Optional and Iterated Types for Pregroup Grammars

Denis Béchet[1], Alexander Dikovsky[1], Annie Foret[2], and Emmanuelle Garel[2]

[1] LINA CNRS – UMR 6241 – Université de Nantes
2, rue de la Houssiniére – BP 92208
44322 Nantes Cedex 03 – France
Denis.Bechet@univ-nantes.fr
Alexandre.Dikovsky@univ-nantes.fr
[2] IRISA – Université de Rennes 1
Campus Universitaire de Beaulieu
Avenue du Général Leclerc
35042 Rennes Cedex – France
Annie.Foret@irisa.fr
Emmanuelle.Garel@irisa.fr

Abstract. Pregroup grammars are a context-free grammar formalism which may be used to describe the syntax of natural languages. However, this formalism is not able to naturally define types corresponding to optional and iterated arguments such as optional complements of verbs or verbs' adverbial modifiers. This paper introduces two constructions that make up for this deficiency.
Keywords: Pregroups, Lambek Categorial Grammars, Categorial Dependency Grammar.

1 Introduction

Pregroup grammars (PG) [1] have been introduced as a simplification of Lambek calculus [2]. They have been used to model fragments of syntax of several natural languages: English [1], Italian [3], French [4], German [5, 6], Japanese [7], Persian [8], etc. PG are based on the idea that the sentences are derived from words using only lexical rules. The syntactic properties of each word in the lexicon are defined as a finite set of its grammatical categories. These grammatical categories are types of a free pregroup generated by a set of basic types together with a partial order on the basic types. A sentence is correct with respect to a PG if for each word of the sentence, one can find in the lexicon such a type that the concatenation of the selected types can be proved in the pregroup calculus to be inferior than or equal to a particular basic type s. The PG are weakly equivalent to CF-grammars [9]. It doesn't mean that they are suitable to define the syntax of natural languages. In particular, any formalism desined for this purpose should naturally handle the optional and iterated constructions such as noun modifiers, attributes and relative clauses or verb's optional arguments, adverbials or location, manner and other circumstantial clauses. All of them are

optional and their number is not bounded. Whereas the CF-grammars handle such constructions rather naturally, they are problematic for the conventional PG. Like the Lambek categorial grammars, the PG are resource sensitive. In particular, in PG proofs, every simple type (except s) should be linked and reduced with exactly one dual type. So a PG cannot define a simple type that is optional or linked to zero, one, or several duals. This effect can be simulated in PG using complex types, however this simulation has serious disadvantages (see a discussion below) .

In this paper, we propose a different solution to this problem adding new rules to the PG calculus. As a result, we obtain a class of PG with simple optional and iterated types, which express the optional and iterated constructions in the way the dependency grammars do ([10–13]). We prove that the new calculus is decidable.

2 Background

Definition 1 (Pregroup). *A* pregroup *is a structure* $(P, \leq, \cdot, l, r, 1)$ *such that* $(P, \leq, \cdot, 1)$ *is a partially ordered monoid*[3] *and* l, r *are two unary operations on* P *that satisfy for all element* $x \in P$, $x^l x \leq 1 \leq x x^l$ *and* $x x^r \leq 1 \leq x^r x$.

Definition 2 (Free Pregroup). *Let* $(P, \leq)$ *be an ordered set of basic types,* $P^{(\mathbb{Z})} = \{p^{(i)} \mid p \in P, i \in \mathbb{Z}\}$ *be the set of simple types and* $T_{(P,\leq)} = \left(P^{(\mathbb{Z})}\right)^* = \{p_1^{(i_1)} \cdots p_n^{(i_n)} \mid 0 \leq k \leq n, p_k \in P \text{ and } i_k \in \mathbb{Z}\}$ *be the set of types. The empty sequence in* $T_{(P,\leq)}$ *is denoted by* 1. *For* X *and* $Y \in T_{(P,\leq)}$, $X \leq Y$ *iff this relation is derivable in the following system where* $p, q \in P$, $n, k \in \mathbb{Z}$ *and* $X, Y, Z \in T_{(P,\leq)}$:

$$X \leq X \ \ (Id) \qquad \frac{X \leq Y \quad Y \leq Z}{X \leq Z}\ (Cut)$$

$$\frac{XY \leq Z}{Xp^{(n)}p^{(n+1)}Y \leq Z}\ (A_L) \qquad \frac{X \leq YZ}{X \leq Yp^{(n+1)}p^{(n)}Z}\ (A_R)$$

$$\frac{Xp^{(k)}Y \leq Z}{Xq^{(k)}Y \leq Z}\ (IND_L) \qquad \frac{X \leq Yq^{(k)}Z}{X \leq Yp^{(k)}Z}\ (IND_R)$$

$q \leq p$ *if* k *is even, and* $p \leq q$ *if* k *is odd*

This construction, proposed by Buskowski [9], defines a pregroup that extends $\leq$ on basic types P to $T_{(P,\leq)}$[4] [5].

[3] We briefly recall that a *monoid* is a structure $< M, \cdot, 1 >$, such that $\cdot$ is associative and has a neutral element 1 ($\forall x \in M : 1 \cdot x = x \cdot 1 = x$). A partially ordered monoid is a monoid $< M, \cdot, 1 >$ with a partial order $\leq$ that satisfies $\forall a, b, c$: $a \leq b \Rightarrow c \cdot a \leq c \cdot b$ and $a \cdot c \leq b \cdot c$.

