FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Tools for the Investigation of Substructural, Intermediate and Paraconsistent Logics

DISSERTATION

zur Erlangung des akademischen Grades

Doktorin der technischen Wissenschaften

eingereicht von

Lara Katharina Spendier
Matrikelnummer 0525611

an der
Fakultät für Informatik der Technischen Universität Wien

Betreuung: Prof. Dr. Agata Ciabattoni

Diese Dissertation haben begutachtet:

(Prof. Dr. Agata Ciabattoni) (Ass. Prof. Elaine Pimentel, PhD)

Wien,

(Lara Katharina Spendier)

FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Tools for the Investigation of Substructural, Intermediate and Paraconsistent Logics

DISSERTATION

submitted in partial fulfillment of the requirements for the degree of

Doktorin der technischen Wissenschaften

by

Lara Katharina Spendier
Registration Number 0525611

to the Faculty of Informatics
at the Vienna University of Technology

Advisor: Prof. Dr. Agata Ciabattoni

The dissertation has been reviewed by:

(Prof. Dr. Agata Ciabattoni)	(Ass. Prof. Elaine Pimentel, PhD)

Wien,

(Lara Katharina Spendier)

Erklärung zur Verfassung der Arbeit

Lara Katharina Spendier
Zirkusgasse 50/15, 1020 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.

(Ort, Datum)	(Unterschrift Verfasserin)

Abstract

Non-classical logics have gained importance in many fields of computer science, engineering and philosophy. They are often employed in applications of artificial intelligence, knowledge representation and formal verification; e.g., when it comes to reasoning in presence of vague information or inconsistencies. There are already many non-classical logics and, due to the increasing demand for such logics, new ones are introduced frequently.

Non-classical logics are often introduced or described by adding Hilbert axioms to well-known systems. The usefulness of these logics however strongly depends on the availability of analytic calculi for them. Analytic calculi are deductive systems in which proof search proceeds by a step-wise decomposition of the formulas to be proved. Such calculi play a paramount role for computational proof search and are also key to establishing essential properties for the formalized logics, like consistency or decidability. Unfortunately, introducing an analytic calculus for a particular logic often requires significant effort: a suitable framework for the calculus has to be chosen and adequate inference rules reflecting the characteristic properties of the considered logic have to be provided. Finally, soundness, completeness and analyticity of the defined calculus must be proved. Since these steps are usually tailored to the specific logic at hand and often difficult to establish, many important logics still lack an analytic calculus. Systematic and algorithmic procedures to generate analytic calculi are therefore highly desirable.

This thesis presents (theoretical and engineering) tools for the investigation of substructural, intermediate and paraconsistent propositional logics. The main contributions are:

(c1) systematic procedures for the automated generation of analytic calculi,

(c2) a further exploitation of the analytic calculi to establish important properties for the formalized logics, and

(c3) the introduction of the framework **TINC** (*Tools for the Investigation of Non-Classical logics*) to provide computer support for the contributions (c1) and (c2).

More precisely, for *substructural logics* we use the analytic hypersequent calculi generated by the procedure in [52] to check whether the corresponding logics are standard complete, i.e. they have a semantics with truth values in $[0, 1]$ (c2).

For *intermediate logics*, we present two systematic procedures (c1) that differ in their (syntactic or semantic) starting points. The first procedure is based on the syntactic

presentation of a logic in terms of a Hilbert system: we combine the method in [52] with a heuristic to transform Hilbert axioms of a certain form into equivalent (structural or logical) hypersequent rules. The second procedure is based on the semantic specification of a logic: it transforms frame conditions into equivalent labelled rules, obtaining a labelled sequent calculus for the corresponding logic.

For *paraconsistent logics*, we establish an algorithmic procedure (c1) to generate sequent calculi by transforming Hilbert axioms of a certain form into equivalent logical rules. In addition to that, we extract new semantics from the obtained calculi using the framework of partial non-deterministic matrices [23]. The introduced semantics is then used to reason about decidability of the corresponding logics and analyticity of the obtained calculi (c2).

Following the spirit of "logic engineering" [139, 140], we provide computer support (c3) to automatize the generation of analytic calculi (c1) and their utilization (c2). The tools of **TINC** — *AxiomCalc* (for substructural logics), *Framinator* (for intermediate logics) and *Paralyzer* (for paraconsistent logics) — are implemented in Prolog. They take a specification of a logic as input and return (a paper written in LaTeX containing) an analytic calculus for the logic and certain properties of the logic and/or the calculus as output.

Kurzfassung

Nichtklassische Logiken haben in vielen Gebieten der Informatik, Ingenieurswissenschaften und Philosophie an Bedeutung gewonnen und werden in Anwendungen der künstlichen Intelligenz, Wissensrepräsentation und formalen Verifikation eingesetzt. Es gibt bereits viele nichtklassische Logiken und aufgrund der stetig wachsenden Nachfrage werden neue häufig eingeführt.

Nichtklassische Logiken werden häufig eingeführt bzw. beschrieben, indem Hilbert Axiome zu bereits bekannten Systemen hinzugefügt werden. Die Brauchbarkeit dieser Logiken hängt jedoch stark davon ab, ob analytische Kalküle für sie verfügbar sind. Analytische Kalküle sind Deduktionssysteme, in welchen Formeln durch schrittweise Zerlegung der zu zeigenden Formeln bewiesen werden können. Solche Kalküle spielen eine immens wichtige Rolle im Bereich der computerunterstützten Beweissuche und sind eine grundlegende Voraussetzung für die Ermittlung essentieller Eigenschaften der formalisierten Logiken, wie beispielsweise Konsistenz oder Entscheidbarkeit. Leider ist die Definition eines analytischen Kalküls für eine bestimmte Logik mit hohem Aufwand verbunden: Zuerst muss ein passender Formalismus gewählt und danach adäquate Inferenzregeln, welche die charakteristischen Eigenschaften der Logik reflektieren, definiert werden. Abschließend müssen Korrektheit, Vollständigkeit und Analytizität für den neuen Kalkül gezeigt werden. Da diese Schritte üblicherweise auf eine bestimmte Logik zugeschnitten und auch oft schwierig zu zeigen sind, gibt es für viele wichtige Logiken noch keinen analytischen Kalkül. Systematische und algorithmische Prozeduren zur automatisierten Generierung analytischer Kalküle sind daher sehr erstrebenswert.

In dieser Dissertation präsentieren wir (theoretische und praktische) Tools für die Untersuchung von substrukturellen, intermediären und parakonsistenten Aussagenlogiken. Unsere wichtigsten Beiträge sind:

(c1) systematische Methoden zur automatisierten Generierung von analytischen Kalkülen,

(c2) weitere Anwendungen der analytischen Kalküle, um wichtige Eigenschaften der formalisierten Logiken zu zeigen, und

(c3) die Einführung des Frameworks **TINC** (*Tools for the Investigation of Non-Classical logics*) zur Computerunterstützung für die Beiträge (c1) und (c2).

Für *substrukturelle Logiken* verwenden wir analytische Hypersequenzkalküle, die mit der Prozedur in [52] generiert werden, um herauszufinden, ob die korrespondierenden Logiken standard-vollständig sind, d.h., ob die Logiken eine Semantik mit Wahrheitswerten im Intervall $[0, 1]$ besitzen (c2).

Für *intermediäre Logiken* präsentieren wir zwei systematische Prozeduren (c1), die sich in ihrem (syntaktischen oder semantischen) Ausgangspunkt unterscheiden. Die erste Methode basiert auf der syntaktischen Beschreibung einer Logik mittels Hilbert System: Wir kombinieren die Methode in [52] mit einer Heuristik, um Hilbert Axiome einer bestimmten Form in äquivalente (strukturelle oder logische) Hypersequenzregeln zu transformieren. Die zweite Methode begründet sich auf der semantischen Spezifikation einer Logik: Sie konvertiert Rahmenbedingungen in äquivalente Inferenzregeln. Auf diese Weise erhält man einen gelabelten Sequenzkalkül für die entsprechende Logik.

Für *parakonsistente Logiken* generieren wir mittels einer systematischen Prozedur (c1) Sequenzkalküle, indem Hilbert Axiome einer bestimmten Form in äquivalente logische Inferenzregeln transformiert werden. Außerdem extrahieren wir aus den generierten Kalkülen eine neue Semantik im Framework der partiellen nicht-deterministischen Matrizen [23]. Die eingeführte Semantik kann dann genutzt werden (c2), um die Entscheidbarkeit der korrespondierenden Logiken und die Analytizität der Kalküle zu erörtern.

Im Sinne des "logic engineering" [139, 140] stellen wir Computerunterstützung (c3) für die automatische Generierung von analytischen Kalkülen (c1) und ihrer weiteren Anwendung (c2) zur Verfügung. Die Tools, die zu **TINC** gehören — *AxiomCalc* (für substrukturelle Logiken), *Framinator* (für intermediäre Logiken) und *Paralyzer* (für parakonsistente Logiken) — nehmen die Spezifikation einer Logik als Input und errechnen einen analytischen Kalkül für die Logik und bestimmte Eigenschaften der Logik und/oder des Kalküls. Diese Ergebnisse werden zusammengefasst und in einem automatisch generierten LaTeX-Paper ausgegeben.

Contents

1 Introduction **1**
1.1 Motivations . . . 1
1.2 Aims of the Thesis . . . 2
1.3 Thesis Outline . . . 3
1.4 Publications . . . 4

2 Preliminaries and Background **5**
2.1 Basic Concepts for Intuitionistic Logic . . . 5
2.2 Analytic Calculi and Cut Elimination . . . 14

3 TINC: Tools for the Investigation of Non-classical Logics **19**
3.1 **TINC** in a Nutshell . . . 20
3.2 Implementation Details . . . 22
3.2.1 Overview Code Examples . . . 27
3.3 Related Work . . . 28

4 Substructural Logics **31**
4.1 Preliminaries . . . 31
4.2 Theoretical Base and Related Work in Proof Theory . . . 34
4.3 An Application: Standard Completeness for Extensions of **MTL** . . . 43
4.3.1 Tool: *AxiomCalc* . . . 50

5 Intermediate Logics **57**
5.1 Preliminaries . . . 58
5.2 Related Work in Proof Theory . . . 62
5.3 Towards the Generation of Logical Hypersequent Rules . . . 65
5.4 Towards a Systematic Procedure for Labelled Calculi . . . 78
5.4.1 Tool: *Framinator* . . . 93

6 Paraconsistent Logics **99**
6.1 Preliminaries . . . 100
6.2 Related Work in Proof Theory . . . 106
6.3 Towards Analytic Calculi for Paraconsistent Logics . . . 109
6.4 Step 1: Automated Generation of Sequent Calculi . . . 109

6.5 Step 2: Automated Extraction of Semantics . . . 116
6.6 A Special Case . . . 126
6.6.1 Tool: *Paralyzer* . . . 135

7 Conclusion **143**
7.1 Summary . . . 143
7.2 Some Open Questions . . . 145

A Substructural Logics **149**

B Paraconsistent Logics **155**

C TINC: Tool Output **159**
C.1 Example LaTeX-output of *AxiomCalc* . . . 159
C.2 Example LaTeX-output of *Framinator* . . . 165
C.3 Example LaTeX-output of *Paralyzer* . . . 169

Bibliography **177**

CHAPTER 1

Introduction

1.1 Motivations

Logic is a system of reasoning and the basis of applications in various fields. Some applications call for systems of reasoning different from the usual classical logic; such logics are called *non-classical logics*. Non-classical logics are often employed in applications of artificial intelligence, knowledge representation and formal verification: for example, fuzzy logics are used for the underlying inference mechanism of medical expert systems [148, 29, 166]; paraconsistent logics have proven to be a successful tool when handling contradicting information [36, 156]; substructural logics provide adequate languages for modeling dynamic data structures or resources [4, 90, 83].

Due to the increasing demand for non-classical logics in the fields of engineering, computer science or philosophy, new non-classical logics are frequently introduced. Before these logics can however be used in potential applications, they must be studied and well understood. A standard approach to studying a logic is to investigate its valid statements (or *theorems*), which are *proved* from axioms via inference rules. Axioms and inference rules compose a *proof system* (or *calculus*) for a logic. For any logic, there can be several proof systems with different features that prove the same theorems. However, not every proof system is useful when it comes to the development of automated reasoning methods or studying essential mathematical properties of the formalized logic, like consistency or decidability. Many proof systems that are used to describe logics are formulated in the framework due to Hilbert and Frege. Such systems consist of a large number of axioms and, in most cases, a single inference rule called *modus ponens*. Roughly speaking, modus ponens allows us to derive a formula ψ from the formulas φ and $\varphi \supset \psi$ ("φ implies ψ"). A non-classical logic is then often formalized by adding, omitting, or modifying certain axioms of a Hilbert system for classical logic.

A Hilbert calculus is a convenient proof system for describing a logic, but since it uses the rule of modus ponens, it is not an analytic calculus. One of the traditional formalisms aiming to overcome this deficiency is Gentzen's framework of sequent calculi [88]. A sequent calculus operates on sequents, which are pairs of multisets of formulas, and often consists of few axioms and a large number of inference rules. An important rule in the sequent calculus is the *cut rule*, which is a reformulation of modus ponens and corresponds to the introduction of intermediate steps (lemmas) in proofs. This rule however often breaks the analyticity property of a sequent calculus, which can be restored by showing that the cut rule is admissible in (or eliminable from) the calculus.

Many important non-classical logics have been successfully formalized using analytic sequent calculi. However, this framework does not seem to be expressive enough to find analytic sequent systems for many other interesting and useful non-classical logics. As a result, a large range of generalizations of the sequent calculus has been introduced to define analytic calculi for logics lacking analytic sequent systems; these include hypersequent calculus [9, 150], labelled sequent calculus [83, 167], or display calculus [34]. The construction of an analytic calculus for a logic then traditionally consists of the following three steps:

(i) **Choose or define a suitable framework.**

(ii) **Find adequate rules that formalize the logic under consideration.** A calculus for the logic is defined within the framework chosen in step (i). This is usually done by finding adequate inference rules that reflect the characteristic properties of the considered logic (e.g., the Hilbert axioms formalizing the logic). Finally, soundness and completeness proofs of the new calculus with respect to the (system formalizing the) logic are required, where it is shown that both systems prove the exact same theorems.

(iii) **Prove analyticity of the defined calculus.** Analyticity is usually obtained as a direct corollary of the cut elimination theorem, which states that the cut rule can be removed from the calculus without weakening the system.

These three steps are usually done ad-hoc for the particular logic under consideration and each new calculus requires its own proofs of soundness, completeness and analyticity. This is required even for calculi obtained by adding similar rules to different "base" systems. The construction of an analytic calculus for a logic is thus often an error-prone and cumbersome task. Algorithmic procedures to obtain analytic calculi in a systematic and uniform way for large classes of logics are therefore highly desirable. These would provide a theoretical basis for the development of efficient automated reasoning tools, and permit the implementation of the otherwise tedious and time-consuming task of finding an analytic calculus.

1.2 Aims of the Thesis

In this PhD thesis, we focus on *substructural*, *intermediate* and *paraconsistent* propositional logics. The overall aim is the development of theoretical and engineering tools for their investigation. More precisely, the goals of this PhD project are threefold:

1. We strive to find systematic and uniform procedures for the automated generation of analytic calculi for large classes of substructural, intermediate and paraconsistent propositional logics.
2. We use the obtained calculi to establish important properties for many logics in a uniform way.
3. We develop the framework **TINC** (*Tools for the Investigation of Non-Classical logics*) to provide computer support for 1. and 2. by exploiting the algorithmic and systematic nature of our procedures and implementing them as tools.

This thesis is a concrete step towards a systematic investigation of non-classical logics and the development of theoretical and engineering tools for designing new application-oriented logics.

1.3 Thesis Outline

The remainder of this thesis is organized as follows.

In Chapter 2, we start by recalling propositional intuitionistic logic and use it to settle the basic definitions and notions that will be used throughout this thesis. We discuss the notion of "analytic calculus" and recall a method to prove cut elimination for sequent and hypersequent calculi.

Chapter 3 contains the design of our system **TINC** (*Tools for the Investigation of Non-Classical logics*) and the description of a general approach to introduce analytic calculi, which is based on the method introduced in [52]. Moreover, we give a short introduction to Prolog to provide a basic understanding of the code examples in the following chapters.

In Chapter 4, we focus on substructural logics, which are logics that lack some (or all) of the structural rules when formalized as sequent systems. We first describe the systematic procedure in [52] that transforms Hilbert axioms of a certain form into equivalent *structural* sequent or hypersequent rules. Using this method, analytic calculi can be obtained for a large class of substructural logics. Then we consider the subclass of the introduced calculi for axiomatic extensions of monoidal t-norm logic **MTL** [78]. We show how these calculi can be further utilized to check whether the corresponding logics are standard complete, i.e., complete with respect to algebras based on truth values in $[0, 1]$ (and hence, whether the corresponding logics are a fuzzy logic in the sense of [97]). The chapter also contains the description of the tool *AxiomCalc*, which implements our theoretical result.

Chapter 5 focuses on intermediate logics, which are logics that lie between intuitionistic and classical logic. We present two approaches for the introduction of analytic calculi, which are distinguished by their (syntactic or semantic) starting points. The first approach is based on the syntactic presentation of a logic in terms of a Hilbert system. We combine the automated procedure in [52] with a heuristic to transform Hilbert axioms of a certain form into equivalent *logical* hypersequent rules. An instance of this method gives a first analytic hypersequent calculus for the logic $\mathbf{Bd}_2$, i.e. the logic of frames with bounded depth at most 2 [49], for which we present ad-hoc proofs of soundness, completeness and cut elimination. The second approach is based on the semantic specification of a logic, which is obtained by imposing conditions on standard intuitionistic Kripke frames. We introduce a systematic procedure, adapting the one in [52] for labelled sequent calculi, to obtain cut-free calculi for large classes of intermediate logics. The method generalizes the results presented in [76] and is subsumed by the new results in [133]. The chapter also contains a description of the implementation of the second procedure in the tool *Framinator*.

Chapter 6 contains an algorithmic procedure to generate sequent calculi for a large class of paraconsistent logics, which are logics that are not trivialized in the presence of inconsistency. In addition to that, we extract semantics in the framework of partial non-deterministic matrices [23] from the obtained calculi and use it to reason about the analyticity of the calculi and the decidability of the logics. We also provide a description of the tool *Paralyzer* that implements our transformation procedure and the extraction of the semantics for a specific subclass of paraconsistent logics.

Chapter 7 summarizes the results of the thesis and discusses future research directions.

1.4 Publications

This thesis is based on the following publications:

1. Agata Ciabattoni, Ori Lahav, Lara Spendier and Anna Zamansky. Taming Paraconsistent (and Other) Logics: An Algorithmic Approach. *ACM Transactions on Computational Logic (TOCL)* 16(1):5:1–5:23, 2015.
2. Agata Ciabattoni and Lara Spendier. Tools for the Investigation of Substructural and Paraconsistent Logics. In *Proceedings of the European Conference on Logics in Artifical Intelligence (JELIA 2014)*, volume 8761 of *LNAI*, pages 18–32, 2014.
3. Agata Ciabattoni, Paolo Maffezioli and Lara Spendier. Hypersequent and Labelled Calculi for Intermediate Logics. In *Proceedings of TABLEAUX 2013*, volume 8123 of *LNCS*, pages 81–96, 2013.
4. Agata Ciabattoni, Ori Lahav, Lara Spendier and Anna Zamansky. Automated Support for the Investigation of Paraconsistent and Other Logics. In *Proceedings of the Symposium on Logical Foundations in Computer Science (LFCS 2013)*, volume 7734 of *LNCS*, pages 119–133, 2013.
5. Paolo Baldi, Agata Ciabattoni and Lara Spendier. Standard Completeness for Extensions of **MTL**: an Automated Approach. In *Proceedings of International Workshop on Logic, Language, Information and Computation (WoLLIC 2012)*, volume 7456 of *LNCS*, pages 154–167, 2012.

CHAPTER 2

Preliminaries and Background

Before advancing to the results of the thesis, we establish a common ground by introducing the basic concepts. The notions that are specific for a certain part are defined in the respective chapters.

We start by recalling propositional intuitionistic logic (abbreviated as **Int**) and some proof systems for it. In particular, we focus on the Gentzen sequent calculus and on the hypersequent calculus, as these are the main formalisms used in this thesis. In the second part of this chapter, we discuss the notion of "analytic calculus" and recall a cut elimination method.

For the basic notions of this chapter, we refer to the following standard references in proof theory [43, 49, 134, 160, 165].

2.1 Basic Concepts for Intuitionistic Logic

Let us start by introducing the language $\mathcal{L}_{int}$ of propositional intuitionistic logic, which consists of infinitely many (possibly indexed) propositional variables $p, q, \ldots$, the binary connectives $\wedge$ (conjunction), $\vee$ (disjunction), $\supset$ (implication) and the constant $\bot$ for falsity. Using this language, we can now define the formulas and subformulas of **Int**:

Definition 1. *Atomic* formulas are (possibly indexed) propositional variables $p, q, \ldots$. A *formula* of **Int** is then defined inductively:

1. Every atomic formula and the logical constant $\bot$ is a formula.
2. If φ and ψ are formulas, then $\varphi \wedge \psi$, $\varphi \vee \psi$ and $\varphi \supset \psi$ are formulas.

As usual, we abbreviate $\bot \supset \bot$ with $\top$ and $\varphi \supset \bot$ with $\neg\varphi$.

We adopt standard conventions for omitting brackets from formulas by assuming that unary operators have the highest precedence, followed by the binary connectives (from highest to lowest precedence) $\wedge$, $\vee$, and $\supset$.

We denote (metavariables for) formulas by (possibly indexed) $\varphi, \psi, \chi, \alpha, \beta, \ldots$ and (metavariables for) finite (possibly empty) multisets of formulas by (possibly indexed) $\Gamma, \Delta, \Sigma, \Pi, \ldots$.

Note that all axioms considered in this thesis are in fact axiom schemas, where $\varphi, \psi, \chi, \alpha, \beta, \ldots$ denote metavariables for formulas which are substituted by any formula in the instances of the schemas.

We use the following notation for formulas and (metavariables for) multisets of formulas with $n \geq 0$:

$$\varphi^n = \overbrace{\{\varphi, \ldots, \varphi\}}^{n} \text{ and } \Sigma^n = \overbrace{\{\Sigma, \ldots, \Sigma\}}^{n}$$

Definition 2. *Subformulas* of a formula φ are defined by

1. φ is a subformula of φ, and
2. if $\psi \diamond \chi$ is a subformula of φ, then ψ and χ are subformulas of φ for $\diamond \in \{\vee, \wedge, \supset\}$.

A *Hilbert-style calculus* consists of a set of axioms and inference rules of the form

$$\frac{\varphi_1 \quad \cdots \quad \varphi_n}{\psi}$$

where $\varphi_1, \ldots, \varphi_n, \psi$ are *metavariables* for formulas (or *schemas*). $\varphi_1, \ldots, \varphi_n$ are the *premises* and ψ is the *conclusion* of the rule, which is inferred from the premises. Note that a Hilbert system usually consists of a small number of inference rules and a high number of (schematic) axioms.

We define the notion of derivability of formulas in a Hilbert system C_H as follows:

Definition 3. A *derivation* of a formula φ in a Hilbert-style calculus is a sequence $\varphi_1, \ldots, \varphi_n$ of formulas such that $\varphi_n = \varphi$ and for every $i, 1 \leq i \leq n$, φ_i is either an axiom or obtained from some of the preceding formulas in the sequence by one of the inference rules.

We use the symbol $\vdash_{C_H}$ to denote *derivability* in the Hilbert calculus C_H. For the derivability of the formula φ in C_H, we write

$$\vdash_{C_H} \varphi$$

By using the Hilbert-style calculus C_H, we can now syntactically specify a *logic* as set of theorems, i.e. as sets of formulas containing the axioms of C_H and closed under the rules of C_H and substitution.

We define propositional intuitionistic logic **Int** in terms of the Hilbert-style calculus Int_H as follows [49]:

(Schematic) Axioms

(A1) $\varphi \supset (\psi \supset \varphi)$
(A2) $(\varphi \supset (\psi \supset \chi)) \supset ((\varphi \supset \psi) \supset (\varphi \supset \chi))$
(A3) $\varphi \wedge \psi \supset \varphi$
(A4) $\varphi \wedge \psi \supset \psi$

(A5) $\varphi \supset (\psi \supset \varphi \wedge \psi)$
(A6) $\varphi \supset \varphi \vee \psi$
(A7) $\psi \supset \varphi \vee \psi$
(A8) $(\varphi \supset \chi) \supset ((\psi \supset \chi) \supset (\varphi \vee \psi \supset \chi))$
(A9) $\bot \supset \varphi$

Inference rules

- *modus ponens* (MP): for given formulas φ and $\varphi \supset \psi$, we obtain ψ; $\dfrac{\varphi \quad \varphi \supset \psi}{\psi}$

Hilbert-style calculi are a convenient way for describing logics. However, they are not optimal for practical use such as proof search or theorem proving. The following example shows that finding a derivation for the formula $\varphi \supset \varphi$ in Int_H is not straightforward and requires some ingenuity.

Example 1. *We show $\vdash_{Int_H} \varphi \supset \varphi$. In the right column of the following derivation (read top-down), we either refer to the axiom that is used as assumption or to an application of modus ponens (MP(x, y), where x and y denote the row number).*

1.	$\varphi \supset (\varphi \supset \varphi)$	(A1)
2.	$\varphi \supset ((\varphi \supset \varphi) \supset \varphi)$	(A1)
3.	$(\varphi \supset ((\varphi \supset \varphi) \supset \varphi)) \supset ((\varphi \supset (\varphi \supset \varphi)) \supset (\varphi \supset \varphi))$	(A2)
4.	$(\varphi \supset (\varphi \supset \varphi)) \supset (\varphi \supset \varphi)$	MP(2,3)
5.	$\varphi \supset \varphi$	MP(1,4)

Sequents and Sequent Calculus

The sequent calculus was introduced by Gentzen [88] as a formalism for studying proofs in classical and intuitionistic logic. It operates on structures that are called sequents, which are defined as follows:

Definition 4. A *sequent* is an expression of the form $\Gamma \Rightarrow \Delta$ where Γ and Δ are finite multisets[1] of formulas. We call Γ the *antecedent* and Δ the *succedent* of the sequent. If the succedent of a sequent contains at most one formula, it is called *single-conclusion*, and *multiple-conclusion*, otherwise.

Intuitively, a sequent is understood as the implication from the conjunction of all formulas in the antecedent to the disjunction of all formulas in the succedent:

Definition 5. In intuistionistic logic **Int**, a sequent

$$S = \Gamma \Rightarrow \Delta$$

[1] Gentzen originally used sequences of formulas.

is interpreted as

$$\mathcal{I}(S) = \bigwedge \Gamma \supset \bigvee \Delta$$

where $\bigwedge \Gamma$ stands for the conjunction of the formulas in Γ ($\top$ when Γ is empty), and $\bigvee \Delta$ stands for the disjunction of the formulas in Δ ($\bot$ when Δ is empty).

A sequent calculus consists of axioms and rules. The rules are actually *rule schemas* and an instance of a rule is a *rule application* or *inference*. A rule is written

$$\frac{S_1 \quad \cdots \quad S_n}{S}$$

which means that a sequent S is inferred from the sequents $S_1, \ldots, S_n$. As in Hilbert-style calculi, we call the sequent S the *conclusion* of the rule while $S_1, \ldots, S_n$ are the *premises*. The notion of a derivation (or *proof*) is then also extended to sequents:

Definition 6. A *derivation* in a sequent calculus is a finite labelled tree with nodes labelled by sequents and a single root (called *end sequent*), axioms at the top nodes, and where each node is connected with the (immediate) successor nodes (if any) according to the inference rules. Let C be a sequent calculus and R be a set of rules. We write $C + R$ to denote the sequent calculus C extended with R. For sequents derived in $C + R$ we write

$$\vdash_{C+R} \Gamma \Rightarrow \Delta$$

$\vdash_{C+R} \varphi$ is defined as $\vdash_{C+R} \Rightarrow \varphi$, i.e., the sequent $\Gamma \Rightarrow \varphi$ with Γ being empty. If a sequent S_0 is derivable from a set of sequents $\mathcal{S}$ in $C + R$, we write

$$\mathcal{S} \vdash_{C+R} S_0$$

We can now define the *equivalence* between a sequent calculus C and a Hilbert calculus C_H by their derivable formulas:

Definition 7. A sequent calculus C is *equivalent* to a Hilbert system C_H if for every finite set $\Gamma \cup \{\varphi\}$ of formulas: φ is provable in C_H from Γ (in symbols $\Gamma \vdash_{C_H} \varphi$) iff $\Gamma \Rightarrow \varphi$ is provable in C (in symbols $\vdash_C \Gamma \Rightarrow \varphi$).

The connection between a logic and a sequent calculus is then established by *soundness* and *completeness*:

Definition 8. A sequent calculus C is *sound* for **Int** if for every sequent $\Gamma \Rightarrow \Delta$, we have

$$\vdash_C \Gamma \Rightarrow \Delta \quad \text{implies} \quad \mathcal{I}(\Gamma \Rightarrow \Delta) \in \mathbf{Int}$$

Similarly, a sequent calculus C is *complete* for **Int** if for every sequent $\Gamma \Rightarrow \Delta$, we have

$$\mathcal{I}(\Gamma \Rightarrow \Delta) \in \mathbf{Int} \quad \text{implies} \quad \vdash_C \Gamma \Rightarrow \Delta$$

We will also need the following notions of the *height* of a derivation and the *complexity* of a formula:

$\varphi \Rightarrow \varphi$	$\dfrac{\Gamma \Rightarrow \varphi_i}{\Gamma \Rightarrow \varphi_1 \vee \varphi_2}\ (\vee_i, r)$	$\dfrac{\Gamma, \varphi \Rightarrow \psi}{\Gamma \Rightarrow \varphi \supset \psi}\ (\supset, r)$	$\dfrac{\Gamma \Rightarrow \varphi \quad \Gamma, \psi \Rightarrow \Pi}{\Gamma, \varphi \supset \psi \Rightarrow \Pi}\ (\supset, l)$
$\bot \Rightarrow \varphi$	$\dfrac{\Gamma, \varphi_i \Rightarrow \Pi}{\Gamma, \varphi_1 \wedge \varphi_2 \Rightarrow \Pi}\ (\wedge_i, l)$	$\dfrac{\Gamma \Rightarrow \varphi \quad \Gamma \Rightarrow \psi}{\Gamma \Rightarrow \varphi \wedge \psi}\ (\wedge, r)$	$\dfrac{\Gamma, \varphi \Rightarrow \Pi \quad \Gamma, \psi \Rightarrow \Pi}{\Gamma, \varphi \vee \psi \Rightarrow \Pi}\ (\vee, l)$
$\dfrac{\Gamma \Rightarrow}{\Gamma \Rightarrow \varphi}\ (w, r)$	$\dfrac{\Gamma \Rightarrow \Pi}{\Gamma, \varphi \Rightarrow \Pi}\ (w, l)$	$\dfrac{\Gamma, \varphi, \varphi \Rightarrow \Pi}{\Gamma, \varphi \Rightarrow \Pi}\ (c, l)$	$\dfrac{\Gamma \Rightarrow \varphi \quad \varphi, \Delta \Rightarrow \Pi}{\Gamma, \Delta \Rightarrow \Pi}\ (cut)$

$i \in \{1, 2\}$; note that Γ, Δ are metavariables for multisets of formulas and Π is a metavariable for a multiset containing at most one formula.

Table 2.1: Sequent calculus *LJ* for intuitionistic logic **Int**

Definition 9. The *height* $|d|$ of a derivation d is the maximal number of inference rules + 1 occurring on any branch of d. The *complexity* $|\varphi|$ of a formula φ is defined as:

1. $|\varphi| = 0$ if φ is atomic and
2. $|\varphi \wedge \psi| = |\varphi \vee \psi| = |\varphi \supset \psi| = max(|\varphi|, |\psi|) + 1$

The first sequent calculus for propositional intuitionistic logic, *LJ*, was introduced by Gentzen in [88]. It is a single-conclusion calculus. The axioms and rules (which are, in fact, schemas) of *LJ* are depicted in Table 2.1.

Before explaining the different types of rules of a sequent calculus, we show that proving theorems in this type of calculi is indeed easier than in a Hilbert-style calculus.

Example 2. *We show* $\vdash_{LJ} \Rightarrow \neg\neg\neg\varphi \supset \neg\varphi$*:*

$$\dfrac{\dfrac{\dfrac{\dfrac{\bot \Rightarrow \bot}{\bot, \varphi \Rightarrow \bot}\ (w,l) \qquad \dfrac{\dfrac{\dfrac{\bot \Rightarrow \bot}{\bot, \varphi \Rightarrow \bot}\ (w,l) \quad \varphi \Rightarrow \varphi}{\varphi, \neg\varphi \Rightarrow \bot}\ (\supset, l)}{\varphi \Rightarrow \neg\neg\varphi}\ (\supset, r)}{\neg\neg\neg\varphi, \varphi \Rightarrow \bot}\ (\supset, l)}{\neg\neg\neg\varphi \Rightarrow \neg\varphi}\ (\supset, r)}{\Rightarrow \neg\neg\neg\varphi \supset \neg\varphi}\ (\supset, r)$$

Note that in this proof (read bottom-up), the end sequent is essentially *decomposed* into its subformulas by using rules matching the outermost logical connectives until we reach an initial axiom.

In a sequent calculus, we distinguish logical rules, structural rules and the cut rule. *Logical rules* are rules that introduce a logical connective. The logical rules of *LJ* are $(\supset, l)$, $(\supset, r)$, $(\wedge_i, l)$, $(\wedge, r)$, $(\vee, l)$, $(\vee_i, r)$. Consider for example the following rule introducing $\supset$ on the left:

$$\dfrac{\Gamma \Rightarrow \varphi \qquad \Gamma, \psi \Rightarrow \Pi}{\Gamma, \varphi \supset \psi \Rightarrow \Pi}\ (\supset, l)$$

The formula $\varphi \supset \psi$ in the conclusion of the rule is called *principal formula*. The formulas φ and ψ in the premises of the rule, from which the principal formula derives, are the *active formulas*. The formulas that remain unchanged by the rule application, i.e. the formulas in Γ, Π, are referred to as the left and right *contexts* of the rule. We use analogous terminology for the other rules.

For every logical connective in the language, there usually exist a left and a right rule depending on whether a connective is introduced on the left or the right side of the sequent. Note that in these rules, a proof of the premises implies a proof of the conclusion. For some rules, the converse also holds, namely a proof of the conclusion implies a proof of the premises. Such rules are called *invertible*:

Definition 10. Let $S_1, \ldots, S_n, S$ be sequents and r a sequent rule of *LJ*. r is *invertible* if for each instance

$$\frac{S_1 \quad \cdots \quad S_n}{S}$$

of r, whenever $\vdash_{LJ} S$, then $\vdash_{LJ} S_i$ for $i = 1, \ldots, n$.

Structural rules do not introduce a logical connective. Examples for such rules are the rules for weakening ((w, l) and (w, r) in Table 2.1), contraction ((c, l) in Table 2.1) or exchange, e.g.:

$$\frac{\Gamma, \psi, \varphi \Rightarrow \Pi}{\Gamma, \varphi, \psi \Rightarrow \Pi}\,(e, l)$$

Note that in this thesis, we only consider commutative logics and we are using multisets (and not sequences) of formulas. The exchange rules are hence superfluous and are omitted in our calculi (see e.g. in *LJ*).

Finally, there is the crucial *cut rule*, which introduces a formula in the premises that does not necessarily occur in the conclusion of the rule (the *cut formula*):

$$\frac{\Gamma \Rightarrow \varphi \qquad \varphi, \Delta \Rightarrow \Pi}{\Gamma, \Delta \Rightarrow \Pi}\,(cut)$$

Cut corresponds to the introduction of intermediate steps (lemmas) into proofs, see e.g. the following example:

Example 3. *We show* $\vdash_{LJ} \neg\neg\neg\varphi \supset (\psi \supset \neg\varphi \wedge \psi)$. *To shorten the proof, we use a cut with cut formula* $\neg\neg\neg\varphi \supset \neg\varphi$ *as follows:*

$$\dfrac{\dfrac{\dfrac{\dfrac{\neg\neg\neg\varphi \Rightarrow \neg\neg\neg\varphi \qquad \dfrac{\neg\varphi \Rightarrow \neg\varphi \qquad \psi \Rightarrow \psi}{\neg\varphi, \psi \Rightarrow \neg\varphi \wedge \psi}\,(\wedge, r)}{\neg\neg\neg\varphi \supset \neg\varphi, \neg\neg\neg\varphi, \psi \Rightarrow \neg\varphi \wedge \psi}\,(\supset, l)}{\neg\neg\neg\varphi \supset \neg\varphi, \neg\neg\neg\varphi \Rightarrow \psi \supset \neg\varphi \wedge \psi}\,(\supset, r)}{\neg\neg\neg\varphi \supset \neg\varphi \Rightarrow \neg\neg\neg\varphi \supset (\psi \supset \neg\varphi \wedge \psi)}\,(\supset, r) \qquad \begin{array}{c}(\textit{Example 2}) \\ \vdots \\ \Rightarrow \neg\neg\neg\varphi \supset \neg\varphi\end{array}}{\Rightarrow \neg\neg\neg\varphi \supset (\psi \supset \neg\varphi \wedge \psi)}\,(cut)$$

Note also that the cut rule can simulate modus ponens (MP):

Example 4. *The rule modus ponens allows for given formulas φ and $\varphi \supset \psi$ to obtain ψ:*

$$\frac{\varphi \quad \varphi \supset \psi}{\psi}$$

To show that modus ponens can be simulated by the cut rule, we only need to derive $\Rightarrow \psi$ from $\Rightarrow \varphi$ and $\Rightarrow \varphi \supset \psi$:

$$\dfrac{\Rightarrow \varphi \supset \psi \qquad \dfrac{\Rightarrow \varphi \qquad \dfrac{\varphi \Rightarrow \varphi \qquad \psi \Rightarrow \psi}{\varphi, \varphi \supset \psi \Rightarrow \psi}\,(\supset, l)}{\varphi \supset \psi \Rightarrow \psi}\,(cut)}{\Rightarrow \psi}\,(cut)$$

Even though the cut rule is useful to shorten proofs, an important result is the possibility to actually *remove* (or eliminate) all instances of the cut rule from derivations. For more details on cut elimination in sequent calculi see Section 2.2.

Finally, we also need the notion of equivalent rules in a sequent calculus (we present it for LJ, but it is straightforwardly generalized to other calculi):

Definition 11. Two rules r and r' are *equivalent* (in LJ) if the derivability relations $\vdash_{LJ+r}$ and $\vdash_{LJ+r'}$ coincide, i.e., when the conclusion of r is derivable from its premises in $LJ+r'$ (and the conclusion of r' is derivable from its premises in $LJ+r$). The definition naturally extends to sets of rules.

Hypersequents and Hypersequent Calculus

Introduced by Avron in [9] (and, independently, by Pottinger in [150]), the *hypersequent calculus* is a simple and natural generalization of the sequent calculus. The hypersequent calculus does not operate on sequents but on *hypersequents*, which are finite multisets of sequents:

Definition 12. A *hypersequent* is a finite multiset $\Gamma_1 \Rightarrow \Delta_1 \mid \cdots \mid \Gamma_n \Rightarrow \Delta_n$ where each $\Gamma_i \Rightarrow \Delta_i, i = 1, \ldots, n$ is a sequent, called a *component* of the hypersequent. A hypersequent is *single-conclusion* if all of its components are single-conclusion and it is *multiple-conclusion* otherwise.

The new structural connective "|" that is used to separate the sequents is usually interpreted as disjuction at the meta-level.

Definition 13. In intuitionistic logic **Int**, a hypersequent

$$G = \Gamma_1 \Rightarrow \Delta_1 \mid \cdots \mid \Gamma_n \Rightarrow \Delta_n$$

is interpreted as:

$$\mathcal{I}(G) = (\bigwedge \Gamma_1 \supset \bigvee \Delta_1) \vee \cdots \vee (\bigwedge \Gamma_n \supset \bigvee \Delta_n)$$

$$\varphi \Rightarrow \varphi \qquad \frac{G \mid \Gamma \Rightarrow \varphi_i}{G \mid \Gamma \Rightarrow \varphi_1 \vee \varphi_2}\,(\vee_i, r) \qquad \frac{G \mid \Gamma, \varphi \Rightarrow \Pi \quad G \mid \Gamma, \psi \Rightarrow \Pi}{G \mid \Gamma, \varphi \vee \psi \Rightarrow \Pi}\,(\vee, l)$$

$$\bot \Rightarrow \varphi \qquad \frac{G \mid \Gamma, \varphi \Rightarrow \psi}{G \mid \Gamma \Rightarrow \varphi \supset \psi}\,(\supset, r) \qquad \frac{G \mid \Gamma \Rightarrow \varphi \quad G \mid \Gamma, \psi \Rightarrow \Pi}{G \mid \Gamma, \varphi \supset \psi \Rightarrow \Pi}\,(\supset, l)$$

$$\frac{G \mid \Gamma \Rightarrow}{G \mid \Gamma \Rightarrow \varphi}\,(w, r) \qquad \frac{G \mid \Gamma, \varphi, \psi \Rightarrow \Pi}{G \mid \Gamma, \varphi \wedge \psi \Rightarrow \Pi}\,(\wedge, l) \qquad \frac{G \mid \Gamma \Rightarrow \varphi \quad G \mid \Gamma \Rightarrow \psi}{G \mid \Gamma \Rightarrow \varphi \wedge \psi}\,(\wedge, r)$$

$$\frac{G \mid \Gamma \Rightarrow \Pi}{G \mid \Gamma, \varphi \Rightarrow \Pi}\,(w, l) \qquad \frac{G \mid \Gamma, \varphi, \varphi \Rightarrow \Pi}{G \mid \Gamma, \varphi \Rightarrow \Pi}\,(c, l) \qquad \frac{G \mid \Gamma \Rightarrow \varphi \quad H \mid \varphi, \Delta \Rightarrow \Pi}{G \mid H \mid \Gamma, \Delta \Rightarrow \Pi}\,(cut)$$

$$\frac{G}{G \mid \Gamma \Rightarrow \Pi}\,(ew) \qquad \frac{G \mid \Gamma \Rightarrow \Pi \mid \Gamma \Rightarrow \Pi}{G \mid \Gamma \Rightarrow \Pi}\,(ec)$$

$i \in \{1, 2\}$; note that Γ, Δ are metavariables for multisets of formulas and Π, Π' are metavariables for a multiset containing at most one formula.

Table 2.2: Hypersequent calculus *HLJ* for intuitionistic logic **Int**

where $\bigwedge \Gamma_i$ is the conjunction $\wedge$ of the formulas in Γ_i ($\top$ when Γ_i is empty), and $\bigvee \Delta_i$ is the disjunction of the formulas in Δ_i ($\bot$ when Δ_i is empty).

As in the case of sequent calculus, the rules of a hypersequent calculus consist of axioms, logical rules, structural rules and the cut rule. Axioms and the cut rule are essentially the same as in the sequent calculus. The only difference is that in the cut rule (possibly empty) side hypersequents G, H may occur. Structural and logical rules are divided into *internal* and *external* rules. Internal rules deal with formulas within one component of a hypersequent. Examples for internal structural rules are the structural rules for weakening and contraction of *LJ* (augmented with the side hypersequent G). External rules manipulate the components of a hypersequent, see for example the rules for external weakening (ew) and external contraction (ec) in Table 2.2.

HLJ is a single-conclusion hypersequent calculus for **Int**. Its axioms and rules are depicted in Table 2.2.

The definitions of derivation, soundness and completeness are naturally extended to hypersequent calculi. We can also establish equivalence between hypersequent rules (in *HLJ*):

Definition 14. Let $\mathcal{S}$ be a set of sequents and S_0 be a sequent. Two hypersequent rules r and r' are *equivalent* in *HLJ* if the derivability relations $\vdash_{HLJ+r}$ and $\vdash_{HLJ+r'}$ coincide when restricted to sequents[2]: $\mathcal{S} \vdash_{HLJ+r} S_0$ iff $\mathcal{S} \vdash_{HLJ+r'} S_0$ for any set $\mathcal{S} \cup \{S_0\}$ of sequents.

[2]We restrict the derivability to sequents because we are mainly interested in the derivability relations of formulas (and sequents).

In hypersequent calculi, we can define rules acting on several components of one or more hypersequents in parallel. This type of rules increases the expressive power of hypersequent calculi compared to ordinary sequent calculi. An example for such a rule is the communication rule (*com*) introduced by Avron in [10]:

$$\frac{G \mid \Gamma, \Delta \Rightarrow \Pi \quad G \mid \Gamma', \Delta' \Rightarrow \Pi'}{G \mid \Gamma, \Gamma' \Rightarrow \Pi \mid \Delta, \Delta' \Rightarrow \Pi'} \ (com)$$

Indeed, by adding (*com*) to the calculus *HLJ* (see Table 2.2), we can prove the prelinearity axiom $(\varphi \supset \psi) \vee (\psi \supset \varphi)$ in this calculus by an application of this rule[3]:

Example 5. *We show* $\vdash_{HLJ+(\mathrm{com})} (\varphi \supset \psi) \vee (\psi \supset \varphi)$ *(we indicate several applications of a rule by a double line):*

$$\dfrac{\dfrac{\dfrac{\dfrac{\varphi \Rightarrow \varphi \qquad \psi \Rightarrow \psi}{\varphi \Rightarrow \psi \mid \psi \Rightarrow \varphi}\ (com)}{\Rightarrow \varphi \supset \psi \mid \Rightarrow \psi \supset \varphi}\ (\supset, r)}{\Rightarrow (\varphi \supset \psi) \vee (\psi \supset \varphi) \mid \Rightarrow (\varphi \supset \psi) \vee (\psi \supset \varphi)}\ (\vee_i, r)}{\Rightarrow (\varphi \supset \psi) \vee (\psi \supset \varphi)}\ (ec)$$

Other Formalisms

We briefly mention some other generalizations of Gentzen's sequent calculus, which can be divided into *syntactic* and *semantic* formalisms. In the syntactic formalisms, sequents are generalized by allowing extra structural connectives in addition to sequents' comma; in the semantic formalisms, the semantic language is explicit part of the syntax in sequents and rules. Nested sequent calculi and display calculi belong to the former class, while labelled sequent calculi belong to the latter class of proof-theoretic frameworks.

Labelled Sequent Calculus. Labelled sequents generalize ordinary sequents by prefixing the formulas in a sequent with indices or labels. The design of *labelled sequent calculi* [83, 167] allows us to incorporate the relational semantics of the logic in the syntax. This is achieved by (a) adding a label to each formula, where each label represents a possible world, and (b) using relational formulas that define the underlying relational semantics of the logic. For more detail on labelled sequent calculi see Section 5.1.

Nested Sequent Calculus. *Nested sequents* allow sequents to appear nested within other sequents. Rules of the nested sequent calculus can then be applied at any depth. Nested sequent calculi were first introduced by Kashima in [109]. Variations of these calculi became popular quite recently [41, 42, 94, 2, 81, 159].

Tree hypersequents were introduced by Poggiolesi in [149]. They are defined by extending the syntax of hypersequents with two new symbols and by assigning importance

[3]Note that by extending *HLJ* with the (*com*)-rule, we obtain a calculus for Gödel-Dummet logic **GD** [91, 71, 10].

to the order of the sequents. Tree hypersequents internalize the structure of tree-frames of Kripke semantics. *Labelled tree sequents* are an instance of labelled sequents. Their underlying graph structure is restricted to a tree. Nested sequents, tree hypersequents and labelled tree sequents turned out to be notational variants of each other, as shown in [95].

Display Calculus. Introduced by Belnap in [34], the display calculus is a powerful generalization of the sequent calculus which has been used to capture a large variety of different logics [110, 169, 93, 170, 40].

