FAKULTÄT FÜR !NFORMATIK

Faculty of Informatics

Abstrakte Beweisstrukturen

Ein einheitliches Framework

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Logic and Computation

eingereicht von

Andreas Humenberger
Matrikelnummer 1026602

an der Fakultät für Informatik
der Technischen Universität Wien

Betreuung: Ass.Prof. Dr.techn. Stefan Hetzl

Wien, 25. August 2016

Andreas Humenberger

Stefan Hetzl

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.ac.at

FAKULTÄT
FÜR !NFORMATIK
Faculty of Informatics

Abstract Proof Structures

A uniform framework

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Logic and Computation

by

Andreas Humenberger
Registration Number 1026602

to the Faculty of Informatics
at the Vienna University of Technology

Advisor: Ass.Prof. Dr.techn. Stefan Hetzl

Vienna, 25th August, 2016

Andreas Humenberger

Stefan Hetzl

Erklärung zur Verfassung der Arbeit

Andreas Humenberger
Tigergasse 33/5
1080 Wien

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 25. August 2016

Andreas Humenberger

Acknowledgements

First and foremost, I would like to thank my supervisor Stefan Hetzl. Not only did he introduce me to this very interesting topic, he also spent a vast amount of time discussing my work and providing help. It was also Stefan, who encouraged me to consider various aspects of my thesis from different points of view which made working on this thesis even more interesting. This is also the main non-math related insight I gained while working on this thesis.

I am also very grateful to my family, especially to my parents Maria and Maximilian, who continuously supported me in every possible way. I would not be at this great point in my life without their support.

Last but not least, I would like to thank my friends and my dear Sophie for making the wonderful time during my studies even better.

Kurzfassung

Abstraktionen von formalen Beweisen, sogenannte *abstrakte Beweisstrukturen*, dienen als anerkanntes Werkzeug um Struktur und Eigenschaften von formalen Beweisen zu untersuchen. Für einen Beweis in einem Sequentialkalkül existieren verschiedenste Abstraktionen, wie *Beweisskelett*, *Beweisnetz* und *Logischer Flussgraph* (im speziellen *Atomarer Flussgraph*). Diese abstrakten Beweisstrukturen haben ihren Ursprung in verschiedenen Gebieten der Beweistheorie und eine gründliche Untersuchung der Zusammenhänge dieser existiert noch nicht.

Durch die Definition einer Tupel-Darstellung von *LK*-Beweisen wird die Grundlage eines einheitlichen Rahmens zur Untersuchung von Beweisstrukturen im Sequentialkalkül *LK* gelegt. Diese Tupel-Darstellung erlaubt es, die oben genannten abstrakten Beweisstrukturen geeignet darzustellen und zu untersuchen.

Wir zeigen, dass es bei geeigneter Behandlung der Abschwächung für jedes Paar der oben genannten Abstraktionen eine Struktur gibt, sodass beide zu dieser reduziert werden können. Neben den Einblicken in die Zusammenhänge der Beweisstrukturen entsteht ein Rahmen zur Verallgemeinerung von Resultaten und Algorithmen. So definierten etwa Krajíček und Pudlák [9] einen Algorithmus zur Herleitung von Schranken bezüglich der minimalen Größe von Beweisen. Durch Verallgemeinerung dieses Algorithmus ergeben sich dieselben Schranken für Beweisnetze.

Außerdem untersuchen wir die Kardinalitäten der Äquivalenzklassen, welche von den Abstraktionen generiert werden. Wir zeigen, dass es endlich viele Beweise gibt, welche dasselbe Beweisnetz (denselben Atomaren Flussgraphen) besitzen. Für Beweisskelette gibt es im Allgemeinen unendlich viele dazugehörige Beweise.

Abstract

Abstractions of formal proofs, so-called *abstract proof structures*, serve as a well-accepted tool for studying structure and properties of formal proofs. Given a sequent calculus proof, there are various abstractions including *proof skeleton*, *proof net* and *logical flow graph* (in particular *atomic flow graph*). These abstract proof structures emerged in different areas of proof theory and a thorough investigation of their interrelationship does not exist so far.

By introducing a tuple-based representation of proofs, which allows a suitable representation of the before-mentioned abstractions, we establish a uniform framework for classical first-order logic which clarifies the relationship between these proof structures in the context of the sequent calculus LK.

We show that, in case of a suitable treatment of the weakening rules, there exists a structure for every pair of the above-mentioned abstractions such that both abstractions can be reduced to it. Besides the gained insights of the interrelationship by defining this uniform framework, we get a framework for generalizing results and algorithms. For instance, Krajíček and Pudlák [9] introduced an algorithm defined on proof skeletons for deriving bounds on the minimal size of proofs. We generalize this result to proof nets by generalizing the algorithm to proof net skeletons. A proof net skeleton is an abstraction of both, proof nets and proof skeletons.

Furthermore, we investigate the cardinalities of the equivalences generated by the abstractions. We show that there exist finitely many proofs having the same proof net (atomic flow graph). For proof skeletons there exists an infinite number of associated proofs.

Contents

CHAPTER 1

Introduction

Abstractions of formal proofs, so-called *abstract proof structures*, serve as a well-accepted tool for studying structure and properties of formal proofs. For instance, Orevkov [11] showed, for a large class of Hilbert-type calculi, that it is undecidable whether a given formula has a proof with a given *proof skeleton* where a proof skeleton is a formal proof without formulas. Krajíček and Pudlák [9] proved a similar result for LK, namely, that it is undecidable whether a given sequent has an LK-proof with a given skeleton. In [9], Krajíček and Pudlák also gave upper bounds on the depth of terms in LK-proofs by using proof skeletons.

The investigations of Orevkov and Krajíček and Pudlák were motivated by work in the context of Kreisel's conjecture [10]: If Peano arithmetic (PA) proves $A(S^n0)$ with a proof of $\leq k$ lines for all n then PA proves $\forall x A(x)$. This well-known conjecture gave also rise to work on the k-provability problem: Given a first-order formula F and an integer k, does there exist a proof of F consisting of $\leq k$ lines. Buss [1] showed that the k-provability problem for LK is undecidable by means of *logical flow graphs*. A logical flow graph is a directed graph tracing the influence of formulas in a proof. In contrast to proof skeletons, a logical flow graph is a labelled graph not containing any inferences.

A *proof net* is a proof without the specific order of rule applications. For a proof net we only consider causal dependencies between inferences, that is, two inferences which are not dependent on each other can be seen as parallel. Girard [6] introduced proof nets in the context of normalization considerations for linear logic. In linear logic, proofs are seen as programs which involves normal form theorems and considerations about the equality of normal forms. Later, Robinson [12] extended proof nets to classical logic.

The above-mentioned abstract proof structures, viz. proof skeletons, logical flow graphs and proof nets, were used and studied independently from each other, and therefore no work about their relationship to each other exists so far. We want to take a first step in this direction.

The two main issues we want to address here are: First, the introduction of a uniform framework for abstract proof structures. Second, relating these abstract proof structures by means of the before-mentioned uniform framework. Both in the context of first-order logic and a sequent calculus LK.

The former task is about introducing a tuple-based representation of proofs. From this point of view, a proof is a mathematical object consisting of elements, such that every element represents a specific type of explicit information contained in a proof. The advantage of this separation of information is the effortless representation of abstractions. To be more precise, we introduce a tuple $\langle F, I, \preccurlyeq, \lceil\cdot\rceil\rangle$ which represents a proof provided it obeys certain conditions. However, the contribution of $\lceil\cdot\rceil$ is the integration of formulas, that is, we have a separation of the proof and the formulas involved. By removing this element from the tuple we get a proof skeleton $\langle F, I, \preccurlyeq\rangle$.

The second task is about defining transformations and functions from one proof structure to another. We will show that there do not exist functions between the above-mentioned abstract proof structures, for instance, in general the proof skeleton of a proof P cannot be transformed into the proof net of P. However, we can define a so-called *proof net skeleton* which is an abstraction of both, proof skeleton and proof net. In fact, we can define such an abstraction for all pairs of proof structures if we consider a variant of LK with annotated weakening rules. In this sequent calculus we annotate the weakening rules with so-called *formula skeletons* of the weakening formulas, which can be seen as the syntax tree of a formula without the atomic formulas. The existence of a base abstraction for all pairs of proof structures implies that there exists a structure such that every abstract proof structure in our considerations can be reduced to it. We call this structure *reduced atomic flow graph skeleton* which is a topologically reduced version of an atomic flow graph without formulas.

This framework of proof structures can also be characterized as a partially ordered set of equivalence relations which are defined on proofs, and where the partial order is given by a refinement relation: informally, $\sim_f$ is finer than $\sim_g$ if $X \sim_f Y$ implies $X \sim_g Y$ for all X and Y. By using this framework, it is possible to put newly defined equivalence relations on a map, that is, relate it to other already defined relations.

Another question we will answer is, how many proofs induce the same abstract proof structure. We distinguish between finitely and infinitely many, and define equivalence classes based on the previously described abstractions and transformations. In general, there infinitely many proofs having the same proof skeleton. For proof nets and atomic flow graphs we show that there exists a finite number of corresponding proofs.

The last part of this thesis is an application of the uniform framework to the generalization of an algorithm defined by Krajíček and Pudlák [9]. The authors of [9] reduced the existence of a proof having a given proof skeleton and a given end-sequent to a unification problem with certain restrictions. By deriving bounds on the maximal depth of unified terms from an algorithm for finding most general unifiers, Krajíček and Pudlák gave bounds on the depth of terms in a proof. This also proves the existence of a most general proof with a cut-free skeleton. We aim at generalizing this result to proof nets. We will see that we cannot retain the reduction from the existence of a proof with a given proof net skeleton and a given end-sequent to the unification problem. However, we show that the existence of a most general proof net with a given cut-free net skeleton still holds, and that we can still apply the before-mentioned bounds on the depth of terms in a proof net.

This thesis is organized as follows: After introducing our sequent calculus LK and giving a survey of proof skeletons, proof nets and atomic flow graphs in Chapter 2, we will start by defining the tuple-based representation of LK-proofs in Chapter 3, in particular in Section 3.1. Based on that we define the abstractions in the same tuple-based manner in Section 3.2.

Chapter 4 is about the relationship between the introduced proof structures. First, we show the existence or non-existence of certain functions obeying commutation in Section 4.1. One example of the non-existence of a function is from proof net skeletons to the reduced atomic flow graph skeleton. We circumvent this fact in Section 4.2 by defining a variant of LK with annotated weakening rules. Furthermore, we show that these relations can be characterized as a partially ordered set of equivalence relations. In Section 4.3 we define equivalence relations based on the abstractions and investigate the cardinalities of their classes.

In Chapter 5 we introduce basic notions of unification in Section 5.1 before we generalize the algorithm of Krajíček and Pudlák [9] in Section 5.2.

CHAPTER 2

Preliminaries

2.1 First-order predicate calculus

2.1.1 Syntax

Definition 1. The language of first-order logic is comprised of the following symbols.

1. Constants:

   1.1. Constant symbols: $k_0, k_1, k_2, \ldots$

   1.2. Function symbols with n arguments: $f_0^n, f_1^n, \ldots, g_0^n, g_1^n, \ldots, h_0^n, h_1^n, \ldots$

   1.3. Predicate symbols with n arguments: $P_0^n, P_1^n, \ldots, Q_0^n, Q_1^n, \ldots, R_0^n, R_1^n, \ldots$

   If the number of arguments is clear from the context, we omit the superscript of the function and predicate symbols.

2. Variables:

   2.1. Free variables: $a_0, a_1, \ldots, b_0, b_1, \ldots, c_0, c_1, \ldots$

   2.2. Bound variables: $x_0, x_1, \ldots, y_0, y_1, \ldots, z_0, z_1, \ldots$

3. Logical symbols:

   3.1. Propositional: $\neg$ (not), $\wedge$ (and), $\vee$ (or) and $\supset$ (implies)

   3.2. Quantifiers: $\forall$ (for all) and $\exists$ (exists)

4. Auxiliary symbols: (,) (parentheses) and , (comma)

Definition 2 (Term)**.** We define the set of terms $\mathcal{T}$ as follows.

1. Every constant is a term.

2. Every free variable is a term.
3. If f is an n-ary function symbol and $t_1, \ldots, t_n$ are terms, then $f(t_1, \ldots, t_n)$ is a term.

Definition 3 (Formula)**.** Let P be an n-ary predicate symbol and $t_1, \ldots, t_n$ terms, then $P(t_1, \ldots, t_n)$ is an *atomic formula.* The set of atomic formulas is denoted as $\mathcal{A}$. Furthermore, we define the set of formulas $\mathcal{F}$ as follows.

1. Every atomic formula is a formula.
2. If F is a formula, then $(\neg F)$ is a formula.
3. If F and G are formulas, then $(F \wedge G)$, $(F \vee G)$ and $(F \supset G)$ are formulas.
4. If F is a formula and x a variable, then $\forall x F$ and $\exists x F$ are formulas.

2.1.2 Sequent calculus

Definition 4 (Indexed formula, sequent)**.** A pair (F, i) is called *indexed formula* where F is a formula and $i \in \mathbb{N}$. Furthermore, we call $\Gamma \vdash \Delta$ an *indexed sequent* where Γ and Δ are sets of indexed formulas s.t. all indices in $\Gamma \vdash \Delta$ are pairwise distinct.

Given an (indexed) sequent $\Gamma \vdash \Delta$, we call Γ the *antecedent* and Δ the *succedent* of the sequent. Furthermore, $\Gamma \vdash \Delta$ is an *axiom* if it is of the form $F^i \vdash F^j$ s.t. F is an atomic formula and $i \neq j$. In addition to axioms the calculus *LK* exhibits the following rules.

1. Structural rules:

1.1. Weakening:

$$\frac{\Gamma \vdash \Delta}{(F, k), \Gamma \vdash \Delta}\ \mathrm{W_L} \qquad\qquad \frac{\Gamma \vdash \Delta}{\Gamma \vdash \Delta, (F, k)}\ \mathrm{W_R}$$

1.2. Contraction:

$$\frac{(F, i), (F, j), \Gamma \vdash \Delta}{(F, k), \Gamma \vdash \Delta}\ \mathrm{C_L} \qquad\qquad \frac{\Gamma \vdash \Delta, (F, i), (F, j)}{\Gamma \vdash \Delta, (F, k)}\ \mathrm{C_R}$$

1.3. Cut:

$$\frac{\Gamma \vdash \Delta, (F, i) \qquad (F, j), \Gamma' \vdash \Delta'}{\Gamma, \Gamma' \vdash \Delta, \Delta'}\ \mathrm{CUT}$$

The formulas (F, i) and (F, j) in the upper sequents of the rules above are called *auxiliary formulas*, the (F, k) in the lower sequent is called *principal formula*, and the formulas in Γ, Γ', Δ and Δ' are called *side formulas* of the respective inference.

The formulas (F, i) and (F, j) in CUT are called *cut-formulas*. Note that, the weakening rules do not have auxiliary formulas, whereas the cut rule does not have a principal formula.

