TECHNISCHE UNIVERSITÄT WIEN
Vienna University of Technology

DIPLOMARBEIT

Cuts Without Quantifier Alternations and Their Effect on Expansion Trees

Ausgeführt am Institut für

Diskrete Mathematik und Geometrie
der Technischen Universität Wien

unter Anleitung von

Privatdoz. Dr.techn. Stefan Hetzl

durch

Sebastian Zivota, BSc
Penzinger Straße 71
1140 Wien

Datum

Unterschrift

Contents

Introduction

In the year 1934, Gerhard Gentzen introduced his *sequent calculus* **LK**, a refinement of his earlier calculus of natural deduction (cf. [Gen34]). His aim was to prove the consistency of arithmetic, and to this end he formulated and proved the cut elimination theorem ("Gentzen's Hauptsatz"). Briefly, the cut elimination theorem shows that any sequent proof can be algorithmically transformed into a proof that does not use the cut rule; proofs with this property are called *cut-free* or *analytic*. The only formulas that can occur in a cut-free proof are instances of subformulas of the end sequent. This so-called *subformula property* significantly simplifies the structure of possible proofs of a given sequent, making cut-free proofs very convenient as theoretical objects. Gentzen himself did not originally intend cut elimination to be applied to actual mathematical proofs; it was Georg Kreisel who advocated using it to extract constructive content from nonconstructive proofs, cf. [Kre51], [Kre52]. Luckhardt used Kreisel's method to obtain a polynomial bound on the number of exceptionally good rational approximations in the Thue-Siegel-Roth theorem (see [Luc89]); a similar bound was found independently by Bombieri and van der Poorten using number-theoretical methods, see [Bom88]. Kohlenbach carries on Kreisel's ideas under the name "Applied Proof Theory", cf. [Koh08].

One way to formalize the "constructive content" of a proof is *Herbrand's Theorem*: In a simple formulation, a formula $\exists\bar{x}A$ where A is quantifier-free is valid iff there is a tautological disjunction $\bigvee_{i=1}^{n} A[\bar{x}\backslash\bar{t}_i]$ of instances of A, a so-called *Herbrand disjunction*. Given a cut-free proof π^* of $\exists\bar{x}A$, we can obtain this disjunction by collecting all the quantifier inferences in π^*. Essentially, the Herbrand disjunction contains all the witness terms that we use to prove an existential formula valid. We can also write this disjunction over instances as a set of instances and call it an *Herbrand set*.

In formalizing actual mathematical proofs, one will inevitably have to use the cut rule. How should we treat such formalized proofs that contain cuts? By the cut elimination theorem, we could certainly transform a given proof into a cut-free one and proceed from there. But cut elimination is a tedious process, and as shown (independently) by Statman and Orevkov, cut-free proofs can be nonelementarily large compared to their counterparts with cuts (see [Sta79], [Ore79], [Pud98]), so one would prefer a method that circumvents cut elimination and extracts an Herbrand set directly from the original proof. The approach that we discuss in this thesis is to extract from a proof π a tree grammar $G(\pi)$ such that the language of $G(\pi)$ is a tautological set of instances. This is accomplished by carrying out cut elimination on π according to a certain strategy. Stefan

Hetzl proved this result in [Het12a] for proofs whose end sequents consist of a single prenex formula and which only contain cuts with at most one quantifier using totally rigid tree grammars, as defined in [Jac11]; in this sense, this thesis generalizes [Het12a].

In [Het12b], Hetzl and Straßburger presented the stronger result that if π can be transformed by cut elimination into a cut-free proof π^*, then the language of $G(\pi)$ is an upper bound for the Herbrand set we would obtain from π^*.

The notion of a Herbrand disjunction can be extended to more complicated formulas: it is well-known that an arbitrary formula is valid iff it has an *expansion proof*, which is a tree that combines constructive information of an Herbrand disjunction with a more complex propositional structure. Expansion trees were first proposed by Dale Miller in [Mil87]. In analogy to the simple prenex case, we can extract an expansion proof of a valid formula from a given cut-free sequent proof of that formula. Moreover, the tree can then be "flattened" in order to obtain a set of instances that is tautological.

The principal result of this thesis is the development of a new type of tree grammar, the so-called *constrained grammars*. A constrained grammar is a tree grammar together with a propositional formula that determines which combinations of productions can and cannot occur in derivations, which is an essential feature if one is dealing with nested quantifiers and logical connectives. It turns out that constrained grammars generalize totally rigid tree grammars, and in fact the constrained grammar $G(\pi)$ associated with a proof π is always totally rigid.

Using constrained grammars, we have managed to extend the results of [Het12a] to proofs whose end sequent consists of arbitrarily many Boolean combinations of prenex formulas and whose cut formulas may not contain quantifier alternations, but are not otherwise restricted. The proof strategy is inspired by [Het12b] in that we work with a local cut reduction procedure.

CHAPTER 1

Proofs

1.1 Basic definitions

Definition 1.1 (Sequent)**.** A *sequent* is an ordered pair of finite multisets of formulas $(\{A_1,\ldots,A_m\},\{B_1,\ldots,B_n\})$, written as $A_1,\ldots,A_m \vdash B_1,\ldots,B_n$. The first component of a sequent is called the *antecedent*, the second is called the *succedent* and both are referred to as *cedents*. Note that while cedents are usually written as lists, they are actually multisets and hence the order in which their elements appear is irrelevant.

The natural interpretation of the sequent $A_1,\ldots,A_m \vdash B_1,\ldots,B_n$ is "if all of the A_i hold, then one of the B_j holds" (or, more formally, this sequent can be interpreted as the formula $\neg A_1 \vee \ldots \vee \neg A_m \vee B_1 \vee \ldots \vee B_n$). As a consequence, concepts such as models, satisfiability and validity of sequents are well-defined. The special cases $A_1,\ldots,A_m \vdash$ and $\vdash$ are interpreted as "$A_1,\ldots,A_m$ lead to a contradiction" and "there is a contradiction", respectively. $\vdash$ is called the *empty sequent* and usually written as $\bot$.
We generally use uppercase Greek letters $\Gamma,\Pi,\Delta,\Lambda,\ldots$ to denote cedents. If we want to emphasize one or more formulas in either cedent, we will use the notation $\Gamma,A,B,C,\ldots$.

The logical calculus we use in this thesis is a variant of the system that Gentzen introduced in [Gen34] under the name **LK**, for "Logistikkalkül klassisch". **LK** is a sequent calculus, which means that it describes a method of deriving sequents from other sequents via certain inference rules. Before we can list the inference rules of **LK**, we need to define the substitution of terms in formulas.

Definition 1.2 (Substitution operator)**.** Let $s_1,\ldots,s_n,t_1,\ldots,t_n$ be terms. The *substitution operator* $[s_1\backslash t_1,\ldots,s_n\backslash t_n]$ acts on a formula A by replacing every occurrence of each s_i in A with t_i. The result is written as $A[s_1\backslash t_1,\ldots,s_n\backslash t_n]$, i.e. the operator is written as a postfix. Note that $A[s_1\backslash t_1,\ldots,s_n\backslash t_n]$ is only defined if the t_i do not contain any variables that would become bound in A.

Definition 1.3 (Inference rules)**.** An *inference rule* is an expression of the form $\dfrac{\Gamma' \vdash \Delta'}{\Gamma \vdash \Delta}\ r$ or $\dfrac{\Gamma' \vdash \Delta' \quad \Gamma'' \vdash \Delta''}{\Gamma \vdash \Delta}\ r$, where r is the name of the rule. The inference rules of **LK** are:

1. Contraction:

$$\frac{A,A,\Gamma \vdash \Delta}{A,\Gamma \vdash \Delta}\ c_l \qquad\qquad \frac{\Gamma \vdash \Delta,A,A}{\Gamma \vdash \Delta,A}\ c_r$$

2. Weakening:

$$\frac{\Gamma \vdash \Delta}{A,\Gamma \vdash \Delta}\ w_l \qquad\qquad \frac{\Gamma \vdash \Delta}{\Gamma \vdash \Delta,A}\ w_r$$

3. Propositional rules:

$$\frac{A,\Gamma \vdash \Delta \quad B,\Pi \vdash \Lambda}{A \vee B,\Gamma,\Pi \vdash \Delta,\Lambda}\ \vee_l \qquad\qquad \frac{\Gamma \vdash \Delta, A, B}{\Gamma \vdash \Delta, A \vee B}\ \vee_r$$

$$\frac{A, B, \Gamma \vdash \Delta}{A \wedge B, \Gamma \vdash \Delta}\ \wedge_l \qquad\qquad \frac{\Gamma \vdash \Delta, A \quad \Pi \vdash \Lambda, B}{\Gamma,\Pi \vdash \Delta,\Lambda, A \wedge B}\ \wedge_r$$

$$\frac{\Gamma \vdash \Delta, A}{\neg A, \Gamma \vdash \Delta}\ \neg_l \qquad\qquad \frac{A, \Gamma \vdash \Delta}{\Gamma \vdash \Delta, \neg A}\ \neg_r$$

4. Quantifier rules:

$$\frac{A[x\backslash t], \Gamma \vdash \Delta}{\forall x A, \Gamma \vdash \Delta}\ \forall_l \qquad\qquad \frac{\Gamma \vdash \Delta, A[x\backslash \alpha]}{\Gamma \vdash \Delta, \forall x A}\ \forall_r$$

$$\frac{A[x\backslash \alpha], \Gamma \vdash \Delta}{\exists x A, \Gamma \vdash \Delta}\ \exists_l \qquad\qquad \frac{\Gamma \vdash \Delta, A[x\backslash t]}{\Gamma \vdash \Delta, \forall x A}\ \exists_r$$

   Here, t is any term, while α is a variable that does not occur in Γ, Δ or A, called an *eigenvariable*. The inferences that use eigenvariables are called *strong quantifier inferences*, the others *weak quantifier inferences*.

5. The cut rule:

$$\frac{\Gamma \vdash \Delta, A \quad A, \Pi \vdash \Lambda}{\Gamma,\Pi \vdash \Delta,\Lambda}\ cut$$

   The formula A is called the *cut formula* of the inference.

In all of these cases, the sequents above the line are called *premises* and the sequent below the line is called the *conclusion*. An inference is called *unary* or *binary* if it has one or two premises, respectively. Moreover, in all of these cases except *cut*, the formula that is emphasized in the conclusion is called the *main formula*, while those that are emphasized in the premises are called *auxiliary formulas*.

It is easy to verify that all of these rules are *sound*, i.e. if their premises are valid, then so are their conclusions.

Strictly speaking, there is a difference between *inference rules* and *inferences*: An inference rule is a template for generating inferences; an inference is a concrete instance of an inference rule. In practice we will mostly use the term "inference" and trust that the meaning can be derived from context.

If c is a cut in a proof π, i.e. a concrete instance of the cut rule, then we may assign various attributes to c according to the form of its cut formula A_c. For example, if A_c is quantifier-free, we call c an *unquantified cut*.

Definition 1.4 (**LK**-proof)**.** Let $\Gamma \vdash \Delta$ be a sequent. An ***LK**-proof* of $\Gamma \vdash \Delta$ is a finite tree π of sequents such that

1. The root of π is $\Gamma \vdash \Delta$.
2. The leaves of π are sequents of the form $A \vdash A$ with A atomic; such sequents are called *axioms*.
3. Every inner node $\Pi \vdash \Lambda$ of π has one or two children $\Pi' \vdash \Lambda'$ (and $\Pi'' \vdash \Lambda''$) such that

$$\frac{\Pi' \vdash \Lambda'}{\Pi \vdash \Lambda} \text{ or } \qquad \frac{\Pi' \vdash \Lambda' \quad \Pi'' \vdash \Lambda''}{\Pi \vdash \Lambda}$$

is an inference of **LK**.

$\Gamma \vdash \Delta$ is called *provable* if there is an **LK**-proof of $\Gamma \vdash \Delta$. A subtree of π that is itself a proof is called a *subproof* of π.
A proof in which no two strong quantifier inferences use the same eigenvariable is said to be *regular*; from this point on we will always assume regularity without explicitly stating it.

Proofs are visualized with the root at the bottom and the leaves at the top; as such, the notions of "upward" and "downward" are well-defined. Note that any formula may occur multiple times within a proof and even within a single inference, but it is often necessary to refer to a concrete occurrence of a formula. Consequently, we use letters μ,ν, . . ., possibly with subscripts, to refer to individual occurrences of formulas in a proof or inference. We will denote such formulas by $A_{[\mu]}$ etc. The same method will be used to refer to concrete inferences within a proof.

Theorem 1.5. ***LK** is sound, i.e. if $\Gamma \vdash \Delta$ is **LK**-provable, it is valid.*

Proof. By induction on the depth of proofs. Proofs of depth 1 consist of a single axiom, which is certainly valid. If π is a proof of depth $n + 1$, then its final inference ι is either unary or binary and the premises of ι have proofs of length at most n. It follows that these premises are valid and because ι preserves validity, so is $\Gamma \vdash \Delta$. □

All inference rules of **LK** apart from *cut* obey the so-called *subformula property*: The only formulas that appear in the premises of these inferences are subformulas (or instances, in the case of quantifier inferences) of formulas in the conclusion. By extension, if a proof does not contain cuts, the only formulas that occur anywhere within it are instances of subformulas of the end sequent. In other words, *cut* is the only rule that completely erases a formula (when reading top-down) or introduces it from nowhere (when reading bottom-up). For this reason, *cut-free proofs*, i.e. proofs that do not contain any cuts, are of particular interest. The question naturally arises whether **LK** loses any of its strength when the *cut* rule is removed. It turns out that **LK** is complete even without *cut*.

Theorem 1.6. ***LK*** *without cut is complete, i.e. if $\Gamma \vdash \Delta$ is valid, then there is a cut-free* ***LK****-proof of $\Gamma \vdash \Delta$. As a consequence,* ***LK*** *is complete as well.*

Sketch of proof. We show the following equivalent theorem: If $\Gamma \vdash \Delta$ is any sequent, then either $\Gamma \vdash \Delta$ has a countermodel or it is provable without cut. The proof proceeds by taking $\Gamma \vdash \Delta$ as the root of a tree and then iteratively enumerating all inferences that could have led to the current top level of the tree. If this process terminates, then the resulting tree can be converted to a cut-free proof of $\Gamma \vdash \Delta$; if it does not terminate, we can use it to construct a countermodel. For a rigorous proof, see [Tak87]. □

1.2 Eliminating cuts

If a sequent $\Gamma \vdash \Delta$ has a proof π, then it is valid by Theorem 1.5 and we can construct a cut-free proof π' of $\Gamma \vdash \Delta$ by Theorem 1.6. This shows that provability and cut-free provability of sequents are equivalent. The problem with this approach is that the cut-free proof π' is constructed from scratch and has no relation to the original proof π. This raises the question of whether we can obtain a cut-free proof of $\Gamma \vdash \Delta$ by transforming π.

In this section we will define a relation $\pi \rightsquigarrow \pi'$ between proofs π, π' of the same end sequent signifying that π' is obtained from π either by reducing a cut or by applying a transformation that makes reduction of a cut more convenient. All of these operations are sound, i.e. they transform correct proofs into correct proofs.

Definition 1.7 (Cut reduction)**.** Let c be a cut in a proof π and let A_c be the cut formula of c. We define the following methods of cut reduction according to the inferences immediately above the cut:

1. On one side of c, there is a unary or binary inference r whose active formula is not A_c:

$$\dfrac{\overset{(\psi_1)}{\Gamma \vdash \Delta, A_c} \qquad \dfrac{\overset{(\psi_2)}{A_c, \Pi' \vdash \Lambda'}}{A_c, \Pi \vdash \Lambda}\, r}{\Gamma, \Pi \vdash \Delta, \Lambda}\, cut_{[c]} \rightsquigarrow \dfrac{\dfrac{\overset{(\psi_1)}{\Gamma \vdash \Delta, A_c} \qquad \overset{(\psi_2)}{A_c, \Pi' \vdash \Lambda'}}{\Gamma, \Pi' \vdash \Delta, \Lambda'}\, cut_{[c]}}{\Gamma, \Pi \vdash \Delta, \Lambda}\, r$$

$$\dfrac{\overset{(\psi_1)}{\Gamma \vdash \Delta, A_c} \qquad \dfrac{\overset{(\psi_2)}{A_c, \Pi_1 \vdash \Lambda_1} \qquad \overset{(\psi_2)}{\Pi_2 \vdash \Lambda_2}}{A_c, \Pi \vdash \Lambda}\, r}{\Gamma, \Pi \vdash \Delta, \Lambda}\, cut_{[c]} \rightsquigarrow \dfrac{\dfrac{\overset{(\psi_1)}{\Gamma \vdash \Delta, A_c} \qquad \overset{(\psi_2)}{A_c, \Pi_1 \vdash \Lambda_1}}{\Gamma, \Pi_1 \vdash \Delta, \Lambda_1}\, cut_{[c]} \qquad \overset{(\psi_3)}{\Pi_2 \vdash \Lambda_2}}{\Gamma, \Pi \vdash \Delta, \Lambda}\, \iota$$

The case where ι is on the left side of c works entirely symmetrically.

2. A_c is introduced by an axiom on one side of c:

$$\dfrac{A_c \vdash A_c \qquad \overset{(\psi)}{A_c, \Gamma \vdash \Delta}}{A_c, \Gamma \vdash \Delta}\, cut_{[c]} \rightsquigarrow \overset{(\psi)}{A_c, \Gamma \vdash \Delta}$$

3. A_c is introduced by a weakening on one side of c:

$$\dfrac{\dfrac{\overset{(\psi_1)}{\Gamma \vdash \Delta}}{\Gamma \vdash \Delta, A_c}\, w_r \qquad \overset{(\psi_2)}{A_c,\Pi \vdash \Lambda}}{\Gamma,\Pi \vdash \Delta,\Lambda}\, cut_{[c]} \quad \rightsquigarrow \quad \dfrac{\overset{(\psi_1)}{\Gamma \vdash \Delta}}{\Gamma,\Pi \vdash \Delta,\Lambda}\, w*$$

The case where the weakening is on the right side is symmetrical.

4. A_c is the main formula of a contraction on one side of c:

$$\dfrac{\dfrac{\overset{(\psi_1)}{\Gamma \vdash \Delta,A_c,A_c}}{\Gamma \vdash \Delta,A_c}\, c_r \qquad \overset{(\psi_2)}{A_c,\Pi \vdash \Lambda}}{\Gamma,\Pi \vdash \Delta,\Lambda}\, cut_{[c]} \quad \rightsquigarrow \quad \dfrac{\dfrac{\dfrac{\overset{(\psi_1)}{\Gamma \vdash \Delta,A_c,A_c} \qquad \overset{(\psi_2'')}{A_c,\Pi \vdash \Lambda}}{\Gamma,\Pi \vdash \Delta,\Lambda,A_c}\, cut_{[c'']} \qquad \overset{(\psi_2')}{A_c,\Pi \vdash \Lambda}}{\Gamma,\Pi,\Pi \vdash \Delta,\Lambda,\Lambda}\, cut_{[c']}}{\Gamma,\Pi \vdash \Delta,\Lambda}\, c*$$

Here, ψ_2' and ψ_2'' each arise from ψ_2 by replacing all eigenvariables occurring in ψ_2 with fresh copies. The case where the contraction is on the right is treated analogously.