In a display calculus, we use many new structural connectives in addition to the structural connective ',' (comma) of sequent calculus. Display calculus rules rules do not operate only on formulas, but also on structures, which are built from formulas and structure constants by using these new structural connectives. A distinctive feature of display calculus is that we can *display* any given formula (structure, resp.) contained in a sequent by transforming the sequent into an equivalent one that has the given formula (structure, resp.) as the whole antecedent or succedent. The main advantage of display calculi lies to a large extent in a general cut elimination theorem that can be verified by checking whether the calculus obeys eight syntactic conditions.

2.2 Analytic Calculi and Cut Elimination

As already mentioned before, the use of cuts corresponds to the introduction of lemmas in a proof. But in a cut-free derivation (i.e. a derivation not containing applications of the cut rule), all formulas occurring in this derivation are subformulas of the end sequent. This property, known as *subformula property*, is key for the development of automated reasoning methods and proof-theoretic applications. Calculi with the subformula property are often referred to as *analytic*.

How can we prove that a calculus is analytic?

One way is to show the completeness of the calculus without the cut rule, i.e., that the cut rule (in a system without cut) is *admissible*. This means that the presence of the cut rule in the calculus does not add any new derivable sequents and the system is hence *closed under cut*.

Another — constructive — method is to prove the *cut elimination theorem* by transforming proofs containing applications of the cut rule into proofs without such applications. Gentzen's original proof of cut elimination (the *Hauptsatz*) for *LJ* [88] proceeds by a *double induction* on the complexity of the cut formula and the sum of its left and right ranks, starting from an *uppermost* cut. The left and right rank of a cut is defined as follows:

Definition 15. The left and right *rank* of a cut is the number of consecutive sequents that contain the cut formula, counting upward from the left and right premise of the cut, respectively.

Gentzen's strategy for cut elimination was (1) to *permute the cut upwards* to reduce the rank, or (2) to *replace the cut formula* with another formula that has lower complexity.

(1) Permute the cut upwards.
Consider for example the following application of a cut:

$$\dfrac{\Gamma \Rightarrow \chi \qquad \dfrac{\Delta, \chi \Rightarrow \varphi}{\Delta, \chi \Rightarrow \varphi \vee \psi}\,(\vee_1, r)}{\Gamma, \Delta \Rightarrow \varphi \vee \psi}\,(cut)$$

The cut can then be permuted upwards as follows:

$$\dfrac{\dfrac{\Gamma \Rightarrow \chi \qquad \Delta, \chi \Rightarrow \varphi}{\Gamma, \Delta \Rightarrow \varphi}\,(cut)}{\Gamma, \Delta \Rightarrow \varphi \vee \psi}\,(\vee_1, r)$$

(2) Replace the cut formula with a cut formula of lower complexity.
Consider the following application of a cut:

$$\dfrac{\dfrac{\Gamma \Rightarrow \varphi}{\Gamma \Rightarrow \varphi \vee \psi}\,(\vee_1, r) \qquad \dfrac{\varphi, \Delta \Rightarrow \Pi \qquad \psi, \Delta \Rightarrow \Pi}{\varphi \vee \psi, \Delta \Rightarrow \Pi}\,(\vee, l)}{\Gamma, \Delta \Rightarrow \Pi}\,(cut)$$

It can be replaced by the following cut:

$$\dfrac{\Gamma \Rightarrow \varphi \qquad \varphi, \Delta \Rightarrow \Pi}{\Gamma, \Delta \Rightarrow \Pi}\,(cut)$$

By repeating these types of replacements, one of the two parameters of the double induction will decrease: either the rank becomes smaller while the complexity remains the same (1) or the complexity of the cut formula decreases (2). Eventually, the derivation will end in an application of the cut rule where the cut formula is introduced by an axiom or an application of (w, r) or (w, l). Either way, the sequent in the conclusion of the cut can be proved without using the cut rule.

A problem occurs, however, when the cut formula is contracted by an application of the rule (c, l). This can be seen in the following case:

$$\dfrac{\Gamma \Rightarrow \varphi \qquad \dfrac{\varphi, \varphi, \Delta \Rightarrow \Pi}{\varphi, \Delta \Rightarrow \Pi}\,(c, l)}{\Gamma, \Delta \Rightarrow \Pi}\,(cut)$$

Here, the cut is not necessarily permuted upwards:

$$\dfrac{\dfrac{\Gamma \Rightarrow \varphi \qquad \dfrac{\Gamma \Rightarrow \varphi \qquad \varphi, \varphi, \Delta \Rightarrow \Pi}{\varphi, \Gamma, \Delta \Rightarrow \Pi}\,(cut)}{\Gamma, \Gamma, \Delta \Rightarrow \Pi}\,(cut)}{\Gamma, \Delta \Rightarrow \Pi}\,(c, l)$$

It can be seen that the right rank of the second cut is at least that of the original cut. To overcome this problem, Gentzen introduced the *multicut rule*, which is a derivable generalization of the cut rule, and allows to cut several occurrences of the cut formula:

$$\frac{\Gamma \Rightarrow \varphi \quad \varphi^n, \Delta \Rightarrow \Pi}{\Gamma, \Delta \Rightarrow \Pi}\ (mcut)$$

By using the multicut rule instead of the cut rule, the following theorem can be proven:

Theorem 1 ([88])**.** *Cut elimination holds for LJ.*

Cut Elimination in Hypersequent Calculi

In hypersequent calculi, cut elimination proofs proceed essentially as in the sequent calculus case. In addition to the problematic case regarding the internal contraction rules, a similar problem arises when permuting the cut upwards over the external contraction rule (ec). Avron solved this problem in [9] for the hypersequent calculus GRM of the logic **RM**[4] by keeping track of the "history of a derivation". Another way to overcome this obstacle is to introduce a multicut hypersequent rule, which allows — similarly to Gentzen's solution for sequents — to cut between one and several components in parallel:

$$\frac{G \mid \Gamma \Rightarrow \varphi \quad H \mid \Delta_1, \varphi^{m_1} \Rightarrow \Pi_1 \mid \cdots \mid \Delta_n, \varphi^{m_n} \Rightarrow \Pi_n}{G \mid H \mid \Delta_1, \Gamma^{m_1} \Rightarrow \Pi_1 \mid \cdots \mid \Delta_n, \Gamma^{m_n} \Rightarrow \Pi_n}\ (mcut)$$

We will briefly sketch a *general* method of proving cut elimination [127, 58, 60] that works for many single-conclusion hypersequent calculi. The idea of this approach is to prove cut elimination for all hypersequent calculi whose rules satisfy some simple syntactic properties: they are "substitutive" and "reductive". Intuitively, substitutivity of a rule ensures that cuts over non-principal formulas can be permuted upwards in the derivation. When the cut formula is principal in both premises, reductivity of the rules allows to replace the cut with a smaller cut on subformulas of the cut formula.

Let G and H be single-conclusion hypersequents. The set $\mathrm{CUT}(G, H)$ consists of hypersequents that are obtained by a cut between one component in H and one or more components in G:

Definition 16 ([127, 60])**.** $\mathrm{CUT}(G, H)$ is the set of hypersequents obtained by saturating $\{G, H\}$ under the following two operations:

1. if $G = (G' \mid \Delta_1, \varphi^{m_1} \Rightarrow \Pi_1 \mid \cdots \mid \Delta_n, \varphi^{m_n} \Rightarrow \Pi_n)$ and $H = (H' \mid \Gamma \Rightarrow \varphi)$, then, for all $0 \leq l_i \leq m_i$ and $i = 1, \ldots, n$ it is the case that $\mathrm{CUT}(G, H) \ni$

$$G' \mid H' \mid \Delta_1, \Gamma^{l_1}, \varphi^{m_1 - l_1} \Rightarrow \Pi_1 \mid \cdots \mid \Delta_n, \Gamma^{l_n}, \varphi^{m_n - l_n} \Rightarrow \Pi_n$$

[4] **RM** is an extension of the relevance logic **R** [4] with the "mingle" axiom $\alpha \supset (\alpha \supset \alpha)$.

2. if $G = (G' \mid \Gamma_1 \Rightarrow \varphi \mid \cdots \mid \Gamma_n \Rightarrow \varphi)$ and $H = (H' \mid \Delta, \varphi \Rightarrow \Pi)$ then it is the case that $\mathrm{CUT}(G, H) \ni$

$$G' \mid H' \mid \Gamma_1, \Delta \Rightarrow \Pi \mid \cdots \mid \Gamma_n, \Delta \Rightarrow \Pi$$

Definition 17 ([127, 60])**.** A hypersequent rule r is *substitutive* if for any

- instance $\dfrac{G_1 \quad \cdots \quad G_n}{G}$ of r,
- single-conclusioned hypersequent H,
- $G' \in \mathrm{CUT}(G, H)$ (with the condition that if r is a logical rule, then G' contains its principal formula), there exist $G'_i \in \mathrm{CUT}(G_i, H)$ for $i = 1, \ldots, n$ such that $\dfrac{G'_1 \quad \cdots \quad G'_n}{G'}$ is an instance of r.

Definition 18 ([127, 60])**.** The logical rules for any n-ary connective $\heartsuit$ are *reductive* if for all instances of left and right rules of $\heartsuit$:

$$\frac{G \mid S_1 \quad \ldots \quad G \mid S_l}{G \mid \Delta, \heartsuit(\varphi_1, \ldots, \varphi_n) \Rightarrow \Pi} \qquad \frac{G' \mid S'_1 \quad \ldots \quad G \mid S'_k}{G' \mid \Gamma \Rightarrow \heartsuit(\varphi_1, \ldots, \varphi_n)}$$

with S_i, S'_j sequents for $i = 1, \ldots, l; j = 1, \ldots, k$, the hypersequent $G' \mid G \mid \Gamma, \Delta \Rightarrow \Pi$ is derivable from $(G \mid S_1), \ldots, (G \mid S_l), (G' \mid S'_1), \ldots, (G' \mid S'_k)$ using only (*cut*) with cut formulas from $\varphi_1, \ldots, \varphi_n$.

The proof of cut elimination is then similar to Gentzen's cut elimination proof: it takes an uppermost cut and shifts it upwards in a specific order. First, the cut is moved upwards in the derivation that has the cut formula on the right side of the sequent (and then it is moved upwards in the derivation that has the cut formula on the left side). This shift is ensured by substitutivity of the rules of the calculus. If the cut formula is principal in both premises, by reductivity the cuts can be replaced by cuts on its subformulas.

Using this procedure, the following can be proved:

Theorem 2 ([127])**.** *Cut elimination holds for any single-conclusion hypersequent calculus that consists of:*

(1) the initial axioms for identity $\varphi \Rightarrow \varphi$ (and possibly for the logical constants),
(2) the rules (cut), (ew), and (ec)
(3) a set of substitutive and reductive logical rules, and
(4) a set of substitutive structural rules.

Since the rules of *HLJ* are reductive and substitutive, we have the following:

Corollary 1. *Cut elimination holds for HLJ.*

CHAPTER 3

TINC: Tools for the Investigation of Non-classical Logics

The idea to use computer support for the creation and investigation of logics has already been around for more than two decades, see e.g. [139, 140]. Following this spirit of "logic engineering", several tools have been introduced that aim at making theoretical results in logic more accessible to researchers and practitioners who might not have deep knowledge about the underlying logical theory, e.g. [22, 163, 164, 136]. A particular example of a "logic engineering" tool addressing the issue of finding analytic calculi in an automated way is the system *MUltlog* [22] which introduces such calculi for the rather restricted class of finite-valued logics.

In this chapter, we describe the system **TINC**, which stands for *Tools for the Investigation of Non-Classical logics*. It is created along the line of *MUltlog* to automatically generate analytic calculi for a wider range of non-classical logics. **TINC** takes the specification of a logic as input, returns an analytic calculus for the logic and states certain properties of the calculus. Since there is not yet a method to create analytic calculi for all non-classical logics uniformly, the task is being done incrementally by covering more and more families of logics — leading to several tools included in **TINC**. **TINC** currently contains three tools that can handle large classes of substructural, intermediate and paraconsistent logics. The tools (and their source code) are available at:

`http://www.logic.at/tinc`

In the following sections, we start by giving an overview of the general structure and idea of the **TINC**-system. We continue by presenting an overview of the implementation of the **TINC**-tools. The last section contains selected related work in the area.

This chapter is based on the publication [61].

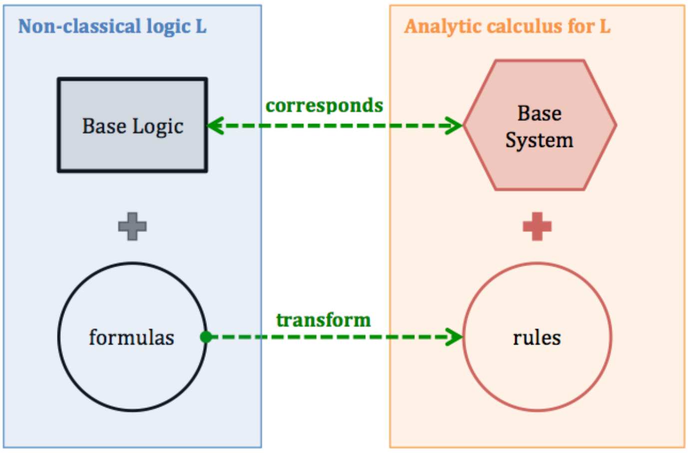


Figure 3.1: Definition of analytic calculi

3.1 TINC in a Nutshell

The theoretical basis of **TINC** is a procedure that has first been introduced in [52] for substructural logics. This procedure transforms Hilbert axioms of a specific shape into analytic sequent and hypersequent calculi in a uniform and systematic way (see Chapter 4 for a description of the method). The method has been further extended and modified to cover other logics, see e.g. [62, 28, 57, 55, 59, 56]. The original transformation procedure from [52] and some of these extended and modified procedures have been implemented in three tools, which are available via **TINC**, and are described in the Sections 4.3.1, 5.4.1 and 6.6.1.

The general idea of the procedure is depicted in Figure 3.1. We start from some *base logic*, e.g. intuitionistic logic **Int**. The logics that we can deal with are then defined by *adding suitable formulas*, in the form of Hilbert axioms or semantic specifications (frame conditions), to this base logic, see Example 6. These formulas are *transformed* into equivalent rules in the framework of the *calculus* for the base logic, e.g. *HLJ*. Adding these rules to the calculus for the base logic gives *analytic calculi* for our logics.

Example 6. *Gödel-Dummet logic* **GD** *is obtained from intuitionistic logic by*

1. *adding the Hilbert axiom* $(\varphi \supset \psi) \vee (\psi \supset \varphi)$ *to the calculus* Int_H*, or*
2. *imposing on intuitionistic frames the condition of strong connectedness of the accessibility relation* $\leqslant$ *(see Section 5.1 for the definitions):* $\forall x \forall y \forall z((x \leqslant y \wedge x \leqslant z) \rightarrow (y \leqslant z \vee z \leqslant y))$

Figure 3.2 shows how to define an analytic calculus for **GD** *starting with the syntactic definition 1., see above.*

The crucial part in this procedure is the transformation of formulas into suitable structural or logical rules to be added to the base calculus. Its key ingredients are:

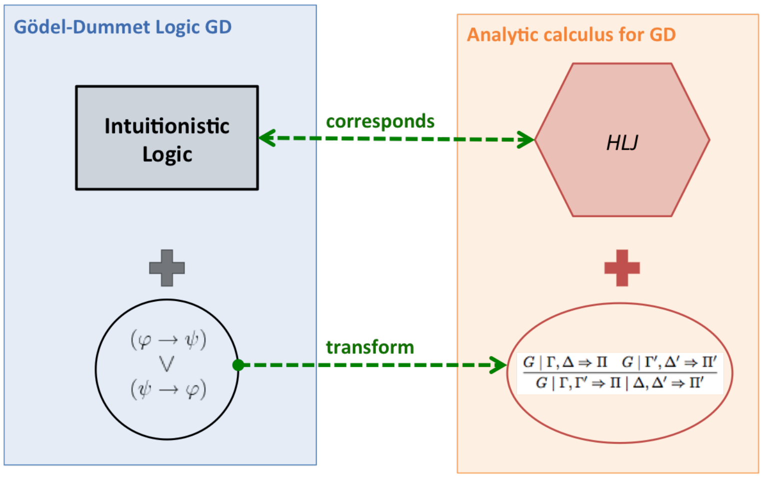


Figure 3.2: Example: definition of an analytic calculus for Gödel-Dummet logic **GD**

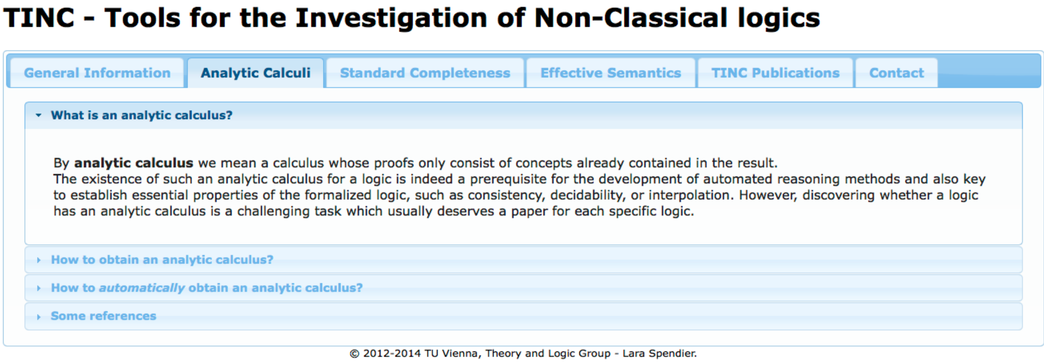


Figure 3.3: Main page of **TINC**

(1) the use of the *invertible rules* of the base calculus, and
(2) the *Ackermann lemma*, which allows formulas to change the side of the sequent by moving them from the rule conclusion to the rule premise. See the table below for the chapters containing various versions of the Ackermann Lemma for specific logics and calculi:

Base logic	Base calculus	Chapter
FLew	Sequent calculus	Chapter 4, Lemma 1
FLew	Hypersequent calculus	Chapter 4, Lemma 2
Int	Hypersequent calculus	Chapter 5, Lemma 4
Int	Labelled calculus	Chapter 5, Lemma 9
$\mathbf{Cl}^+$	Label-based sequent calculus	Chapter 6, Lemma 14

TINC (see its main page in Figure 3.3) currently contains three tools that implement this transformation procedure:

Tool	Overview	Chapter
AxiomCalc	• *AxiomCalc* takes as input Hilbert axioms defining a substructural logic. • The tool transforms these axioms into structural sequent or hypersequent rules, hence providing an *analytic sequent or hypersequent calculus* for the logic. • Moreover, it exploits the generated calculus by *checking a sufficient condition* for standard completeness of the input logic.	Chapter 4, Section 4.3.1
Framinator	• *Framinator* takes as input semantic specifications (frame conditions) that define intermediate logics. • The tool transforms these formulas into labelled rules, hence providing a *cut-free labelled sequent calculus* for the logic.	Chapter 5, Section 5.4.1
Paralyzer	• *Paralyzer* takes as input large classes of Hilbert axioms defining paraconsistent (and related) logics. • The tool transforms these axioms into *sequent calculus rules.* • Moreover, it extracts *non-deterministic, finite-valued semantics* from the obtained calculi. This new semantics is also used to show the decidability of the logics and reveal whether the calculi are analytic. • *Paralyzer* also provides an *encoding* of the introduced calculi for the proof-assistant *Isabelle* [138, 171] that can be used for semi-automated proof search within the considered logics.	Chapter 6, Section 6.6.1

All tools generate papers written in LaTeX that contain the resulting calculi, some basic explanations about them and their applications.

3.2 Implementation Details

All three tools are implemented in the declarative programming language Prolog (in particular, we used swi-prolog by Jan Wielemaker[1]). For the notions in this section (and also for more detailed information), we refer to the Prolog standard references [158, 68].

[1] `http://www.swi-prolog.org`

A (Very) Short Introduction to Prolog

The basic data structure in Prolog is a *term*, which is is an atom, a number, a variable or a compound term:

- An *atom* starts with a lowercase latter or is enclosed in single quotes, like `a`, `test`, or `'Something'`.
- A *number* is an integer or a float.
- A *variable* can stand for an arbitrary term and starts with an uppercase letter or an underscore `\_`. Examples are `A`, `\_test` or `Something`.
- A *compound term* consists of an atom called "functor" and a number of one or more arguments which are again terms, such as, for instance, `birthday(5,1,1989)` or `married('Patrizia','Ingmar')`. *Lists* and *strings* are specific types of compound terms. The former is an ordered collection of terms enclosed in square brackets `[ ]`. Its elements are separated by commas, e.g. `[ultimate,frisbee,fun]`. The latter is a sequence of characters within double quotes, e.g. `"Hello, World!"`.

Atoms and numbers are also called *atomic terms*, whereas atoms and compound terms are referred to as *predicates*.

A *logic program* consists of a finite set of *clauses*, which are *rules* or *facts*. A rule is a statement of the form

```
head :- body.
```

with the intuitive meaning "`head` is true if `body` is true". The head consists of one predicate, which can take any number of terms as arguments, and the body consists of (possibly) several predicates. The predicates are also referred to as *goals*. Several goals can be separated by a comma (`,`), which stands for the conjunction of the goals, or a semicolon (`;`), which stands for the disjunction of the goals. A rule with an empty body, i.e. a body not containing any goals, is a *fact*.

A *query* is a statement of the form

```
?- query.
```

It consists of one or more goals, also separated by commas or semicolons. Given a query, Prolog will try to *satisfy* these goals based on the information that is given in the program. If it succeeds, the query is a logical consequence of the program. Note however that Prolog works according to the *closed world assumption*: only known statements are true; statements that are not known are false.

In the following example, we show a small Prolog program to make these notions clearer.

Example 7. *The predicate* `child_of` *takes two arguments: the first is the child and the second is the parent. We start our program by providing some facts using this predicate:*

```
child_of(moses,marlene).
child_of(moses,nues).
child_of(dominik,patrizia).
child_of(dominik,ingmar).
child_of(lara,patrizia).
```

We can now pose the following query:

```
?- child_of(moses,nues).
```

Prolog will answer `yes` *since we have this fact in our program. For the following query, Prolog will return* `false` *because of the closed world assumption:*

```
?- child_of(ingmar,emma).
```

We will now extend our Prolog program by defining the notion of a `sibling`*:*

```
sibling(X,Z) :- child_of(X,Y), child_of(Z,Y), X \= Z.
```

With this rule, we say that siblings must share at least one parent (`child_of(X,Y)` *and* `child_of(Z,Y)`*)* and *that someone can not be a sibling to himself (*`X \= Z`*). We can now ask whether* `dominik` *has a sibling according to our program by using a variable:*

```
?- sibling(dominik,Y).
```

Prolog tries to instantiate a value to this variable according to the knowledge provided by the program. Indeed we will get the following answer:

```
Y = lara.
```

We can then also pose the following queries and will get the respective answers (note that in the first case the predicates are concatenated by `,` *while in the second case, we use* `;`*):*

```
?- sibling(moses,X),sibling(lara,Y).
false.

?- sibling(moses,X);sibling(lara,Y).
Y = dominik .
```

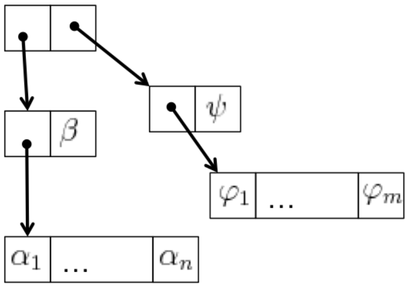


Figure 3.4: Representation of the hypersequent $\alpha_1, \dots, \alpha_n \Rightarrow \beta \mid \varphi_1, \dots, \varphi_m \Rightarrow \psi$ as a list

Data structure

The data structure we will use for the representation of (hyper)sequents and rules are lists. For instance, a single-conclusion sequent of the form $\varphi_1, \dots, \varphi_n \Rightarrow \psi$ (or $\varphi_1, \dots, \varphi_n \Rightarrow$) is simply represented as a list of two elements where the first element (that is representing the antecedent of the sequent) is another list:

```
[[_, _, _, _, _, ...], _]
```

A hypersequent is then considered a list of such lists. A single-conclusion hypersequent of the form $\alpha_1, \dots, \alpha_n \Rightarrow \beta \mid \varphi_1, \dots, \varphi_m \Rightarrow \psi$ is then represented as follows (see also Figure 3.4)

```
[[[alpha_1, ..., alpha_n], beta], [[phi_1, ..., phi_m], psi]]
```

Note however that in the formulation of hypersequent rules, the side hypersequent G is omitted in the list representation.

Example 8. *The list representation of the sequent $\varphi_1, \varphi_2, \varphi_3 \Rightarrow \psi$ is as follows (we use* `a` *for φ_1,* `b` *for φ_2,* `c` *for φ_3 and* `z` *for ψ):*

```
[[a, b, c], z]
```

The list representation of the hypersequent $G \mid \varphi_1, \varphi_2 \Rightarrow \mid \psi_1 \Rightarrow \psi_2$ is as follows (we use `a` *for φ_1,* `b` *for φ_2,* `c` *for ψ_1 and* `d` *for ψ_2):*

```
[ [[a, b], ''], [[c], d] ]
```

Rules are then represented as a list with two elements, where the first element is a list containing the premises and the second element is a list containing the conclusion.

Example 9. *Consider the following sequent rule application of $(\vee, l)$:*

$$\frac{\alpha, \varphi_1 \Rightarrow \beta \qquad \alpha, \varphi_2 \Rightarrow \beta}{\alpha, \varphi_1 \vee \varphi_2 \Rightarrow \beta}$$

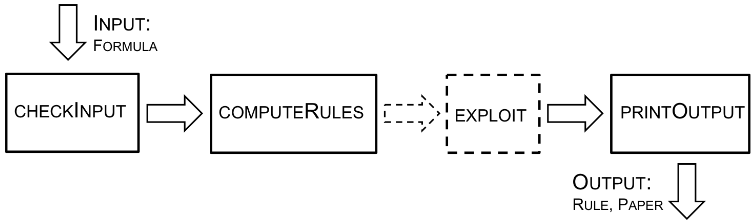


Figure 3.5: General structure of **TINC**

Its representation as a list is as follows (we use `a` *for* α*,* `b` *for* β*,* `c1` *for* φ_1 *and* `c2` *for* φ_2*):*

```
[ [[[a,c1],b], [[a,c2],b] ], [[a,c1 v c2],b]]
```

Recall the application of the hypersequent rule (*com*) *as used in Example 5:*

$$\frac{\varphi \Rightarrow \varphi \qquad \psi \Rightarrow \psi}{\varphi \Rightarrow \psi \mid \psi \Rightarrow \varphi}$$

We can represent it in our list notation in the following way (we use `a` *for* φ *and* `b` *for* ψ*):*

```
[ [ [[[a],a]], [[[b],b]] ], [[[a],b], [[b],a]] ]
```

Analogously, multiple-conclusion sequents will be represented as lists of two elements with the second element being a list as well.

Design of TINC

The implementation of all **TINC**-tools follows the structure depicted in Figure 3.5. Here we only give an outline of the general design of **TINC** — we will explain the specific instantiations for the implementation of each tool in the respective tool-section.

We provide two interfaces for **TINC**: the standard command-line interface for users who want to run the tools on their own computer and a web interface. The user provides the input formula for the tools via the respective interface. The input formula is a formula of a specific form depending on the class of logics the tool can handle. It is given as parameter to the first component, CHECKINPUT, which checks whether the input is correct according to the syntactic requirements of the input formula.

The second component, COMPUTERULES, is the core component of the whole tool. It contains the implementation of the algorithm to transform the input formula into an equivalent set of rules (recall the transformation step in Figure 3.1).

The third component, EXPLOIT, is optional. When present in the tool, it implements methods that utilize the computed calculus to establish properties of the formalized logic.

The fourth and last component, PRINTOUTPUT, contains everything that is related to presenting the results to the user: it prints the results of the transformation procedure and the investigation either on the command-line or on the web interface. Moreover, it also produces a paper written in LaTeX, which contains the obtained calculus and the results of the EXPLOIT-component.

Input and CHECKINPUT

The input formula is provided by the user following a tool-specific syntax. The component CHECKINPUT then contains methods to:

(i) check if the input is indeed a formula, and
(ii) check whether the input formula satisfies certain criteria to be transformed by our algorithm.

In step (i), we perform a basic syntax check to see if the input indeed has the shape of a formula of the defined language. To implement this, we use a definite clause grammar (DCG) to (a) check whether the input can be parsed as a formula, and (b) rewrite the input formulas to LaTeX-code, which will be used for the output later. An example of a DCG can be seen in Code Example 1 in Section 4.3.1 on page 52.

If the input formula does not pass the syntactic checks (i) or (ii), an error message is printed on the screen.

COMPUTERULES and EXPLOIT

The component COMPUTERULES contains the implementation of the algorithm to transform the input formula into an equivalent rule. For more information on the implementation of the transformation procedure, see Code Example 4 in Section 5.4.1 on page 95 and Code Example 5 in Section 6.6.1 on page 139.

The optional component EXPLOIT (recall Figure 3.5) is currently only implemented for the tools *AxiomCalc* and *Paralyzer*. We will give more details on its implementation in the respective tool section of Chapters 4 and 6.

Output and PRINTOUTPUT

The component PRINTOUTPUT contains the implementation of the presentation of the results. The results (the obtained calculus and – if applicable – the results of the EXPLOIT component) are (1) printed on the command-line or on the web interface, and (2) summarized in a LaTeX-paper, which can be downloaded from the web when generated by the web interface. See Code Example 3 in Section 4.3.1 on page 55 for the implementation of the LaTeX-paper generation.

3.2.1 Overview Code Examples

The following table gives an overview of the various code examples and their location in this thesis. Note that we only present code snippets to give an intuition of the implementation – for greater detail see the source code, which is freely available at http://www.logic.at/tinc

Component	Method	Tool	Chapter
CHECKINPUT	axiom2tex	*AxiomCalc*	Chapter 4, Code Example 1
COMPUTERULES	isClass	*Framinator*	Chapter 5, Code Example 4
COMPUTERULES	axioms2rules	*Paralyzer*	Chapter 6, Code Example 5
EXPLOIT	isConvergent	*AxiomCalc*	Chapter 4, Code Example 2
EXPLOIT	isAnalytic	*Paralyzer*	Chapter 6, Code Example 6
PRINTOUTPUT	texOut	*AxiomCalc*	Chapter 4, Code Example 3

3.3 Related Work

One of the first milestones to advance the development of tools for the design of application-oriented logics are the papers [139, 140], which focus on computer support for the generation and investigation of logics. These papers describe several approaches for automated methods in various fields of logic, including reasoning in Hilbert systems or the translation of Hilbert axioms into frame properties and vice versa. The methods are further assisted by the usage of tools like the automated theorem prover *Otter* [103] (the predecessor of *Prover9* [125]) and the implementation of the SCAN-algorithm[2] [84] for quantifier-elimination in second-order logic. Since then, several new or extended approaches and implementations to provide automated support for the development and investigation of non-classical logics have been provided. A website containing a (non-exhaustive) list of computational tools for (mainly) modal logics is, for example, http://www.cs.man.ac.uk/~schmidt/tools/.

We give an overview of selected tools that implement theoretical results in proof theory and can be considered "logic engineering tools" in the spirit of [139]. The common aim that these tools share with **TINC** is that they can be used for finding and/or investigating non-classical logics in an automated way. Note also that our listing is by no means complete: we only present tools that could possibly be used together with **TINC** (i.e., with *AxiomCalc*, *Framinator* or *Paralyzer*).

MUltlog and MUltseq. The predecessor of **TINC** is the Prolog program *MUltlog*[3] [22], which introduces analytic calculi automatically for the class of finitely-valued logics. *MUltlog* takes as input the specification of a finitely-valued first-order logic and produces as output a paper written in LaTeX containing a sequent calculus, a natural deduction system and clause formation rules for the specified logic.

The Prolog tool *MUltseq*[4] [89] is a generic sequent prover for propositional many-valued logics and was initially intended to accompany *MUltlog*. *MUltseq* takes as input

[2] http://www.mettel-prover.org/scan
[3] http://www.logic.at/multlog/
[4] http://www.logic.at/multseq/

(1) the rules of a many-valued sequent calculus (as e.g. produced by the system *MUlt-log*) and (2) sequents or formulas, and then decides the validity of the latter. Moreover, it is possible to decide the consequence relations associated with the logic and the sequent calculus given as input. Unfortunately, both tools (*MUltlog* and *MUltseq*) are not maintained any more.

MUltlog and the tools of **TINC** are rather unique since they start from the specification of a logic and create (and investigate) an analytic calculus based on this specification. Many of the other logic engineering tools that are currently available start already with the calculus for a logic and investigate it.

Investigative Tools: TATU and QUATI. The tool *TATU*[5] [136], which is implemented in OCaml, is a logic engineering tool that allows to reason about sequent calculi of a specific form. It takes as input an encoding of a proof system in the framework of linear logic with subexponentials (SELLF) [135]. *TATU* then automatically checks whether the specified proof system admits cut elimination, i.e. whether the system is analytic, and whether it is complete using only atomic axioms. Encodings for sequent calculi of some logics have already been provided, e.g. a multiple-conclusion calculus for intuitionistic logic, or proof systems for modal logic **S4** or intuitionistic Lax logic. However, the encodings have to be found manually and require some basic theoretical knowledge from the user.

Similar to *TATU*, the system *QUATI*[6] [137] takes as input the encoding of a sequent calculus (of a specific form) in SELLF. *QUATI* then allows us to prove permutation lemmas for the encoded proof system, which is usually a tedious exercise. The tool computes this automatically and gives as output the permutation transformation in LaTeX. *QUATI* is implemented in OCaml and makes use of the *DLV* system[7].

Once we have obtained an analytic calculus for a logic, we naturally would like to use it to do automated proof search. For this, we can either use interactive theorem provers (see the last paragraph below), or simpler tools that are easier to handle, but might also be more restricted in their functionality. In recent years, several generic tableau provers have been introduced that fall into this latter category.

Tableau Provers: Tableaux WorkBench, LoTREC and MetTeL.

The *Tableaux WorkBench*[8] [1], which is implemented in OCaml, is a generic tableau prover allowing the user to specify his own rules and strategies to experiment with proof search. It can handle propositional modal-type logics, such as the modal logics **K** and **S4**, or computational tree logic **CTL**.

Similarly, the system *LoTREC*[9] [87], implemented in Java, aims at providing tableau provers for logics with Kripke semantics – in particular modal and description logics.

[5] `https://www.logic.at/staff/giselle/tatu/`
[6] `https://www.logic.at/staff/giselle/quati/`
[7] `http://www.dlvsystem.com/dlv/`
[8] `http://twb.rsise.anu.edu.au`
[9] `http://www.irit.fr/ACTIVITES/LILaC/Lotrec/`

It works analogous to the *Tableaux WorkBench*, as it allows the user to determine the tableau rules and the search strategy. The main difference between the two tools is their approach: while the *Tableaux WorkBench* is based on a purely syntactic tableau algorithm, *LoTREC* is semantic-based and tries to build (counter-)models for a proof.

Another tableau prover for various propositional modal-type logics is the Java-tool *MetTeL*[10] [163]. It also allows the user to specify his own rules of a tableau calculus and use *MetTeL* to generate a prover for it. Its successor *MetTeL*^{2} [164] generates tableau provers from the user specification of a tableau calculus for a logic. *MetTeL*^{2} extends its predecessor not only by allowing a flexible specification language (that is, in contrast to *MetTeL*, not restricted to a fixed set of logical operators), but also by adding several more features to optimize proof search.

Proof Assistants: Isabelle, COQ and TWELF. To complete the picture, we briefly mention another important category of software tools, namely proof assistants. Prominent examples are *Isabelle*[11] [171], *COQ*[12] [37], or *TWELF*[13] [147]. Proof assistants are more general (and, thus, more complex) than automated theorem provers. While the latter allow to do proof search within a specific logic automatically, the former require interaction with the user. Hence they are also called *interactive theorem provers*. Proof assistants usually utilize higher-order logics for the development of proofs of mathematical statements.

These powerful systems can also be used to perform interactive proof search within or to reason about a logic. This is achieved by first encoding the calculus of a logic in the (higher-order logic of the) proof assistant. Here we have to distinguish between a *shallow* and a *deep* embedding of the logic:

In a shallow embedding, the calculus is encoded by a more or less direct translation into the logic of the respective proof assistant. It is easier to achieve and allows to prove theorems *within* the logic. For example in [35], Gödel's argument for the existence of God has recently been formalized (amongst others) in *Isabelle/HOL* by using a shallow embedding of higher-order modal logic [86].

In a deep embedding, syntax and semantics of the logic are modelled separately in the meta-language of the proof assistant. Even though a deep embedding is more complicated to establish, it enables the user to reason *about* properties of the (calculus of the) logic and prove proof-theoretic results such as soundness, completeness, or cut elimination, e.g. see [66, 67, 162].

[10] `http://www.mettel-prover.org/`
[11] `https://isabelle.in.tum.de/`
[12] `http://coq.inria.fr/`
[13] `http://twelf.org/`

CHAPTER 4

Substructural Logics

Substructural logics are logics lacking some (or all) of the structural rules when formalized as sequent systems. They encompass, among others, classical, intuitionistic, fuzzy, intermediate, linear or relevant logics. Substructural logics provide languages for modelling dynamic data structures or resources and are therefore of particular interest for various areas of computer science. Moreover, they are also widely studied in other fields, such as philosophy (in particular relevant logic) or linguistics (e.g., the Lambek calculus, which is used to represent linguistic expressions).

In this chapter, we present an application of the analytic calculi that are generated by the procedure in [52] for axiomatic extensions of the logic **MTL** [78]. We explain how to use the obtained calculi to investigate the corresponding logics, in particular, to check in an automated way whether they are standard complete, i.e., whether the logics are complete with respect to algebras based on truth values in $[0, 1]$. We also show the implementation of this result in the **TINC**-tool *AxiomCalc*.

We start the chapter by recalling the basic logic we deal with: (propositional) substructural logic **FLew** (Section 4.1). In Section 4.2, we give an overview of related work in proof theory that focuses on the (automated) introduction of analytic calculi for substructural logics. In particular, we present the systematic procedure from [52], which forms the theoretical base for the other algorithms that are introduced in this thesis. In Section 4.3, we utilize the calculi generated by this procedure for axiomatic extensions of **MTL** and identify properties that ensure standard completeness of the corresponding logics. The check of these properties is completely automatized in our tool *AxiomCalc*, which we describe in Section 4.3.1.

The results of this chapter are contained in [28].

4.1 Preliminaries

We use the language $\mathcal{L}_{FLew}$, i.e. the language of Full Lambek calculus with exchange and weakening. $\mathcal{L}_{FLew}$ consists of infinitely many (possibly indexed) propositional variables

$p, q, \ldots$, the binary connectives $\wedge$ (additive conjunction), $\cdot$ (multiplicative[1] conjunction or fusion), $\vee$ (disjunction), $\supset$ (implication) and the constants $\top$ and $\bot$. Formulas are built from propositional variables and constants by using the logical connectives. As usual, we abbreviate $\alpha \supset \bot$ to $\neg\alpha$. In the following, (possibly indexed) $\varphi, \psi, \chi, \alpha, \beta, \ldots$ will stand for (metavariables for) formulas and (possibly indexed) $\Gamma, \Delta, \Sigma, \Theta, \Lambda, \ldots$ denote multisets of formulas. To distinguish between rule applications and rule schemas, (only) in this chapter we denote metavariables for multisets of formulas with $\overline{\Gamma}, \overline{\Delta}, \overline{\Sigma}, \overline{\Theta}, \overline{\Lambda}, \ldots$.

We use the following notation for formulas and (metavariables for) multisets of formulas with $n \geq 0$:

$$\varphi^n = \overbrace{\{\varphi, \ldots, \varphi\}}^{n} \text{ and } \Sigma^n = \overbrace{\{\Sigma, \ldots, \Sigma\}}^{n} \text{ and } \overline{\Sigma}^n = \overbrace{\{\overline{\Sigma}, \ldots, \overline{\Sigma}\}}^{n}$$

The Substructural Logic FLew

Basic substructural systems are defined by removing the structural rules from the calclus *LJ* for (propositional) intuitionistic logic **Int**. Recall the standard sequent calculus for *LJ* (see Table 2.1 in Chapter 2) and its structural rules (w, l), (w, r) and (c, l). Moreover, recall the exchange rule, which is implicitly used in *LJ* since we use multisets of formulas:

$$\frac{\overline{\Gamma}, \psi, \varphi, \overline{\Sigma} \Rightarrow \overline{\Pi}}{\overline{\Gamma}, \varphi, \psi, \overline{\Sigma} \Rightarrow \overline{\Pi}}\ (e, l)$$

If we remove all of these structural rules from *LJ* ((e, l) is "removed" by using sequences instead of multisets of formulas), we retrieve the system for the most basic substructural logic, *Full Lambek calculus* **FL** [113], that is non-commutative intuitionistic linear logic without exponentials. If we remove the rules for contraction from *LJ*, we obtain the system for Full Lambek calculus with exchange and weakening, **FLew** (intuitionistic linear logic with weakening and without exponentials). We will use **FLew** as base logic in this chapter.

By removing some or all of these structural rules, the number of formulas (due to missing weakening and/or contraction rules) and the order of formulas (due to missing exchange rules) in a sequent matters. This also leads to a different interpretation of the comma "," in a sequent (note that in this chapter we only consider single-conclusion sequents): in **Int**, the comma of an arbitrary sequent is interpreted as the additive conjunction in the antecedent. However, when the contraction or weakening rules are missing, it is interpreted as the multiplicative conjunction "$\cdot$". E.g., the interpretation of a sequent in *FLew* is as follows:

Definition 19. In the substructural logic **FLew**, a sequent

$$S = \Gamma \Rightarrow \Pi$$

[1] We refer to [165] for the terminology of "additive" and "multiplicative".

$\varphi \Rightarrow \varphi$	$\dfrac{\overline{\Gamma} \Rightarrow}{\overline{\Gamma} \Rightarrow \varphi}\ (w,r)$	$\dfrac{\overline{\Gamma}, \varphi, \psi \Rightarrow \overline{\Pi}}{\overline{\Gamma}, \varphi \cdot \psi \Rightarrow \overline{\Pi}}\ (\cdot, l)$	$\dfrac{\overline{\Gamma} \Rightarrow \varphi \quad \overline{\Delta} \Rightarrow \psi}{\overline{\Gamma}, \overline{\Delta} \Rightarrow \varphi \cdot \psi}\ (\cdot, r)$
$\bot \Rightarrow$	$\dfrac{\overline{\Gamma} \Rightarrow \overline{\Pi}}{\overline{\Gamma}, \varphi \Rightarrow \overline{\Pi}}\ (w,l)$	$\dfrac{\overline{\Gamma}, \varphi \Rightarrow \psi}{\overline{\Gamma} \Rightarrow \varphi \supset \psi}\ (\supset, r)$	$\dfrac{\overline{\Gamma} \Rightarrow \varphi \quad \psi, \overline{\Delta} \Rightarrow \overline{\Pi}}{\overline{\Gamma}, \varphi \supset \psi, \overline{\Delta} \Rightarrow \overline{\Pi}}\ (\supset, l)$
$\Rightarrow \top$	$\dfrac{\overline{\Gamma} \Rightarrow \varphi_i}{\overline{\Gamma} \Rightarrow \varphi_1 \vee \varphi_2}\ (\vee_i, r)$	$\dfrac{\overline{\Gamma}, \varphi_i \Rightarrow \overline{\Pi}}{\overline{\Gamma}, \varphi_1 \wedge \varphi_2 \Rightarrow \overline{\Pi}}\ (\wedge_i, l)$	$\dfrac{\overline{\Gamma} \Rightarrow \varphi \quad \overline{\Gamma} \Rightarrow \psi}{\overline{\Gamma} \Rightarrow \varphi \wedge \psi}\ (\wedge, r)$
$\dfrac{\overline{\Gamma} \Rightarrow}{\overline{\Gamma} \Rightarrow \bot}\ (\bot, r)$	$\dfrac{\overline{\Gamma} \Rightarrow \overline{\Pi}}{\overline{\Gamma}, \top \Rightarrow \overline{\Pi}}\ (\top, l)$	$\dfrac{\overline{\Gamma} \Rightarrow \varphi \quad \varphi, \overline{\Delta} \Rightarrow \overline{\Pi}}{\overline{\Gamma}, \overline{\Delta} \Rightarrow \overline{\Pi}}\ (cut)$	$\dfrac{\overline{\Gamma}, \varphi \Rightarrow \overline{\Pi} \quad \overline{\Delta}, \psi \Rightarrow \overline{\Pi}}{\overline{\Gamma}, \overline{\Delta}, \varphi \vee \psi \Rightarrow \overline{\Pi}}\ (\vee, l)$

$i \in \{1, 2\}$; $\overline{\Pi}$ is a metavariable for a multiset containing at most one formula.

Table 4.1: Sequent calculus *FLew*

is interpreted as

$$\mathcal{I}(S) = \bigodot \Gamma \supset \Pi$$

where $\bigodot \Gamma$ stands for the multiplicative conjunction of the formulas in Γ ($\top$ when Γ is empty), and Π is one formula ($\bot$ when Π is empty).

The sequent calculus *FLew* for **FLew** is then obtained by removing from *LJ* the structural rule for contraction (c, l) and adding the rules for $\cdot$, see Table 4.1.

Systems for substructural logics can also be defined by removing some of the structural rules from the standard sequent calculus for classical propositional logic, *LK* [88]. E.g, **CFLe** (or **MALL**) is the multiplicative and additive fragment of linear logic introduced in [90]. Its sequent system is obtained by removing the structural rules for weakening and contraction from *LK*. However, in this chapter we only concentrate on extensions of **FLew**.

Examples of Substructural Logics

Generally speaking, a substructural logic is any axiomatic extension of **FL**. Below we list some examples of substructural logics obtained by extending **FLew**.

Logic		**Axiomatization**
Int	Intuitionistic logic	**FLew**+$\varphi \supset \varphi \cdot \varphi$
MTL	Monoidal t-norm logic [78] is the logic of left continuous t-norms and their residua.	**FLew**+$(\varphi \supset \psi) \vee (\psi \supset \varphi)$

GD	Gödel-Dummett logic [91, 71] is the logic of linear order and one of the main fuzzy logics.	**Int**+$(\varphi \supset \psi) \lor (\psi \supset \varphi)$ or **MTL**+$\varphi \supset \varphi \cdot \varphi$
SMTL	Strict MTL [77] is the logic of left-continuous t-norms satisfying the pseudo-complementation property.	**MTL**+$\neg\varphi \lor \neg\neg\varphi$
WNM	Weak nilpotent minimum logic [78] is the logic of left-continuous t-norms satisfying the weak nilpotent minimum property.	**MTL**+$\neg(\varphi \cdot \psi) \lor (\varphi \land \psi \supset \varphi \cdot \psi)$
BL	Basic logic [97] characterizes logics based on continuous t-norms.	**MTL**+$(\psi \cdot (\psi \supset \varphi)) \supset (\varphi \cdot (\varphi \supset \psi))$
Ł	Łukasiewicz logic [119, 120] is the logic of magnitude and also one of the main fuzzy logics.	**FLew**+$((\varphi \supset \psi) \supset \psi) \supset ((\psi \supset \varphi) \supset \varphi)$ or **BL**+$\neg\neg\varphi \supset \varphi$

4.2 Theoretical Base and Related Work in Proof Theory

Cut-free calculi for substructural logics are traditionally introduced on a case-by-case basis, tailored to the specific logic at hand. As mentioned in the introduction, this is usually achieved by the following three steps:

(i) Choose or define a suitable formalism,
(ii) find suitable rules for formalizing the specific logic under consideration, and
(iii) prove soundness, completeness and cut-elimination for the defined calculus.