2. Logical rules:

2.1. Propositional rules:

$$\frac{\mathit{\Gamma} \vdash \mathit{\Delta}, (F, i)}{(\neg F, k), \mathit{\Gamma} \vdash \mathit{\Delta}} \neg_{\mathrm{L}} \qquad \frac{(F, i), \mathit{\Gamma} \vdash \mathit{\Delta}}{\mathit{\Gamma} \vdash \mathit{\Delta}, (\neg F, k)} \neg_{\mathrm{R}}$$

$$\frac{(F, i), (G, j), \mathit{\Gamma} \vdash \mathit{\Delta}}{(F \wedge G, k), \mathit{\Gamma} \vdash \mathit{\Delta}} \wedge_{\mathrm{L}} \qquad \frac{\mathit{\Gamma} \vdash \mathit{\Delta}, (F, i) \qquad \mathit{\Gamma}' \vdash \mathit{\Delta}', (G, j)}{\mathit{\Gamma}, \mathit{\Gamma}' \vdash \mathit{\Delta}, \mathit{\Delta}', (F \wedge G, k)} \wedge_{\mathrm{R}}$$

$$\frac{(F, i), \mathit{\Gamma} \vdash \mathit{\Delta} \qquad (G, j), \mathit{\Gamma}' \vdash \mathit{\Delta}'}{(F \vee G, k), \mathit{\Gamma}, \mathit{\Gamma}' \vdash \mathit{\Delta}, \mathit{\Delta}'} \vee_{\mathrm{L}} \qquad \frac{\mathit{\Gamma} \vdash \mathit{\Delta}, (F, i), (G, j)}{\mathit{\Gamma} \vdash \mathit{\Delta}, (F \vee G, k)} \vee_{\mathrm{R}}$$

$$\frac{\mathit{\Gamma} \vdash \mathit{\Delta}, (F, i) \qquad (G, j), \mathit{\Gamma}' \vdash \mathit{\Delta}'}{(F \supset G, k), \mathit{\Gamma}, \mathit{\Gamma}' \vdash \mathit{\Delta}, \mathit{\Delta}'} \supset_{\mathrm{L}} \qquad \frac{(F, i), \mathit{\Gamma} \vdash \mathit{\Delta}, (G, j)}{\mathit{\Gamma} \vdash \mathit{\Delta}, (F \supset G, k)} \supset_{\mathrm{R}}$$

The formulas (F, i) and (F, j) in the rules above are the auxiliary formulas, whereas the formulas in the lower sequent are the principal formulas of the respective inference. The formulas in $\mathit{\Gamma}, \mathit{\Gamma}', \mathit{\Delta}$ and $\mathit{\Delta}'$ are the side formulas.

2.2. Quantifier rules:

$$\frac{(F(t), i), \mathit{\Gamma} \vdash \mathit{\Delta}}{(\forall x F(x), k), \mathit{\Gamma} \vdash \mathit{\Delta}} \forall_{\mathrm{L}} \qquad \frac{\mathit{\Gamma} \vdash \mathit{\Delta}, (F(a), i)}{\mathit{\Gamma} \vdash \mathit{\Delta}, (\forall x F(x), k)} \forall_{\mathrm{R}}$$

$$\frac{(F(a), i), \mathit{\Gamma} \vdash \mathit{\Delta}}{(\exists x F(x), k), \mathit{\Gamma} \vdash \mathit{\Delta}} \exists_{\mathrm{L}} \qquad \frac{\mathit{\Gamma} \vdash \mathit{\Delta}, (F(t), i)}{\mathit{\Gamma} \vdash \mathit{\Delta}, (\exists x F(x), k)} \exists_{\mathrm{R}}$$

Again, the formulas in the upper sequent are the auxiliary formulas, the formulas in the lower sequent are the principal formulas, and the formulas in $\mathit{\Gamma}$ and $\mathit{\Delta}$ are the side formulas. The variable a in $\forall_{\mathrm{R}}$ and $\exists_{\mathrm{L}}$ is called *eigenvariable* and must not occur in the lower sequent of the inference.

We call an *LK*-rule *binary* if it has two upper sequents and *unary* if it has one upper sequent.

The indices of the auxiliary and principal formulas in the rules above are meant to be different, e.g. i, j, k are pairwise distinct in C_{L}. That is, all auxiliary and principal formulas of an inference have distinct indices, whereas the indices of the side formulas are propagated from the upper to the lower sequent. Furthermore, k does not occur in the subproofs above, and for binary rules we have that the set of indices occurring in the subproofs are disjoint.

Remark 1. Note that having indexed formulas and indexed sequents allows us to relinquish the exchange rules usually defined in other variants of LK.

For simplicity we use F^i instead of (F, i) from now on.

Definition 5 (LK-proof)**.** An *LK-proof* is directed tree whose vertices are labelled with indexed sequents, and edges are labelled with rules. Additionally, the leaves of the tree must be axioms, and the edges are oriented towards the axioms. An LK-proof where no edge is labelled with CUT is called *cut-free.*

2.2 A survey of abstract proof structures

In this section we give an overview of the proof structures we are investigating. Therefore we will have a look at the basic idea, motivation and the usage of *proof skeletons*, *proof nets* and *atomic flow graphs.*

Proof skeletons and atomic flow graphs were introduced in the context of the k-provability problem: *Given a first-order formula F and an integer k, does F has a proof in less than or equal k lines?* Proof nets were introduced for normalization studies in linear logic by Girard.

To illustrate these proof structures we will use the following LK-proof. We will also refer to this proof for illustrating the concepts we are introducing in the upcoming sections.

$$\supset_{\mathrm{L}}\dfrac{\neg_{\mathrm{L}}\dfrac{\neg_{\mathrm{R}}\dfrac{A^1 \vdash A^2}{\vdash A^2, \neg A^3}}{\neg A^4 \vdash \neg A^3} \qquad B^5 \vdash B^6}{\neg A^4, \neg A \supset B^7 \vdash B^6} \tag{$\mathit{\Pi}_\sharp$}$$

Furthermore, we distinguish between two different measurements of the size of a proof. First, the number of lines in a proof, also called *length* of a proof. Second, the actual *size* of a proof meaning the number of symbols involved.

2.2.1 Proof skeletons

A proof skeleton is an LK-proof without formulas. That is, given an LK-proof, the actual formulas involved are removed whereas the structure and the applied LK-rules remain. Furthermore, information about the active formulas is kept which means for our LK-calculus that we keep the indices to know the active formula occurrences for each LK-inference. Hence, we get the following proof skeleton for $\mathit{\Pi}_\sharp$.

$$\supset_{\mathrm{L}} \dfrac{\neg_{\mathrm{L}} \dfrac{\neg_{\mathrm{R}} \dfrac{.^{1} \vdash .^{2}}{\vdash .^{2}, .^{3}}}{.^{4} \vdash .^{3}} \qquad .^{5} \vdash .^{6}}{.^{4}, .^{7} \vdash .^{6}}$$

Proof skeletons were introduced in the context of the k-provability problem by Orevkov [11]. Orevkov showed for a large class of Hilbert-type calculi that it is undecidable whether a given formula has a proof with a given proof skeleton.

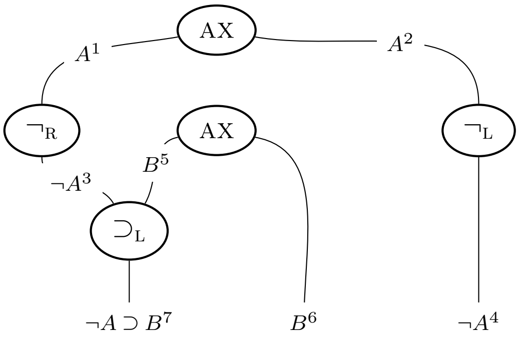
Despite the fact that a proof skeleton has an infinite number of corresponding proofs, Krajíček and Pudlák [9] showed that given an end-sequent $\Gamma \vdash \Delta$ and a skeleton S of a cut-free proof, there is a most general proof of $\Gamma \vdash \Delta$ with skeleton S such that every proof of $\Gamma \vdash \Delta$ with S is an instance of it. The main purpose, however, was to give bounds on the minimal size of proofs. More specifically, given a sequent $\Gamma \vdash \Delta$ of size m with an LK-proof having k lines, the aim was to find bounds on the size of a minimal proof of $\Gamma \vdash \Delta$ depending on m and k.

By a reduction to the first-order unification problem an exponential bound in $k + m$ for cut-free proofs was proven, i.e. given a cut-free proof skeleton of $\Gamma \vdash \Delta$ of size m with k proof lines, there exists a proof of $\Gamma \vdash \Delta$ whose size is bounded by an exponential in $k+m$. The bound is obtained by an estimate of the depth of the most general unifier. In general, i.e. for proofs with cut, Krajíček and Pudlák showed a primitive recursive bound in $k + m$, as a cut-free proof has to be obtained from the proof first. It is unknown, whether there exists a fixed times iterated exponential bound.

Furthermore, the authors of [9] investigated the problem whether a given sequent has a proof with a given proof skeleton. This problem was shown to be undecidable by a reduction from the second-order unification problem which was proven to be undecidable in [8].

2.2.2 Proof nets

A proof net is a geometric representation of a proof which enables a view of proof identity closer to the intuitive meaning. To be more precise, proof nets are immune to a certain kind of rule permutations, i.e. proofs which only differ by simple rule transpositions exhibit the same proof net. For instance, transposing the first and the second inference of $\Pi_\sharp$ yields the following proof net, so does $\Pi_\sharp$ itself.



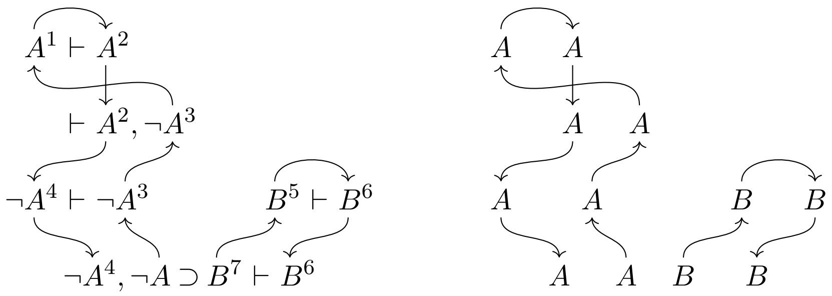
Rule applications in sequent calculi proofs have to be sequentialized which has the disadvantage that two proofs might be formally different but intuitively the same. In certain aspects of proof theory it is desirable to have a device which neglects this differences. In case of (a suitable selection of logical connectives in) intuitionistic logic, natural deduction plays this role w.r.t. a sequent calculus, where natural deduction has a strong correspondence to the simply typed lambda calculus which is interesting from a computational point of view.

Girard [6] was faced with the same problem concerning linear logic. In the context of this considerations he developed the concept of proof nets for the multiplicative fragment of linear logic (MLL). That is, proof nets play the same role for linear logic as simply typed lambda terms do for intuitionistic logic. The work of Girard also involves a normal form theorem. Normalization in the context of proof nets is the process of eliminating cuts, and he showed that proof nets are strongly normalizing. Furthermore, the reduction satisfies the Church-Rosser property, i.e. the order of cuts to eliminate does not matter.

Such a device is certainly desirable for classical logic as well. Robinson [12] extended the notion of proof net to classical logic. Unlike for MLL proof nets do not behave as satisfactory for classical logic. In contrast to MLL, sequent calculi for classical logic contain contraction and weakening rules which yields the question of where to attach the weakening rules (cf. Straßburger [13]).

2.2.3 Atomic flow graphs

An atomic flow graph (AFG) is a special case of a *logical flow graph*, which is a directed graph whose set of vertices consists of all subformula occurrences in the proof, whereas the edges track the influence of the subformulas. As the name suggests, an atomic flow graph is a subgraph of the logical flow graph containing only the atomic subformulas. The following graph on the right is the atomic flow graph of $\Pi_\sharp$.



Buss [1] introduced logical flow graphs in the context of the k-provability problem. In particular, the author showed that the k-provability problem is undecidable for LK. To do so, he reduced the *second-order unification problem with partial substitution* to the k-provability problem. The undecidability of the second-order unification problem was proven in [8] and was extended to partial substitution in [1], which yields the undecidability of the k-provability problem.

Logical flow graphs, in particular atomic flow graphs, were used by Carbone and Semmes [4] for investigating the geometric aspects of cut-elimination. In particular, they studied the patterns emerging during the process of cut-elimination. The motivation in doing so lies in the fact that proofs can be made essentially smaller by introducing cuts. Hence, these cuts must capture some essential patterns of the underlying proof structure. Let Π be a proof and Π' a cut-free version of it obtained by a particular method for cut-elimination. In [2], Carbone showed that certain subgraphs of the AFG of Π' are contained in the AFG of Π, and both subgraphs correspond to the same sequent. This locality property is then used to derive results about the complexity of interpolants. Furthermore, it is shown in [2] that cut-free proofs have acyclic atomic flow graphs, and the same holds for contraction-free proofs. In other words, only proofs containing both, cut and contraction, may contain cycles. In [3] it was shown by Carbone that there exist arbitrarily complex non-oriented graphs G such that G can be embedded in a proof, i.e. G is topologically contained in the AFG of a proof.

CHAPTER 3

A Uniform Framework

In order to compare abstract proof structures, a suitable representation of proofs and their abstractions is needed. Therefore we introduce a tuple-based representation of proofs where abstractions can be performed and denoted effortless.

3.1 Proof

Making the abstractions convenient in terms of notation, requires a separation of the information contained in an LK-proof. On the one hand we have to distinguish formula occurrences and formulas for dealing with proof skeletons, on the other hand we have to distinguish between inferences and their specific order of application to be able to characterize proof nets. The former can be established by distinguishing formulas and their indices.

Definition 6 (Formula index)**.** Let F^i be an indexed formula. Then i is the *formula index* (or *index* for short) of F^i.

Definition 7 (Inference)**.** Let $\mathbf{x}_i = x_{i,1}, \ldots, x_{i,k_i}$ and $\mathbf{y} = y_1, \ldots, y_l$ be pairwise distinct indices for $i = 1, \ldots, n$. Then $r(\mathbf{x}_1; \ldots; \mathbf{x}_n; \mathbf{y})$ is an *inference* where r is an n-ary LK-rule with $k = \Sigma_{i=1}^{n} k_i$ auxiliary formulas and l principal formulas.

Example 1. Translating the LK-proof $\Pi_\sharp$ we get the following set of inferences where the indices used in the inferences are the formula indices of the formulas in the proof.

$$\begin{array}{cc}
\textsc{ax}(\varnothing; 1, 2) & \\
\neg_{\textsc{r}}(1; 3) & \\
\neg_{\textsc{l}}(2; 4) & \textsc{ax}(\varnothing; 5, 6) \\
\multicolumn{2}{c}{\supset_{\textsc{l}}(3; 5; 7)}
\end{array}$$

Note that in our sequent calculus LK we have $k \in \{0, 1, 2\}$ and $l \in \{0, 1\}$. Given an inference $\iota = r(\mathbf{x}_1; \ldots; \mathbf{x}_n; \mathbf{y})$, $\mathbf{x}_1; \ldots; \mathbf{x}_n$ correspond to the auxiliary formulas of r and $\mathbf{y}$ to the principal formulas. In particular the formulas in $\mathbf{x}_i$ are the formulas contained in the i-th upper sequent denoted by $Aux_i(\iota) = \{\mathbf{x}_i\}$. Furthermore, $Aux(\iota) = \bigcup_{i=1}^{n} Aux_i(\iota)$ and $Princ(\iota) = \{\mathbf{y}\}$ denote the auxiliary and principal formulas of ι respectively.

Note that every inference $r(\mathbf{x}_1; \ldots; \mathbf{x}_n; \mathbf{y})$ induces an LK-inference with upper sequents $S_1, \ldots, S_n$ and one lower sequent S. Furthermore, every LK-rule r uniquely determines whether a given auxiliary or principal formula is in the antecedent or the succedent of a sequent. Thus, we have that r uniquely determines whether $x \in \{\mathbf{x}_i\}$ is in the antecedent or the succedent of S_i, and whether $y \in \{\mathbf{y}\}$ is in the antecedent or succedent of S.

Accordingly we can define a notion of polarity for formula indices. We say an index x occurs *negatively* (*positively*) in an inference ι if x occurs in the antecedent (succedent) of the corresponding sequent involved in ι.

Definition 8 (Well-polarization)**.** Given a set of inferences I, we call I *well-polarized* if there exists no formula index x such that x occurs positively (negatively) in ι_1 and negatively (positively) in ι_2 for $\iota_1, \iota_2 \in I$.

Intuitively, well-polarization ensures that a formula cannot jump from the antecedent to the succedent or vice versa. This concept assures the applicability of inferences.

Example 2. Assume we have a set of inferences I containing two inferences $\textsc{ax}(\varnothing; 1, 2)$ and $\neg_{\textsc{l}}(1; 3)$. On the one hand, the LK-rule $\textsc{ax}$ expects the index 1 to be in the antecedent of the sequent. On the other hand, $\neg_{\textsc{l}}$ expects the index 1 to be in the succedent of the sequent. That is, in the former case the index 1 occurs negatively, whereas in the latter case it is positive, i.e. the polarities do not coincide. In conclusion, I is not well-polarized.