5. $A_c = \exists x B$ and A_c is introduced by $\exists$-inferences immediately above the cut:

$$\dfrac{\dfrac{\overset{(\psi_1)}{\Gamma \vdash \Delta,B[x\backslash t]}}{\Gamma \vdash \Delta,\exists x B}\, \exists_r \qquad \dfrac{\overset{(\psi_2)}{B[x\backslash\alpha],\Pi \vdash \Lambda}}{\exists x B,\Pi \vdash \Lambda}\, \exists_l}{\Gamma,\Pi \vdash \Delta,\Lambda}\, cut_{[c]} \quad \rightsquigarrow \quad \dfrac{\overset{(\psi_1)}{\Gamma \vdash \Delta,B[x\backslash t]} \qquad \overset{(\psi_2[\alpha\backslash t])}{B[x\backslash t],\Pi \vdash \Lambda}}{\Gamma,\Pi \vdash \Delta,\Lambda}\, cut_{[c']}$$

6. $A_c = \forall x B$: Analogous to the previous case, but with switched sides.

7. $A_c = B \wedge C$ and A_c is introduced by $\wedge$-inferences immediately above the cut:

$$\dfrac{\dfrac{\overset{(\psi_1)}{\Gamma_1 \vdash \Delta_1,B} \qquad \overset{(\psi_2)}{\Gamma_2 \vdash \Delta_2,C}}{\Gamma \vdash \Delta,B \wedge C}\, \wedge_r \qquad \dfrac{\overset{(\psi_3)}{B,C,\Pi \vdash \Lambda}}{B \wedge C,\Pi \vdash \Lambda}\, \wedge_l}{\Gamma,\Pi \vdash \Delta,\Lambda}\, cut_{[c']} \quad \rightsquigarrow$$

$$\rightsquigarrow \quad \dfrac{\overset{(\psi_1)}{\Gamma_1 \vdash \Delta_1,B} \qquad \dfrac{\overset{(\psi_2)}{\Gamma_2 \vdash \Delta_2,C} \qquad \overset{(\psi_3)}{C,B,\Pi \vdash \Lambda}}{B,\Gamma_2,\Pi \vdash \Delta_2,\Lambda}\, cut_{[c'']}}{\Gamma,\Pi \vdash \Delta,\Lambda}\, cut_{[c']}$$

8. $A_c = B \vee C$: Analogous to the previous case.

9. $A_c = \neg B$ and both $\neg$-inferences introducing A_c are immediately above the cut:

$$\dfrac{\dfrac{\overset{(\psi_1)}{B,\Gamma \vdash \Delta}}{\Gamma \vdash \Delta,\neg B}\, \neg_r \qquad \dfrac{\overset{(\psi_2)}{\Pi \vdash \Lambda,B}}{\neg B,\Pi \vdash \Lambda}\, \neg_l}{\Gamma,\Pi \vdash \Delta,\Lambda}\, cut_{[c]} \quad \rightsquigarrow \quad \dfrac{\overset{(\psi_2)}{\Pi \vdash \Lambda,B} \qquad \overset{(\psi_1)}{B,\Gamma \vdash \Delta}}{\Gamma,\Pi \vdash \Delta,\Lambda}\, cut_{[c']}$$

If π' arises from π by finitely many applications of these rules, then we write $\pi \rightsquigarrow^* \pi'$. If no instances of rule 3 are used, we write $\pi \rightsquigarrow^*_{ne} \pi'$ (for *non-erasing*).

Theorem 1.8 (Gentzen). *Let π be a proof of $\Gamma \vdash \Delta$. Then there is a cut-free proof π^* of $\Gamma \vdash \Delta$ such that $\pi \rightsquigarrow^* \pi^*$.*

A system of reducing objects to certain normal forms is called *weakly normalizing* if a normal form can always be reached in a finite number of steps for any initial object. Thus, Gentzen's theorem can be reformulated as "Cut reduction is weakly normalizing". On the other hand, cut reduction is not *strongly normalizing*, i.e. there are transformation sequences that do not terminate. Moreover, cut reduction is not *confluent*, which means that given a proof π, it is possible to end up with different cut-free proofs π', π''.

Example 1.9. Consider the proof

$$\dfrac{\dfrac{\begin{matrix}(\psi_1)\\ \Gamma \vdash \Delta\end{matrix}}{\Gamma \vdash \Delta, A}\,w_r \qquad \dfrac{\begin{matrix}(\psi_2)\\ \Pi \vdash \Lambda\end{matrix}}{A, \Pi \vdash \Lambda}\,w_l}{\Gamma, \Pi \vdash \Delta, \Lambda}\,cut$$

and assume that ψ_1 and ψ_2 are cut-free. Since the cut formula is introduced by a weakening on both sides, this proof can be reduced to two different cut-free proofs

$$\dfrac{\begin{matrix}(\psi_1)\\ \Gamma \vdash \Delta\end{matrix}}{\Gamma, \Pi \vdash \Delta, \Lambda}\,w* \qquad\qquad \dfrac{\begin{matrix}(\psi_2)\\ \Pi \vdash \Lambda\end{matrix}}{\Gamma, \Pi \vdash \Delta, \Lambda}\,w*$$

This shows the non-confluence of cut reduction.

Definition 1.10 (Pruning). Let π and π' be proofs of the same end sequent. We say that π' is the result of *"pruning"* π, written as $\pi \overset{pr}{\rightsquigarrow} \pi'$, if π' is obtained from π by the following subproof transformation:

$$\begin{matrix}
\dfrac{\begin{matrix}(\psi)\\ A[x\backslash\beta], \Gamma'' \vdash \Delta''\end{matrix}}{\exists x A, \Gamma'' \vdash \Delta''}\,\exists_l \\
\vdots \\
\dfrac{A[x\backslash\alpha], \Gamma' \vdash \Delta'}{\exists x A, \Gamma' \vdash \Delta'}\,\exists_l \\
\vdots \\
\dfrac{C[\exists x A], C[\exists x A], \Gamma \vdash \Delta}{C[\exists x A], \Gamma \vdash \Delta}\,c_l
\end{matrix}
\quad \overset{pr}{\rightsquigarrow} \quad
\begin{matrix}
\dfrac{\begin{matrix}(\psi[\beta\backslash\alpha])\\ A[x\backslash\alpha], \Gamma'' \vdash \Delta''\end{matrix}}{\exists x A, A[x\backslash\alpha], \Gamma'' \vdash \Delta''}\,w_l \\
\vdots \\
\dfrac{\dfrac{A[x\backslash\alpha], A[x\backslash\alpha], \Gamma' \vdash \Delta'}{A[x\backslash\alpha], \Gamma' \vdash \Delta'}\,c_l}{\exists x A, \Gamma' \vdash \Delta'}\,\exists_l \\
\vdots \\
\dfrac{C[\exists x A], C[\exists x A], \Gamma \vdash \Delta}{C[\exists x A], \Gamma \vdash \Delta}\,c_l
\end{matrix}$$

We say that a proof is *"pruned"* if it cannot be pruned further.

CHAPTER 2

Expansion Trees

Expansion trees were originally presented by Miller in [Mil87] as a generalization of Herbrand disjunctions for higher-order logic. For our purposes, it will be sufficient to develop them only with respect to first-order logic.

Expansion trees are conceptually very similar to formulas in that they are constructed as trees from atomic formulas and logical connectives. The connectives used in expansion trees are $\neg, \vee, \wedge, \forall, \exists$, as in first-order formulas, and additionally a new symbol $+^t$ where any term may be inserted for t.

Definition 2.1 (Expansion tree, dual expansion tree, deep formula)**.** Let A be a formula. *Expansion trees* and *dual expansion trees* of A are trees recursively defined by the rules below. Each (dual) expansion tree also has an associated *deep formula*, denoted by Dp, that is constructed along with it.

1. $\bot$ is an expansion tree of any formula and $Dp(\bot) := \bot$. $\top$ is a dual expansion tree of any formula and $Dp(\top) := \top$.
2. If A is atomic, then A is an expansion tree and a dual expansion tree of itself and $Dp(A) := A$.
3. If E is an expansion tree of A, then $\neg E$ is a dual expansion tree of $\neg A$. If E is a dual expansion tree of A, then $\neg E$ is an expansion tree of $\neg A$. In both cases, $Dp(\neg E) := \neg Dp(E)$.
4. Let $\diamond \in \{\vee, \wedge\}$. If E and F are (dual) expansion trees of A and B respectively, then $E \diamond F$ is a (dual) expansion tree of $A \diamond B$ and $Dp(E \diamond F) := Dp(E) \diamond Dp(F)$.
5. If $A[x \backslash \alpha]$ has an expansion tree E, then $\forall x A +^{\alpha} E$ is an expansion tree of $\forall x A$ and $Dp(\forall x A +^{\alpha} E) := Dp(E)$.
   If $A[x \backslash \alpha]$ has a dual expansion tree E, then $\exists x A +^{\alpha} E$ is a dual expansion tree of $\exists x A$ and $Dp(\exists x A +^{\alpha} E) := Dp(E)$.
6. If $t_1, \dots, t_n$ are distinct terms such that $E_1, \dots, E_n$ are expansion trees of $A[x \backslash t_1], \dots, A[x \backslash t_n]$, then $\exists x A +^{t_1} E_1, \dots, +^{t_n} E_n$ is an expansion tree of $\exists x A$ and $Dp(\exists x A +^{t_1} E_1, \dots, +^{t_n} E_n) := \bigvee_{i=1}^{n} Dp(E_i)$.

If $E_1, \dots, E_n$ are dual expansion trees of $A[x\backslash t_1], \dots, A[x\backslash t_n]$, then $\forall x A +^{t_1} E_1, \dots, +^{t_n} E_n$ is a dual expansion tree of $\forall x A$ and $Dp(\forall x A +^{t_1} E_1, \dots, +^{t_n} E_n) = \bigwedge_{i=1}^{n} Dp(E_i)$

$+^{\alpha}$ (in 5.) and $+^{t_i}$ (in 6.) are called *strong* and *weak expansions*, respectively. They are typically visualized as labeled arcs connecting the node QxA with the (dual) expansion trees immediately following them. We say that $+^{\alpha}$ *dominates* every expansion in E and $+^{t_i}$ dominates every expansion in E_i. α is referred to as an eigenvariable; we stipulate that each strong expansion in a (dual) expansion tree use a unique eigenvariable.

The concept of expansion trees can be generalized to sequents.

Definition 2.2 (Expansion sequent)**.** Let $\Gamma \vdash \Delta = A_1, \dots, A_m \vdash B_1, \dots, B_n$ be a sequent. A formal sequent $\mathcal{E} \vdash \mathcal{F} = E_1, \dots, E_m \vdash F_1, \dots, F_n$ is called an *expansion sequent* of $\Gamma \vdash \Delta$ if it fulfills the following two conditions:

1. Each E_i is a dual expansion tree of A_i and each F_i is an expansion tree of B_i.
2. No eigenvariable occurs more than once in the whole sequent.

If $\mathcal{E} \vdash \mathcal{F}$ is an expansion sequent, then

$$Dp(\mathcal{E} \vdash \mathcal{F}) := Dp(E_1), \dots, Dp(E_m) \vdash Dp(F_1), \dots, Dp(F_n)$$

is its *deep sequent*.

Definition 2.3 (Dependency relation)**.** Let E be a (dual) expansion tree and $+^s, +^t$ weak expansions in E. The relation $+^s <^0_E +^t$ is defined by

$$+^s <^0_E +^t \text{ if } +^s \text{ dominates a strong expansion } +^{\alpha} \text{ such that } \alpha \text{ occurs in } t.$$

The transitive closure of $<^0_E$ is written as $<_E$ and called the *dependency relation* of E. We will omit the subscript if the meaning is clear from the context. E is called *acyclic* if $<_E$ is acyclic.

If $\mathcal{E} \vdash \mathcal{F}$ is an expansion sequent, the relations $<^0_{\mathcal{E}\vdash\mathcal{F}}$ and $<_{\mathcal{E}\vdash\mathcal{F}}$ are defined in an analogous manner between all weak expansions of (dual) expansion trees in $\mathcal{E} \vdash \mathcal{F}$.

Definition 2.1 requires the terms that are used to expand a weak quantifier to be unique. This fact necessitates some thought on how we define the union of (dual) expansion trees of the same formula. We will first expand the definition of (dual) expansion trees by a new type of node that indicates that the node's child trees need to be merged. Then we will give a reduction procedure describing how these merge nodes can be shifted further and further down until a (dual) expansion tree according to the original definition remains. In order to define this reduction, we will need some terminology: If E and F are expansion trees (either or both might be dual), we write $E[F]$ to indicate that E contains F as a subtree.

Definition 2.4 (Expansion tree with merge)**.** Let A be any formula. A *(dual) expansion tree with merge* is a tree together with an associated deep formula that is defined by the following rules:

1. If E is a (dual) expansion tree of A, then E is a (dual) expansion tree with merge of A.
2. If E, F are expansion trees with merge of A, then so is $E \sqcup F$ and $Dp(E \sqcup F) = Dp(E) \vee Dp(F)$.
   If E,F are dual expansion trees with merge of A, then so is $E \sqcup F$ and $Dp(E \sqcup F) = Dp(E) \wedge Dp(F)$.

Given this definition of expansion trees with merge, expansion sequents with merge can be defined analogously to expansion sequents.

We can apply substitution operators to (dual) expansion trees. This operation is less straightforward than one might imagine, due to the fact that the terms in weak expansions of the same quantifier must be unique.

Definition 2.5 (Substitution for expansion trees)**.** Let α,β be distinct variables and t a term. We define the application of the substitution $[\alpha\backslash t]$ to (dual) expansion trees with merge inductively:

1. $\bot[\alpha\backslash t], \top[\alpha\backslash t], A[\alpha\backslash t]$ are just substitutions of formulas.
2. $(E \diamond F)[\alpha\backslash t] = E[\alpha\backslash t] \diamond F[\alpha\backslash t]$ and $(\neg E)[\alpha\backslash t] = \neg(E[\alpha\backslash t])$.
3. $(QxA +^{\alpha} E)[\alpha\backslash t]$ is not defined if t is not a variable; otherwise it is equal to $QxA +^{t} E[\alpha\backslash t]$.
4. $(QxA +^{\beta} E)[\alpha\backslash t] = QxA +^{\beta} E[\alpha\backslash t]$.
5. Let $s_1, \ldots, s_n$ be distinct terms and $J_1, \ldots, J_m$ the equivalence classes of the relation $i \sim j$ iff $s_i[\alpha\backslash t] = s_j[\alpha\backslash t]$ for $i,j \in \{1, \ldots, n\}$ and let $l_1, \ldots, l_m \in \{1, \ldots, n\}$ such that $l_i \in J_i$. Then

$$(QxA +^{s_1} E_1 \ldots +^{s_n} E_n)[\alpha\backslash t] = QxA +^{s_{l_1}[\alpha\backslash t]} (\bigsqcup_{j \in J_1} E_j[\alpha\backslash t]) \ldots \\ \ldots +^{s_{l_m}[\alpha\backslash t]} (\bigsqcup_{j \in J_m} E_j[\alpha\backslash t]).$$

Note that due to 3., $E[\alpha\backslash t]$ is not defined if E contains the strong expansion $+^{\alpha}$ and t is not a variable.

Definition 2.6 (Merge reduction)**.** We define a reduction relation $\overset{\sqcup}{\mapsto}$ between expansion sequents with merge. Note that in each case except strong expansion, only the expansion tree that is being reduced needs to be taken into consideration, as the rest of the sequent is unaffected. For this reason, we formulate these cases for expansion trees instead of sequents:

1. $E[F \sqcup \bot] \overset{\sqcup}{\mapsto} E[F]$ and $E[\bot \sqcup F] \overset{\sqcup}{\mapsto} E[F]$.
   Likewise, $E[F \sqcup \top] \overset{\sqcup}{\mapsto} E[F]$ and $E[\top \sqcup F] \overset{\sqcup}{\mapsto} E[F]$.
2. If A is atomic, then $E[A \sqcup A] \overset{\sqcup}{\mapsto} E[A]$.

3. $E[(\neg E_1) \sqcup (\neg E_2)] \overset{\sqcup}{\mapsto} E[\neg(E_1 \sqcup E_2)]$.
4. If $\diamond \in \{\vee,\wedge\}$, then $E[(E_1 \diamond E_2) \sqcup (F_1 \diamond F_2)] \overset{\sqcup}{\mapsto} E[(E_1 \sqcup F_1) \diamond (E_2 \sqcup F_2)]$.
5. $(\mathcal{E} \vdash \mathcal{F})[(QxA +^{\alpha} E_1) \sqcup (QxA +^{\beta} E_2)] \overset{\sqcup}{\mapsto} (\mathcal{E} \vdash \mathcal{F})[QxA +^{\alpha} (E_1 \sqcup E_2)][\beta\backslash\alpha]$.
6. If $r_1,\dots,r_l,s_1,\dots,s_m,t_1,\dots,t_n$ are terms such that $\{s_1,\dots,s_m\} \cap \{t_1,\dots,t_n\} = \emptyset$ and

$$E_1 = QxA +^{r_1} E_{1,1} \dots +^{r_l} E_{1,l} +^{s_1} F_1 \dots +^{s_m} F_m,$$
$$E_2 = QxA +^{r_1} E_{2,1} \dots +^{r_l} E_{2,l} +^{t_1} G_1 \dots +^{t_n} G_n,$$

then

$$E[E_1 \sqcup E_2] \overset{\sqcup}{\mapsto} E[QxA +^{r_1} (E_{1,1} \sqcup E_{2,1}) \dots +^{r_l} (E_{1,l} \sqcup E_{2,l}) +^{s_1} F_1 \dots +^{s_m} F_m +^{t_1} G_1 \dots +^{t_n} G_n]$$

We write $\overset{\sqcup}{\to}$ for the reflexive and transitive closure of $\overset{\sqcup}{\mapsto}$.

Theorem 2.7.

1. *For acyclic (dual) expansion sequents with merge, the reduction $\overset{\sqcup}{\to}$ is strongly normalizing and confluent. Its normal forms are proper (dual) expansion sequents, i.e. they do not contain any $\sqcup$-nodes.*
2. *Let E, F be acyclic expansion trees with merge. If $E \sqcup F \overset{\sqcup}{\to} G$, then there are variables $\alpha_1,\dots,\alpha_n,\beta_1,\dots,\beta_n$ such that*

$$Dp(G) \leftrightarrow (Dp(E) \vee Dp(F))[\beta_1\backslash\alpha_1,\dots,\beta_n\backslash\alpha_n].$$

If E and F are dual expansion trees with merge, then

$$Dp(G) \leftrightarrow (Dp(E) \wedge Dp(F))[\beta_1\backslash\alpha_1,\dots,\beta_n\backslash\alpha_n].$$

Proof.