As a consequence, there are many papers devoted to the introduction of an analytic calculus for a specific logic, see e.g. [11, 141, 21, 128, 126, 127] for a rather incomplete list.

To move the introduction of analytic calculi from a logic-specific to a more systematic manner, general methods are needed. Such methods have been investigated e.g. in the following works:

Goré [93, 92] utilizes the formalism of display calculus [34] to capture some substructural logics in a unified way. In [93], generalized cut-free display calculi are given for (intuitionistic) Bi-Lambek calculus[2] [114] and its extensions with few structural rules. In [92], this idea is generalized and display calculi are extracted by starting from the algebraic Gaggle-theoretic[3] semantics of a logic.

[2](Intuitionistic) Bi-Lambek calculus is (intuitionisic) Lambek calculus extended with duals of each logical connective of the Lambek calculus.

[3]Gaggle Theory generalizes the algebraic notions of residuation and Galois connections to obtain a uniform relational semantics for a substructural logic, see [74, 75].

Three other semantic-based approaches to provide analytic calculi are presented in [131], [121] and [64] (note however that the latter two allow analytic cuts in their obtained calculi). In [131], labelled sequent calculi are introduced for basic relevant logic and its extensions. The method turns frame properties of a specific shape, which characterize the considered logics, into equivalent labelled sequent rules. Note that we discuss this semantic approach in greater detail for intermediate logics (Chapter 5).

In [121], which extends the method in [142], relational proof systems for substructural logics are introduced by using relational semantics where the formulas are interpreted as ternary relations. The obtained relational proof systems, which are tableau-style theorem provers, are provided for extensions of **FL** and for linear logic with exponentials.

In [64], a combination of (a generalized version of) the classical refutation system *KE* [65] with labelled deductive systems [83] is used to provide a "uniform and transparent system of analytic deduction" for substructural logics. The basic idea of this "labelled analytic deduction" is to represent the deduction problems in the algebra of the labels and seek for a solution of algebraic equations among the labels.

Even though these approaches are interesting, we do not go into too much detail here. Instead, we concentrate on the systematic procedure of converting axioms into sequent and hypersequent calculi, which serves as the *starting point* for the methods established in this thesis.

Theoretical Base: A Systematic Procedure

In Chapter 3, we already gave the general idea of the procedure to obtain analytic calculi for large classes of logics. In this section, we present the method of [52] where this idea was originally introduced for substructural logics. It transforms Hilbert axioms defining substructural logics into equivalent structural sequent and hypersequent rules.

The foundation for the systematic procedure in [52] is the *substructural hierarchy*. It is based on the *polarity* [5] of logical connectives of the base sequent calculus *FLew*. Their negative or positive polarities depend on whether their right or left logical rule is invertible[4]. The logical connectives of *FLew* can be divided into two groups of negative $(\supset, \wedge, \bot)$ and positive $(\cdot, \vee, \top)$ connectives. Axioms with an outermost logical connective of positive (negative, resp.) polarity belong to a positive class $\mathcal{P}$ (negative class $\mathcal{N}$, resp.) of the hierarchy and each class contains an infinite number of axioms. The intuition behind this classification is that the different classes account for the difficulty to deal with the corresponding axioms proof theoretically (and, as shown in [54], with the preservation under suitable order theoretic completions of the corresponding algebraic equations). Note that this procedure was originally introduced using **FL** as base logic [54] and is described below in its simplified version based on **FLew**.

[4] A rule is *invertible* when the premises are derivable from the conclusion of the rule.

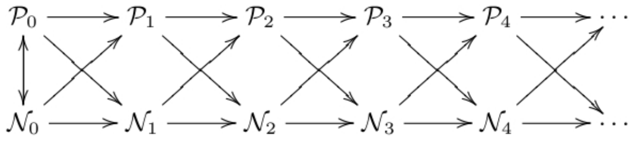


Figure 4.1: The substructural hierarchy [52]

Definition 20 ([52])**.** Let $\mathbf{A}_0$ be a set of atomic formulas. For $n \geq 0$, the sets $\mathcal{P}_n, \mathcal{N}_n$ of formulas are defined as follows:

$$\begin{aligned} \mathcal{P}_0 ::= \mathcal{N}_0 &::= \mathbf{A}_0 \\ \mathcal{P}_{n+1} &::= \mathcal{N}_n \mid \mathcal{P}_{n+1} \cdot \mathcal{P}_{n+1} \mid \mathcal{P}_{n+1} \vee \mathcal{P}_{n+1} \mid \top \\ \mathcal{N}_{n+1} &::= \mathcal{P}_n \mid \mathcal{P}_{n+1} \supset \mathcal{N}_{n+1} \mid \mathcal{N}_{n+1} \wedge \mathcal{N}_{n+1} \mid \bot \end{aligned}$$

A graphical representation of the substructural hierarchy is depicted in Figure 4.1. Note that the arrows $\rightarrow$ stand for inclusions $\subseteq$ of the classes.

Example 10. *Examples of Hilbert axioms and their classification within the substructural hierarchy:*

Name	*Axiom*	*Class*
weakening $(w), (w')$	$\varphi \supset \top,\ \bot \supset \varphi$	$\mathcal{N}_2$
contraction (c)	$\varphi \supset \varphi \cdot \varphi$	$\mathcal{N}_2$
weak contraction (wc)	$\neg(\varphi \wedge \neg\varphi)$	$\mathcal{N}_2$
excluded middle (em)	$\varphi \vee \neg\varphi$	$\mathcal{P}_2$
prelinearity $(prel)$	$(\varphi \supset \psi) \vee (\psi \supset \varphi)$	$\mathcal{P}_2$
Łukasiewicz axiom (Ł)	$((\varphi \supset \psi) \supset \psi) \supset ((\psi \supset \varphi) \supset \varphi)$	$\mathcal{N}_3$
Nelson axiom (nel)	$((\varphi \cdot \varphi \cdot \varphi \supset \psi) \wedge (\neg\psi \cdot \neg\psi \cdot \neg\psi \supset \neg\varphi)) \supset (\varphi \supset \psi)$	$\mathcal{N}_3$
weak excluded middle (wem)	$\neg\varphi \vee \neg\neg\varphi$	$\mathcal{P}_3$
weak nilpotent minimum (wnm)	$\neg(\varphi \cdot \psi) \vee (\varphi \wedge \psi \supset \varphi \cdot \psi)$	$\mathcal{P}_3$
Kripke model with 2 *worlds* $(bc2)$	$\varphi_0 \vee (\varphi_0 \supset \varphi_1) \vee (\varphi_0 \wedge \varphi_1 \supset \varphi_2)$	$\mathcal{P}_3$

Since the transformation procedure introduced in [52] works up to the classes $\mathcal{N}_2$ and $\mathcal{P}_3$ of the substructural hierarchy, the normal form of axioms within these classes is of particular interest:

$\mathcal{N}_2$: Axioms have the form $\bigwedge_{1\leq i\leq n}\delta_i$, in which every δ_i is a $\alpha_1\cdots\alpha_m \supset \beta$ where:
- $\beta = \bot$ or $\beta_1 \vee \cdots \vee \beta_k$ and each β_l is a multiplicative conjunction of propositional variables and
- each α_i is of the form $\bigwedge_{1\leq j\leq p}\gamma_i^j \supset \beta_i^j$ where
  - ∘ $\beta_i^j = \bot$ or a propositional variable, and
  - ∘ γ_i^j is a multiplicative conjunction or a disjunction of propositional variables (or $\top$).

$\mathcal{P}_2$: Axioms have the form $\bigvee_{1\leq i\leq n}\delta_i$, where each δ_i is of the form $\bigwedge_{1\leq j\leq m}\alpha_j \supset \beta_j$ or $\bigwedge_{1\leq j\leq m}\alpha_j$ where:
- each α_j is a multiplicative conjunction or disjunction of propositional variables and $\top$, and
- $\beta_j = \bot$ or a propositional variable.

$\mathcal{P}_3$: Axioms have the form $\delta_1 \vee \cdots \vee \delta_n$, where each δ_i is in $\mathcal{N}_2$.

From Axioms to Analytic Rules

The procedure in [52] transforms each axiom within the class $\mathcal{N}_2$ into equivalent structural sequent calculus rules that preserve cut elimination when added to the calculus *FLew*. These rules are however not powerful enough to capture axioms beyond the class $\mathcal{N}_2$. Indeed, as shown in [52, 54], they can only formalize properties that are already valid in intuitionistic logic ([52]) and among them only those corresponding to algebraic equations which are closed under the order theoretic completion known as Dedekind-MacNeille completion[5] in the context of integral residuated lattices ([54]). These results ensure, for instance, that no structural sequent rule can capture the prelinearity axiom, which is within $\mathcal{P}_2$ (see Example 10).

For axioms up to the class $\mathcal{P}_3$, the transformation procedure is adapted to a different base calculus, namely to the hypersequent calculus *HFLew* depicted in Table 4.2. These axioms can then be transformed into equivalent structural hypersequent rules. The calculi are obtained by adding the generated rules to the base sequent calculus *FLew* and the base hypersequent calculus *HFLew*, respectively.

The transformation procedure uses the following two key ingredients (recall the general procedure described in Chapter 3):

(1) the invertible logical rules of the base calculus *FLew* (or *HFLew*), that are $(\vee, l)$, $(\cdot, l)$, $(\top, l)$, $(\bot, r)$, $(\supset, r)$, and $(\wedge, r)$.
(2) the Ackermann lemma for *FLew*:

Lemma 1 ([52])**.** *Let* $\Phi_1, \ldots, \Phi_m$ *be (meta)sequents consisting of metavariables, and* $\psi_1, \ldots, \psi_n \Rightarrow \xi$ *be metavariables for formulas. Then the rules*

[5] Dedekind-MacNeille completion generalizes Dedekind's embedding of the rational numbers into the reals to various ordered algebraic structures [122].

$\varphi \Rightarrow \varphi$	$\dfrac{G \mid \overline{\Gamma} \Rightarrow}{G \mid \overline{\Gamma} \Rightarrow \overline{\Pi}}\ (w,r)$	$\dfrac{G \mid \varphi, \psi, \overline{\Gamma} \Rightarrow \overline{\Pi}}{G \mid \varphi \cdot \psi, \overline{\Gamma} \Rightarrow \overline{\Pi}}\ (\cdot, l)$	$\dfrac{G \mid \overline{\Gamma} \Rightarrow \varphi \quad G \mid \overline{\Delta} \Rightarrow \psi}{G \mid \overline{\Gamma}, \overline{\Delta} \Rightarrow \varphi \cdot \psi}\ (\cdot, r)$
$\Rightarrow \top$	$\dfrac{G \mid \overline{\Gamma} \Rightarrow \overline{\Pi}}{G \mid \overline{\Gamma}, \varphi \Rightarrow \overline{\Pi}}\ (w,l)$	$\dfrac{G \mid \varphi, \overline{\Gamma} \Rightarrow \psi}{G \mid \overline{\Gamma} \Rightarrow \varphi \supset \psi}\ (\supset, r)$	$\dfrac{G \mid \overline{\Gamma} \Rightarrow \varphi \quad G \mid \psi, \overline{\Delta} \Rightarrow \overline{\Pi}}{G \mid \overline{\Gamma}, \varphi \supset \psi, \overline{\Delta} \Rightarrow \overline{\Pi}}\ (\supset, l)$
$\bot \Rightarrow$	$\dfrac{G \mid \overline{\Gamma} \Rightarrow}{G \mid \overline{\Gamma} \Rightarrow \bot}\ (\bot, r)$	$\dfrac{G \mid \varphi_i, \overline{\Gamma} \Rightarrow \overline{\Pi}}{G \mid \varphi_1 \wedge \varphi_2, \overline{\Gamma} \Rightarrow \overline{\Pi}}\ (\wedge, l)$	$\dfrac{G \mid \overline{\Gamma} \Rightarrow \varphi \quad G \mid \overline{\Gamma} \Rightarrow \psi}{G \mid \overline{\Gamma} \Rightarrow \varphi \wedge \psi}\ (\wedge, r)$
	$\dfrac{G \mid \overline{\Gamma} \Rightarrow \overline{\Pi}}{G \mid \overline{\Gamma}, \top \Rightarrow \overline{\Pi}}\ (\top, l)$	$\dfrac{G \mid \overline{\Gamma} \Rightarrow \varphi_i}{G \mid \overline{\Gamma} \Rightarrow \varphi_1 \vee \varphi_2}\ (\vee, r)$	$\dfrac{G \mid \varphi, \overline{\Gamma} \Rightarrow \overline{\Pi} \quad G \mid \psi, \overline{\Gamma} \Rightarrow \overline{\Pi}}{G \mid \varphi \vee \psi, \overline{\Gamma} \Rightarrow \overline{\Pi}}\ (\vee, l)$
	$\dfrac{G}{G \mid \overline{\Gamma} \Rightarrow \overline{\Pi}}\ (ew)$	$\dfrac{G \mid \overline{\Gamma} \Rightarrow \overline{\Pi} \mid \overline{\Gamma} \Rightarrow \overline{\Pi}}{G \mid \overline{\Gamma} \Rightarrow \overline{\Pi}}\ (ec)$	$\dfrac{G \mid \overline{\Gamma} \Rightarrow \varphi \quad G \mid \varphi, \overline{\Delta} \Rightarrow \overline{\Pi}}{G \mid \overline{\Gamma}, \overline{\Delta} \Rightarrow \overline{\Pi}}\ (cut)$

$i \in \{1,2\}$, $\overline{\Pi}$ is a metavariable for a multiset containing at most one formula.

Table 4.2: Hypersequent calculus *HFLew* for **FLew**

$$\frac{\Phi_1 \quad \cdots \quad \Phi_m}{\psi_1, \ldots, \psi_n \Rightarrow \xi} \qquad \frac{\Phi_1 \quad \cdots \quad \Phi_m \qquad \xi \Rightarrow \beta}{\psi_1, \ldots, \psi_n \Rightarrow \beta}$$

$$\frac{\Phi_1 \quad \cdots \quad \Phi_m \qquad \alpha_1 \Rightarrow \psi_1 \qquad \cdots \qquad \alpha_n \Rightarrow \psi_n}{\alpha_1, \ldots, \alpha_n \Rightarrow \xi}$$

are equivalent by (*cut*) *and the identity axiom, where* $\alpha_1, \ldots, \alpha_n, \beta$ *are fresh metavariables for formulas.*

and the Ackermann lemma for *HFLew*:

Lemma 2 ([52])**.** *Let* $\Phi, \Phi_1, \ldots, \Phi_m$ *be (meta)hypersequents consisting of metavariables,* Υ_i *be a fresh metavariable* α_i *or* $\overline{\Gamma_i}$ *and* $\Upsilon \Rightarrow \Xi$ *is either* $\Rightarrow \beta$ *or* $\overline{\Sigma} \Rightarrow \overline{\Pi}$ *with* $\beta, \overline{\Sigma}, \overline{\Pi}$ *fresh* ($\overline{\Pi}$ *is either one formula or empty). Then the following rules are equivalent:*

$$\frac{G \mid \Phi_1 \ \cdots \ G \mid \Phi_m}{G \mid \Phi \mid \psi_1, \ldots, \psi_n \Rightarrow \xi} \qquad \frac{G \mid \Phi_1 \ \cdots \ G \mid \Phi_m \quad G \mid \Upsilon_1 \Rightarrow \psi_1 \ \cdots \ G \mid \Upsilon_n \Rightarrow \psi_n}{G \mid \Phi \mid \Upsilon_1, \ldots, \Upsilon_n \Rightarrow \xi}$$

$$\frac{G \mid \Phi_1 \ \cdots \ G \mid \Phi_m \qquad G \mid \xi, \Upsilon \Rightarrow \Xi}{G \mid \Phi \mid \psi_1, \ldots, \psi_n, \Upsilon \Rightarrow \Xi}$$

These ingredients are then integrated in the transformation procedure, which is described in an informal way as follows. Given any axiom $\varphi \in \mathcal{N}_2$ or $\varphi \in \mathcal{P}_3$:

(i) If $\varphi \in \mathcal{N}_2$, we start with the sequent $\Rightarrow \varphi$. If $\varphi \in \mathcal{P}_3$, its normal form is a disjunction of formulas of the form $\varphi_1 \vee \cdots \vee \varphi_n$ where each $\varphi_1, \ldots, \varphi_n$ is within $\mathcal{N}_2$. In this case, we start with the hypersequent $G \mid\Rightarrow \varphi_1 \mid \cdots \mid\Rightarrow \varphi_n$. By utilizing the invertibility of the logical rules, we decompose φ as much as possible and obtain an equivalent set of (hyper)sequent rules R without premises. As an example, consider the axiom for weak contraction $\neg(\varphi \wedge \neg\varphi) \in \mathcal{N}_2$:

$$\Rightarrow (\varphi \wedge (\varphi \supset \bot)) \supset \bot \quad \longrightarrow^{(i)} \quad \frac{}{\varphi \wedge (\varphi \supset \bot) \Rightarrow \bot}$$

$$\longrightarrow^{(i)} \quad \frac{}{\varphi \wedge (\varphi \supset \bot) \Rightarrow}$$

(ii) Next, we apply the Ackermann lemma (Lemma 1 or Lemma 2) to each $r \in R$ to change side of the sequents of those formulas that cannot be decomposed by logical rules in their current position. Continuing our example, we move $\varphi \wedge (\varphi \supset \bot)$ to the succedent of a premise and get

$$\longrightarrow^{(ii)} \quad \frac{\alpha \Rightarrow \varphi \wedge (\varphi \supset \bot)}{\alpha \Rightarrow}$$

(iii) Then we utilize again the invertibility of the logical rules to decompose the compound formulas in the premises of each rule, resulting in a set of structural (hyper)sequent rules R_s. In the previous example, we get:

$$\longrightarrow^{(iii)} \quad \frac{\alpha \Rightarrow \varphi \qquad \alpha \Rightarrow \varphi \supset \bot}{\alpha \Rightarrow} \quad \longrightarrow^{(iii)} \quad \frac{\alpha \Rightarrow \varphi \qquad \alpha, \varphi \Rightarrow}{\alpha \Rightarrow}$$

(iv) The final step is a *completion procedure* to transform the structural rules resulting from steps (*i*)–(*iii*) into equivalent rules that preserve cut-elimination and the subformula property once they are added to the base calculus. This completion procedure again contains three more steps (here we describe the completion procedure for hypersequent rules; it works analogously for sequent rules, as shown in the running example):

(iv.a) In the preliminary step, each metavariable for multisets of formulas $\overline{\Gamma}$ or $\overline{\Pi}$ (where $\overline{\Pi}$ is a metavariable for a multiset containing at most one formula) is replaced by a fresh metavariable β_Γ or β_Π for formulas. This step can be skipped if the structural rule does not contain any $\overline{\Gamma}$, nor $\overline{\Pi}$.

(iv.b) In the restructuring step, we replace each component

$$(\alpha_1, \ldots, \alpha_n \Rightarrow \beta)$$

in its conclusion with

$$(\overline{\Gamma}_1, \ldots, \overline{\Gamma}_n, \overline{\Sigma}_\beta \Rightarrow \overline{\Pi}_\beta)$$

and add $n+1$ premises

$$(G \mid \overline{\Gamma}_1 \Rightarrow \alpha_1), \ldots, (G \mid \overline{\Gamma}_n \Rightarrow \alpha_n), (G \mid \beta, \overline{\Sigma}_\beta \Rightarrow \overline{\Pi}_\beta)$$

where $\overline{\Gamma}_1, \ldots, \overline{\Gamma}_n, \overline{\Sigma}_\beta, \overline{\Pi}_\beta$ are fresh and mutually distinct metavariables and $\overline{\Pi}_\beta$ is a metavariable for a multiset containing at most one formula. Similarly, we replace each component

$$(\alpha_1, \ldots, \alpha_n \Rightarrow)$$

with

$$(\overline{\Gamma}_1, \ldots, \overline{\Gamma}_n \Rightarrow)$$

and add n premises

$$(G \mid \overline{\Gamma}_1 \Rightarrow \alpha_1), \ldots, (G \mid \overline{\Gamma}_n \Rightarrow \alpha_n)$$

Back to our example, we get:

$$\longrightarrow^{(iv.b)} \qquad \frac{\alpha \Rightarrow \varphi \qquad \alpha, \varphi \Rightarrow \qquad \overline{\Gamma} \Rightarrow \alpha}{\overline{\Gamma} \Rightarrow}$$

(iv.c) In the cutting step, we remove all the constants and variables that appear in the premises and not in the conclusion. When such variables appear on the left and on the right hand side of different premises, we close the obtained rules under all possible applications of (*cut*): Let each Υ denote either a metavariable for a formula or for a multiset of formulas and each Ξ is either empty or a metavariable for a formula. Let $\mathcal{S}$ be the set of premises of the structural rule for a formula α not occurring in the conclusion, let $\mathcal{S}_S = \{G \mid \Upsilon'_i \Rightarrow \alpha : 1 \leq i \leq k\}$ ($\mathcal{S}_A = \{G \mid \Upsilon_j, \alpha, \ldots, \alpha \Rightarrow \Xi_j : 1 \leq j \leq m\}$, resp.) be the subset of premises which have one occurrence of α in the succedent (one or more occurrences of α in the antecedent and Υ_j does not contain α, resp.). If $\mathcal{S}_S = \emptyset$ ($\mathcal{S}_A = \emptyset$, resp.), we remove all premises in $\mathcal{S}_A$ ($\mathcal{S}_S$, resp.) from $\mathcal{S}$. Else, let $\mathcal{S}_{cut}$ be the set of all hypersequents of the form $G \mid \Upsilon_j, \Upsilon'_{i_1}, \ldots, \Upsilon'_{i_p} \Rightarrow \Xi_j$ where $1 \leq j \leq m$ and $1 \leq i_1, \ldots, i_p \leq k$. By replacing $\mathcal{S}_S \cup \mathcal{S}_A$ with $\mathcal{S}_{cut}$, we obtain a new structural rule. This step is repeated until all variables not occurring in the conclusion are removed.
For the rule above we therefore get:

$$\longrightarrow^{(iv.c)} \qquad \frac{\overline{\Gamma} \Rightarrow \varphi \qquad \overline{\Gamma}, \varphi \Rightarrow}{\overline{\Gamma} \Rightarrow} \qquad \longrightarrow^{(iv.c)} \qquad \frac{\overline{\Gamma}, \overline{\Gamma} \Rightarrow}{\overline{\Gamma} \Rightarrow} \ (wc)$$

By adding the rule (*wc*) to *FLew*, we obtain a sound and complete calculus for the logic **FLew** extended with the axiom for weak contraction where the cut rule is admissible. For example, the axiom for weak contraction is indeed derivable in the new calculus:

$$\dfrac{\dfrac{\dfrac{\varphi \Rightarrow \varphi \qquad \dfrac{\bot \Rightarrow}{\bot \Rightarrow \bot}\,(w,r)}{\varphi, \varphi \supset \bot \Rightarrow \bot}\,(\supset,l)}{\dfrac{\varphi \wedge (\varphi \supset \bot), \varphi \wedge (\varphi \supset \bot) \Rightarrow \bot}{(\varphi \wedge (\varphi \supset \bot)) \Rightarrow \bot}\,(wc)}\,(\wedge,l)}{\Rightarrow (\varphi \wedge (\varphi \supset \bot)) \supset \bot}\,(\supset,r)$$

Theorem 3 ([52])**.** *Given any axiom $\varphi \in \mathcal{N}_2$ ($\varphi \in \mathcal{P}_3$, resp.), the rules generated by the algorithm in [52] are sound and complete for the substructural logic axiomatized by* **FLew** $+ \varphi$ *and they preserve cut elimination when added to the sequent calculus FLew (hypersequent calculus HFLew, resp.).*

Example 11. *The axiom for weak nilpotent minimum $\neg(\varphi \cdot \psi) \vee (\varphi \wedge \psi \supset \varphi \cdot \psi)$ (cf. Example 10) is within the class $\mathcal{P}_3$. The algorithm constructs an equivalent structural hypersequent rule as follows:*

$$G \mid\Rightarrow (\varphi \cdot \psi) \supset \bot \mid\Rightarrow \varphi \wedge \psi \supset \varphi \cdot \psi \quad \rightarrow^{(i)} \quad \dfrac{}{G \mid \varphi \cdot \psi \Rightarrow \bot \mid \varphi \wedge \psi \Rightarrow \varphi \cdot \psi}$$

$$\rightarrow^{(i)} \quad \dfrac{}{G \mid \varphi, \psi \Rightarrow \mid \varphi \wedge \psi \Rightarrow \varphi \cdot \psi} \quad \rightarrow^{(ii)} \quad \dfrac{G \mid \alpha \Rightarrow \varphi \wedge \psi \qquad G \mid \varphi \cdot \psi \Rightarrow \beta}{G \mid \varphi, \psi \Rightarrow \mid \alpha \Rightarrow \beta}$$

$$\rightarrow^{(iii)} \quad \dfrac{G \mid \alpha \Rightarrow \varphi \qquad G \mid \alpha \Rightarrow \psi \qquad G \mid \varphi, \psi \Rightarrow \beta}{G \mid \varphi, \psi \Rightarrow \mid \alpha \Rightarrow \beta}\,(wnm_s)$$

$$\rightarrow^{(iv.b)} \quad \dfrac{\begin{array}{ccc} & & G \mid \varphi, \psi \Rightarrow \beta \\ G \mid \alpha \Rightarrow \varphi & G \mid \alpha \Rightarrow \psi & G \mid \overline{\Lambda} \Rightarrow \alpha \\ G \mid \overline{\Gamma} \Rightarrow \varphi & G \mid \overline{\Delta} \Rightarrow \psi & G \mid \overline{\Sigma}, \beta \Rightarrow \overline{\Pi} \end{array}}{G \mid \overline{\Gamma}, \overline{\Delta} \Rightarrow \mid \overline{\Lambda}, \overline{\Sigma} \Rightarrow \overline{\Pi}}$$

$$\rightarrow^{(iv.c)} \quad \dfrac{\begin{array}{ccc} G \mid \overline{\Lambda} \Rightarrow \varphi & G \mid \overline{\Lambda} \Rightarrow \psi & \\ G \mid \overline{\Gamma} \Rightarrow \varphi & G \mid \overline{\Delta} \Rightarrow \psi & G \mid \varphi, \psi, \overline{\Sigma} \Rightarrow \overline{\Pi} \end{array}}{G \mid \overline{\Gamma}, \overline{\Delta} \Rightarrow \mid \overline{\Lambda}, \overline{\Sigma} \Rightarrow \overline{\Pi}}$$

$$\rightarrow^{(iv.c)} \quad \dfrac{\begin{array}{cc} G \mid \overline{\Lambda}, \overline{\Lambda}, \overline{\Sigma} \Rightarrow \overline{\Pi} & G \mid \overline{\Lambda}, \overline{\Delta}, \overline{\Sigma} \Rightarrow \overline{\Pi} \\ G \mid \overline{\Gamma}, \overline{\Lambda}, \overline{\Sigma} \Rightarrow \overline{\Pi} & G \mid \overline{\Gamma}, \overline{\Delta}, \overline{\Sigma} \Rightarrow \overline{\Pi} \end{array}}{G \mid \overline{\Gamma}, \overline{\Delta} \Rightarrow \mid \overline{\Lambda}, \overline{\Sigma} \Rightarrow \overline{\Pi}}\,(wnm)$$

By adding the rule (wnm) *and* (com) *(see Table 4.3) to HFLew, we obtain a sound and complete calculus for* **WNM** *where the cut rule is admissible.*

Note that the other axioms depicted in Example 10 are transformed similarly into the equivalent analytic rules of Table 4.3.

Note that the transformation procedure works for all axioms within the class $\mathcal{N}_2$ of the substructural hierarchy, even when choosing *FL* (instead of *FLew*) as base calculus,

$$\frac{\overline{\Gamma}, \overline{\Gamma}, \overline{\Delta} \Rightarrow \overline{\Pi}}{\overline{\Gamma}, \overline{\Delta} \Rightarrow \overline{\Pi}}\ (c) \qquad \frac{G \mid \overline{\Gamma}, \overline{\Delta} \Rightarrow \overline{\Pi}}{G \mid \overline{\Gamma} \Rightarrow \mid \overline{\Delta} \Rightarrow \overline{\Pi}}\ (em) \qquad \frac{G \mid \overline{\Gamma}_1, \overline{\Gamma}_2 \Rightarrow}{G \mid \overline{\Gamma}_1 \Rightarrow \mid \overline{\Gamma}_2 \Rightarrow}\ (wem)$$

$$\frac{\overline{\Gamma} \Rightarrow \overline{\Pi}}{\overline{\Gamma}, \overline{\Delta} \Rightarrow \overline{\Pi}}\ (w) \qquad \frac{G \mid \overline{\Gamma}_1, \overline{\Delta}_2 \Rightarrow \overline{\Pi}_2 \qquad G \mid \overline{\Gamma}_2, \overline{\Delta}_1 \Rightarrow \overline{\Pi}_1}{G \mid \overline{\Gamma}_1, \overline{\Delta}_1 \Rightarrow \overline{\Pi}_1 \mid \overline{\Gamma}_2, \overline{\Delta}_2 \Rightarrow \overline{\Pi}_2}\ (com)^1$$

$$\frac{\overline{\Gamma} \Rightarrow}{\overline{\Gamma} \Rightarrow \overline{\Pi}}\ (w') \qquad \frac{G \mid \overline{\Gamma}_1, \overline{\Delta}_2 \Rightarrow \overline{\Pi}_2 \qquad G \mid \overline{\Gamma}_1, \overline{\Delta}_3 \Rightarrow \overline{\Pi}_3 \qquad G \mid \overline{\Gamma}_2, \overline{\Delta}_3 \Rightarrow \overline{\Pi}_3}{G \mid \overline{\Delta}_3 \Rightarrow \overline{\Pi}_3 \mid \overline{\Gamma}_2, \overline{\Delta}_2 \Rightarrow \overline{\Pi}_2 \mid \overline{\Gamma}_1, \overline{\Delta}_1 \Rightarrow \overline{\Pi}_1}\ (bc2)$$

$\overline{\Pi}, \overline{\Pi_1}, \overline{\Pi_2}, \overline{\Pi_3}$ are either one formula or empty

[1] The rule generated from the axiom (*prel*) is actually Avron's (*com*)-rule.

Table 4.3: Analytic rules corresponding to the axioms in Example 10

see [53]. The absence of the weakening rule in *FL* however forces us to consider a proper subclass of $\mathcal{P}_3$ axioms.

In [62], the transformation procedure is introduced for Hilbert axioms in the language of **CFLe** (better known as **MALL**), linear logic without exponentials. As in the case of **FLew**, the substructural hierarchy for **CFLe** is based on the polarity of its logical connectives. The algorithm to transform the axioms into equivalent analytic (hyper)sequent rules works analogously to the one described above. The advantage of the shift from the intuitionistic, single-conclusion setting to the classical, multiple-conclusion setting is that some axioms belonging to higher classes in the former setting are brought down to lower classes in the latter setting.

In addition to this result, a heuristic principle to generate logical (instead of structural) rules is introduced in [62]. Note that the generation of logical rules requires some additional effort for the proofs of soundness, completeness and cut elimination and in finding the "right" logical rule(s). The heuristic principle works roughly in the following three steps:

(1) The desired axiom is transformed into a structural or logical rule using the transformation procedure presented before.
(2) If the obtained system is not cut-free, a counterexample φ for cut elimination, which explains the failing proof, needs to be found.
(3) If $\varphi \in \mathcal{N}_2$ or $\varphi \in \mathcal{P}_3$, by the transformation procedure, φ is turned into a structural or logical rule and the procedure is started over from step (2).

This way, [62] rediscovered the calculus for Łukasiewicz logic in [129]. Note that the peculiar axiom of this logic is within $\mathcal{N}_3$ (see Example 10).

4.3 An Application: Standard Completeness for Extensions of MTL

We now show how to utilize the analytic calculi introduced by the procedure in [52] for axiomatic extensions of the Monoidal t-norm based logic **MTL** [78] to *standard completeness*, i.e., completeness of the formalized logics with respect to algebras based on truth values in the unit real interval $[0, 1]$; this property makes these logics fuzzy logics in the sense of [97].

The Logic MTL

Introduced in [78], monoidal t-norm based logic **MTL** is the logic of left continuous t-norms[6] and their residua. **MTL** is defined in terms of the Hilbert-style calculus MTL_H as follows [78]:

(Schematic) Axioms

(A1) $(\varphi \supset \psi) \supset ((\psi \supset \chi) \supset (\varphi \supset \chi))$
(A2) $(\varphi \cdot \psi) \supset \varphi$
(A3) $(\varphi \cdot \psi) \supset (\psi \cdot \varphi)$
(A4) $(\varphi \wedge \psi) \supset \varphi$
(A5) $(\varphi \wedge \psi) \supset (\psi \wedge \varphi)$
(A6) $(\varphi \cdot (\varphi \supset \psi)) \supset (\varphi \wedge \psi)$
(A7a) $(\varphi \supset (\psi \supset \chi)) \supset ((\varphi \cdot \psi) \supset \chi))$
(A7b) $((\varphi \cdot \psi) \supset \chi) \supset (\varphi \supset (\psi \supset \chi))$
(A8) $((\varphi \supset \psi) \supset \chi) \supset (((\psi \supset \varphi) \supset \chi) \supset \chi)$
(A9) $\bot \supset \varphi$

Inference rules

- *modus ponens* (MP): for given formulas φ and $\varphi \supset \psi$, we obtain ψ; $\dfrac{\varphi \quad \varphi \supset \psi}{\psi}$

As mentioned in Section 4.1, an alternative axiomatization of **MTL** is obtained by extending **FLew** with the prelinearity axiom $(\varphi \supset \psi) \vee (\psi \supset \varphi)$. A proof system for **MTL** is therefore the (cut-free) hypersequent calculus *HMTL*, which is obtained by adding to the hypersequent calculus *HFLew* Avron's communication rule (*com*), see Table 4.4.

We also need the following definitions of algebras and standard completeness (we refer to [127, 53, 85] for more details).

Definition 21. An **MTL**-*algebra* is a structure $\mathcal{A} = (A, \wedge, \vee, \cdot, \supset, 0, 1)$, where

- $(A, \wedge, \vee, 0, 1)$ is a lattice with 0 and 1 as the least and greatest element, respectively.

[6] A *t*-norm is a commutative, associative, increasing function $* : [0, 1]^2 \to [0, 1]$ with identity element 1. $*$ is *left continuous* iff whenever $\{x_n\}, \{y_n\}$ $(n \in N)$ are increasing sequences in $[0, 1]$ s.t. their suprema are x and y, then $\sup\{x_n * y_n : n \in N\} = x * y$. The residuum of $*$ is a function $\to^*$ where $x \to^* y = \max\{z \mid x * z \leq y\}$.

$\varphi \Rightarrow \varphi$	$\dfrac{G \mid \overline{\Gamma} \Rightarrow}{G \mid \overline{\Gamma} \Rightarrow \overline{\Pi}}\ (w, r)$	$\dfrac{G \mid \varphi, \psi, \overline{\Gamma} \Rightarrow \overline{\Pi}}{G \mid \varphi \cdot \psi, \overline{\Gamma} \Rightarrow \overline{\Pi}}\ (\cdot, l)$	$\dfrac{G \mid \overline{\Gamma} \Rightarrow \varphi \quad G \mid \overline{\Delta} \Rightarrow \psi}{G \mid \overline{\Gamma}, \overline{\Delta} \Rightarrow \varphi \cdot \psi}\ (\cdot, r)$
$\Rightarrow \top$	$\dfrac{G \mid \overline{\Gamma} \Rightarrow \overline{\Pi}}{G \mid \overline{\Gamma}, \varphi \Rightarrow \overline{\Pi}}\ (w, l)$	$\dfrac{G \mid \varphi, \overline{\Gamma} \Rightarrow \psi}{G \mid \overline{\Gamma} \Rightarrow \varphi \supset \psi}\ (\supset, r)$	$\dfrac{G \mid \overline{\Gamma} \Rightarrow \varphi \quad G \mid \psi, \overline{\Delta} \Rightarrow \overline{\Pi}}{G \mid \overline{\Gamma}, \varphi \supset \psi, \overline{\Delta} \Rightarrow \overline{\Pi}}\ (\supset, l)$
$\bot \Rightarrow$	$\dfrac{G \mid \overline{\Gamma} \Rightarrow \overline{\Pi}}{G \mid \overline{\Gamma}, \top \Rightarrow \overline{\Pi}}\ (\top, l)$	$\dfrac{G \mid \varphi_i, \overline{\Gamma} \Rightarrow \overline{\Pi}}{G \mid \varphi_1 \wedge \varphi_2, \overline{\Gamma} \Rightarrow \overline{\Pi}}\ (\wedge, l)$	$\dfrac{G \mid \overline{\Gamma} \Rightarrow \varphi \quad G \mid \overline{\Gamma} \Rightarrow \psi}{G \mid \overline{\Gamma} \Rightarrow \varphi \wedge \psi}\ (\wedge, r)$
	$\dfrac{G \mid \overline{\Gamma} \Rightarrow}{G \mid \overline{\Gamma} \Rightarrow \bot}\ (\bot, r)$	$\dfrac{G \mid \overline{\Gamma} \Rightarrow \varphi_i}{G \mid \overline{\Gamma} \Rightarrow \varphi_1 \vee \varphi_2}\ (\vee, r)$	$\dfrac{G \mid \varphi, \overline{\Gamma} \Rightarrow \overline{\Pi} \quad G \mid \psi, \overline{\Gamma} \Rightarrow \overline{\Pi}}{G \mid \varphi \vee \psi, \overline{\Gamma} \Rightarrow \overline{\Pi}}\ (\vee, l)$
	$\dfrac{G}{G \mid \overline{\Gamma} \Rightarrow \overline{\Pi}}\ (ew)$	$\dfrac{G \mid \overline{\Gamma} \Rightarrow \overline{\Pi} \mid \overline{\Gamma} \Rightarrow \overline{\Pi}}{G \mid \overline{\Gamma} \Rightarrow \overline{\Pi}}\ (ec)$	$\dfrac{G \mid \overline{\Gamma} \Rightarrow \varphi \quad G \mid \varphi, \overline{\Delta} \Rightarrow \overline{\Pi}}{G \mid \overline{\Gamma}, \overline{\Delta} \Rightarrow \overline{\Pi}}\ (cut)$
	$\dfrac{G \mid \overline{\Gamma}_1, \overline{\Delta}_1 \Rightarrow \overline{\Pi}_1 \quad G \mid \overline{\Gamma}_2, \overline{\Delta}_2 \Rightarrow \overline{\Pi}_2}{G \mid \overline{\Gamma}_1, \overline{\Gamma}_2 \Rightarrow \overline{\Pi}_1 \mid \overline{\Delta}_1, \overline{\Delta}_2 \Rightarrow \overline{\Pi}_2}\ (com)$		

$i \in \{1, 2\}$, $\overline{\Pi}, \overline{\Pi}_1, \overline{\Pi}_2$ are metavariables for multisets containing at most one formula.

Table 4.4: Hypersequent calculus *HMTL* for **MTL**

- $(A, \cdot, 1)$ is a commutative monoid.
- For any $x, y, z \in A$ the *residuation* property holds, i.e., $x \cdot z \leq y$ iff $z \leq x \supset y$, with $\leq$ a binary relation on A defined as $x \leq y$ iff $x = x \wedge y$.
- The *prelinearity* equation holds, i.e., $1 = (x \supset y) \vee (y \supset x)$ for any $x, y \in A$

The **MTL**-algebra $\mathcal{A} = (A, \wedge, \vee, \cdot, \supset, 0, 1)$ is said to be

- a *chain* if for every $x, y \in A$, either $x \leq y$ or $y \leq x$.
- *dense* if, for every $x, y \in A$, whenever $x \not\leq y$, there is a $z \in A$ such that $x \not\leq z$ and $z \not\leq y$.

An $\mathcal{A}$-*valuation* is a homomorphism from formulas to $\mathcal{A}$, i.e., a mapping e from formulas to A such that $e(0) = 0$, $e(1) = 1$ and $e(\varphi \diamond \psi) = e(\varphi) \diamond e(\psi)$, where $\diamond \in \{\wedge, \vee, \cdot, \supset\}$.

Definition 22. Let $\mathbf{L}$ be an axiomatic extension of **MTL** with a set **Ax** of additional axioms and HL be a suitable proof system for $\mathbf{L}$. An ***L**-algebra* $\mathcal{A}_{\mathbf{L}} = (A, \wedge, \vee, \cdot, \supset, 0, 1)$ is an **MTL**-algebra which satisfies $1 = e(\varphi)$, for every $\varphi \in \mathbf{Ax}$ and every $\mathcal{A}$-valuation e. We say that $\mathbf{L}$ is *standard complete* if, for every set of formulas $\Gamma \cup \varphi$, the following are equivalent:

- $\Gamma \vdash_{HL} \varphi$
- For every $\mathbf{L}$-algebra over the real interval $[0, 1]$, $\mathcal{A}_{\mathbf{L}} = ([0, 1], \wedge, \vee, \cdot, \supset, 0, 1)$ and $\mathcal{A}_{\mathbf{L}}$-valuation e, if $e(\psi) = 1$ for every $\psi \in \Gamma$, then $e(\varphi) = 1$.

Traditionally, the proof that a logic is standard complete uses *semantic techniques* which are inherently logic-specific, e.g., [102, 50, 99]. Given a logic **L** described in a Hilbert-style system, such semantic proofs usually consist of the following four steps (see, e.g., [78, 97, 102, 50, 130, 77]):

1. The algebraic semantics of the logic is identified (**L**-algebras).
2. It is shown that if a formula is not valid in an **L**-algebra, then it is not valid in a countable **L**-chain (linearly ordered **L**-algebra).
3. It is shown that any countable **L**-chain can be embedded into a countable *dense* **L**-chain by adding countably many new elements to the algebra and extending the operations appropriately. This establishes *rational completeness*: a formula is derivable in **L** iff it is valid in all dense **L**-chains.
4. Finally, a countable dense **L**-chain is embedded into a standard **L**-algebra, that is an **L**-algebra with lattice reduct $[0, 1]$, using a Dedekind-MacNeille-style completion.

Note that the most difficult step is to establish rational completeness (step 3), as it relies on finding the right embedding, if there is one. A different approach to step 3 using *proof-theoretic techniques* was proposed in [127]: the idea is that the admissibility of the so-called *density rule* in a logic **L** can lead to a proof of rational completeness for **L**; for instance, this is the case when **L** is any axiomatic extension of **MTL**. The density rule, which is syntactically similar to the cut rule, was introduced by Takeuti and Titani in their axiomatization of first-order Gödel logic [161]. In hypersequent calculi (an instance of) this rule has the form:

$$\frac{G' \mid \Sigma, p \Rightarrow \Pi \mid \Lambda \Rightarrow p}{G' \mid \Sigma, \Lambda \Rightarrow \Pi} \ (D)$$

where p is a propositional variable not occurring in Σ, Λ, Π or G', i.e. p is an *eigenvariable*. Adding the density rule to a hypersequent calculus can have a dramatic effect. Consider e.g. *HMTL + (em)* from Table 4.3. By adding the density rule (D) we are able to prove the empty sequent as follows:

$$\cfrac{\cfrac{\cfrac{}{p \Rightarrow p}\ (init)}{p \Rightarrow \mid \Rightarrow p}\ (em)}{\Rightarrow}\ (D)$$

A similar situation arises for *HMTL + (bc2) + (c)*. This is not really a surprise because the addition and subsequent elimination of (D) from an extension of *HMTL* leads to rational completeness for the formalized logic, as shown in [127]. However, the two calculi mentioned above formalize logics that are *not* rational complete: they are classical logic, and 3-valued Gödel logic. On the other hand, for many extensions of *HMTL*, adding (D) has no effect on the derivable hypersequents: applications of (D) can be *eliminated* from derivations.

Following the approach in [127], to establish standard completeness for a logic **L** we need to

(a) define a suitable proof system *HL* for **L** extended with the density rule
(b) check that this rule can be eliminated (or is admissible) in *HL*, i.e. that density does not enlarge the set of provable formulas. Density elimination implies rational completeness for the formalized logic [127].
(c) prove standard completeness in many cases (but not in general) by means of the Dedekind-MacNeille completion, which is ensured by the results in [53].

We use these three steps to automate standard completeness proofs for large classes of axiomatic extensions of **MTL**. This is achieved by identifying sufficient conditions on hypersequent rules that ensure density elimination (and, by [127], rational completeness) for the formalized logics. We call the rules meeting these conditions "convergent rules". Moreover, the tool *AxiomCalc* (Section 4.3.1) automates steps (a)-(c) above for propositional logics extending **MTL** by any Hilbert axiom within the class $\mathcal{P}_3$ in the substructural hierarchy.

Step (a): Automated Proof Theory for Extensions of MTL

By applying the algorithm of [52] that we described previously, we can find a proof system *HL* for any logic defined by extending **MTL** with Hilbert axioms within the class $\mathcal{P}_3$ of the substructural hierarchy.

Example 12. *Let* $\mathbf{WNM}^n$ *be the logic defined by adding to* **MTL** *the axiom* $\neg(\varphi\cdot\psi)^n \vee ((\varphi\wedge\psi)^{n-1} \supset (\varphi\cdot\psi)^n)$ *for any* $n \geq 2$*, where* $(\varphi\diamond\psi)^n$ *stands for* $(\varphi\diamond\psi)\diamond\cdots\diamond(\varphi\diamond\psi), \diamond \in \{\wedge,\vee,\supset,\cdot\}$ *n times. The equivalent analytic rule obtained by the algorithm in Section 4.2 is*

$$\frac{\begin{array}{c}\{G \mid (\overline{\Gamma}_i, \overline{\Gamma}_j)^n, \overline{\Sigma} \Rightarrow \overline{\Pi}\}_{1\leq i,j\leq(n-1)}\\ \{G \mid (\overline{\Gamma}_i, \overline{\Gamma}_{n+2j-1})^n, \overline{\Sigma} \Rightarrow \overline{\Pi}\}_{1\leq i\leq(n-1);1\leq j\leq n}\\ \{G \mid (\overline{\Gamma}_{n+2i-2}, \overline{\Gamma}_j)^n, \overline{\Sigma} \Rightarrow \overline{\Pi}\}_{1\leq i\leq n;1\leq j\leq(n-1)}\\ \{G \mid (\overline{\Gamma}_{n+2i-2}, \overline{\Gamma}_{n+2j-1})^n, \overline{\Sigma} \Rightarrow \overline{\Pi}\}_{1\leq i,j\leq n}\end{array}}{G \mid \overline{\Gamma}_n, \cdots, \overline{\Gamma}_{(3n-1)} \Rightarrow \mid\ \overline{\Gamma}_1, \cdots, \overline{\Gamma}_{n-1}, \overline{\Sigma} \Rightarrow \overline{\Pi}}\ (wnm)^n$$

Let HWNM^n *be the hypersequent calculus for* $\mathbf{WNM}^n$ *obtained by extending HMTL with the rule above. For instance, HWNM*^{2} *is the calculus obtained by extending HMTL with the rule*

$$\frac{\{G \mid \overline{\Gamma}_1^2, \overline{\Gamma}_i^2, \overline{\Sigma} \Rightarrow \overline{\Pi}\}_{1\leq i\leq 5} \qquad \{G \mid \overline{\Gamma}_i^2, \overline{\Gamma}_{i+1}^2, \overline{\Sigma} \Rightarrow \overline{\Pi}\}_{2\leq i\leq 4} \qquad \{G \mid \overline{\Gamma}_2^2, \overline{\Gamma}_5^2, \overline{\Sigma} \Rightarrow \overline{\Pi}\}}{G \mid \overline{\Gamma}_2, \overline{\Gamma}_3, \overline{\Gamma}_4, \overline{\Gamma}_5 \Rightarrow\mid \overline{\Gamma}_1, \overline{\Sigma} \Rightarrow \overline{\Pi}}\ (wnm)^2$$

which is equivalent to the axiom $\neg(\varphi\cdot\psi\cdot\varphi\cdot\psi) \vee ((\varphi\wedge\psi) \supset (\varphi\cdot\psi\cdot\varphi\cdot\psi))$.