Definition 9. Given a partial order $\preccurlyeq$ over some set P, then $\preccurlyeq$ is a *tree order* over P if

(i) $\preccurlyeq$ has a smallest element r (called *root*), and

(ii) for each $y \in P$ where $y \succcurlyeq r$ there exists a unique path from r to y.

Since it is possible to transpose independent rules in an LK-proof without changing the end-sequent, an order of the inferences is crucial to uniquely determine an LK-proof. As defined in Definition 5, a proof is a tree. Hence, a set of inferences I must be partially ordered by a tree order $\preccurlyeq$, in particular $\preccurlyeq$ must be an inference order (see below). As usual, we write $\iota \prec \kappa$ if $\iota \preccurlyeq \kappa$ and $\iota \neq \kappa$, and $\iota \prec^1 \kappa$ if $\iota \prec \kappa$ and there exists no κ' s.t. $\iota \prec \kappa' \prec \kappa$.

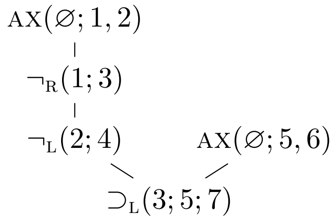
Definition 10 (Inference order)**.** Given a set of inferences I, a tree order $\preccurlyeq$ over I is called *inference order* over I if

(i) for every n-ary inference $\iota = r(\mathbf{x}_1; \ldots; \mathbf{x}_n; \mathbf{y}) \in I$ there exist exactly n distinct inferences $\kappa_1, \ldots, \kappa_n$ s.t. $\kappa_1, \ldots, \kappa_n \succ^{-1} \iota$ and for every $x \in \{\mathbf{x}_i\}$ there exists precisely one inference $\kappa_i' \succcurlyeq \kappa_i$ where $x \in Princ(\kappa_i')$, and

(ii) for every inference $\iota \in I$ and $x \in Princ(\iota)$ there exists at most one $\kappa \in I$ s.t. $\kappa \prec \iota$ and $x \in Aux(\kappa)$.

An inference order ensures that a formula occurrence is introduced by some inference, i.e. it is a principal formula occurrence of some inference. Furthermore, a formula occurrence cannot be auxiliary for more than one inference.

Example 3. The order of inferences given by the *LK*-proof $\Pi_\sharp$ yields an inference order. Note that the minimal element of the order is the last inference in $\Pi_\sharp$ and all maximal elements are axioms.


$\textsc{ax}(\varnothing; 1, 2)$
$\neg_{\mathrm{R}}(1; 3)$
$\neg_{\mathrm{L}}(2; 4)$
$\textsc{ax}(\varnothing; 5, 6)$
$\supset_{\mathrm{L}}(3; 5; 7)$

So far, we only used formula indices, that is, we require the integration of actual formulas. Therefore, a mapping $\lceil\cdot\rceil : F \to \mathcal{F}$ from indices to formulas is used.

Definition 11 (Formula map)**.** A map $\lceil\cdot\rceil : F \to \mathcal{F}$ is called *formula map* where F is a set of indices and $\mathcal{F}$ is a set of formulas.

Before defining the tuple representing an *LK*-proof we have to ensure that for a given set of inferences I the formula map exhibits correct inferences w.r.t. *LK*.

Let $\lceil\cdot\rceil$ be a formula map. For simplicity, we denote the indexed formula $\lceil x\rceil^x$ as $\lceil x\rceil$ where x is an index. Furthermore, for a list of indices $\mathbf{x} = x_1, \ldots, x_n$, we write $\lceil\mathbf{x}\rceil$ instead of $\lceil x_1\rceil, \ldots, \lceil x_n\rceil$.

Definition 12 (Correct map)**.** We call a formula map $\lceil\cdot\rceil$ *correct* w.r.t. an inference ι if $\lceil\mathbf{y}\rceil$ is the result of applying r to $\lceil\mathbf{x}_1\rceil, \ldots, \lceil\mathbf{x}_n\rceil$ for $\iota = r(\mathbf{x}_1; \ldots; \mathbf{x}_n, \mathbf{y})$. The map $\lceil\cdot\rceil$ is called correct w.r.t. a set I of inferences if $\lceil\cdot\rceil$ is correct w.r.t. κ for all $\kappa \in I$.

Example 4. Let $\iota = \supset_{\mathrm{L}}(3; 5; 7) \in I$ be an inference and let the formula maps $\lceil\cdot\rceil_1$ and $\lceil\cdot\rceil_2$ be defined as follows.

$$\begin{aligned} \lceil 3\rceil_1 &= A & \lceil 3\rceil_2 &= \neg A \\ \lceil 5\rceil_1 &= B & \lceil 5\rceil_2 &= B \\ \lceil 7\rceil_1 &= \neg A \supset B & \lceil 7\rceil_2 &= \neg A \supset B. \end{aligned}$$

Then $\lceil\cdot\rceil_1$ and $\lceil\cdot\rceil_2$ applied to ι translate into

$$\supset_{\mathrm{L}} \frac{\Gamma_1 \vdash \Delta_1, A \qquad B, \Gamma_2 \vdash \Delta_2}{\Gamma_1, \Gamma_2, \neg A \supset B \vdash \Delta_1, \Delta_2} \qquad \text{and} \qquad \supset_{\mathrm{L}} \frac{\Gamma_1 \vdash \Delta_1, \neg A \qquad B, \Gamma_2 \vdash \Delta_2}{\Gamma_1, \Gamma_2, \neg A \supset B \vdash \Delta_1, \Delta_2}$$

respectively. The former is obviously not a valid *LK*-inference, whereas the result of $\lceil\cdot\rceil_2$ yields a valid *LK*-inference. Therefore, $\lceil\cdot\rceil_2$ is correct w.r.t. ι, and $\lceil\cdot\rceil_1$ is not. Consequently, $\lceil\cdot\rceil_1$ is not correct w.r.t. I, and $\lceil\cdot\rceil_2$ is correct w.r.t. I if $\lceil\cdot\rceil_2$ is correct w.r.t. to all other inferences $\kappa \in I$ as well.

The notions and concepts introduced so far, viz. well-polarization, inference order and correct maps, suffice to characterize propositional *LK*-proofs. For first-order *LK*-proofs we have to deal with the eigenvariable condition. Therefore, we restrict the formula map s.t. the eigenvariables in strong quantifier inferences, viz. $\exists_{\mathrm{L}}$ and $\forall_{\mathrm{R}}$, do not occur free in the side formulas of the strong quantifier inference.

In general, side formulas of a given inference ι are recognized as those formulas introduced above ι which are neither auxiliary for any inference above ι nor for ι itself.

Definition 13 (Eigenvariable-preserving map)**.** Let $\langle I, \preccurlyeq\rangle$ be a partially ordered set of inferences and $\lceil\cdot\rceil$ a formula map. We call $\lceil\cdot\rceil$ *eigenvariable-preserving* w.r.t. $\langle I, \preccurlyeq\rangle$ if for every strong quantifier inference $\iota \in I$ where $\{x\} = \mathit{Aux}(\iota)$ and $\lceil x\rceil = F(a)$ the following holds: a does not occur free in $\lceil y\rceil$ for every $y \in \mathit{Princ}(\kappa)$ s.t. $\kappa \succ \iota$ and $y \notin \mathit{Aux}(\lambda)$ for all $\lambda \not\succcurlyeq \iota$.

Example 5. Consider the following proof skeleton and the corresponding inferences, and a formula map $\lceil\cdot\rceil$.

$$\begin{array}{rl}
\hline
.^{1} \vdash .^{2} & \mathrm{AX} \doteq \iota_1 \\
\hline
.^{1} \vdash .^{3} & \exists_{\mathrm{R}} \doteq \iota_2 \\
\hline
.^{4}, .^{1} \vdash .^{3} & \mathrm{W}_{\mathrm{L}} \doteq \iota_3 \\
\hline
.^{4}, .^{5} \vdash .^{3} & \exists_{\mathrm{L}} \doteq \iota_4
\end{array}$$

Now the eigenvariable of the strong quantifier inference ι_4 must not occur free in $\lceil 4\rceil$ and $\lceil 3\rceil$, as the indices 3 are 4 are not auxiliary for any inference $\not\succcurlyeq \iota_4$. In other words, as the indices 1 and 2 are auxiliary for some inference $\not\succcurlyeq \iota_4$, the eigenvariable might occur free in $\lceil 1\rceil$ or $\lceil 2\rceil$.

With the notion of eigenvariable-preserving maps present, we can represent first-order *LK*-proofs as quadruples satisfying certain restrictions.

Definition 14 (Proof)**.** Let F be a set of formula indices, I a set of inferences, $\preccurlyeq$ a partial order over I and $\lceil\cdot\rceil$ a formula map. Then $\langle F, I, \preccurlyeq, \lceil\cdot\rceil\rangle$ is a *proof* if

(i) $F = \bigcup_{\iota \in I} \mathit{Aux}(\iota) \cup \mathit{Princ}(\iota)$,

(ii) I is well-polarized,

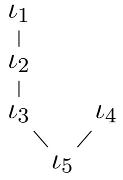
(iii) $\preccurlyeq$ is an inference order and its maximal elements are axioms,

(iv) $\lceil\cdot\rceil$ is correct w.r.t. I, and

(v) $\lceil\cdot\rceil$ is eigenvariable-preserving w.r.t. $\langle I, \preccurlyeq\rangle$.

Example 6. The tuple $P_\sharp = \langle F, I, \preccurlyeq, \lceil\cdot\rceil\rangle$ representing the *LK*-proof $\Pi_\sharp$ is defined as follows where $F = \{1, \ldots, 7\}$ and $I = \{\iota_1, \ldots, \iota_5\}$.

$$\begin{aligned}
\iota_1 &= \textsc{ax}(\varnothing; 1, 2) \\
\iota_2 &= \neg_{\mathrm{R}}(1; 3) \\
\iota_3 &= \neg_{\mathrm{L}}(2; 4) \\
\iota_4 &= \textsc{ax}(\varnothing; 5, 6) \\
\iota_5 &= \supset_{\mathrm{L}}(3; 5; 7)
\end{aligned}$$



$$\begin{aligned}
\lceil 1\rceil = \lceil 2\rceil &= A \\
\lceil 3\rceil = \lceil 4\rceil &= \neg A \\
\lceil 5\rceil = \lceil 6\rceil &= B \\
\lceil 7\rceil &= \neg A \supset B
\end{aligned}$$

We use $\mathscr{P}^{LK}$ and $\mathscr{P}$ to denote the set of *LK*-proofs and proofs respectively. Before continuing with defining the abstractions based on the definition of a proof, it is necessary to show that Definition 14 indeed captures the properties of an *LK*-proof. Therefore, we define maps $\pi^{LK} : \mathscr{P} \to \mathscr{P}^{LK}$ and $\pi : \mathscr{P}^{LK} \to \mathscr{P}$, and by showing that they are inverse to each other we obtain the equivalence of proofs and *LK*-proofs.

Let $S_1 = \Gamma_1 \vdash \Delta_1$ and $S_2 = \Gamma_2 \vdash \Delta_2$ be sequents. Then $S_1 \cup S_2 = \Gamma_1 \cup \Gamma_2 \vdash \Delta_1 \cup \Delta_2$ and $S_1 \setminus S_2 = \Gamma_1 \setminus \Gamma_2 \vdash \Delta_1 \setminus \Delta_2$. Furthermore, let $\iota = r(\mathbf{x}_1; \ldots; \mathbf{x}_n; \mathbf{y})$ be an inference and $\lceil\cdot\rceil : F \to \mathcal{F}$ a formula map. As r uniquely determines whether a given auxiliary or principal formula of ι is in the antecedent or the succedent of a sequent, we can write $S_1 \cup \{\lceil z_1\rceil^{z_1}, \ldots, \lceil z_m\rceil^{z_m}\}$ and $S_1 \setminus \{\lceil z_1\rceil^{z_1}, \ldots, \lceil z_m\rceil^{z_m}\}$ without being ambiguous for $\{z_1, \ldots, z_m\} \subseteq \mathit{Aux}(\iota) \cup \mathit{Princ}(\iota)$.

Transformation 1. Given a proof $\langle F, I, \preccurlyeq, \lceil\cdot\rceil\rangle$, the associated *LK*-proof is constructed inductively as follows:

1. For every axiom $\iota = \textsc{ax}(\varnothing; y_1, y_2) \in I$ create a sequent $S_\iota = \lceil y_1\rceil \vdash \lceil y_2\rceil$.
2. Let $\iota = r(\mathbf{x}_1; \ldots; \mathbf{x}_n; \mathbf{y}) \in I$ be an n-ary inference where $\{\mathbf{x}\} = \mathit{Aux}(\iota)$, and $\kappa_1, \ldots, \kappa_n \succ^1 \iota$. Then $S_\iota = ((S_{\kappa_1} \cup \cdots \cup S_{\kappa_n}) \setminus \{\lceil\mathbf{x}\rceil\}) \cup \{\lceil\mathbf{y}\rceil\}$ is the lower sequent of r and $S_{\kappa_1}, \ldots, S_{\kappa_n}$ are the upper sequents.

Transformation 2. Given an *LK*-proof Π, the associated proof $\langle F, I, \preccurlyeq, \lceil\cdot\rceil\rangle$ is constructed inductively as follows:

1. For every initial sequent $S = A^{y_1} \vdash A^{y_2}$ in Π create an inference $\iota_S = \textsc{ax}(\varnothing; y_1, y_2)$, and let $\lceil y_1\rceil = \lceil y_2\rceil = A$.
2. For every n-ary *LK*-inference $\; r\,\dfrac{S_1 \qquad \ldots \qquad S_n}{S}\;$:
   Let $A_{i,1}^{x_{i,1}}, \ldots, A_{i,k_i}^{x_{i,k_i}}$ be the indexed formulas in $S_i \setminus S$ and $B_1^{y_1}, \ldots, B_l^{y_l}$ the indexed formulas in $S \setminus (S_1 \cup \cdots \cup S_n)$. Then create an inference $\iota_S = r(\mathbf{x}_1; \ldots; \mathbf{x}_n; \mathbf{y})$, and let $\lceil x_{i,1}\rceil = A_{i,1}, \ldots, \lceil x_{i,k_i}\rceil = A_{i,k_i}$ and $\lceil y_j\rceil = B_j$ for $1 \leq i \leq n$ and $1 \leq j \leq l$.

3. Let $\preccurlyeq$ be the reflexive, transitive closure of $\prec^1$ where $\prec^1$ is defined as follows: $\iota_{S_2} \prec^1 \iota_{S_1}$ iff there exists an *LK*-inference in $\varPi$ s.t. S_1 is an upper sequent and S_2 the lower sequent.

Next, we have to show that Transformations 1 and 2 indeed define maps from $\mathscr{P}$ to $\mathscr{P}^{LK}$ and vice versa.

Lemma 1. *Transformation 1 defines a map $\pi^{LK} : \mathscr{P} \to \mathscr{P}^{LK}$.*

Proof. Let $P = \langle F, I, \preccurlyeq, \lceil\cdot\rceil\rangle$ be a proof and $\pi^{LK}(P) = \varPi$. We prove by induction on the number of inferences in I that $\varPi$ indeed is an *LK*-proof.

Case 1. I contains a single inference of the form $\textsc{ax}(\varnothing; x_1, x_2)$. As $\lceil\cdot\rceil$ is correct w.r.t. I, $\pi^{LK}(P)$ yields an *LK*-proof of the form $A^{x_1} \vdash A^{x_2}$.

Case 2. Assume the smallest element w.r.t. $\preccurlyeq$ is an n-ary inference $\iota = r(\mathbf{x}_1; \ldots; \mathbf{x}_n; \mathbf{y})$ s.t. $\kappa_1, \ldots, \kappa_n \succ^1 \iota$. By the induction hypothesis there exist *LK*-proofs $\varPi_1, \ldots, \varPi_n$ with end-sequents $S_{\kappa_1}, \ldots, S_{\kappa_n}$. As $\preccurlyeq$ is an inference order, in particular because of Definition 10 (i) and (ii), we have that $\lceil\mathbf{x}_i\rceil$ is contained in S_{κ_i}. Furthermore, as I is well-polarized, every $x \in \{\mathbf{x}_i\}$ is on the side of the sequent S_{κ_i} where r expects it to be, and r is by definition an n-ary *LK*-rule, i.e. r can be applied. As $\lceil\cdot\rceil$ is correct w.r.t. I we get an *LK*-inference $r\,\dfrac{S_{\kappa_1} \quad \ldots \quad S_{\kappa_n}}{S_\iota}$.