[4] Left and right adjoints are defined by $(p^{(n)})^l = p^{(n-1)}$, $(p^{(n)})^r = p^{(n+1)}$, $(XY)^l = Y^l X^l$ and $(XY)^r = Y^r X^r$. We write p for $p^{(0)}$. We also iterate left and right adjoints for every $X \in T_{(P,\leq)}$: $X^{(0)} = X$, $X^{(n+1)} = (X^r)^{(n)}$ and $X^{(n-1)} = (X^l)^{(n)}$

[5] $\leq$ is only a preorder. Thus, in fact, the pregroup is the quotient of $T_{(P,\leq)}$ by the equivalence relation $X \leq Y \ \& \ Y \leq X$.

The Cut Elimination. The cut rule in the Free Pregroup calculus can be eliminated: every derivable inequality has a cut-free derivation.

Definition 3 (Pregroup Grammar). *Let $(P, \leq)$ be a finite partially ordered set. A* pregroup grammar *based on $(P, \leq)$ is a lexicalized*[6] *grammar $G = (\Sigma, I, s)$ such that $s \in T_{(P,\leq)}$. G assigns a type X to a string $v_1 \cdots v_n$ of Σ^* iff for $1 \leq i \leq n$, $\exists X_i \in I(v_i)$ such that $X_1 \cdots X_n \leq X$ in the free pregroup $T_{(P,\leq)}$.* The language $\mathcal{L}(G)$ *is the set of strings in Σ^* that are assigned s by G.*

Example 4. Let us see the following sentence taken from "Un amour de Swann" by M. Proust: *Maintenant, tous les soirs, quand il l'avait ramenée chez elle, il fallait qu'il entrât.* [7] In Fig. 1 we show a proof of correctness of assignment of types to its fragment. The primitive types used in this proof are: π_3 and $\overline{\pi_3}$ = third person (subject) with $\pi_3 \leq \overline{\pi_3}$, p_2 = past participle, ω = object, s = sentence, s_5 = subjunctive clause, with $s_5 \leq s$, σ = complete subjunctive clause, τ = adverbial phrase. This grammar assigns s to the following sentence:

quand il l' avait ramenée chez-elle il fallait qu' il entrât

$\tau s^l \quad \pi_3 \quad \pi_3^r s \omega^{ll} s^l \pi_3 \quad \overline{\pi_3}^r s p_2^l \quad p_2 \omega \lambda^l \quad \lambda \quad \pi_3 \quad \overline{\pi_3}^r \tau^r s \sigma^l \quad \sigma s_5^l \quad \pi_3 \quad \overline{\pi_3}^r s_5$

Figure 1

In more details, this grammar assigns σ to "qu'il entrât", due to:

$$\dfrac{\dfrac{\dfrac{\sigma = \sigma^{(0)} \leq \sigma^{(0)}}{\sigma s_5^l s_5 = \sigma^{(0)} s_5^{(-1)} s_5^{(0)} \leq \sigma^{(0)}}\,(A_L)}{\sigma s_5^l \pi_3 \pi_3{}^r s_5 = \sigma^{(0)} s_5^{(-1)} \pi_3{}^{(0)} \pi_3{}^{(1)} s_5^{(0)} \leq \sigma^{(0)}}\,(A_L)}{\sigma s_5^l \pi_3 \overline{\pi_3}^r s_5 = \sigma^{(0)} s_5^{(-1)} \pi_3{}^{(0)} \overline{\pi_3}^{(1)} s_5^{(0)} \leq \sigma^{(0)}}\,(IND_L)$$

figured as:

qu' il entrât

$\sigma s_5^l \quad \pi_3 \quad \overline{\pi_3}^r s_5$

3 Optional and Iterated Primitive Types

In Fig. 2, we present a more traditional analysis of the sentence in Example 4 represented as a dependency tree.

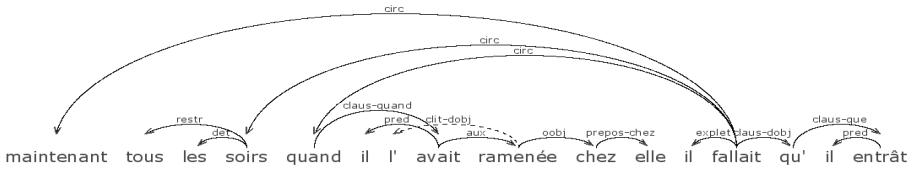

Figure 2

[6] A lexicalized grammar is a triple (Σ, I, s): Σ is a finite alphabet, I assigns a finite set of categories (or types) to each $c \in \Sigma$, s is a category (or type) associated to correct sentences.

[7] [FR: *Now, every evening when he took back her to her home, he ought to enter*]

We see that the verb *fallait* governs three consecutive circumstantial phrases: *Maintenant*, *tous les soirs* and *quand il l'avait ramenée chez elle*. All the three are optional and there might be some more dependent circumstantial phrases in this sequence. We can also remark that the oblique object *chez elle* is optional for the verb *ramener*.