In the next step we identify a property that, when satisfied by the hypersequent rules generated using the algorithm in [52], ensures density elimination of the corresponding extensions of *HMTL*. Rules satisfying this property are called *convergent*.

Step (b): Sufficient Condition for Density Elimination

Given a sequent S henceforth we will denote by $L(S)$ its left hand side (the antecedent) and by $R(S)$ its right hand side (the succedent). Let $S := \Gamma_1, \Gamma_2 \Rightarrow \Pi$, we indicate by $S[^{\Gamma_1}/_{\Lambda}]^l[^{\Pi}/_{\Sigma\Rightarrow\Psi}]^r$ the sequent $\Lambda, \Gamma_2, \Sigma \Rightarrow \Psi$. The notations apply also to metasequents, i.e., sequents built from metavariables.

In what follows we will refer to any hypersequent rule generated by the procedure in [52] as *completed.*

Definition 23. Let r be a completed hypersequent rule of the form

$$\frac{G \mid S_1 \quad \ldots \quad G \mid S_m}{G \mid C_1 \mid \ldots \mid C_q}$$

and let $G \mid S_i$ and $G \mid S_j$ be among its premises.

(0) $G \mid S_i$ is a *0-pivot* if there is an $s \in \{1, \ldots, q\}$ such that $R(S_i) = R(C_s)$ and the different metavariables in $L(S_i)$ are contained in those of $L(C_s)$.

(n) $G \mid S_j$ is an *n-pivot for* $G \mid S_i$, for $n > 0$, if the following conditions hold:

- $G \mid S_j$ is a *0-pivot*
- $R(S_i) = R(S_j)$
- $L(S_j) = L(S_i[^{\overline{\Gamma}_1}/_{\overline{\Delta}_1}, \ldots ^{\overline{\Gamma}_n}/_{\overline{\Delta}_n}]^l)$ for $\overline{\Gamma}_1, \ldots \overline{\Gamma}_n \in L(S_i)$ and $\overline{\Delta}_1, \ldots \overline{\Delta}_n \in L(S_j)$
- If $n > 1$, $G \mid S_j$ is a *(n-1)-pivot* for n premises $G \mid S_{j_1} \ldots G \mid S_{j_n}$, and $L(S_j) = L(S_{j_i}[^{\overline{\Gamma}_1}/_{\overline{\Delta}_1}, \ldots, ^{\overline{\Gamma}_{i-1}}/_{\overline{\Delta}_{i-1}}, ^{\overline{\Gamma}_{i+1}}/_{\overline{\Delta}_{i+1}}, \ldots, ^{\overline{\Gamma}_n}/_{\overline{\Delta}_n}]^l)$ for $\overline{\Gamma}_1, \ldots, \overline{\Gamma}_{i-1}, \overline{\Gamma}_{i+1}, \ldots, \overline{\Gamma}_n \in L(S_{j_i})$, $\overline{\Delta}_1, \ldots, \overline{\Delta}_{i-1}, \overline{\Delta}_{i+1}, \ldots, \overline{\Delta}_n \in L(S_j)$ and $i = 1, \ldots, n$.

Example 13. *Recall the rule* $(wnm)^2$ *from Example 12:*

$$\frac{\{G \mid \overline{\Gamma}_1^2, \overline{\Gamma}_i^2, \overline{\Sigma} \Rightarrow \overline{\Pi}\}_{1\leq i\leq 5} \qquad \{G \mid \overline{\Gamma}_i^2, \overline{\Gamma}_{i+1}^2, \overline{\Sigma} \Rightarrow \overline{\Pi}\}_{2\leq i\leq 4} \qquad \{G \mid \overline{\Gamma}_2^2, \overline{\Gamma}_5^2, \overline{\Sigma} \Rightarrow \overline{\Pi}\}}{G \mid \overline{\Gamma}_2, \overline{\Gamma}_3, \overline{\Gamma}_4, \overline{\Gamma}_5 \Rightarrow \mid \overline{\Gamma}_1, \overline{\Sigma} \Rightarrow \overline{\Pi}} \, (wnm)^2$$

- *All different metavariables in the premise* $P_1 = G \mid \overline{\Gamma}_1^2, \overline{\Gamma}_1^2, \overline{\Sigma} \Rightarrow \overline{\Pi}$ *are contained in the component* $\overline{\Gamma}_1, \overline{\Sigma} \Rightarrow \overline{\Pi}$ *of the conclusion of the rule* $(wnm)^2$*. Therefore,* P_1 *is a* 0-pivot.
- *The premise* P_1 *is a* 1-pivot *for all premises* $G \mid \overline{\Gamma}_1^2, \overline{\Gamma}_i^2, \overline{\Sigma} \Rightarrow \overline{\Pi}, 2 \leq i \leq 5$ *as they differ from* P_1 *only by one metavariable.*
- P_1 *is a* 2-pivot *for the remaining premises of* $(wnm)^2$.

Definition 24. A completed hypersequent rule r is *convergent* if for each premise $G \mid S_i$ one of the following conditions holds:

(1) $R(S_i) = \emptyset$,
(2) $G \mid S_i$ is a *0-pivot*, or
(3) there is a premise $G \mid S_j$ which is an *n-pivot* for $G \mid S_i$, with $n > 0$.

Intuitively, the conclusion of a convergent rule results from a "minimal interplay" among its premises. Indeed for a premise $G \mid S_i$, in which $R(S_i)$ is not empty, two cases can arise: either the metavariables contained in it are already present in a component of the rule's conclusion, or there is a premise $G \mid S_j$ having this property that allows us to obtain $G \mid S_i$ by suitable replacements of the metavariables.

Example 14. *All internal structural rules, e.g.* (c), (w), (w'), *and* (wem) *from Table 4.3,* (wnm) *from Example 11 and* $(wnm)^n$ *from Example 12 are convergent. The completed rules* (em) *and* $(bc2)$*, see Table 4.3, are not convergent.*

Note that the tool *AxiomCalc* (see Section 4.3.1) does not only implement the transformation procedure from Hilbert axioms to equivalent structural (hyper)sequent rules, but also performs the check for convergent rules automatically.

Below we sketch that *HMTL* extended with any set of convergent rules admits density elimination (the formal proofs can be found in Appendix A). Density elimination was proved in [24, 127] for various calculi, including *HMTL*. These proofs are calculi-specific and use heavy combinatorial arguments, in close analogy with Gentzen-style cut elimination proofs. A different method to eliminate applications of the density rule from derivations was introduced in [58] and is called *density elimination by substitution*. We use and refine this method for extensions of *HMTL* with large classes of external *hypersequent* rules.

The method in [58] works roughly as follows: Let d be a subderivation ending in the following uppermost application of density

$$\dfrac{\begin{array}{c}\vdots d' \\ G' \mid \Sigma, p \Rightarrow \Pi \mid \Lambda \Rightarrow p\end{array}}{G' \mid \Sigma, \Lambda \Rightarrow \Pi}\ (D)$$

(D) is removed by substituting the occurrences of p in d in an "asymmetric" way, according to whether p occurs in the left or in the right hand side of a sequent. More precisely, each component S of a hypersequent in d is replaced by $S[{}^{p}/_{\Lambda}]^l[{}^{p}/_{\Sigma\Rightarrow\Pi}]^r$. This way, the application of (D) above is simply replaced by (ec).

However, the resulting labelled tree, denoted by d^*, might not be a correct derivation anymore. The reason for that is that there might be external structural rules in d, different from (ew) and (ec), that mix the content of various conclusion components. This might lead to *p-axioms* in their premises, i.e. to hypersequents of the form

$$G \mid \Theta, p^k \Rightarrow p$$

Such p-axioms are derivable from axioms simply by using weakenings; the problem is that the asymmetric substitution on a p-axiom leads to a sequent that is no longer derivable in the same way, e.g., $G \mid \Theta, \Lambda^k, \Sigma \Rightarrow \Pi$.

The proof for density elimination in [58] was done for calculi containing only (ec), (ew) and (com) as external structural rules. Then the *only* problematic case was when in d one of the premises of (*com*) led to a p-axiom, which was handled by discarding in d^* this application of (*com*) and replacing it with a suitable (sub)derivation starting from the other premise.

We show below that the addition of a convergent rule r behaves well with respect to these asymmetric substitutions, even though r can manipulate more components at once (and, hence, might lead to a p-axiom). Indeed, assume that one or more premises of r lead to a p-axiom in d, e.g. $G \mid \Theta, p^k \Rightarrow p$. Then the *x-pivot* premises of r can be used to derive the substituted version $G \mid \Theta, \Lambda^k, \Sigma \Rightarrow \Pi$, allowing to correctly apply r.

A (D)-*free derivation* is a derivation not containing the (D)-rule. The following lemma, which allows us to suitably "move" multisets of formulas between components, is the key for our main proof of the theorem of density elimination. Both proofs can be found in the Appendix A.

Lemma 3. *Let R be any set of convergent rules extending the calculus HMTL and let H be the calculus defined by HMTL+R.*

1. *Any derivation d of H can be transformed into a derivation of $H[{}^{p}/_{\alpha}]^{l}[{}^{p}/_{\Rightarrow\alpha}]^{r}$, for any formula α and propositional variable p.*
2. *Let d' and d_1 be derivations of $G' \mid \Sigma, p \Rightarrow \Pi \mid \Lambda \Rightarrow p$ (where $p \notin G', \Sigma, \Pi, \Lambda$) and $G' \mid \Theta, \Delta \Rightarrow \Psi$. We can find a derivation of $G' \mid \Theta, \Lambda \Rightarrow \Psi \mid \Sigma, \Delta \Rightarrow \Pi$.*

Theorem 4 (Density Elimination)**.** *HMTL extended with any set R of convergent rules admits density elimination, i.e., there is a procedure which transforms every derivation of HMTL+R into a (D)-free derivation of HMTL+R with the same end sequent.*

Step (c): From Density Elimination to Standard Completeness

Theorem 4 together with the results in [127, 53] lead to standard completeness for any logic **L** extending **MTL** with any set **Ax** of axioms having equivalent convergent rules. We refer to [127, 53, 85] for more detail on concepts of universal algebra.

As shown in [127], density elimination is indeed a uniform method to establish rational completeness for any extension of **MTL**. From Theorem 4 we can therefore state the following: let $\mathcal{A}_{\mathbf{L}}$ be an **MTL**-algebra (see [78] and steps 1-4 at the beginning of this section) satisfying the equations $\mathcal{E}_{\mathbf{Ax}}$ corresponding to axioms in **Ax**. Then for every formula φ we have:

$$\varphi \text{ is satisfied in each dense } \mathcal{A}_{\mathbf{L}}\text{-chain} \Leftrightarrow \varphi \text{ is derivable in } \mathbf{MTL}{+}\mathbf{Ax}.$$

Standard completeness is then achieved through Dedekind-MacNeille completion. It is shown, e.g. in [127], that the Dedekind-MacNeille completion of a dense **MTL**-chain is still a dense **MTL**-chain — in other words, the property of being a dense **MTL**-chain is preserved by the Dedekind-MacNeille-completion. The results in [53] on the preservation of equations by this completion hold for the equations $\mathcal{E}_{\mathbf{Ax}}$ when restricting to **MTL**-chains. Hence the Dedekind-MacNeille completion of a dense $\mathcal{A}_{\mathbf{L}}$-chain is still a dense $\mathcal{A}_{\mathbf{L}}$-chain, and, in addition, it is order-isomorphic to $[0, 1]$. This leads to *standard completeness* for the logic **L**.

Corollary 2. *Let* **L** *be a logic extending* **MTL** *with a set of axioms* **Ax** *such that every axiom* $\varphi \in \mathbf{Ax}$ *corresponds to a set of convergent rules by the algorithm of Theorem 3. Then* **L** *is standard complete.*

Example 15. *From our results (Corollary 2) it follows that the family of logics obtained by extending* **MTL** *with the axioms* $\neg(\varphi \cdot \psi)^n \vee ((\varphi \wedge \psi)^{n-1} \supset (\varphi \cdot \psi)^n)$ *for any* $n \geq 2$ *from Example 12 are standard complete and hence they are fuzzy logics in the sense of [97]. This new family of logics contains infinitely many different logics. This can be easily seen by noticing that the axiom above for any* n *is valid in the* m*-valued logic of Łukasiewicz if and only if* $m \leq n + 1$*.*

Note that our results allows us to prove standard completeness for (infinitely) many logics and also allows for the automated discovery of new fuzzy logics.

4.3.1 Tool: *AxiomCalc*

The **TINC**-tool *AxiomCalc* implements the systematic procedure introduced in [52] (and recalled in Section 4.2) and the check for the sufficient conditions for standard completeness presented in Section 4.3. *AxiomCalc* takes as input a Hilbert axiom specified in the language of Full Lambek calculus with exchange and weakening **FLew** and, if possible, transforms it into equivalent analytic sequent or hypersequent rules. If required by the user, the tool also checks whether the logic obtained by extending monoidal t-norm logic **MTL** [78] with this axiom is a fuzzy logic in the sense of [97] (i.e., whether the resulting rule is convergent).

For example, *AxiomCalc* can be used to construct analytic calculi for Gödel logic [91], the logics of Kripke models with k worlds $\mathbf{Bc}_k$ [49] or the logics of Kripke models with width $\leq k$ $\mathbf{Bw}_k$ [49]. Moreover, it was used to introduce an analytic calculus for the standard complete weak nilpotent minimum logic **WNM** [78]. By experimenting with the tool, we discovered the new family of fuzzy logics defined by extending **MTL** with the axioms $\neg(\varphi \cdot \psi)^n \vee ((\varphi \wedge \psi)^{n-1} \supset (\varphi \cdot \psi)^n)$ for $n \geq 2$, see Section 4.3.

AxiomCalc is available at

`http://www.logic.at/tinc/webaxiomcalc/`

Example

The main screen of the tool *AxiomCalc* is depicted in Figure 4.2. The user can enter in the text field an axiom according to the syntax (see below). Moreover, he can check if the resulting logic is a fuzzy logic by ticking the checkbox "Check for Standard Completeness".

After the computation, a dialog box containing the results pops up, see Figure 4.3. It contains the class of the axiom in the substructural hierarchy, the computed rule in text format, as well as a link to the generated paper containing the obtained calculus along with a basic description of the system.

When the program is started via the command-line, the user simply types `compute` (for the check of standard completeness: `computesc`) and enters the axiom. While the

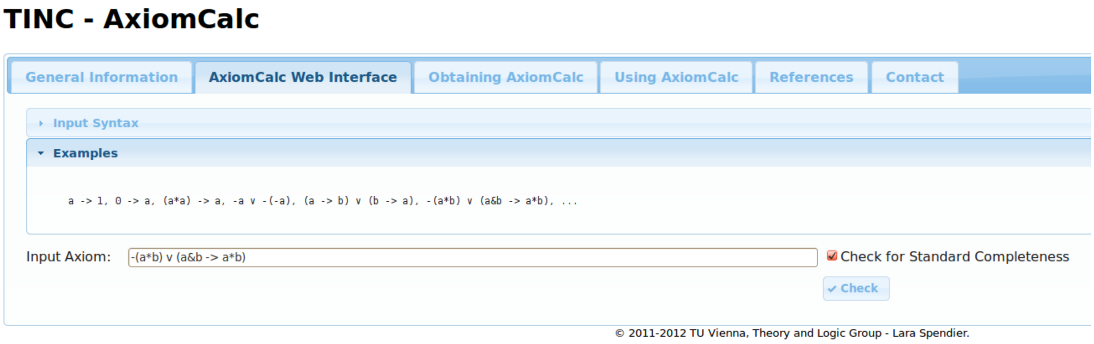


Figure 4.2: Main screen of *AxiomCalc*

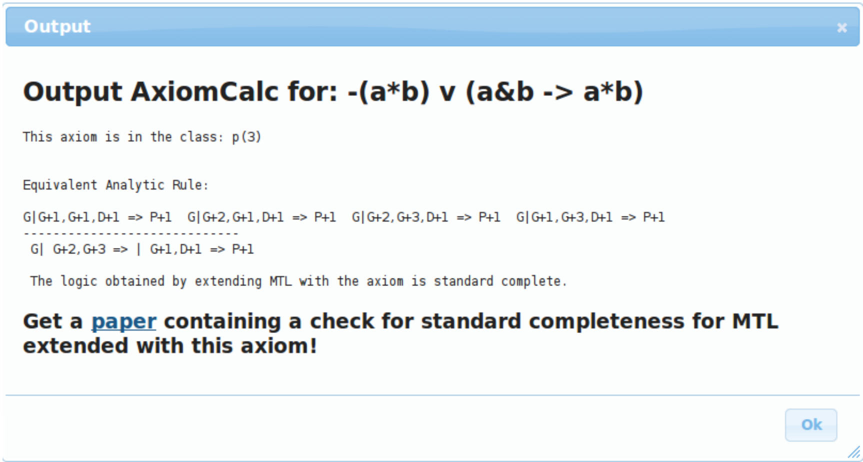


Figure 4.3: Dialog box containing the results

class of the axiom in the substructural hierarchy and the corresponding rule is printed on the screen, the LaTeX-file is saved in a program folder on the computer. Note that in the text representation of the rule, `G+i` and `D+i` (`P+i`, resp.) for any $i \geq 1$ stands for multisets of formulas Γ_i and Δ_i (or Π_i which contains at most one formula, resp.) and `G` stands for a side hypersequent. Below we show the output generated for the formula $\neg(\varphi \cdot \psi) \vee (\varphi \wedge \psi \supset \varphi \cdot \psi)$:

```
?- compute.
|: -(a*b) v (a&b -> a*b).

This axiom is in the class: p(3)
```

```
Equivalent Analytic Rule:

G|G+1,G+1,D+1 => P+1  G|G+2,G+1,D+1 => P+1
G|G+2,G+3,D+1 => P+1  G|G+1,G+3,D+1 => P+1
-----------------------------
G| G+2,G+3 => | G+1,D+1 => P+1
```

Implementation Details

AxiomCalc is implemented in Prolog. The implementation consists of 11 files and roughly 1700 lines of code (including documentation). The implementation of *AxiomCalc* follows the general **TINC**-structure described in Chapter 3 (recall Figure 3.5). The specific instantiation for *AxiomCalc* is depicted in Figure 4.4.

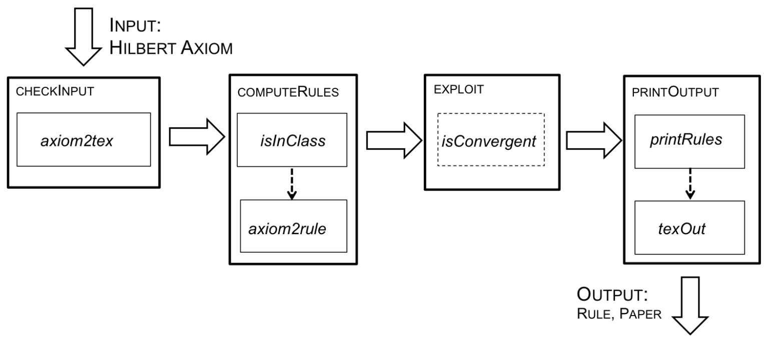


Figure 4.4: Design of *AxiomCalc*

Input and CHECKINPUT. The input formula is provided as a parameter to the first component, CHECKINPUT, which ensures that the axiom has the correct form to be handled by the algorithm. The syntax of the input formula is as follows:

- the letters `[a-z]` except `v` for (atomic) formulas
- `bot` and `top` for logical constants
- logical connectives: `&` (additive and), `*` (fusion/multiplicative and), `v` (or), `->` (implication) and `-` (negation).

The component CHECKINPUT implements a syntactic check of the input formula using a definite clause grammar:

Code Example 1. *We use* `axiom2tex` *as start symbol of the DCG. On the right side of the arrow we can see the translation into LaTeX code. In the following code snippet, we only show the definition of the start symbol and omit the definition of the other nonterminal symbols (*`axiom2texP1`*,* `axiom2texP2`*, ...,* `axiom2texP5` *and* `bs`*, which stands for* *the ASCII code of the backslash \).*

```
axiom2tex(-X)     --> " ", bs, "lnot ", axiom2texP1(X).
axiom2tex(X & Y)  --> axiom2texP2(X), " ", bs, "land ", axiom2texP2(Y).
axiom2tex(X v Y)  --> axiom2texP3(X), " ", bs, "lor ", axiom2texP3(Y).
axiom2tex(X * Y)  --> axiom2texP4(X), " ", bs, "cdot ", axiom2texP4(Y).
axiom2tex(X -> Y) --> axiom2texP5(X), " ", bs, "ra ", axiom2texP5(Y).
axiom2tex(top)      --> bs,"top".
axiom2tex(bot)      --> bs,"bot".
axiom2tex(a)    --> bs,"alpha".
axiom2tex(b)    --> bs,"beta".
...
```

computeRules and exploit. The second component, computeRules, contains the implementation of the algorithm:

- `isInClass` identifies the class of the input axiom within the substructural hierarchy and indicates it as `n(i)` for $\mathcal{N}_i$ or `p(i)` for $\mathcal{P}_i$.
- If the class of the axiom identified before is within $\mathcal{P}_3$, `axiom2rule` transforms the axiom into equivalent analytic (hyper)sequent rules.

The third component exploit is optional and will only be called when requested by the user. It contains the goal `isConvergent` that checks if the new rule is a convergent rule according to Definition 24.

Code Example 2. *Below we show the implementation of* `isConvergent`. *We check*

(1) `is_empty_RHS`*: if the right hand side is empty,*
(2) else `is_pivot0`*: if the sequent is a 0-pivot, and*
(3) else `is_npivot_check`*: if there exists another premise which is an n-pivot for the sequent.*

```
%% isConvergent(+[Prem,Con], -Convergent, -[PremR,ConvR])
%%    + ... parameter given as input, - ... return value
%%    [Prem,Con] ... premises and conclusion of the analytic rule
%%        that has to be checked if it is convergent
%%    Convergent ... 1 if premises are convergent, 0 otherwise
%%    [PremR,ConR]... resulting premises and conclusion

isConvergent([Premises,Con], Convergent, [PremisesR,ConvR]) :-
    remove_double(Premises, [], Prem),
```

```
    is_empty_RHS(Prem, ConvPrem),       % check condition (1)
    length(ConvPrem, LConvPrem),
    length(Prem, LPrem),
    ( (LPrem = LConvPrem) ->
      Convergent = 1,
      PremisesR = [],
      ConvR = []
    ; % else: check condition (2)
      is_pivot0(Prem, Con, ConvPrem, ConvPrem1),
      length(ConvPrem1, LConvPrem1),
      ( (LPrem = LConvPrem1) ->
        Convergent = 1,
        PremisesR = [],
        ConvR = [[0, ConvPrem1]]
      ; ( (LConvPrem1 = LConvPrem) ->
          Convergent = 0,
          PremisesR = Prem,
          ConvR = []
        ; %else: check condition (3)
          is_npivot_check(Prem, ConvPrem1, Check, CountP),
          ( (Check = 0) ->
            Convergent = 0,
            PremisesR = Prem,
            ConvR = []
           ; % differentiate between 0-pivot, 1-pivot, ... premises
            ConvPrem2 = [[0, ConvPrem1]],
            recursive_pivotn(Prem, ConvPrem2, ConvPrem3, CountP, 1),
            is_convergent_length(ConvPrem3, LConvPrem3),
            ( (LPrem = LConvPrem3) ->
                Convergent = 1
              ; Convergent = 0
            ),
            PremisesR = Prem,
            ConvR = ConvPrem3
    )))).
```

Output and printOutput. The last component, printOutput, contains the method to print the generated rule on the command-line or web interface (`printRules`); moreover, the method `texOut` generates a LaTeX-paper containing the resulting analytic (hyper-)sequent calculus and, if requested, the results of the check for standard completeness.

The method `texOut` is used for the generation of the LaTeX-paper. It takes as arguments the input formula and the computed rules and rewrites them into LaTeX-code by using the DCG that we already used in the CHECKINPUT-component. The (LaTeX version of the) axioms and rules are written to a style file, which is included in the main tex file. However, the LaTeX code of the axioms and rules is not the only thing that needs to be "written" by the tool: according to the results, also the contents of the final tex file need to be adjusted. For example, in *AxiomCalc*, the base calculus could be either a sequent or a hypersequent calculus, according to the input axiom.

Code Example 3. *We show below how the* `texOut` *goal is implemented in AxiomCalc for axioms that are transformed into hypersequent rules.*

```
%% texOut(+Axiom, +Rules, +C)
%%  + ... parameter given as input, - ... return value
%%  Axiom   ... Axiom the user provided
%%  Rules   ... Analytic rules for Axiom
%%  C       ... Class of the Axiom
texOut(Axiom, Rules, C) :-
    member(C, [p(2),p(3)],      % C is either p(2) or p(3)
    axiom2tex(Axiom,TAxiom,[]), % use the DCG to rewrite the Axiom
    tell('tex/AxiomCalc.sty'),  % open the .sty file
    length(Rules, N),           % number of rules
    % retrieve content according to the number of rules:
    constants2tex("HFLew", N, C),
    print_phrase(texNewAxiom(TAxiom)), % write to .sty file
    print_phrase(texNewAxiomNotMath(TAxiom)), % write to .sty file:
    nl, print_phrase(texNewRuleStart), % write to .sty file:
    createHSeqRules(Rules),    % create LaTeX code for hypersequent rules
    print_phrase(texNewRuleEnd), %write to .sty file
    told.  %close .sty file
```

CHAPTER 5

Intermediate Logics

Intermediate logics lie between intuitionistic and classical logic. They can be described in two different ways by using *syntactic* or *semantic* methods. Syntactically, they are axiomatic extensions of (propositional) intuitionistic logic **Int**. Semantically, they are defined by imposing additional conditions to the accessibility relation $\leqslant$ of the standard intuitionistic Kripke frame [49]. Intermediate logics are closely connected to modal logics, from a technical and a philosophical viewpoint [49]. Moreover, some intermediate logics have found applications in computer science, e.g. in non-monotonic reasoning and logic programming [146, 143], and some others are among the most important formalizations of fuzzy logic [97].

In this chapter, we present two approaches for the introduction of analytic calculi for intermediate logics. The two approaches are distinguished by their starting point. The first combines an automated procedure with a heuristic method and is based on the syntactic definition of the logic. As an example, we apply this method to the Hilbert system for the logic $\mathbf{Bd}_2$, i.e. the logic of frames with bounded depth of at most 2 [49]. This way, we give a first analytic hypersequent calculus for this logic. The second approach is a systematic procedure that is based on the semantic specification of the considered logic. It generalizes the method in [76]. Finally, we present the **TINC**-tool *Framinator* that implements the latter approach for a large class of intermediate logics.

We start by settling the basic notions and give some examples of (propositional) intermediate logics. In Section 5.2, we give an overview of related work on the (automated) introduction of analytic calculi for intermediate logics. The following two sections contain our theoretical contributions: In Section 5.3, we introduce a (heuristic) method to create logical hypersequent rules out of Hilbert axioms. Section 5.4 contains the systematic procedure to transform frame conditions defining intermediate logics into labelled rules, hence obtaining cut-free labelled sequent calculi for a large class of intermediate logics. Section 5.4.1 contains a description of the tool *Framinator* that implements the systematic procedure to generate labelled calculi.

The results of this chapter are based on [57].

5.1 Preliminaries

The language of propositional intermediate logics is the language $\mathcal{L}_{int}$ of Section 2.1.

An *intuitionistic frame* is a pair $\mathfrak{F} = \langle W, \leqslant \rangle$ where W is a non-empty set (the set of possible worlds), and $\leqslant \subseteq W \times W$ is a reflexive and transitive (accessibility) relation on W. An *intuitionistic model* is a triple $\mathfrak{M} = \langle W, \leqslant, \Vdash \rangle$, where $\Vdash$ is a binary relation (called *forcing*) between elements of W, the possible worlds, and atomic formulas. Intuitively, $x \Vdash p$ means that the atom p is true (holds) at world x. Forcing is assumed to be monotonic w.r.t. the relation $\leqslant$, namely, if $x \leqslant y$ and $x \Vdash p$ then also $y \Vdash p$. The forcing relation is defined inductively on arbitrary formulas as follows:

$(\Vdash \bot)$	$x \Vdash \bot$	for no x	
$(\Vdash \wedge)$	$x \Vdash \varphi \wedge \psi$	iff	$x \Vdash \varphi$ and $x \Vdash \psi$
$(\Vdash \vee)$	$x \Vdash \varphi \vee \psi$	iff	$x \Vdash \varphi$ or $x \Vdash \psi$
$(\Vdash \supset)$	$x \Vdash \varphi \supset \psi$	iff	$x \leqslant y$ and $y \Vdash \varphi$ implies $y \Vdash \psi$.

Frame conditions are imposed on the relation $\leqslant$ on intuitionistic Kripke frames to define intermediate logics semantically. They are usually expressed as formulas in the language of first-order classical logic: Atomic formulas are *relational atoms* of the form $x \leqslant y$, denoted by (possibly indexed) P, Q. Compound formulas $A, B, C, M, \ldots$ are built from relational atoms using the propositional connectives & (conjunction), $\vee$ (disjunction), $\rightarrow$ (implication), $\neg$ (negation), and the quantifiers $\forall$ (universal quantifier) and $\exists$ (existential quantifier). Variables are interpreted as elements of W and the binary predicate $\leqslant$ denotes the accessibility relation of $\mathfrak{F}$.

We use $Q[x/y]$ ($\Gamma[x/y]$) to denote Q (Γ) after substituting the variable x for the variable y.

Labelled Sequent Calculus

The *labelled sequent calculus* [83, 167, 134] is a semantic formalism generalizing Gentzen's sequent calculus. In a labelled sequent calculus, the logic semantics is made explicit part of the syntax. Each formula φ receives a label x, written as $x : \varphi$. The labels are interpreted as possible worlds, and a labelled formula $x : \varphi$ intuitively means that φ holds in world x, i.e. $x \Vdash \varphi$. Moreover, labels may occur also in expressions for the accessibility relation (relational atoms) like, e.g., $x \leqslant y$.

Definition 25. A *labelled sequent* is a sequent consisting of labelled formulas and relational atoms.

As in the case of sequent and hypersequent calculus, the rules of a labelled sequent calculus consist of initial axioms, logical rules and structural rules. Moreover, there are also rules that correspond to the peculiar frame conditions of the considered logics; as an example, see the rules (ref) and $(trans)$ in Table 5.1 for relational atoms corresponding to the assumptions of reflexivity and transitivity of the accessibility relation $\leqslant$.

$$x \leqslant y, x : p, \Gamma \Rightarrow \Delta, y : p \qquad \frac{x : \varphi, x : \psi, \Gamma \Rightarrow \Delta}{x : \varphi \wedge \psi, \Gamma \Rightarrow \Delta}\ (\wedge, l) \qquad \frac{\Gamma \Rightarrow \Delta, x : \varphi \quad \Gamma \Rightarrow \Delta, x : \psi}{\Gamma \Rightarrow \Delta, x : \varphi \wedge \psi}\ (\wedge, r)$$

$$\frac{}{x : \bot, \Gamma \Rightarrow \Delta}\ (\bot, l) \qquad \frac{\Gamma \Rightarrow \Delta, x : \varphi, x : \psi}{\Gamma \Rightarrow \Delta, x : \varphi \vee \psi}\ (\vee, r) \qquad \frac{x : \varphi, \Gamma \Rightarrow \Delta \quad x : \psi, \Gamma \Rightarrow \Delta}{x : \varphi \vee \psi, \Gamma \Rightarrow \Delta}\ (\vee, l)$$

$$\frac{x \leqslant x, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}\ (ref) \qquad \frac{x \leqslant y, y : \varphi, \Gamma \Rightarrow \Delta, y : \psi}{\Gamma \Rightarrow \Delta, x : \varphi \supset \psi}\ (\supset, r) \qquad \frac{x \leqslant z, x \leqslant y, y \leqslant z, \Gamma \Rightarrow \Delta}{x \leqslant y, y \leqslant z, \Gamma \Rightarrow \Delta}\ (trans)$$

$$\frac{x \leqslant y, x : \varphi \supset \psi, \Gamma \Rightarrow \Delta, y : \varphi \quad x \leqslant y, x : \varphi \supset \psi, y : \psi, \Gamma \Rightarrow \Delta}{x \leqslant y, x : \varphi \supset \psi, \Gamma \Rightarrow \Delta}\ (\supset, l)$$

y in the conclusion of $(\supset, r)$ is an eigenvariable

Table 5.1: Labelled calculus *G3I* [76]

Table 5.1 depicts the labelled sequent calculus *G3I* for **Int**. Note that its logical rules are obtained directly from the inductive definition of forcing and that the rule $(\supset, r)$ must satisfy the *eigenvariable* condition (y does not occur in the conclusion).

Definition 26. Let C_L be a labelled sequent calculus and R be a set of rules. We write $C_L + R$ to denote the labelled sequent calculus C_L extended with R. A derivation in a labelled sequent calculus is a labelled finite tree with a single root (called end sequent), axioms at the top nodes, and where each node is connected with the (immediate) successor nodes (if any) according to the inference rules.

For labelled sequents derived in $C_L + R$ we write

$$\vdash_{C_L+R} \Gamma \Rightarrow \Delta$$

The notation $\vdash_{C_L+R} \varphi$ is defined as $\vdash_{C_L+R} \Rightarrow \varphi$, i.e., derivability of the labelled sequent $\Gamma \Rightarrow \varphi$ with Γ being empty. If a labelled sequent S_0 is derivable from a set of labelled sequents $\mathcal{S}$ in $C_L + R$, we write

$$\mathcal{S} \vdash_{C_L+R} S_0$$

The notions of *height* of a derivation and *complexity* of a formula are as for the sequent calculus. Moreover, we also need the *equivalence* of rules:

Definition 27. Two labelled rules r and r' are *equivalent* (in *G3I*) if the derivability relations $\vdash_{G3I+r}$ and $\vdash_{G3I+r'}$ coincide, i.e., when the conclusion of r is derivable from its premises in $G3I + r'$ (and the conclusion of r' is derivable from its premises in $G3I + r$), then r and r' are equivalent. The definition naturally extends to sets of rules.

Finally, we introduce the notions of *height-preserving admissibility* and *height-preserving invertibility*:

Name	Axiom
em	$\neg\varphi \vee \varphi$
wem	$\neg\varphi \vee \neg\neg\varphi$
prel	$(\varphi \supset \psi) \vee (\psi \supset \varphi)$
bw_n	$\bigvee_{i=0}^{n}(\varphi_i \supset \bigvee_{j \neq i} \varphi_j)$
bd_n	$bd_1 = \varphi_1 \vee (\varphi_1 \supset \varphi_0)$, and $bd_n = \varphi_n \vee (\varphi_n \supset bd_{n-1})$
bd_2	$\varphi_2 \vee (\varphi_2 \supset \varphi_1 \vee (\varphi_1 \supset \varphi_0))$
bc_n	$\varphi_0 \vee (\varphi_0 \supset \varphi_1) \vee \ldots \vee (\varphi_0 \wedge \ldots \wedge \varphi_{n-1} \supset \varphi_n)$
gs	$(\varphi \supset \psi) \vee (\psi \supset \varphi) \vee ((\varphi \supset \neg\psi) \wedge (\neg\psi \supset \varphi))$
sm	$(\neg\psi \supset \varphi) \supset (((\varphi \supset \psi) \supset \varphi) \supset \varphi)$
kp	$(\neg\varphi \supset (\psi \vee \chi)) \supset ((\neg\varphi \supset \psi) \vee (\neg\varphi \supset \chi))$

Table 5.2: (Schematic) Hilbert axioms defining intermediate logics

Definition 28. Whenever derivability of the premises of a rule implies derivability of its conclusion with at most the same derivation height, the rule is called *height-preserving admissible* (abbreviated hp-admissible).

Let $S_1, \ldots, S_n, S$ be labelled sequents and r a labelled sequent rule of *G3I*. r is *height-preserving invertible* (abbreviated hp-invertible) if for each instance

$$\frac{S_1 \quad \cdots \quad S_n}{S}$$

of r, whenever $\vdash_{G3I} S$, then $\vdash_{G3I} S_i$ for $i = 1, \ldots, n$ with at most the same derivation height.

Examples of Intermediate Logics

(Propositional) Intermediate logics lie between (propositional) intuitionistic logic **Int** and (propositional) classical logic **Cl**. The latter is considered to be the strongest intermediate logic in which the accessibility relation $\leqslant$ is an equivalence relation.

Below we describe some intermediate logics and present their syntactic (Hilbert axioms) and semantic (frame conditions) definitions.

Logic		Axiomatization*	Semantic Characterization
Cl	Classical logic	**Int**+ *em*	$\forall x, y(x \leqslant y \to y \leqslant x)$
LQ	Jankov (or De Morgan) logic [100] is the logic of strongly directed frames.	**Int**+ *wem*	$\forall x, y, z((x \leqslant y \& x \leqslant z) \to \exists w(y \leqslant w \& z \leqslant w))$

GD	Gödel-Dummett logic [91, 71] is the logic of linear order and one of the main fuzzy logics.	**Int**+*prel*	$\forall x, y, z((x \leqslant y \& x \leqslant z) \to (y \leqslant z \vee z \leqslant y))$
$\mathbf{Bc}_n$	are the logics of frames with cardinality at most n for $n \geq 1$ [49].	**Int**+bc_n	$\forall x_0, x_1, \ldots, x_n(\&_{i=1}^{n}(x_0 \leqslant x_i) \to \bigvee_{i \neq j}(x_i \leqslant x_j \& x_j \leqslant x_i))$
$\mathbf{G}_n$	$(n+1)$-valued Gödel logics [91] are the logics of linear frames with at most n nodes.	**Int**+$prel + bc_n$	frame conditions of **GD** and $\mathbf{Bc}_n$
$\mathbf{Bw}_n$	are the logics of frames with bounded width of at most n for $n \geq 1$ [49]. $\mathbf{Bw}_1$ coincides with **GD**.	**Int**+bw_n	$\forall x, x_0, \ldots, x_n(\&_{i=0}^{n}(x \leqslant x_i) \to \bigvee_{i \neq j}(x_i \leqslant x_j \& x_j \leqslant x_i))$
$\mathbf{Bd}_n$	are the logics of frames with bounded depth of at most n for $n \geq 1$ [49].	**Int**+ bd_n	$\forall x_0, \ldots, x_n(\&_{i=0}^{n-1}(x_i \leqslant x_{i+1}) \to \bigvee_{i \neq j}(x_i \leqslant x_j \& x_j \leqslant x_i))$
$\mathbf{Bd}_2$	is the logic of frames with bounded depth at most 2.	**Int**+bd_2	$\forall x, y, z((x \leqslant y \& y \leqslant z) \to (y \leqslant x \vee z \leqslant y))$
GS	is the greatest semi-constructive logic (also called $\mathbf{Bd}_2\mathbf{F}_2$) [79].	**Int**+$bd_2 + gs$	frame condition of $\mathbf{Bd}_2$ and $\forall x, y, z \exists v((x \leqslant v \& y \leqslant v) \vee (y \leqslant v \& z \leqslant v) \vee (x \leqslant v \& z \leqslant v))$
SM	Smetanich logic [49] is the greatest intermediate logic that is properly included in **Cl**. It is also known as 3-valued Gödel logic.	**Int**+*sm* **Int**+$prel + bd_2$	frame conditions of **GD** and $\mathbf{Bd}_2$
KP	Kreisel-Putnam logic [111]	**Int**+*kp*	$\forall x, y, z((x \leqslant y \& x \leqslant z) \to (y \leqslant z \vee z \leqslant y \vee \exists u(x \leqslant u \& u \leqslant y \& u \leqslant z \& F(u, y, z))))$ where $F(u, y, z) = \forall v(u \leqslant v \to \exists w(v \leqslant w \& (y \leqslant w \vee z \leqslant w)))$.

* The Hilbert axioms are given in Table 5.2 on page 60.

The intermediate logics **Int**, **Cl**, **LQ**, **GD**, $\mathbf{Bd}_2$, **GS** and **SM** are the only seven interpolable propositional intermediate logics [123] and were among the first intermediate logics to be introduced in the literature.

5.2 Related Work in Proof Theory

We give a brief overview of related work on the systematic introduction of analytic calculi for intermediate logics. We distinguish between syntactic and semantic approaches.

Syntactic Calculi

A formalism that has frequently been used to define calculi for intermediate logics is the tableau calculus, see [157, 80]. Tableau calculi can be easily obtained by dualizing sequent calculi. In [8] (duplication-free) tableau calculi and their related cut-free sequent calculi are defined for all the interpolable intermediate logics. As the introduction of these calculi was tailored to the specific logic at hand, a more systematic approach was presented in [51]. There, for the logics $\mathbf{Bw}_n$, $\mathbf{Bc}_n$, $\mathbf{G}_n$ and $\mathbf{LQ}$, the conditions characterizing these logics were translated into rules for hypertableaux, which are a natural generalization of tableaux, analogous to how hypersequents are a natural generalization of sequents. However, similar to the case of hypersequents, the same procedure cannot be applied to all intermediate logics, e.g. to $\mathbf{Bd}_2$. For the particular case of the logic $\mathbf{Bd}_n$, a *path-hypertableau calculus* was introduced, which is a hypertableau calculus where the order of the components matters. The path-hypertableau calculus for $\mathbf{Bd}_n$ is then defined by adding to the path-hypertableau calculus for **Int** a structural rule corresponding to the axiom characterizing $\mathbf{Bd}_n$.

As already mentioned in Section 4.2, the procedure in [52] was a first step in the direction of automated procedures for large classes of various non-classical logics. The results in [52] also include many intermediate logics, e.g. the logics **LQ**, **GD**, $\mathbf{Bw}_n$ or $\mathbf{Bc}_n$. The Hilbert axioms characterizing these logics are all within the class $\mathcal{P}_3$ of the substructural hierarchy and can thus be transformed into equivalent analytic hypersequent rules.

The systematic procedure from [52] has been adapted to *display calculi* in [59], transforming axioms into equivalent structural display rules. Note that hypersequent calculi can be translated into display calculi [152] and hence the results on display calculus subsume those for the hypersequent calculus. Moreover, the procedure proposed in [59] can also be applied to axioms that cannot be captured using the procedure in [52], e.g., the axioms defining the logic $\mathbf{Bd}_n$ for $n \geq 2$.

Semantic Calculi

A modular approach to define cut-free labelled calculi, which in particular applies to a large class of intermediate logics, has been proposed in [76, 131]. The resulting calculi are obtained by adding to the labelled intuitionistic system *G3I* (see Table 5.1) new rules, which correspond to the peculiar frame conditions of the considered logics.

The (formulas defining) frame conditions, to which the method in [76] applies, are called *regular* and *geometric formulas*. Regular formulas are conjunctions of formulas of the form

$$reg \qquad \forall \overline{x}(P_1 \& \ldots \& P_m \to Q_1 \vee \ldots \vee Q_n)$$

whereas geometric formulas consist of conjunctions of formulas of the shape

$$geom \qquad \forall\overline{x}(P_1\&\ldots\&P_m \to \exists\overline{y}(M_1 \vee \ldots \vee M_n))$$

where in both cases $\overline{x}, \overline{y}$ are sequences of bound variables, each P_i is a relational atom, each M_j is a conjunction of relational atoms $Q_{j_1}, \ldots, Q_{j_k}$ and $\overline{y}$ does not appear in $P_1, \ldots, P_m$. Note that regular formulas are equivalent to geometric formulas where $\overline{y}$ does not appear in M_i (for all $i = 1, \ldots, n$). As shown in [76], the rule scheme corresponding to regular formulas has the form

$$\frac{Q_1, P_1, \ldots, P_m, \Gamma \Rightarrow \Delta \quad \cdots \quad Q_n, P_1, \ldots, P_m, \Gamma \Rightarrow \Delta}{P_1, \ldots, P_m, \Gamma \Rightarrow \Delta}\ (reg)$$

while the rule scheme corresponding to geometric formulas is of the shape

$$\frac{\overline{Q}_1[z_1/y_1], P_1, \ldots, P_m, \Gamma \Rightarrow \Delta \quad \cdots \quad \overline{Q}_n[z_n/y_n], P_1, \ldots, P_m, \Gamma \Rightarrow \Delta}{P_1, \ldots, P_m, \Gamma \Rightarrow \Delta}\ (geom)$$

where each $\overline{Q_j}$ indicates the multiset of relational atoms $Q_{j_1}, \ldots, Q_{j_k}$ and $z_1, \ldots, z_n$ are *eigenvariables*, i.e., variables not occurring in the conclusion. Note that we refer to eigenvariables also as *fresh* variables.

Note that the accessibility relation $\leqslant$ in all intermediate logics presented in Section 5.1 (except **KP**) is characterized by universal or so-called geometric axioms corresponding to regular or geometric formulas.

Example 16 (**LQ**)**.** *The logic* ***LQ*** *is semantically characterized by the frame condition* $\forall xyz((x \leqslant y \& x \leqslant z) \to \exists w(y \leqslant w \& z \leqslant w))$. *The corresponding geometric rule of the form* $(geom)$ *according to [76] is* (w' *is a fresh variable*)*:*

$$\frac{y \leqslant w', z \leqslant w', x \leqslant y, x \leqslant z, \Gamma \Rightarrow \Delta}{x \leqslant y, x \leqslant z, \Gamma \Rightarrow \Delta}\ (geom_{lq})$$

Theorem 5 ([76])**.** *Cut is admissible in any extension of G3I by rules of the form* (reg) *and* $(geom)$. *Weakening and contraction are height-preserving (hp-) admissible and all rules are hp-invertible (see Definition 28).*

The work of [76] has been further extended in [133] where *systems of labelled rules* are employed. The idea behind a system of rules is that two or more rules are combined and need to be applied in a specific order in the derivation. The (generalized) geometric formulas that can be handled with systems of rules are inductively defined as follows

$$GF_0 \equiv \forall\overline{x}(\&P_i \to \exists\overline{y_1}M_1 \vee \cdots \vee \exists\overline{y_m}M_m)$$
$$GF_1 \equiv \forall\overline{x}(\&P_i \to \exists\overline{y_1}\&GA_0 \vee \cdots \vee \exists\overline{y_m}\&GF_0)$$
$$GF_{n+1} \equiv \forall\overline{x}(\&P_i \to \exists\overline{y_1}\&GF_{k_1} \vee \cdots \vee \exists\overline{y_m}\&GF_{k_m})$$

where $\overline{x}, \overline{y_i}$ are sequences of bound variables, $\& P_i$ denotes a conjunction of atomic formulas, each M_j is a conjunction of atomic formulas, the variables in $\overline{y_j}$ are not free in any P_i, $\& GF_{k_i}$ denotes a conjunction of GF_{k_i}-formulas and $k_1, \ldots, k_m \leq n$.

The corresponding geometric rule schemes (GRS_{n+1}) for the respective geometric formula GF_{n+1} is inductively defined as follows [133]:

$$\dfrac{\begin{array}{c}\Gamma'_1 \Rightarrow \Delta'_1 \\ \vdots \\ \mathcal{D}^1_{k_1} \\ \vdots \\ \Gamma''_1 \Rightarrow \Delta''_1 \\ \vdots \\ \mathcal{D}^1 \\ \vdots \\ z_1 = z_1, \overline{P}, \Gamma \Rightarrow \Delta \end{array} \quad \ldots \quad \begin{array}{c}\Gamma'_m \Rightarrow \Delta'_m \\ \vdots \\ \mathcal{D}^m_{k_m} \\ \vdots \\ \Gamma''_m \Rightarrow \Delta''_m \\ \vdots \\ \mathcal{D}^m \\ \vdots \\ z_m = z_m, \overline{P}, \Gamma \Rightarrow \Delta \end{array}}{\overline{P}, \Gamma \Rightarrow \Delta} \; (GRS_{n+1})$$

where z_i are eigenvariables, the derivations $\mathcal{D}^i_{k_i}$ use only logical rules and rules of the form (GRS_{k_i}) corresponding to the geometric formulas GF_{k_i}, and the derivations $\mathcal{D}^i$ use only logical rules.