Furthermore, if ι is a strong quantifier inference. Then ι is of the form $r(x; y)$ and the last *LK*-inference in $\varPi$ is of the form $\dfrac{S_\kappa}{S_\iota}$ where $\kappa \succ^1 \iota$. Let $\lceil z\rceil$ be an arbitrary formula in S_κ where $z \neq x$. Then there exists an inference $\lambda \succcurlyeq \kappa$ s.t. $z \in \mathit{Princ}(\lambda)$. As $\lceil z\rceil$ is contained in S_κ there exists no $\kappa' \succcurlyeq \kappa$ s.t. $z \in \mathit{Aux}(\kappa')$. Therefore, as $\lceil\cdot\rceil$ is eigenvariable-preserving w.r.t. $\langle I, \preccurlyeq\rangle$, the eigenvariable of ι does not occur free in $\lceil z\rceil$.

Eventually, as $\preccurlyeq$ is a tree order, we get a tree-shaped *LK*-proof $\pi^{LK}(P)$. □

Lemma 2. *Transformation 2 defines a map $\pi : \mathscr{P}^{LK} \to \mathscr{P}$.*

Proof. Analogous to the previous lemma, we have to show that Transformation 2 indeed constructs a proof. Let $\varPi$ be an *LK*-proof and $\pi(\varPi) = \langle F, I, \preccurlyeq, \lceil\cdot\rceil\rangle$.

Note that, due to the definition of the *LK*-rules, no indexed formula changes its polarity within a proof. As a consequence, and by definition of Transformation 2, no formula occurrence in $\pi(\varPi)$ changes its polarity, i.e. I is well-polarized.

As $\varPi$ is a tree and by definition of $\preccurlyeq$ it follows that $\preccurlyeq$ is a tree order. Furthermore, if a formula F^i occurs as an auxiliary formula of some *LK*-inference, then F^i does not occur below that *LK*-inference, i.e. Definition 10 (ii) is satisfied. Now let J be an n-ary *LK*-inference and F^j an auxiliary formula of J contained in the i-th upper sequent S_i. Then, by definition of *LK*, there exists precisely one *LK*-inference J' above S_i s.t. F^j is a principal formula of J'. Consequently, and by definition of Transformation 2, condition (i) of Definition 10 is satisfied.

Furthermore, by inspection of Transformation 2, it is clear that π yields a mapping $\lceil\cdot\rceil : F \to \mathcal{F}$ which is correct w.r.t. I.

Now let $\iota_S \in I$ be a strong quantifier inference obtained by Transformation 2. Furthermore let $y \in Princ(\lambda)$ for some $\lambda \succcurlyeq \iota_S$. If there does not exist an inference $\kappa \succcurlyeq \iota_S$ s.t. $y \in Aux(\kappa)$, then $\lceil y \rceil$ has to be contained in S. Since Π satisfies the eigenvariable condition, the eigenvariable of ι_S is not contained free in $\lceil y \rceil$. Hence, $\lceil \cdot \rceil$ is eigenvariable-preserving w.r.t. $\langle I, \preccurlyeq \rangle$. □

Theorem 1. *π is the inverse of π^{LK}, i.e. $\pi^{LK}(\pi(\Pi)) = \Pi$.*

Proof. We proceed by induction on the number of LK-inferences in Π.

Case 1. Π is of the form $A^{y_1} \vdash A^{y_2}$. Then $\pi(\Pi) = \langle \{y_1, y_2\}, \{\iota = \text{AX}(\varnothing; y_1, y_2)\}, \{(\iota, \iota)\}, \lceil \cdot \rceil \} \rangle$ where $\lceil y_1 \rceil = A$ and $\lceil y_2 \rceil = A$. Therefore $\pi^{LK}(\pi(\Pi))$ consists of a single axiom $A^{y_1} \vdash A^{y_2}$.

Case 2. Assume the last LK-inference in Π is of the following form:

$$r \, \frac{S_1 \qquad \dots \qquad S_n}{S}$$

By the induction hypothesis, each subproof with end-sequent $S_1, \dots, S_n$ retranslates to itself. It remains to show that the last LK-inference is reconstructed. Let $\bar{S} = S_1 \cup \cdots \cup S_n$. By definition of π, $\iota_S = r(\mathbf{x}_1; \dots; \mathbf{x}_n; \mathbf{y})$ s.t. $\lceil \mathbf{x}_i \rceil$ are the formulas in $S_i \setminus S$ and $\lceil \mathbf{y} \rceil$ are the formulas in $S \setminus \bar{S}$. Let $\bar{X} = \{\lceil \mathbf{x}_1 \rceil\} \cup \cdots \cup \{\lceil \mathbf{x}_n \rceil\}$, then

$$\bar{X} = (S_1 \setminus S) \cup \cdots \cup (S_n \setminus S) = \bar{S} \setminus S.$$

Putting pieces together we get $S_{\iota_S} = S$:

$$\begin{aligned} S_{\iota_S} &= ((S_{\iota_{S_1}} \cup \cdots \cup S_{\iota_{S_n}}) \setminus \bar{X}) \cup \{\lceil \mathbf{y} \rceil\} && \text{(by definition of } \pi^{LK}) \\ &= (\bar{S} \setminus \bar{X}) \cup \{\lceil \mathbf{y} \rceil\} && \text{(by induction hypothesis } S_i = S_{\iota_{S_i}}) \\ &= (\bar{S} \setminus (\bar{S} \setminus S)) \cup (S \setminus \bar{S}) && \text{(by replacing } \bar{X} \text{ and } \{\lceil \mathbf{y} \rceil\}) \\ &= S \end{aligned}$$

□

Before establishing the equivalence of LK-proofs and proofs by proving the next result, we introduce a notion of subproofs analogous to LK-subproofs. As for LK-proofs we can locate subproofs of a proof P by restricting P to a certain set of inferences.

Proposition 1. *Let $P = \langle F, I, \preccurlyeq, \lceil \cdot \rceil \rangle$ be a proof, and let $P\!\restriction_\iota$ be the proof P restricted to the inferences in $\{\kappa \mid \kappa \succcurlyeq \iota\}$. Then $P\!\restriction_\iota$ is a proof for any $\iota \in I$.*

Proof. Conditions (ii), (iii), (iv) and (v) of Definition 14 follow immediately from the fact that P is a proof, i.e. as these conditions hold for P they must hold for $P\!\restriction_\iota$. As F is restricted to indices occurring in some inference $\kappa \succcurlyeq \iota$, condition (i) holds as well. □

Theorem 2. *π^{LK} is the inverse of π, i.e. $\pi(\pi^{LK}(P)) = P$.*

Proof. We proceed by induction on the number of inferences in P.

Case 1. P consists of a single inference $\iota = \textsc{ax}(\varnothing; y_1, y_2)$. Then $\pi^{LK}(P)$ is of the form $A^{y_1} \vdash A^{y_2}$, and clearly $\pi(\pi^{LK}(P)) = P$.

Case 2. The smallest inference $\iota = r(\mathbf{x}_1; \dots ; \mathbf{x}_n; \mathbf{y})$ in P is n-ary, i.e. there exist inferences $\kappa_1, \dots, \kappa_n \succ^1 \iota$. Then, by induction hypothesis, we have $\pi(\pi^{LK}(P\restriction_{\kappa_i})) = P\restriction_{\kappa_i}$ for $1 \leq i \leq n$. It remains to show that $\iota = \iota_{S_\iota}$. Let $\bar{S} = S_1 \cup \dots \cup S_n$ where S_i is the end-sequent of $\pi^{LK}(P\restriction_{\kappa_i})$, and let $\bar{X} = \{\lceil \mathbf{x}_1 \rceil\} \cup \dots \cup \{\lceil \mathbf{x}_n \rceil\}$. By definition in Transformation 1 we have that $S_\iota = (\bar{S} \setminus \bar{X}) \cup \{\lceil \mathbf{y} \rceil\}$. Then, $\iota_{S_\iota} = r(\mathbf{u}_1; \dots ; \mathbf{u}_n; \mathbf{v})$ where

$$\begin{aligned}
\{\lceil \mathbf{u}_i \rceil\} &= S_i \setminus S_\iota \\
&= S_i \setminus ((\bar{S} \setminus \bar{X}) \cup \{\lceil \mathbf{y} \rceil\}) \\
&= S_i \setminus (\bar{S} \setminus \bar{X}) \\
&= S_i \setminus (S_i \setminus \bar{X}) \\
&= S_i \setminus (S_i \setminus \{\lceil \mathbf{x}_i \rceil\}) \\
&= \{\lceil \mathbf{x}_i \rceil\}
\end{aligned}$$

and

$$\{\lceil \mathbf{v} \rceil\} = S_\iota \setminus \bar{S} = ((\bar{S} \setminus \bar{X}) \cup \{\lceil \mathbf{y} \rceil\}) \setminus \bar{S} = \{\lceil \mathbf{y} \rceil\}.$$

□

3.2 Abstractions

Based on the tuple representation of *LK*-proofs in Definition 14, we are now able to characterize the abstract proof structures of interest.

We will see that, from a different point of view, these abstractions can be treated as equivalence relations on proofs. To be more precise, these equivalence relations form a partially ordered set and our abstract proof structures can be seen as reference points for classifying a newly defined equivalence relation on proofs.

3.2.1 Partially ordered equivalence relations

We will show that a set of equivalence relations can be partially ordered by means of the refinement relation.

Definition 15 (Refinement). Let $\sim_1$ and $\sim_2$ be equivalence relations on a set $\mathscr{A}$. Then $\sim_1$ is *finer* than $\sim_2$, denoted $\sim_1 \preceq \sim_2$, if $X \sim_1 Y$ implies $X \sim_2 Y$ for all $X, Y \in \mathscr{A}$.

Proposition 2. *Let E be a set of equivalence relations on a fixed set and $\preceq$ the refinement relation on E. Then $\langle E, \preceq \rangle$ is a partially ordered set.*

Proof. Let $\sim_1$, $\sim_2$ and $\sim_3$ be equivalence relations.

(Reflexivity) Clearly we have $X \sim_1 Y$ implies $X \sim_1 Y$ and therefore $\sim_1 \preceq \sim_1$.

(Antisymmetry) Assume $\sim_1 \preceq \sim_2$ and $\sim_2 \preceq \sim_1$, then $X \sim_1 Y$ implies $X \sim_2 Y$ and vice versa for all X, Y. Therefore $\sim_1 = \sim_2$.

(Transitivity) Assume $\sim_1 \preceq \sim_2$ and $\sim_2 \preceq \sim_3$, then $X \sim_1 Y$ implies $X \sim_2 Y$ implies $X \sim_3 Y$. Therefore $\sim_1 \preceq \sim_3$. □

A function induces an equivalence relation as follows.

Definition 16 (Equivalence relation)**.** Let $f : \mathscr{A} \to \mathscr{B}$ be a function and $X, Y \in \mathscr{A}$. Then $X \sim_f Y$ if $f(X) = f(Y)$.

Next we show that $\sim_f$ is finer than $\sim_g$ if and only if g is a composition of f and some other function.

Proposition 3. *Let $f : \mathscr{A} \to \mathscr{B}$ and $g : \mathscr{A} \to \mathscr{C}$ be functions. Then $\sim_f \preceq \sim_g$ iff $g = h \circ f$ for some function $h : \mathscr{B} \to \mathscr{C}$.*

Proof. ($\Rightarrow$) Assume $\sim_f \preceq \sim_g$. Then $X \sim_f Y$ implies $X \sim_g Y$ for every X, Y. Thus there do not exist elements X_0, Y_0 s.t. $X_0 \sim_f Y_0$ and $X_0 \nsim_g Y_0$ holds. Hence there must be a function h s.t. $g = h \circ f$.

($\Leftarrow$) Assume $X \sim_f Y$. Then by definition of $\sim_f$ we have $f(X) = f(Y)$ and clearly $h(f(X)) = h(f(Y))$. Therefore $X \sim_g Y$. □

In the following, we will discuss the partially ordered set (poset) of equivalence relations on the set of proofs, i.e. we consider various equivalence relations on proofs induced by the respective abstract proof structure. Note that the least element, and therefore finest equivalence relation in this poset is $\sim_{id}$ where id is the identity function.

Definition 17 (Equivalence class)**.** Let $f : \mathscr{A} \to \mathscr{B}$ be a function and $X \in \mathscr{A}$. Then $[X]_f = \{Y \mid Y \sim_f X\}$ denotes the equivalence class of X w.r.t. f.

Note that for any equivalence relation $\sim_f$ on a set $\mathscr{A}$ we have that the set of equivalence classes $[X]_f$ is a partition of $\mathscr{A}$. Moreover, being a refinement is equivalent to all equivalence classes being subsets:

Proposition 4. *Let $\sim_f$ and $\sim_g$ be equivalence relations on $\mathscr{A}$. Then $[X]_f \subseteq [X]_g$ for every $X \in \mathscr{A}$ iff $\sim_f \preceq \sim_g$.*

Proof. By definition of refinement and equivalence classes. □

3.2.2 Proof skeleton and proof net

Distinguishing between formula indices and actual formulas in a proof makes it straightforward to obtain a proof skeleton from a given proof. In addition, we separated inferences from their specific order of application, i.e. a proof net can be defined as simply as a proof skeleton.

Definition 18 (Proof skeleton)**.** Let F be a set of indices, I a set of inferences and $\preccurlyeq$ a partial order over I. Then $\langle F, I, \preccurlyeq \rangle$ is called *proof skeleton* if

(i) $F = \bigcup_{\iota \in I} Aux(\iota) \cup Princ(\iota)$

(ii) I is well-polarized, and

(iii) $\preccurlyeq$ is an inference order and its maximal elements are axioms.

Definition 19 (Proof net)**.** Let F be a set of indices, I a set of inferences and $\lceil\cdot\rceil : F \to \mathcal{F}$ a formula map. Then $\langle F, I, \lceil\cdot\rceil\rangle$ is called *proof net* if

(i) $F = \bigcup_{\iota \in I} Aux(\iota) \cup Princ(\iota)$

(ii) I is well-polarized, and

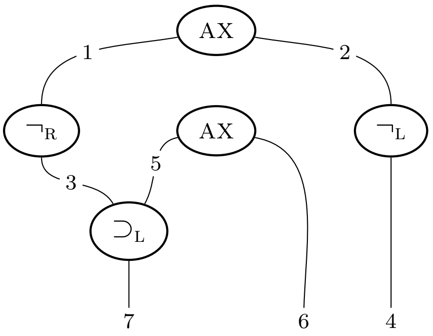
(iii) $\lceil\cdot\rceil$ is correct w.r.t. I.

What we refer to as proof nets is usually called pre-net or proof-structure in the literature (cf. Girard [6, 7] and Robinson [12]). In contrast to the definition of Girard and Robinson we do not impose any soundness criterion on proof nets. That is, we are only interested in the structure itself, not in characterizing which of them correspond to proofs.

Note that Definitions 18 and 19 are projections of Definition 14, that is, the elegance of this tuple representation lies in the fact that we define proof nets and proof skeletons of a proof by eliminating an element from the tuple. Thus, given a proof $P = \langle F, I, \preccurlyeq, \lceil\cdot\rceil\rangle$, we call $\langle F, I, \preccurlyeq\rangle$ the proof skeleton of P, and $\langle F, I, \lceil\cdot\rceil\rangle$ the proof net of P. Analogously, we call $\langle F, I\rangle$ the proof net skeleton of P. A proof net skeleton serves as a common base for proof nets and proof skeletons, and is defined as follows.

Definition 20 (Proof net skeleton)**.** Let F be a set of indices and I a well-polarized set of inferences. Then $\langle F, I\rangle$ is called *proof net skeleton.*

Example 7. The graphical representation of the proof net skeleton of $P_\sharp$ is the following. Note that we have formula indices instead of indexed formulas.