1. If m is a merge node in any (dual) expansion tree E in $\mathcal{E} \vdash \mathcal{F}$, then let $|m|$ be the number of nodes below m. By extension we define the weight $|\mathcal{E} \vdash \mathcal{F}|$ of $\mathcal{E} \vdash \mathcal{F}$ to be the sum of $|m|$ over all merge nodes anywhere in the sequent. Now observe that all reduction steps except 5. reduce $|\mathcal{E} \vdash \mathcal{F}|$ by shifting merge nodes downward. Step 5. may introduce new merge nodes anywhere in the sequent, thereby increasing the weight, but the resulting sequent has one eigenvariable fewer. It follows that if $\mathcal{E} \vdash \mathcal{F} \overset{\sqcup}{\mapsto} \mathcal{E}' \vdash \mathcal{F}'$, then $\mathcal{E}' \vdash \mathcal{F}' <_{lex} \mathcal{E} \vdash \mathcal{F}$, where $<_{lex}$ is the order on expansion sequents induced by the lexicographic order on $(|EV(\mathcal{E} \vdash \mathcal{F})|, |\mathcal{E} \vdash \mathcal{F}|)$. Since this order is clearly well-founded, it follows that every reduction sequence terminates.
2. We show that each of the reduction steps in Definition 2.6 preserves the equivalence. The cases of $E = \bot$ or E atomic are trivial. If $E = \neg E'$, $F = \neg F'$ and $E \sqcup F \overset{\sqcup}{\mapsto} G$,

then

$$Dp(G) = Dp(\neg(E' \sqcup F')) = \neg Dp(E' \sqcup F') = \neg(Dp(E') \wedge Dp(F'))$$
$$= \neg Dp(E') \vee \neg Dp(F') = Dp(E) \vee Dp(F).$$

The other propositional cases are proved similarly. Now, let $E = \forall x A +^{\alpha} E'$, $F = \forall x A +^{\beta} F'$ and $E \sqcup F \overset{\sqcup}{\mapsto} G$. It follows that $Dp(G) = Dp(E' \sqcup F') = (Dp(E') \vee Dp(F'))[\beta \backslash \alpha]$.
The only case left to deal with is that of weak expansions. Let E_1, E_2 as in Definition 2.6, 6., and let $E_1 \sqcup E_2 \overset{\sqcup}{\mapsto} E_3$. Then

$$\begin{aligned} Dp(E_3) = {} & Dp(E_{1,1} \sqcup E_{2,1}) \vee \ldots \vee Dp(E_{1,l} \sqcup E_{2,l}) \\ & \vee Dp(F_1) \vee \ldots \vee Dp(F_m) \vee Dp(G_1) \vee \ldots \vee Dp(G_n) \\ & \leftrightarrow Dp(E_{1,1}) \vee \ldots \vee Dp(E_{1,l}) \vee Dp(F_1) \vee \ldots \vee Dp(F_m) \\ & \vee Dp(E_{2,1}) \vee \ldots \vee Dp(E_{2,l}) \vee Dp(G_1) \vee \ldots \vee Dp(G_n) \\ & \leftrightarrow Dp(E_1) \vee Dp(E_2). \end{aligned}$$

□

Definition 2.8 (Union of expansion trees)**.** By Theorem 2.7, if $(\mathcal{E} \vdash \mathcal{F})[E_1 \sqcup E_2]$ is an acyclic expansion sequent with merge, then it has a unique $\overset{\sqcup}{\to}$-normal form. We define the *union* $(\mathcal{E} \vdash \mathcal{F})[E_1 \cup E_2]$ to be this normal form.

Definition 2.9 (Expansion proof)**.** Let $\Gamma \vdash \Delta$ be a sequent and $\mathcal{E} \vdash \mathcal{F}$ an expansion sequent of $\Gamma \vdash \Delta$. $\mathcal{E} \vdash \mathcal{F}$ is called an *expansion proof* if $<$ is acyclic and $Dp(\mathcal{E} \vdash \mathcal{F})$ is tautological.

We shall now define a method for extracting an expansion sequent from a cut-free proof. This expansion sequent will actually turn out to be an expansion proof.

Definition 2.10 (Expansion proof extraction)**.** Let π be a proof of some sequent. Then we can construct an expansion sequent $Ex(\pi)$ of the same sequent by the following inductive procedure:

1. If π is a one-line proof of $A \vdash A$, then we let $Ex(\pi) = A \vdash A$.
2. If the bottommost inference of π is a contraction on the left, i.e.

$$\pi = \cfrac{\begin{matrix}(\psi)\\ A,A,\Gamma \vdash \Delta\end{matrix}}{A,\Gamma \vdash \Delta}\ c_l$$

and $Ex(\psi) = E_1, E_1', E_2, \ldots, E_m \vdash F_1, \ldots, F_n$, then

$$Ex(\pi) = E_1 \cup E_1', E_2, \ldots, E_m \vdash F_1, \ldots, F_n.$$

Contractions on the right are treated symmetrically (also using $\cup$).

3. If π ends with a weakening on the left, i.e.

$$\pi = \cfrac{\begin{matrix}(\psi)\\ \Gamma \vdash \Delta\end{matrix}}{A,\Gamma \vdash \Delta}\; w_l$$

and $Ex(\psi) = E_1,E_2,\ldots,E_m \vdash F_1,\ldots,F_n$, then

$$Ex(\pi) := \top, E_1,E_2,\ldots,E_m \vdash F_1,\ldots,F_n,$$

and analogously for a weakening on the right, using *bot*.

4. Suppose that

$$\pi = \cfrac{\begin{matrix}(\psi)\\ \Gamma \vdash \Delta,A\end{matrix}}{\neg A, \Gamma \vdash \Delta}\; \neg_l$$

and let $Ex(\psi) = E_1,\ldots,E_m \vdash F_1,\ldots,F_n$. Then

$$Ex(\pi) = \neg F_n, E_1,\ldots,E_m \vdash F_1,\ldots,F_{n-1}.$$

$\neg_r$-inferences are treated analogously.

5. If π ends with a unary $\wedge$-inference, i.e.

$$\pi = \cfrac{\begin{matrix}(\psi)\\ A,B,\Gamma \vdash \Delta\end{matrix}}{A \wedge B, \Gamma \vdash \Delta}\; \wedge_l$$

and $Ex(\psi) = E_1,E_2,\ldots,E_m \vdash F_1,\ldots,F_n$, then

$$Ex(\pi) = E_1 \wedge E_2, E_3,\ldots,E_m \vdash F_1,\ldots,F_n.$$

Unary $\vee$-inferences are treated in the same way.

6. Suppose that

$$\pi = \cfrac{\begin{matrix}(\psi_1)\\ A, \Gamma \vdash \Delta\end{matrix} \quad \begin{matrix}(\psi_2)\\ B, \Pi \vdash \Lambda\end{matrix}}{A \vee B, \Gamma,\Pi \vdash \Delta,\Lambda}\; \vee_l$$

and

$$Ex(\psi_1) = E_1,\ldots,E_m \vdash F_1,\ldots,F_n,$$
$$Ex(\psi_2) = G_1,\ldots,G_k \vdash H_1,\ldots,H_l$$

Then

$$Ex(\pi) := E_1 \vee G_1, E_2,\ldots,E_m,G_2,\ldots,G_k \vdash F_1,\ldots,F_n,H_1,\ldots,H_l.$$

The case of a binary $\wedge$-inference is treated analogously.

7. If π ends with a strong quantifier inference, say

$$\pi = \dfrac{\begin{matrix}(\psi)\\ A[x\backslash\alpha],\Gamma \vdash \Delta\end{matrix}}{\exists x A,\Gamma \vdash \Delta}\ \exists_l$$

and $Ex(\psi) = E_1,\ldots,E_m \vdash F_1,\ldots,F_n$, then

$$Ex(\pi) := \exists x A +^{\alpha} E_1, E_2,\ldots,E_m \vdash F_1,\ldots,F_n.$$

8. If π ends with a weak quantifier inference, say

$$\pi = \dfrac{\begin{matrix}(\psi)\\ \Gamma \vdash \Delta, A[x\backslash t]\end{matrix}}{\Gamma \vdash \Delta, \exists x A}\ \exists_r$$

and $Ex(\psi) = E_1,\ldots,E_m \vdash F_1,\ldots,F_n$, then

$$Ex(\pi) := E_1, E_2,\ldots,E_m \vdash F_1,\ldots,F_{n-1},\exists x A +^{t} F_n.$$

9. If π ends with a cut, as in

$$\pi = \dfrac{\begin{matrix}(\psi_1)\\ \Gamma \vdash \Delta, A\end{matrix} \quad \begin{matrix}(\psi_2)\\ A, \Pi \vdash \Lambda\end{matrix}}{\Gamma,\Pi \vdash \Delta,\Lambda}\ cut$$

and

$$Ex(\psi_1) = E_1,\ldots,E_m \vdash F_1,\ldots,F_n,$$
$$Ex(\psi_2) = G_1,\ldots,G_k \vdash H_1,\ldots,H_l,$$

then

$$Ex(\pi) := E_1,\ldots,E_m,G_2,\ldots,G_k \vdash F_1,\ldots F_{n-1},H_1,\ldots,H_l.$$

Theorem 2.11. *If π is a proof of $\Gamma \vdash \Delta$ in which all cuts are either unquantified or have their cut formula introduced by a weakening, then $Ex(\pi)$ is an expansion proof of $\Gamma \vdash \Delta$.*

Proof. Clearly, $Ex(\pi)$ is an expansion sequent of $\Gamma \vdash \Delta$. We need to show that $Dp(Ex(\pi))$ is a tautology and that $<_{Ex(\pi)}$ is acyclic. For acyclicity, observe that if $+^t <^0_{Ex(\pi)} +^s$, then there is a strong expansion $+^\alpha$ dominated by $+^t$ such that s contains α. But this is only possible if the weak quantifier inference that gave rise to $+^s$ is above the strong inference that produced $+^\alpha$, because π is regular. It follows that if $+^t <_{Ex(\pi)} +^s$, the inference corresponding to s is above that corresponding to t. Thus, a cycle would imply that two inferences are each above the other, which is impossible.

For the validity of $Dp(Ex(\pi))$, we use induction on the length of π to show that each rule in Definition 2.10 preserves the validity of $Dp(Ex(\pi))$.

1. If π is a one-line proof of $A \vdash A$, then $Dp(Ex(\pi)) = A \vdash A$ is tautological.
2. By Theorem 2.7,

$$\begin{aligned}
&Dp(E_1 \cup E_1', E_2, \ldots, E_m \vdash F_1, \ldots, F_n) \leftrightarrow \\
&\leftrightarrow Dp(E_1 \cup E_1'), Dp(E_2), \ldots, Dp(E_m) \vdash Dp(F_1), \ldots, Dp(F_n) \leftrightarrow \\
&\leftrightarrow Dp(E_1) \wedge Dp(E_1')[\beta_1 \backslash \alpha_1, \ldots, \beta_n \backslash \alpha_n], Dp(E_2), \ldots, Dp(E_m) \vdash Dp(F_1), \ldots, Dp(F_n) \leftrightarrow \\
&\leftrightarrow Dp(E_1[\beta_1 \backslash \alpha_1, \ldots, \beta_n \backslash \alpha_n], E_1'[\beta_1 \backslash \alpha_1, \ldots, \beta_n \backslash \alpha_n], E_2, \ldots, E_m \vdash F_1, \ldots, F_n)
\end{aligned}$$

   for some α_i,β_i. Note that applying a variable substitution to a quantifier-free formula preserves validity and consequently, so does the contraction rule.
3. $Ex(\pi)$ arises from $Ex(\psi)$ by adding the dual expansion tree $\top$ to the antecedent. Since $Dp(Ex(\pi)) = \top, Dp(Ex(\psi)) \leftrightarrow Dp(Ex(\psi))$, $Dp(Ex(\pi))$ is a tautology.
4. This step transforms a deep sequent of the form $\Pi \vdash \Lambda, B$ to the logically equivalent $\neg B, \Pi \vdash \Lambda$.
5.

$$\begin{aligned}
Dp(Ex(\pi)) &= Dp(E_1 \wedge E_2, E_3, \ldots, E_m \vdash F_1, \ldots, F_n) = \\
&= Dp(E_1) \wedge Dp(E_2), Dp(E_3), \ldots, Dp(E_m) \vdash Dp(F_1), \ldots, Dp(F_n) \\
&\leftrightarrow Dp(E_1), Dp(E_2), Dp(E_3), \ldots, Dp(E_m) \vdash Dp(F_1), \ldots, Dp(F_n) = \\
&= Dp(Ex(\psi))
\end{aligned}$$

   It follows that if $Dp(Ex(\psi))$ is a tautology, then so is $Dp(Ex(\pi))$.
6. First, note that

$$\begin{aligned}
Dp(Ex(\pi)) &= Dp(E_1) \vee Dp(G_1), Dp(E_2), \ldots, Dp(E_m), Dp(G_2), \ldots, Dp(G_k) \vdash \\
&\vdash Dp(F_1), \ldots, Dp(F_n), Dp(H_1), \ldots, Dp(H_l).
\end{aligned}$$

   Now assume that $Dp(Ex(\psi_1))$ and $Dp(Ex(\psi_2))$ are tautologies and let $\mathcal{I}$ be any interpretation of $Dp(Ex(\pi))$. If the whole antecedent of either $Dp(Ex(\psi_1))$ or $Dp(Ex(\psi_2))$ is true under $\mathcal{I}$, then so is one of the F_i or one of the H_i, respectively, and hence $Dp(Ex(\pi))$ is true under $\mathcal{I}$. If neither antecedent is true under $\mathcal{I}$, there are two possibilites. First, if both $Dp(E_1)$ and $Dp(G_1)$ are false under $\mathcal{I}$, then so is $Dp(Ex(E_1)) \vee Dp(Ex(G_1))$. If either of them is true, then one of the E_i or the G_i for $i \geq 2$ must be false. In both cases, $Dp(Ex(\pi))$ is true under $\mathcal{I}$. Since $\mathcal{I}$ was arbitrary, $Dp(Ex(\pi))$ is a tautology.
7. Since $Dp(\exists x A +^{\alpha} E_1) = Dp(E_1)$, the deep sequent is not changed by this rule.
8. Again, the deep sequent does not change in this step.
9. If A is quantifier-free, then each expansion tree of A is also dual and vice versa. Thus, if F_n and G_1 are (dual) expansion trees of A, then there is a (dual) expansion

tree M of A such that $Dp(M)$ implies both $Dp(F_n)$ and $Dp(G_1)$. Consequently, we may assume that $F_n = G_1$. Now it is straightforward to prove that if $Dp(Ex(\psi_1))$ and $Dp(Ex(\psi_2))$ are tautological, so is $Dp(Ex(\pi))$.

If A is introduced by a weakening on the left side of the cut, then $F_n = \bot$. It follows that $Dp(Ex(\psi_1)) \leftrightarrow Dp(E_1, \ldots, E_m \vdash F_1, \ldots, F_{n-1})$. But if the latter formula is valid, then clearly so is $Dp(Ex(\pi))$. Note that in this case we do not actually use any information about $Ex(\psi_2)$, which is reminiscent of the weakening case of cut reduction. The case where A is introduced on the right-hand side of the cut is treated analogously.

□

Definition 2.12 (Deep set)**.** Let E be a (dual) expansion tree of A. The *deep set* $Dp^*(E)$ is defined as follows:

- If E contains no weak expansions, then $Dp^*(E) = \{Dp(E)\}$.
- $Dp^*(\neg E) = \neg Dp^*(E)$.
- If $E = E_1 \diamond E_2$, then $Dp^*(E) = Dp^*(E_1) \diamond Dp^*(E_2)$.
- If $E = QxA +^{t_1} E_1 \ldots +^{t_n} E_n$, then $Dp^*(E) = \bigcup_{i=1}^n Dp^*(E_i)$.
- $Dp^*(QxA +^{\alpha} E) = Dp^*(E)$.

If $\mathcal{E} \vdash \mathcal{F}$ is an expansion sequent, then $Dp^*(\mathcal{E} \vdash \mathcal{F}) := \bigcup_{E \in \mathcal{E}} \neg Dp^*(E) \cup \bigcup_{F \in \mathcal{F}} Dp^*(F)$.

Lemma 2.13. *Let $\mathcal{E} \vdash \mathcal{F}$ be an expansion sequent. Then $Dp(\mathcal{E} \vdash \mathcal{F}) \leftrightarrow \bigvee Dp^*(\mathcal{E} \vdash \mathcal{F})$.*

Proof. First of all, it is clear that $Dp(\mathcal{E} \vdash \mathcal{F}) \leftrightarrow \bigvee_{E \in \mathcal{E}} \neg Dp(E) \vee \bigvee_{F \in \mathcal{F}} Dp(F)$. As a consequence, we only need to show that $Dp(E) \leftrightarrow \bigvee Dp^*(E)$ for any expansion tree E and $Dp(E) \leftrightarrow \bigwedge Dp^*(E)$ for any dual expansion tree E. We prove this for expansion trees by induction on the structure of E:

- The case where E contains no weak expansions is trivial.
- If E is an expansion tree, $Dp(\neg E) = \neg Dp(E) \leftrightarrow \neg \bigvee Dp^*(E) \leftrightarrow \bigwedge \neg Dp^*(E) = \bigwedge Dp^*(\neg E)$. The case where E is a dual expansion tree is dealt with analogously.
- If E_1, E_2 are expansion trees, then $Dp(E_1 \diamond E_2) = Dp(E_1) \diamond Dp(E_2) \leftrightarrow (\bigvee Dp^*(E_1)) \diamond (\bigvee Dp^*(E_2)) \leftrightarrow \bigvee (Dp^*(E_1) \diamond Dp^*(E_2))$. The case of dual expansion trees is treated in the same way.
- If $E_1, \ldots, E_n$ are expansion trees, then $Dp(QxA +^{t_1} E_1 \ldots +^{t_n} E_n) = \bigvee_{i=1}^n Dp(E_i) \leftrightarrow \bigvee_{i=1}^n \bigvee Dp^*(E_i) = \bigvee \bigcup_{i=1}^n Dp^*(E_i)$.
- Straightforward.

□

Definition 2.14 (Herbrand set)**.** Let π be a proof of the form described in Theorem 2.11. Then $\mathcal{H}(\pi) := Dp^*(Ex(\pi))$ is called the *Herbrand set* of π.

Corollary 2.15. *If π is a proof of the form described in Theorem 2.11, then $\mathcal{H}(\pi)$ is a tautological set, i.e. $\bigvee \mathcal{H}(\pi)$ is a tautology.*

Proof. By combining Theorem 2.11 and Lemma 2.13. □

CHAPTER 3

Grammars

In this chapter we will develop the theory of regular tree grammars as applicable to this thesis. We will need the notion of a *ranked alphabet*, i.e. a set Σ of symbols together with their respective arities. We write $\mathfrak{T}_\Sigma$ for the set of terms that can be constructed from Σ and $\mathfrak{T}_\Sigma(X)$ for the set of terms that can be constructed from Σ together with a set X of variables. For any symbol x and term t, we write $x \in t$ to express that x occurs in t.

Definition 3.1 (Regular tree grammar)**.** A *regular tree grammar* is a tuple $G = \langle \varphi, N, \Sigma, P \rangle$, where

1. Σ is a finite ranked alphabet; its elements are called *terminal symbols* (or *terminals* for short);
2. N is a finite set, disjoint from Σ; its elements are called *nonterminals*;
3. $\varphi \in N$ is the *starting symbol*;
4. $P \subseteq N \times \mathfrak{T}_\Sigma(N)$, is the finite set of *production rules* (*productions*). Its elements are usually written as $\alpha \to t$ instead of (α, t).