Example 17 ([133]). *The frame condition for join semi-lattices has the form GF_1:*

$$\forall xy \exists z((x \leqslant z \& y \leqslant z) \& \forall w(x \leqslant w \& y \leqslant w \to z \leqslant w))$$

The corresponding system of rules is

$$\begin{array}{c}\dfrac{z \leqslant w, x \leqslant w, y \leqslant w, \Gamma' \Rightarrow \Delta'}{x \leqslant w, y \leqslant w, \Gamma' \Rightarrow \Delta'} \; (GRS_1^2) \\ \vdots \\ \dfrac{x \leqslant z, y \leqslant z, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta} \; (GRS_1^1)\end{array}$$

where z in (GRS_1^1) is fresh and (GRS_1^2) needs to be applied above (GRS_1^1).

Example 18 (**KP**). *The frame condition for* **KP** *has the form:*

$$\forall xyz((x \leqslant y \& x \leqslant z) \to (y \leqslant z \lor z \leqslant y \lor \exists u(x \leqslant u \& u \leqslant y \& u \leqslant z \& F(u, y, z))))$$

where $F(u, y, z)$ abbreviates $\forall v(u \leqslant v \to \exists w(v \leqslant w \& (y \leqslant w \lor z \leqslant w)))$.

The corresponding system of rules is:

$$\dfrac{x \leqslant y, x \leqslant z, y \leqslant z, \Gamma \Rightarrow \Delta \quad x \leqslant y, x \leqslant z, z \leqslant y, \Gamma \Rightarrow \Delta \quad \begin{array}{c}\dfrac{u \leqslant v, v \leqslant w, \Gamma' \Rightarrow \Delta' \quad u \leqslant v, y \leqslant w, z \leqslant w, \Gamma' \Rightarrow \Delta'}{u \leqslant v, \Gamma' \Rightarrow \Delta'} \\ \vdots \\ x \leqslant y, x \leqslant z, x \leqslant u, u \leqslant y, u \leqslant z, \Gamma \Rightarrow \Delta\end{array}}{x \leqslant y, x \leqslant z, \Gamma \Rightarrow \Delta}$$

where u is fresh in the lower rule and w is fresh in the upper rule.

Theorem 6 ([133]). *Cut is admissible in any extension of G3I by systems of rules following the geometric scheme (GRS_{n+1}).*

With the extension of the geometric rule scheme to the generalized geometric rule scheme for systems of rules, a larger class of intermediate logics can be captured (including the class of intermediate logics that is captured by our procedure in Section 5.4). Moreover, as pointed out in a remark in [133], the method to create systems of rules is not only applicable to frame conditions, but also to axioms without quantifiers that are in GF_1, but not in GF_0.

Example 19 ([133]). *The Hilbert axiom for prelinearity has the form GF_1:*

$$(P \to Q) \vee (Q \to P)$$

The corresponding system of rules is

$$\begin{array}{c} \dfrac{\dfrac{Q, P, \Gamma' \Rightarrow \Delta'}{P, \Gamma' \Rightarrow \Delta'} \qquad \dfrac{P, Q, \Gamma'' \Rightarrow \Delta''}{Q, \Gamma'' \Rightarrow \Delta''}}{} \\ \vdots \qquad\qquad\qquad \vdots \\ \dfrac{\Gamma \Rightarrow \Delta \qquad\qquad \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}\ (GRS_1) \end{array}$$

Note that the system of rules in the previous example has some similarity with Avron's communication rule [10], i.e.

$$\frac{G \mid \Gamma, \Gamma' \Rightarrow \Delta \qquad G \mid \Sigma', \Sigma \Rightarrow \Delta'}{G \mid \Gamma, \Sigma \Rightarrow \Delta \mid \Gamma', \Sigma' \Rightarrow \Delta'}\ (com)$$

5.3 Towards the Generation of Logical Hypersequent Rules

As mentioned in the previous section, the systematic procedure to generate structural rules from Hilbert axioms introduced in [52] does not work for axioms belonging to a class above $\mathcal{P}_3$, such as the axiom $bd_2 = \xi \vee (\xi \supset (\varphi \vee (\varphi \supset \psi))) \in \mathcal{P}_4$.

Example 20. *The axioms defining the logics $\mathbf{Bd}_n$ ($n > 1$), **SM** and **KP** are above the class $\mathcal{P}_3$ and hence cannot be handled by the algorithm described in Theorem 3.*

Thus, to try to capture these logics, one has to think of new ways to suitably adapt the systematic procedure in [52] to create rules out of axioms. Possible solutions are to change the base calculus, which was done e.g. in [62, 59], and/or to create *logical* instead of structural rules out of the given axioms, see e.g. the general strategy from [62] described in Section 5.2.

We now introduce a transformation procedure that combines a heuristic method with the procedure in [62]. Since adding logical rules to the base calculus requires some further

$$\varphi \Rightarrow \varphi \qquad \bot \Rightarrow \qquad \frac{G \mid \Gamma \Rightarrow \Delta \mid \Gamma \Rightarrow \Delta}{G \mid \Gamma \Rightarrow \Delta}\ (ec) \qquad \frac{G}{G \mid \Gamma \Rightarrow \Delta}\ (ew)$$

$$\frac{G \mid \Gamma \Rightarrow \varphi, \Delta \quad G \mid \Gamma, \psi \Rightarrow \Delta}{G \mid \Gamma, \varphi \supset \psi \Rightarrow \Delta}\ (\supset, l) \qquad \frac{G \mid \Gamma, \varphi \Rightarrow \psi}{G \mid \Gamma \Rightarrow \varphi \supset \psi, \Delta}\ (\supset, r) \qquad \frac{G \mid \Gamma \Rightarrow \Delta}{G \mid \Gamma \Rightarrow \bot, \Delta}\ (\bot, r)$$

$$\frac{G \mid \Gamma \Rightarrow \varphi, \Delta \quad G \mid \Gamma \Rightarrow \psi, \Delta}{G \mid \Gamma \Rightarrow \varphi \wedge \psi, \Delta}\ (\wedge, r) \qquad \frac{G \mid \varphi, \psi, \Gamma \Rightarrow \Delta}{G \mid \varphi \wedge \psi, \Gamma \Rightarrow \Delta}\ (\wedge, l) \qquad \frac{G \mid \Gamma \Rightarrow \varphi, \varphi, \Delta}{G \mid \Gamma \Rightarrow \varphi, \Delta}\ (c, r)$$

$$\frac{G \mid \varphi, \Gamma \Rightarrow \Delta \quad G \mid \psi, \Gamma \Rightarrow \Delta}{G \mid \varphi \vee \psi, \Gamma \Rightarrow \Delta}\ (\vee, l) \qquad \frac{G \mid \Gamma \Rightarrow \varphi, \psi, \Delta}{G \mid \Gamma \Rightarrow \varphi \vee \psi, \Delta}\ (\vee, r) \qquad \frac{G \mid \Gamma, \varphi, \varphi \Rightarrow \Delta}{G \mid \Gamma, \varphi \Rightarrow \Delta}\ (c, l)$$

$$\frac{G \mid \Gamma \Rightarrow \varphi, \Delta \quad H \mid \varphi, \Sigma \Rightarrow \Pi}{G \mid H \mid \Gamma, \Sigma \Rightarrow \Pi, \Delta}\ (cut) \qquad \frac{G \mid \Gamma \Rightarrow \Delta}{G \mid \Gamma \Rightarrow \varphi, \Delta}\ (w, r) \qquad \frac{G \mid \Gamma \Rightarrow \Delta}{G \mid \Gamma, \varphi \Rightarrow \Delta}\ (w, l)$$

Table 5.3: Multiple-conclusion hypersequent calculus *HLJm*

investigation on how the various logical rules interact with each other, we cannot provide uniform proofs of soundness, completeness and cut elimination. As a case study, we apply the combined method to the logic $\mathbf{Bd}_2$ and present ad-hoc proofs of these results. This way we define a first cut-free hypersequent calculus for this particular logic.

From Axioms to Logical Rules

Inspired by [154], we use as base calculus (the hypersequent version of) Maehara's calculus *LJm* for intuitionistic logic [160]. This is a multiple-conclusion version of *LJ* where the intuitionistic restriction, i.e., that the consequent of a sequent contains at most one formula, applies only to the right rule of $\supset$ (and $\forall$, in the first order case). The rule schemas for the hypersequent version of *LJm* (we call this calculus *HLJm*) are depicted in Table 5.3.

Our transformation procedure to create logical rules out of axioms again uses the following two key ingredients:

(1) the invertibility of the logical rules for the connectives $\wedge, \vee$, $(\bot, r)$ and the restricted invertibility of the rule $(\supset, r)$ [1] in *HLJm*, and
(2) the Ackermann lemma (see Lemma 4).

Lemma 4 ([52])**.** *The hypersequent* (0) $G \mid \varphi_1, \ldots, \varphi_n \Rightarrow \psi_1, \ldots, \psi_m$ *interpreted as a zero-premise rule is equivalent to the rules*

$$\frac{G \mid \psi_1, \Sigma \Rightarrow \Pi \quad \cdots \quad G \mid \psi_m, \Sigma \Rightarrow \Pi}{G \mid \varphi_1, \ldots, \varphi_n, \Sigma \Rightarrow \Pi}\ (1) \qquad \frac{G \mid \Gamma \Rightarrow \varphi_1 \quad \cdots \quad G \mid \Gamma \Rightarrow \varphi_n}{G \mid \Gamma \Rightarrow \psi_1, \ldots, \psi_m}\ (2)$$

[1] Note that an application of the rule $(\supset, r)$ in *HLJm* deletes the context in the succedent of the sequent. Hence, this rule is invertible only if the sequent in the conclusion is already single-conclusion.

where Γ, Σ, Π *are fresh metavariables for multisets of formulas.*

Proof. "(0) $\Rightarrow$ (1)": Follows by m applications of (cut) (and applications of contraction and weakening), i.e.

$$
\dfrac{\dfrac{\dfrac{\dfrac{G \mid \varphi_1, \dots, \varphi_n \Rightarrow \psi_1, \dots, \psi_m}{G \mid \varphi_1, \dots, \varphi_n, \Sigma \Rightarrow \Pi, \psi_1, \dots, \psi_m} \quad G \mid \psi_m, \Sigma \Rightarrow \Pi}{G \mid \varphi_1, \dots, \varphi_n, \Sigma \Rightarrow \Pi, \psi_1, \dots, \psi_{m-1}}\,(cut) \quad G \mid \psi_{m-1}, \Sigma \Rightarrow \Pi}{G \mid \varphi_1, \dots, \varphi_n, \Sigma \Rightarrow \Pi, \psi_1, \dots, \psi_{m-2}}\,(cut) \quad \vdots}{\begin{array}{c}\vdots \\ G \mid \varphi_1, \dots, \varphi_n, \Sigma \Rightarrow \Pi\end{array}}
$$

"(0) $\Rightarrow$ (2)": Analogous – follows by n applications of (cut) (and contraction and weakening).

"(0) $\Leftarrow$ (1)": Follows by instantiating $\Sigma = \emptyset$ and $\Pi = \psi_1, \dots, \psi_m$.

"(0) $\Leftarrow$ (2)": Follows by instantiating $\Gamma = \varphi_1, \dots, \varphi_n$. □

However, the axioms that can be handled with this procedure need to be of a specific form due to the restricted invertibility of the $(\supset, r)$ rule. Hence we define the following grammar:

Definition 29. Let $\mathbf{Ax}_I$ be the set of axioms generated by the following grammar (I is the initial variable):

$I = R_1 \vee I \mid I \vee R_1 \mid P_1$ $\qquad$ $R_1 = P_2 \mid C_2$
$P_1 = C_2 \supset C_1 \mid C_2 \supset C_2$ $\qquad$ $P_2 = C_2 \supset C_2$
$C_1 = P_2 \diamond C_2 \mid C_2 \diamond P_2$ $\qquad$ $C_2 = C_2 \diamond C_2 \mid p_i$
for $\diamond \in \{\wedge, \vee\}$, $i \in \{1, \dots, n\}$

Note that the outermost disjunction ($\vee$ in I) corresponds to the $\mid$ in hypersequents, whereas the other disjunctions ($\vee$ in C_2) correspond to the logical connective $\vee$.

For example, the bd_2 axiom, the law of excluded middle axiom *em* or the axiom for prelinearity *prel* (see Table 5.2) are in $\mathbf{Ax}_I$. The following algorithm now shows that every axiom within $\mathbf{Ax}_I$ can be transformed into a set of equivalent logical (or structural) hypersequent rules.

Theorem 7. *Every axiom* $\alpha \in \mathbf{Ax}_I$ *can be transformed into a set of equivalent logical (or structural) hypersequent rules.*

Proof. Let $\alpha \in \mathbf{Ax}_I$. α is hence of the form $\alpha_1 \vee \dots \vee \alpha_n$ where one α_j is generated by P_1 and all other $\alpha_1, \dots, \alpha_{j-1}, \alpha_{j+1}, \dots \alpha_n$ are generated by R_1 in the grammar of Definition 29. Hence, α is equivalent to a hypersequent of the form

$$G \mid \Rightarrow \alpha_1 \mid \dots \mid \Rightarrow \alpha_n$$

W.l.o.g. we assume that $\alpha_n = \alpha_j$ is generated by P_1.

Step (i): By the invertibility of the rules $(\wedge, l)$, $(\wedge, r)$, $(\vee, l)$, $(\vee, r)$, $(\bot, r)$ and (when there is no context present) $(\supset, r)$, we obtain rules of the form

$$\frac{}{G \mid \alpha_{1_{1_1}}, \ldots, \alpha_{1_{k_1}} \Rightarrow \alpha'_{1_{l_1}}, \ldots, \alpha'_{1_{m_1}} \mid \cdots \mid \alpha_{n_{1_n}}, \ldots, \alpha_{n_{k_n}} \Rightarrow \alpha'_{n_{l_n}}, \ldots, \alpha'_{n_{m_n}}, \alpha^*} \; (1)$$

where each $\alpha_{i_{j_i}}, i = 1 \ldots n$ does not contain any logical connective and α^* is of the form $\varphi \supset \psi$. Note that α^* is not necessarily present in the hypersequent; in that case we obtain rules of the form

$$\frac{}{G \mid \alpha_{1_{1_1}}, \ldots, \alpha_{1_{k_1}} \Rightarrow \alpha'_{1_{l_1}}, \ldots, \alpha'_{1_{m_1}} \mid \cdots \mid \alpha_{n_{1_n}}, \ldots, \alpha_{n_{k_n}} \Rightarrow \alpha'_{n_{l_n}}, \ldots, \alpha'_{n_{m_n}}} \; (1')$$

Step (ii): Next, we replace each component

$$\alpha_{i_{1_i}}, \ldots, \alpha_{i_{k_i}} \Rightarrow \alpha'_{i_{l_i}}, \ldots, \alpha'_{i_{m_i}}$$

in its conclusion with

$$\Gamma_i, \Sigma_i \Rightarrow \Delta_i$$

and, if present, replace the component

$$\alpha_{n_{1_n}}, \ldots, \alpha_{n_{k_n}} \Rightarrow \alpha'_{n_{l_n}}, \ldots, \alpha'_{n_{m_n}}, \alpha^*$$

in its conclusion with

$$\Gamma_n, \Sigma_n \Rightarrow \Delta_n, \alpha^*$$

We add i_{m_i} new premises $(G \mid \Gamma_i \Rightarrow \alpha_{i_{1_1}}), \ldots, (G \mid \Gamma_i \Rightarrow \alpha_{i_{k_i}}), (G \mid \alpha'_{i_{l_i}}, \Sigma_i \Rightarrow \Delta_i), \ldots, (G \mid \alpha'_{i_{m_i}}, \Sigma_i \Rightarrow \Delta_i)$ for $i = 1, \ldots, n$, thus obtaining the rule

$$\frac{G \mid \Gamma_1 \Rightarrow \alpha_{1_{1_1}} \quad \cdots \quad G \mid \Sigma_n, \alpha'_{n_{m_n}} \Rightarrow \Delta_n}{G \mid \cdots \mid \Gamma_1, \Sigma_1 \Rightarrow \Delta_1 \mid \cdots \mid \Gamma_n, \Sigma_n \Rightarrow \Delta_n, \alpha^*} \; (2)$$

If α^* is not present in any component of the hypersequent, we obtain a rule

$$\frac{G \mid \Gamma_1 \Rightarrow \alpha_{1_{1_1}} \quad \cdots \quad G \mid \Sigma_n, \alpha'_{n_{m_n}} \Rightarrow \Delta_n}{G \mid \cdots \mid \Gamma_1, \Sigma_1 \Rightarrow \Delta_1 \mid \cdots \mid \Gamma_n, \Sigma_n \Rightarrow \Delta_n} \; (2')$$

The equivalence of the rule (2) (or (2′), resp.) with rule (1) (or (1′), resp.) is ensured by Lemma 4. Note that after this step we only have metavariables $\Gamma_i, \Sigma_i, \Delta_i$ and at most one occurrence of α^* containing logical connectives in the conclusion of the rule r.

Step (iii): Let $\mathcal{S}$ be the set of premises of rule (2), resp. (2′). Now we eliminate from $\mathcal{S}$ all metavariables β not occurring in the conclusion of r (note that we do not eliminate metavariables from $\mathcal{S}$ that are subformulas of the logical formula occurring in the conclusion of r). Let $\mathcal{S}_S = \{G \mid \Gamma'_i \Rightarrow \beta : 0 \leq i \leq k\}$ ($\mathcal{S}_A = \{G \mid \Sigma'_j, \beta \Rightarrow \Delta'_j : 0 \leq j \leq m\}$, resp.) be the subset of premises which have at least one occurrence of β in the succedent (antecedent, resp.). If $\mathcal{S}_S = \emptyset$, we remove $\mathcal{S}_A$ from $\mathcal{S}$. Similar, if $\mathcal{S}_A = \emptyset$, we remove $\mathcal{S}_S$. Furthermore, let $\mathcal{S}_{cut}$ be the set of all hypersequents of the form

$G \mid \Sigma_j, \Gamma_i \Rightarrow \Delta_j$ where $1 \leq j \leq m$ and $1 \leq i \leq k$ and $G \mid \gamma, \Sigma \Rightarrow \Delta$ (or $G \mid \Gamma \Rightarrow \gamma$) where γ is a subformula of the logical formula in the conclusion of r. Let r' be the rule obtained by replacing $\mathcal{S}_A \cup \mathcal{S}_S$ with $\mathcal{S}_{cut}$. The number of variables not occurring in the conclusion decreases by one. We show that r' is equivalent to r.

"$\Rightarrow$": Follows by using cut on the premises.

"$\Leftarrow$": Let $\beta' = \bigvee_{i=1}^{k} \Gamma_i$ and instantiate all β with β'. Then all the premises $G \mid \Gamma_i \Rightarrow \beta'$ are provable. $G \mid \beta', \Sigma_j \Rightarrow \Delta_j$ is derivable using the $(\vee, l)$ rule. By applying r, we get the conclusion of r'.

Note that the resulting rules are *logical* if α^* occurs in a component of the hypersequent and they are *structural*, otherwise. □

This algorithm produces logical (or structural) rules for axioms within $\mathbf{Ax}_I$. For example, it transforms the bd_2 axiom into a logical hypersequent rule, and the axioms *em* or *prel* into structural hypersequent rules. However, these newly generated rules do not necessarily preserve cut elimination when they are added to *HLJm*. Therefore, we need to add a heuristic step to our procedure as was done for the calculi extended with logical rules in [62]. This heuristic step was in particular inspired by the work in [155], where a cut free sequent calculus for the modal logic **K4** was introduced.

(Step 1) For any axiom $\alpha \in \mathbf{Ax}_I$, we first apply the algorithm in Theorem 7 to obtain a logical rule r equivalent to α.

(Step 2) If *HLJm* extended with r is a cut-free system, we are done. Otherwise, we search for a concrete counterexample, which is a hypersequent only derivable using cuts, by inspecting the case where cut elimination fails.

(Step 3) If we have a counterexample and its derivation with applications of cut, we see if we can make the original rule stronger by cutting over the premises, and go to (Step 2) (else, we get stuck).

An Example: a Cut-free Hypersequent Calculus for $\mathbf{Bd}_2$

Among the seven interpolable propositional intermediate logics, $\mathbf{Bd}_2$ was the only one still lacking a cut-free hypersequent calculus. $\mathbf{Bd}_2$ is obtained by extending **Int** with bd_2.

We show how to construct a hypersequent calculus for $\mathbf{Bd}_2$ according to three steps explained above.

Step 1: The algorithm in Theorem 7 works as follows:

$$\Rightarrow \xi \vee \xi \supset (\varphi \vee (\varphi \supset \psi))$$

$$\longrightarrow \quad G \mid\Rightarrow \xi \mid\Rightarrow \xi \supset (\varphi \vee (\varphi \supset \psi))$$

$$\longrightarrow^{(i)} \quad \frac{}{G \mid\Rightarrow \xi \mid \xi \Rightarrow \varphi \vee (\varphi \supset \psi)}$$

$$\longrightarrow^{(i)} \quad \frac{}{G \mid\Rightarrow \xi \mid \xi \Rightarrow \varphi, \varphi \supset \psi}$$

$$\longrightarrow^{(ii)} \qquad \frac{G \mid \xi, \Sigma' \Rightarrow \Delta' \qquad G \mid \Gamma \Rightarrow \xi \qquad G \mid \varphi, \Sigma \Rightarrow \Delta}{G \mid \Sigma' \Rightarrow \Delta' \mid \Gamma, \Sigma \Rightarrow \Delta, \varphi \supset \psi}$$

$$\longrightarrow^{(iii)} \qquad \frac{G \mid \Gamma, \Sigma' \Rightarrow \Delta' \qquad G \mid \varphi, \Sigma \Rightarrow \Delta}{G \mid \Sigma' \Rightarrow \Delta' \mid \Gamma, \Sigma \Rightarrow \Delta, \varphi \supset \psi}\ (bd_2)''$$

We can simplify the rule as follows (by contraction and weakening):

$$\frac{G \mid \Gamma, \Sigma' \Rightarrow \Delta' \qquad G \mid \varphi, \Gamma \Rightarrow \Delta}{G \mid \Sigma' \Rightarrow \Delta' \mid \Gamma \Rightarrow \Delta, \varphi \supset \psi}\ (bd_2)'$$

However $(bd_2)'$ *does not* preserve cut elimination when added to *HLJm*.

Step 2: A concrete counterexample is

$$\Rightarrow \alpha \mid \alpha \Rightarrow \beta, \alpha \supset ((\alpha \supset \beta) \supset \delta)$$

which can be proved with a cut,

$$\dfrac{\dfrac{\alpha \Rightarrow \alpha \qquad \dfrac{\beta \Rightarrow \beta}{\alpha, \beta \Rightarrow \beta}}{\Rightarrow \alpha \mid \alpha \Rightarrow \beta, \neg\beta}\ (bd_2)' \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\alpha \Rightarrow \alpha}{\alpha \Rightarrow \delta, \beta, \alpha} \qquad \dfrac{\beta \Rightarrow \beta}{\beta, \alpha \Rightarrow \delta, \beta}}{\alpha, \alpha \supset \beta \Rightarrow \delta, \beta}\ (\supset, l) \qquad \dfrac{\bot \Rightarrow}{\bot, \alpha, \alpha \supset \beta \Rightarrow \delta}}{\neg\beta, \alpha, \alpha \supset \beta \Rightarrow \delta}\ (\supset, l)}{\neg\beta, \alpha \Rightarrow (\alpha \supset \beta) \supset \delta}\ (\supset, r)}{\neg\beta \Rightarrow \alpha \supset ((\alpha \supset \beta) \supset \delta)}\ (\supset, r)}{\Rightarrow \alpha \mid \alpha \Rightarrow \beta, \alpha \supset ((\alpha \supset \beta) \supset \delta)}\ (cut)$$

but not without it, as is shown by inspecting all possible derivations, e.g.:

$$\frac{\alpha \Rightarrow \alpha \qquad \dfrac{\lightning}{\alpha, \alpha \Rightarrow \beta}}{\Rightarrow \alpha \mid \alpha \Rightarrow \beta, \alpha \supset ((\alpha \supset \beta) \supset \delta)}\ (bd_2)'$$

Step 3: The derivation of the counterexample above can be rewritten as

$$\dfrac{\dfrac{\overset{\vdots d_1}{G \mid \Gamma, \Gamma' \Rightarrow \Delta'} \qquad \overset{\vdots d_2}{G \mid \Gamma, \beta \Rightarrow \Delta}}{G \mid \Gamma' \Rightarrow \Delta' \mid \Gamma \Rightarrow \Delta, \neg\beta}\ (bd_2)' \qquad \dfrac{\dfrac{\overset{\vdots d_3}{G \mid \varphi \Rightarrow \psi, \beta} \qquad G \mid \bot, \varphi \Rightarrow \psi}{G \mid \neg\beta, \varphi \Rightarrow \psi}\ (\supset, l)}{G \mid \neg\beta \Rightarrow \varphi \supset \psi}\ (\supset, r)}{G \mid \Gamma' \Rightarrow \Delta' \mid \Gamma \Rightarrow \Delta, \varphi \supset \psi}\ (cut)$$

We can cut over the premises in d_2 and d_3, hence obtaining (together with the premise in d_1 – the fourth premise is initial) the following rule $(bd_2)^*$:

$$\frac{G \mid \Gamma', \Gamma \Rightarrow \Delta' \qquad G \mid \Gamma, \varphi \Rightarrow \psi, \Delta}{G \mid \Gamma' \Rightarrow \Delta' \mid \Gamma \Rightarrow \varphi \supset \psi, \Delta}\ (bd_2)^*$$

The hypersequent calculus HBd_2 is then obtained by extending $HLJm$ with $(bd_2)^*$. We present ad-hoc proofs of soundness, completeness and cut elimination for HBd_2. We start by showing that HBd_2 is sound and complete for the logic $\mathbf{Bd}_2$.

Theorem 8 (Soundness and Completeness)**.** *For any hypersequent G*

$$\vdash_{HBd_2} G \quad \textit{iff} \quad \vdash_{LJ+\{\Rightarrow bd_2\}} \mathcal{I}(G)$$

Proof. "$\Rightarrow$": For any hypersequent G, we show that if $\vdash_{HBd_2} G$ then $\vdash_{LJ+\{\Rightarrow bd_2\}} \mathcal{I}(G)$ (recall Definition 13). By induction on the height (see Definition 9) of a derivation of G. The base case (G is an axiom) is easy. For the inductive case it suffices to see that for each inference rule in HBd_2 with premise(s) G_1 (and G_2), the sequent $\mathcal{I}(G_1) \Rightarrow \mathcal{I}(G)$ ($\mathcal{I}(G_1), \mathcal{I}(G_2) \Rightarrow \mathcal{I}(G)$) is derivable in $LJ+\{\Rightarrow bd_2\}$. The only non-trivial case to show is $(bd_2)^*$:

$$\vdash_{LJ+\{\Rightarrow bd_2\}} \mathcal{I}(G \mid \Gamma', \Gamma \Rightarrow \Delta'), \mathcal{I}(G \mid \Gamma, \varphi \Rightarrow \psi, \Delta) \Rightarrow \mathcal{I}(G \mid \Gamma' \Rightarrow \Delta' \mid \Gamma \Rightarrow \varphi \supset \psi, \Delta)$$

which follows by a (cut) with the axiom bd_2, i.e., $\Rightarrow \bigwedge \Gamma \vee (\bigwedge \Gamma \supset (\varphi \vee (\varphi \supset \psi)))$. Note that for simplicity we omit the context G and write Γ, Γ' (Δ, Δ') instead of $\bigwedge \Gamma, \bigwedge \Gamma'$ $(\bigvee \Delta, \bigvee \Delta')$ in the following derivation:

$$\dfrac{\dfrac{\dfrac{\varphi \Rightarrow \varphi \qquad \Gamma \Rightarrow \Gamma}{\varphi, \Gamma \Rightarrow \Gamma \wedge \varphi}(\wedge, r) \qquad \dfrac{\dfrac{\dfrac{\psi, \varphi \Rightarrow \psi}{\psi \Rightarrow \varphi \supset \psi}(\supset, r)}{\psi \Rightarrow (\varphi \supset \psi) \vee \Delta}(\vee, r) \qquad \dfrac{\Delta \Rightarrow \Delta}{\Delta \Rightarrow (\varphi \supset \psi) \vee \Delta}(\vee, r)}{\psi \vee \Delta \Rightarrow (\varphi \supset \psi) \vee \Delta}(\vee, l)}{\Gamma \wedge \varphi \supset \psi \vee \Delta, \varphi, \Gamma \Rightarrow (\varphi \supset \psi) \vee \Delta}(\supset, l) \qquad \dfrac{\dfrac{\psi, \varphi \Rightarrow \psi}{\psi \Rightarrow \varphi \supset \psi}(\supset, r)}{\psi \Rightarrow (\varphi \supset \psi) \vee \Delta}(\vee, r)}{(**)\ \Gamma \wedge \varphi \supset \psi \vee \Delta, \varphi \supset \psi, \Gamma \Rightarrow (\varphi \supset \psi) \vee \Delta}(\supset, l)$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Gamma \Rightarrow \Gamma}{\Gamma \wedge \varphi \supset \psi \vee \Delta, \Gamma \Rightarrow \Gamma}(w, l) \qquad \dfrac{\dfrac{\dfrac{\varphi \Rightarrow \varphi \qquad \Gamma \Rightarrow \Gamma}{\varphi, \Gamma \Rightarrow \Gamma \wedge \varphi}(\wedge, r) \qquad \dfrac{\dfrac{\dfrac{\psi, \varphi, \Rightarrow \psi}{\psi, \varphi, \Rightarrow \varphi \supset \psi}(\supset, r)}{\psi, \varphi, \Rightarrow (\varphi \supset \psi) \vee \Delta}(\vee, r) \qquad \dfrac{\Delta \Rightarrow \Delta}{\Delta, \varphi \Rightarrow (\varphi \supset \psi) \vee \Delta}(\vee, r)}{\psi \vee \Delta, \varphi \Rightarrow (\varphi \supset \psi) \vee \Delta}(\vee, l)}{\Gamma \wedge \varphi \supset \psi \vee \Delta, \varphi, \Gamma \Rightarrow (\varphi \supset \psi) \vee \Delta}(\supset, l) \qquad \vdots(**)}{\Gamma \wedge \varphi \supset \psi \vee \Delta, \varphi \vee (\varphi \supset \psi), \Gamma \Rightarrow (\varphi \supset \psi) \vee \Delta}(\vee, l)}{\Gamma \wedge \varphi \supset \psi \vee \Delta, \Gamma \supset (\varphi \vee (\varphi \supset \psi)), \Gamma \Rightarrow (\varphi \supset \psi) \vee \Delta}(\supset, l)}{\Gamma \wedge \varphi \supset \psi \vee \Delta, \Gamma \supset (\varphi \vee (\varphi \supset \psi)) \Rightarrow \Gamma \supset (\varphi \supset \psi) \vee \Delta}(\supset, r)}{\Gamma \wedge \varphi \supset \psi \vee \Delta, \Gamma \supset (\varphi \vee (\varphi \supset \psi)) \Rightarrow (\Gamma' \supset \Delta') \vee (\Gamma \supset (\varphi \supset \psi) \vee \Delta)}(\vee, r)}{(*)\ \Gamma' \wedge \Gamma \supset \Delta', \Gamma \wedge \varphi \supset \psi \vee \Delta, \Gamma \supset (\varphi \vee (\varphi \supset \psi)) \Rightarrow (\Gamma' \supset \Delta') \vee (\Gamma \supset (\varphi \supset \psi) \vee \Delta)}(w, l)$$

$$\dfrac{\Rightarrow \Gamma \vee (\Gamma \supset (\varphi \vee (\varphi \supset \psi))) \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Gamma \Rightarrow \Gamma \qquad \Gamma' \Rightarrow \Gamma'}{\Gamma, \Gamma' \Rightarrow \Gamma' \wedge \Gamma}(\wedge, r) \qquad \Delta' \Rightarrow \Delta'}{\Gamma' \wedge \Gamma \supset \Delta', \Gamma, \Gamma' \Rightarrow \Delta'}(\supset, l)}{\Gamma' \wedge \Gamma \supset \Delta', \Gamma \Rightarrow \Gamma' \supset \Delta'}(\supset, r)}{\Gamma' \wedge \Gamma \supset \Delta', \Gamma \Rightarrow (\Gamma' \supset \Delta') \vee (\Gamma \supset (\varphi \supset \psi) \vee \Delta)}(\vee, r)}{\Gamma' \wedge \Gamma \supset \Delta', \Gamma \wedge \varphi \supset \psi \vee \Delta, \Gamma \Rightarrow (\Gamma' \supset \Delta') \vee (\Gamma \supset (\varphi \supset \psi) \vee \Delta)}(w, l) \qquad \vdots(*)}{\Gamma' \wedge \Gamma \supset \Delta', \Gamma \wedge \varphi \supset \psi \vee \Delta, \Gamma \vee (\Gamma \supset (\varphi \vee (\varphi \supset \psi))) \Rightarrow (\Gamma' \supset \Delta') \vee (\Gamma \supset (\varphi \supset \psi) \vee \Delta)}(\vee, l)}{\Gamma' \wedge \Gamma \supset \Delta', \Gamma \wedge \varphi \supset \psi \vee \Delta \Rightarrow (\Gamma' \supset \Delta') \vee (\Gamma \supset (\varphi \supset \psi) \vee \Delta)}(cut)$$

"⇐": We show that if $\vdash_{LJ+\{\Rightarrow bd_2\}} \mathcal{I}(G)$ then $\vdash_{HBd_2} G$. All rules of LJ are derivable in HBd_2. Hence it only remains to show a proof of the axiom bd_2 using $(bd_2)^*$:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\varphi \Rightarrow \varphi \qquad \varphi, \psi \Rightarrow \psi, \xi}{\Rightarrow \varphi \mid \varphi \Rightarrow \psi, \psi \supset \xi}\,(bd_2)^*}{\Rightarrow \varphi \mid \varphi \Rightarrow \psi \vee (\psi \supset \xi)}\,(\vee, r)}{\Rightarrow \varphi \mid \Rightarrow \varphi, \varphi \supset (\psi \vee (\psi \supset \xi))}\,(\supset, r)}{\Rightarrow \varphi \vee (\varphi \supset (\psi \vee (\psi \supset \xi))) \mid \Rightarrow \varphi \vee (\varphi \supset (\psi \vee (\psi \supset \xi)))}\,(\vee, r), (w, r)}{\Rightarrow \varphi \vee (\varphi \supset (\psi \vee (\psi \supset \xi)))}\,(ec)$$

□

Cut elimination for *HBd*$_2$

Recall the general cut elimination theorem for hypersequent calculi that contain reductive and substitutive rules (see Theorem 2). This theorem does not apply to HBd_2 (and $HLJm$) because not all logical rules of the calculus are substitutive (see Definition 17, extended to the multiple-conclusion case).

Lemma 5. *The rule*

$$\dfrac{G \mid \Gamma, \varphi \Rightarrow \psi}{G \mid \Gamma \Rightarrow \varphi \supset \psi, \Delta}\,(\supset, r)$$

of the calculus HLJm is not substitutive.

Proof. Assume that $(\supset, r)$ is substitutive, i.e., for any instance $\frac{G_1}{G}$ of $(\supset, r)$, any multiple-conclusioned hypersequent H and $G' \in CUT(G, H)$, there exists $G'_1 \in CUT(G_1, H)$ such that $\frac{G'_1}{G'}$ is an instance of $(\supset, r)$. Consider the following instance of the rule:

$$\dfrac{\alpha, \varphi \Rightarrow \psi}{\alpha \Rightarrow \varphi \supset \psi}\,(\supset, r)$$

and let $H = \gamma \Rightarrow \alpha, \delta$. Then $G' = \gamma \Rightarrow \varphi \supset \psi, \delta$ and $G'_1 = \gamma, \varphi \Rightarrow \psi, \delta$, obtaining:

$$\dfrac{\gamma, \varphi \Rightarrow \psi, \delta}{\gamma \Rightarrow \varphi \supset \psi, \delta}$$

This is however clearly not an instance of $(\supset, r)$ and $(\supset, r)$ is thus not substitutive. □

The cut elimination proof for the calculus HBd_2 therefore requires another strategy. Consider for example the following application of cut with cut formula α:

$$\dfrac{\begin{array}{c}\vdots d_r \\ G \mid \Gamma \Rightarrow \Delta, \alpha\end{array} \qquad \begin{array}{c}\vdots d_l \\ H \mid \alpha, \Sigma \Rightarrow \Pi\end{array}}{G \mid H \mid \Gamma, \Sigma \Rightarrow \Delta, \Pi}\,(cut)$$

We call d_l (d_r, resp.) the derivation ending in the premise of cut with the cut formula on the left (right, resp.) side of the sequent. Our cut elimination strategy distinguishes by cases according to the cut formula:

For non-atomic cut formulas having $\wedge$ and $\vee$ as outermost connective, we use the invertibility of their respective logical rules to replace the cut by smaller ones.

Cut formulas having $\supset$ as outermost connective and atomic cut formulas require a different handling. In this case we proceed by shifting the cut upwards in a specific order. First we move the cut upwards in the derivation d_r in which the cut formula is on the right side of the sequents (Lemma 8). If the cut formula is atomic, we can move the cut upward and then finally remove it.

When the cut formula is introduced by a weakening or in the context of an application of $(\supset, r)$, the cut can be removed, e.g.:

$$\dfrac{\dfrac{\overset{\vdots d_r'}{G \mid \Gamma, \varphi \Rightarrow \psi}}{G \mid \Gamma \Rightarrow \varphi \supset \psi, \alpha}(\supset, r) \qquad \overset{\vdots d_l}{H \mid \alpha, \Sigma \Rightarrow \Pi}}{G \mid H \mid \Gamma, \Sigma \Rightarrow \varphi \supset \psi, \Pi} \quad \longrightarrow \quad \dfrac{\dfrac{\overset{\vdots d_r'}{G \mid \Gamma, \varphi \Rightarrow \psi}}{G \mid \Gamma \Rightarrow \varphi \supset \psi}(\supset, r)}{\overline{\overline{G \mid H \mid \Gamma, \Sigma \Rightarrow \varphi \supset \psi, \Pi}}}(ew), (w, l), (w, r)$$

If the cut formula is introduced by $(\supset, r)$ or $(bd_2)^*$ we proceed by shifting the cut upwards in the left derivation d_l until the cut formula is introduced on the left side of the sequent. Finally, we cut the premises of the two last applied rules to replace the cut by smaller ones (Lemma 7). However, moving the cut upwards can be problematic in presence of $(\supset, r)$ or $(bd_2)^*$ in d_l. E.g., in the following situation:

$$\dfrac{\overset{\vdots d_r}{G \mid \Gamma' \Rightarrow \Delta' \mid \Gamma \Rightarrow \alpha \supset \beta, \Delta} \qquad \dfrac{\overset{\vdots d_l}{H \mid \Sigma, \alpha \supset \beta, \varphi \Rightarrow \psi}}{H \mid \Sigma, \alpha \supset \beta \Rightarrow \varphi \supset \psi, \Pi}(\supset, r)}{G \mid H \mid \Gamma' \Rightarrow \Delta' \mid \Gamma, \Sigma \Rightarrow \Delta, \varphi \supset \psi, \Pi}(cut)$$

The problem is the presence of the context Δ that does not permit the subsequent application of $(\supset, r)$ to the following derivation

$$\dfrac{\overset{\vdots d_r}{G \mid \Gamma' \Rightarrow \Delta' \mid \Gamma \Rightarrow \alpha \supset \beta, \Delta} \qquad \overset{\vdots d_l}{H \mid \Sigma, \alpha \supset \beta, \varphi \Rightarrow \psi}}{G \mid \Gamma' \Rightarrow \Delta' \mid H \mid \Gamma, \Sigma, \varphi \Rightarrow \Delta, \psi}(cut)$$

We show that it is always possible to shift the cut upwards over d_l when the right premise ends in an introduction rule of the cut formula, i.e. $(\supset, r)$ or $(bd_2)^*$. For instance, assume in the above case that the cut formula is introduced by $(bd_2)^*$:

$$\dfrac{\dfrac{\overset{\vdots d_r'}{G \mid \Gamma', \Gamma \Rightarrow \Delta'} \qquad \overset{\vdots d_r''}{G \mid \Gamma, \alpha \Rightarrow \beta, \Delta}}{G \mid \Gamma' \Rightarrow \Delta' \mid \Gamma \Rightarrow \alpha \supset \beta, \Delta}(bd_2)^* \qquad \overset{\vdots d_l}{H \mid \Sigma, \alpha \supset \beta, \varphi \Rightarrow \psi}}{G \mid \Gamma' \Rightarrow \Delta' \mid H \mid \Gamma, \Sigma, \varphi \Rightarrow \Delta, \psi}(cut)$$

The original cut above is then shifted upwards as follows (we omit the contexts G and H for simplicity):

$$\dfrac{\dfrac{\dfrac{\dfrac{\vdots d_r'}{\Gamma', \Gamma \Rightarrow \Delta'}}{\Gamma' \Rightarrow \Delta' \mid \Gamma', \Gamma, \Sigma \Rightarrow \Delta'}\,(w,l),(ew) \qquad \dfrac{\dfrac{\vdots d_r}{\Gamma' \Rightarrow \Delta' \mid \Gamma \Rightarrow \alpha \supset \beta, \Delta} \quad \dfrac{\vdots d_l}{\Sigma, \alpha \supset \beta, \varphi \Rightarrow \psi}}{\Gamma' \Rightarrow \Delta' \mid \Gamma, \Sigma, \varphi \Rightarrow \Delta, \psi}\,(cut)}{\Gamma' \Rightarrow \Delta' \mid \Gamma' \Rightarrow \Delta' \mid \Gamma, \Sigma \Rightarrow \Delta, \varphi \supset \psi}\,(bd_2)^*}{\Gamma' \Rightarrow \Delta' \mid \Gamma, \Sigma \Rightarrow \Delta, \varphi \supset \psi}\,(ec)$$

To formalize this cut elimination proof, we need the following definition of *cut grade*:

Definition 30. The *cut grade* $\rho(d)$ of a derivation d is the maximal complexity of cut formulas in $d+1$ ($\rho(d) = 0$ if d is cut free).

Lemma 6 (Inversion)**.**

(i) *If* $d \vdash_{HBd_2} G \mid \Gamma, \varphi \vee \psi \Rightarrow \Delta$ *then one can find* $d_1 \vdash_{HBd_2} G \mid \Gamma, \varphi \Rightarrow \Delta$ *and* $d_2 \vdash_{HBd_2} G \mid \Gamma, \psi \Rightarrow \Delta$.

(ii) *If* $d \vdash_{HBd_2} G \mid \Gamma \Rightarrow \varphi \vee \psi, \Delta$ *then one can find* $d_1 \vdash_{HBd_2} G \mid \Gamma \Rightarrow \varphi, \psi, \Delta$.

(iii) *If* $d \vdash_{HBd_2} G \mid \Gamma, \varphi \wedge \psi \Rightarrow \Delta$ *then one can find* $d_1 \vdash_{HBd_2} G \mid \Gamma, \varphi, \psi \Rightarrow \Delta$.

(iv) *If* $d \vdash_{HBd_2} G \mid \Gamma \Rightarrow \varphi \wedge \psi, \Delta$ *then one can find* $d_1 \vdash_{HBd_2} G \mid \Gamma \Rightarrow \varphi, \Delta$ *and* $d_2 \vdash_{HBd_2} G \mid \Gamma \Rightarrow \psi, \Delta$.

(v) *If* $d \vdash_{HBd_2} G \mid \Gamma \Rightarrow \varphi \supset \psi$ *then one can find* $d_1 \vdash_{HBd_2} G \mid \Gamma, \varphi \Rightarrow \psi$.

(vi) *If* $d \vdash_{HBd_2} G \mid \Gamma \Rightarrow \bot, \Delta$ *then one can find* $d_1 \vdash_{HBd_2} G \mid \Gamma \Rightarrow \Delta$.

such that $\rho(d_i) \leq \rho(d)$ *and* $|d_i| \leq |d|$ *for* $i = 1, 2$.

Proof. (ii) To deal with internal and external contraction, we have to prove a more general statement, namely:
(ii)' If $d \vdash_{HBd_2} G \mid \Gamma_1 \Rightarrow (\varphi \vee \psi)^{n_1}, \Delta_1 \mid \cdots \mid \Gamma_k, \Rightarrow (\varphi \vee \psi)^{n_k}, \Delta_k$ then one can find $d_1 \vdash_{HBd_2} G \mid \Gamma_1 \Rightarrow (\varphi, \psi)^{n_1}, \Delta_1 \mid \cdots \mid \Gamma_k \Rightarrow (\varphi, \psi)^{n_k}, \Delta_k$. We assume $n_j > 0$ for some $j, 1 \leq j \leq k$. By induction on $|d|$. Consider the last inference rule r in d:

- r is $(\vee, r)$ and some $\varphi \vee \psi$ is the principal formula. Then we get $d' \vdash_{HBd_2} G \mid \Gamma_1 \Rightarrow \Delta_1, \varphi, \psi, (\varphi \vee \psi)^{n_1 - 1} \mid \cdots \mid \Gamma_k \Rightarrow \Delta_k, (\varphi \vee \psi)^{n_k}$. The claim follows by application of the inductive hypothesis.
- r is any other rule. The claim follows by application of the inductive hypothesis and subsequent application(s) of r.

(i) and *(iii)–(vi)* can be shown analogously. Note that in the case of (v), we only have invertibility when there is no context present in the succedent. □

Lemma 7 (Shift Left and Reduction of $\supset$)**.** *Let* d_l *and* d_r *be derivations in* HBd_2 *such that:*

- d_l *is a derivation of* $H \mid \Sigma_1, (\alpha \supset \beta)^{n_1} \Rightarrow \Pi_1 \mid \cdots \mid \Sigma_k, (\alpha \supset \beta)^{n_k} \Rightarrow \Pi_k$,

- *d_r is a derivation of $G \mid \Gamma \Rightarrow \alpha \supset \beta, \Delta$,*
- *$\rho(d_l) \leq |\alpha \supset \beta|$ and $\rho(d_r) \leq |\alpha \supset \beta|$,*
- *d_r ends with an application of $(\supset, r)$ or $(bd_2)^*$ introducing $\alpha \supset \beta$.*

Then we can find a derivation d of $G \mid H \mid \Gamma^{n_1}, \Sigma_1 \Rightarrow \Delta^{n_1}, \Pi_1 \mid \cdots \mid \Gamma^{n_k}, \Sigma_k \Rightarrow \Delta^{n_k}, \Pi_k$ in HBd$_2$ with $\rho(d) \leq |\alpha \supset \beta|$.

Proof. By induction on $|d_l|$. If d_l ends in an axiom, we are done. Otherwise, consider the last inference rule r applied in d_l.