In terms of the before-mentioned poset of equivalence relations, we have here three different equivalence relations defined on proofs: Let σ_P (ν_P) be the projection from a proof to its proof skeleton (proof net), and σ_N (ν_S) the projection from a proof net (proof skeleton) to its proof net skeleton. First, we have $\sim_{\sigma_P}$ denoting the relation of two proofs having the same proof skeleton. Second, $\sim_{\nu_P}$ for having the same proof nets. Third, $\sim_{\nu_S \circ \sigma_P}$ if proofs have the same proof net skeleton. Note that in the strict sense we would also have $\sim_{\sigma_N \circ \nu_P}$, however, in Section 4.1 we show that $\nu_S \circ \sigma_P = \sigma_N \circ \nu_P$. Furthermore, we have by Proposition 3 that $\sim_{\sigma_P}$ and $\sim_{\nu_P}$ are finer than $\sim_{\nu_S \circ \sigma_P}$ and $\sim_{\sigma_N \circ \nu_P}$.

3.2.3 Atomic flow graph

This section is about formally defining AFGs and defining a transformation from proofs to AFGs. This transformation is divided into pieces which makes it possible to use these transformations in a more modular fashion, e.g. it is possible to define a transformation from proof nets to topologically reduced AFGs.

Definition 21 (Atomic flow graph)**.** Let $F[x/t]$ denote the formula where every occurrence of the variable x is replaced by the term t. An *atomic flow graph* $\langle V, E, \lceil\cdot\rceil_{\mathcal{A}}\rangle$ is a directed graph where V is a set of vertices, E a set of edges and $\lceil\cdot\rceil_{\mathcal{A}} : V \to \mathcal{A}$ a map from vertices to atomic formulas. Furthermore, for every pair of adjacent vertices u and v there must exist a variable x and a term t s.t. $\lceil u\rceil_{\mathcal{A}}[x/t] = \lceil v\rceil_{\mathcal{A}}[x/t]$.

Note that in the propositional case we have that adjacent vertices have to exhibit the same formula as there are no variables involved, that is, in the definition above we would have $\lceil u\rceil_{\mathcal{A}} = \lceil v\rceil_{\mathcal{A}}$.

Let x^p_ι denote a vertex x which is annotated with the polarity p and inference ι, where either $p = +$ or $p = -$. Furthermore, $\bar{p}$ denotes the dual of p, i.e. $\bar{+} = -$ and $\bar{-} = +$. If V is a set of annotated vertices, then $V^p_\iota \subseteq V$ is the set of those vertices in V annotated with polarity p and inference ι.

The following construction of an annotated graph serves as the basis for transforming a proof into an atomic flow graph. Intuitively, this first transformation just considers occurrences of active formulas. That is, we get edges between auxiliary and principal formulas only. To get the full AFG, additional vertices have to be inserted according to the inference order of the proof (cf. Transformation 4). These additional vertices correspond to the side formulas in the proof.

This modular composition of the following transformations enables us to construct an AFG from a proof net of a proof P, which is a subdivision of the AFG of P.

Transformation 3. Let I be a set of inferences and $\lceil\cdot\rceil : F \to \mathcal{F}$ a formula map. We inductively construct the *annotated graph* $\langle V, E\rangle$ as follows.

1. For every axiom $\iota = \textsc{ax}(\varnothing; x, y)$ create vertices x^-_ι and y^+_ι, and an edge from x^-_ι to y^+_ι.
2. For every weakening inference $\iota = r(\varnothing; x)$ and every atomic subformula $\lceil y\rceil$ of $\lceil x\rceil$: create a vertex y^-_ι if y occurs negatively in ι, and y^+_ι if it occurs positively.

3. For every $\iota = r(x_1, x_2; z)$ where r is either $\textsc{c}_\textsc{l}$ or $\textsc{c}_\textsc{r}$: Let κ_1, κ_2 be inferences s.t. $x_1 \in \mathit{Princ}(\kappa_1)$ and $x_2 \in \mathit{Princ}(\kappa_2)$. Then for every $x \in V^p_{\kappa_1}$ let $y \in V^p_{\kappa_2}$ be the vertex corresponding to the same subformula as x. Then create a vertex z^p_ι, and an edge from x to z^p_ι and from y to z^p_ι if $p = +$ and vice versa otherwise.
4. For every $\textsc{cut}(x_1; x_2; \varnothing)$: Let ι, κ be the inferences s.t. $x_1 \in \mathit{Princ}(\iota)$ and $x_2 \in \mathit{Princ}(\kappa)$. Then for every $x \in V^p_\iota$ let $y \in V^{\bar{p}}_\kappa$ be the vertex corresponding to the same subformula as x. Then create an edge from x to y if $p = +$ and vice versa otherwise.
5. For an n-ary logical inference $\iota = r(\mathbf{x}_1; \ldots; \mathbf{x}_n; y)$ where $\{x_1, \ldots, x_k\} = \mathit{Aux}(\iota)$: Let κ_i be the inference s.t. $x_i \in \mathit{Princ}(\kappa_i)$ for $i = 1, \ldots, k$. Then for every $x \in V^p_{\kappa_i}$ create a vertex x^p_ι, and an edge from x to x^p_ι if $p = +$ and vice versa otherwise.

Example 8. Applying Transformation 3 to the proof $P_\sharp$ yields to following annotated graph.

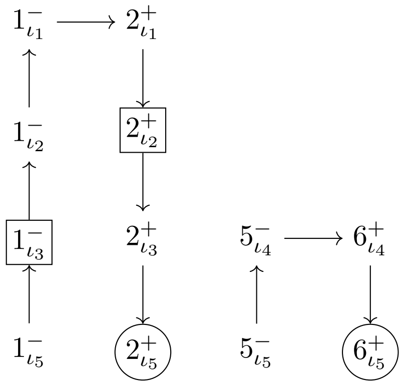
$$\begin{tikzcd}
1^-_{\iota_1} \arrow[r] & 2^+_{\iota_1} \arrow[d] & & \\
1^-_{\iota_2} \arrow[u] & 2^+_{\iota_3} & 5^-_{\iota_4} \arrow[r] & 6^+_{\iota_4} \\
1^-_{\iota_5} \arrow[u] & & 5^-_{\iota_5} \arrow[u] &
\end{tikzcd}$$

Transformation 4. Let $\langle V, E \rangle$ be an annotated graph and $\preccurlyeq$ an inference order. We create the *extended* annotated graph as follows.

1. Let λ be the root of $\preccurlyeq$. For every vertex $x^p_\iota \in V$ s.t. $\iota \neq \lambda$ and there exists no $(x^p_\iota, y^q_\kappa) \in E$ or $(y^q_\kappa, x^p_\iota) \in E$: Create a vertex x^p_λ; and if $p = +$ create an edge (x^p_ι, x^p_λ), otherwise create (x^p_λ, x^p_ι).
2. For every edge $e = (x^p_\iota, y^q_\kappa) \in E$: Let $\lambda_1, \ldots, \lambda_n$ be inferences s.t. $\iota \prec^1 \lambda_1 \prec^1 \cdots \prec^1 \lambda_n \prec^1 \kappa$ or $\kappa \prec^1 \lambda_1 \prec^1 \cdots \prec^1 \lambda_n \prec^1 \iota$. Then remove e, create vertices $z^p_\lambda, \ldots, z^p_{\lambda_n}$ and edges $(x^p_\iota, z^p_{\lambda_1}), (z^p_{\lambda_1}, z^p_{\lambda_2}), \ldots, (z^p_{\lambda_n}, y^q_\kappa)$.

As already mentioned above, the vertices created in Transformation 3 correspond to active formulas only, i.e. occurrences for side formulas are not considered. Therefore, the intuitive meaning of the first step in Transformation 4 is to create an occurrence for every side formula of the last inference. As a second step the edges have to be expanded according to the number of inferences in which the formulas have not been active, i.e. the number of inferences between its use as principal and auxiliary index (or the last inference of the proof if it is not auxiliary for any inference).

Example 9. Applying Transformation 4 to the graph depicted in Example 8 yields the following extended annotated graph. The circled vertices were introduced by step 1 of the transformation, and the rectangular vertices are the result of step 2.



Transformation 5. Let $\langle V, E\rangle$ be an (extended) annotated graph, $\preccurlyeq$ a partial order over a set of inferences and $\lceil\cdot\rceil : F \to \mathcal{F}$ a formula map. We create the *labelled* (extended) annotated graph $\langle V, E, \lceil\cdot\rceil_{\mathcal{A}}\rangle$ as follows.

1. For a vertex x^p_ι where $\iota = \textsc{ax}(\varnothing; x_1, x_2)$, let $\lceil x^p_\iota\rceil_{\mathcal{A}} = \lceil x_q\rceil$.
2. For a vertex x^p_ι where $\iota = r(\varnothing; x)$ is a weakening inference, let $\lceil x^p_\iota\rceil_{\mathcal{A}}$ be the associated atomic subformula of $\lceil x\rceil$.
3. For a vertex y^p_ι where $\iota = r(x; y)$ is a quantifier inference, let $\lceil y^p_\iota\rceil_{\mathcal{A}}$ be the associated atomic subformula of $\lceil y\rceil$.
4. For vertices x^p_ι and y^q_κ where $\lceil y^q_\kappa\rceil_{\mathcal{A}}$ is defined, ι is a quantifier inference and $\iota \prec \kappa$: Let $z_1, \ldots, z_n$ be the vertices between x^p_ι and y^q_κ. Then $\lceil z_i\rceil_{\mathcal{A}} = \lceil y^q_\kappa\rceil_{\mathcal{A}}$ for all $i = 1, \ldots, n$.
5. For adjacent vertices x and y where $\lceil x\rceil_{\mathcal{A}}$ is defined and $\lceil y\rceil_{\mathcal{A}}$ is not, let $\lceil y\rceil_{\mathcal{A}} = \lceil x\rceil_{\mathcal{A}}$.

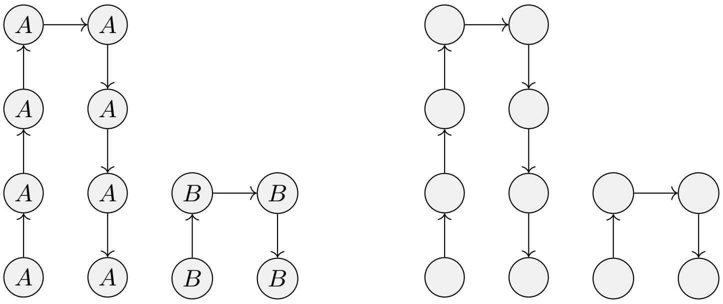
Definition 22. The unannotated result of applying Transformations 3, 4 and 5 to a proof P is the *atomic flow graph of* P.

Definition 22 defines a function γ_P transforming a proof into an atomic flow graph. That is, $R \sim_{\gamma_P} S$ holds if the proofs R and S have the same atomic flow graph. Analogously, we can define equivalence relations based on the following abstract proof structures.

Definition 23 (Atomic flow graph skeleton)**.** An *atomic flow graph skeleton* $\langle V, E\rangle$ is a directed graph where V is the set of vertices and E the set of edges.

The atomic flow graph relates to the atomic flow graph skeleton as a proof does to a proof skeleton. Hence, given a proof P and its AFG $\langle V, E, \lceil\cdot\rceil_{\mathcal{A}}\rangle$, we call $\langle V, E\rangle$ the AFG skeleton of P.

Example 10. The atomic flow graph of $P_\sharp$ and its skeleton.



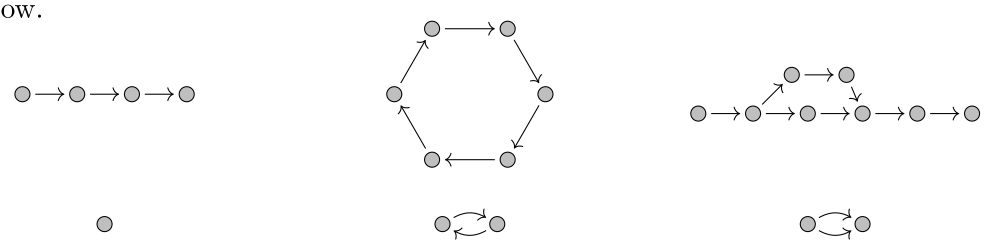
Note that the partial order in Transformation 5 need not necessarily be an inference order. As step 4 of the transformation handles the labels of occurrences between axioms (weakenings) and quantifier inferences, it suffices to know which axioms and weakenings are consumed by which quantifier inferences. Hence, the partial order induced by the set of inferences suffices for Transformation 5.

As a consequence, it is possible to construct an atomic flow graph from a proof net, which will be discussed in more detail in Section 4.1.

Definition 24 (Topological graph reduct)**.** The *topological reduct* of a directed graph $\langle V, E\rangle$ is obtained by contracting adjacent vertices $u, v \in V$ where $(u, v) \in E$, $(v, u) \notin E$ and the indegree of v is equal to 1.

In case of a labelled graph with labelling function $\lceil\cdot\rceil$ we have the additional condition that the vertices exhibit the same label, i.e. $\lceil u\rceil = \lceil v\rceil$.

Example 11. The original graphs are in the top row and the corresponding reductions are below.



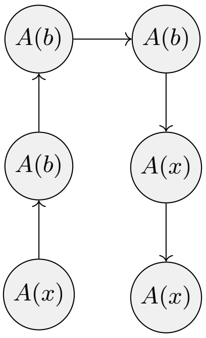
Besides the already introduced structures, we also consider topologically reduced versions of the graphs, i.e. *reduced AFG* and *reduced AFG skeleton* (of a given proof P).

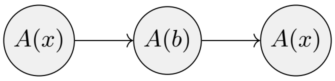
Example 12. The AFG of $P_\sharp$ collapses to two single vertices after reduction.

Example 13. Let P be the proof of the following LK-proof.

$$\dfrac{\dfrac{A(b) \vdash A(b)}{A(b) \vdash \exists x A(x)}\,\exists_{\mathrm{R}}}{\exists x A(x) \vdash \exists x A(x)}\,\exists_{\mathrm{L}}$$

Then we have the following AFG (left) and reduced AFG (right) of P.





CHAPTER 4

Relationships Between Proof Structures

4.1 Commutation

In the previous section we introduced various proof structures. The relationship between some of these structures might be obvious whereas for others it is certainly not. For instance, the relation between a proof and a proof skeleton is clear as the latter is just a substructure of the former. However, it is for example not obvious if it is possible to construct an AFG skeleton from a proof skeleton. We will clarify these relationships.

We say there exists an *arrow* from A to B, denoted as $A \rightarrow B$, if there exists a function s.t. the respective diagram commutes. Analogously, we say there does not exist an arrow, denoted as $A \nrightarrow B$, if such a function does not exist.

Furthermore, we distinguish proof-based structures (proof, proof skeleton, proof net, proof net skeleton) and graph-based structures (AFG, AFG skeleton, reduced AFG, reduced AFG skeleton). We start with the former and show that the natural mappings commute.

Proposition 5. *Let σ_P (ν_P) be the projection from a proof to its proof skeleton (proof net), and σ_N (ν_S) the projection from a proof net (proof skeleton) to its proof net skeleton. Then the following diagram commutes.*

$$\begin{tikzcd}
\text{Proof} \arrow[r, "\sigma_P"] \arrow[d, "\nu_P"] & \text{Skeleton} \arrow[d, "\nu_S"] \\
\text{Net} \arrow[r, "\sigma_N"] & \text{Net Skeleton}
\end{tikzcd}$$

Proof. By definition of σ_P, σ_N, ν_P and ν_S. □

Proposition 6. *Let σ_G denote the projection from an AFG to its skeleton, and ρ_G (ρ_S) the reduction of a labelled (unlabelled) graph. Furthermore, let $\sigma_R = \rho_S \circ \sigma_G$. Then the following diagram commutes.*

$$\begin{tikzcd}
\text{AFG} \arrow[r, "\sigma_G"] \arrow[d, "\rho_G"] & \text{AFG Skeleton} \arrow[d, "\rho_S"] \\
\text{Reduced AFG} \arrow[r, "\sigma_R"] & \text{Reduced AFG Skeleton}
\end{tikzcd}$$

Proof. As ρ_G can be seen as a pre-reduction step of ρ_S, it follows that $\rho_S \circ \sigma_G \circ \rho_G = \rho_S \circ \sigma_G$. □

Note that for propositional proofs we would have $\sigma_R = \sigma_G$ as all vertices of a path exhibit the same label. In other words, in case of quantifiers we have to reduce the graph again after skeletonizing (cf. Example 13).