A production of the form $\alpha \to t$ is said to *begin* with α. The set of productions beginning with α will be denoted by P_α. If $P_\alpha = \{\alpha \to t_1, \dots, \alpha \to t_n\}$, we can concisely denote these productions by $\alpha \to t_1 | \dots | t_n$.

Remark 3.2. We can and frequently will apply the substitution operator defined in Definition 1.2 to productions; the substitution is always understood as being applied to the second term, i.e. $\alpha \to t[\beta \backslash r] = \alpha \to (t[\beta \backslash r])$. If Q is any set of productions, $Q[\beta \backslash r]$ is defined as $\{p[\beta \backslash r] \mid p \in Q\}$.

Now we shall define the *derivability relation* of a grammar G. Let $r, s \in \mathfrak{T}_\Sigma(N)$. We say that s is derivable from r—written as $r \to_G s$—if there is a production $\beta \to t \in P$ such that s can be obtained by replacing one occurence of β in r by t. The reflexive and transitive closure of this relation will be denoted by $r \to_G^* s$. If the grammar is clear from the context, the subscript will be omitted.

It is clear from the above that every step in a derivation can be viewed as an application of a specific production rule to one term in order to receive a new term; we shall occasionally write $r \xrightarrow{p} s$ to express that s arose from r through application of the rule p. Consequently, the notion of a production rule that is used in a derivation is well-defined.

Definition 3.3 (Language of a grammar)**.** Let G be a tree grammar of any type. Since every class of grammar carries with it a notion of what a valid derivation is, the *language* of G, denoted by $L(G)$, is always definable as the set of all $t \in \mathfrak{T}_\Sigma$ such that $\varphi \to_G^* t$.

In the sequel we shall need types of grammars that arise from regular tree grammars by restricting the productions—or, more precisely, the combinations of productions—that can be used in derivations. The first of these types is that of *rigid grammars*. The notion of rigidity was first introduced in the form of automata in [Jac11].

Definition 3.4 (Rigid tree grammar)**.** A *rigid tree grammar* is a tuple $G = \langle \varphi, N, R, \Sigma, P \rangle$ such that $\langle \varphi, N, \Sigma, P \rangle$ is a regular tree grammar (called the *underlying* regular grammar of G) and $R \subseteq N$ is the set of *rigid nonterminals*. In the case of $R = N$, G is called *totally rigid*; when talking about totally rigid grammars, we shall leave the set R out of the definition.

The important difference between rigid and nonrigid grammars lies in the derivation relation. A derivation $\varphi \to t_1 \to \ldots \to t_n \to t$ of the underlying regular grammar is a valid derivation of G if the following *rigidity condition* holds: If β is a rigid nonterminal and q,q' are positions such that $t_i|_q = t_j|_{q'} = \beta$ for some i,j, then $t|_q = t|_{q'}$. This condition ensures that if an occurence of a rigid nonterminal is replaced by a certain term in the end product, all other occurences of the same nonterminal are replaced by the same term there.

Lemma 3.5.

1. *If G is a rigid grammar and $t \in L(G)$, then there is a derivation of t that uses at most one production beginning with each rigid nonterminal.*
2. *If G is a grammar and every term $t \in L(G)$ can be derived using at most one production beginning with each nonterminal, then G can be interpreted as a totally rigid grammar.*

Proof.

1. Suppose that $\varphi \to t_1 \to \ldots \to t_n \to t$ is a derivation of G and $\beta \in R$ such that two different productions $\beta \to r, \beta \to r'$ are used in steps $t_i \to t_{i+1}$ and $t_j \to t_{j+1}$ respectively. We know that in the end term t, every occurrence of β has been replaced by the same term s, so obviously we can replace either of the subderivations $\beta \to r \to^* s$, $\beta \to r' \to^* s$ with the other and end up with a valid derivation of t.
2. If a derivation of a term t only includes one rule for each terminal, then it clearly satisfies the rigidity condition for all terminals.

□

Example 3.6. Let $\Sigma = \{f/2, g/1, a/0\}$ and $L = \{f(g^n(a),g^n(a)) \mid n \geq 0\}$. Using the pumping lemma for tree grammars found, for instance, in [Com07], it is easy to show that L is not generated by a regular tree grammar. On the other hand, the following rigid grammar generates L:

- $N = \{\varphi,\alpha,\beta\}$
- $R = \{\alpha\}$
- Productions:
  - $\varphi \to f(\alpha,\alpha)$
  - $\alpha \to \beta$
  - $\beta \to a|g(\beta)$.

In a rigid grammar, the choice we make for one nonterminal has no influence on what productions we can use for the others. It turns out that it is possible to define a class of grammars that allows us to enforce any relationship between nonterminals (or their productions, respectively) that we might want. We call this the class of *constrained grammars*.

Definition 3.7 (Constrained tree grammar)**.** A *constrained tree grammar* is a tuple $G = \langle\varphi,N,\Sigma,P,\mathcal{C}\rangle$ consisting of a regular tree grammar $\langle\varphi,N,\Sigma,P\rangle$ together with a *constraint formula* $\mathcal{C}$, which is a propositional formula that uses productions as atoms.

As in the case of rigid grammars, the salient point of this definition lies in the derivation relation. Every derivation d of the underlying regular tree grammar naturally induces a partial truth assignment v_d that assigns $\top$ to all productions used in d and leaves the other atoms unassigned. We define the language of G to be the set of all terms $t \in \mathfrak{T}_\Sigma(N)$ such that there is a derivation d ending with t and v_d is a maximal interpretation of $\mathcal{C}$, i.e. if d' is the extension of v by a production that is not in v, then $v_{d'}(\mathcal{C})$ is a contradiction. Note that by this definition, the language of a constrained grammar may contain nonterminals.

By Lemma 3.5, totally rigid grammars are characterized by the fact that derivations may use at most one production for each nonterminal. It follows that if $G = \langle\varphi, N, \Sigma, P\rangle$ is a totally rigid grammar, then the constrained grammar $G' = \langle\varphi, N, \Sigma, P, TR_P\rangle$, where

$$TR_P := \bigwedge_{\alpha\in N} \bigwedge_{\substack{p,q\in P_\alpha \\ p\neq q}} \neg(p \wedge q),$$

generates the same language as G. By extension, if $G = \langle\varphi, N, \Sigma, P, \mathcal{C}\rangle$ is a constrained grammar, then $G' = \langle\varphi, N, \Sigma, P, \mathcal{C} \wedge TR_P\rangle$ is a totally rigid constrained grammar that enforces the same constraints as G.

From time to time we will need to substitute both sides of a production rule at the same time. To this end, we define a variant substitution operator.

Definition 3.8 (Variant substitution operator)**.** If $\alpha \to t$ is a production rule of a grammar, then $\alpha \to t\{\beta\backslash r\}$ is defined as the production that results from replacing all occurences of β in both α and t with r. Concatenation of this operator and its application to a set of productions are defined as in Definition 1.2 and Remark 3.2, respectively.

Both substitution operators can be applied in a natural way to constraint formulas; indeed, this is the primary use of the variant operator.

We shall now turn our attention to defining a certain relation on grammars that will in practice turn out to be a partial order.

Definition 3.9 (Dependency relation)**.** Let G be a tree grammar (of any kind) and N its set of nonterminals. We define the relation $\prec_G^0$ on N as

$$\alpha \prec_G^0 \beta \text{ iff there is a term } t \text{ such that } \alpha \to t \in P \text{ and } \beta \in t.$$

The transitive closure of this relation is written as $\prec_G$ and called the *dependency relation* of G. As in the case of the derivation relation, the subscript will be omitted if the grammar is clear from the context. The dependency relation is transitive by definition. If it is also acyclic, i.e. there are no $\alpha, \gamma_1, \ldots, \gamma_k \in N, k \geq 0$, such that $\alpha \prec \gamma_1 \prec \ldots \prec \gamma_k \prec \alpha$, it is called the *dependency order* and G is called an *acyclic grammar*.

If M is any nonempty subset of N, then a nonterminal $\alpha \in M$ is said to be *minimal with respect to* M if there is no $\beta \in M$ with $\beta \prec \alpha$.

Lemma 3.10. *Let G be a totally rigid grammar and $\alpha \in N$ minimal with $P_\alpha = \alpha \to t_1|\ldots|t_n$. Define a new totally rigid grammar $G' = \langle \varphi, N', \Sigma', P' \rangle$, where*

1. $N' = N \setminus \{\alpha\}$,
2. $\Sigma' = \Sigma$,
3. $P' = \bigcup_{\beta \in N' \setminus \{\varphi\}} P_\beta \cup \bigcup_{i=1}^{n} P_\varphi[\alpha\backslash t_i]$.

Then $L(G) = L(G')$.

Proof. We need to show that any valid derivation of G can be transformed into a valid derivation of G' and vice versa. Suppose that $\varphi \to r_1 \to \ldots \to r_k \to s$ is a derivation of G that uses only one production for each terminal, cf. Lemma 3.5. If α does not occur in any r_i, it is easy to see that this derivation is just as valid in G'. If, on the other hand, the derivation uses α, then the first step $\varphi \to r_1$ must introduce all the occurences of α since α is minimal. Any two steps where α is replaced use the same production beginning with α, say $p := \alpha \to t_j$. Thus, we can replace the first step by $\varphi \to r_1[\alpha\backslash t_j]$ and remove all steps that use p; the result will be a valid G'-derivation ending in s.
Conversely, assume that $\varphi \to r_1 \to \ldots \to r_k \to s$ is a G'-derivation. Then $r_1 = q_1[\alpha\backslash t_i]$ for some $q_1 \in \mathfrak{T}_\Sigma(N)$ and $1 \leq i \leq n$. There are two cases to consider, according to whether or not q_1 contains α. If it does not, then we already have a G-derivation. If it does, then we replace the first step by $\varphi \to q_1$ (which is a G-production) and immediately afterwards insert as many applications of $\alpha \to t_i$ as we need to arrive at r_1. Again, the result is a valid derivation of G. □

Corollary 3.11. *Let G be totally rigid and acyclic. Then there is an enumeration $\alpha_1,\dots,\alpha_n$ of $N \setminus \{\varphi\}$ such that*

$$L(G) = \{s[\alpha_1\backslash t_1,\dots,\alpha_n\backslash t_n] \mid \varphi \to s, \alpha_i \to t_i \in P \textit{ for } 1 \le i \le n\}\,. \tag{3.1}$$

Proof. By induction on the number n of nonterminals, excluding φ. The case $n = 0$ is trivial.
Now suppose that G has n nonterminals, not counting φ, and that 3.1 holds up to $n - 1$. Since G is acyclic, there is a minimal element $\alpha_1 \in N \setminus \{\varphi\}$. We can apply Lemma 3.10 to G and α_1 to obtain a grammar G' with fewer nonterminals that is again totally rigid and acyclic and satisfies $L(G) = L(G')$. By the induction hypothesis,

$$L(G) = L(G') = \left\{s[\alpha_2\backslash t_2,\dots,\alpha_n\backslash t_n] \,\middle|\, \varphi \to s, \alpha_i \to t_i \in P' \text{ for } 2 \le i \le n\right\};$$

note that $\alpha_i \to t_i \in P'$ is obviously equivalent to $\alpha_i \to t_i \in P$ and $\varphi \to s \in P'$ is equivalent to $s = r[\alpha_1\backslash t_1]$ for some r, t_1 such that $\varphi \to r, \alpha_1 \to t_1 \in P$. Substituting these equivalent conditions in the above equation yields equation (3.1) and concludes the proof. □

Remark. Equation (3.1) shows that totally rigid acyclic grammars—and by extension, those constrained grammars that we are actually going to use—always generate finite languages. Since any finite language can be trivially generated by a tree grammar without resorting to rigidity, these grammars do not give us any additional expressive power. The reason they are useful rather lies in the fact that constrained grammars allow us to represent some languages in a more compact form than general tree grammars would; in particular, the size of a grammar that corresponds to a proof is polynomially bounded by the size of that proof.

CHAPTER 4

Proofs and Grammars

We impose some restrictions on the proofs we consider in this chapter:

1. In order to avoid having to deal with strong expansions in the end-sequent, we only consider proofs of sequents that do not contain any strong quantifiers. This is not a significant restriction, as we can always Skolemize a sequent with strong quantifiers into an equivalid one that only contains weak quantifiers.
2. For every cut c of π, if the cut formula of c contains quantifiers, then it is either Π_1 or Σ_1; we call c a *Π_1-cut* or a *Σ_1-cut* accordingly.

A large part of this chapter will be devoted to constructing a constrained grammar $G(\pi) = \langle \varphi, N(\pi), \Sigma, P(\pi), \mathcal{C}(\pi) \rangle$. The nonterminals of $G(\pi)$ will belong to two distinct types: those given by the end sequent and those arising from quantified cuts in π.

We will denote the set of quantified cut inferences of π by $\mathrm{QCuts}(\pi)$. Let $c \in \mathrm{QCuts}(\pi)$, A_c be the cut formula of c and π_1, π_2 be the left and right subproofs above c respectively, then exactly one of π_1 and π_2 uses strong quantifier inferences to introduce the quantifiers in A. We call this subproof the *strong side* and the other one the *weak side* of c. The strong side is on the left for Π_1-cuts and on the right for Σ_1-cuts.

The relation defined in the following paragraph bears a close relationship to the dependency relation defined in 3.9, cf. Lemma 4.12.

Definition 4.1 (Scope relation)**.** Let π be a simple proof. We define a relation $<^0_\pi$ on $\mathrm{QCuts}(\pi)$ by

$$c <^0_\pi c' \text{ if the weak side of } c \text{ and the strong side of } c' \text{ have nonempty intersection.}$$

The transitive closure of this relation is written as $<_\pi$ and called the *scope relation* of π. We will omit the subscript if the proof is clear from the context.

The one-step scope relation $<^0$ of a proof π can easily be determined from the types of cuts and their spatial relationship in the proof.

Lemma 4.2. *Let c,$c' \in \mathrm{QCuts}(\pi)$.*

1. *$c \not<^0 c$, i.e. $<^0$ is irreflexive.*
2. *If c and c' are not on a common branch in π, then neither $c <^0 c'$ nor $c' <^0 c$ hold.*
3. *If c and c' are on a common branch, then there are three cases, depending on their respective types:*
   - *c and c' are both Σ_1: w.l.o.g. c' is above c; if c' is to the right, then $c' <^0 c$, otherwise $c <^0 c'$ (i.e. the cut "further right" is smaller).*
   - *c and c' are both Π_1: The converse of the previous case holds, i.e. the cut "further left" is smaller.*
   - *c is Π_1, c' is Σ_1: If either of them is above and to the right of the other, $c <^0 c'$, otherwise $c' <^0 c$.*

Proof.

1. It is clear that the strong and weak sides of a cut do not intersect.
2. If c and c' are not on a common branch, then their respective strong and weak subproofs obviously cannot intersect.
3. All three cases are easily verified by using the fact that the strong side of Π_1-cuts is left and the strong side of Σ_1-cuts is right.

□

Lemma 4.3. *$<_\pi$ is acyclic.*

Proof. By induction on the cardinality of QCuts(π). If QCuts(π) $= \emptyset$, then $<_\pi$ is trivially acyclic. Now let ι be the lowest inference in π such that either ι is itself a cut or both its subproofs π_1 and π_2 contain cuts. In either case, both π_1 and π_2 contain fewer cuts than π and consequently their scope relations are acyclic. In the second case, $<_\pi = <_{\pi_1} \cup <_{\pi_2}$ and hence $<_\pi$ is clearly acyclic. In the first case, assume there is a cycle $c_1 <^0 \ldots <^0 c_n <^0 c_1$ in QCuts(π). Since both $<_{\pi_1}$ and $<_{\pi_2}$ are acyclic by the induction hypothesis and $<^0$ is irreflexive, there must be i,j such that $\iota <^0 c_i$ and $\iota <^0 c_j$ and $c_i \in$ QCuts(π_1); $c_j \in$ QCuts(π_2) (i.e. π_1 and π_2 must each contain a cut larger than ι). Now, we can apply Lemma 4.2 to see that either possibility for the type of ι leads to a contradiction. Thus, $<_\pi$ is acyclic. □

Lemma 4.3 implies that $<_\pi$ is in fact a strict partial order; we will henceforth call it the *scope order* of π.

Before we proceed to constructing the grammar of a proof, we need to fix some terminology. Let π be a proof and $c \in$ QCuts(π) with cut formula A_c. As we have noted before, introducing the quantifiers of A_c requires strong inferences on one side of c (the "strong side"). The eigenvariables of these inferences will be called the eigenvariables of c and denoted by $EV(c)$. Due to contractions, each quantifier might be introduced by several inferences and because of regularity, the eigenvariables of these inferences are all distinct.

Moving on, on the weak side of c each quantifier is introduced one or more times by inferences of the form

$$\frac{\Gamma \vdash \Delta, A[x\backslash t]}{\Gamma \vdash \Delta, \exists x A}\ \exists_r \qquad\qquad \text{or} \frac{A[x\backslash t], \Gamma \vdash \Delta}{\forall x A, \Gamma \vdash \Delta}\ \forall_l,$$

according to the type of c. If $\alpha \in EV(c)$ is used to introduce $\exists x$ on the strong side of c, then each such term t is said to be *associated* with α. Likewise, if x is a bound variable in the end sequent of π and the weak quantifier Qx is introduced by a term t in π, then we call x and t associated.

We can now work towards defining the grammar of a proof.

Definition 4.4 (Nonterminals of the end sequent)**.** Let π be a proof of $A_1, \ldots, A_m \vdash B_1, \ldots, B_n$ and let $BV(A)$ be the set of bound variables in the formula A. Then

$$N_{ES}(\pi) = \bigcup_{i=1}^{m} BV(A_i) \cup \bigcup_{i=1}^{n} BV(B_i).$$

Definition 4.5 (Nonterminals of cuts)**.** Let π be a proof and $EV(\pi) := \bigcup_{c \in \mathrm{QCuts}(\pi)} EV(c)$. Furthermore, let $\alpha_1, \ldots, \alpha_n$ be an enumeration of $EV(\pi)$ such that if $c <_\pi c'$, then all elements of $EV(c)$ have lower indices than all elements of $EV(c')$. We inductively define sets $N_1(\pi), \ldots, N_n(\pi)$ in the following manner:
Consider the set

$$B(\alpha_1) := \{x \in N_{ES} | x \text{ is associated with a term } t(\alpha_1)\}.$$

Now let $x_{1,1}, \ldots, x_{1,k_1}$ be those elements of $B(\alpha_1)$ whose quantifiers are outermost. Then

$$N_1(\pi) := \{\alpha_1^{x_{1,1}}, \ldots, \alpha_1^{x_{1,k_1}}\},$$

where the $\alpha_1^{x_{1,j}}$ are new symbols.
Now suppose that we have already defined $B(\alpha_1), \ldots, B(\alpha_i)$ and $N_1(\pi), \ldots, N_i(\pi)$. Let

$$B(\alpha_{i+1}) := \{x \in N_{ES} | x \text{ is associated with a term } t(\alpha_{i+1})\} \cup \\ \cup \bigcup \{B(\alpha_j) \mid \alpha_j \text{ is associated with a term } t(\alpha_{i+1}), j \leq i\}$$

and $x_{i+1,1}, \ldots, x_{i+1,k_{i+1}}$ those variables in $B(\alpha_{i+1})$ whose quantifiers are outermost. Then

$$N_{i+1}(\pi) := \{\alpha_{i+1}^{x_{i+1,1}}, \ldots, \alpha_{i+1}^{x_{i+1,k_{i+1}}}\},$$

where the $\alpha_{i+1}^{x_{i+1,j}}$ are new symbols.
Finally, $N_{Cuts}(\pi) := \bigcup_{i=1}^{n} N_i(\pi)$.