- Suppose that r acts only on H, or r is any rule other than $(\supset, l)$, $(\supset, r)$ or $(bd_2)^*$ introducing $\alpha \supset \beta$. Then the claim follows by applications of the inductive hypothesis, r and, if needed, weakening and contraction.
- When $r = (\supset, l)$ and $\alpha \supset \beta$ is the principal formula the claim follows by applying the inductive hypothesis and subsequent cuts with cut formulas α and β.
- When d_r ends in an application of $(\supset, r)$, the required derivation is simply obtained by applying the inductive hypothesis and r (note that in this case Δ can be taken to be empty and hence no context is added to the premises by the inductive hypothesis).
- If d_r ends with $(bd_2)^*$ and $r = (\supset, r)$ the case is handled as described previously. Assume that d_r ends with $(bd_2)^*$ and $r = (bd_2)^*$ as in the following derivation (we omit the contexts for simplicity):

$$\dfrac{\dfrac{\dfrac{\vdots\, d_r'}{\Gamma', \Gamma \Rightarrow \Delta'} \quad \dfrac{\vdots\, d_r''}{\Gamma, \alpha \Rightarrow \beta, \Delta}}{\Gamma' \Rightarrow \Delta' \mid \Gamma \Rightarrow \alpha \supset \beta, \Delta}(bd_2)^* \quad \dfrac{\dfrac{\vdots\, d_l'}{\Sigma', \Sigma_1, (\alpha \supset \beta)^{n_1} \Rightarrow \Pi'} \quad \dfrac{\vdots\, d_l''}{\Sigma_1, (\alpha \supset \beta)^{n_1}, \varphi \Rightarrow \psi, \Pi_1}}{\Sigma' \Rightarrow \Pi' \mid \Sigma_1, (\alpha \supset \beta)^{n_1} \Rightarrow \varphi \supset \psi, \Pi_1}(bd_2)^*}{\Gamma' \Rightarrow \Delta' \mid \Sigma' \Rightarrow \Pi' \mid \Gamma^{n_1}, \Sigma_1 \Rightarrow \varphi \supset \psi, \Pi_1, \Delta^{n_1}}(cut)$$

The cut is moved upwards as follows:

$$\dfrac{\dfrac{(w,l),(w,r)\,\dfrac{\dfrac{\vdots d_r'}{\Gamma', \Gamma \Rightarrow \Delta'}}{\Gamma', \Gamma^{n_1}, \Sigma_1 \Rightarrow \Delta'} \quad \dfrac{\dfrac{\vdots d_r}{\Gamma' \Rightarrow \Delta' \mid \Gamma \Rightarrow \alpha \supset \beta, \Delta} \quad \dfrac{\vdots d_l''}{\Sigma_1, (\alpha \supset \beta)^{n_1}, \varphi \Rightarrow \psi, \Pi_1}}{\Gamma' \Rightarrow \Delta' \mid \Gamma^{n_1}, \Sigma_1, \varphi \Rightarrow \psi, \Delta^{n_1}, \Pi_1}(cut)}{\Gamma' \Rightarrow \Delta' \mid \Gamma' \Rightarrow \Delta' \mid \Gamma^{n_1}, \Sigma_1 \Rightarrow \varphi \supset \psi, \Pi_1, \Delta^{n_1}}(bd_2)^*}{\Gamma' \Rightarrow \Delta' \mid \Sigma' \Rightarrow \Pi' \mid \Gamma^{n_1}, \Sigma_1 \Rightarrow \varphi \supset \psi, \Pi_1, \Delta^{n_1}}(ew),(ec)$$

□

Lemma 8 (Shift Right)**.** *Let d_l and d_r be derivations in HBd$_2$ such that:*

- *d_l is a derivation of $H \mid \Sigma, \varphi \Rightarrow \Pi$,*
- *φ is either atomic or of the form $\alpha \supset \beta$,*
- *d_r is a derivation of $G \mid \Gamma_1 \Rightarrow \varphi^{n_1}, \Delta_1 \mid \cdots \mid \Gamma_k \Rightarrow \varphi^{n_k}, \Delta_k$,*
- *$\rho(d_l) \leq |\varphi|$ and $\rho(d_r) \leq |\varphi|$.*

Then we can find a derivation d of $G \mid H \mid \Gamma_1, \Sigma^{n_1} \Rightarrow \Delta_1, \Pi^{n_1} \mid \cdots \mid \Gamma_k, \Sigma^{n_k} \Rightarrow \Delta_k, \Pi^{n_k}$ in HBd_2 with $\rho(d) \leq |\varphi|$.

Proof. By induction on $|d_r|$. If d_r ends in an axiom, we are done. Otherwise, consider the last inference rule r in d_r.

– If r acts only on G or r is any rule other than a logical rule introducing φ then the claim follows by applications of the inductive hypothesis, r and, if needed, weakening or contraction. For example, consider the following case where r is $(\vee, r)$ (we omit the contexts for better readability):

$$\dfrac{\dfrac{\dfrac{\vdots}{\Gamma, \Rightarrow \varphi, \psi, \xi, \Delta}}{\Gamma \Rightarrow \varphi, \psi \vee \xi, \Delta}\,(\vee, r) \qquad \dfrac{\vdots}{\Sigma, \varphi \Rightarrow \Pi}}{\Gamma, \Sigma \Rightarrow \Delta, \Pi, \psi \vee \xi}\,(cut) \quad \longrightarrow \quad \dfrac{\dfrac{\dfrac{\vdots}{\Gamma \Rightarrow \varphi, \psi, \xi, \Delta} \qquad \dfrac{\vdots}{\Sigma, \varphi \Rightarrow \Pi}}{\Gamma, \Sigma \Rightarrow \Delta, \Pi, \psi, \xi}\,(cut)}{\Gamma, \Sigma \Rightarrow \Delta, \Pi, \psi \vee \xi}\,(\vee, r)$$

– If r is $(\supset, r)$ and φ is not the principal formula, the claim follows by several applications of weakening, see e.g. the following example (again, we omit the contexts):

$$\dfrac{\dfrac{\dfrac{\vdots}{\Gamma, \psi \Rightarrow \xi}}{\Gamma \Rightarrow \varphi, \psi \supset \xi, \Delta}\,(\supset, r) \qquad \dfrac{\vdots}{\Sigma, \varphi \Rightarrow \Pi}}{\Gamma, \Sigma \Rightarrow \Delta, \Pi, \psi \supset \xi}\,(cut) \quad \longrightarrow \quad \dfrac{\dfrac{\dfrac{\vdots}{\Gamma, \psi \Rightarrow \xi}}{\Gamma \Rightarrow \psi \supset \xi}\,(\supset, r)}{\Gamma, \Sigma \Rightarrow \Delta, \Pi, \psi \supset \xi}\,(w, l) + (w, r)$$

– If r is $(\supset, r)$ or $(bd_2)^*$ and φ is the principal formula, the claim follows by applications of the inductive hypothesis, the corresponding rule r and Lemma 7.

□

Theorem 9 (Cut elimination)**.** *Cut elimination holds for HBd_2.*

Proof. Let d be a derivation in HBd_2 with $\rho(d) > 0$. The proof proceeds by a double induction on $\langle \rho(d), \#\rho(d) \rangle$, where $\#\rho(d)$ is the number of applications of (cut) in d with cut formulas of complexity $\rho(d)$. Consider an uppermost application of (cut) in d with a cut formula of complexity $\rho(d)$. Let d_l and d_r be its premises, where d_l is a derivation of $H \mid \Sigma, \varphi \Rightarrow \Pi$, and d_r is a derivation of $G \mid \Gamma \Rightarrow \varphi, \Delta$:

$$\dfrac{\overset{\vdots\, d_r}{G \mid \Gamma \Rightarrow \varphi, \Delta} \qquad \overset{\vdots\, d_l}{H \mid \Sigma, \varphi \Rightarrow \Pi}}{G \mid H \mid \Gamma, \Sigma \Rightarrow \Delta, \Pi}\,(cut)$$

We can find a proof of $G \mid H \mid \Gamma, \Sigma \Rightarrow \Delta, \Pi$ in which either $\rho(d)$ decreases, or $\rho(d)$ stays the same and $\#\rho(d)$ decreases. Indeed we distinguish the following cases according to the main connective of φ:

- φ is an atomic formula or $\varphi = \alpha \supset \beta$. The claim follows by Lemma 8.

- Suppose $\varphi = \alpha \vee \beta$. By Lemma 6,we can find the derivations $d'_r \vdash G \mid \Gamma \Rightarrow \alpha, \beta, \Delta$, as well as $d'_l \vdash H \mid \alpha, \Sigma \Rightarrow \Pi$ and $d''_l \vdash H \mid \beta, \Sigma \Rightarrow \Pi$, such that $\rho(d'_r), \rho(d'_l), \rho(d''_l) < |\varphi|$. The claim follows by replacing the cut with cut formula $\alpha \vee \beta$ with cuts on α and β.
- The case $\varphi = \alpha \wedge \beta$ is similar since $\wedge$ is also invertible on both sides (Lemma 6).

□

5.4 Towards a Systematic Procedure for Labelled Calculi

While in the previous section we have outlined a method that takes the syntactic specification of a logic as starting point, we now discuss a systematic method that starts from the semantic specification of the logic. Inspired by the procedures to generate hypersequent calculi from Hilbert axioms in [52, 62], we introduce a similar approach to generate labelled sequent calculi from frame conditions. We provide proofs for soundness, completeness and cut elimination for the generated calculi, which subsume those introduced in [76] for geometric formulas.

A Classification of Frame Conditions

In a first step, we start by classifying frame conditions, describing intermediate logics in a hierarchy similar to the substructural hierarchy in [52] (see Section 5.2). This hierarchy intuitively accounts for the difficulty to deal proof-theoretically with the corresponding formulas of first-order classical logic. Our classification is based on the invertibility of the logical/quantifier rules of the base calculus. In this case, we use the base calculus *LK'*, i.e. a variant of the Gentzen *LK* calculus for first-order classical logic, which is called *G3c* in [134] and is depicted in Table 5.4. All logical rules of *LK'* are invertible, while the universal (existential, respectively) quantifier is invertible on the right (on the left, respectively) [134].

Note that w.l.o.g. we will only consider formulas in prenex normal form[2]. The class of a formula is then determined solely by the alternation of universal and existential quantifiers in the prefix. Intuitively, any formula within a class Π_i will start with a universal quantifier while any formula within a class Σ_j will start with an existential quantifier. This leads to a classification which is essentially the arithmetical hierarchy.

Definition 31. Let A be a formula in first-order classical logic. The classes Π_k and Σ_k are defined as follows: $A \in \Sigma_0$ and $A \in \Pi_0$, if A is quantifier-free. Otherwise:

- if A is classically equivalent to $\exists \overline{x} B$ where $B \in \Pi_n$ then $A \in \Sigma_{n+1}$
- if A is classically equivalent to $\forall \overline{x} B$ where $B \in \Sigma_n$ then $A \in \Pi_{n+1}$

[2]Recall that a formula is in prenex normal form if it is of the form $\Diamond_1 x_1 \ldots \Diamond_n x_n A$ where $\Diamond_1 \ldots \Diamond_n$ are quantifiers $\forall$ or $\exists$ and the formula A is quantifier-free.

$P, \Gamma \Rightarrow \Delta, P$	$\dfrac{A[y/x], \Gamma \Rightarrow \Delta}{\exists x A, \Gamma \Rightarrow \Delta}\ (\exists, l)$	$\dfrac{\Gamma \Rightarrow \Delta, A[y/x]}{\Gamma \Rightarrow \Delta, \forall x A}\ (\forall, r)$	$\dfrac{\Gamma \Rightarrow \Delta, A \quad \Gamma \Rightarrow \Delta, B}{\Gamma \Rightarrow \Delta, A \& B}\ (\&, r)$
$\dfrac{\Gamma \Rightarrow \Delta, A}{\neg A, \Gamma \Rightarrow \Delta}\ (\neg, l)$	$\dfrac{\Gamma \Rightarrow \Delta, A[t/x]}{\Gamma \Rightarrow \Delta, \exists x A}\ (\exists, r)$	$\dfrac{A[t/x], \Gamma \Rightarrow \Delta}{\forall x A, \Gamma \Rightarrow \Delta}\ (\forall, l)$	$\dfrac{A, \Gamma \Rightarrow \Delta \quad B, \Gamma \Rightarrow \Delta}{A \vee B, \Gamma \Rightarrow \Delta}\ (\vee, l)$
$\dfrac{A, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta, \neg A}\ (\neg, r)$	$\dfrac{\Gamma \Rightarrow \Delta, A, B}{\Gamma \Rightarrow \Delta, A \vee B}\ (\vee, r)$	$\dfrac{A, B, \Gamma \Rightarrow \Delta}{A \& B, \Gamma \Rightarrow \Delta}\ (\&, l)$	$\dfrac{\Gamma \Rightarrow \Delta, A \quad B, \Gamma \Rightarrow \Delta}{A \rightarrow B, \Gamma \Rightarrow \Delta}\ (\rightarrow, l)$
$\dfrac{\Gamma \Rightarrow \Delta}{A, \Gamma \Rightarrow \Delta}\ (w, l)$	$\dfrac{A, A, \Gamma \Rightarrow \Delta}{A, \Gamma \Rightarrow \Delta}\ (c, l)$	$\dfrac{A, \Gamma \Rightarrow \Delta, B}{\Gamma \Rightarrow \Delta, A \rightarrow B}\ (\rightarrow, r)$	$\dfrac{\Gamma \Rightarrow \Delta, A \quad A, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}\ (cut)$
$\dfrac{\Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta, A}\ (w, r)$	$\dfrac{\Gamma \Rightarrow \Delta, A, A}{\Gamma \Rightarrow \Delta, A}\ (c, r)$	y in $(\exists, l)$ and $(\forall, r)$ is an eigenvariable, P is atomic	

Table 5.4: Sequent calculus *LK'* [134]

All frame conditions for intermediate propositional logics presented in Section 5.1 with the exception of **KP** are within the class Π_2. Moreover, note that all regular formulas (universal axioms, see page 63) are within Π_1, while the geometric formulas of [76, 131] (see page 63) are within Π_2. In particular, note that any formula within Π_1 has only universal quantifiers $\forall$ in the prefix, while any formula within Π_2 is of the form $\forall x_1, \ldots, x_n \exists y_1, \ldots, y_m$ for $m, n \geq 1$.

We show below how to transform all formulas within the class Π_2 into labelled rules that preserve cut elimination when added to (a slightly modified version of) the calculus *G3I*, see Table 5.1. The resulting rules are equivalent to the corresponding axioms, i.e., *LK'* extended with the defined rules and *LK'* extended with the original formula prove the same sequents.

From Frame Conditions to Labelled Rules

Recall again the key ingredients for our transformation procedure described in Section 3.1:

(1) the *invertibility* of all logical rules, and the quantifier rules $(\forall, r)$ (i.e., introduction of $\forall$ on the right) and $(\exists, l)$ (i.e., introduction of $\exists$ on the left) in *LK'* [134], and
(2) the Ackermann lemma:

Lemma 9. *The sequent* $A_1, \ldots, A_n \Rightarrow B_1, \ldots, B_m$ *is equivalent to the rule*

$$\frac{B_1, \Gamma \Rightarrow \Delta \quad \cdots \quad B_m, \Gamma \Rightarrow \Delta}{A_1, \ldots, A_n, \Gamma \Rightarrow \Delta}$$

where Γ, Δ *are fresh metavariables standing for multisets of formulas.*

Proof. "$\Rightarrow$": Follows by m applications of (cut) (and weakening), i.e.

$$
\dfrac{\dfrac{\dfrac{\dfrac{A_1,\dots,A_n \Rightarrow B_1,\dots,B_m}{A_1,\dots,A_n,\Gamma \Rightarrow \Delta, B_1,\dots,B_m}\,(w,r),(w,l) \qquad B_m,\Gamma \Rightarrow \Delta}{A_1,\dots,A_n,\Gamma \Rightarrow \Delta, B_1,\dots,B_{m-1}}\,(cut) \qquad B_{m-1},\Gamma \Rightarrow \Delta}{A_1,\dots,A_n,\Gamma \Rightarrow \Delta, B_1,\dots,B_{m-2}}\,(cut) \qquad \vdots}{\begin{array}{c}\vdots\\ A_1,\dots,A_n,\Gamma \Rightarrow \Delta\end{array}}
$$

"$\Leftarrow$": Follows by instantiating $\Gamma = \emptyset$ and $\Delta = B_1,\dots,B_m$. □

Theorem 10. *Every frame condition $\mathcal{F}$ within the class Π_2 can be transformed into a set of equivalent labelled sequent rules.*

Proof. Let $\mathcal{F} = \forall\overline{x}\exists\overline{y}A$, where A is a quantifier-free formula, $\overline{x} = x_1,\dots,x_h$ and $\overline{y} = y_1,\dots,y_l$. W.l.o.g. we assume that A is in disjunctive normal form and has the shape $B_1 \vee \dots \vee B_k$ where every B_i has the form $Q_{i_1}\&\cdots\&Q_{i_n}\&\neg P_{i_1}\&\cdots\&\neg P_{i_m}$. Recall that (possibly indexed) P, Q indicate relational atoms. By the invertibility of the rule $(\forall, r)$, $\Rightarrow \mathcal{F}$ is equivalent to $\Rightarrow \exists\overline{y}A$.

We distinguish two cases according to whether $\mathcal{F}$ contains at least one existential quantifier, i.e. $\mathcal{F} \in \Pi_2$, or it does not, i.e. $\mathcal{F} \in \Pi_1$.

- $\mathcal{F} \in \Pi_1$ ($l = 0$): By the invertibility of $(\vee, r)$, $(\&, r)$ and $(\neg, r)$, $\Rightarrow A$ is equivalent to a set of atomic sequents $\overline{P} \Rightarrow \overline{Q}$ with $\overline{P}, \overline{Q}$ multisets of relational atoms P_{i_r}, Q_{i_s}. By Lemma 9, these sequents are equivalent to rules of the form:

$$\dfrac{\overline{Q}, \Gamma \Rightarrow \Delta}{\overline{P}, \Gamma \Rightarrow \Delta}\,(\Pi_1')$$

- $\mathcal{F} \in \Pi_2$ ($l > 0$): By Lemma 9, $\Rightarrow \exists\overline{y}A$ is equivalent to $\dfrac{\exists\overline{y}A, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}$ which is in turn equivalent to $\dfrac{A', \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}$ where A' is obtained by replacing in A all $y_1,\dots,y_l$ with fresh variables $y_1',\dots,y_l'$ (eigenvariable condition). By the invertibility of $(\vee, l), (\wedge, l)$ and $(\neg, l)$ we get

$$\dfrac{\{Q_{i_1},\dots,Q_{i_n},\Gamma \Rightarrow \Delta, P_{i_1},\dots,P_{i_m}\}_{i=1\dots k}}{\Gamma \Rightarrow \Delta}\,(\Pi_2)$$

The resulting rules are equivalent to $\mathcal{F}$. □

Note that the (Π_2) rule (which is, in fact, a rule schema) is invertible. To make (Π_1') rules invertible we simply repeat $\overline{P}$ in its premises, thus obtaining

$$\dfrac{\overline{P}, \overline{Q}, \Gamma \Rightarrow \Delta}{\overline{P}, \Gamma \Rightarrow \Delta}\,(\Pi_1)$$

which is equivalent with the rule (Π_1') in *LK'*.

Proposition 1. *Let $\overline{P}, \overline{Q}$ be multisets of relational atoms P_i, Q_i. Any (Π_1') rule of the form*

$$\frac{\overline{Q}, \Gamma \Rightarrow \Delta}{\overline{P}, \Gamma \Rightarrow \Delta}\ (\Pi_1')$$

is equivalent to the rule (Π_1) of the form

$$\frac{\overline{P}, \overline{Q}, \Gamma \Rightarrow \Delta}{\overline{P}, \Gamma \Rightarrow \Delta}\ (\Pi_1)$$

Proof. "$\Rightarrow$": For the first direction, we simply use several applications of contraction and an application of (Π_1').

$$\cfrac{\cfrac{\overline{P}, \overline{Q}, \Gamma \Rightarrow \Delta}{\overline{P}, \overline{P}, \Gamma \Rightarrow \Delta}\ (\Pi_1')}{\overline{P}, \Gamma \Rightarrow \Delta}\ (c, l)$$

"$\Leftarrow$": For the other direction, we use an application of (Π_1) and several applications of weakening.

$$\cfrac{\cfrac{\overline{Q}, \Gamma \Rightarrow \Delta}{\overline{P}, \overline{Q}, \Gamma \Rightarrow \Delta}\ (w, l)}{\overline{P}, \Gamma \Rightarrow \Delta}\ (\Pi_1)$$

□

Example 21 (LQ). *The logic **LQ** is characterized by the frame condition $\mathcal{F} = \forall xyz((x \leqslant y \& x \leqslant z) \to \exists w(y \leqslant w \& z \leqslant w)) = \forall xyz\exists w(\neg(x \leqslant y) \vee \neg(x \leqslant z) \vee (y \leqslant w \& z \leqslant w))$, where $\mathcal{F} \in \Pi_2$. The algorithm described in Theorem 10 works as follows:*

$$\frac{}{\Rightarrow \forall xyz\exists w(\neg(x \leqslant y) \vee \neg(x \leqslant z) \vee (y \leqslant w \& z \leqslant w))}$$

$$\longrightarrow \quad \frac{}{\Rightarrow \exists w(\neg(x \leqslant y) \vee \neg(x \leqslant z) \vee (y \leqslant w \& z \leqslant w))}$$

$$\longrightarrow \quad \frac{\exists w(\neg(x \leqslant y) \vee \neg(x \leqslant z) \vee (y \leqslant w \& z \leqslant w)), \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}$$

$$\longrightarrow \quad \frac{\neg(x \leqslant y) \vee \neg(x \leqslant z) \vee (y \leqslant w \& z \leqslant w), \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}$$

$$\longrightarrow \quad \frac{\neg(x \leqslant y), \Gamma \Rightarrow \Delta \qquad \neg(x \leqslant z), \Gamma \Rightarrow \Delta \qquad y \leqslant w \& z \leqslant w, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}$$

$$\longrightarrow \quad \frac{\Gamma \Rightarrow \Delta, x \leqslant y \qquad \Gamma \Rightarrow \Delta, x \leqslant z \qquad y \leqslant w, z \leqslant w, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}$$

Note that, while the rule schema (Π_1) coincides with the rule schema defined in [76] for regular formulas (see Section 5.2), this is not the case for geometric formulas. The rules of the form (Π_2) generated by our procedure from geometric formulas might contain relational atoms ($P_{i_1}, \dots, P_{i_m}$) on the right hand side of premises, see e.g. the rule generated for **LQ** in Example 21. Such rules are not of the form (*geom*) presented in [76], recall e.g. the rule from Example 16:

$$\frac{y \leqslant w, z \leqslant w, x \leqslant y, x \leqslant z, \Gamma \Rightarrow \Delta}{x \leqslant y, x \leqslant z, \Gamma \Rightarrow \Delta}\ (\Pi_2^G)$$

However, geometric formulas are Π_2 formulas of a *particular* shape. We show below that the (Π_2) rules that are generated by Theorem 10 for these formulas can easily be transformed into rules without relational atoms on the right hand side of the sequent. The resulting rules then coincide with the (*geom*) rules in [76].

Corollary 3. *Geometric axioms are equivalent to rules of the form* (*geom*).

Proof. Recall that geometric axioms are formulas in Π_2 of the form $\forall\overline{x}\exists\overline{y}A_G$, where A_G is a quantifier-free formula, see Section 5.2. W.l.o.g. we assume that A_G is in disjunctive normal form and hence has the shape $B_1 \vee \cdots \vee B_n \vee C_1 \vee \cdots \vee C_m$ where each B_i is $Q_{i_1} \& \cdots \& Q_{i_k}$ and each C_j is $\neg P_j$. Theorem 10 transforms such a geometric axiom into the equivalent rule

$$\frac{\{Q_{i_1}, \dots, Q_{i_k}, \Gamma \Rightarrow \Delta\}_{i=1\dots n} \quad \{\Gamma \Rightarrow \Delta, P_j\}_{j=1\dots m}}{\Gamma \Rightarrow \Delta}\ (\Pi_2')$$

The claim follows by showing that (Π_2') can be transformed into a rule

$$\frac{\overline{Q}_1, P_1, \dots, P_m, \Gamma \Rightarrow \Delta \quad \cdots \quad \overline{Q}_n, P_1, \dots, P_m, \Gamma \Rightarrow \Delta}{P_1, \dots, P_m, \Gamma \Rightarrow \Delta}\ (\Pi_2^G)$$

where each $\overline{Q}_i$ is a multiset of $Q_{i_1}, \dots, Q_{i_k}$. Observe that (Π_2^G) has no relational atom P_j on the right hand side and that it is a (*geom*) rule [76].

For one direction, we use (Π_2') and m initial sequents:

$$\frac{\{\overline{Q}_i, P_1, \dots, P_m, \Gamma \Rightarrow \Delta\}_{i=1\dots n} \quad \{P_1, \dots, P_m, \Gamma \Rightarrow \Delta, P_j\}_{j=1\dots m}}{P_1, \dots, P_m, \Gamma \Rightarrow \Delta}\ (\Pi_2')$$

For the other direction, we first apply (Π_2^G) followed by m applications of (*cut*) (and weakening):

$$\dfrac{\cdots \qquad \dfrac{\dfrac{\Gamma \Rightarrow \Delta, P_m}{P_1, \dots, P_{m-1}, \Gamma \Rightarrow \Delta, P_m}\,(w,l) \qquad \dfrac{\{\overline{Q}_i, P_1, \dots, P_m, \Gamma \Rightarrow \Delta\}_{i=1\dots n}}{P_1, \dots, P_m, \Gamma \Rightarrow \Delta}\,(\Pi_2^G)}{P_1, \dots, P_{m-1}, \Gamma \Rightarrow \Delta}\,(cut)}{\vdots}\,(cut)$$

$$\Gamma \Rightarrow \Delta$$

□

Example 22 (**LQ**). *The rule that we get by applying the algorithm in Theorem 10 for the frame condition defining the logic* ***LQ*** *is (see Example 21):*

$$\frac{\Gamma \Rightarrow \Delta, x \leqslant y \qquad \Gamma \Rightarrow \Delta, x \leqslant z \qquad y \leqslant w, z \leqslant w, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta} \; (\Pi_2')$$

while the rule obtained in [76] is:

$$\frac{y \leqslant w, z \leqslant w, x \leqslant y, x \leqslant z, \Gamma \Rightarrow \Delta}{x \leqslant y, x \leqslant z, \Gamma \Rightarrow \Delta} \; (\Pi_2^G)$$

(Π_2^G) *is then obtained from* (Π_2') *as follows:*

$$\frac{x \leqslant y, x \leqslant z, \Gamma \Rightarrow \Delta, x \leqslant y \qquad x \leqslant y, x \leqslant z, \Gamma \Rightarrow \Delta, x \leqslant z \qquad y \leqslant w, z \leqslant w, x \leqslant y, x \leqslant z, \Gamma \Rightarrow \Delta}{x \leqslant y, x \leqslant z, \Gamma \Rightarrow \Delta} \; (\Pi_2')$$

whereas (Π_2') *is derived from* (Π_2^G) *in the following way:*

$$\dfrac{\Gamma \Rightarrow \Delta, x \leqslant y \qquad \dfrac{\dfrac{\Gamma \Rightarrow \Delta, x \leqslant z}{x \leqslant y, x \leqslant z, \Gamma \Rightarrow \Delta, x \leqslant z} \; (w,l) \qquad \dfrac{y \leqslant w, z \leqslant w, x \leqslant y, x \leqslant z, \Gamma \Rightarrow \Delta}{x \leqslant y, x \leqslant z, \Gamma \Rightarrow \Delta} \; (\Pi_2^G)}{x \leqslant y, \Gamma \Rightarrow \Delta} \; (cut)}{\Gamma \Rightarrow \Delta} \; (cut)$$

Like the labelled sequent calculi for quantified modal and temporal logics presented in [48], our labelled calculi use rules that manipulate relational atoms in both sides of the sequent. In particular, the rules for non-geometric formulas within Π_2 are the only rules of our newly generated calculi that introduce relational atoms in the succedent. We will show that this is not an obstacle for the admissibility results analogous to those in Theorem 5. The base calculus we will use is a slightly modified version of *G3I* (see Table 5.1), which is obtained by adding initial sequents of the form

$$x \leqslant y, \Gamma \Rightarrow \Delta, x \leqslant y$$

to *G3I*. Note that these initial sequents are not present in the labelled systems for intermediate logics of [76]. We also consider structural rules manipulating labelled formulas and relational atoms on both sides of a sequent. Thus, in our base calculus, we use the following structural rules for contraction and weakening where Z is either a labelled formula $u : \varphi$ or a relational atom $x \leqslant y$:

$$\frac{\Gamma \Rightarrow \Delta}{Z, \Gamma \Rightarrow \Delta} \; (w,l) \quad \frac{\Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta, Z} \; (w,r) \quad \frac{Z, Z, \Gamma \Rightarrow \Delta}{Z, \Gamma \Rightarrow \Delta} \; (c,l) \quad \frac{\Gamma \Rightarrow \Delta, Z, Z}{\Gamma \Rightarrow \Delta, Z} \; (c,r)$$

From now on, we denote by *G3SI** the (super-intuitionistic) system obtained by adding to our base calculus rules of the form (Π_1) and (Π_2) which are defined by applying Theorem 10 to the set $*$ of formulas within the class Π_2 corresponding to the properties of the accessibility relation.

Lemma 10. *Substitution of variables is hp-admissible (see Definition 28) in G3SI*.*

Proof. We show that if $\Gamma \Rightarrow \Delta$ is derivable in *G3SI** and y is free for x in every formula in $\Gamma \Rightarrow \Delta$, then so is $\Gamma[y/x] \Rightarrow \Delta[y/x]$ with the same derivation height. Let d be the derivation of $\Gamma \Rightarrow \Delta$ in *G3SI**. We proceed by induction on $|d|$. When d ends in an axiom or the conclusion of $(\bot, l)$, we are done. Otherwise, consider the last inference rule r in d.

- If r is (Π_1) or any logical rule except $(\supset, r)$, the claim follows by application of the inductive hypothesis and r.
- If r is $(\supset, r)$ with the eigenvariable condition, then either the substitution is vacuous (and the result of the substitution is equivalent to $\Gamma \Rightarrow \Delta$), or otherwise, x is not an eigenvariable. Then we have to consider two cases: If y is not the eigenvariable that is introduced in the premise, we apply the inductive hypothesis and $(\supset, r)$. Else, y coincides with the eigenvariable, e.g. in the following situation:

$$\dfrac{\begin{array}{c}\vdots\\ x \leqslant y, y : \varphi, \Gamma' \Rightarrow \Delta', y : \psi\end{array}}{\Gamma' \Rightarrow \Delta', x : \varphi \supset \psi}\ (\supset, r)$$

  We first apply the inductive hypothesis, replacing y with a fresh metavariable z. Then we apply the inductive hypothesis to substitute x with y and $(\supset, r)$. We get

$$\dfrac{\begin{array}{c}\vdots\\ y \leqslant z, z : \varphi, \Gamma \Rightarrow \Delta', z : \psi\end{array}}{\Gamma' \Rightarrow \Delta', y : \varphi \supset \psi}\ (\supset, r)$$

- Finally, if r is a (Π_2) rule, we need some care to avoid clashing of eigenvariables. Suppose the last rule applied in d has premises of the form

$$\{Q_{i_1}, \ldots, Q_{i_n}, \Gamma' \Rightarrow \Delta', P_{i_1}, \ldots, P_{i_m}\}_{i=1,\ldots,k}$$

  with fresh variables $y'_1, \ldots, y'_n$ in each Q_{i_j} (eigenvariable condition). If $y \neq y'_h$ for all $h = 1, \ldots, n$, we use the inductive hypothesis and (Π_2). Otherwise, suppose that $y = y'_i$ for some i, then we first apply the inductive hypothesis, replacing y'_i with a fresh metavariable z. Then we use the inductive hypothesis to substitute x with y and (Π_2).

□

Lemma 11. *The rules of weakening are hp-admissible in G3SI*.*

Proof. Let d_p be the derivation of the premise of the weakening rule (let Z be either a labelled formula or a relational atom):

$$\dfrac{\begin{array}{c}\vdots d_p\\ \Gamma \Rightarrow \Delta\end{array}}{Z, \Gamma \Rightarrow \Delta}\ (w, l) \quad \text{or} \quad \dfrac{\begin{array}{c}\vdots d_p\\ \Gamma \Rightarrow \Delta\end{array}}{\Gamma \Rightarrow \Delta, Z}\ (w, r)$$

By induction on $|d_p|$. When d_p ends in an axiom or the conclusion of $(\bot, l)$, we are done. Otherwise, consider the last inference rule r in d_p.

- If r is any rule without eigenvariable condition, the claim follows by application of the inductive hypothesis and r.
- If r is $(\supset, r)$ or (Π_2), we have to avoid a clash of the eigenvariable with the variable in Z. For instance, consider the following situation:

$$\dfrac{\dfrac{\begin{matrix}\vdots\\ x \leqslant y, y : \varphi, \Gamma \Rightarrow \Delta, y : \psi\end{matrix}}{\Gamma \Rightarrow \Delta, x : \varphi \supset \psi}\ (\supset, r)}{\Gamma \Rightarrow \Delta, x : \varphi \supset \psi, y : \chi}\ (w, r)$$

  Then we first replace the eigenvariable with a fresh variable by Lemma 10. The claim then follows by applications of the inductive hypothesis and r, e.g. in the above situation:

$$\dfrac{\begin{matrix}\vdots\\ x \leqslant z, z : \varphi, \Gamma \Rightarrow \Delta, y : \chi, z : \psi\end{matrix}}{\Gamma \Rightarrow \Delta, x : \varphi \supset \psi, y : \chi}\ (\supset, r)$$

□

Lemma 12. *All rules of G3SI* are hp-invertible.*

Proof. We show that each rule is hp-invertible:

- Consider first the rule $(\wedge, l)$. We show that if $d \vdash_{G3SI^*} x : \varphi \wedge \psi, \Gamma \Rightarrow \Delta$ then one can find $d_1 \vdash_{G3SI^*} x : \varphi, x : \psi, \Gamma \Rightarrow \Delta$ such that $|d_1| \leq |d|$. By induction on $|d|$. Consider the last inference rule r.
  - If $x : \varphi \wedge \psi$ is the principal formula of the rule r, we are done.
  - Otherwise, $x : \varphi \wedge \psi$ is not the principal formula. Then the claim follows by applications of the inductive hypothesis and r.

  The proof is analogous for the logical rules of $(\wedge, r)$, $(\vee, l)$ and $(\vee, r)$.
- $(\supset, l)$ and all (Π_1)- and (Π_2) rules are hp-invertible since their premise(s) can be obtained by weakening from the conclusion which is hp-admissible (Lemma 11).
- For the rule $(\supset, r)$ we show the following: if $d \vdash_{G3SI^*} \Gamma \Rightarrow \Delta, x : \varphi \supset \psi$ then $d_1 \vdash_{G3SI^*} x \leqslant y, y : \varphi, \Gamma \Rightarrow \Delta, y : \psi$ such that $|d_1| \leq |d|$. We proceed by induction on $|d|$. When $\Gamma \Rightarrow \Delta, x : \varphi \supset \psi$ is an axiom or the conclusion of $(\bot, l)$, we are done. Otherwise, consider the last inference rule r in d.
  - If r is any logical rule and $x : \varphi \supset \psi$ is not the principal formula, e.g.

$$\dfrac{\begin{matrix}\vdots\\ \Gamma \Rightarrow \Delta, x : \varphi \supset \psi, v : \alpha, v : \beta\end{matrix}}{\Gamma \Rightarrow \Delta, x : \varphi \supset \psi, v : \alpha \vee \beta}\ (\vee, r)$$

The claim follows by applications of the inductive hypothesis and r:

$$\dfrac{\begin{array}{c}\vdots\\ x \leqslant y, y : \varphi, \Gamma \Rightarrow \Delta, v : \alpha, v : \beta, y : \psi\end{array}}{x \leqslant y, y : \varphi, \Gamma \Rightarrow \Delta, v : \alpha \vee \beta, y : \psi}\ (\vee, r)$$

– Else, let r be $(\supset, r)$ and $x : \varphi \supset \psi$ the principal formula. Then we are done (we might first have to use Lemma 10 to obtain the desired variables).

□

To ensure hp-admissibility of contraction in the resulting systems, the following closure condition needs to be satisfied:

Definition 32 (Closure Condition [76, 134]). Given a system with (Π_1)-rules, for every rule schema of the form

$$\frac{\overline{Q}, P_1, \ldots, P_{i-1}, P_i, P_i, P_{i+1}, \ldots, P_m, \Gamma \Rightarrow \Delta}{P_1, \ldots, P_{i-1}, P_i, P_i, P_{i+1}, \ldots, P_m, \Gamma \Rightarrow \Delta}$$

a rule schema

$$\frac{\overline{Q}, P_1, \ldots, P_{i-1}, P_i, P_{i+1}, \ldots, P_m, \Gamma \Rightarrow \Delta}{P_1, \ldots, P_{i-1}, P_i, P_{i+1}, \ldots, P_m, \Gamma \Rightarrow \Delta}$$

has to be included in the system.

This condition is not problematic as the number of rule schemas that need to be added to the system is finite.

Lemma 13. *The rules of contraction are hp-admissible in G3SI* provided the closure condition is satisfied.*

Proof. Let d_p be the derivation of the premise of the contraction rule (let Z be either a labelled formula or a relational atom):

$$\dfrac{\begin{array}{c}\vdots d_p\\ Z, Z, \Gamma \Rightarrow \Delta\end{array}}{Z, \Gamma \Rightarrow \Delta}\ (c, l) \qquad \text{or} \qquad \dfrac{\begin{array}{c}\vdots d_p\\ \Gamma \Rightarrow \Delta, Z, Z\end{array}}{\Gamma \Rightarrow \Delta, Z}\ (c, r)$$

By induction on $|d_p|$. When d_p ends in an axiom or $(\bot, l)$, then also the contracted sequent is an axiom or the conclusion of $(\bot, l)$. Otherwise, consider the last inference rule r in d_p. We have to distinguish two cases.

- If the contracted formula Z is not principal in r, both occurrences of Z are in the premise(s) of r. The claim then follows by applications of the inductive hypothesis and r.
- Else, Z is principal in r. Then we distinguish three cases:
  (1) r is a rule where the contracted formula also occurs in the premise(s), e.g. a (Π_1) rule or $(\supset, l)$. Consider the following case:

$$\dfrac{\begin{array}{cc}\vdots & \vdots \\ x \leqslant y, x : \varphi \supset \psi, x : \varphi \supset \psi, \Gamma \Rightarrow \Delta, y : \varphi & x \leqslant y, x : \varphi \supset \psi, x : \varphi \supset \psi, y : \psi, \Gamma \Rightarrow \Delta\end{array}}{x \leqslant y, x : \varphi \supset \psi, x : \varphi \supset \psi, \Gamma \Rightarrow \Delta}\ (\supset, l)$$

The claim follows by applications of the inductive hypothesis and r, i.e.:

$$\dfrac{\begin{array}{cc}\vdots & \vdots \\ x \leqslant y, x : \varphi \supset \psi, \Gamma \Rightarrow \Delta, y : \varphi & x \leqslant y, x : \varphi \supset \psi, y : \psi, \Gamma \Rightarrow \Delta\end{array}}{x \leqslant y, x : \varphi \supset \psi, \Gamma \Rightarrow \Delta}\ (\supset, l)$$

When both contracted formulas are principal in the rule (e.g. in (Π_1) rules), the claim holds by the closure condition (Definition 32).

(2) The premise(s) of r contains subformulas of the principal formula, i.e. for the rules $(\wedge, r)$, $(\wedge, l)$, $(\vee, r)$ or $(\vee, l)$. Consider the case for $(\wedge, l)$:

$$\dfrac{\begin{array}{c}\vdots \\ x : \varphi, x : \psi, x : \varphi \wedge \psi, \Gamma \Rightarrow \Delta\end{array}}{x : \varphi \wedge \psi, x : \varphi \wedge \psi, \Gamma \Rightarrow \Delta}\ (\wedge, l)$$

By hp-invertibility (Lemma 12) of $(\wedge, l)$, we obtain

$$x : \varphi, x : \psi, x : \varphi, x : \psi, \Gamma \Rightarrow \Delta$$

The claim then follows by applications of the inductive hypothesis and $(\wedge, l)$. Analogous for the rules $(\wedge, r)$, $(\vee, r)$ and $(\vee, l)$.

(3) The premise of r contains a subformula of the contracted formula and a new relational atom $x \leqslant y$, i.e., r is $(\supset, r)$:

$$\dfrac{\begin{array}{c}\vdots \\ x \leqslant y, y : \varphi, \Gamma \Rightarrow \Delta, y : \psi, x : \varphi \supset \psi\end{array}}{\Gamma \Rightarrow \Delta, x : \varphi \supset \psi, x : \varphi \supset \psi}\ (\supset, r)$$

As $(\supset, r)$ is hp-invertible (Lemma 12), we obtain

$$x \leqslant y, x \leqslant y, y : \varphi, y : \varphi, \Gamma \Rightarrow \Delta, y : \psi, y : \psi$$

The claim then follows by applications of the inductive hypothesis and $(\supset, r)$.

□

We now show the proofs for soundness and completeness for the calculi *G3SI**, which are similar to those in [132].

Let $\mathfrak{F}_{SI^*} = \langle W, \leqslant \rangle$ be a frame with the properties of the accessibility relation expressed as (Π_2 and Π_1) formulas in $*$. Let $L = \{x, y, z \ldots\}$ be the labels occurring in *G3SI**-derivations, $w \in W$. An *interpretation* $\mathcal{I}_l$ of L in $\mathfrak{F}_{SI^*}$ is a function $\mathcal{I}_l : L \to W$, s.t. $\mathcal{I}_l(x) = w$ and $\mathcal{I}_l(\leqslant) = \leqslant$.

Definition 33. Let $\mathfrak{M}_{SI^*} = \langle \mathfrak{F}_{SI^*}, \Vdash \rangle$ be a model and $\mathcal{I}_l$ an interpretation. A labelled sequent $\Gamma \Rightarrow \Delta$ is *valid in* $\mathfrak{M}_{SI^*}$ if for every interpretation $\mathcal{I}_l$ we have: if for all labelled formulas $x : \varphi$ and relational atoms $y \leqslant z$ in Γ, $\mathcal{I}_l(x) \Vdash \varphi$ and $\mathcal{I}_l(y) \leqslant \mathcal{I}_l(z)$ hold, then for some $w : \psi$, $u \leqslant v$ in Δ we have $\mathcal{I}_l(w) \Vdash \psi$ or $\mathcal{I}_l(u) \leqslant \mathcal{I}_l(v)$. A sequent $\Gamma \Rightarrow \Delta$ is *valid* in a frame $\mathfrak{F}_{SI^*}$ when it is valid in every model $\mathfrak{M}_{SI^*}$. based on the frame $\mathfrak{F}_{SI^*}$.

Theorem 11 (Soundness)**.** *For any sequent* $\Gamma \Rightarrow \Delta$*,*

$$\textit{if } \vdash_{G3SI^*} \Gamma \Rightarrow \Delta, \textit{ then } \Gamma \Rightarrow \Delta \textit{ is valid in every frame } \mathfrak{F}_{SI^*}.$$

Proof. Let d be the derivation of $\Gamma \Rightarrow \Delta$ in *G3SI**. By induction on $|d|$. If $\Gamma \Rightarrow \Delta$ is an axiom, there is either a labelled atom $x : p$ or a relational atom $x \leqslant y$ in both, Γ and Δ, and we are done. Similarly, if $\Gamma \Rightarrow \Delta$ is the conclusion of $(\bot, l)$. Otherwise, consider the last inference rule r in d.

- Let r be any logical rule except $(\supset, r)$, e.g. $(\wedge, l)$:

$$\dfrac{\begin{array}{c}\vdots\\ x : \varphi, x : \psi, \Gamma' \Rightarrow \Delta\end{array}}{x : \varphi \wedge \psi, \Gamma' \Rightarrow \Delta}\ (\wedge, l)$$

  By inductive hypothesis, we have the validity of the sequent $x : \varphi, x : \psi, \Gamma' \Rightarrow \Delta$. Since $x \Vdash \varphi$ and $x \Vdash \varphi$ iff $x \Vdash \varphi \wedge \psi$, the conclusion is also valid. Analogous for the other logical rules.
- If r is $(\supset, r)$, consider the following situation (note that y is an eigenvariable):

$$\dfrac{\begin{array}{c}\vdots\\ x \leqslant y, y : \varphi, \Gamma \Rightarrow \Delta', y : \psi\end{array}}{\Gamma \Rightarrow \Delta', x : \varphi \supset \psi}\ (\supset, r)$$

  Let $\mathcal{I}_l$ be an interpretation that makes all the formulas in Γ true. Let z be arbitrary s.t. $\mathcal{I}_l(x) \leqslant z$. By inductive hypothesis, we have validity of the premise. We show that either $x : \varphi \supset \psi$ or a formula in Δ' is valid. Let $\mathcal{I}'_l$ be an interpretation identical to $\mathcal{I}_l$ except for y, where $\mathcal{I}'_l(y)$ is assigned the value z. Since $\mathcal{I}'_l$ validates the antecedent of the premise, it also validates either a formula in Δ' (and hence, also $\mathcal{I}_l$ validates a formula in Δ') or $y : \psi$ (then, since z was arbitrary, $\mathcal{I}_l$ validates $x : \varphi \supset \psi$).
- If r is a (Π_1) rule (no eigenvariable condition), by inductive hypothesis we have validity of the premise. As the conclusion is contained in the premise, we also have validity of the conclusion.
- Finally, if r is a (Π_2) rule with eigenvariable condition, we have premise(s) of the form

$$\{\overline{Q_i}, \Gamma \Rightarrow \Delta, \overline{P_i}\}_{i=1\dots k}$$

  where $\overline{Q_i}$ are sets of relational atoms containing eigenvariables. E.g., consider the rule for **LQ** with w an eigenvariable:

$$\dfrac{\begin{array}{ccc}\vdots & \vdots & \vdots \\ \Gamma \Rightarrow \Delta, x \leqslant y & \Gamma \Rightarrow \Delta, x \leqslant z & y \leqslant w, z \leqslant w, \Gamma \Rightarrow \Delta\end{array}}{\Gamma \Rightarrow \Delta} \ (\Pi_2)$$

Let $\mathcal{I}_l$ be an interpretation that validates all the formulas in the antecedent of the premise(s). Let w' be arbitrary s.t. $\mathcal{I}_l(y) \leqslant w', \mathcal{I}_l(z) \leqslant w'$ (the $\overline{Q_i}$ from the premise(s)). By inductive hypothesis, we have validity of the premise(s). Let $\mathcal{I}'_l$ be an interpretation identical to $\mathcal{I}_l$ except on w where $\mathcal{I}'_l(w)$ is assigned the value w'. Since $\mathcal{I}'_l$ validates the antecedent of the premise, it also validates a formula in Δ (and hence, also $\mathcal{I}_l$ validates a formula in Δ).

□

To prove completeness, we use a direct method that shows how root-first proof search in the sequent system either gives a proof or leads to the construction of a counter model.

Theorem 12 (Completeness)**.** *For any sequent* $\Gamma \Rightarrow \Delta$

$$\textit{If } \Gamma \Rightarrow \Delta \textit{ is valid in every frame } \mathfrak{F}_{SI^*}, \textit{ then } \vdash_{G3SI^*} \Gamma \Rightarrow \Delta.$$

Proof. We show that each sequent $\Gamma \Rightarrow \Delta$ is either derivable in *G3SI** or it has a countermodel in a frame with properties expressed by formulas in $*$, i.e. a model that makes all labelled formulas and relational atoms in Γ true and all labelled formulas and relational atoms in Δ false.