A Dependency Grammar Appoach

In rule-based dependency grammars (cf. [13]), as well as in the local environment dependency grammars of Sleator and Temperly [12], such optional and iterated constructions are defined using restricted regular expressions. In categorial dependency grammars (CDG) [14, 15], which are a kind of categorial grammars with dependency types as categories, these constructions are defined using rather traditional reduction rules. For instance, the left iterated types are defined by the following two rules:

$$\mathbf{I^l}.\ \ a[a^*\backslash\alpha] \vdash [a^*\backslash\alpha]$$
$$\mathbf{\Omega^l}.\ \ [a^*\backslash\alpha] \vdash \alpha$$

Several Pregroup Approaches

Below, we denote the optional types by $a^?$ and the iterated types by a^*. Let us show how such types might be defined in the free pregroup. We see at least three approaches, among which only the last one is considered below in full detail.

A Simulation with Compound Types. The first way to define the optional types might be using the following definitions:

$$a^* \stackrel{\mathbf{Def}}{=} x_{a^*} x^r_{a^*}$$
$$a^? \stackrel{\mathbf{Def}}{=} x_{a^?} y_{a^?} y^r_{a^?} z_{a^?} z^r_{a^?} x^r_{a^?}$$

Here, the basic types x_{a^*}, $x_{a^?}$, $y_{a^?}$ and $z_{a^?}$ must not be used for any other purpose. This simulation is not perfect because the duals of the optional type $x_{a^*} x^r_{a^*}$ or of the iterated type $x_{a^?} y_{a^?} y^r_{a^?} z_{a^?} z^r_{a^?} x^r_{a^?}$ are not simple but compound types and that is problematic. In fact, we have to simulate a^*, a^r and a^l, such as we can obtain the composition of a^*, on the right with a^r and on the left with a^l. We have an other simulation to do with $a^?$, a^r and a^l.

A List-like Simulation In order to simulate an iterated type $[\alpha/a^*]\, a^* \vdash \alpha$ we can distinguish two types, one type a for a first use in a sequence and one type $a^r a$ for next uses in a sequence of elements of type a (this encodes in fact one or more iterations of a). To fully encode a^*, we may add assignments b, whenever ba^* was intended. As in

John	*run*	*fast*	*yesterday*
n	$n^r s a^l$	a	$a^r a$

We have two assignments for run: in "John run", run $\mapsto n^r s$ but in "John run fast, yesterday", "run" $\mapsto n^r s a^l$. Unfortunately, this approach increases the number of types in the lexicon: if a type has k iterated simple types, the simulation associates 2^k types. The same problem occurs with a simulation of an optional simple type using two types, one with the optional simple type and the second without it.

Adding Rules to Pregroup Grammars We propose another definition of the optional and iterated types adding to the PG calculus new rules. Our purpose is to ensure properties such as $a \leq a^?$, $a^*a \leq a^*$, $aa^* \leq a^*$, $1 \leq a^?$, $1 \leq a^*$ (see Corollary 7).

Definition 5 (PG with Optional and Iterated Types). *We add the following rules to a PG that define $p^?$ and p^* for p a basic type*[8]*:*

$$\frac{XY \leq Z}{Xp^{?^{(2k+1)}}Y \leq Z}\,(?-W_L) \qquad \frac{X \leq YZ}{X \leq Yp^{?^{(2k)}}Z}\,(?-W_R)$$

$$\frac{Xp^{(2k+1)}Y \leq Z}{Xp^{?^{(2k+1)}}Y \leq Z}\,(?-D_L) \qquad \frac{X \leq Yp^{(2k)}Z}{X \leq Yp^{?^{(2k)}}Z}\,(?-D_R)$$

$$\frac{XY \leq Z}{Xp^{*^{(2k+1)}}Y \leq Z}\,(*-W_L) \qquad \frac{X \leq YZ}{X \leq Yp^{*^{(2k)}}Z}\,(*-W_R)$$

$$\frac{Xp^{*^{(2k+1)}}p^{(2k+1)}Y \leq Z}{Xp^{*^{(2k+1)}}Y \leq Z}\,(*-C_L) \qquad \frac{X \leq Yp^{(2k)}p^{*^{(2k)}}Z}{X \leq Yp^{*^{(2k)}}Z}\,(*-C_R)$$

$$\frac{Xp^{(2k+1)}p^{*^{(2k+1)}}Y \leq Z}{Xp^{*^{(2k+1)}}Y \leq Z}\,(*-C'_L) \qquad \frac{X \leq Yp^{*^{(2k)}}p^{(2k)}Z}{X \leq Yp^{*^{(2k)}}Z}\,(*-C'_R)$$

As desired, this system enjoys the following property and corollary.

Proposition 6. *Let $U_i \leq 1$ for $0 \leq i \leq n$ and $C_j \leq a$ for $1 \leq j \leq n$. Then $U_0C_1U_1 \leq a^?$ and $\forall k, 1 \leq k \leq n : U_0C_1U_1C_2 \cdots U_k a^* U_{k+1} \cdots C_nU_n \leq a^*$.*

Proof. It is easy to check that if $X_i \leq Y_i$ for $1 \leq i \leq n$, $n \in \mathbb{N}$ then $X_1 \cdots X_n \leq Y_1 \cdots Y_n$. Thus $U_0C_1U_1 \leq a$ and using $(?-D_R)$, we find $U_0C_1U_1 \leq a^?$. Similarly, we have $U_0C_1U_1C_2 \cdots U_k a^* U_{k+1} C_nU_n \leq a \cdots aa^*a \cdots a$. Using $(*-C_R)$ and $(*-C'_R)$, we find $U_0C_1U_1C_2 \cdots U_k a^* U_{k+1} \cdots C_nU_n \leq a^*$.