Definition 4.6 (Productions of a proof)**.** Let π be a proof and $\alpha_1, \ldots, \alpha_n$ and the $B(\alpha_i)$ as in the previous definition. We define a family $(o_i)_{i \in \{1, \ldots, n\}}$ of functions such that $o_i : B(\alpha_i) \to B(\alpha_i)$ maps each variable $x \in B(\alpha_i)$ to the unique outermost variable

$y \in B(\alpha_i)$ above or equal to x.

1. For $x \in N_{ES}(\pi)$, let

$$P_x(\pi) := \left\{ x \to t\left(\alpha_{i_1}^{o_{i_1}(x)}, \ldots, \alpha_{i_k}^{o_{i_k}(x)}\right) \,\middle|\, t(\alpha_{i_1}, \ldots, \alpha_{i_k}) \text{ is associated with } x \right\}.$$

2. For $\beta^x \in N_{Cuts}(\pi)$, let

$$P_{\beta^x}(\pi) := \left\{ \beta^x \to t\left(\alpha_{i_1}^{o_{i_1}(x)}, \ldots, \alpha_{i_k}^{o_{i_k}(x)}\right) \,\middle|\, t(\alpha_{i_1}, \ldots, \alpha_{i_k}) \text{ is associated with } \beta \right\}.$$

Definition 4.7 (Constraint formula of the end sequent)**.** Let π be a proof of $A_1, \ldots, A_m \vdash B_1, \ldots, B_n$. We construct a constraint formula for the end sequent by traversing the ancestor trees of the A_i and B_i. To this end, let μ be an occurrence of any formula in π.

- If μ is quantifier-free, then

$$\mathcal{C}_{ES}(\mu,\pi) := \top.$$

- If μ is introduced by a weakening, then let $z_1, \ldots, z_k$ be the bound variables in μ and

$$\mathcal{C}_{ES}(\mu,\pi) := \bigwedge_{j=1}^{k} \bigwedge \neg P_{z_j}(\pi).$$

- If μ is introduced by a $\wedge_r$-rule, as in

$$\frac{\Gamma_1 \vdash \Delta_1, A_{[\nu_1]} \quad \Gamma_2 \vdash \Delta_2, B_{[\nu_2]}}{\Gamma \vdash \Delta, (A \wedge B)_{[\mu]}} \wedge_r,$$

  then

$$\mathcal{C}_{ES}(\mu,\pi) := \mathcal{C}_{ES}(\nu_1,\pi) \wedge \mathcal{C}_{ES}(\nu_2,\pi).$$

  The same holds in the case of a $\vee_l$-inference.

- If μ is introduced by a $\wedge_l$-rule, as in

$$\frac{A_{[\nu_1]}, B_{[\nu_2]}, \Gamma \vdash \Delta}{(A \wedge B)_{[\mu]}, \Gamma \vdash \Delta} \wedge_l,$$

  then

$$\mathcal{C}_{ES}(\mu,\pi) := \mathcal{C}_{ES}(\nu_1,\pi) \wedge \mathcal{C}_{ES}(\nu_2,\pi),$$

  and analogously for $\vee_r$.

- If μ arises from a contraction on the right, i.e.

$$\frac{\Gamma \vdash \Delta, A_{[\nu_1]}, A_{[\nu_2]}}{\Gamma \vdash \Delta, A_{[\mu]}}\ c_r,$$

then

$$\mathcal{C}_{ES}(\mu,\pi) := \mathcal{C}_{ES}(\nu_1,\pi) \vee \mathcal{C}_{ES}(\nu_2,\pi),$$

and analogously for a contraction on the left.

- If μ is introduced by a $\neg_r$ rule, as in

$$\frac{\Gamma, A_{[\nu]} \vdash \Delta}{\Gamma \vdash \Delta, (\neg A)_{[\mu]}}\ \neg_r,$$

then

$$\mathcal{C}_{ES}(\mu,\pi) := \mathcal{C}_{ES}(\nu,\pi).$$

The same is true for a $\neg_l$-inference.

- If μ is introduced by a quantifier rule, i.e.

$$\frac{\Gamma \vdash \Delta, (A[x \backslash t(\alpha_{i_1}, \ldots, \alpha_{i_k})])_{[\nu]}}{\Gamma \vdash \Delta, (\exists x A)_{[\mu]}}\ \exists_r,$$

then

$$\mathcal{C}_{ES}(\mu,\pi) := x \to t\left(\alpha_{i_1}^{o_{i_1}(x)}, \ldots, \alpha_{i_k}^{o_{i_k}(x)}\right) \wedge \mathcal{C}_{ES}(\nu,\pi).$$

The case of a $\forall_l$-inference is analogous.

- We skip over all inferences whose active formula is not μ.

Now we let $\mu_1, \ldots, \mu_m, \nu_1, \ldots, \nu_n$ be the occurrences of $A_1, \ldots, A_m, B_1, \ldots B_n$ in the end sequent. Then

$$\mathcal{C}_{ES}(\pi) := \bigwedge_{i=1}^{m} \mathcal{C}_{ES}(\mu_i,\pi) \wedge \bigwedge_{i=1}^{n} \mathcal{C}_{ES}(\nu_i,\pi).$$

4.1 The case of simple end sequents

For the remainder of this section, we only consider proofs whose end sequents are of the form $\vdash \exists x_1 \ldots \exists x_m A \vee \exists y_1 \ldots \exists y_n B$ with A,B quantifier-free. For such a proof π it follows that $N_{ES}(\pi) = \{x_1, \ldots, x_m, y_1, \ldots, y_n\}$ and each $\alpha \in EV(\pi)$ has at most two copies in $N_{Cuts}(\pi)$, one originating from the x_i and one from the y_i. Thus, $N_{Cuts}(\pi) = \{\alpha_{i_1}^x, \ldots, \alpha_{i_k}^x\} \cup \{\alpha_{j_1}^y, \ldots, \alpha_{j_l}^y\}$.

If t is any term, then let

$$t^x := t[\alpha_{i_1}\backslash\alpha^x_{i_1},\ldots,\alpha_{i_k}\backslash\alpha^x_{i_k}],$$
$$t^y := t[\alpha_{j_1}\backslash\alpha^y_{j_1},\ldots,\alpha_{j_l}\backslash\alpha^y_{j_l}]$$

The productions of π are now easy to compute:

$$P_{x_i}(\pi) = \{x_i \to t^x \mid t \text{ is associated with } x_i\},$$
$$P_{y_i}(\pi) = \{y_i \to t^y \mid t \text{ is associated with } y_i\},$$
$$P_{\alpha^x_i}(\pi) = \{\alpha^x_i \to t^x \mid t \text{ is associated with } \alpha_i\},$$
$$P_{\alpha^y_j}(\pi) = \left\{\alpha^y_j \to t^y \,\middle|\, t \text{ is associated with } \alpha_i\right\}.$$

In the sequel, we will use "z_i" to denote "x_i or y_i" and "z" to denote "x or y".

Definition 4.8 (Constraint formula of a cut)**.** Let $c \in \mathrm{QCuts}(\pi)$ and A_c be the cut formula of c. We construct a new formula of productions by induction on the ancestor tree of A_c. If μ is an occurrence of a formula on the weak side of c, we define $\mathcal{C}^x_c(\mu,\pi)$ in the following manner:

- If μ is quantifier-free , then $\mathcal{C}^x_c(\mu,\pi) = \top$.
- If μ is introduced by a weakening, then let $z_1,\ldots,z_k$ be the bound variables in μ and

$$\mathcal{C}^x_c(\pi,\pi) := \bigwedge_{j=1}^{k} \bigwedge \neg P_{z_j}(\pi).$$

- If μ is introduced by a $\wedge_r$-rule, as in

$$\frac{\Gamma_1 \vdash \Delta_1, B_{[\nu_1]} \quad \Gamma_2 \vdash \Delta_2, C_{[\nu_2]}}{\Gamma \vdash \Delta,(B \wedge C)_{[\mu]}} \wedge_r,$$

  then $\mathcal{C}^x_c(\mu,\pi) = \mathcal{C}^x_c(\nu_1,\pi) \wedge \mathcal{C}^x_c(\nu_2,\pi)$. The same holds in the case of a $\vee_l$-inference.
- If μ is introduced by a $\wedge_l$-rule, as in

$$\frac{B_{[\nu_1]},C_{[\nu_2]},\Gamma \vdash \Delta}{(B \wedge C)_{[\mu]},\Gamma \vdash \Delta} \wedge_l,$$

  then $\mathcal{C}^x_c(\mu,\pi) = \mathcal{C}^x_c(\nu_1,\pi) \wedge \mathcal{C}^x_c(\nu_2,\pi)$, and analogously for $\vee_r$.
- If μ arises from a contraction on the right, i.e.

$$\frac{\Gamma \vdash \Delta, B_{[\nu_1]},B_{[\nu_2]}}{\Gamma \vdash \Delta,B_{[\mu]}} c_r,$$

  then $\mathcal{C}^x_c(\mu,\pi) = \mathcal{C}^x_c(\nu_1,\pi) \vee \mathcal{C}^x_c(\nu_2,\pi)$, and analogously for a contraction on the left.

- If μ is introduced by a $\neg_r$ rule, as in

$$\frac{\Gamma, B_{[\nu]} \vdash \Delta}{\Gamma \vdash \Delta, (\neg B)_{[\mu]}} \neg_r,$$

then $\mathcal{C}_c^x(\mu,\pi) = \mathcal{C}_c^x(\nu,\pi)$. The same is true for a $\neg_l$-inference.

- If μ is introduced by a quantifier rule, i.e.

$$\frac{\Gamma \vdash \Delta, (B[z\backslash t])_{[\nu]}}{\Gamma \vdash \Delta, (\exists z B)_{[\mu]}} \exists_r,$$

then

$$\mathcal{C}_c^x(\mu,\pi) = (\beta_1^x \to t^x \vee \ldots \vee \beta_r^x \to t^x) \wedge \mathcal{C}_c^x(\nu,\pi),$$

where $\beta_1, \ldots, \beta_k \in EV(c)$ are the eigenvariables associated with the quantifier $\exists z$ that have x-copies in $N_{Cuts}(\pi)$, if such eigenvariables exist. If there is no such eigenvariable, then $\mathcal{C}_c^x(\mu,\pi) = \mathcal{C}_c^x(\nu,\pi)$. The case of a $\forall_l$-inference is analogous.

- We skip over all unary inferences whose active formula is not μ, i.e. inferences that only operate on the context.

A formula $\mathcal{C}_c^y(\mu,\pi)$ is defined analogously. Finally, we set $\mathcal{C}_c(\pi) := \mathcal{C}_c^x(\mu_0,\pi) \wedge \mathcal{C}_c^y(\mu_0,\pi)$, where μ_0 is the active occurence of A_c in the weak premise of c.

Definition 4.9 (Grammar of a proof)**.** Let π be a proof of $A_1, \ldots, A_m \vdash B_1, \ldots, B_n$. The constrained grammar $G(\pi) = \langle \varphi, N(\pi), \Sigma, P(\pi), \mathcal{C}(\pi) \rangle$, where

φ is a new symbol;
$N(\pi) = N_{ES}(\pi) \cup N_{Cuts}(\pi) \cup \{\varphi\}$;
Σ is the language of π;

$$P(\pi) = \bigcup_{x \in N_{ES}(\pi)} P_x(\pi) \cup \bigcup_{\alpha \in N_{Cuts}(\pi)} P_\alpha(\pi) \cup \{\varphi \to A \vee B\};$$

$$\mathcal{C}(\pi) = \mathcal{C}_{ES}(\pi) \wedge \bigwedge_{c \in \mathrm{QCuts}(\pi)} \mathcal{C}_c(\pi) \wedge TR_{P(\pi)},$$

is called the *grammar* of π.

Note that the definition of $\mathcal{C}(\pi)$ implies the total rigidity of $G(\pi)$.

From this definition, it is immediately clear that the first step of any valid derivation instantiates φ as the single formula in the end sequent. It follows that $L(G_{ES}(\pi))$ consists of partial instances of that formula; as we noted in the definition of the language of a constrained grammar, it is possible that there are nonterminals remaining at the end of a derivation. Nonterminals that no productions can be applied to may result from weakenings in the proof.

Let A, B be formulas. We write $A \leq B$ to express that B arises from A by substituting positive instances of $\bot$ and negative instances of $\top$ by other formulas. If M,N are sets of formulas, then $M \leq N$ means that for every $A \in M$, there is a $B \in N$ such that $A \leq B$ (i.e. M is bounded above by N).

Lemma 4.10. *Let π be a proof and $\mathcal{C}(\pi)$ the constraint formula of its grammar $G(\pi)$.*

1. *d is a valid derivation of $G(\pi)$ iff d uses at most one production for each nonterminal and each of the formulas $v_d(\mathcal{C}_{ES}(\pi))$ and $v_d(\mathcal{C}_c(\pi)), c \in \mathrm{QCuts}(\pi)$ is satisfiable.*
2. *$\mathcal{C}_{ES}(\pi)$ and each $\mathcal{C}_c(\pi)$ are satisfiable.*

Proof.

1. First, note that the formula $v_d(\mathcal{C}_{ES}(\pi) \wedge \bigwedge_{c\in \mathrm{QCuts}(\pi)} \mathcal{C}_c(\pi) \wedge TR_{P(\pi)})$ is satisfiable iff $v_d(\mathcal{C}_{ES}(\pi) \wedge \bigwedge_{c\in \mathrm{QCuts}(\pi)} \mathcal{C}_c(\pi))$ is satisfiable and d obeys total rigidity. Since the constraint formulas of the end sequent and those of cuts have pairwise disjoint sets of atoms, the satisfiability of $v_d(\mathcal{C}_{ES}(\pi) \wedge \bigwedge_{c\in \mathrm{QCuts}(\pi)} \mathcal{C}_c(\pi))$ is equivalent to the individual satisfiability of $v_d(\mathcal{C}_{ES}(\pi))$ and all $v_d(\mathcal{C}_c(\pi))$.
2. Let us first consider $\mathcal{C}_{ES}(\pi)$. By the same argument as above, it is sufficient to show that each $\mathcal{C}_{ES}(\mu,\pi)$, where μ is a formula occurrence in the end sequent, is satisfiable. We show this by induction on the complexity of $\mathcal{C}_{ES}(\mu,\pi)$. In the cases of μ quantifier-free or μ introduced by weakening, $\mathcal{C}_{ES}(\mu,\pi)$ is satisfied by the empty interpretation or by interpreting each atom occurring in it as $\bot$, respectively.

   If μ is introduced from ν_1 and ν_2 by a $\vee$- or $\wedge$-inference and $\mathcal{C}_{ES}(\nu_1,\pi), \mathcal{C}_{ES}(\nu_2,\pi)$ are satisfied by interpretations I_1, I_2 respectively, then $\mathcal{C}_{ES}(\mu,\pi)$ is satisfied by $I_1 \cup I_2$ since I_1 and I_2 do not share any atoms.

   If μ is introduced by a quantifier inference from ν and $\mathcal{C}_{ES}(\nu,\pi)$ is satisfied by I, $\mathcal{C}_{ES}(\mu,\pi)$ is of the form $p \wedge \mathcal{C}_{ES}(\nu,\pi)$ and can be satified by the interpretation $I \cup \{p\}$. Note that this is an actual interpretation because p does not occur in $\mathcal{C}_{ES}(\nu,\pi)$ and hence is not assigned in I.

   In the case of a contraction, $\mathcal{C}_{ES}(\mu,\pi)$ is of the form $\mathcal{C}_{ES}(\nu_1,\pi) \vee \mathcal{C}_{ES}(\nu_2,\pi)$ and $\mathcal{C}_{ES}(\nu_1,\pi), \mathcal{C}_{ES}(\nu_2,\pi)$ are satisfiable by the induction hypothesis. It follows that any satisfying interpretation of either also satisfies $\mathcal{C}_{ES}(\mu,\pi)$.

   The remaining cases are trivial. The argument for the constraint formulas of cuts proceeds analogously.

□

Theorem 4.11. *Let π be a proof of $\vdash \exists x_1 \ldots \exists x_m A \vee \exists y_1 \ldots \exists y_n B$ in which all cuts are unquantified or have their cut formulas introduced by weakening. Then $\mathcal{H}(\pi) \leq L(G(\pi))$.*

Proof. First, let us consider $Ex(\pi) = \vdash E$. Since $\vdash Dp(E) = Dp(\vdash E)$ must be tautological by Theorem 2.11, E cannot be of the form $\bot$ or $\bot \vee \bot$; in fact, if $E = E_1 \vee E_2$, then either E_1 or E_2 must expand all quantifiers in $\exists \bar{x} A$ or $\exists \bar{y} B$, respectively, as the only alternative would be $Dp(E) \leftrightarrow \bot$.

Next, suppose that c is a cut whose cut formula is introduced by a weakening on the weak side. The consequence is that none of the nonterminals of c will have any productions and hence cannot be eliminated once they are introduced in a derivation of $G(\pi)$. If the cut formula of c is introduced by weakening on the strong side, c will simply not contribute anything to the grammar of π.

Now let C be an element of $\mathcal{H}(\pi)$. Due to the above considerations, C is certainly of the form $A' \vee B'$. Moreover, there are numbers $m_0 \leq m$ and $n_0 \leq n$ and terms $s_1,\ldots,s_{m_0},t_1,\ldots,t_{n_0}$ such that for each $i \in \{1,\ldots,m_0\}$, x_i is replaced by s_i in C, and analogously for $y_1,\ldots,y_{n_0}$. Clearly, $x_i \to s_i$ and $y_i \to t_i$ are productions of $G(\pi)$. Therefore, let d be a derivation that begins with $\varphi \to A \vee B$ and then uses the productions $x_i \to s_i$ and $y_i \to t_i$ in any order and as many times as necessary to eliminate all $x_1,\ldots,x_{m_0}$ and $y_1,\ldots,y_{n_0}$ from C.