We first construct in the usual manner a derivation tree for $\Gamma \Rightarrow \Delta$ by applying the rules of *G3SI** root first. If the derivation tree is finite, i.e., all leaves are axioms or conclusions of $(\bot, l)$, we have a proof in *G3SI**. Assume that the derivation tree is infinite. By König's lemma, it has an infinite branch that is used to build the needed counterexample. The derivation tree is constructed inductively in stages following [132]: Stage 0 has $\Gamma \Rightarrow \Delta$ at the root of the tree. Stage $s > 0$ has two cases: (1) If every topmost sequent is an axiom or conclusion of a zero-premise rule, the construction ends. (2) Otherwise, we continue constructing the tree by applying to the topmost sequents root-first the rules of *G3SI** in a particular order: There are $6 + p$ different stages, 6 for the rules of *G3I* and p for the (Π_1)- and (Π_2)-rules. The stages are then repeated, i.e., $s = 6 + p + 1$ is stage 1, etc.

- Stage $s = 1$ with $(\wedge, l)$: Let $x_1 : \varphi_1 \wedge \psi_1, \dots, x_m : \varphi_m \wedge \psi_m$ be all formulas in Γ having $\wedge$ as outermost connective. We add above any topmost sequent of the form $x_1 : \varphi_1 \wedge \psi_1, \dots, x_m : \varphi_m \wedge \psi_m, \Gamma' \Rightarrow \Delta$, a sequent $x_1 : \varphi_1, x_1 : \psi_1, \dots, x_m : \varphi_m, x_m : \psi_m, \Gamma' \Rightarrow \Delta$. This corresponds to applying the rule $(\wedge, l)$ m times root first.
- Stage $s = 2$ with $(\vee, r)$: Analogous to stage $s = 1$.
- Stage $s = 3$ with $(\wedge, r)$: Let $x_1 : \varphi_1 \wedge \psi_1, \dots, x_m : \varphi_m \wedge \psi_m$ be all formulas in topmost Δ having $\wedge$ as outermost connective. We add above any of such sequents 2^m sequents of the form $\Gamma \Rightarrow \Delta', x_1 : \chi_1, \dots, x_m : \chi_m$ where χ_i can be either φ_i or ψ_i.

- Stage $s = 4$ with $(\vee, l)$: Analogous to stage $s = 3$.
- Stage $s = 5$ with $(\supset, l)$: We add above any topmost sequent of the form $x_1 \leqslant y_1, x_1 : \varphi_1 \supset \psi_1, \ldots, x_m \leqslant y_m, x_m : \varphi_m \supset \psi_m, \Gamma' \Rightarrow \Delta$ 2^m sequents of the form $x_1 \leqslant y_1, x_1 : \varphi_1 \supset \psi_1, \ldots, x_m \leqslant y_m, x_m : \varphi_m \supset \psi_m, y_{i_1} : \psi_{i_1}, \ldots, y_{i_h} : \psi_{i_h}, \Gamma' \Rightarrow \Delta, y_{j_1} : \varphi_{j_1}, \ldots, y_{j_k} : \varphi_{j_k}$ where $i_1, \ldots, i_h \in \{1, \ldots, m\}$ and $j_1, \ldots, j_k \in \{1, \ldots, m\} \setminus \{i_1, \ldots, i_h\}$. This corresponds to applying the rule $(\supset, l)$ m times root first.
- Stage $s = 6$ with $(\supset, r)$: Let $x_1 : \varphi_1 \supset \psi_1, \ldots, x_m : \varphi_m \supset \psi_m$ be all formulas in topmost Δ having $\supset$ as outermost connective. We add above any of such sequents a sequent $x_1 \leqslant y_1, \ldots, x_m \leqslant y_m, y_1 : \varphi_1, \ldots, y_m : \varphi_m, \Gamma \Rightarrow \Delta', y_1 : \psi_1, \ldots, y_m : \psi_m$ with $y_1, \ldots, y_m$ fresh variables.
- Stage $s = 6 + j$ with (Π_1): We add above any topmost sequent of the form $P_1, \ldots, P_m, \Gamma' \Rightarrow \Delta$ a sequent $Q_1, \ldots, Q_n, P_1, \ldots, P_m, \Gamma \Rightarrow \Delta$.
- Stage $s = 6 + j$ with (Π_2): We add above any topmost sequent k sequents of the form $\{Q_{i_1}, \ldots, Q_{i_n}, \Gamma \Rightarrow \Delta, P_{i_1}, \ldots, P_{i_m}\}$, for $i = 1, \ldots, k$ and with $y_1, \ldots, y_n$ fresh variables.

For any s, if the sequent is neither an axiom, nor $(\bot, l)$ and none of the stages apply, we write the sequent itself above it.

If the derivation tree is infinite, it has an infinite branch. Now let $\Gamma \Rightarrow \Delta = \Gamma_0 \Rightarrow \Delta_0, \Gamma_1 \Rightarrow \Delta_1, \ldots, \Gamma_i \Rightarrow \Delta_i, \ldots$ be one such branch. Consider the sets $\mathbf{\Gamma} \equiv \bigcup \Gamma_i$ and $\mathbf{\Delta} \equiv \bigcup \Delta_i$ for $i \geq 0$. We now construct a countermodel, i.e. a model that makes all labelled formulas and relational atoms in $\mathbf{\Gamma}$ true and all labelled formulas and relational atoms in $\mathbf{\Delta}$ false. Let $\mathfrak{F}_{SI^*}$ be a frame whose elements are all the labels occurring in $\mathbf{\Gamma}, \mathbf{\Delta}$. The interpretation $\mathcal{I}_l$ on $\mathfrak{F}_{SI^*}$, the relation $\leqslant$ of $\mathfrak{F}_{SI^*}$ and the relation $\Vdash$ are defined as follows:

(i) for all $x : p$ in $\mathbf{\Gamma}$ it holds that $\mathcal{I}_l(x) \Vdash p$ in $\mathfrak{F}_{SI^*}$,
(ii) for all $x \leqslant y$ in $\mathbf{\Gamma}$ we have $\mathcal{I}_l(x) \leqslant \mathcal{I}_l(y)$ in $\mathfrak{F}_{SI^*}$,
(iii) for all $x' : p'$ in $\mathbf{\Delta}$ we have $\mathcal{I}_l(x') \nVdash p'$ in $\mathfrak{F}_{SI^*}$, and finally
(iv) for all $x' \leqslant y'$ in $\mathbf{\Delta}$ it holds that $\mathcal{I}_l(x') \nleqslant \mathcal{I}_l(y')$ in $\mathfrak{F}_{SI^*}$.

$\mathfrak{F}_{SI^*}$ is well defined as it is not the case that either $x \leqslant y$ is in Γ_i and $x \leqslant y$ is in Δ_j or $x \leqslant y, x : p$ is in Γ_i and $y : p$ is in Δ_j (for any i and j), as otherwise we would have an axiom and therefore the branch would be finite, against the hypothesis. We then show that for any formula φ, φ is forced at $\mathcal{I}_l(x)$ if $x : \varphi$ is in $\mathbf{\Gamma}$ and φ is not forced at $\mathcal{I}_l(x)$ if $x : \varphi$ is in $\mathbf{\Delta}$. As all relational atoms in $\mathbf{\Gamma}$ are true and those in $\mathbf{\Delta}$ are false by definition of $\mathfrak{F}_{SI^*}$ we have a countermodel to $\Gamma \Rightarrow \Delta$. We proceed by induction on $|\varphi|$.

- If φ is $\bot$, it cannot be in $\mathbf{\Gamma}$ because no sequent in the branch contains $x : \bot$ in the antecedent, so it is not forced at any node of the model. If φ is an atom p in $\mathbf{\Gamma}$ then $\mathcal{I}_l(x) \Vdash p$ by definition; and $\mathcal{I}_l(x) \nVdash p$ if it is in $\mathbf{\Delta}$.
- If $x : \varphi \equiv x : \varphi_1 \wedge \varphi_2$ is in $\mathbf{\Gamma}$, there exists i such that $x : \varphi_1 \wedge \varphi_2$ appears first in Γ_i, and therefore, for some $j \geq 0$, $x : \varphi_1$ and $x : \varphi_2$ are in Γ_{i+j} . By inductive hypothesis, $x \Vdash \varphi_1$ and $x \Vdash \varphi_2$ and therefore $x \Vdash \varphi_1 \wedge \varphi_2$ (analogous for $x : \varphi \equiv x : \varphi_1 \vee \varphi_2$ in $\mathbf{\Delta}$).

- If $x : \varphi \equiv x : \varphi_1 \wedge \varphi_2$ is in $\mathbf{\Delta}$ then either $x : \varphi_1$ or $x : \varphi_2$ is in $\mathbf{\Delta}$. By inductive hypothesis, $x \nVdash \varphi_1$ or $x \nVdash \varphi_2$ and therefore $x \nVdash \varphi_1 \wedge \varphi_2$ (analogous for $x : \varphi \equiv x : \varphi_1 \vee \varphi_2$ in $\mathbf{\Gamma}$).
- If $x : \varphi \equiv x : \varphi_1 \supset \varphi_2$ is in $\mathbf{\Gamma}$, we consider all the relational atoms $x \leqslant y$ that occur in $\mathbf{\Gamma}$. If there is no such atom then $x \Vdash \varphi_1 \supset \varphi_2$ is in the model. Else, for any occurrence of $x \leqslant y$ in $\mathbf{\Gamma}$, by construction of the tree either $y : \varphi_1$ is in $\mathbf{\Delta}$ or $y : \varphi_2$ is in $\mathbf{\Gamma}$. By inductive hypothesis $y \nVdash \varphi_1$ or $y \Vdash \varphi_2$, and since this holds for all y with $x \leqslant y$ we have $x \Vdash \varphi_1 \supset \varphi_2$ in the model.
- If $x : \varphi \equiv x : \varphi_1 \supset \varphi_2$ is in $\mathbf{\Delta}$, due to one of the following steps of the derivation tree we have that $x \leqslant y$ and $y : \varphi_1$ are in $\mathbf{\Gamma}$, whereas $y : \varphi_2$ is in $\mathbf{\Delta}$. By inductive hypothesis this gives $x \leqslant y$ and $y \Vdash \varphi_1$ but $y \nVdash \varphi_2$, i.e. $x \nVdash \varphi_1 \supset \varphi_2$.

□

Corollary 4. *For any sequent* $\Gamma \Rightarrow \Delta$, $\vdash_{G3SI^*} \Gamma \Rightarrow \Delta$ *iff* $\Gamma \Rightarrow \Delta$ *is valid in every frame* $\mathfrak{F}_{SI^*}$.

Theorem 13 (Cut elimination)**.** *The cut rule*

$$\frac{\Gamma \Rightarrow \Delta, Z \quad Z, \Gamma' \Rightarrow \Delta}{\Gamma, \Gamma' \Rightarrow \Delta', \Delta}\ (cut)$$

where Z *is either a labelled formula* $y : \varphi$ *or a relational atom* $x \leqslant y$*, can be eliminated from G3SI*-derivations.*

Proof. We distinguish two cases according to the cut formula Z.

When Z is a *labelled formula* $y : \varphi$, we proceed by a double induction on the complexity of Z and the sum of the derivation heights of the premises of the cut.

In the base case, i.e. one or both of the derivations end in an axiom or are the conclusion of $(\bot, l)$, we are done. The only interesting case is when the cut formula Z is $y : p$ and both premises are axioms, e.g.:

$$\frac{x \leqslant y, x : p, \Gamma \Rightarrow \Delta, y : p \qquad y \leqslant z, y : p, \Gamma' \Rightarrow \Delta', z : p}{x \leqslant y, y \leqslant z, x : p, \Gamma', \Gamma \Rightarrow \Delta, \Delta', z : p}\ (cut)$$

Then we replace (cut) by an application of $(trans)$ to the axiom

$$x \leqslant y, y \leqslant z, x \leqslant z, x : p, \Gamma', \Gamma \Rightarrow \Delta, \Delta', z : p$$

Otherwise, let r_l (r_r) be the last rule applied in the derivation of the left (right) premise, i.e., the premise having the cut formula $y : \varphi$ on the left (right).

- If the cut formula is the principal formula of a logical rule in any or both of the premises, we apply invertibility of the logical rules (Lemma 12) and replace the cut by smaller ones (note that for the case of $Z = y : \varphi_1 \supset \varphi_2$, we also need to apply Lemma 10 to avoid a clash of variables).

- Else, the claim follows by applications of the inductive hypothesis and r_l (r_r). Note that when permuting a cut with a rule with eigenvariable condition (i.e., (Π_2) or ($\supset, r$)), we also have to use appropriate substitutions to avoid a clash of variables (Lemma 10).

When Z is a *relational atom* $x \leqslant y$, the proof proceeds by induction on the derivation height of the right premise of the cut, i.e., $\Gamma \Rightarrow \Delta, x \leqslant y$.

In the base case, i.e. the derivation ends in an axiom or as conclusion of ($\bot, l$), we have either

$$(i)\ u \leqslant v, u : p, \Gamma'' \Rightarrow \Delta'', v : p, x \leqslant y$$
$$(ii)\ u \leqslant v, \Gamma'' \Rightarrow \Delta'', u \leqslant v, x \leqslant y$$
$$(iii)\ u : \bot, \Gamma'' \Rightarrow \Delta'', x \leqslant y \text{ or}$$
$$(iv)\ x \leqslant y, \Gamma'' \Rightarrow \Delta'', x \leqslant y$$

If (i)–(iii), the conclusion of (cut) is also an axiom. Otherwise, if (iv), the conclusion of (cut) can be obtained by weakening (Lemma 11).

Otherwise, let r_r be the last rule applied in the derivation of the right premise, i.e., the premise having the cut formula $x \leqslant y$ on the right. We show that the cut can then be shifted over the premise(s) of r_r. The main observation is that $x \leqslant y$ is not affected by the application of r_r since the rules of *G3SI** do not change relational atoms appearing on the right hand side of its conclusion.

- If r_r is a rule other than ($\supset, r$) or (Π_2), the claim follows by applications of the inductive hypothesis and r_r. For instance, let r_r be (Π_1). Then the derivation

$$\dfrac{\dfrac{\begin{matrix}\vdots\\ Q_1,\dots,Q_m,P_1,\dots,P_n,\Gamma'' \Rightarrow \Delta'', x \leqslant y\end{matrix}}{P_1,\dots,P_n,\Gamma'' \Rightarrow \Delta'', x \leqslant y}(\Pi_1) \qquad \begin{matrix}\vdots\\ x \leqslant y, \Gamma' \Rightarrow \Delta'\end{matrix}}{P_1,\dots,P_n,\Gamma',\Gamma'' \Rightarrow \Delta',\Delta''}(cut)$$

is transformed into

$$\dfrac{\dfrac{\begin{matrix}\vdots\\ Q_1,\dots,Q_m,P_1,\dots,P_n,\Gamma'' \Rightarrow \Delta'', x \leqslant y\end{matrix} \qquad \begin{matrix}\vdots\\ x \leqslant y, \Gamma' \Rightarrow \Delta'\end{matrix}}{Q_1,\dots,Q_m,P_1,\dots,P_n,\Gamma',\Gamma'' \Rightarrow \Delta',\Delta''}(cut)}{P_1,\dots,P_n,\Gamma',\Gamma'' \Rightarrow \Delta',\Delta''}(\Pi_1)$$

- If r_r is ($\supset, r$) or a rule following the (Π_2) scheme, the claim follows by applications of the inductive hypothesis, weakenings and r_r. Note that we first have to replace the eigenvariables in the premise(s) of r_r (Lemma 10) and then permute cut and r_r.

□

5.4.1 Tool: *Framinator*

The tool *Framinator* (FRAMe condItioNs Automatically TO Rules) implements the algorithm introduced in Section 5.4. It takes as input frame conditions in the language of first-order classical logic and, if possible, transforms it into equivalent labelled sequent rules.

Framinator can be used to construct cut-free calculi for many intermediate logics, such as Gödel logic [91], the logics of Kripke models with k worlds $\mathbf{Bc}_k$ [49] or the logics of Kripke models with width $\leq k$ $\mathbf{Bw}_k$ [49]. It is available at

`www.logic.at/tinc/webframinator/`

Example

In the main screen of *Framinator* (see Figure 5.1) the user can enter the frame condition in the text field.

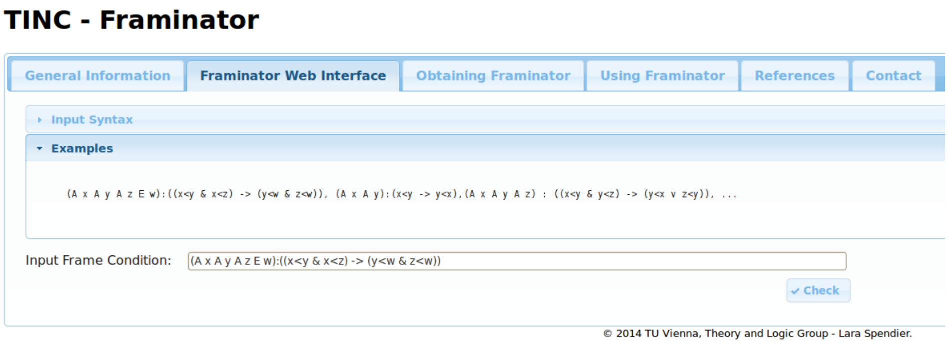


Figure 5.1: Main screen of *Framinator*

After the computation, a dialog box containing the results pops up, see Figure 5.2. The output contains the class of the frame conditions in the hierarchy, the computed rules in text format, as well as a link to the generated paper containing the obtained calculus along with a basic description of the system.

On the command-line, the program is started by typing `compute`. A prompt tells the user that he can enter the frame condition. The formula must have the form `(prefix) : (matrix)`, the quantifiers in the prefix are written `A` (for $\forall$) and `E` (for $\exists$) and the accessibility relation $\leq$ is abbreviated with the symbol `<`. While the class of the frame conditions and the corresponding rules are printed on the screen, the LaTeX-file is saved in a program folder on the computer. Note that in the text representation of the rule, `G` and `D` stand for multisets of formulas Γ and Δ.

In this example, we want to generate the rule equivalent to the frame condition $\forall x, y, z((x \leqslant y \& x \leqslant z) \rightarrow \exists w(y \leqslant w \& z \leqslant w))$ for Jankov logic $\mathbf{LQ}$.

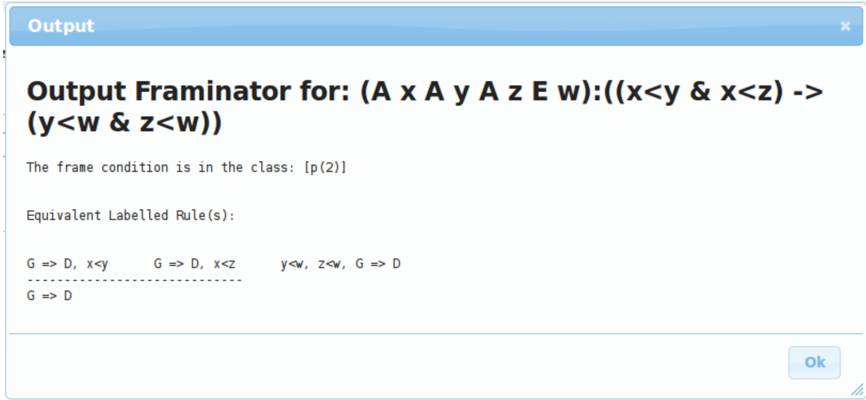


Figure 5.2: Dialog box containing the results

```
?- compute.
|: ('A' x 'A' y 'A' z 'E' w):((x<y & x<z) -> (y<w & z<w)).

The frame condition is in the class: [p(2)]

Equivalent Labelled Rule(s):

G => D,x<y      G => D,x<z      y<w,z<w,G => D
------------------------------
G => D
```

Implementation Details

Framinator is implemented in Prolog. The implementation consists of 10 files and roughly 1000 lines of code (including documentation) and follows the general **TINC**-structure described in Chapter 3 (recall Figure 3.5). The specific instantiation for *Framinator* is depicted in Figure 5.3.

Input and CHECKINPUT. As described in the general section, the first component, CHECKINPUT, takes as input a frame condition and checks whether it has the correct form to be handled by the transformation procedure. This is implemented by using a definite clause grammar (`frames2tex`).

Moreover, the input formula is required to be in prenex normal form where the prefix is separated from the matrix by an ':', i.e. `(prefix) : (matrix)`. Note that the prefix and the matrix have to be enclosed in brackets `()`. The syntax of the input formula consists of:

- the letters `[a-z]` except `v` for variables,
- the symbol `<` denoting the accessibility relation $\leq$,

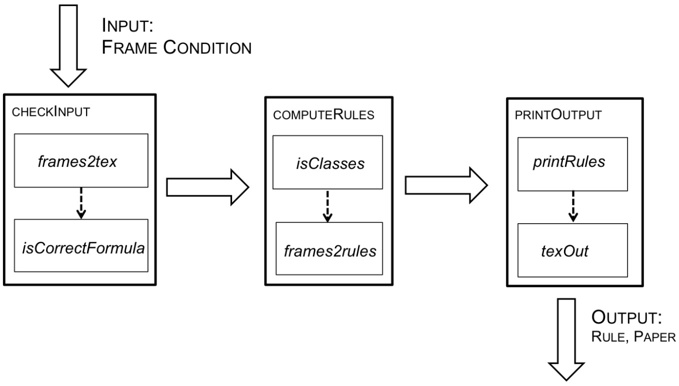


Figure 5.3: Design of *Framinator*

- logical connectives: `&` (and), `v` (or), `->` (implication) and `-` (negation),
- quantifiers: `A` (universal quantifier) and `E` (existential quantifier).

Each formula should be closed, i.e. no free variables should occur in the matrix. Moreover, the user can concatenate several frame conditions by separating them with a semicolon '`;`'. In the method `isCorrectFormula`, we check whether the input formula satisfies all these syntactic criteria.

computeRules. The second component, computeRules, contains the implementation of the algorithm:

- `isClasses` identifies the classes of the frame conditions given as input within the hierarchy of Definition 31 and indicates them as `s(i)` for Σ_i or `p(i)` for Π_i. Note that for the computation of the class, only the prefix of the formula needs to be considered since the hierarchy depends on the alternation of the quantifiers. See Code Example 4 below for the implementation of the class check.
- If the (highest) class of the frame conditions is within Π_2, `frames2rules` transforms the frame conditios into equivalent labelled sequent rules.

Code Example 4. `isClasses` *first determines if there is more than one frame condition given as input. For every frame condition, we determine the class of the input formula with* `isClass(Prefix,Class)` *and then use* `checkClasses(Class,OK)` *to see whether we can apply our algorithm (*`OK` *is the result of the check). We show (parts of) the code to determine whether the input formula is within a class* Π_N *(the formula might also be within a class* Σ_N*, but we omit these parts):*

```
%%% isClasses(+Formula, -Class)
%%  + ... parameter given as input, - ... return value
%%  compute the classes of the (possibly concatenated) frame
%%  conditions based on the quantifier alternation in the prefix
```

```
%%  Formula ... input formula
%%  Class  ... Class of the formula: Pi_N: p(N) or Sigma_N: s(N)
isClasses(X ';' Y, Class) :- % if two formulas are concatenated
    X = Prefix ':' _,    % we only need the prefix of a formula
    isClass(Prefix, C1), % and check the class level
    isClasses(Y, C2),    % same is done for the second formula
    append([C1], C2, Class).
isClasses(X, Class) :-  % if there is only one formula
    X = Prefix ':' _,   % we only need to consider this prefix
    isClass(Prefix, C1),  % and determine the class
    Class = [C1].

%%% isClass(+Prefix, -Class)
%%%    compute the class of the frame condition based on
%%%    the quantifier alternation in the prefix
%%%     +Prefix ... prefix of the input formula
%%%     -Class  ... Class of the formula: Pi_N: p(N) or Sigma_N: s(N)
isClass(Prefix, Class) :-
    Prefix = 'A' _,          % if the prefix starts with A
    isPiClass(Prefix, 0, N), % then the formula is in Pi
    Class = p(N).
isClass(Prefix, Class) :-
    Prefix = 'E' _,          % if the prefix starts with E
    isSigmaClass(Prefix, 0, N), % then the formula is in Sigma
    Class = s(N).

%%% isPiClass(+Prefix, +Class, -ClassNew)
%%%     compute the class level
%%%     +Prefix ... prefix of the input formula
%%%     +Class  ... Class of the formula: Pi_N: p(N) or Sigma_N: s(N)
%%%     -ClassNew ... final class of the formula
isPiClass(Prefix, P0, P0) :-
    % if the formula is a formula in pi, the class remains the same
    isPiFormula(Prefix, P0).
isPiClass(Prefix, P0, P2) :-
    % the if formula is NOT a formula in pi, level has to be increased
    \+ isPiFormula(Prefix, P0),
    P1 is P0+1,
    isPiClass(Prefix, P1, P2).

%%% isPiFormula(+Prefix, -Level)
%%% computes the class level
%%%     +Prefix ... quantifier part of the input formula
```

```
%%%     -Level  ... Class level based on the quantifier alternation
isPiFormula(F, 0) :-
    atomic(F).    % if F is atomic, level = 0
isPiFormula('A' F, 1) :-
    atomic(F).    % if F only contains one quantifier A, level = 1
isPiFormula('A' X 'A' F, P1) :-
    atomic(X),    % if X is just a variable
    P1 > 0,       % and the old level > 0
    isPiFormula('A' F, P1). % then we only need to check level of F
isPiFormula('A' X 'E' F, P1) :-
    atomic(X),   % if X is just a variable
    P1 > 0,      % and the old level > 0
    P is P1-1,   % we decreise the old level and check
    isSigmaFormula('E' F, P). % the level of the "sigma formula"

%%% checkClasses(+Class, -OK)
%%%  recursively checks if the classes of all frame conditions can be
%%%  handled by our algorithm (i.e., if they are in s(1), p(1) or p(2)
%%%    +Class ... List of classes of the various frame conditions
%%%    -OK ... 1 if all classes are within p(2), 0 if not
checkClasses([], 1).
checkClasses([H|T], OK) :-  % we take the first list element H
    % and check if it is equivalent to s(1), p(1) or p(2)
    member(H, [s(1), p(1), p(2)]),
% if yes, we check the other list elements
    checkClasses(T, OK).
checkClasses([H|_], OK) :-
    % if the first list element is not within p(2)
    \+ member(H, [s(1), p(1), p(2)]),
% we stop and set OK = 0
    OK = 0.
```

Output and printOutput. The last component, printOutput, contains the method to print the generated rules on the command-line or web interface (`printRules`); moreover, the method `texOut` generates a LaTeX-paper containing the resulting cut-free labelled sequent calculus.

CHAPTER 6

Paraconsistent Logics

Paraconsistent logics are logics which are not trivialized in the presence of inconsistency. This means that there are some formulas ψ, φ, such that $\psi, \neg\psi \nvdash \varphi$. As paraconsistent logics do not "explode" in presence of contradictions, they are widely used as tool to handle inconsistencies in many areas of computer science [36, 98, 156, 108], e.g. in software engineering, or when merging information from several agents or multiple sources. Paraconsistent logics are usually introduced Hilbert-style by extending the positive fragment of propositional classical logic $\mathbf{Cl}^+$ with suitable axioms.

In this chapter, we describe a procedure to introduce calculi and semantics for a large class of paraconsistent logics automatically. In the first step of our method, we transform the Hilbert axioms describing the paraconsistent logic into equivalent sequent rules, hence generating a sequent calculus for it. In the second step, we extract semantics out of the sequent calculus using the framework of partial non-deterministic matrices (PNmatrices) [23]. The semantics allows us to reason about important properties, e.g. decidability of the logic or analyticity of the calculus. We also present the implementation of this procedure for a particular case in the **TINC**-tool *Paralyzer*.

We first settle the basic notions and then present some examples of propositional paraconsistent logics, as well as related work in proof theory regarding the (semi-)automated introduction of analytic calculi for these logics. The following sections contain our theoretical contributions: In Section 6.4, we introduce the first step of our systematic procedure to automatically generate a sequent calculus. The second step of our procedure, i.e. extracting a partial non-deterministic matrix from the obtained calculus, is described in Section 6.5. We also explain how the semantics can be used to investigate the logic and the generated calculus, by showing that the PNmatrix induces a decision procedure for the corresponding logic, and allows to reason about the analyticity of the calculus. Section 6.6 contains a refinement of the general results for a specific subclass, for which the semantics extracted from the calculi is simpler and allows a better characterization of the analyticity property. Our **TINC**-tool *Paralyzer* is described in Section 6.6.1.

This chapter is based on the publications [55, 56].

6.1 Preliminaries

Syntax

The language $\mathcal{L}_{cl}^{+}$ of the considered logics is that of the positive fragment of propositional classical logic extended with finitely many unary connectives. Recall that $\mathcal{L}_{cl}^{+}$ consists of infinitely many (possibly indexed) propositional variables $p, q, \ldots$, the connectives $\wedge$ (conjunction), $\vee$ (disjunction) and $\supset$ (implication). As usual, formulas are built from atoms using the logical connectives. (Metavariables for) Formulas are denoted by (possibly indexed) $\varphi, \psi, \alpha, \beta, \ldots$ and (metavariables of) multisets of formulas are written as $\Gamma, \Delta, \ldots$.

We identify a language with its set of formulas, e.g. we write $\varphi \in \mathcal{L}$. $\mathcal{U}_{\mathcal{L}}$ denotes the set of unary connectives of $\mathcal{L}$, whereas $\mathcal{U}_{\mathcal{L}}^{*}$ denotes the set of all finite sequences of connectives from $\mathcal{U}_{\mathcal{L}}$ with the empty sequence denoted by ϵ and $\bar{\star}, \bar{\triangleright}$ for arbitrary such finite sequences. We also employ standard notations for their concatenation (e.g., when writing expressions like $\bar{\star}\bar{\triangleright}$).

Let HCL^{+} be a Hilbert system for $\mathbf{Cl}^{+}$, e.g.:

(Schematic) Axioms

(A1) $\varphi \supset (\psi \supset \varphi)$
(A2) $(\varphi \supset (\psi \supset \chi)) \supset ((\varphi \supset \psi) \supset (\varphi \supset \chi))$
(A3) $\varphi \wedge \psi \supset \varphi$
(A4) $\varphi \wedge \psi \supset \psi$
(A5) $\varphi \supset (\psi \supset \varphi \wedge \psi)$
(A6) $\varphi \supset \varphi \vee \psi$
(A7) $\psi \supset \varphi \vee \psi$
(A8) $(\varphi \supset \chi) \supset ((\psi \supset \chi) \supset (\varphi \vee \psi \supset \chi))$
(A9) $((\psi \supset \varphi) \supset \psi) \supset \psi$

Inference rules

- *modus ponens* (MP): for given formulas φ and $\varphi \supset \psi$, we obtain ψ; $\dfrac{\varphi \quad \varphi \supset \psi}{\psi}$

In this chapter we use a label-based formulation of ordinary Gentzen sequent calculi to simplify the presentation of our results:

Definition 34. Let $\mathcal{L}$ be a propositional language.

1. A *labelled $\mathcal{L}$-formula* has the form $\mathtt{b} : \psi$, where $\mathtt{b} \in \{\mathtt{f}, \mathtt{t}\}$ and ψ is an $\mathcal{L}$-formula.
2. An *$\mathcal{L}$-sequent* is a finite set of labelled $\mathcal{L}$-formulas. The usual sequent notation

$$\psi_1, \ldots, \psi_n \Rightarrow \varphi_1, \ldots, \varphi_m$$

corresponds to the set of labelled formulas

$$\{\mathtt{f} : \psi_1, \ldots, \mathtt{f} : \psi_n, \mathtt{t} : \varphi_1, \ldots, \mathtt{t} : \varphi_m\}$$

(id)	$\emptyset/\{\mathtt{f}:p_1,\mathtt{t}:p_1\}$	(cut)	$\{\{\mathtt{f}:p_1\},\{\mathtt{t}:p_1\}\}/\emptyset$
$(\mathtt{f}:W)$	$\{\emptyset\}/\{\mathtt{f}:p_1\}$	$(\mathtt{t}:W)$	$\{\emptyset\}/\{\mathtt{t}:p_1\}$
$(\mathtt{f}:\wedge)$	$\{\{\mathtt{f}:p_1,\mathtt{f}:p_2\}\}/\{\mathtt{f}:p_1\wedge p_2\}$	$(\mathtt{t}:\wedge)$	$\{\{\mathtt{t}:p_1\},\{\mathtt{t}:p_2\}\}/\{\mathtt{t}:p_1\wedge p_2\}$
$(\mathtt{f}:\vee)$	$\{\{\mathtt{f}:p_1\},\{\mathtt{f}:p_2\}\}/\{\mathtt{f}:p_1\vee p_2\}$	$(\mathtt{t}:\vee)$	$\{\{\mathtt{t}:p_1,\mathtt{t}:p_2\}\}/\{\mathtt{t}:p_1\vee p_2\}$
$(\mathtt{f}:\supset)$	$\{\{\mathtt{t}:p_1\},\{\mathtt{f}:p_2\}\}/\{\mathtt{f}:p_1\supset p_2\}$	$(\mathtt{t}:\supset)$	$\{\{\mathtt{f}:p_1,\mathtt{t}:p_2\}\}/\{\mathtt{t}:p_1\supset p_2\}$

Table 6.1: Label-based sequent calculus LK^+

3. An $\mathcal{L}$-*substitution* is a function $\sigma:\mathcal{L}\to\mathcal{L}$, such that

$$\sigma(\heartsuit(\psi_1,\ldots,\psi_n)) = \heartsuit(\sigma(\psi_1),\ldots,\sigma(\psi_n))$$

for every n-ary connective $\heartsuit$ of $\mathcal{L}$. $\mathcal{L}$-substitutions are naturally extended to labelled $\mathcal{L}$-formulas, $\mathcal{L}$-sequents, and sets of $\mathcal{L}$-sequents.
4. An $\mathcal{L}$-*rule* is an expression of the form Q/s, where Q is a finite set of $\mathcal{L}$-sequents (called *premises*) and s is an $\mathcal{L}$-sequent (called *conclusion*). An *application* of an $\mathcal{L}$-rule Q/s is any inference step inferring the $\mathcal{L}$-sequent $\sigma(s)\cup c$ from the set of $\mathcal{L}$-sequents $\{\sigma(q)\cup c \mid q\in Q\}$, where σ is an $\mathcal{L}$-substitution, and c is an $\mathcal{L}$-sequent.
5. A label-based sequent calculus G for $\mathcal{L}$ consists of a finite set of $\mathcal{L}$-rules. We write $\mathcal{S}\vdash_G s$ whenever the $\mathcal{L}$-sequent s is derivable from the set $\mathcal{S}$ of $\mathcal{L}$-sequents in G.

Example 23. *Examples of rules and their applications (in standard sequent notation) are:*

$(\mathtt{f}:\neg\neg)$ $\quad \{\{\mathtt{f}:p_1\}\}/\{\mathtt{f}:\neg\neg p_1\}$ $\quad \dfrac{\Gamma,\varphi\Rightarrow\Delta}{\Gamma,\neg\neg\varphi\Rightarrow\Delta}$

$(\mathtt{t}:\neg\neg\neg)$ $\quad \{\{\mathtt{t}:\neg p_1\}\}/\{\mathtt{t}:\neg\neg\neg p_1\}$ $\quad \dfrac{\Gamma\Rightarrow\neg\varphi,\Delta}{\Gamma\Rightarrow\neg\neg\neg\varphi,\Delta}$

$(\mathtt{t}:\neg\neg\wedge)$ $\quad \{\{\mathtt{t}:\neg\neg p_1\},\{\mathtt{t}:\neg\neg p_2\}\}/\{\mathtt{t}:\neg\neg(p_1\wedge p_2)\}$ $\quad \dfrac{\Gamma\Rightarrow\neg\neg\varphi,\Delta\quad \Gamma\Rightarrow\neg\neg\psi,\Delta}{\Gamma\Rightarrow\neg\neg(\varphi\wedge\psi),\Delta}$

The sequent calculus LK^+ for $\mathbf{Cl}^+$ is the standard calculus LK without the rules for negation and $\bot$. Its label-based formulation is presented in Table 6.1. Note that LK^+ is equivalent (in the sense of Definition 7) to HCL^+.

Let H be a Hilbert system and G be a sequent calculus. We denote by $H\cup\{\varphi\}$ ($H\setminus\{\varphi\}$ resp.) the Hilbert system obtained from H by adding (removing) the axiom φ, and by $G\cup R$ the sequent calculus extending G with the set R of $\mathcal{L}$-rules. Moreover, we define the *equivalence* of rules with respect to a sequent calculus:

Definition 35. Let R and R' be two finite sets of $\mathcal{L}$-rules, and G be a sequent calculus for $\mathcal{L}$. R and R' are *equivalent in* G if $Q\vdash_{G\cup R'} s$ for every $Q/s\in R$, and $Q'\vdash_{G\cup R} s'$ for every $Q'/s'\in R'$.

The definition of equivalence between rules for label-based sequent calculi can also be reformulated by considering rule applications:

Proposition 2. *Let R and R' be two finite sets of $\mathcal{L}$-rules, and G be a sequent calculus for $\mathcal{L}$. R and R' are equivalent in G iff the following hold for every $\mathcal{L}$-sequent c and substitution σ: $\sigma(Q) \cup c \vdash_{G\cup R'} \sigma(s) \cup c$ for every $Q/s \in R$, and $\sigma(Q') \cup c \vdash_{G\cup R} \sigma(s') \cup c$ for every $Q'/s' \in R'$.*

Proof. "⇒": Suppose that R and R' are equivalent in G. Let σ be a substitution and c an $\mathcal{L}$-sequent. For every $Q/s \in R$, by the equivalence of R and R', $Q \vdash_{G\cup R'} s$. Since $\{\sigma(q) \cup c \mid q \in Q\} \vdash_{G\cup R} \sigma(s) \cup c$, it holds that $\{\sigma(q) \cup c \mid q \in Q\} \vdash_{G\cup R'} \sigma(s) \cup c$. Analogously, for every $Q'/s' \in R'$, it holds that $\{\sigma(q') \cup c \mid q' \in Q'\} \vdash_{G\cup R} \sigma(s') \cup c$.

"⇐": Follows by taking σ to be identity and $c = \emptyset$. □

Definition 36. An $\mathcal{L}^+_{cl}$-rule Q/s is *invertible in LK^+* if $s \vdash_{LK^+} q$ for every $q \in Q$.

All the logical rules of LK^+, i.e. the rules for the connectives $\wedge$, $\vee$ and $\supset$, are invertible in LK^+.

Semantics

The semantic framework that we use in this chapter is that of *partial non-deterministic matrices* (PNmatrices) [23]. PNmatrices are a natural generalization of ordinary multi-valued logical matrices, in which connectives can have non-deterministic and partial interpretations. This means that truth values assigned to compound formulas can be chosen non-deterministically out of a given, possibly empty[1], set of options.

Definition 37 ([23])**.** A *partial non-deterministic matrix* (*PNmatrix*) $\mathcal{M}$ for a propositional language $\mathcal{L}$ consists of:

- A set $\mathcal{V}_\mathcal{M}$ of truth values,
- a subset $\mathcal{D}_\mathcal{M} \subseteq \mathcal{V}_\mathcal{M}$ of designated truth values, and
- a truth table $\heartsuit_\mathcal{M} : {\mathcal{V}_\mathcal{M}}^n \to P(\mathcal{V}_\mathcal{M})$ for every n-ary connective $\heartsuit$ of $\mathcal{L}$.

Definition 38 ([23])**.** Let $\mathcal{M}$ be a PNmatrix for $\mathcal{L}$.

- An *$\mathcal{M}$-valuation* for $\mathcal{L}$ is a function $v : \mathcal{L} \to \mathcal{V}_\mathcal{M}$ that respects the truth tables of $\mathcal{M}$, i.e. $v(\heartsuit(\varphi_1, \ldots, \varphi_n)) \in \heartsuit_\mathcal{M}(v(\varphi_1), \ldots, v(\varphi_n))$ for every compound formula $\heartsuit(\varphi_1, \ldots, \varphi_n) \in \mathcal{L}$.
- An $\mathcal{M}$-valuation v for $\mathcal{L}$ *satisfies* (with respect to $\mathcal{M}$):
  - an $\mathcal{L}$-formula φ (denoted by $v \models_\mathcal{M} \varphi$) if $v(\varphi) \in \mathcal{D}_\mathcal{M}$;
  - a finite set Γ of $\mathcal{L}$-formulas (denoted by $v \models_\mathcal{M} \Gamma$) if $v \models_\mathcal{M} \varphi$ for every $\varphi \in \Gamma$;
  - an $\mathcal{L}$-sequent s (denoted by $v \models_\mathcal{M} s$) if either $v \models_\mathcal{M} \varphi$ for some $\mathtt{t} : \varphi \in s$, or $v \not\models_\mathcal{M} \varphi$ for some $\mathtt{f} : \varphi \in s$.

[1] The possibility of having empty sets in the matrices make PNmatrices a generalization of non-deterministic matrices (Nmatrices) [19, 20, 15].

- Given a set Γ of $\mathcal{L}$-formulas and a single $\mathcal{L}$-formula φ, $\Gamma \vdash_{\mathcal{M}} \varphi$ if for every $\mathcal{M}$-valuation v for $\mathcal{L}$: $v \models_{\mathcal{M}} \varphi$ whenever $v \models_{\mathcal{M}} \Gamma$.
- Given an $\mathcal{L}$-sequent s, $\vdash_{\mathcal{M}} s$ if $v \models_{\mathcal{M}} s$ for every $\mathcal{M}$-valuation v for $\mathcal{L}$.

Note that every ordinary matrix can be identified with a PNmatrix. All truth tables of the PNmatrix contain only singletons, see e.g. the following example.

Example 24. *The PNmatrix $\mathcal{M}_{\boldsymbol{Cl^+}}$ for the positive fragment of classical logic is defined as follows:*

- *The set of truth values $\mathcal{V}_{\mathcal{M}_{\boldsymbol{Cl^+}}} = \{\mathtt{f}, \mathtt{t}\}$,*
- *the set of designated truth values $\mathcal{D}_{\mathcal{M}_{\boldsymbol{Cl^+}}} = \{\mathtt{t}\}$, and*
- *truth tables for every connective $\diamond \in \{\wedge, \vee, \supset\}$. $\wedge_{\mathcal{M}_{\boldsymbol{Cl^+}}}$, $\vee_{\mathcal{M}_{\boldsymbol{Cl^+}}}$, and $\supset_{\mathcal{M}_{\boldsymbol{Cl^+}}}$ are defined according to the classical truth tables where singletons are used instead of values, e.g. $\wedge_{\mathcal{M}_{\boldsymbol{Cl^+}}}(\mathtt{t}, \mathtt{f}) = \{\mathtt{f}\}$.*

Note that $\mathcal{M}_{\boldsymbol{Cl^+}}$ is sound and complete for HCL^+ (i.e. $\Gamma \vdash_{HCL^+} \varphi$ iff $\Gamma \vdash_{\mathcal{M}_{\boldsymbol{Cl^+}}} \varphi$), as well as for LK^+ (i.e. $\vdash_{LK^+} s$ iff $\vdash_{\mathcal{M}_{\boldsymbol{Cl^+}}} s$) [12].

PNmatrices can be used to provide a decision procedure for the logics they characterize, see the following result established in [23]:

Proposition 3. *Let $\mathcal{M}$ be a finite PNmatrix for a propositional language $\mathcal{L}$. Given an $\mathcal{L}$-sequent s, the question whether $\vdash_{\mathcal{M}} s$ is decidable. Similarly, given a finite set $\Gamma \cup \{\varphi\}$ of $\mathcal{L}$-formulas, the question whether $\Gamma \vdash_{\mathcal{M}} \varphi$ is decidable.*

Examples of Paraconsistent Logics

Logics of Formal Inconsistency

The *Logics of Formal Inconsistency* (LFIs) [63, 47, 45] are one of the most important classes of paraconsistent logics. A well-known subclass is the family of C-systems [63, 47, 45, 13, 18], where the notion of consistency is internalized in the object language by a unary consistency operator $\circ$. $\circ\varphi$ has the intuitive meaning that "φ is consistent". Below we will present the definitions of some LFIs (and in particular, C-systems) which are described by Hilbert systems extending the positive fragment of classical propositional logic with axioms of Table 6.2.

Logic	Axiomatization*	Remark
B, **mbC**	**Cl**$^+$+(**n₁**) + (**b**)	**B** is the basic paraconsistent logic. It is called **mbC** in [47, 45].
BK	**B** +(**k**)	**BK** is also considered a basic paraconsistent logic [16, 17, 18].
bC	**B**+(**c**)	**bC** is the basic logic considered in [47].

(n$_1$)	$\varphi \vee \neg\varphi$	**(n$_2$)**	$\varphi \supset (\neg\varphi \supset \psi)$	**(k)**	$\circ\varphi \vee (\varphi \wedge \neg\varphi)$
(r$_\diamond$)	$\circ(\varphi \diamond \psi) \supset (\circ\varphi \vee \circ\psi)$	**(b)**	$\varphi \supset (\neg\varphi \supset (\circ\varphi \supset \psi))$	**(o$^1_\diamond$)**	$\circ\varphi \supset \circ(\varphi \diamond \psi)$
(o$^2_\diamond$)	$\circ\psi \supset \circ(\varphi \diamond \psi)$	**(i)**	$\neg \circ \varphi \supset (\varphi \wedge \neg\varphi)$	**(c)**	$\neg\neg\varphi \supset \varphi$
(a$_\neg$)	$\circ\varphi \supset \circ\neg\varphi$	**(a$_\diamond$)**	$(\circ\varphi \wedge \circ\psi) \supset \circ(\varphi \diamond \psi)$	**(e)**	$\varphi \supset \neg\neg\varphi$
(o$^l_\supset$)	$\neg(\varphi \supset \psi) \supset (\varphi \wedge \neg\psi)$	**(o$^r_\supset$)**	$(\varphi \wedge \neg\psi) \supset \neg(\varphi \supset \psi)$	**(l)**	$\neg(\varphi \wedge \neg\varphi) \supset \circ\varphi$
(o$^r_\wedge$)	$(\neg\varphi \vee \neg\psi) \supset \neg(\varphi \wedge \psi)$	**(o$^l_\wedge$)**	$\neg(\varphi \wedge \psi) \supset (\neg\varphi \vee \neg\psi)$	**(d)**	$\neg(\neg\varphi \wedge \varphi) \supset \circ\varphi$
(o$^l_\vee$)	$\neg(\varphi \vee \psi) \supset (\neg\varphi \wedge \neg\psi)$	**(o$^r_\vee$)**	$(\neg\varphi \wedge \neg\psi) \supset \neg(\varphi \vee \psi)$		
(i$^l_\wedge$)	$\neg \circ (\varphi \wedge \psi) \supset (\neg \circ \varphi \wedge \psi) \vee (\neg \circ \psi \wedge \varphi))$				
(i$^r_\wedge$)	$(\neg \circ \varphi \wedge \psi) \vee (\neg \circ \psi \wedge \varphi)) \supset \neg \circ (\varphi \wedge \psi)$				
(i$^l_\vee$)	$\neg \circ (\varphi \vee \psi) \supset (\neg \circ \varphi \wedge \neg\psi) \vee (\neg \circ \psi \wedge \neg\varphi))$				
(i$^r_\vee$)	$(\neg \circ \varphi \wedge \neg\psi) \vee (\neg \circ \psi \wedge \neg\varphi)) \supset \neg \circ (\varphi \vee \psi)$				
(i$^l_\supset$)	$\neg \circ (\varphi \supset \psi) \supset (\varphi \wedge \neg \circ \psi))$				
(i$^r_\supset$)	$(\varphi \wedge \neg \circ \psi)) \supset \neg \circ (\varphi \supset \psi)$				

Table 6.2: (Schematic) Hilbert axioms defining paraconsistent logics ($\diamond \in \{\vee, \wedge, \supset\}$)

Ci	**bC** + **(i)** **mCi** + **(c)**	In **Ci**, the notions of inconsistency and contradiction coincide [47].
C$_1$	**bC** + **(a$_\diamond$)** **BK** + **(c)**,**(l)**,**(i)**,**(a$_\diamond$)**	**C$_1$** is Da Costa's first paraconsistent logic [63].
LFI1	**Ci** + **(e)**, **(i$^l_\diamond$)**, **(i$^r_\diamond$)**	**LFI1** is a three-valued maximal paraconsistent logic [47].
* see Table 6.2 for the axioms.		