Proposition 7. *Let τ_1 (τ_2, τ_3) be the function defined by Transformation 3 (4, 5), and let the reduction of a labelled graph be denoted by the function ρ_G. Now let $\gamma_P = \tau_3 \circ \tau_2 \circ \tau_1$ and $\gamma_N = \rho_G \circ \tau_3 \circ \tau_1$, and let ν_P be the projection from a proof to its proof net. Then the following diagram commutes.*

$$\begin{tikzcd}
\text{Proof} \arrow[r, "\gamma_P"] \arrow[d, "\nu_P"] & \text{AFG} \arrow[d, "\rho_G"] \\
\text{Net} \arrow[r, "\gamma_N"] & \text{Reduced AFG}
\end{tikzcd}$$

Proof. We have to show that

$$\rho_G \circ \gamma_P = \rho_G \circ \tau_3 \circ \tau_2 \circ \tau_1 = \rho_G \circ \tau_3 \circ \tau_1 \circ \nu_P = \gamma_N \circ \nu_P$$

holds. Note that τ_2 either attaches a vertex u with indegree 0 to a vertex v s.t. v then has indegree 1, or inserts vertices with indegree 1. In both cases we have that the introduced vertices are being contracted in the reduction procedure. Hence, the reduct of an AFG yields the same result regardless of whether τ_2 was applied or not, i.e. $\rho_G \circ \tau_3 \circ \tau_2 \circ \tau_1 = \rho_G \circ \tau_3 \circ \tau_1$. Furthermore, as the inference order of a proof is needed for τ_2 only, we have $\rho_G \circ \tau_3 \circ \tau_1 = \rho_G \circ \tau_3 \circ \tau_1 \circ \nu_P$. □

Combining Propositions 5, 6 and 7 we get Diagram 4.1. In fact, the arrows depicted in the diagram are the only abstractions existing between these proof structures. The rest of this subsection will be devoted to proving this.

First we show a basic property of commuting squares.

Lemma 3. *Let the following diagram commute, and assume $B \nrightarrow C$ and $C \nrightarrow B$ holds.*

$$\begin{tikzcd}
\text{A} \arrow[r] \arrow[d] & \text{B} \arrow[d] \\
\text{C} \arrow[r] & \text{D}
\end{tikzcd}$$

$$\begin{tikzcd}
 & \text{Proof} \arrow[rr, "\gamma_P"] \arrow[dl, "\sigma_P"'] \arrow[dd, "\nu_P"] & & \text{AFG} \arrow[dl, "\sigma_G"] \arrow[dd, "\rho_G"] \\
\text{Skeleton} \arrow[dd, "\nu_S"] & & \text{AFG Skeleton} \arrow[dd, "\rho_S"] & \\
 & \text{Net} \arrow[rr, "\gamma_N"] \arrow[dl, "\sigma_N"] & & \text{Reduced AFG} \arrow[dl, "\sigma_R"] \\
\text{Net Skeleton} & & \text{Reduced AFG Skeleton} &
\end{tikzcd}$$

Figure 4.1: Abstractions between proof structures for *LK*.

Then there are no other arrows than those depicted in the diagram and those following from transitivity.

Proof. First we show that the reverse direction of the depicted arrows do not hold. We prove only one case, all others are treated analogously. Assume $C \to A$, then we have $C \to A \to B$ which contradicts the assumption that $C \not\rightarrow B$.

It remains to show that $D \not\rightarrow A$ holds. Assume $D \to A$ holds, then also $D \to B$ which is a contradiction. □

Showing the non-existence of the following arrows, which correspond to diagonals of Figure 4.1, is the most basic step for proving that there are no other arrows, as most of the other non-existence results follow from these ones. We will see in the next section (cf. Lemma 4) that these are the only ones needed provided we have a complete commuting cube. As we do not have a complete cube here, we have to give counterexamples in some more cases.

In the following we will write *by X* to indicate that something follows from Equation (X) by a transitivity argument of the following form: If $A \to B$ and $A \not\rightarrow C$ hold, then $B \not\rightarrow C$ must hold.

Proposition 8. *The following holds:*

$$\text{Net} \not\rightarrow \text{AFG Skeleton} \tag{4.1}$$

$$\text{Skeleton} \not\rightarrow \text{Reduced AFG} \tag{4.2}$$

$$\text{AFG} \not\rightarrow \text{Net Skeleton} \tag{4.3}$$

Proof. First we prove (4.1). Let R and S be the proofs associated with the following *LK*-proofs.

$$\neg_{\mathrm{R}}\,\cfrac{\mathrm{w_L}\,\cfrac{A^1 \vdash A^2}{B^3, A^1 \vdash A^2}}{B^3 \vdash A^2, \neg A^4} \qquad\qquad \mathrm{w_L}\,\cfrac{\neg_{\mathrm{R}}\,\cfrac{A^1 \vdash A^2}{\vdash A^2, A^4}}{B^3 \vdash A^2, \neg A^4}$$

Hence, R and S have the same proof net, i.e. $\nu_P(R) = \nu_P(S)$. In contrast, the graph component in the AFG skeleton of R corresponding to the atomic formula B contains two vertices, and the AFG skeleton of S contains only one vertex corresponding to B, i.e. $\sigma_G(\gamma_P(R)) \neq \sigma_G(\gamma_P(S))$.

To prove (4.2) let P and Q be proofs differentiated by the predicate symbols only. Then, on the one hand, P and Q have the same proof skeleton, i.e. $\sigma_P(P) = \sigma_P(Q)$. On the other hand, $\rho_G(\gamma_P(P)) \neq \rho_G(\gamma_P(Q))$ as P and Q contain different atomic formulas, i.e. P and Q have different reduced AFGs.

To show (4.3) consider the proofs P and Q associated with the following LK-proofs.

$$\mathrm{W_L}\ \frac{A^1 \vdash A^2}{B^3, A^1 \vdash A^2} \qquad\qquad \mathrm{W_R}\ \frac{A^1 \vdash A^2}{A^1 \vdash A^2, B^3}$$

Then $\gamma_P(P) = \gamma_P(Q)$ but $\sigma_N(\nu_P(P)) \neq \sigma_N(\nu_P(Q))$ as P and Q do not contain the same inferences, i.e. they have the same AFG but different net skeletons. □

The following propositions, viz. Proposition 10, 9 and 11, show the non-existence of the diagonal arrows for the diagrams in Proposition 5, 6 and 7 s.t. we can apply Lemma 3.

Proposition 9. *The following holds:*

$$\text{Net} \not\twoheadrightarrow \text{Skeleton} \tag{4.4}$$

$$\text{Skeleton} \not\twoheadrightarrow \text{Net} \qquad \textit{by } 4.2 \tag{4.5}$$

Proof. To show (4.4) consider the proofs R and S from the proof of Proposition 8. Then $\sigma_P(R) \neq \sigma_P(S)$ but $\nu_P(R) = \nu_P(S)$. □

Proposition 10. *The following holds:*

$$\text{AFG Skeleton} \not\twoheadrightarrow \text{Reduced AFG} \tag{4.6}$$

$$\text{Reduced AFG} \not\twoheadrightarrow \text{AFG Skeleton} \qquad \textit{by } 4.1 \tag{4.7}$$

Proof. To prove (4.6) let P and Q be proofs differentiated by the predicate symbols only. Then, on the one hand, P and Q have the same AFG skeleton, i.e. $\sigma_G(\gamma_P(P)) = \sigma_G(\gamma_P(Q))$. On the other hand, $\rho_G(\gamma_P(P)) \neq \rho_G(\gamma_P(Q))$ as P and Q contain different atomic formulas, i.e. P and Q have different reduced AFGs. □

Proposition 11. *The following holds:*

$$\text{Net} \not\twoheadrightarrow \text{AFG} \qquad \textit{by } 4.1 \tag{4.8}$$

$$\text{AFG} \not\twoheadrightarrow \text{Net} \qquad \textit{by } 4.3 \tag{4.9}$$

Next we show that the cube in Figure 4.1 is in fact not complete, i.e. we do not have an arrow from Skeleton (Net Skeleton) to AFG Skeleton (Reduced AFG Skeleton). We are doing so by showing that two of the diagonals of the front face do not exist, and the remaining ones follow.

Proposition 12. *The following holds:*

$$\text{Skeleton} \not\twoheadrightarrow \text{Reduced AFG Skeleton} \tag{4.10}$$

$$\text{AFG Skeleton} \not\twoheadrightarrow \text{Net Skeleton} \qquad \textit{by } 4.3 \tag{4.11}$$

Proof. To prove (4.10) let P and Q be the proofs associated with the following LK-proofs.

$$\text{W}_\text{L}\,\frac{A^1 \vdash A^2}{B \wedge C^3, A^1 \vdash A^2} \qquad\qquad \text{W}_\text{L}\,\frac{A^1 \vdash A^2}{B^3, A^1 \vdash A^2}$$

Then both have the same proof skeleton, however, P and Q have different reduced AFG skeletons as P contains three atomic formulas and Q just two. □

Proposition 13. *The following holds:*

$$\text{Skeleton} \not\twoheadrightarrow \text{AFG Skeleton} \qquad \textit{by } 4.10 \tag{4.12}$$

$$\text{Net Skeleton} \not\twoheadrightarrow \text{Reduced AFG Skeleton} \qquad \textit{by } 4.10 \tag{4.13}$$

$$\text{AFG Skeleton} \not\twoheadrightarrow \text{Skeleton} \qquad \textit{by } 4.11 \tag{4.14}$$

$$\text{Reduced AFG Skeleton} \not\twoheadrightarrow \text{Net Skeleton} \qquad \textit{by } 4.11 \tag{4.15}$$

$$\text{Net Skeleton} \not\twoheadrightarrow \text{AFG Skeleton} \qquad \textit{by } 4.13 \tag{4.16}$$

$$\text{Reduced AFG Skeleton} \not\twoheadrightarrow \text{Skeleton} \qquad \textit{by } 4.15 \tag{4.17}$$

Next we show that we do not have the diagonals on the top and bottom face of the diagram.

Proposition 14. *The following holds:*

$$\text{Skeleton} \not\twoheadrightarrow \text{AFG} \qquad \textit{by } 4.2 \tag{4.18}$$

$$\text{AFG} \not\twoheadrightarrow \text{Skeleton} \qquad \textit{by } 4.3 \tag{4.19}$$

$$\text{Net Skeleton} \not\twoheadrightarrow \text{Reduced AFG} \qquad \textit{by } 4.2 \tag{4.20}$$

$$\text{Reduced AFG} \not\twoheadrightarrow \text{Net Skeleton} \qquad \textit{by } 4.3 \tag{4.21}$$

It remains to show that we do not have the reverse directions of Proposition 8, and that we do not have an arrow from Reduced AFG Skeleton to Proof. Furthermore, we have to show that AFG Skeleton $\not\twoheadrightarrow$ Proof and Reduced AFG Skeleton $\not\twoheadrightarrow$ Net hold.

Proposition 15. *The following holds:*

$$\text{AFG Skeleton} \not\twoheadrightarrow \text{Net} \qquad \textit{by } 4.6 \tag{4.22}$$

$$\text{Reduced AFG} \not\twoheadrightarrow \text{Skeleton} \qquad \textit{by } 4.21 \tag{4.23}$$

$$\text{Net Skeleton} \not\twoheadrightarrow \text{AFG} \qquad \textit{by } 4.20 \tag{4.24}$$

$$\text{Reduced AFG Skeleton} \not\twoheadrightarrow \text{Proof} \qquad \textit{by } 4.17 \tag{4.25}$$

$$\text{AFG Skeleton} \not\twoheadrightarrow \text{Proof} \qquad \textit{by } 4.22 \tag{4.26}$$

$$\text{Reduced AFG Skeleton} \not\twoheadrightarrow \text{Net} \qquad \textit{by } 4.15 \tag{4.27}$$

Note that the above counterexamples used for showing the non-existence of an arrow can also be formulated by means of the poset of equivalence relations: For instance, let P, Q be proofs and f, g functions producing a proof net and an AFG skeleton respectively. If $P \sim_f Q$ and $P \nsim_g Q$ then $\sim_f \npreceq \sim_g$. This is equivalent to saying that there is no arrow from Net to AFG Skeleton. In fact, we have the following correspondence between commuting diagrams and the refinement of equivalence relations.

Proposition 16. *Assume the following diagram holds. Then $B \to C$ iff $\sim_f \preceq \sim_g$.*

$$\begin{tikzcd} A \arrow[r, "f"] \arrow[dr, "g"'] & B \\ & C \end{tikzcd}$$

Proof. Follows immediately from Proposition 3: $B \to C$ iff $g = h \circ f$ iff $\sim_f \preceq \sim_g$. □

4.2 Annotated weakening

As already shown in the previous section, it is not possible to construct an AFG skeleton from a proof skeleton. This is due to the weakening rules where we abstract information about the atomic formulas involved in the weakening formula when skeletonizing.

Example 14. Given the following proof and its skeleton where A, B and C are atomic formulas.

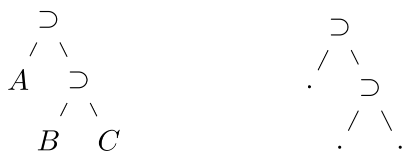
$$\mathrm{w_L}\,\frac{A^1 \vdash A^2}{B \wedge C^3, A^1 \vdash A^2} \qquad\qquad \mathrm{w_L}\,\frac{.^{1} \vdash .^{2}}{.^{3}, .^{1} \vdash .^{2}}$$

In the proof itself we know how many atomic formulas are involved, whereas for the proof skeleton we just know that there is a weakening formula, but we do not know the structure of it. Therefore it is not possible to create a vertex for every occurrence of an atomic formula when given a proof skeleton.

We can circumvent this issue by considering weakening rules annotated with formula skeletons.

Definition 25 (Formula skeleton)**.** Let F be a formula. Then deleting the atomic formulas from the syntax tree of F yields the *formula skeleton* of F, denoted as $\|F\|$.

Example 15. Let the formula F be $A \supset (B \supset C)$. Then the syntax tree and the formula skeleton of F are of the following form.



Note that in Transformation 3 the mapping $\lceil\cdot\rceil$ is only needed for the weakening rules to obtain the atomic subformulas. In a calculus where the weakening inferences are annotated with formula skeletons of the weakening formulas, the mapping $\lceil\cdot\rceil$ would not be necessary for constructing the graph obtained by Transformation 3. Hence, we define $LK^{\|\cdot\|}$ to be the sequent calculus obtained from LK by replacing the weakening rules with the following *annotated weakening rules.*

$$\frac{\Gamma \vdash \Delta}{(F,k), \Gamma \vdash \Delta}\,\mathrm{w}_{\mathrm{L}}^{\|F\|} \qquad\qquad \frac{\Gamma \vdash \Delta}{\Gamma \vdash \Delta, (F,k)}\,\mathrm{w}_{\mathrm{R}}^{\|F\|}$$

Note that the logical connectives in the formula skeletons are necessary for determining the polarity of atomic formulas, which in turn is essential for the direction of the arcs in an AFG.

Now with the concept of annotated weakening rules we can revoke Equations (4.12) and (4.13) for $LK^{\|\cdot\|}$ and prove the following result.