Corollary 7 (Optional and Iterated Basic Types). *For a, a basic type:*

$$\begin{array}{cc} & a^*a \leq a^* \\ a \leq a^? & aa^* \leq a^* \\ 1 \leq a^? & 1 \leq a^* \end{array}$$

[8] $p^?$ and p^* are considered as incomparable (with respect to the PG order) basic types: Rules (A_L) and (A_R) are valid. (IND_L) and (IND_R) are useless (if $p \neq q$ then $p^? \not\leq q$, $p^? \not\leq q^?$, $p^? \not\leq q^*$, etc)

Theorem 8. *This construction defines a pregroup that extends the free pregroup based on* $(P, \leq)$.

Proof. The structure is a monoid. It is partially ordered (the Cut rule implies transitivity) and moreover with a deduction of $X_1 \leq Y_1$ and a deduction of $X_2 \leq Y_2$, we can build a deduction of $X_1X_2 \leq Y_1Y_2$: the structure is a partially ordered monoid. Finally l and r define the left and right adjoints: the proofs use (A_L), (A_R), (Id).

Example 9. In FiG. 3, we show an analysis of the sentence of Proust in the calculus. The primitive types used below are: π_3 = third person (subject), p_2 = past participle, ω = object, s = sentence, s_5 = subjunctive clause, σ = complete subjunctive clause, d = determinant, ρ = restrictive adjective, τ = adverbial phrase.
$\rho^{?^r}$ corresponds to a left optional restrictive adjective argument, $\lambda^{?^l}$ to a right locative argument and τ^{*^l} (or τ^{*^r}) to right (or left) iterated adverbial phrase arguments.

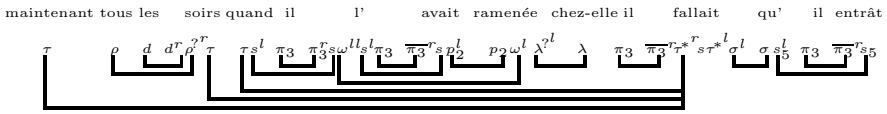

Figure 3

Theorem 10 (The Cut Elimination). *The cut rule can be eliminated in the extended calculus: every derivable inequality has a cut-free derivation.*

Proof. The proof is given in Appendix A.

Corollary 11 (Decidability). *The provability of an inequality in this system is decidable*

Proof. The provability of $X \leq Y$ for the free pregroup based on $(P, \leq)$ (without iterated or optional basic type) is a direct consequence of the elimination of the cut rule because the set of possible premises appearing in a cut-free deduction of $X \leq Y$ is finite. The proof is also correct with the addition of optional basic types. However, for a free pregroup with iterated basic types, this argument in not valid because $(*{-}C_L)$, $(*{-}C_R)$, $(*{-}C'_L)$, $(*{-}C'_R)$ introduce new occurrences of basic types in a cut-free derivation. However, we can limit the number of uses of these rules by the number of basic types in X and Y (this is a consequence of the parity condition on the exponent of the iterated or optional simple type). Thus, even if the search space is infinite, we can limit it to a finite subset.

4 Conclusion

This paper introduces in pregroups two new type constructors ? (option) and $*$ (iteration) allowing to handle in a natural way the optional and iterated constructions such as optional noun modifiers and complements of verbs or their

circumstantials (adverbs, time or location clauses etc.). The extended sequent calculus for pregroups formalising the two constructors has natural properties and is decidable. The future work will concern the complexity of this calculus (see [16]) and the development of a parser for this new class of PG.

References

1. Lambek, J.: Type grammars revisited. In Lecomte, A., Lamarche, F., Perrier, G., eds.: Logical aspects of computational linguistics: Second International Conference, LACL '97, Nancy, France ; selected papers. Volume 1582., Springer-Verlag (1999)
2. Lambek, J.: The mathematics of sentence structure. American Mathematical Monthly **65** (1958) 154–170
3. Casadio, C., Lambek, J.: An algebraic analysis of clitic pronouns in italian. In de Groote, P., Morill, G., Retoré, C., eds.: Logical aspects of computational linguistics: 4th International Conference, LACL 2001, Le Croisic, France, June 2001. Volume 2099., Springer-Verlag (2001)
4. Bargelli, D., Lambek, J.: An algebraic approach to french sentence structure. In de Groote, P., Morill, G., Retoré, C., eds.: Logical aspects of computational linguistics: 4th International Conference, LACL 2001, Le Croisic, France, June 2001. Volume 2099., Springer-Verlag (2001)
5. Lambek, J.: Type grammar meets german word order. Theoretical Linguistics **26** (2000) 19–30
6. Lambek, J., Preller, A.: An algebraic approach to the german noun phrase. Linguistic Analysis **31** (2003) 3–4
7. Cardinal, K.: An algebraic study of Japanese grammar. Master's thesis, McGill University, Montreal (2002)
8. Sadrzadeh, M.: Pregroup analysis of persian sentences (2007)
9. Buszkowski, W.: Lambek grammars based on pregroups. In Groote, P., Morrill, G., Retoré, C., eds.: 4th Intern. Conf. "Logical Aspects of Computational Linguists". Number 2099 in LNAI (2001) 95–109
10. Tesnière, L.: Éléments de syntaxe structurale. Librairie C. Klincksiek, Paris (1959)
11. Hays, D.: Dependency theory: A formalism and some observations. Language **40** (1964) 511–525
12. Sleator, D.D., Temperly, D.: Parsing English with a Link Grammar. In: Proc. IWPT'93. (1993) 277–291
13. Kahane, S., ed.: Les grammaires de dépendance. In Kahane, S., ed.: Traitement automatique des langues. Volume 41., Paris, Hermes (2000) n. 1/2000.
14. Dikovsky, A.: Dependencies as categories. In Kruiff, G.J., Duchier, D., eds.: Proc. of Workshop "Recent Advances in Dependency Grammars". In conjunction with COLING 2004, Geneva, Switzerland (August, 28th 2004) 90–97
15. Dekhtyar, M., Dikovsky, A.: Categorial dependency grammars. In Moortgat, M., Prince, V., eds.: Proc. of Int. Conf. on Categorial Grammars, Montpellier. (2004)
16. Béchet, D., Dikovsky, A., Foret, A., Garel, E.: Introduction of option and iteration into pregroup grammars. In Casadio, C., Lambek, J., eds.: Computational Algebric Approaches to Morphology and Syntax. Polimetrica, Monza (Milan), Italy (2008)
17. Došen, K.: Cut Elimination in Categories. Kluwer Academic publishers, Dordrecht, Boston, London (1999)
18. Buszkowski, W.: Cut elimination for the lambek calculus of adjoints. In Abrusci, V., Casadio, C., eds.: New Perspectives in Logic and Formal Linguisitics, Proceedings Vth ROMA Workshop, Bulzoni Editore (2001)