The result is a formula $A^* \vee B^*$. Note that due to the above considerations about cuts, the terms s_i and t_i, as well as $A^* \vee B^*$ itself, may contain nonterminals of cuts that cannot be eliminated. We need to show that $A^* \vee B^* \in L(G(\pi))$—i.e., that d is a valid derivation of $G(\pi)$—and that $C \leq A^* \vee B^*$. Concerning the first point, we observe that $\mathcal{C}_{ES}(\pi)$ is certainly equivalent to a formula of the form $\mathcal{C}_x \wedge \mathcal{C}_y$ with $\mathcal{C}_x$ only containing x_i-productions and $\mathcal{C}_y$ only y_i-productions. The fact that in A' the variables $x_1,\ldots,x_{m_0}$ are assigned terms $s_1,\ldots,s_{m_0}$ while all x_i with $i > m_0$ remain unassigned implies that there is a branch in π that contains inferences

$$\frac{\Gamma \vdash \Delta, A_i[x_i \backslash s_i]}{\Gamma \vdash \Delta, \exists x_i A_i}\ \exists_r$$

for each $i \in \{1,\ldots,m_0\}$ and on which the formula $\exists x_{m_0+1} \ldots \exists x_m A$ is introduced by a weakening. It follows that there is a clause in $\mathcal{C}_x$ that contains exactly the literals $x_1 \to s_1, \ldots, x_{m_0} \to s_{m_0}$ and $\neg x_i \to s$ for any $i > m_0$ and $x_i \to s \in P_{x_i}(\pi)$. An analogous result holds for the structure of $\mathcal{C}_y$. From this observation, it is clear that v_d is both a partial interpretation of $\mathcal{C}$ and maximal, as extending d would necessitate the use of a production $x_i \to s$ or $y_i \to t$ with $i > m_0$ or $i > n_0$, respectively.

Now suppose that C contains $\bot$ as a subformula. This implies that $\bot$ is an expansion tree of some formula D in E, which in turn means that D is introduced by a weakening somewhere in π. If D is a quantifier-free formula, then D is a subformula of $A^* \vee B^*$ and occupies the same position there as $\bot$ in C. If D contains any quantifiers, then the corresponding variables have not been used in the production d and hence D is again a subformula of $A^* \vee B^*$. All in all, we obtain $C \leq A^* \vee B^*$.

□

We can now investigate the relationship between the scope order of π and the dependency relation of $G(\pi)$ that was alluded to earlier.

Lemma 4.12. *Let $c, c' \in \mathrm{QCuts}(\pi)$ and $\alpha \in EV(c), \alpha' \in EV(c')$.*

1. *If $\alpha \prec^0_{G(\pi)} \alpha'$, then $c <^0_\pi c'$.*
2. *If $\alpha \prec_{G(\pi)} \alpha'$, then $c <_\pi c'$.*

3. *$G(\pi)$ is acyclic.*
4. *If c is minimal in $<_\pi$, then α is minimal with respect to N_{Cuts} in $\prec_{G(\pi)}$.*

Proof.

1. $\alpha \prec^0_{G(\pi)} \alpha$ means that there is a production $\alpha \to t(\beta)$ in $G(\pi)$. This can only be the case if $t(\beta)$ is associated with α. Since $t(\beta)$ is associated with an eigenvariable of c and contains an eigenvariable of c', it is in both the weak side of c and the strong side of c', which entails $c <^0_\pi c'$.
2. Follows immediately from (1).
3. Follows immediately from (2) and Lemma 4.3.
4. Follows immediately from (2).

□

Remark 4.13. For the remainder of this chapter, we will simply call those cut nonterminals that are minimal with respect to N_{Cuts} *minimal*.

Lemma 4.14. *Let π and π' be proofs such that $\pi \overset{pr}{\rightsquigarrow} \pi'$, cf. Definition 1.10. If α, β are as depicted there, then $L(G(\pi')) \subseteq L(G(\pi))$.*

Proof. Let $d' = \varphi \to^* s$ be a valid derivation of $G(\pi')$. Suppose that there is a term $t(\alpha,\beta)$ in ψ such that the production $\sigma^z \to (t[\beta\backslash\alpha])^z$ is used in d'. It follows that $\sigma^z \to t^z$ is a production of $G(\pi)$. Since α and β are associated with the same terms, the α^z and β^z must have the same productions in $G(\pi)$. We can thus obtain a derivation d of $G(\pi)$ by first replacing $\sigma^z \to (t[\beta\backslash\alpha])^z$ with $\sigma^z \to t^z$. If no production for α^z is used in d', we are done; if a production $\alpha^z \to r^z$ is used, we add $\beta^z \to r^z$ at any point after $\sigma^z \to t^z$. The validity of d is easy to verify; in the case where a production $\beta^z \to r^z$ is added, v_d does not invalidate the constraint formula of $G(\pi)$ because $v_d = v_{d'} \cup \{\beta^z \to t^z\}$ and whenever $\mathcal{C}(\pi')$ contains a positive instance of $\alpha^z \to t^z$, $\mathcal{C}(\pi)$ must contain a positive instance of $\alpha^z \to t^z \vee \beta^z \to t^z$ in the same position.

□

Theorem 4.15. *Let π, π' be proofs of $\exists\bar{x}A \vee \exists\bar{y}B$ and $\pi \rightsquigarrow \pi'$ by one of the cut reduction steps defined in 1.7, except contraction and weakening. Then $L(G(\pi)) = L(G(\pi'))$.*

Proof. First of all, observe that no cut reduction step changes the end sequent part of $G(\pi)$ and consequently, we only need to consider what happens to the parts of $G(\pi)$ that originate from cuts. We shall consider each of the steps in 1.7 in turn. In each case, we will assume that A_c is Σ_1 and note the changes that need to be made in case of a Π_1-formula.

1. Rule permutations obviously have no effect on the grammar, regardless of the type of A_c.
2. If A_c is an axiom, then $c \notin \mathrm{QCuts}(\pi)$ and eliminating c has no effect on the grammar.

5. Obviously, there is only a single production for α^x in $G(\pi)$, namely $\alpha^x \to t^x$, and $\mathcal{C}_c^x$ has the form $\alpha^x \to t^x \wedge \mathcal{B}^x$. In $G(\pi')$, α^x and its single production are deleted and any production $\beta^x \to s^x \in P(\pi)$ is replaced by $\beta^x \to s^x[\alpha^x \backslash t^x]$. Of course, analogous statements hold for α^y. Moreover, the constraint formula of $G(\pi')$ is obtained by replacing $\mathcal{C}_c^x$ and $\mathcal{C}_c^y$ in $\mathcal{C}$ with $\mathcal{B}^x$ and $\mathcal{B}^y$, respectively. Now it is easy to see that both grammars generate the same language, cf. the proof of Lemma 3.10. If A_c is Π_1, this case cannot occur.
6. $A_c = \forall x B$ is impossible for a Σ_1-formula. The case where A_c is Π_1 is treated analogously to 5.
7. Let $EV(c) = \{\beta_1, \dots, \beta_n, \gamma_1, \dots, \gamma_m\}$ such that the β_i and γ_i occur in B and C respectively. Clearly, $\mathcal{C}_c^x(\pi) = \mathcal{B}^{x\prime} \wedge \mathcal{B}^{x\prime\prime}$. In π', c is replaced by two new cuts c', c'' with cut formulas B and C respectively; this replacement leaves all nonterminals and their productions unchanged. It follows that $\mathcal{C}_{c'}^x(\pi') = \mathcal{B}^{x\prime}$ and $\mathcal{C}_{c''}^x(\pi') = \mathcal{B}^{x\prime\prime}$, which implies $\mathcal{C}(\pi')^x = \mathcal{C}(\pi)^x$. The same argument can be made for y and hence $G(\pi') = G(\pi)$. This proof also works in the case of a Π_1-cut.
8. Disjunction is handled analogously to conjunction, in either case of the type of A_c.
9. It is straightforward to see that removing $\neg$-inferences does not change the grammar at all.

□

The following lemma is the main technical result of this thesis.

Lemma 4.16. *Let π be a pruned proof of $\exists \bar{x} A \vee \exists \bar{y} B$ and c a minimal cut in π. If $\pi \rightsquigarrow \pi'$ by reducing c according to a contraction rule, then $L(G(\pi)) = L(G(\pi'))$.*

Proof. We assume that c is Σ_1; the case of a Π_1-cut can be treated by switching the strong and weak sides. Let $G(\pi') = \langle \varphi, N', \Sigma, P', \mathcal{C}' \rangle$.

First, suppose that the contraction that is reduced is on the left-hand side of c; This situation is pictured in Definition 1.7. The first thing we note is that the only nonterminals that are affected by the proof transformation are those introduced in ψ_2. Due to the minimality of c, there are no quantified cuts in ψ_2 and hence the only eigenvariables therein are those of cuts below c and those of c itself. Let $EV(c) = \{\alpha_1, \dots, \alpha_n\}$. In $G(\pi')$, each α_i is replaced by two new copies α_i' and α_i''. Moreover, if $\tilde{c}$ is a cut in π such that c is on the strong side of $\tilde{c}$, then there might be eigenvariables of $\tilde{c}$ that are introduced within ψ_2. Let $\beta_1, \dots, \beta_m$ be all such eigenvariables; it follows that π' contains two new copies β_i', β_i'' for each of them.

Let us now consider the effects of the reduction on the end sequent part of the grammar. The nonterminals are obviously unchanged, but the productions and the constraint formula are not. If $z_j \to t^z$ is a production of the end sequent, then it is replaced in $G(\pi')$ by two new productions

$$z_j \to t^z[\alpha_1^z \backslash \alpha_1'^z, \dots, \alpha_n^z \backslash \alpha_n'^z, \beta_1^z \backslash \beta_1'^z, \dots, \beta_m^z \backslash \beta_m'^z] \text{ and}$$
$$z_j \to t^z[\alpha_1^z \backslash \alpha_1''^z, \dots, \alpha_n^z \backslash \alpha_n''^z, \beta_1^z \backslash \beta_1''^z, \dots, \beta_m^z \backslash \beta_m''^z].$$

By the same token, if ν is any formula occurrence in the conclusion of c, then

$$\mathcal{C}_{ES}(\nu,\pi') = \mathcal{C}_{ES}(\nu,\pi)[\alpha_1^z\backslash{\alpha'_1}^z,\dots,\alpha_n^z\backslash{\alpha'_n}^z,\beta_1^z\backslash{\beta'_1}^z,\dots,\beta_m^z\backslash{\beta'_m}^z]\vee$$
$$\vee\, \mathcal{C}_{ES}(\nu,\pi)[\alpha_1^z\backslash{\alpha''_1}^z,\dots,\alpha_n^z\backslash{\alpha''_n}^z,\beta_1^z\backslash{\beta''_1}^z,\dots,\beta_m^z\backslash{\beta''_m}^z].$$

Now we consider the rest of the grammar. If μ' and μ'' are the two occurrences of A_c on the weak side of c, then one of them is arbitrarily designated as the cut formula of c' and the other as the cut formula of c''; w.l.o.g we assume that μ' is the cut formula of c' and μ'' the cut formula of c''. The productions of α_i are split between α'_i and α''_i accordingly, that is, if $\alpha_i^z \to t^z$ is a production of $G(\pi)$ and t introduces a quantifier in μ', then ${\alpha'_i}^z \to t^z$ is a production of $G(\pi')$ and analogously if t introduces a quantifier in μ''.

As for the β_i, each of them originates from a cut below c whose weak side is entirely unaffected by the duplication of ψ_2, so ${\beta'_i}^z$ and ${\beta''_i}^z$ simply inherit the productions of β_i^z.

The constraint formula of c is necessarily of the form $(\mathcal{B}'^x \vee \mathcal{B}''^x) \wedge (\mathcal{B}'^y \vee \mathcal{B}''^y)$; it follows that the constraint formulas of c' and c'' are $\mathcal{B}'^x \wedge B'^y$ and $\mathcal{B}''^x \wedge B''^y$, respectively, up to replacement of nonterminals by their fresh copies:

$$\mathcal{C}'_{c'} = \mathcal{B}'^x\{\alpha_1^x\backslash{\alpha'_1}^x,\dots,\alpha_n^x\backslash{\alpha'_n}^x\} \wedge \mathcal{B}'^y\{\alpha_1^y\backslash{\alpha'_1}^y,\dots,\alpha_n^y\backslash{\alpha'_n}^y\},$$
$$\mathcal{C}'_{c''} = \mathcal{B}''^x\{\alpha_1^x\backslash{\alpha''_1}^x,\dots,\alpha_n^x\backslash{\alpha''_n}^x\} \wedge \mathcal{B}'^y\{\alpha_1^y\backslash{\alpha''_1}^y,\dots,\alpha_n^y\backslash{\alpha''_n}^y\}.$$

If c is above the strong side of $\tilde{c}$, then eigenvariables of $\tilde{c}$ might be duplicated, as noted above. In that case, we obtain the new constraint formula of $\tilde{c}$ by replacing each $\beta_i^z \to t^z$ in $\mathcal{C}_{\tilde{c}}(\pi)$ with ${\beta'_i}^z \to t^z \vee {\beta''_i}^z \to t^z$.

Summing up, $G(\pi')$ has the following components:

- Nonterminals:

$$N' = (N \setminus \{\alpha_1^z,\dots,\alpha_n^z,\beta_1^z,\dots,\beta_m^z\}) \cup$$
$$\cup\, \{{\alpha'_1}^z,\dots,{\alpha'_n}^z,{\beta'_1}^z,\dots,{\beta'_m}^z\}\cup$$
$$\cup\, \{{\alpha''_1}^z,\dots,{\alpha''_n}^z,{\beta''_1}^z,\dots,{\beta''_m}^z\}$$

- Productions:

$$
\begin{aligned}
P' = P \setminus \Bigg(& \bigcup_{z_i \in N_{ES}(\pi)} P_{z_i} \cup \bigcup_{i=1}^{n} P_{\alpha_i^z} \cup \bigcup_{i=1}^{m} P_{\beta_i^z} \Bigg) \cup \\
\cup & \bigcup_{z_i \in N_{ES}(\pi)} P_{z_i}[\alpha_1^z \backslash {\alpha'_1}^z, \dots, \alpha_n^z \backslash {\alpha'_n}^z, \beta_1^z \backslash {\beta'_1}^z, \dots, \beta_m^z \backslash {\beta'_m}^z] \cup \\
\cup & \bigcup_{z_i \in N_{ES}(\pi)} P_{z_i}[\alpha_1^z \backslash {\alpha''_1}^z, \dots, \alpha_n^z \backslash {\alpha''_n}^z, \beta_1^z \backslash {\beta''_1}^z, \dots, \beta_m^z \backslash {\beta''_m}^z] \cup \\
\cup & \bigcup_{i=1}^{n} \left\{ {\alpha'_i}^z \to t^z \,\middle|\, t \text{ is associated with } \alpha_i \text{ and introduces a quantifier in } \mu', z \in \{x, y\} \right\} \cup \\
\cup & \bigcup_{i=1}^{n} \left\{ {\alpha''_i}^z \to t^z \,\middle|\, t \text{ is associated with } \alpha_i \text{ and introduces a quantifier in } \mu'', z \in \{x, y\} \right\} \cup \\
\cup & \bigcup_{i=1}^{m} P_{\beta_i} \{\beta_1^z \backslash {\beta'_1}^z, \dots, \beta_m^z \backslash {\beta'_m}^z\} \\
\cup & \bigcup_{i=1}^{m} P_{\beta_i} \{\beta_1^z \backslash {\beta''_1}^z, \dots, \beta_m^z \backslash {\beta''_m}^z\}
\end{aligned}
$$

- Constraint formula:

$$
\begin{aligned}
\mathcal{C}' = & \, \mathcal{C}_{ES}(\pi') \wedge \\
\wedge & \bigwedge_{\substack{\tilde{c} \in \mathrm{QCuts}(\pi) \\ \tilde{c} \neq c}} \mathcal{C}_{\tilde{c}(\pi)} \wedge \\
\wedge & \, \mathcal{B}'^x \{\alpha_1^x \backslash {\alpha'_1}^x, \dots, \alpha_n^x \backslash {\alpha'_n}^x\} \wedge \mathcal{B}'^y \{\alpha_1^y \backslash {\alpha'_1}^y, \dots, \alpha_n^y \backslash {\alpha'_n}^y\} \wedge \\
\wedge & \, \mathcal{B}''^x \{\alpha_1^x \backslash {\alpha''_1}^x, \dots, \alpha_n^x \backslash {\alpha''_n}^x\} \wedge \mathcal{B}''^y \{\alpha_1^y \backslash {\alpha''_1}^y, \dots, \alpha_n^y \backslash {\alpha''_n}^y\} \wedge \\
\wedge & \, TR_{P'}
\end{aligned}
$$

Now let $d = \varphi \to^* s$ be a valid derivation of $G(\pi)$. If no nonterminals belonging to c are used in d then all we have to do to obtain a valid derivation of $G(\pi')$ is replace each β_i that occurs in d with β'_i. If, on the other hand, such nonterminals are used, then all of them must be produced from nonterminals of the end sequent due to the minimality of c. Let $\alpha_{i_1}^x, \dots, \alpha_{i_l}^x, \alpha_{i_{l+1}}^y, \dots, \alpha_{i_r}^y$ be those nonterminals of c that occur in d. If any of the $\alpha_{i_j}^x$ are later replaced by terms, then either all of these terms are above μ' or all of them are above μ''. To see this, assume w.l.o.g. that $\alpha_{i_1}^x$ is later replaced by a term t_1^x that introduces a quantifier in μ', but not in μ'' and $\alpha_{i_2}^x$ by a term t_2^x for which the converse is true. Since d is valid, the atom $\alpha_{i_j}^x \to t_j^x$ in $\mathcal{C}_c(\pi)$ is assigned the value $\top$ by v_d and no other atoms beginning with $\alpha_{i_j}^x$ can have that value, due to total rigidity. $\mathcal{C}_c^x(\pi)$ is certainly of the form $\tilde{\mathcal{B}}' \vee \tilde{\mathcal{B}}''$ with $\alpha_{i_j}^x \to t_j^x$ only occurring in $\tilde{\mathcal{B}}^{(j)}$ because of our stipulations about the terms t_j^x. It follows that on the one hand, no positive literal beginning with $\alpha^x_{i_2}$ within $\tilde{\mathcal{B}}'$ can be assigned the value $\top$, while on the other hand the literal $\neg\alpha^x_{i_2} \to t^x_2$ certainly evaluates to $\bot$ (if it occurs at all) in $\tilde{\mathcal{B}}'$. As a consequence, $\tilde{\mathcal{B}}'$ is unsatisfiable under v_d. An analogous argument shows the unsatisfiability of $\tilde{\mathcal{B}}''$ under v_d, which leads to a contradiction with the validity of d.

We now consider the case where all terms produced from the $\alpha^x_{i_j}$ and $\alpha^y_{i_j}$ introduce quantifiers in μ'. In this case, replacing all $\alpha^x_{i_j}$ in d with $\alpha'_{i_j}{}^x$ yields productions of $G(\pi')$. An analogous substitution applied to the $\alpha^y_{i_j}$ gives a new derivation d'. The derivation d might also contain some of the β^z_i. Since the β'^x_i and the β''^x_i have the same productions in P' as the β^x_i do in P, we can simply replace all β^x_i with β'^x_i and analogously for the β^y_i. In the case where the terms produced from the $\alpha^x_{i_j}$ or $\alpha^y_{i_j}$ (or both) introduce quantifiers in μ'', we replace the corresponding $\alpha^z_{i_j}$ and β^z_i by their respective $''$-versions instead.