Proposition 17. *Let τ_1 (τ_2) be the function defined by Transformation 3 (4), and let the reduction of an AFG skeleton denoted by the function ρ_S. Now let $\gamma_S = \tau_2 \circ \tau_1$ and $\gamma_{NS} = \rho_S \circ \tau_1$, and let ν_S be the projection from a proof to its proof net. Then the following diagram commutes.*

$$\begin{tikzcd}
\text{Skeleton} \arrow[r, "\gamma_S"] \arrow[d, "\nu_S"] & \text{AFG Skeleton} \arrow[d, "\rho_S"] \\
\text{Net Skeleton} \arrow[r, "\gamma_{NS}"] & \text{Reduced AFG Skeleton}
\end{tikzcd}$$

Proof. We have to show that

$$\rho_S \circ \gamma_S = \rho_S \circ \tau_2 \circ \tau_1 = \rho_S \circ \tau_1 \circ \nu_S = \gamma_{NS} \circ \nu_S$$

holds. Note that τ_2 either attaches a vertex u with indegree 0 to a vertex v s.t. v then has indegree 1, or inserts vertices with indegree 1. In both cases we have that the introduced vertices are being contracted in the reduction procedure. Hence, the reduct of an AFG yields the same result regardless of whether τ_2 was applied or not, i.e. $\rho_S \circ \tau_2 \circ \tau_1 = \rho_S \circ \tau_1$. Furthermore, as the inference order of a proof is needed for τ_2 only, we have $\rho_S \circ \tau_1 = \rho_S \circ \tau_1 \circ \nu_P$. □

Note that the annotated weakening rules add information to skeletonized proof-based structures only. In case of proofs and proof nets the formula skeleton is implicitly given by the formula map, and graph-based structures do not contain information about inferences. Hence, the commutation results of the previous section are still valid, and combined with Proposition 17 we get the following commutative diagram for $LK^{\|\cdot\|}$.

Figure 4.2 illustrates three different dimensions of abstraction. First, we have the σ-functions for abstracting formulas, i.e. skeletonizing proof structures. Second, transformations of proof-based structures into graph-based structures, denoted by the γ-functions. Third, the ν- and ρ-functions, for abstracting the order of inferences and

$$\begin{tikzcd}
 & \text{Proof} \arrow[rr, "\gamma_P"] \arrow[dl, "\sigma_P"'] \arrow[dd, "\nu_P"] & & \text{AFG} \arrow[dl, "\sigma_G"] \arrow[dd, "\rho_G"] \\
\text{Skeleton} \arrow[rr, "\gamma_S"] \arrow[dd, "\nu_S"] & & \text{AFG Skeleton} \arrow[dd, "\rho_S"] & \\
 & \text{Net} \arrow[rr, "\gamma_N"] \arrow[dl, "\sigma_N"] & & \text{Reduced AFG} \arrow[dl, "\sigma_R"] \\
\text{Net Skeleton} \arrow[rr, "\gamma_{NS}"] & & \text{Reduced AFG Skeleton} &
\end{tikzcd}$$

Figure 4.2: Abstractions between proof structures for $LK^{\|\cdot\|}$.

reducing graphs respectively. Both can be seen as removal of what is called in literature bureaucracy. As for proof nets, where we do not mind the specific order of inferences (cf. identity of proofs), we reduce the length of paths for AFGs by applying the ρ-functions. However, we do not alter the basic structure of the graph.

Note that Proposition 8 still holds for $LK^{\|\cdot\|}$. Thus, by applying the following Lemma 4 we get that the arrows depicted in Figure 4.2 are the only ones.

Lemma 4. *Let the following diagram commute, and assume $A \nrightarrow H$, $G \nrightarrow B$ and $F \nrightarrow C$.*

$$\begin{tikzcd}
 & E \arrow[rr] \arrow[dl] \arrow[dd] & & F \arrow[dl] \arrow[dd] \\
A \arrow[rr] \arrow[dd] & & B \arrow[dd] & \\
 & G \arrow[rr] \arrow[dl] & & H \arrow[dl] \\
C \arrow[rr] & & D &
\end{tikzcd}$$

Then there are no other arrows than those depicted in the diagram and those following from transitivity.

Proof. The proof idea is to show that the diagonals of every face of the cube do not exist s.t. we can apply Lemma 3.

We treat only one face of the cube, all others are shown analogously. Assume $C \rightarrow B$ holds then $G \rightarrow C \rightarrow B$ holds which contradicts the initial assumption. Now assume $B \rightarrow C$ holds then $F \rightarrow B \rightarrow C$ would hold as well, contradicting the initial assumption. Hence we have $C \nrightarrow B$ and $B \nrightarrow C$, that is, we can apply Lemma 3 to the commuting square $A - B - C - D$.

Furthermore, we have to show that the reverse directions of the non-existent arrows in the assumption do not exist either, viz. $H \nrightarrow A$, $B \nrightarrow G$ and $C \nrightarrow F$. As $H \nrightarrow C$

holds we have $H \not\twoheadrightarrow A$. As $B \not\twoheadrightarrow C$ holds we have $B \not\twoheadrightarrow G$. Finally, as $C \not\twoheadrightarrow H$ holds we have $C \not\twoheadrightarrow F$.

It remains to show that $D \not\twoheadrightarrow E$ holds. Assume $D \to E$, then we have $D \to E \to A$ contradicting Lemma 3. □

4.3 Equivalence classes

This section is about determining the cardinalities of the equivalence classes generated by the abstractions defined in Section 4.1 and 4.2. Hence, we deal with the question of how many instances of a given proof structure induce the same abstraction.

For the following results we distinguish between finite (n) and infinite (∞) cardinalities, furthermore we consider the sequent calculus $LK^{\|\cdot\|}$ with annotated weakening rules as defined in Section 4.2. Note that the following results also hold for LK except those where we do not have abstractions in LK, viz. skeleton to AFG skeleton and net skeleton to reduced AFG skeleton.

Definition 26 (Path graph)**.** A *path graph* $\langle V, E\rangle$ is a tree having the following shape where $V = \{v_0, \ldots, v_n\}$ and $E = \{e_1, \ldots, e_n\}$.

$$v_0 \xrightarrow{e_1} v_1 \xrightarrow{e_2} \cdots \xrightarrow{e_n} v_n$$

Note that in a path graph, all arcs are directed in the same direction.

Lemma 5. *Let F be an arbitrary formula and k the number of atomic formulas contained in F. There exists a proof Π^F of $F \vdash F$ s.t. the AFG of Π^F is a forest consisting of k path graphs.*

Proof. By induction on the complexity of F.

If F is an atomic formula, then the AFG of Π^F contains a single path graph $F \longrightarrow F$.

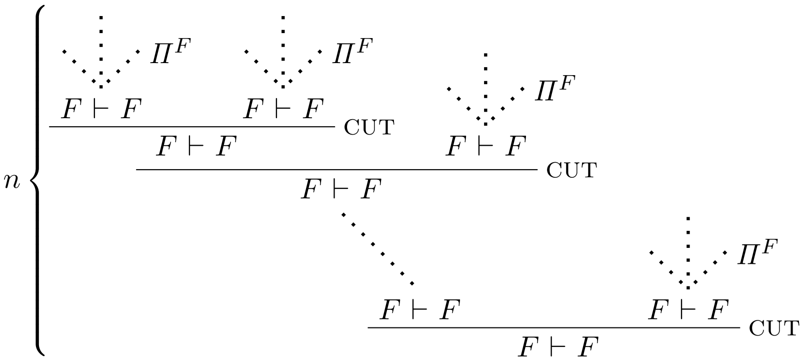
Let F be of the form $G \supset H$ where G and H are arbitrary formulas. Then the proof of $F \vdash F$ has the following form

$$\dfrac{\dfrac{\overset{\vdots\ \Pi^G}{G \vdash G} \qquad \overset{\vdots\ \Pi^H}{H \vdash H}}{G \supset H, G \vdash H}\supset_{\mathrm{L}}}{G \supset H \vdash G \supset H}\supset_{\mathrm{R}}$$

where Π^G and Π^H, by the induction hypothesis, consist of k_G and k_H path graphs respectively. As the atomic formulas in G and H have the same polarities in Π^F as in Π^G and Π^H, the path graphs of Π^G and Π^H are just lengthened. That is, the path graphs of Π^G and Π^H are subgraphs of the AFG of Π^F. Furthermore, the AFG of Π^F consists of $k_G + k_H$ path graphs which coincides with the number of atomic formulas in $G \supset H$. The connectives $\wedge$, $\vee$, $\neg$, $\forall$ and $\exists$ are treated analogously.

□

Definition 27. Let Π^F be a proof of $F \vdash F$ where F is an arbitrary formula. Then Π_n^F denotes the LK-proof consisting of n consecutive applied cut inferences. The equivalent proof is denoted as P_n^F.



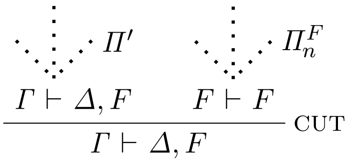
Lemma 6. *Let F be an arbitrary formula and k the number of atomic formulas contained in F. The AFG of a proof of P_n^F is a forest consisting of k path graphs.*

Proof. Follows from Lemma 6 and the definition of Transformation 3. □

Proposition 18. *Let W be an AFG, X an AFG skeleton, Y a proof net and Z a proof net skeleton.*

$$|[W]_{\rho_G}| = \infty \tag{4.28}$$
$$|[X]_{\rho_S}| = \infty \tag{4.29}$$
$$|[Y]_{\gamma_N}| = \infty \tag{4.30}$$
$$|[Z]_{\gamma_{NS}}| = \infty \tag{4.31}$$

Proof. Let Π be an arbitrary proof and let $S = \Gamma \vdash \Delta, F$ be a sequent in Π where F is quantifier-free and Π' the subproof of Π proving S. Note that such a sequent exists in every proof since we require atomic axioms. The case for $F, \Gamma \vdash \Delta$ is symmetric. Now consider the following proof Π'_n.



By Lemma 6, the AFG G_n^F of Π_n^F consists of k path graphs where k is the number of atomic subformulas in F. In particular, as F is quantifier-free, all adjacent vertices exhibit the same label, i.e. the reduct of G_n^F consists of k single vertices. Therefore, the reducts of the AFGs of Π'_n and Π are equal.

It is clear that there exist infinitely many proofs Π'_n, all exhibiting the same reduced AFG.

Similar arguments hold for (4.29), (4.30) and (4.31). □

There is a strong indication that we get finite cardinalities for the equivalence classes in Proposition 18 for cut-free proofs. This is due to the fact that cut inferences are needed to construct arbitrarily long path graphs.

Proposition 19. *Let W be a proof, X a proof net, Y an AFG and Z a reduced AFG.*

$$|[W]_{\sigma_P}| = \infty \tag{4.32}$$

$$|[X]_{\sigma_N}| = \infty \tag{4.33}$$

$$|[Y]_{\sigma_G}| = \infty \tag{4.34}$$

$$|[Z]_{\sigma_R}| = \infty \tag{4.35}$$

Proof. Since there exist an infinite number of predicate symbols, there are infinitely many proofs exhibiting the same proof skeleton. Analogous arguments hold for (4.33), (4.34) and (4.35). □

Proposition 20. *Let X be a proof and Y a proof skeleton.*

$$|[X]_{\nu_P}| = n \tag{4.36}$$

$$|[Y]_{\nu_S}| = n \tag{4.37}$$

Proof. As the number inferences contained in a proof P is finite, there are only finitely many permutations of these inferences. Consequently, there are finitely many possible partial orders of the inferences. An analogous argument holds for (4.37). □

Proposition 21. *Let X be a proof and Y a proof skeleton.*

$$|[X]_{\gamma_P}| = n \tag{4.38}$$

$$|[Y]_{\gamma_S}| = n \tag{4.39}$$

Proof. Let G be the AFG of a proof P. Since P is finite, also G is finite, i.e. the finite number of vertices in G can only result in a finite number of possible sequents. Furthermore, there are only finitely many permutations of these sequents. An analogous argument holds for (4.39). □

An overview of Propositions 18, 19, 20 and 21 is illustrated in Figure 4.3.

$$\begin{tikzcd}
 & \text{Proof} \arrow[rr, "n"] \arrow[dl, "\infty"'] \arrow[dd, "n"] & & \text{AFG} \arrow[dl, "\infty"] \arrow[dd, "\infty"] \\
\text{Skeleton} \arrow[rr, crossing over, "n"] \arrow[dd, "n"] & & \text{AFG Skeleton} & \\
 & \text{Net} \arrow[rr, "\infty"] \arrow[dl, "\infty"] & & \text{Reduced AFG} \arrow[dl, "\infty"] \\
\text{Net Skeleton} \arrow[rr, "\infty"] & & \text{Reduced AFG Skeleton} \arrow[from=uu, crossing over, "\infty"] &
\end{tikzcd}$$

Figure 4.3: Cardinalities of the equivalence classes induced by the abstractions for $LK^{\|\cdot\|}$.

CHAPTER 5

Most General Proof Nets

One of the advantages of defining a uniform framework of proof structures is the ability to generalize algorithms in an appropriate manner. In this chapter we are doing so by generalizing an algorithm defined on proof skeletons in [9]. Krajíček and Pudlák defined an algorithm for constructing a proof from a given cut-free proof skeleton S and a given end-sequent $\Gamma \vdash \Delta$, and derived bounds on the maximal depth of terms used in the proof. From a different point of view, this algorithm produces a most general proof s.t. every proof of $\Gamma \vdash \Delta$ with skeleton S is an instance of it. We generalize this concept to proof net skeletons, and get most general proof nets with a cut-free net skeleton.

Before defining the algorithm we have to introduce unification which is needed within the algorithm.

5.1 Unification

Besides the unification problem itself we define the notions substitution, unifier and most general unifier as usual.

Definition 28 (Unification problem)**.** A *unification problem* U is given by a set of pairs of terms $(s_1, t_1), \dots, (s_n, t_n)$.

Definition 29 (Substitution)**.** A *substitution* is a mapping $\sigma : V \to T$ where V is a set of variables and T a set of terms s.t. $x\sigma \neq x$ for finitely many $x \in V$.

Definition 30 (Unifier)**.** A solution of a unification problem U is given by a substitution σ s.t. $s\sigma = t\sigma$ for all (s, t) in U. Then σ is called *unifier* of U.

Definition 31 (Most general unifier)**.** Let U be a unification problem and σ_0 a unifier of U. Then σ_0 is a *most general unifier (mgu)* of U if every unifier σ of U can be decomposed into $\sigma = \sigma_0\sigma_1$ for some substitution σ_1.

The following bounds for the maximal depth of unified terms are extracted from an algorithm for finding most general unifiers (Chang and Lee [5]). For the proof we refer to [9].

Let t be a term. Then $\mathrm{dp}(t)$ denotes the depth of t and $|t|$ the size of t, i.e. the number of symbols in t.

Proposition 22 (Krajíček and Pudlák [9]). *Let U be a unification problem. Let T be the set of all terms in U and n the number of variables in U. Then every most general unifier σ of U satisfies the following inequalities:*

$$\max_{t \in T} \mathrm{dp}(t\sigma) \leq \sum_{t \in T} |t| \tag{5.1}$$

$$\max_{t \in T} \mathrm{dp}(t\sigma) \leq (n+1) \cdot \max_{t \in T} \mathrm{dp}(t) \tag{5.2}$$

5.2 Computing a most general proof net with a cut-free net skeleton

The algorithm defined in Krajíček and Pudlák [9] reduces the existence of a proof of a given sequent with a given cut-free proof skeleton to a unification problem with certain restrictions. The inequalities in Proposition 22 are then applied to a most general unifier of the unification problem to get bounds on the maximal depth of terms in a proof with a given skeleton and a given end-sequent. Such a restricted unification problem $U{\restriction}_R$ is given by a unification problem U and a set R of pairs (a, c) where a is a variable and c a constant. A unifier σ of U is a solution of $U{\restriction}_R$ if $a\sigma$ does not contain c for (a, c) in R. By adding certain restrictions (a, α) where a is a free variable and α an eigenvariable, the existence of a proof with a given skeleton and a given end-sequent can be reduced to a restricted unification problem. However, this set of restrictions is dependent on the specific inference order. To be more precise, a pair (a, α) is contained in R if α is the eigenvariable of an *LK*-inference J and a occurs below J. It is clear, that this set of restrictions is responsible for satisfying the eigenvariable condition. As a consequence, we cannot use this concept to find most general proof nets where we do not have an inference order at hand.

However, we show that the algorithm in [9] for deskeletonizing cut-free proof skeletons can also be applied to proof net skeletons, and that we can prove, with a slightly different approach, the same bounds for cut-free proof nets as for cut-free proofs. In contrast to Krajíček and Pudlák, we use an ordinary first-order unification problem U and show that every most general unifier of U produces a proof with inferences I, inference order $\preccurlyeq$ and end-sequent $\Gamma \vdash \Delta$ provided there exists a corresponding proof. Note that we do not have a one-to-one correspondence between the solvability of the unification problem and the existence of a proof of a given sequent with a given skeleton (see Remark 2). However, one direction suffices to show the existence of a most general proof net, and to prove the before-mentioned bounds.