Appendix A - Cut elimination in S' : proof details

This proof assumes lemmas detailed in the Appendix B. Take again the systems precedently introduced and consider S the system without **Cut** and S' the system with the rule **Cut** Clearly a proof in S is also a proof in S'.

To show the converse, we proceed by induction on the number of **Cut** and on the length of a derivation γ_l, ending in **Cut** in S':

$$\dfrac{\gamma_l \left\{ \begin{array}{cc} \dfrac{\cdots}{X \leq Y} R_l & \dfrac{\cdots}{Y \leq Z} R_r \end{array} \right\} \gamma_r}{X \leq Z}\ \textbf{Cut}$$

– If R_l is the axiom rule, the last rule (cut) can be suppressed since R_r has the same conclusion as $\mathcal{D}$. If R_r is the axiom rule, the last rule (cut) can also be suppressed since R_l has the same conclusion as $\mathcal{D}$. We now assume that neither R_l nor R_r is the axiom rule.

– If R_l is the **Cut** rule, the induction hypothesis applies to γ_l, this **Cut** can be suppressed, in a proof γ_l', and the final **Cut** can be suppressed in this deduction. If R_r is the **Cut** rule, we proceed similarly.
We now assume that neither γ_l or γ_r has a **Cut** rule.

– We consider the remaining possibilities for R_l (left part) and R_r (right part): these cases are detailed below.
- If R_l is a left rule,
Y remains in the antecedent,
we can easily permute R_l with **Cut** .
- If R_l is a right rule,
we apply a lemma or a rule on the right then **Cut** with the antecedent of R_l

R_l	R_r	method
A_L, ...$_L$	–	permute R_l with **Cut**
A_R	–	**Cut** and lemma (B1)
IND_R	–	**Cut** and IND_L
$? - W_R$	–	**Cut** and lemma (B2)
$? - D_R$	–	**Cut** and lemma (B3)
$* - W_R$	–	**Cut** and lemma (B4)
$* - C_R$	–	**Cut** and lemma (B5)
$* - C'_R$	–	**Cut** and lemma (B6)

– A typical case for a left rule is : [$R_l = ? - W_L$]

$$\dfrac{\dfrac{X_1Y_1 \leq Y}{X = X_1p^{?(2k+1)}Y_1 \leq Y}\ ? - W_L \quad \left\{ \begin{array}{c} \dfrac{\cdots}{Y \leq Z} R_r \end{array} \right\} \gamma_r}{X \leq Z}\ \textbf{Cut} \quad \mapsto \quad \dfrac{\dfrac{X_1Y_1 \leq Y \quad \left\{ \begin{array}{c} \dfrac{\cdots}{Y \leq Z} R_r \end{array} \right\} \gamma_r}{X_1Y_1 \leq Z}\ \textbf{Cut}}{X_1p^{?(2k+1)}Y_1 \leq Z}\ ? - W_L$$

– A typical case for a right rule is : [$R_l = ? - W_R$]

$$\dfrac{\dfrac{X_1 \leq Y_1Z_1}{X_1 \leq Y_1p^{?(2k)}Z_1 = Y}\ ? - W_R \quad \left\{ \begin{array}{c} \dfrac{\cdots}{Y \leq Z} R_r \end{array} \right\} \gamma_r}{X \leq Z}\ \textbf{Cut} \quad \mapsto \quad \dfrac{X_1 \leq Y_1Z_1 \quad \dfrac{Y_1p^{?(2k)}Z_1 \leq Z}{Y_1Z_1 \leq Z}\ (lemmaB2)}{X_1 \leq Z}\ \textbf{Cut}$$

The other cases are similar ■

Appendix B - Lemmas for Cut elimination in S'

