathematicarum Hungarica 6, 243–249 (1971)
Pentus, M.: Lambek calculus is L-complete. Tech. Rep. LP-93-14, Institute for Logic, Language and Computation, Universiteit van Amsterdam (1993a)
Pentus, M.: Lambek grammars are context-free. In: Logic in Computer Science. IEEE Computer Society Press (1993b)
Pentus, M.: Product-free Lambek calculus and context-free grammars. Journal of Symbolic Logic 62(2), 648–660 (1997)
Pentus, M.: Lambek calculus is NP-complete. Theoretical Computer Science 357(1), 186–201 (2006)
Retoré, C., Stabler, E.: Generative grammar in resource logics. Research on Language and Computation 2(1), 3–25 (2004)
Roorda, D.: Resource logic: proof theoretical investigations. PhD thesis, FWI, Universiteit van Amsterdam (1991)
Thatcher, J.W.: Characterizing derivation trees of context free grammars through a generalization of finite automata theory. Journal of Computer and System Sciences 1, 317–322 (1967)
Tiede, H.-J.: Lambek Calculus Proofs and Tree Automata. In: Moortgat, M. (ed.) LACL 1998. LNCS (LNAI), vol. 2014, pp. 251–265. Springer, Heidelberg (2001)
Zielonka, W.: Axiomatizability of the Adjukiewicz-Lambek calculus by means of cancellation schemes. Zeitschrift für Matematische Logik und Grundlagen der Mathematik 27(13-14), 215–224 (1981)
Zielonka, W.: A simple and general method of solving the finite axiomatizability problems for Lambek's syntactic calculi. Studia Logica 48(1), 35–39 (1989)