In the following, we will distinguish between *terms* and *semiterms*: while terms contain free variables only, semiterms may also contain bound variables.

Note that we can identify, from a given set of inferences, those indices occurring in the end-sequent. That is, for a proof $P = \langle F, I, \preccurlyeq, \lceil\cdot\rceil\rangle$ we define X_I to be the set of indices contained in the end-sequent of P.

Definition 32. Given a set of inferences I we denote the set $X_I = \{x \mid \nexists\kappa \text{ s.t. } x \in \mathit{Aux}(\kappa)\}$ as the *end-indices of I*.

Given a cut-free set of inferences I and an indexed sequent $\mathit{\Gamma} \vdash \mathit{\Delta}$ we can construct a formula map by assigning the formulas in $\mathit{\Gamma} \vdash \mathit{\Delta}$ to the end-indices and propagating the formulas bottom-up.

Algorithm 1. Let I be a cut-free set of inferences and $\mathit{\Gamma} \vdash \mathit{\Delta}$ an indexed sequent. We construct a formula map $\lceil\cdot\rceil$ as follows:

1. Assign the formulas of $\mathit{\Gamma} \vdash \mathit{\Delta}$ to the indices in X_I accordingly. If the set of indices in $\mathit{\Gamma} \vdash \mathit{\Delta}$ does not match X_I, then there does not exist a proof of $\mathit{\Gamma} \vdash \mathit{\Delta}$ with inferences I.
2. For an inference ι where $\lceil y\rceil$ is defined for every $y \in \mathit{Princ}(\iota)$, set $\lceil x\rceil$ for every $x \in \mathit{Aux}(\iota)$ accordingly. If ι is a quantifier inference, introduce a new free variable.

Definition 33 (Variable-distinct map)**.** A formula map is called *variable-distinct* w.r.t. I if a new free variable is introduced at every quantifier inference. A free variable introduced at a weak quantifier inference is called *weak variable.*

If there exists a proof of $\mathit{\Gamma} \vdash \mathit{\Delta}$ with inferences I, then Algorithm 1 constructs a variable-distinct map. Furthermore, it produces a map which is correct w.r.t. all inferences except axioms.

Definition 34 (Semi-correct map)**.** A formula map $\lceil\cdot\rceil$ is *semi-correct* w.r.t. I if $\lceil\cdot\rceil$ is correct w.r.t. ι for every $\iota \in I$ where ι is not an axiom inference. Furthermore, for an axiom inference $\textsc{ax}(\varnothing; x, y)$ we require that $\lceil x\rceil = P(s_1, \ldots, s_n)$ and $\lceil y\rceil = P(t_1, \ldots, t_n)$.

Proposition 23. *If there exists a proof of $\mathit{\Gamma} \vdash \mathit{\Delta}$ with a cut-free set of inferences I, then Algorithm 1 yields a variable-distinct and semi-correct map.*

Proof. As every inference uniquely determines the formulas (up to the introduction of variables) for the auxiliary indices from a given principal formula, we end up with axioms whose formulas only differ w.r.t. terms, i.e. $\lceil\cdot\rceil$ is semi-correct. Furthermore, as we introduce a new free variable for every quantifier inference we get a variable-distinct map. □

Now given a cut-free set of inferences and a semi-correct map, we can construct a unification problem U s.t. U has a solution if there exists a corresponding proof.

For simplicity we use $\mathit{\Gamma} \vdash \mathit{\Delta}$ to denote the result of applying $\lceil\cdot\rceil$ to X_I where $\lceil\cdot\rceil$ is a formula map and I a set of inferences.

Definition 35. Let $\lceil\cdot\rceil$ be a semi-correct map with end-sequent $\Gamma \vdash \Delta$. We construct a unification problem U as follows:

(i) Eigenvariables, bound variables and free variables of $\Gamma \vdash \Delta$ are considered to be constants.

(ii) Weak variables are the variables of U.

(iii) For every semi-correct axiom $P(s_1, \dots, s_n) \vdash P(t_1, \dots, t_n)$ the pair (s_i, t_i) is contained in U for $i = 1, \dots, n$.

Lemma 7. *Let $\preccurlyeq$ be a partial order over a cut-free set of inferences I and $\lceil\cdot\rceil$ a semi-correct and variable-distinct map. If there exists a proof $\langle F, I, \preccurlyeq, \lceil\cdot\rceil_P\rangle$, then $\lceil\cdot\rceil_P$ is obtained from $\lceil\cdot\rceil$ by replacing weak variables with suitable terms and renaming eigenvariables.*

Proof. As $\lceil\cdot\rceil$ is semi-correct and I is cut-free we have that every formula is uniquely determined up to the terms introduced at quantifier inferences. Furthermore, as $\lceil\cdot\rceil$ is variable-distinct we can replace every free variable in $\lceil\cdot\rceil$ by the corresponding term in $\lceil\cdot\rceil_P$. □

Lemma 8. *Let $\preccurlyeq$ be a partial order over a cut-free set of inferences I and $\lceil\cdot\rceil$ a semi-correct and variable-distinct map. If there exists a proof with I and $\preccurlyeq$, then there exists a unifier σ of U.*

Proof. Follows from the definition of U and Lemma 7. □

Remark 2. Note that the converse direction of Lemma 8 does obviously not hold in general, just consider the following simple counterexample.

$$\exists_{\mathrm{R}}\,\dfrac{\exists_{\mathrm{L}}\,\dfrac{F(a)^1 \vdash F(b)^2}{\exists x F(x)^3 \vdash F(b)^4}}{\exists x F(x)^5 \vdash \exists x F(x)^6}$$

As we already mentioned above, a fine-grained construction of the set of restrictions for the unification problem is required in order to get the second direction. This construction is heavily dependent on the specific inference order, and since we are interested in generalizing this concept to proof net skeletons where we do not have a specific inference order at hand we avoid such a dependent set of restrictions.

However, if there exists a proof of $\Gamma \vdash \Delta$ with a given cut-free set of inferences and a given inference order, we can show that any most general unifier of the associated unification problem produces a regular proof.

Definition 36 (Regular map)**.** Let $\preccurlyeq$ be a partial order over a set of inferences I. A formula map is called *regular* w.r.t. $\langle I, \preccurlyeq\rangle$ if

(i) all eigenvariables are distinct, and

(ii) an eigenvariable α of an inference ι only occurs above ι.

Lemma 9. *Let $\preccurlyeq$ be a partial order over a cut-free set of inferences I. Furthermore, let $\lceil\cdot\rceil$ be a semi-correct and variable-distinct map with end-sequent $\Gamma \vdash \Delta$, U the associated unification problem and σ a most general unifier of U. If there exists a proof of $\Gamma \vdash \Delta$ with I and $\preccurlyeq$, then $\lceil\cdot\rceil\sigma$ is eigenvariable-preserving, regular and correct.*

Proof. By definition of U we have that $\lceil\cdot\rceil\sigma$ is correct.

Now assume $\lceil\cdot\rceil\sigma$ is not eigenvariable-preserving. Then there exist inferences $\iota \prec \kappa$ s.t. ι is a strong quantifier inferences with eigenvariable α and there is an index $x \in \mathit{Princ}(\kappa)$ s.t. $\lceil x\rceil\sigma$ contains α and x is not consumed by ι, i.e. there does not exist a $\lambda \not\succcurlyeq \iota$ s.t. $x \in \mathit{Aux}(\lambda)$. As $\lceil\cdot\rceil$ is variable-distinct and α is considered to be constant in U, there exists a weak variable a in $\lceil x\rceil$ s.t. $a\sigma = t(\alpha)$. As σ is a mgu of U there does not exist a unifier σ' of U s.t. $a\sigma' = t(b)$ where b is a variable. Hence α must be contained in $a\sigma$ in order to unify U and consequently to get correct axioms. This in turn means that there does not exist a proof with I and $\preccurlyeq$.

Assume $\lceil\cdot\rceil\sigma$ is not regular. Note that all eigenvariables are pairwise distinct as $\lceil\cdot\rceil$ is variable-distinct and eigenvariables are considered to be constants in U. Thus, there exist inferences $\kappa \prec \iota$ s.t. ι is a strong quantifier inference with eigenvariable α and α is contained in $\lceil x\rceil\sigma$ for some $x \in \mathit{Princ}(\kappa)$. As $\lceil\cdot\rceil$ is variable-distinct and α is considered to be constant, there must be a weak variable a in $\lceil x\rceil$ s.t. $a\sigma = t(\alpha)$. As σ is a mgu of U there does not exist a unifier σ' of U s.t. $a\sigma' = t(b)$ where b is a variable. Thus α must be contained in $a\sigma$ in order to unify U and to get correct axioms. However, α must then be contained in the lower sequent of ι, which cannot be the case as $\lceil\cdot\rceil\sigma$ is eigenvariable-preserving. □

Lemma 9 shows that there exists a most general proof net with a given cut-free net skeleton and a given end-sequent. Furthermore, by combining Proposition 23 with Lemmata 8 and 9 we can apply the inequalities in Proposition 22 to the maximal depth of terms in proof nets.

Theorem 3. *Suppose $\Gamma \vdash \Delta$ has a cut-free proof P with proof net skeleton $P_{NS} = \langle F, I\rangle$. Let T be the set of maximal semiterms of $\Gamma \vdash \Delta$, let n be the number of inferences in I and m the number of weak quantifier inferences. Then there exists a proof net P'_N of a proof P' with proof net skeleton P_{NS} s.t. the depth of every semiterm in P_N is bounded from above by*

$$n \cdot \sum_{t\in T} |t| \text{ and} \tag{5.3}$$

$$(m+1) \cdot \max_{t\in T} \mathrm{dp}(t). \tag{5.4}$$

Proof. Let $P = \langle F, I, \preccurlyeq, \lceil\cdot\rceil\rangle$. By Algorithm 1 we can construct a variable-distinct map $\lceil\cdot\rceil_v$ from P_{NS} and $\Gamma \vdash \Delta$. As P_{NS} is the proof net skeleton of P we have by Lemma 8 that the associated unification problem U is unifiable. Furthermore, by Lemma 9, every mgu of U produces a map $\lceil\cdot\rceil_v\sigma$ s.t. $P' = \langle F, I, \preccurlyeq, \lceil\cdot\rceil_v\sigma\rangle$ is a proof and $P'_N = \langle F, I, \lceil\cdot\rceil_v\sigma\rangle$ is the proof net of P'.

(5.3) As we introduce a new variable at every quantifier inference (and not more complex terms) we have that $\sum_{t\in T}|t|$ is a bound on the sum of sizes of maximal semiterms for any possible sequent. Furthermore, as there are n inferences and therefore n sequents we have that $n \cdot \sum_{t\in T}|t|$ is an upper bound on the sum of sizes of the terms in the unification problem, and by Proposition 22 an upper bound on the maximal depth of the unified terms.

(5.4) Note that $\max_{t\in T}\mathrm{dp}(t)$ is an upper bound on the depth of any semiterm occurring in the variable-distinct map $\lceil\cdot\rceil_v$. Therefore, and since m is equal to the number of variables in U we have that Proposition 22 implies $(m+1)\cdot\max_{t\in T}\mathrm{dp}(t)$. □

CHAPTER 6

Conclusion

This thesis is about gaining insights into the relationship between proofs, proof skeletons, proof nets and atomic flow graphs in the context of first-order logic and sequent calculi. We showed that it is, under certain conditions, useful to deviate from the classical definition of proofs. To be more precise, in our setting the representation of proofs as tuples allowed a straightforward definition of abstractions, and led to a first insight into the relationship between these proof structures. This tuple-based representation generates a uniform framework consisting of several abstract proof structures such that every pair of proof structures has a common abstraction. From a different point of view, this framework can be seen as a lattice where the order is given by the abstractions. The maximum and minimum of this lattice are given by the proof and the reduced atomic flow graph skeleton respectively. Note that we have this lattice only if we consider annotated weakening rules, that is, weakening rules annotated with the formula skeletons of the weakening formulas. For the standard sequent calculus we have that a proof skeleton (net skeleton) cannot be transformed into an AFG skeleton (reduced AFG skeleton).

Furthermore, we saw that the defined proof structures generate equivalence relations. These equivalence relations in combination with a so-called refinement order form a poset of equivalence relations. Therefore, if in the context of questions around the identity of proofs one introduces a new equivalence relation $\sim$ defined on proofs, one can use this poset in order to relate $\sim$ to already existing ones. We also saw a correspondence between commuting diagrams and the refinement order.

These equivalence relations in turn generate equivalence classes. We showed, on the one hand, that there exists an infinite number of proofs having the same proof skeleton. On the other hand, for proof nets and atomic flow graphs there exist finitely many corresponding proofs. Furthermore, we showed that, in general, there are infinitely many atomic flow graphs having the same topological reduct. We suggested that this would not be the case for cut-free proofs, however, a proof of that is not in the scope of this thesis.

Finally, we saw an application of the uniform framework. We generalized an algorithm defined by Krajíček and Pudlák [9] for proof skeletons to proof net skeletons. By doing so, we derived the same bounds on the minimal size of cut-free proof nets as Krajíček and Pudlák showed for cut-free proofs, and we showed that there exists a most general proof net with a given cut-free proof net skeleton.

Further work in this direction would include studying the properties of the equivalence classes generated by the abstractions. For instance, for proof nets we have that two proofs induce the same proof net if one can be obtained from the other by certain rule transpositions. A similar result would interesting for atomic flow graphs, and also for the rest of the equivalence classes.

A starting point for related work would also be a more fine-grained investigation of the cardinalities of the equivalence classes. On the one hand, the influence of cuts, that is, the investigation of the cardinalities in a cut-free setting, on the other hand, the determination of concrete bounds on the cardinalities.

Another open question concerns the algorithm for computing a most general proof net. In contrast to Krajíček and Pudlák [9], we used an ordinary unification problem without restrictions since these restrictions are dependent on the inference order. That is, there are different sets of restrictions depending on the inference order. However, does there exist a minimal set of restrictions common to all inference orders, and does this minimal set of restrictions yield a proof?

Bibliography

[1] S. R. Buss. The undecidability of k-provability. *Annals of Pure and Applied Logic*, 53(1):75–102, 1991.

[2] A. Carbone. Interpolants, cut elimination and flow graphs for the propositional calculus. *Annals of Pure and Applied Logic*, 83(3):249–299, 1997.

[3] A. Carbone. Logical structures and genus of proofs. *Annals of Pure and Applied Logic*, 161(2):139–149, 2009.

[4] A. Carbone and S. Semmes. *A graphic apology for symmetry and implicitness.* Oxford University Press, 2000.

[5] C.-L. Chang and R. C.-T. Lee. *Symbolic logic and mechanical theorem proving.* Academic press, 1973.

[6] J.-Y. Girard. Linear logic. *Theoretical Computer Science*, 50(1), 1987.

[7] J.-Y. Girard. Quantifiers in linear logic II. *Nuovi problemi della logica e della filosofia della scienza*, 2:1, 1991.

[8] W. D. Goldfarb. The undecidability of the second-order unification problem. *Theoretical Computer Science*, 13(2):225–230, 1981.

[9] J. Krajíček and P. Pudlák. The number of proof lines and the size of proofs in first order logic. *Archive for Mathematical Logic*, 27(1):69–84, 1988.

[10] G. Kreisel. Proof theory: some personal recollections. Appendix to [14].

[11] V. P. Orevkov. Reconstitution of the proof from its scheme (russian abstract). *8th Soviet Conference on Mathematical Logic*, 1984.

[12] E. Robinson. Proof nets for classical logic. *Journal of Logic and Computation*, 13 (5):777–797, 2003.

[13] L. Straßburger. What is the problem with proof nets for classical logic? In *Programs, Proofs, Processes, 6th Conference on Computability in Europe, CiE 2010. Proceedings*, volume 6158 of *LNCS*, pages 406–416. 2010.

[14] G. Takeuti. *Proof theory.* North-Holland, 2nd edition, 1987.