(1) if $Up^{(n+1)}p'^{(n)}V \leq Z$ with $p'^{(n)} \leq p^{(n)}$ and p, p' are primitive not an iterated or an optional type, then $UV \leq Z$
(1') if $Up^{(n+1)}p^{(n)}V \leq Z$ with p an iterated or an optional type, then $UV \leq Z$
(1") if $UP_{(2k+1)}p^{*(2k)}V \leq Z$ or $Up^{*(2k+2)}P_{(2k+1)}V \leq Z$
where $P_{(2k+1)}$ has the form $\underbrace{p^{(2k+1)}}_{n_1 \text{ times}} \underbrace{p^{*(2k+1)}}_{0 \text{ or } 1 \text{occ.}} \underbrace{p^{(2k+1)}}_{n_2 \text{ times}}$, then $UV \leq Z$
(2) if $Up^{?(2k)}V \leq Z$ then $UV \leq Z$
(3) if $Up^{?(2k)}V \leq Z$ then $Up^{(2k)}V \leq Z$
(4) if $Up^{*(2k)}V \leq Z$ then $UV \leq Z$
(5) if $Up^{*(2k)}V \leq Z$ then $Up^{(2k)}p^{*(2k)}V \leq Z$
(6) if $Up^{*(2k)}V \leq Z$ then $Up^{*(2k)}p^{(2k)}V \leq Z$

Proof These properties are shown *for the system without* **Cut** by induction on the premise of the inequality, according to the last applied rule.

We show (1') and (1") separately after.

– The axiom cases are gathered below, *including those for (1') and (1").*

(1), (1'), (1") case $Up^{(n+1)}p'^{(n)}V = Z$ with $p'^{(n)} \leq p^{(n)}$ or $p = p'$ is an iterated or an optional type

then $\dfrac{\dfrac{\dfrac{UV \leq UV}{}\,A_R}{UV \leq Up^{(n+1)}p^{(n)}V}\,IND_R}{UV \leq Up^{(n+1)}p'^{(n)}V}$ or $\dfrac{\dfrac{UV \leq UV}{}\,A_R}{UV \leq Up^{(n+1)}p^{(n)}V}$ (for $p = p'$ iterated or optional)

(1") Axiom case with $p = q^*$ let $Z = UQ_{(2k+1)}q^{*(2k)}V$ (first form) or $Z = Uq^{*(2k+2)}Q_{(2k+1)}V$ (second form)
where $Q_{(2k+1)}$ has the form $\underbrace{q^{(2k+1)}}_{n_1 \text{ times}} \underbrace{q^{*(2k+1)}}_{n_3} \underbrace{q^{(2k+1)}}_{n_2 \text{ times}}$
We proceed by induction on n_1, n_2, to show that in the first form of Z, if $Z = U \underbrace{q^{(2k+1)}}_{n_1 \text{ times}} \underbrace{q^{*(2k+1)}}_{n_3} \underbrace{q^{(2k+1)}}_{n_2 \text{ times}} q^{*(2k)}V$, then $UV \leq Z$

* for $n_3 = 0; n_1 = n_2 = 0$, we get $UV \leq Z = Uq^{*(2k)}V$, as conclusion of rule $* - W_R$ on $UV \leq UV$
* for $n_3 = 0; n_1 + n_2 > 0$,

$\dfrac{\dfrac{\dfrac{\dfrac{UV \leq UV}{}\,A_R}{UV \leq Uq^{(2k+1)}q^{(2k)}V}\,*W_R}{UV \leq Uq^{(2k+1)}q^{(2k)}q^{*(2k)}V}\,*C_R}{UV \leq Uq^{(2k+1)}q^{*(2k)}V}$ then for $n_1+n_2>1$: $\dfrac{\dfrac{\dfrac{UV \leq U \underbrace{q^{(2k+1)}}_{n_1+n_2-1} q^{*(2k)}V}{}\,A_R}{UV \leq U \underbrace{q^{(2k+1)}}_{n_1+n_2-1} \mathbf{q^{(2k+1)}q^{(2k)}} q^{*(2k)}V}\,*C_R}{UV \leq U \underbrace{q^{(2k+1)}}_{n_1+n_2} q^{*(2k)}V}$

* for $n_3 = 1; n_1 = n_2 = 0$, it is shown above, applying A_R
* for $n_3 = 1$; $n_1 > 0$ or $n_2 > 0$, we start from above when $n_3 = 0$

$$\dfrac{\dfrac{\dfrac{UV \leq U\underbrace{q^{(2k+1)}}_{n_1}q^{*(2k)}V}{UV \leq U\underbrace{q^{(2k+1)}}_{n_1}q^{*(2k+1)}q^{*(2k)}V}*W_L}{UV \leq U\underbrace{q^{(2k+1)}}_{n_1}q^{*(2k+1)}\mathbf{q^{(2k+1)}q^{(2k)}}q^{*(2k)}V}A_R}{UV \leq U\underbrace{q^{(2k+1)}}_{n_1}q^{*(2k+1)}\mathbf{q^{(2k+1)}}q^{*(2k)}V}*C_R$$

we then repeat these last two steps if $n_2 > 1$.

The second form is similar.