Thus, we obtain a derivation d'' that certainly consists of productions of $G(\pi')$; we now need to show that it is in fact valid. By Lemma 4.10, it is sufficient to show the satisfiability of the separate conjuncts of $v_{d''}(\mathcal{C}(\pi'))$. First of all, note that d'' does not invalidate $TR_{P'}$. It is easy to see that $v_{d''}(\mathcal{C}_{ES}(\pi')$ is satisfiable. If $\tilde{c}$ is any cut with an eigenvariable among the β_i, say β_{i_0}, and β_{i_0} has an associated term t, then the production $\beta^z_{i_0} \to t^z$ in $\mathcal{C}_{\tilde{c}}$ has been replaced with $\beta'_{i_0}{}^z \to t^z \vee \beta''_{i_0}{}^z \to t^z$ in $G(\pi')$ and since $v_d(\mathcal{C}_{\tilde{c}}(\pi))$ is satisfiable, so is $v_{d''}(\mathcal{C}_{\tilde{c}}(\pi'))$. The formulas $\mathcal{B}'^x\{\alpha^x_1\backslash\alpha'_1{}^x, \ldots, \alpha^x_n\backslash\alpha'_n{}^x\}$ and $\mathcal{B}'^y\{\alpha^y_1\backslash\alpha'_1{}^y, \ldots, \alpha^y_n\backslash\alpha'_n{}^y\}$ are clearly satisfiable under d'' because they contain exactly the same substitutions relative to $\mathcal{B}'^x$ and $\mathcal{B}'^y$, respectively, as d'' does relative to d. $\mathcal{B}''^x\{\alpha^x_1\backslash\alpha''^x_1, \ldots, \alpha^x_n\backslash\alpha''^x_n\}$ and $\mathcal{B}''^y\{\alpha^y_1\backslash\alpha''^y_1, \ldots, \alpha^y_n\backslash\alpha''^y_n\}$ are trivially satisfiable under d'' because none of their literals are assigned. The maximality of d'' follows immediately from the maximality of d.

Conversely, suppose that we have a derivation d' of $G(\pi')$. The first thing we need to establish is that if nonterminals $\alpha'_{i_1}{}^{z_1}$ and $\alpha''_{i_2}{}^{z_2}$ of c are used in d', then $z_1 \neq z_2$; that is to say, any subderivation that uses only the x_i or the y_i from the end-sequent cannot contain nonterminals of both c' and c''. This is the case because on the one hand, no production of $G(\pi')$ contains nonterminals of both c' and c'' and on the other hand, once a production resulting in nonterminals of either cut is used for some nonterminal of the end sequent, $\mathcal{C}_{ES}(\pi')$ prevents productions of the other kind from being used. An analogous result holds for the β^z_i.

We thus obtain a derivation d of $G(\pi)$ by replacing all $\alpha'^z_i, \beta'^z_i, \alpha''^z_i, \beta''^z_i$ with their original versions. This d does not violate total rigidity because due to the considerations above, d' cannot contain both α'^z_i and α''^z_i for any given i, and analogously for the β^z_i. As in the argument for the other direction, the satisfiability under d of the various parts of $\mathcal{C}$ follows readily from the satisfiability of the corresponding parts of $\mathcal{C}'$.

Now suppose that the contraction happens on the strong side of c. Reducing the contraction leaves us with two new cuts c', c'' whose cut formulas are both A_c. Let μ' and μ'' be the occurrences of A_c that serve as cut formulas for c' and c'' respectively. Each eigenvariable α of c introduces a quantifier in either μ' or μ'' and consequently belongs to either c' or c'' accordingly. Consequently, $EV(c) = EV(c')\dot{\cup}EV(c'')$, where either set on the right might be empty. Thus, let $EV(c) = \{\alpha_1, \ldots, \alpha_n\}$ and assume for the sake of simplicity that $EV(c') = \{\alpha_1, \ldots, \alpha_k\}$ and $EV(c'') = \{\alpha_{k+1}, \ldots, \alpha_n\}$. Each α_i has a duplicate belonging to x or y in $G(\pi')$ iff it has one in $G(\pi)$.

The duplication of the left subproof ψ_1 has extensive effects on the grammar. We will discuss these effects separately for each $\tilde{c} \in \mathrm{QCuts}(\pi)$. First, if $\tilde{c}$ is below c, then c must be on the strong side of $\tilde{c}$ due to c's minimality. As a consequence, it is possible that there are eigenvariables of $\tilde{c}$ that are introduced within ψ_1. If γ is such an eigenvariable, then γ is duplicated, giving rise to eigenvariables γ' and γ''. This duplication naturally carries over to γ's x- and y-versions. Each such γ'^z and γ''^z inherits the productions of γ^z in $G(\pi)$. The constraint formula of $\tilde{c}$ changes in a straightforward manner, by replacing $\gamma^z \to t^z$ with $\gamma'^z \to t^z \vee \gamma''^z \to t^z$ for each γ that is duplicated. In the sequel, let $\{\gamma_1, \ldots, \gamma_l\}$ be all eigenvariables of the original proof duplicated in this manner.

Next, assume that $\tilde{c}$ is located in ψ_1. In this case, $\tilde{c}$ is replaced with two new cuts $\tilde{c}'$ and $\tilde{c}''$. If $\{\beta_1, \ldots, \beta_m\}$ are all eigenvariables that belong to such cuts, then clearly each of them is replaced by two new copies β_i' and β_i''. The productions of the x- and y-versions of these duplicates work out to

$$P'_{\beta'^z_i} = P_{\beta^z_i}\{\bar{\beta}^z \backslash \bar{\beta}'^z, \bar{\gamma}^z \backslash \bar{\gamma}'^z\},$$
$$P'_{\beta''^z_i} = P_{\beta^z_i}\{\bar{\beta}^z \backslash \bar{\beta}''^z, \bar{\gamma}^z \backslash \bar{\gamma}''^z\}$$

for each $i \in \{1, \ldots, m\}$. Similarly, $\tilde{c}'$ and $\tilde{c}''$ have the constraint formulas $\mathcal{C}_{\tilde{c}'} = \mathcal{C}_{\tilde{c}}\{\beta_1^z \backslash {\beta_1'}^z, \ldots, \beta_m^z \backslash {\beta_m'}^z\}$ and $\mathcal{C}_{\tilde{c}''} = \mathcal{C}_{\tilde{c}}\{\beta_1^z \backslash {\beta_1''}^z, \ldots, \beta_m^z \backslash {\beta_m''}^z\}$ respectively.

The final case to consider is that of c itself: The productions of the α_i in $G(\pi')$ work out to

$$P'_{\alpha^z_i} = P_{\alpha^z_i}[\beta_1^z \backslash {\beta_1'}^z, \ldots, \beta_m^z \backslash {\beta_m'}^z, \gamma_1^z \backslash {\gamma_1'}^z, \ldots, \gamma_k^z \backslash {\gamma_l'}^z] \text{ for } i \leq k,$$
$$P'_{\alpha^z_i} = P_{\alpha^z_i}[\beta_1^z \backslash {\beta_1''}^z, \ldots, \beta_m^z \backslash {\beta_m''}^z, \gamma_1^z \backslash {\gamma_1''}^z, \ldots, \gamma_k^z \backslash {\gamma_l''}^z] \text{ for } i > k.$$

The constraint formula of c' can be obtained from $\mathcal{C}_c$ by replacing each literal $\alpha_i^z \to t^z$ that occurs in it with ${\alpha_i}^z \to t^z[\beta_1^z \backslash {\beta_1'}^z, \ldots, \beta_m^z \backslash {\beta_m'}^z, \gamma_1^z \backslash {\gamma_1'}^z, \ldots, \gamma_k^z \backslash {\gamma_k'}^z]$ (for $i \leq k$) removing it (for $i > k$). An analogous transformation yields $\mathcal{C}_{c''}$. If $\tilde{c}$ is any other cut with quantifiers, then $\tilde{c}$ is either within the strong side of c or on a different branch of the proof from c. The first case is impossible due to minimality of c and in the second case, $\tilde{c}$ is unaffected by the proof transformation.

The last thing that needs to be taken care of are the productions and constraint formula of the end sequent. Each production $z_i \to t^z$ is replaced by

$$z_i \to t^z[\beta_1^z \backslash \beta_1' z, \ldots, \beta_m^z \backslash {\beta_m'}^z, \gamma_1^z \backslash {\gamma_1'}^z, \ldots, \gamma_k^z \backslash {\gamma_k'}^z] \text{ and}$$
$$z_i \to t^z[\beta_1^z \backslash \beta_1'' z, \ldots, \beta_m^z \backslash {\beta_m''}^z, \gamma_1^z \backslash {\gamma_1''}^z, \ldots, \gamma_k^z \backslash {\gamma_k''}^z].$$

If t does not contain any β_i or γ_i, then both of these duplicates obviously coincide with the original production and it simply carries over to $G(\pi')$. As for $\mathcal{C}_{ES}(\pi')$, there are formulas $\mathcal{B}_1,\dots,\mathcal{B}_r$ such that

$$\begin{aligned}
\mathcal{C}_{ES}(\pi) &= \mathcal{C}[\mathcal{B}_1,\dots\mathcal{B}_r] \text{ and} \\
\mathcal{C}_{ES}(\pi') &= \mathcal{C}[\mathcal{B}_1[\beta_1^z\backslash\beta_1'z,\dots,\beta_m^z\backslash{\beta_m'}^z,\gamma_1^z\backslash{\gamma_1'}^z,\dots,\gamma_k^z\backslash{\gamma_k'}^z]\vee \\
&\quad \vee \mathcal{B}_1[\beta_1^z\backslash\beta_1''z,\dots,\beta_m^z\backslash{\beta_m''}^z,\gamma_1^z\backslash\gamma_1''^z,\dots,\gamma_k^z\backslash\gamma_k''^z], \\
&\quad \dots \\
&\quad \mathcal{B}_r[\beta_1^z\backslash\beta_1'z,\dots,\beta_m^z\backslash{\beta_m'}^z,\gamma_1^z\backslash{\gamma_1'}^z,\dots,\gamma_k^z\backslash{\gamma_k'}^z]\vee \\
&\quad \vee \mathcal{B}_r[\beta_1^z\backslash\beta_1''z,\dots,\beta_m^z\backslash{\beta_m''}^z,\gamma_1^z\backslash\gamma_1''^z,\dots,\gamma_k^z\backslash\gamma_k''^z]].
\end{aligned}$$

As in the previous case, we can now sum up the contents of the new grammar:

- Nonterminals:

$$\begin{aligned}
N' = N \setminus \{\beta^z \mid \beta \text{ is introduced in } \psi_1\} \cup \\
\cup \left\{\beta'^z, \beta''^z \,\middle|\, \beta \text{ is introduced in } \psi_1\right\}
\end{aligned}$$

- Productions:

$$\begin{aligned}
P' = P \setminus \left(\bigcup_{z_i \in N_{ES}(\pi)} P_{z_i} \cup \bigcup_{i=1}^{n} P_{\alpha_i^z} \cup \bigcup_{\beta \in EV(\psi_1)} P_{\beta^z} \cup \right) \cup \\
\cup \bigcup_{z_i \in N_{ES}(\pi)} P_{z_i}[\beta_1^z\backslash{\beta_1'}^z,\dots,\beta_m\backslash{\beta_m'}^z] \cup \\
\cup \bigcup_{z_i \in N_{ES}(\pi)} P_{z_i}[\beta_1^z\backslash\beta_1''^z,\dots,\beta_m\backslash{\beta_m''}^z] \cup \\
\cup \bigcup_{i=1}^{k} P_{\alpha_i^z}[\bar{\beta}^z\backslash\bar{\beta}'^z,\bar{\gamma}^z\backslash\bar{\gamma}'^z] \cup \\
\cup \bigcup_{i=k+1}^{n} P_{\alpha_i^z}[\bar{\beta}^z\backslash\bar{\beta}''^z,\bar{\gamma}^z\backslash\bar{\gamma}''^z] \cup \\
\cup \bigcup_{i=1}^{m} P_{\beta_i^z}\{\bar{\beta}^z\backslash\bar{\beta}'^z,\bar{\gamma}^z\backslash\bar{\gamma}'^z\} \cup \\
\cup \bigcup_{i=1}^{m} P_{\beta_i^z}\{\bar{\beta}^z\backslash\bar{\beta}''^z,\bar{\gamma}^z\backslash\bar{\gamma}''^z\} \cup \\
\cup \bigcup_{i=1}^{l} P_{\gamma_i^z}\{\bar{\gamma}^z\backslash\bar{\gamma}'^z\} \cup \\
\cup \bigcup_{i=1}^{l} P_{\gamma_i^z}\{\bar{\gamma}^z\backslash\bar{\gamma}''^z\}
\end{aligned}$$

- Constraint formula:

$$
\begin{aligned}
\mathcal{C}' = {} & \mathcal{C}_{ES}(\pi') \\
& \wedge \bigwedge_{\substack{\tilde{c}\in \mathrm{QCuts}(\pi) \\ \tilde{c} \text{ below or parallel to } c}} (\mathcal{C}_{\tilde{c}}\{\bar{\gamma}^z \backslash \bar{\gamma}'^z\} \wedge \mathcal{C}_{\tilde{c}}\{\bar{\gamma}^z \backslash \bar{\gamma}'^z\}) \\
& \wedge \bigwedge_{\tilde{c}\in \mathrm{QCuts}(\psi_1)} \mathcal{C}_{\tilde{c}}\{\bar{\beta}_1 \backslash \bar{\beta}'_1, \ldots, \bar{\beta}_k \backslash \bar{\beta}'_k\} \wedge \\
& \wedge \bigwedge_{\tilde{c}\in \mathrm{QCuts}(\psi_1)} \mathcal{C}_{\tilde{c}}\{\bar{\beta}_1 \backslash \bar{\beta}''_1, \ldots, \bar{\beta}_k \backslash \bar{\beta}''_k\} \wedge \\
& \wedge \mathcal{C}_{c'} \wedge \mathcal{C}_{c''} \wedge \\
& \wedge TR_{P'}.
\end{aligned}
$$

Let d be a valid derivation of $G(\pi)$. If nonterminals of c occur in d, then due to the minimality of c they can only be introduced from nonterminals of the end sequent. Let $\alpha^z_{i_1}, \ldots, \alpha^z_{i_r}$ be those nonterminals of c that are used in d and are later replaced by terms $t^z_1, \ldots, t^z_r$. For each i_j, we replace the production $\alpha^z_{i_j} \to t^z_j$ with $\alpha^z_{i_j} \to t^z_j[\bar{\beta}^z \backslash \bar{\beta}'^z, \bar{\gamma}^z \backslash \bar{\gamma}'^z]$ if $i_j \leq k$ or $\alpha^z_{i_j} \to t^z_j[\bar{\beta}^z \backslash \bar{\beta}''^z, \bar{\gamma}^z \backslash \bar{\gamma}''^z]$ if $i_j > k$. Also, if $z_i \to t^z$ is a production of the end sequent in d, we replace it with $z_i \to t^z[\bar{\beta}^z \backslash \bar{\beta}'^z, \bar{\gamma}^z \backslash \bar{\gamma}'^z]$, obtaining a new derivation d'. This can lead to d' containing both β'_i and β''_i for some i, and similarly for the γ_i. Due to total rigidity, d uses at most one production for each β_i and γ_i and we can simply replace any such production by one or both of its two variants in the new grammar, according to whether one or both copies of the respective nonterminal occur in d'. We call the derivation obtained by this process d''.

As before, it is sufficient to show that d'' is totally rigid and does not invalidate the conjuncts of $\mathcal{C}(\pi')$. $v_{d''}(\mathcal{C}_{c'})$ is satisfiable because up to renaming, the literals of $\mathcal{C}_{c'}$ are a subset of those of $\mathcal{C}_c$ and $v_d(\mathcal{C}_c)$ is satisfiable. The satisfiability of $v_{d''}(\mathcal{C}_{c''})$ is shown in an analogous manner. The constraint formulas of all other cuts are similarly easy to deal with because they contain the same substitutions relative to their original counterparts as d'' does to d. The satisfiability of $v_{d''}(\mathcal{C}_{ES}(\pi'))$ is immediately obvious.

Now suppose that we have a valid derivation d' of $G(\pi')$. First of all, there are some important conclusions to be drawn from the form of $\mathcal{C}_{ES}(\pi')$: If some production $x_i \to t^x(\bar{\alpha}^x)$ is used in d', no production of a nonterminal x_j with $j > i$ that is used in d' can contain any of the β'^x_i or γ'^x_i (or their $''$-versions), and vice versa. Moreover, if there is a production $x_i \to t^x_i(\bar{\beta}'^x, \bar{\gamma}'^x)$ in d', then productions $x_j \to t^x_j(\bar{\beta}''^x, \bar{\gamma}''^x)$ with $j > i$ cannot occur in d', and analogously with the $'$- and $''$-nonterminals changed around. Since π is pruned, no term in ψ_2 contains two eigenvariables that introduce the same quantifier. These facts imply that the x-part of d' only uses $'$- or $''$-nonterminals, but not both; the same is naturally true for the y-part. It follows that we can simply replace all $'$- and $''$-nonterminals by their original versions without violating total rigidity. The argument that the resulting derivation d is valid then goes through just as in the previous cases. □

Corollary 4.17. *Let π be a proof of $\vdash \exists\bar{x}A \vee \exists\bar{y}B$. Then $L(G(\pi))$ is tautological.*

Proof. The result is obtained as a combination of Theorem 4.15, Lemma 4.14, Theorem 4.16 and Theorem 4.11. □

4.2 The general case

Theorem 4.18. *Let C be a formula such that C is a Boolean combination of prenex formulas and all quantifiers in C are weak. If π is a proof of $\vdash C$, then $L(G(\pi))$ is tautological.*

Sketch of proof. First, it is clear that Lemmas 4.14 and 4.16 and Theorems 4.15 and 4.11 still hold if the disjunction in $\exists\bar{x}A \vee \bar{y}B$ is replaced with a conjunction or if either block of existential quantifiers is replaced with negated universal quantifiers. Second, each additional block of quantifiers simply induces (at most) another copy for each cut nonterminal; all the proofs still proceed in exactly the same manner. □

Let $\Gamma \vdash \Delta$ be a sequent where every formula is of the form described in Theorem 4.18. If π is a proof of $\Gamma \vdash \Delta$, the only amendment we need to make to the definition of $G(\pi)$ is

$$P_\varphi(\pi) := \{\varphi \to \neg A^* \mid A \in \Gamma\} \cup \{\varphi \to B^* \mid B \in \Delta\}$$

instead of the single production we had for φ in the simple case.

Theorem 4.19. *Let π be a proof of an end sequent consisting only of formulas of the form described in Theorem 4.18. Then $L(G(\pi))$ is tautological.*

Sketch of proof. The proofs of Lemmas 4.14 and 4.16 and Theorems 4.15 and 4.11 still work in exactly the same manner. □

The previous two theorems enable us to present an example of a proof with cuts and its grammar.