(2) case $Up^{?(2k)}V = Z$ then $\dfrac{UV \leq UV}{UV \leq Up^{?(2k)}V = Z} ? - W_R$

(3) case $Up^{?(2k)}V = Z$ then $\dfrac{Up^{(2k)}V \leq Up^{(2k)}V}{Up^{(2k)}V \leq Up^{?(2k)}V = Z} ? - D_R$

(4) case $Up^{*(2k)}V = Z$ then $\dfrac{UV \leq UV}{UV \leq Up^{*(2k)}V = Z} * - W_R$

(5) case $Up^{*(2k)}V = Z$ then $\dfrac{Up^{(2k)}p^{*(2k)}V \leq Up^{(2k)}p^{*(2k)}V}{Up^{(2k)}p^{*(2k)}V \leq Up^{?(2k)}V = Z} * - C_R$

(6) case $Up^{*(2k)}V = Z$ then $\dfrac{Up^{*(2k)}p^{(2k)}V \leq Up^{*(2k)}p^{(2k)}V}{Up^{*(2k)}p^{(2k)}V \leq Up^{?(2k)}V = Z} * - C'_R$

– If *the last rule is a right rule*, it is easy to permute the induction hypothesis with this rule.
– In *all cases distinct from (1), if the last rule is a left rule distinct from* A_L, it cannot create the type $p^{?(2k)}$ or $p^{*(2k)}$ involved in the lemma. We can then permute the induction hypothesis with this rule. The same remark holds in *all cases distinct from (1), if the last rule is* A_L*, but does not create the type* $p^{?(2k)}$ or $p^{*(2k)}$ involved in the lemma.
– In *all cases distinct from (1), if the last rule is* A_L*, and it creates* $p^{?(2k)}$ or $p^{*(2k)}$, let $p' = p^?$ according to case (2)(3), or $p' = p^*$ for (4)(5)(6) such that: $\dfrac{Up'^{(2k)}V = U'V' \leq Z}{U'p'^{(n)}p'^{(n+1)}V' \leq Z} A_L$ with $Up'^{(2k)} = Up'^{(n)}$ or $p'^{(2k)}V = p'^{(n+1)}V'$ *We then apply appropriate rules on* $U'V' \leq Z$*:*
  –in case(2), if $(n = 2k)$ we show $U'p^{?(2k+1)}V' \leq Z$, from $(? - W_L)$ on $U'V' \leq Z$
  –in case(2), if $(n = 2k-1)$ we show $U'p^{?(2k-1)}V' \leq Z$ similarly
  –in case(3), if $(n = 2k)$ we show $U'p^{(2k)}p^{?(2k+1)}V' \leq Z$ by A_R then $(? - D_L)$
  –in case(3), if $(n = 2k-1)$ we show $U'p^{?(2k-1)}p^{(2k)}V' \leq Z$ by A_R then $(? - D_L)$
  –in case(4), if $(n = 2k)$ we show $U'p^{*(2k+1)}V' \leq Z$, from $(* - W_L)$ on $U'V' \leq Z$

–in case(4), if $(n = 2k-1)$ we show $U'p^{*(2k+1)}V' \leq Z$ similarly
–in case(5), if $(n = 2k)$ we show $U'p^{(2k)}p^{*(2k)}p^{*(2k+1)}V' \leq Z$, by A_L on $U'V' \leq Z$ we get : $U'p^{(2k)}p^{(2k+1)}V' \leq Z$,
then by A_L again : $U'p^{(2k)}p^{*(2k)}p^{*(2k+1)}p^{(2k+1)}V' \leq Z$, finally by $* - C_L$.
–in case(5), if $(n = 2k-1)$ we show $U'p^{*(2k-1)}p^{(2k)}p^{*(2k)}V' \leq Z$, similarly : by A_L on $U'V' \leq Z$ we get : $U'p^{*(2k-1)}p^{*(2k)}V' \leq Z$,
then by A_L again : $U'p^{*(2k-1)}p^{(2k-1)}p^{(2k)}p^{*(2k)}V' \leq Z$, finally by $* - C_L$.
–in case(6), if $(n = 2k)$ we show $U'p^{*(2k)}p^{(2k)}p^{*(2k+1)}V' \leq Z$, by A_L on $U'V' \leq Z$ we get : $U'p^{*(2k)}p^{*(2k+1)}V' \leq Z$,
then by A_L again : $U'p^{*(2k)}p^{(2k)}p^{(2k+1)}p^{*(2k+1)}V' \leq Z$, finally by $* - C'_L$.
–in case(6), if $(n = 2k-1)$ we show $U'p^{*(2k-1)}p^{*(2k)}p^{(2k)}V' \leq Z$, similarly : by A_L on $U'V' \leq Z$ we get : $U'p^{(2k-1)}p^{(2k)}V' \leq Z$,
then by A_L again : $U'p^{(2k-1)}p^{*(2k-1)}p^{*(2k)}p^{(2k)}V' \leq Z$, finally by $* - C'_L$.

– In *the inductive case for Lemma(1)*, we consider applications of the rule that can interfere with $p^{(n+1)}p'^{(n)}$ (in the other cases we can then permute the induction hypothesis with this rule) :

- A_L case $\dfrac{U'p'^{(n)}V \leq Z}{U\ \ p^{(n+1)}p'^{(n)}V = U'p^{(n)}\ \ p^{(n+1)}p'^{(n)}V \leq Z}\ A_L$

  $p^{(n)} \leq p'^{(n)}$, if $p = p'$, the premise is the desired inequality, otherwise we apply IND_L
- A_L case (second possibility) the premise is the desired inequality :