Example 4.20. Let $\mathcal{L}$ be the language consisting of six constant symbols a,b,c,d,e,f and a binary relation symbol $\sim$. Let $\mathcal{A}$ be the set of axioms expressing that the constants are all distinct and that $\sim$ is irreflexive and symmetric. We define a few abbreviations:

$$\begin{aligned}
{\sim}\{x_1,x_2,x_3\} &:\leftrightarrow\ x_1{\sim}x_2 \wedge x_2{\sim}x_3 \wedge x_1{\sim}x_3 \\
{\not\sim}\{x_1,x_2,x_3\} &:\leftrightarrow\ x_1{\not\sim}x_2 \wedge x_2{\not\sim}x_3 \wedge x_1{\not\sim}x_3 \\
{\neq}\{x_1,\dots,x_k\} &:\leftrightarrow\ \bigwedge_{i=1}^{k}\bigwedge_{j=i+1}^{k} x_i{\neq}x_j \\
K(x_1,x_2,x_3) &:\leftrightarrow\ {\neq}\{x_1,x_2,x_3\} \wedge {\sim}\{x_1,x_2,x_3\} \\
N(x_1,x_2,x_3) &:\leftrightarrow\ {\neq}\{x_1,x_2,x_3\} \wedge {\not\sim}\{x_1,x_2,x_3\} \\
K_a(z_1,z_2,z_3) &:\leftrightarrow {\neq}\{a,z_1,z_2,z_3\} \wedge \bigwedge_{i=1}^{3} a \sim z_i \\
N_a(z_1,z_2,z_3) &:\leftrightarrow {\neq}\{a,z_1,z_2,z_3\} \wedge \bigwedge_{i=1}^{3} a{\not\sim}z_i
\end{aligned}$$

Then the sequent $\mathcal{A} \vdash \exists\bar{x}\ K(\bar{x}) \vee \exists\bar{y}\ N(\bar{y})$ can be interpreted as "in a group of six people there are always three who all know each other or three who don't know each other". We shall give a proof π containing a cut with the cut formula $\exists\bar{z}\ (K_a(\bar{z}) \vee N_a(\bar{z}))$, or "there are three distinct people who know a or three distinct people who don't know a". This is a well-known special case of Ramsey's Theorem.

In order to construct the grammar of π, we need to discuss the structure of the subproofs $\psi_0,\dots,\psi_{32}$ and $\varphi_0,\dots,\varphi_3$, defined at the end of the example. First, ψ_0 contributes nothing to the grammar. Note that there are 32 distinct possibilities for which of b,c,d,e,f know a. In each of these cases, there either are three who know a or three who don't; this is easy to see using a pigeonhole argument. In each of the subproofs $\psi_1,\dots,\psi_{32}$, we prove the cut formula $\exists\bar{z}\ (K_a \vee N_a)$ under the assumption of one of these constellations. We choose to always instantiate the variables z_1, z_2, z_3 in $K_a \vee N_a$ as the first three elements that do or don't know a in alphabetical order. We also choose to enumerate the 32 constellations by converting them into binary numbers $m_f m_e m_d m_c m_b$ where $m_i = 1$ if i knows a and 0 otherwise. For instance, the constellation where b, c and e know a is converted to 01011, so this is constellation number 11. See the table at the end of the example for how z_1,z_2,z_3 are instantiated in each case.

In φ_1, we prove that if β,γ,δ are distinct and know a and two of them also know each other, there are three people who know each other. We always instantiate x_1 as a and x_2, x_3 alphabetically as the first two elements of $\{\beta,\gamma,\delta\}$ who also know each other. In φ_2, we instantiate (y_1,y_2,y_3) as (β,γ,δ). The proof φ_3 is entirely analogous to the part above the left side of τ, but with $\sim$ and $\not\sim$ switched.

Now we have enough information to compute the grammar of π, restricting ourselves to the formula in the succedent:

- Nonterminals: $N(\pi) = \{\varphi,x_1,x_2,x_3,y_1,y_2,y_3,\beta^x,\beta^y,\gamma^x,\gamma^y,\delta^x,\delta^y\}$

- Productions:

$$\begin{aligned} x_1 &\to a|\beta^x & y_1 &\to a|\beta^y \\ x_2 &\to \beta^x|\gamma^x & y_2 &\to \beta^y|\gamma^y \\ x_3 &\to \gamma^x|\delta^x & y_3 &\to \gamma^y|\delta^y \\ \beta^x &\to b|c|d & \beta^y &\to b|c|d \\ \gamma^x &\to c|d|e & \gamma^y &\to c|d|e \\ \delta^x &\to d|e|f & \delta^y &\to d|e|f \\ \varphi &\to K \vee N \end{aligned}$$

- Constraint formula:

$$\begin{aligned} \mathcal{C}_{ES}(\pi) = \big[&(x_1 \to a \wedge ((x_2 \to \beta^x \wedge x_3 \to \gamma^x) \vee (x_2 \to \beta^x \wedge x_3 \to \delta^x) \vee (x_2 \to \gamma^x \wedge x_3 \to \delta^x))) \\ &\wedge (y_1 \to \beta^y \wedge y_2 \to \gamma^y \wedge y_3 \to \delta^y)\big] \\ &\vee \big[(y_1 \to a \wedge ((y_2 \to \beta^y \wedge y_3 \to \gamma^y) \vee (y_2 \to \beta^y \wedge y_3 \to \delta^y) \vee (y_2 \to \gamma^y \wedge y_3 \to \delta^y))) \\ &\wedge (x_1 \to \beta^x \wedge x_2 \to \gamma^x \wedge x_3 \to \delta^x)\big] \end{aligned}$$

$\mathcal{C}_c^x(\pi)$ is a disjunction of 32 formulas of the form $\beta^x \to t_1 \wedge \gamma^x \to t_2 \wedge \delta^x \to t_3$, where (t_1,t_2,t_3) ranges over the columns in the table. $\mathcal{C}_c^y(\pi)$ is identical up to the superscripts of the nonterminals.

TR_P is straightforward.

We claim that

$$\begin{aligned} L(G(\pi)) = &\overbrace{\{K(a,s_2,s_3) \vee N(t_1,t_2,t_3) \mid s_i,t_i \in \{b,c,d,e,f\},\ s_2 < s_3,\ t_1 < t_2 < t_3\}}^{=:U} \cup \\ \cup &\underbrace{\{K(s_1,s_2,s_3) \vee N(a,t_2,t_3) \mid s_i,t_i \in \{b,c,d,e,f\},\ s_1 < s_2 < s_3,\ t_2 < t_3\}}_{=:V} \end{aligned}$$

("$<$" refers to alphabetical order in the above). For the "$\supseteq$" direction, consider U and let $s_2 < s_3$, $t_1 < t_2 < t_3$. We first use the production $x_1 \to a$. Now $\mathcal{C}_{ES}(\pi)$ allows us to produce $(\beta^x,\gamma^x), (\beta^x,\delta^x)$ or (γ^x,δ^x) from (x_2,x_3). We choose one of the three based on what s_2 and s_3 are. We also have to produce $(\beta^y, \gamma^y, \delta^y)$ from (y_1,y_2,y_3) and can then use those nonterminals to produce (t_1,t_2,t_3). V is treated analogously.

For the "$\subseteq$" direction, let $d : \varphi \to^* K(s_1,s_2,s_3) \vee N(t_1,t_2,t_3)$ be a valid derivation of $G(\pi)$. Due to $\mathcal{C}_{ES}(\pi)$, either s_1 or t_1 must be a; assume s_1 w.l.o.g. It follows that one of the pairs of productions $(x_2 \to \beta^x, x_3 \to \gamma^x), (x_2 \to \beta^x, x_3 \to \delta^x)$ or $(x_2 \to \gamma^x, x_3 \to \delta^x)$ must occur in d and due to $\mathcal{C}_c^x(\pi)$, $s_2 < s_3$, as can easily be seen from the table. The derivation d also certainly contains the productions $y_1 \to \beta^y, y_2 \to \gamma^y$ and $y_3 \to \delta^y$ and hence $t_1 < t_2 < t_3$ by the same argument for $\mathcal{C}_c^y(\pi)$.

Note that $L(G(\pi))$ is not strictly speaking a tautology, as we neglected the axiom set and hence did not compute the entire language. As a consequence, we obtain the result $\mathcal{A} \vDash \bigvee L(G(\pi))$. Why is this the case? We know that in every model of $\mathcal{A}$, there are three people that know each other or three that don't. If a knows two other people s_2 and s_3 who also know each other, all formulas $K(a,s_2,s_3) \vee N(t_1,t_2,t_3)$ in U will be true for any t_1,t_2,t_3. If s_1,s_2,s_3 all know each other and none of them is a, then all $K(s_1,s_2,s_3) \vee N(t_1,t_2,t_3)$ in V will be true. The case where three people don't know each other is treated analogously.

This example illustrates a useful property of $G(\pi)$: It is possible to restrict oneself to computing the part of the grammar (and the language) that is actually interesting and consider the other parts of the end sequent only implicitly. Working out the complete grammar of π and verifying that its language is a proper tautology is left as an exercise for the reader.

π_1:

$$\dfrac{\overset{(\psi_0)}{\mathcal{A} \vdash \bigvee_{i=1}^{32} C_i} \qquad \dfrac{\dfrac{\overset{(\psi_1)}{C_1 \vdash \exists \bar{z}\ (K_a(\bar{z}) \vee N_a(\bar{z}))} \quad \dots \quad \overset{(\psi_{32})}{C_{32} \vdash \exists \bar{z}\ (K_a(\bar{z}) \vee N_a(\bar{z}))}}{\bigvee_{i=1}^{32} C_i \vdash (\exists \bar{z}\ (K_a(\bar{z}) \vee N_a(\bar{z})))^{32}}\ \vee_l \times 31}{\bigvee_{i=1}^{32} C_i \vdash \exists \bar{z}\ (K_a(\bar{z}) \vee N_a(\bar{z}))}\ c_r \times 31}{\mathcal{A} \vdash \exists \bar{z}\ (K_a(\bar{z}) \vee N_a(\bar{z}))}\ cut$$

π_2:

$$\dfrac{\dfrac{\overset{(\varphi_0)}{\mathcal{A} \vdash (\beta{\sim}\gamma \vee \beta{\sim}\delta \vee \gamma{\sim}\delta) \vee \not\sim\{\beta,\gamma,\delta\}} \qquad \dfrac{\dfrac{\overset{(\varphi_1)}{\mathcal{A}, K_a(\beta,\gamma,\delta), \beta{\sim}\gamma \vee \beta{\sim}\delta \vee \gamma{\sim}\delta \vdash \exists \bar{x}\ K\bar{x}} \quad \overset{(\varphi_2)}{\mathcal{A}, K_a(\beta,\gamma,\delta), \not\sim\{\beta,\gamma,\delta\} \vdash \exists \bar{y}\ N\bar{y}}}{\mathcal{A}, (K_a(\beta,\gamma,\delta))^2, \beta{\sim}\gamma \vee \beta{\sim}\delta \vee \gamma{\sim}\delta \vee \not\sim\{\beta,\gamma,\delta\} \vdash \exists \bar{x}\ K\bar{x} \vee \exists \bar{y}\ N\bar{y}}\ \vee_l}{\mathcal{A}, K_a(\beta,\gamma,\delta), \beta{\sim}\gamma \vee \beta{\sim}\delta \vee \gamma{\sim}\delta \vee \not\sim\{\beta,\gamma,\delta\} \vdash \exists \bar{x}\ K\bar{x} \vee \exists \bar{y}\ N\bar{y}}\ c_l}{\mathcal{A}, K_a(\beta,\gamma,\delta) \vdash \exists \bar{x}\ K\bar{x} \vee \exists \bar{y}\ N\bar{y}}\ cut \qquad \overset{(\varphi_3)}{\mathcal{A}, N_a(\beta,\gamma,\delta) \vdash \exists \bar{x}\ K\bar{x} \vee \exists \bar{y}\ N\bar{y}}}{\dfrac{\dfrac{\mathcal{A}, K_a(\beta,\gamma,\delta) \vee N_a(\beta,\gamma,\delta) \vdash (\exists \bar{x}\ K\bar{x} \vee \exists \bar{y}\ N\bar{y})^2}{\mathcal{A}, \exists \bar{z}\ (K_a(\bar{z}) \vee N_a(\bar{z})) \vdash (\exists \bar{x}\ K\bar{x} \vee \exists \bar{y}\ N\bar{y})^2}\ \exists_l \times 3}{\mathcal{A}, \exists \bar{z}\ (K_a(\bar{z}) \vee N_a(\bar{z})) \vdash \exists \bar{x}\ K\bar{x} \vee \exists \bar{y}\ N\bar{y}}\ c_r}\ \vee_{l[\tau]}$$

π:

$$\dfrac{\overset{(\pi_1)}{\mathcal{A} \vdash \exists \bar{z}\ (K_a(\bar{z}) \vee N_a(\bar{z}))} \quad \overset{(\pi_2)}{\mathcal{A}, \exists \bar{z}\ (K_a(\bar{z}) \vee N_a(\bar{z})) \vdash \exists \bar{x}\ K\bar{x} \vee \exists \bar{y}\ N\bar{y}}}{\mathcal{A} \vdash \exists \bar{x}\ K\bar{x} \vee \exists \bar{y}\ N\bar{y}}\ cut_{[c]}$$

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
z_1	b	c	b	d	b	c	b	b	b	c	b	b	b	b	c	b	b	c	b	b	b	b	c	b	b	b	c	b	d	b	c	b
z_2	c	d	d	e	c	e	e	c	c	d	d	c	c	d	d	c	c	d	d	c	c	d	d	c	c	e	e	c	e	d	d	c
z_3	d	e	e	f	e	f	f	d	d	f	f	e	f	e	e	d	d	e	e	f	e	f	f	d	d	f	f	e	f	e	e	d
rel	n	n	n	n	n	n	n	k	n	n	n	k	n	k	k	k	n	n	n	k	n	k	k	k	n	k	k	k	k	k	k	k

CHAPTER 5

Conclusion

This thesis generalizes the results of [Het12a] in two ways: First, we use the concept of the constrained grammar to address cuts with more than one quantifier. It is clear that even in the case of several quantifiers, the resulting grammar should be totally rigid, since each eigenvariable of a cut is instantiated at most once in any given instance of the end sequent. But totally rigid tree grammars are not sufficient to describe the relationships between the terms of different eigenvariables: Choosing a term for some eigenvariable α constrains our choices for eigenvariables that are used within the scope of α's quantifier. Constraint formulas are a natural way of expressing these restrictions.

The other generalization concerns Herbrand's Theorem. A simple version of the theorem states that if $\exists\bar{x}A$ with A quantifier-free is a valid formula, there is a tautological set $\{A[\bar{x}\backslash\bar{t}_1], \ldots, A[\bar{x}\backslash\bar{t}_n]\}$ of instances of A. If one wants to generalize this fact to non-prenex formulas, a question naturally arises: What do we actually mean by "instance" in that case? Given an expansion tree $E \diamond F$ of a formula $A \diamond B$ where $\diamond \in \{\vee, \wedge\}$, the natural definition of an instance of $E \diamond F$ would be "any formula $C \diamond D$ where C is an instance in E and D is an instance in F"; that is, we would essentially "multiply out" the tree every time a binary connective is encountered. The problem with this approach is that if $E \vee F$ was obtained by extraction from a proof π, then not all these combinations necessarily occur in π itself—there may be dependencies between formulas that are parallel in the expansion tree. To deal with this disparity, we essentially define separate copies of the cut part of the grammar that do not interact, ensuring that all combinations of instances of C and D can be generated. In the case of a formula $C \vee D$ with C and D prenex, this is easy: There simply are two copies of the part of the grammar that is generated by cuts. It is also easy to see that this fact can be generalized to any Boolean combination of prenex formulas; the number of copies of the grammar will increase, but apart from that everything stays the same.

Extending the result to arbitrary formulas seems to be possible, but significantly more involved, since different cut nonterminals may vary in the number and of and relationships between their duplicates. A part of the required work is already present in this thesis, as the definitions of all parts of the grammar of a proof, apart from the constraint formula, are already very general. Completing the generalization is left as future work.

Bibliography

[Bom88] E. Bombieri and A.J. van der Poorten: 'Some Quantitative Results Related to Roth's Theorem'. In: *Journal of the Australian Mathematical Society* (1988), vol. 45(2): pp. 233–248 (cit. on p. 1).

[Com07] H. Comon, M. Dauchet, R. Gilleron, C. Löding, F. Jacquemard, D. Lugiez, S. Tison, and M. Tommasi: *Tree Automata Techniques and Applications*. Available on: http://www.grappa.univ-lille3.fr/tata. release October, 12th 2007. 2007 (cit. on p. 21).

[Gen34] G. Gentzen: 'Untersuchungen über das logische Schließen I'. ger. In: *Mathematische Zeitschrift* (1934), vol. 39: pp. 176–210 (cit. on pp. 1, 3).

[Het12a] Stefan Hetzl: 'Applying Tree Languages in Proof Theory'. In: *Language and Automata Theory and Applications*. Ed. by Adrian-Horia Dediu and Carlos Martín-Vide. Vol. 7183. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2012: pp. 301–312 (cit. on pp. 2, 49).

[Het12b] Stefan Hetzl and Lutz Strassburger: 'Herbrand-Confluence for Cut Elimination in Classical First Order Logic'. In: *Computer Science Logic (CSL'12) - 26th International Workshop/21st Annual Conference of the EACSL*. Ed. by Patrick Cégielski and Arnaud Durand. Vol. 16. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2012: pp. 320–334 (cit. on p. 2).

[Jac11] Florent Jacquemard, Francis Klay, and Camille Vacher: 'Rigid tree automata and applications'. In: *Inform. and Comput.* (2011), vol. 209(3): pp. 486–512 (cit. on pp. 2, 20).

[Koh08] Ulrich Kohlenbach: *Applied Proof Theory: Proof Interpretations and their Use in Mathematics*. Springer, 2008 (cit. on p. 1).

[Kre51] G. Kreisel: 'On the Interpretation of Non-Finitist Proofs - Part I'. In: *J. Symb. Log.* (1951), vol. 16(4): pp. 241–267 (cit. on p. 1).

[Kre52] G. Kreisel: 'On the Interpretation of Non-Finitist Proofs: Part II. Interpretation of Number Theory. Applications'. In: *J. Symb. Log.* (1952), vol. 17(1): pp. 43–58 (cit. on p. 1).

[Luc89] Horst Luckhardt: 'Herbrand-Analysen zweier Beweise des Satzes von Roth: Polynomiale Anzahlschranken'. In: *Journal of Symbolic Logic* (1989), vol. 54(1): pp. 234–263 (cit. on p. 1).

[Mil87] DaleA. Miller: 'A compact representation of proofs'. English. In: *Studia Logica* (1987), vol. 46(4): pp. 347–370 (cit. on pp. 2, 9).

[Ore79] V.P. Orevkov: 'Lower Bounds for increasing Complexity of Derivations after Cut Elimination'. In: *Zapiski Nauchnykh Seminarov Leningradskogo Otdeleniya Matematicheskogo Instituta* (1979), vol. 88: pp. 137–161 (cit. on p. 1).

[Pud98] Pavel Pudlák: 'The Lengths of Proofs'. In: *Handbook of Proof Theory*. Ed. by Sam Buss. Elsevier, 1998: pp. 547–637 (cit. on p. 1).

[Sta79] Richard Statman: 'Lower bounds on Herbrand's theorem'. In: *Proceedings of the American Mathematical Society* (1979), vol. 75: pp. 104–107 (cit. on p. 1).

[Tak87] Gaisi Takeuti: *Proof theory*. Second. Vol. 81. Studies in Logic and the Foundations of Mathematics. With an appendix containing contributions by Georg Kreisel, Wolfram Pohlers, Stephen G. Simpson and Solomon Feferman. Amsterdam: North-Holland Publishing Co., 1987: pp. x+490 (cit. on p. 6).

Index