  $\dfrac{Up'^{(n+1)}V' \leq Z}{Up^{(n+1)}p'^{(n)}\ \ V = Up^{(n+1)}p'^{(n)}\ \ p'^{(n+1)}V' \leq Z}\ A_L$
- IND_L case if $p'^{(n)} \leq p''^{(n)}$ $\dfrac{Up^{(n+1)}p''^{(n)}V \leq Z}{Up^{(n+1)}p'^{(n)}V \leq Z}\ IND_L$

  we have $p^{(n)} \leq p'^{(n)} \leq p''^{(n)}$ and apply the induction hypothesis on the premise using $p^{(n)} \leq p''^{(n)}$.
- IND_L case if $p^{(n+1)} \leq q^{(n+1)}$ $\dfrac{Uq^{(n+1)}p'^{(n)}V \leq Z}{Up^{(n+1)}p'^{(n)}V \leq Z}\ IND_L$

  we have $q^{(n)} \leq p^{(n)} \leq p'^{(n)}$ and apply the induction hypothesis on the premise using $q^{(n)} \leq p'^{(n)}$.

– *Separate proof for* (1') and (1") . We proceed similarly, with all other cases of the lemma already proved. The axiom cases are already shown. We consider below the case of a left rule that interferes with the formula involved in the lemma (otherwise we can permute induction and the rule):

- $? - W_L$ case, we can write $p = q^?$,

  subcase $n = 2k$ $\dfrac{Uq^{?(2k)}V \leq Z}{Uq^{?(2k+1)}q^{?(2k)}V \leq Z}\ ? - W_L$

  subcase $n = 2k-1$ $\dfrac{Uq^{?(2k)}V \leq Z}{Uq^{?(2k)}q^{?(2k-1)}V \leq Z}\ ? - W_L$

  we then apply lemma B(2) on the premise.

- $? - D_L$ case, we can write $p = q^?$,
  subcase $n = 2k$ subcase $n = 2k-1$

  $$\boxed{\dfrac{Uq^{(2k+1)}q^{?(2k)}V \le Z}{Uq^{?(2k+1)}q^{?(2k)}V \le Z}\ ? - D_L} \qquad \boxed{\dfrac{Uq^{?(2k)}q^{(2k-1)}V \le Z}{Uq^{?(2k)}q^{?(2k-1)}V \le Z}\ ? - D_L}$$

  we then apply lemma B(3) on the premise then B(1) and get the result.
- rules $* - W_L$
  Let $Q_{(2k+1)}$ has the form $\underbrace{q^{(2k+1)}}_{n_1\ times}\underbrace{q^{*(2k+1)}}_{n_3}\underbrace{q^{(2k+1)}}_{n_2\ times}$, where $n_3 \le 1$, and non empty (if empty, we apply lemma (4))

  $$\boxed{\dfrac{U'q^{*(2k)}V \le Z}{UQ_{(2k+1)}\ q^{*(2k)}V = U'\mathbf{q}^{*(\mathbf{2k+1})}\ q^{*(2k)}V \le Z}\ * - W_L}$$

  we have $UQ_{(2k+1)} = U'q^{*(2k+1)}$, $n_2 = 0$, $U' = U\ \underbrace{q^{(2k+1)}}_{n_1\ times}$ we then apply the induction on the premise, that has a similar form.
  The case $Uq^{*(2k+2)}Q_{(2k+1)}V \le Z$ is similar.
- $* - C_L$ case, subcase $n = 2k$.
  Let $Q_{(2k+1)}$ has the form $\underbrace{q^{(2k+1)}}_{n_1\ times}\underbrace{q^{*(2k+1)}}_{n_3}\underbrace{q^{(2k+1)}}_{n_2\ times}$, where $n_3 \le 1$, and non empty (if empty, we apply lemma (4))

  $$\boxed{\dfrac{UQ_{(2k+1)}q^{(2k+1)}\ q^{*(2k)}V = U'\mathbf{q}^{*(\mathbf{2k+1})}\mathbf{q}^{(\mathbf{2k+1})}q^{*(2k)}V \le Z}{UQ_{(2k+1)}\ q^{*(2k)}V = U'\mathbf{q}^{*(\mathbf{2k+1})}\ q^{*(2k)}V \le Z}\ * - C_L}$$

  we have $UQ_{(2k+1)} = U'q^{*(2k+1)}$, $U' = U\ \underbrace{q^{(2k+1)}}_{n_1\ times}$,
  we then apply the induction on the premise, that has a similar form.
- $* - C_L$ case, subcase $n = 2k-1$.
  Let $Q_{(2k-1)}$ has the form $\underbrace{q^{(2k-1)}}_{n_1\ times}\underbrace{q^{*(2k-1)}}_{n_3}\underbrace{q^{(2k-1)}}_{n_2\ times}$, where $n_3 \le 1$ and non empty (if empty, we apply lemma (4))

  $$\boxed{\dfrac{Uq^{*(2k)}\ \mathbf{q}^{*(\mathbf{2k-1})}\mathbf{q}^{(\mathbf{2k-1})}V' \le Z}{Uq^{*(2k)}\ Q_{(2k-1)}V = Uq^{*(2k)}\ \mathbf{q}^{*(\mathbf{2k-1})}V' \le Z}\ * - C_L}$$

  we have $Q_{(2k-1)}V = q^{*(2k-1)}V'$, $V' = \underbrace{q^{(2k-1)}}_{n_2\ times}V$,
  we then apply the induction on the premise, that has a similar form.
- rules $* - C'_L$ can be treated similarly to $* - C_L$