Other Paraconsistent Logics

Discussive Logic. Introduced by Jaśkowski [101], *discussive logic* **D$_2$**, was one of the first paraconsistent logics. The basic idea of this logic (and hence, also its name) comes from the fact that in a discussion, different opinions might lead to inconsistent information: $\neg\varphi$ and φ can be true at the same time. In discussive logic, this is reflected by blocking the rule of adjunction, i.e., $\varphi, \neg\varphi \not\vdash \varphi \wedge \neg\varphi$. A similar non-adjunctive approach was suggested in [153]. Note that the logic **D$_2$** is, in fact, a logic of formal inconsistency [45].

Many-valued Logics. Some *many-valued logics* are also paraconsistent. For example, the logic of paradox **LP** [7, 151] has three (instead of the classical two) truth values and formulas can either be "true", "false", or "both" (and "true", as well as "both" are the designated truth values). Other widely studied many-valued paraconsistent logics

$(\mathbf{e}\omega^l_{\diamond_i})$	$\sim^i(\varphi\diamond\psi)\supset(\sim^i\varphi\diamond\sim^i\psi)$	$(\mathbf{e}\omega^r_{\diamond_i})$	$(\sim^i\varphi\diamond\sim^i\psi)\supset\sim^i(\varphi\diamond\psi)$
$(\mathbf{o}\omega^l_{\supset_j})$	$\sim^j(\varphi\supset\psi)\supset(\sim^{j-1}\varphi\wedge\sim^j\psi)$	$(\mathbf{o}\omega^r_{\supset_j})$	$(\sim^{j-1}\varphi\wedge\sim^j\psi)\supset\sim^j(\varphi\supset\psi)$
$(\mathbf{o}\omega^l_{\wedge_j})$	$\sim^j(\varphi\wedge\psi)\supset(\sim^j\varphi\vee\sim^j\psi)$	$(\mathbf{o}\omega^r_{\wedge_j})$	$(\sim^j\varphi\vee\sim^j\psi)\supset\sim^j(\varphi\wedge\psi)$
$(\mathbf{o}\omega^l_{\vee_j})$	$\sim^j(\varphi\vee\psi)\supset(\sim^j\varphi\wedge\sim^j\psi)$	$(\mathbf{o}\omega^r_{\vee_j})$	$(\sim^j\varphi\wedge\sim^j\psi)\supset\sim^j(\varphi\vee\psi)$
$(\mathbf{pc}_k)$	$\sim^k(\neg\varphi)\supset\neg(\sim^k\varphi)$	$(\mathbf{cp}_k)$	$\neg(\sim^k\varphi)\supset\sim^k(\neg\varphi)$
$(\mathbf{cc}^l_n)$	$\sim^2\sim^n\varphi\supset\sim^n\varphi$	$(\mathbf{cc}^r_n)$	$\sim^n\varphi\supset\sim^2\sim^n\varphi$

Table 6.3: Hilbert axioms provable in the logics $\mathbf{L}_\omega$ [105] or $\mathbf{L}2^{n+2}$ [106], $\diamond\in\{\supset,\wedge,\vee\}$

are Belnap's and Dunn's four-valued logic **B4** [72, 33], Nelson's paraconsistent logic **N4** [3], first-degree entailment **FDE** [4], logics of logical bilattices [6] or Shramko-Wansing's 16-valued logic [156].

In [105, 106], the paraconsistent many-valued logics $\mathbf{L}_\omega$ and $\mathbf{L}2^{n+2}$ are presented: $\mathbf{L}_\omega$ [105] combines classical ($\neg$) and paraconsistent ($\sim$) negations. It is axiomatized over $\mathbf{Cl}^+$ with the axioms

$$(\mathbf{cp_k}),(\mathbf{pc_k}),(\mathbf{e}\omega^{\mathbf{l}}_{\diamond_\mathbf{i}}),(\mathbf{e}\omega^{\mathbf{r}}_{\diamond_\mathbf{i}}),(\mathbf{o}\omega^{\mathbf{l}}_{\supset_\mathbf{j}}),(\mathbf{o}\omega^{\mathbf{r}}_{\supset_\mathbf{j}}),(\mathbf{o}\omega^{\mathbf{l}}_{\wedge_\mathbf{j}}),(\mathbf{o}\omega^{\mathbf{r}}_{\wedge_\mathbf{j}}),(\mathbf{o}\omega^{\mathbf{l}}_{\vee_\mathbf{j}}),(\mathbf{o}\omega^{\mathbf{r}}_{\vee_\mathbf{j}})$$

depicted in Table 6.3, for any even $i\in\omega$, any odd $j\in\omega$ and any $k\in\omega$ (where ω represents the set of natural numbers). $\mathbf{L}2^{n+2}$ [106] contains weak double negation axioms. It is axiomatized over $\mathbf{Cl}^+$ by the Hilbert axioms

$$(\mathbf{cc^l_n}),(\mathbf{cc^r_n}),(\mathbf{e}\omega^{\mathbf{l}}_{\diamond_\mathbf{i}}),(\mathbf{e}\omega^{\mathbf{r}}_{\diamond_\mathbf{i}}),(\mathbf{o}\omega^{\mathbf{l}}_{\supset_\mathbf{j}}),(\mathbf{o}\omega^{\mathbf{r}}_{\supset_\mathbf{j}}),(\mathbf{o}\omega^{\mathbf{l}}_{\wedge_\mathbf{j}}),(\mathbf{o}\omega^{\mathbf{r}}_{\wedge_\mathbf{j}}),(\mathbf{o}\omega^{\mathbf{l}}_{\vee_\mathbf{j}}),(\mathbf{o}\omega^{\mathbf{r}}_{\vee_\mathbf{j}})$$

depicted in Table 6.3 for any even $i\in\omega^{n+1}$ and any odd $j\in\omega^{n+1}$ (ω^k denotes the set $\{l\in\omega\mid l\leq k\}$).

Relevance Logics. Another important type of paraconsistent logics are *relevance logics* [4, 73]. In these logics a prerequisite for a valid argument is a (relevant) connection between premises and conclusion, and hence $\psi,\neg\psi\not\vdash\varphi$. The main relevance logics are the logic of relevant entailment **E** [4], and the logic of relevant implication **R** [4].

Adaptive Logics. Introduced by Batens [30, 31], an *adaptive logic* "adapts itself to the specific premise set to which it is applied" [32]. An adaptive logic uses two logical systems: an "upper limit logic" (ULL), e.g. classical logic, and a "lower limit logic" (LLL), which is weaker than the ULL, e.g. some paraconsistent logic. Reasoning in an adaptive logic is done by switching between these two logics. Depending on the premises, either the rules from the ULL or LLL are applied in a proof. For example, when the premises contain a contradiction, the LLL is used, while in a "non-problematic" case, rules of the ULL are applied.

$\varphi \Rightarrow \varphi$	$\dfrac{\varphi, \Gamma \Rightarrow \Delta, \psi}{\Gamma \Rightarrow \Delta, \varphi \supset \psi}\ (\supset, r)$	$\dfrac{\Gamma \Rightarrow \Delta, \varphi \qquad \psi, \Gamma \Rightarrow \Delta}{\varphi \supset \psi, \Gamma \Rightarrow \Delta}\ (\supset, l)$
$\dfrac{\Gamma \Rightarrow \Delta}{\varphi, \Gamma \Rightarrow \Delta}\ (w, l)$	$\dfrac{\Gamma \Rightarrow \Delta, \varphi, \psi}{\Gamma \Rightarrow \Delta, \varphi \vee \psi}\ (\vee, r)$	$\dfrac{\varphi, \Gamma \Rightarrow \Delta \qquad \psi, \Gamma \Rightarrow \Delta}{\varphi \vee \psi, \Gamma \Rightarrow \Delta}\ (\vee, l)$
$\dfrac{\Gamma \Rightarrow \Delta}{\varphi, \Gamma \Rightarrow \Delta}\ (w, l)$	$\dfrac{\varphi, \psi, \Gamma \Rightarrow \Delta}{\varphi \wedge \psi, \Gamma \Rightarrow \Delta}\ (\wedge, l)$	$\dfrac{\Gamma \Rightarrow \Delta, \varphi \qquad \Gamma \Rightarrow \Delta, \psi}{\Gamma \Rightarrow \Delta, \varphi \wedge \psi}\ (\wedge, r)$
$\dfrac{\varphi, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta, \neg\varphi}\ (\neg, r)$	$\dfrac{\varphi, \neg\varphi, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta, \circ\varphi}\ (\circ, r)$	$\dfrac{\Gamma \Rightarrow \Delta, \varphi \qquad \Gamma \Rightarrow \Delta, \neg\varphi}{\circ\varphi, \Gamma \Rightarrow \Delta}\ (\circ, l)$
$\dfrac{\Gamma \Rightarrow \Delta, \varphi \qquad \varphi, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}\ (cut)$		

Table 6.4: Sequent calculus G_k for **BK**

6.2 Related Work in Proof Theory

We give a brief overview of related work that focuses on the introduction of analytic calculi for paraconsistent logics.

Most of the existing results are however tailored to the specific (class of) logics at hand and do not provide hints for an automated introduction of analytic calculi for them, see e.g. [6, 104, 69, 70, 45, 105, 107, 108, 106].

In contrast to these logic-tailored approaches, a modular procedure introducing analytic calculi and semantics using the framework of Nmatrices for the logics of formal inconsistency is introduced in [14, 16, 17, 18]. The procedure in [16, 17] works for C-systems which are obtained by extending **BK** with any axiom contained in Table 6.2 except for **(l)**, **(d)**, $(\mathbf{i}^l_\diamond)$ and $(\mathbf{i}^r_\diamond)$. This method consists of the following two steps (note that they cannot be done in a fully automated way):

(Step 1): A finitely-valued non-deterministic matrix (Nmatrix) $\mathcal{M}$ is introduced by extracting semantic conditions from the axioms extending **BK** in each specific logic. The semantic conditions are then used to refine the standard Nmatrix for **BK**, i.e., they reduce the level of non-determinism in the Nmatrix.

(Step 2): A cut-free sequent calculus is constructed based on the finitely-valued Nmatrix $\mathcal{M}$ created in the first step. This is achieved by the algorithm in [14] that works in two stages: (i) Every entry of every truth table of the Nmatrix is translated into rules of the calculus. In the case of the Nmatrix $\mathcal{M}$ for (extensions of) **BK** this step can be simplified by translating the semantic conditions of each axiom from *(Step 1)* in a modular way into corresponding rules using the six facts mentioned in the example below[2]. (ii) When necessary, the resulting rules are then combined and simplified.

The sequent rules obtained in *(Step 2)* are then added to the sequent calculus G_k for **BK** depicted in Table 6.4.

[2]Note that the obtained rules are equivalent.

Example 25 ([16]). *We assume truth values of the form $\langle x, y\rangle$ where $x = 1$ iff φ is "true", and $y = 1$ iff $\neg\varphi$ is "true". Hence there are four possible truth values* $\mathtt{t} = \langle 1, 0\rangle, \mathtt{f} = \langle 0, 1\rangle$, $\top = \langle 1, 1\rangle$ *and* $\bot = \langle 0, 0\rangle$.

The logic **BK** *is characterized by the Nmatrix* $\mathcal{M} = (\{\mathtt{t}, \mathtt{f}, \top\}, \{\mathtt{t}, \top\}, \mathcal{T})$, *where* $\{\mathtt{t}, \top\}$ *are the designated truth values. Note that the fourth truth value* $\bot$ *is excluded by the axiom* $(\mathbf{n_1})\varphi \vee \neg\varphi$ *contained in* **BK**. *The non-deterministic truth tables* $\mathcal{T}$ *of* $\mathcal{M}$ *are defined as follows:*

p	$\neg p$	$\circ p$
t	{f}	{t, ⊤}
⊤	{t, ⊤}	{f}
f	{t, ⊤}	{t, ⊤}

∧	t	⊤	f
t	{t, ⊤}	{t, ⊤}	{f}
⊤	{t, ⊤}	{t, ⊤}	{f}
f	{f}	{f}	{f}

⊃	t	⊤	f
t	{t, ⊤}	{t, ⊤}	{f}
⊤	{t, ⊤}	{t, ⊤}	{f}
f	{t, ⊤}	{t, ⊤}	{t, ⊤}

∨	t	⊤	f
t	{t, ⊤}	{t, ⊤}	{t, ⊤}
⊤	{t, ⊤}	{t, ⊤}	{t, ⊤}
f	{t, ⊤}	{t, ⊤}	{f}

Let us consider the extension of **BK** *with the axiom* $(\mathbf{o^1_\vee}) \circ\varphi \supset \circ(\varphi \vee \psi)$ *to show how this axiom is transformed into an equivalent set of sequent rules.*

(Step 1) We first extract the semantic conditions from $(\mathbf{o^1_\vee})$. *Let v be a valuation in* $\mathcal{M}$. *We must ensure that* $v(\circ\varphi \supset \circ(\varphi \vee \psi)) \in \{\mathtt{t}, \top\}$:

1. *v satisfies* $(\mathbf{o^1_\vee})$, *if* $v(\circ\varphi) = \mathtt{f}$ *(then the implication is true, i.e.,* t *or* $\top$*). Hence it is satisfied if* $v(\varphi) = \top$, *since* $v(\circ\top) = \mathtt{f}$.
2. *Otherwise,* $v(\circ(\varphi \vee \psi))$ *must be true. We have to ensure that* $v(\circ(\varphi \vee \psi)) \neq \mathtt{f}$, *i.e.* $v(\varphi \vee \psi) \neq \top$:
   a) *If* $v(\varphi) = v(\psi) = \mathtt{f}$, $v(\circ(\varphi \vee \psi)) \neq \mathtt{f}$ *and hence v satisfies* $(\mathbf{o^1_\vee})$.
   b) *Otherwise, we know that* $v(\varphi \vee \psi)$ *will be chosen non-deterministically out of* $\{\mathtt{t}, \top\}$. *Thus we have to ensure* $v(\varphi \vee \psi) = \mathtt{t}$ *by defining the following five conditions:*
   (Case $\varphi = \mathtt{t}$*):* $\mathtt{t} \vee \mathtt{t} = \mathtt{t} \vee \mathtt{f} = \mathtt{t} \vee \top = \mathtt{t} \vee x = \{\mathtt{t}\}$ *and*
   (Case $\varphi = \mathtt{f}$*):* $\mathtt{f} \vee \mathtt{t} = \mathtt{f} \vee \top = \{\mathtt{t}\}$.

Now the truth table for $\vee$ *needs to be refined according to these five semantic conditions. We obtain the following new truth table for* $\vee$:

∨	t	⊤	f
t	{t}	{t}	{t}
⊤	{t, ⊤}	{t, ⊤}	{t, ⊤}
f	{t}	{t}	{f}

(Step 2) We show how to generate the sequent rules by analyzing the semantic conditions using the following six facts:

1. $v(\varphi) = \mathtt{t}$ *iff* $\neg\varphi \Rightarrow$ *is true in v.*
2. $v(\varphi) = \mathtt{f}$ *iff* $\varphi \Rightarrow$ *is true in v.*

3. $v(\varphi) = \top$ *iff* $\Rightarrow \varphi$ *and* $\Rightarrow \neg\varphi$ *are both true in* v*.*
4. $v(\varphi) \in \{\mathtt{f}, \top\}$ *iff* $\Rightarrow \neg\varphi$ *is true in* v*.*
5. $v(\varphi) \in \{\mathtt{t}, \top\}$ *iff* $\Rightarrow \varphi$ *is true in* v*.*
6. $v(\varphi) \in \{\mathtt{t}, \mathtt{f}\}$ *iff* $\varphi, \neg\varphi \Rightarrow$ *is true in* v*.*

We can now "read off" the sequent rules that arise from our semantic conditions. We start by analyzing the first condition of the form $\varphi \vee \psi$, $\mathtt{t} \vee x = \{\mathtt{t}\}$*, and match it with the facts described above. The premise of the new rule is obtained by considering fact 1 above:* $v(\varphi) = \mathtt{t}$ *iff* $\neg\varphi \Rightarrow$ *is true in* v*. To obtain the conclusion of our rule, we have to analyze the whole formula:* $v(\varphi \vee \psi) = \mathtt{t}$ *iff* $\neg(\varphi \vee \psi) \Rightarrow$ *is true in* v*. Thus we obtain the rule:*

$$\frac{\neg\varphi, \Gamma \Rightarrow \Delta}{\neg(\varphi \vee \psi), \Gamma \Rightarrow \Delta}$$

Analogously, we analyze the second semantic condition, which is also of the form $\varphi \vee \psi$*:* $\mathtt{f} \vee \mathtt{t} = \mathtt{f} \vee \top = \{\mathtt{t}\}$*. Considering fact 2 above, we know that* $v(\varphi) = \mathtt{f}$ *iff* $\varphi \Rightarrow$ *is true in* v*. Moreover, from fact 5, we know that* $v(\psi) \in \{\mathtt{t}, \top\}$ *iff* $\Rightarrow \psi$ *is true in* v*. These two facts give us the two premises of our new rule. The conclusion is again obtained by fact 1, since* $v(\varphi \vee \psi) = \mathtt{t}$ *iff* $\neg(\varphi \vee \psi) \Rightarrow$ *is true in* v*:*

$$\frac{\varphi, \Gamma \Rightarrow \Delta \qquad \Gamma \Rightarrow \Delta, \psi}{\neg(\varphi \vee \psi), \Gamma \Rightarrow \Delta}$$

A cut-free sequent calculus for the logic $\mathbf{BK}$+$(\mathbf{o}^{\mathbf{1}}_{\vee})$ *is obtained by extending the calculus* G_k *with the two rules stated above.*

The method proposed in [16, 18] works for logics that have a semantic characterization in terms of finitely-valued Nmatrices. However, some important LFIs can only be characterized by infinitely-valued Nmatrices, e.g. logics including the axioms **(l)** or **(d)** from Table 6.2, such as $\mathbf{C}_1$ (see page 104). [17] extends the procedure of [16, 18] to extract cut-free sequent calculi for these logics by adjusting the two-step procedure as follows:

(Step 1): This step basically remains the same except for the fact that the Nmatrix requires an infinite number of truth values — thus, the three truth values for **BK** of the original procedure ($\mathtt{t}, \mathtt{f}, \top$) are replaced with three *sets* of truth values.

(Step 2): Since the algorithm proposed in [14] cannot be applied, as it does not work for infinitely-valued Nmatrices, the corresponding analytic sequent calculus is obtained by translating the semantic conditions of each axiom into a sequent rule using six facts similar to those introduced for the finitely-valued case.

Note however, that the procedure in [16, 17, 18] is not fully automated, since the construction of the semantics has to be done manually and requires some ingenuity.

6.3 Towards Analytic Calculi for Paraconsistent Logics

We describe a procedure to generate sequent calculi and semantics in terms of PNmatrices. Our procedure can be applied to a large class of paraconsistent logics formulated as Hilbert systems. The logics we can deal with are obtained by extending the positive fragment of classical propositional logic $\mathbf{Cl}^+$ with axioms of a certain shape. More precisely, these logics are induced by a family $\mathbb{H}$ of Hilbert calculi that are obtained by (i) extending the language of $\mathbf{Cl}^+$ with finitely many unary connectives, and (ii) adding to HCL^+ axioms over the extended language of a certain general form.

Our procedure is fully automated and works in two steps:

(Step 1) We adapt the systematic procedure of [52] (see Section 4.2) to transform the Hilbert axioms defining $H \in \mathbb{H}$ into sequent calculus rules. In contrast to the method in [52], the rules that we generate are *logical rules* in Gentzen's terminology, i.e., they introduce logical connectives. Since the new rules may introduce more than one connective, the analyticity of the calculus depends on the interaction between all (new and existing) rules involving the same connectives. This requires a "*global view*" on the obtained calculus, which is provided by the semantics introduced in step 2.

(Step 2) We extract semantics from the introduced sequent calculi using the framework of partial non-deterministic matrices (PNmatrices). The semantics allows us to reason about the analyticity of the obtained calculus. We show that if the PNmatrix constructed for H is an Nmatrix (i.e., it has no empty spot in the truth tables) then the corresponding sequent calculus is analytic. Furthermore, the PNmatrix guarantees that the system H is decidable.

With this procedure, we can automatically create sequent calculi and new semantic foundations for many well-studied logics, as well as infinitely many new ones. For example, the method works for classical logic, many C-systems [63, 45, 46] or the paraconsistent logics investigated in [106]. While analytic sequent calculi and/or adequate semantics for some of these logics were already available, the main feature of our approach is its *full automation*. Moreover, it is not tailored to the C-systems (as [17, 18]) or to other specific logics (as [106]).

Note that our procedure reverses the two steps of the method in [16, 17, 18], where first an Nmatrix is constructed for each system and then it is used for introducing a corresponding analytic sequent calculus. Moreover, in contrast to (almost) all existing work on paraconsistent logics, and in particular [16, 17, 18], it is fully automated.

6.4 Step 1: Automated Generation of Sequent Calculi

We consider logics in the language $\mathcal{L}_{\mathbf{Cl}^+}$ extended with new unary connectives from $\mathcal{U}_{\mathcal{L}}$. Our algorithm transforms Hilbert axioms of a special form into equivalent logical sequent rules. The transformation procedure once again utilizes the two key ingredients:

(1) the *invertible rules* of the base calculus LK^+ to decompose the axiom, and
(2) the *Ackermann lemma*:

Lemma 14. *Let G be a sequent calculus for $\mathcal{L}$ extending LK^+, and let*

$$r = \emptyset/\{\mathtt{b}_1 : \varphi_1, \ldots, \mathtt{b}_n : \varphi_n\} \qquad \text{and} \qquad r' = \{\{\widehat{\mathtt{b}_2} : \varphi_2\}, \ldots, \{\widehat{\mathtt{b}_n} : \varphi_n\}\}/\{\mathtt{b}_1 : \varphi_1\}$$

be $\mathcal{L}$-rules where $\widehat{\mathtt{f}} = \mathtt{t}$ and $\widehat{\mathtt{t}} = \mathtt{f}$. Then $\{r\}$ and $\{r'\}$ are equivalent in G.

Proof. "$\Rightarrow$": To show

$$\{\{\widehat{\mathtt{b}_2} : \varphi_2\}, \ldots, \{\widehat{\mathtt{b}_n} : \varphi_n\}\} \vdash_{G \cup \{r\}} \{\mathtt{b}_1 : \varphi_1\}$$

we use an application of r, weakenings and $n-1$ cuts:

$$\dfrac{\dfrac{r\ \dfrac{\emptyset}{\{\mathtt{b}_1 : \varphi_1, \ldots, \mathtt{b}_n : \varphi_n\}} \qquad \dfrac{\{\widehat{\mathtt{b}_n} : \varphi_n\}}{\{\mathtt{b}_1 : \varphi_1, \ldots, \mathtt{b}_{n-1} : \varphi_{n-1}, \widehat{\mathtt{b}_n} : \varphi_n\}}\,(\mathtt{b}_i : W)_{1 \leq i < n}}{\{\mathtt{b}_1 : \varphi_1, \ldots, \mathtt{b}_{n-1} : \varphi_{n-1}\}}\,(cut) \qquad \vdots}{\vdots}\,(cut)$$

$$\{b_1 : \varphi_1\}$$

"$\Leftarrow$": For the other direction, i.e.

$$\vdash_{G \cup \{r'\}} \{\mathtt{b}_1 : \varphi_1, \ldots, \mathtt{b}_n : \varphi_n\}$$

we use (id), weakenings and r':

$$\dfrac{\dfrac{\{\widehat{\mathtt{b}_2} : \varphi_2, \mathtt{b}_2 : \varphi_2\}}{\{\widehat{\mathtt{b}_2} : \varphi_2, \mathtt{b}_2 : \varphi_2, \ldots, \mathtt{b}_n : \varphi_n\}}\,(\mathtt{b}_i : W)_{2 < i \leq n} \quad \ldots \quad \dfrac{\{\widehat{\mathtt{b}_n} : \varphi_n, \mathtt{b}_n : \varphi_n\}}{\{\mathtt{b}_2 : \varphi_2, \ldots, \mathtt{b}_n : \varphi_n, \widehat{\mathtt{b}_n} : \varphi_n\}}\,(\mathtt{b}_i : W)_{2 \leq i < n}}{\{\mathtt{b}_1 : \varphi_1, \mathtt{b}_2 : \varphi_2, \ldots, \mathtt{b}_n : \varphi_n\}}\,r'$$

□

We define a grammar for the class of axioms that can be handled with our procedure. Our procedure works for Hilbert systems that are defined by extending HCL^+ with axioms of the set $\mathbf{Ax}_{\mathcal{L}}$:

Definition 39. $\mathbf{Ax}_{\mathcal{L}}$ is the set of $\mathcal{L}$-formulas that:

1. are generated by the following grammar (I is the initial variable):

$$\begin{array}{ll} I = R_1 \mid R_2 & P_1 = (P_1 \diamond P_1) \mid \bar{\star} p_1 \mid p_1 \mid p_2 \\ R_1 = (R_1 \diamond P_1) \mid (P_1 \diamond R_1) \mid \bar{\star} p_1 & P_2 = (P_2 \diamond P_2) \mid \bar{\star} p_1 \mid p_1 \mid \bar{\star} p_2 \mid p_2 \\ R_2 = (R_2 \diamond P_2) \mid (P_2 \diamond R_2) \mid \bar{\star}(p_1 \diamond p_2) & \text{for } \diamond \in \{\wedge, \vee, \supset\}, \bar{\star} \in \mathcal{U}^*_{\mathcal{L}} \setminus \{\epsilon\} \end{array}$$

2. and satisfy the following conditions: for some subformula $\varphi = \bar{\star} p_1$ of an $\mathcal{L}$-formula arising from the start symbol R_1 (and for the subformula $\varphi = \bar{\star}(p_1 \diamond p_2)$ of an $\mathcal{L}$-formula arising from R_2, resp.): φ must not be contained
   (a) in a positively[3] occurring (sub)formula of the form $\psi_1 \wedge \psi_2$, and
   (b) in a negatively occurring (sub)formula of the form $\psi_1 \vee \psi_2$ or $\psi_1 \supset \psi_2$.

Roughly speaking, the axioms[4] in $\mathbf{Ax}_{\mathcal{L}}$ contain

(R_1) at least one propositional variable p_1 prefixed with a non-empty sequence of connectives from $\mathcal{U}_{\mathcal{L}}$ and possibly the propositional variables p_1, p_2, or
(R_2) exactly one formula $(p_1 \diamond p_2)$ prefixed with a non-empty sequence of connectives from $\mathcal{U}_{\mathcal{L}}$ and possibly the propositional variables p_1, p_2, possibly prefixed with sequences of connectives from $\mathcal{U}_{\mathcal{L}}$.

Example 26. *The following axioms are covered by the grammar in Definition 39:*

$$\begin{aligned} p_1 \vee \neg p_1 & \qquad (\mathbf{n_1}) \\ \circ p_1 \supset \circ\neg p_1 & \qquad (\mathbf{a}_\neg) \\ \neg \circ p_1 \supset (p_1 \wedge \neg p_1) & \qquad (\mathbf{i}) \\ \neg(p_1 \vee p_2) \supset (\neg p_1 \wedge \neg p_2) & \qquad (\mathbf{o}^{\mathbf{l}}_\vee) \\ (\neg \circ p_1 \wedge \neg p_2) \vee (\neg \circ p_2 \wedge \neg p_1)) \supset \neg \circ (p_1 \vee p_2) & \qquad (\mathbf{i}^{\mathbf{r}}_\vee) \end{aligned}$$

Note that these axioms are the ones depicted in Table 6.2. In general, all axioms from Table 6.2 fall into the class $\mathbf{Ax}_{\mathcal{L}}$ *with the exception of the following two:*

$$\begin{aligned} \neg(p_1 \wedge \neg p_1) \supset \circ p_1 & \qquad (\mathbf{l}) \\ \neg(\neg p_1 \wedge p_1) \supset \circ p_1 & \qquad (\mathbf{d}) \end{aligned}$$

Definition 40. $\mathbb{H}$ is the family of Hilbert calculi obtained by extending HCL^+ with any finite set of axioms from $\mathbf{Ax}_{\mathcal{L}}$ for some language $\mathcal{L}$.

$\mathbb{H}$ includes many well-known Hilbert calculi, e.g.:

- the standard calculus for (propositional) classical logic **Cl** (that is obtained by adding the axioms ($\mathbf{n_1}$) and ($\mathbf{n_2}$) from Table 6.2, page 104 to HCL^+).
- the Hilbert calculi for the logics **B**, **BK**, **bC**, **Ci**, $\mathbf{C}_1$ or **LFI1** from page 104.
- the Hilbert calculi for other C-systems that are defined by adding to HCL^+ the axioms (**b**) and ($\mathbf{n_1}$), as well as different subsets of the other axioms (except **(l)** and **(d)**) from Table 6.2.

[3]Recall that a subformula φ occurs *negatively* (*positively*, resp.) in an $\mathcal{L}$-formula ψ if there is an odd (even, resp.) number of implications $\supset$ in ψ having φ as a subformula of its antecedent, see e.g., [44].

[4]Note that the formulas in $\mathbf{Ax}_{\mathcal{L}}$ can actually be interpreted as *axiom schemas* in which p_1, p_2 are replaced with metavariables φ, ψ that can be substituted by any $\mathcal{L}$-formula in the instances of the schema. To make the presentation of the semantics simpler, we keep using p_1, p_2 in the following sections.

- the Hilbert calculi for the logics $\mathbf{L2}^{n+2}$ for each $n \geq 0$ discussed in [106], which are obtained by adding to HCL^+ the axioms depicted in Table 6.3 (except $(\mathbf{cp}_k)$ and $(\mathbf{pc}_k)$).

For every Hilbert system $H \in \mathbb{H}$, we define a set Θ_H containing prefixes of the sequences of the unary connectives to keep track of the unary connectives occurring in H. The meaning of this set will become clear in the second step of our procedure. In short, Θ_H will determine the number and the shape of the truth values for (the PNmatrix of) H.

Definition 41. For an $\mathcal{L}$-formula ψ, let Θ_ψ denote the set of all prefixes (including the empty one ϵ) of the maximal sequences of connectives from $\mathcal{U}_\mathcal{L}$ that occur in ψ. For $H \in \mathbb{H}$, $\Theta_H = \bigcup_{\psi \in H} \Theta_\psi$.

Example 27. *Let* $\mathcal{U}_\mathcal{L} = \{\circ, \star, \neg\}$. Θ_ψ *is as follows:*

$$\psi_1 = \star \circ \neg p_1 \supset p_1 \qquad \Theta_{\psi_1} = \{\epsilon, \star, \star\circ, \star \circ \neg\}$$
$$\psi_2 = \star\star\,(\star \circ p_1 \vee \neg \circ p_2) \supset \neg p_1 \qquad \Theta_{\psi_2} = \{\epsilon, \star, \neg, \star\star, \star\circ, \neg\circ\}$$

From axioms to logical sequent rules

We now show how to construct a sequent calculus G_H that is equivalent (in the sense of Definition 7) to a given Hilbert system $H \in \mathbb{H}$. The idea is to transform the axioms of H belonging to $\mathbf{Ax}_\mathcal{L}$ into equivalent logical rules and add the obtained rules to LK^+.

Given any axiom $\varphi \in \mathbf{Ax}_\mathcal{L}$, our transformation procedure roughly works as follows:

(i) We start from the rule $r_0 = \emptyset/\{\mathtt{t} : \varphi\}$. By utilizing the invertibility of the logical rules of LK^+ as much as possible, we obtain a set of rules R equivalent to $\{r_0\}$, where each $r \in R$ has the form $\emptyset/\{\mathtt{b}_1 : \varphi_1, \ldots, \mathtt{b}_n : \varphi_n\}$ with $\mathtt{b}_i \in \{\mathtt{t}, \mathtt{f}\}$. Note that due to the shape of φ (see Definition 39) it must be the case that each φ_i is either of the form $\bar{\star} p_1$ or $\bar{\star} p_2$ with $\bar{\star} \in \Theta_H$, and there is at most one φ_i of the form $\bar{\star}(p_1 \diamond p_2)$ for $\bar{\star} \in \Theta_H \setminus \{\epsilon\}$.

(ii) Next, we remove each rule $r \in R$ whose conclusion contains $\{\mathtt{t} : p_i, \mathtt{f} : p_i\}$ for $i \in \{1, 2\}$. Moreover, for each remaining rule, if the conclusion does not contain $\bar{\star}(p_1 \diamond p_2)$ for $\bar{\star} \in \Theta_H \setminus \{\epsilon\}$, we remove all variables p_2 and use Lemma 15 below to ensure that the resulting rule is equivalent to r.

(iii) Finally, we use the Ackermann lemma. In each rule, we choose one labelled formula and move all remaining formulas but this one to the premises of the rule, changing their side of the sequent (see Lemma 14). The formula that remains in the conclusion will be the one introduced by the rule and will be either of the form $\bar{\star} p_1$ or $\bar{\star}(p_1 \diamond p_2)$ for $\bar{\star} \in \Theta_H \setminus \{\epsilon\}$.

We first give an example to illustrate the steps of the algorithm:

Example 28. *Let* φ *be the axiom* $(\mathbf{n_2})$ $p_1 \supset (\neg p_1 \supset p_2)$ *(see Table 6.2). The algorithm works as follows:*

	Rule	Application form
Θ-unary	$\mathcal{P}/\{\mathtt{t} : \bar{\star}p_1\}$	$\dfrac{\Gamma, \bar{\triangleright}_1\varphi \Rightarrow \Delta \ \dots \ \Gamma, \bar{\triangleright}_n\varphi \Rightarrow \Delta \ \Gamma \Rightarrow \bar{\bullet}_1\varphi, \Delta \ \dots \ \Gamma \Rightarrow \bar{\bullet}_m\varphi, \Delta}{\Gamma \Rightarrow \bar{\star}\varphi, \Delta}$
	$\mathcal{P}/\{\mathtt{f} : \bar{\star}p_1\}$	$\dfrac{\Gamma, \bar{\triangleright}_1\varphi \Rightarrow \Delta \ \dots \ \Gamma, \bar{\triangleright}_n\varphi \Rightarrow \Delta \ \Gamma \Rightarrow \bar{\bullet}_1\varphi, \Delta \ \dots \ \Gamma \Rightarrow \bar{\bullet}_m\varphi, \Delta}{\Gamma, \bar{\star}\varphi \Rightarrow \Delta}$
	where $\mathcal{P} = \{\{\mathtt{f} : \bar{\triangleright}_1 p_1\}, \dots, \{\mathtt{f} : \bar{\triangleright}_n p_1\}, \{\mathtt{t} : \bar{\bullet}_1 p_1\}, \dots, \{\mathtt{t} : \bar{\bullet}_m p_1\}\}$	
Θ-binary	$\mathcal{Q}/\{\mathtt{t} : \bar{\star}(p_1 \diamond p_2)\}$	$\dfrac{\Gamma, \bar{\triangleright}_1\varphi_{i_1} \Rightarrow \Delta \ \dots \ \Gamma, \bar{\triangleright}_n\varphi_{i_n} \Rightarrow \Delta \ \Gamma \Rightarrow \bar{\bullet}_1\varphi_{j_1}, \Delta \ \dots \ \Gamma \Rightarrow \bar{\bullet}_m\varphi_{j_m}, \Delta}{\Gamma \Rightarrow \bar{\star}(\varphi_1 \diamond \varphi_2), \Delta}$
	$\mathcal{Q}/\{\mathtt{f} : \bar{\star}(p_1 \diamond p_2)\}$	$\dfrac{\Gamma, \bar{\triangleright}_1\varphi_{i_1} \Rightarrow \Delta \ \dots \ \Gamma, \bar{\triangleright}_n\varphi_{i_n} \Rightarrow \Delta \ \Gamma \Rightarrow \bar{\bullet}_1\varphi_{j_1}, \Delta \ \dots \ \Gamma \Rightarrow \bar{\bullet}_m\varphi_{j_m}, \Delta}{\Gamma, \bar{\star}(\varphi_1 \diamond \varphi_2) \Rightarrow \Delta}$
	where $\mathcal{Q} = \{\{\mathtt{f} : \bar{\triangleright}_1 p_{i_1}\}, \dots, \{\mathtt{f} : \bar{\triangleright}_n p_{i_n}\}, \{\mathtt{t} : \bar{\bullet}_1 p_{j_1}\}, \dots, \{\mathtt{t} : \bar{\bullet}_m p_{j_m}\}\}$	

Table 6.5: The general form of our rules ($\bar{\star} \in \Theta \setminus \{\epsilon\}, \bar{\triangleright}_i, \bar{\bullet}_j \in \Theta, \diamond \in \{\wedge, \vee, \supset\}$, $i_1, \dots, i_n, j_1, \dots, j_m \in \{1, 2\}$)

$$
\begin{array}{rl}
\emptyset/\{\mathtt{t} : p_1 \supset (\neg p_1 \supset p_2)\} & \\
\longrightarrow^{(i)} & \emptyset/\{\mathtt{f} : p_1, \mathtt{t} : \neg p_1 \supset p_2)\} \\
\longrightarrow^{(i)} & \emptyset/\{\mathtt{f} : p_1, \mathtt{f} : \neg p_1, \mathtt{t} : p_2\} \\
\longrightarrow^{(ii)} & \emptyset/\{\mathtt{f} : p_1, \mathtt{f} : \neg p_1\} \\
\longrightarrow^{(iii)} & \{\{\mathtt{t} : p_1\}\}/\{\mathtt{f} : \neg p_1\}
\end{array}
$$

The rule that will be added to the base calculus is hence $\{\{\mathtt{t} : p_1\}\}/\{\mathtt{f} : \neg p_1\}$.

The rules that are generated by this algorithm have a certain general form due to the special format of their equivalent axiom $\varphi \in \mathbf{Ax}_{\mathcal{L}}$. Depending on the shape of the axiom, we distinguish between two types of rules, which are defined in Table 6.5:

- Θ-*unary* rules arise from axioms generated from R_1 in the grammar of Definition 39.
- Θ-*binary* rules are generated starting from R_2.

The distinction between these types of rules will be crucial for the definitions of the semantics in step 2 (see Section 6.5).

Due to this special shape of the generated rules, we call the resulting sequent calculi Θ-*simple*:

Definition 42. Let Θ be a non-empty subset of $\mathcal{U}^*_{\mathcal{L}}$ that is closed under prefixes (in particular, $\epsilon \in \Theta$). An $\mathcal{L}$-rule Q/s is called Θ-*simple* if it is either Θ-unary or Θ-binary (see Table 6.5). A sequent calculus for $\mathcal{L}$ is called Θ-*simple* if it is obtained by augmenting

LK^+ with a finite set of Θ-simple $\mathcal{L}$-rules. We shall omit Θ when it is clear from the context.

We now show that our transformation procedure indeed generates Θ-simple rules from axioms within $\mathbf{Ax}_{\mathcal{L}}$.

Note that if we decompose axioms that are generated by R_1 in the grammar of Definition 39, p_2 can appear only as $\mathtt{b} : p_2$ with $\mathtt{b} \in \{\mathtt{t}, \mathtt{f}\}$, see e.g. Example 28. With the following lemma we show that in step (ii), $\mathtt{b} : p_2$ can be removed from the rule:

Lemma 15. *Let G be a sequent calculus for $\mathcal{L}$ extending LK^+. Let s be an $\mathcal{L}$-sequent, and let $s' = s \cup \{\mathtt{b} : p\}$, where $\mathtt{b} \in \{\mathtt{f}, \mathtt{t}\}$ and p is an atomic formula that does not occur in s. Then, $\vdash_{G \cup \{\emptyset/s'\}} \Gamma \Rightarrow \varphi$ iff $\vdash_{G \cup \{\emptyset/s\}} \Gamma \Rightarrow \varphi$, for every $\mathcal{L}$-sequent $\Gamma \Rightarrow \varphi$.*

Proof. "$\Rightarrow$": Suppose that $\vdash_{G \cup \{\emptyset/s'\}} \Gamma \Rightarrow \varphi$. We can simulate applications of $\emptyset/s'$ by using applications of weakenings and $\emptyset/s$. Hence, $\vdash_{G \cup \{\emptyset/s\}} \Gamma \Rightarrow \varphi$ clearly holds.

"$\Leftarrow$": For the converse direction, suppose that $\vdash_{G \cup \{\emptyset/s\}} \Gamma \Rightarrow \varphi$. Moreover, let P be a derivation of $\Gamma \Rightarrow \varphi$ in $G \cup \{\emptyset/s\}$. We distinguish two cases:

- $\mathtt{b} = \mathtt{f}$. Then every application of $\emptyset/s$ in P deriving $\sigma(s)$ can be simulated in $G \cup \{\emptyset/s'\}$ by using (*cut*) on $\sigma(s) \cup \{\mathtt{f} : p_1 \supset p_1\}$ (obtained by $\emptyset/s'$ in which p is substituted with $p_1 \supset p_1$) and $\sigma(s) \cup \{\mathtt{t} : p_1 \supset p_1\}$ (which is derivable in LK^+).
- $\mathtt{b} = \mathtt{t}$. Then every application of $\emptyset/s$ in P is replaced with an application of $\emptyset/s'$, in which p is substituted with φ. $\mathtt{t} : \varphi$ is then propagated to the end sequent.

$\square$

Theorem 14. *Let $H \in \mathbb{H}$ be a Hilbert calculus for $\mathcal{L}$. There is an algorithm for constructing an equivalent Θ_H-simple sequent calculus G_H for $\mathcal{L}$.*

Proof. Let $H \in \mathbb{H}$, $\psi \in \mathbf{Ax}_{\mathcal{L}} \cap H$ and G_H^- be the sequent calculus equivalent to $H \setminus \{\psi\}$ in the sense of Definition 7. We construct a sequent calculus G_H equivalent to H by extending G_H^- with Θ_H-simple rules that are obtained from ψ. We transform ψ into a set R_ψ of Θ_H-simple rules such that H and $G_H = G_H^- \cup R_\psi$ are equivalent. The theorem follows by repetitive applications of this claim.

First, let $r_\psi = \emptyset/\{\mathtt{t} : \psi\}$. We start by showing that H is equivalent to $G_H^- \cup \{r_\psi\}$. The first direction, $\Gamma \vdash_H \varphi$ implies $\vdash_{G_H^- \cup \{r_\psi\}} \Gamma \Rightarrow \varphi$, is easy and proceeds by induction on the height of the derivation in H. For the converse direction, suppose we have a proof P in $G_H^- \cup \{r_\psi\}$ of the sequent $\Gamma \Rightarrow \varphi$. Then there are substitutions $\sigma_1, \dots, \sigma_n$, for which we can transform P into a proof of $\Gamma, \sigma_1(\psi), \dots, \sigma_n(\psi) \Rightarrow \varphi$ in G_H^-, by replacing every application of r_ψ with the identity axiom $\{\mathtt{f} : \sigma_i(\psi), \mathtt{t} : \sigma_i(\psi)\}$ (and weakening), and propagating $\mathtt{f} : \sigma_i(\psi)$ through the derivation to the end sequent. The equivalence of $H \setminus \{\psi\}$ and G_H^- entails that $\Gamma, \sigma_1(\psi), \dots, \sigma_n(\psi) \vdash_{H \setminus \{\psi\}} \varphi$, and it immediately follows that $\Gamma \vdash_H \varphi$.

The algorithm to transform r_ψ into a set of Θ_H-simple rules works in three steps:

(Step i): We use the logical rules for $\wedge, \vee$ and $\supset$ in LK^+ to obtain a finite set of rules R such that (i) R is equivalent to $\{r_\psi\}$ and (ii) each $r \in R$ has the form $\emptyset/s$, where s

has one of the following forms (depending on whether ψ is generated by R_1 or R_2 in the grammar of Definition 39):

1. s consists of at least one labelled formula of the form $\mathtt{b} : \bar{\star}p_1$ for $\mathtt{b} \in \{\mathtt{f}, \mathtt{t}\}$ and $\bar{\star} \in \Theta_H \setminus \{\epsilon\}$ and any number of labelled formulas of the form $\mathtt{c} : p_2$ or $\mathtt{c} : \bar{\triangleright}p_1$ for $\mathtt{c} \in \{\mathtt{f}, \mathtt{t}\}$ and $\bar{\triangleright} \in \Theta_H$
2. s consists of exactly one labelled formula of the form $\mathtt{b} : \bar{\star}(p_1 \diamond p_2)$ for $\mathtt{b} \in \{\mathtt{f}, \mathtt{t}\}$, $\bar{\star} \in \Theta_H \setminus \{\epsilon\}$ and $\diamond \in \{\wedge, \vee, \supset\}$, and any number of labelled formulas of the form $\mathtt{c} : \bar{\triangleright}p_i$ for $i \in \{1, 2\}, \mathtt{c} \in \{\mathtt{f}, \mathtt{t}\}$ and $\bar{\triangleright} \in \Theta_H$.

The equivalence between $\{r_\psi\}$ and R easily follows by the invertibility of the logical rules in LK^+ (and, hence, in G^-_H). We prove that, when ψ is generated by R_2, s has the form (2) above (the proof for (1) is similar). Indeed if $(*)$ each $r'_\psi \in R$ contains exactly one labelled formula of the form $\mathtt{b} : \bar{\star}(p_1 \diamond p_2)$, $\bar{\star} \in \Theta_H \setminus \{\epsilon\}$, then we are done. Otherwise, we apply the logical rules of LK^+ according to the outermost binary connective of some $\mathtt{b} : \psi_j$ in $r'_\psi \in R$ until we reach condition $(*)$. We distinguish the following cases:

$\mathtt{b} : \psi_j =$ $\mathtt{t} : \varphi_1 \supset \varphi_2$ (or $\mathtt{t} : \varphi_1 \vee \varphi_2$ or $\mathtt{f} : \varphi_1 \wedge \varphi_2$, resp.). By using $(\mathtt{t} : \supset)$ (or $(\mathtt{t} : \vee)$ or $(\mathtt{f} : \wedge)$, resp.), we obtain a new rule $r^1_\psi = r'_\psi$ where $\mathtt{b} : \psi_j$ is replaced by $\mathtt{f} : \varphi_1, \mathtt{t} : \varphi_2$ (or $\mathtt{t} : \varphi_1, \mathtt{t} : \varphi_2$ or $\mathtt{f} : \varphi_1, \mathtt{f} : \varphi_2$, resp.) and hence it contains one binary connective less.

$\mathtt{b} : \psi_j =$ $\mathtt{t} : \varphi_1 \wedge \varphi_2$ (or $\mathtt{f} : \varphi_1 \supset \varphi_2$ or $\mathtt{f} : \varphi_1 \vee \varphi_2$, resp.). By using $(\mathtt{t} : \wedge)$ (or $(\mathtt{f} : \supset)$ or $(\mathtt{f} : \vee)$, resp.), we obtain two rules $\{r^1_\psi, r^2_\psi\}$. $r^1_\psi = r'_\psi$ where $\mathtt{b} : \psi_j$ is replaced by $\mathtt{t} : \varphi_1$ (or $\mathtt{t} : \varphi_1$ or $\mathtt{f} : \varphi_1$, resp.) and $r^2_\psi = r'_\psi$ where $\mathtt{b} : \psi_j$ is replaced by $\mathtt{t} : \varphi_2$ (or $\mathtt{f} : \varphi_2$ or $\mathtt{f} : \varphi_2$, resp.). Note that $\bar{\star}(p_1 \diamond p_2)$ is *not* a subformula of ψ_j by condition (i) ((ii), resp.) in Definition 39.

(Step ii): Obviously, we can discard from R all rules $\emptyset/s$ for which $\{\mathtt{f} : p_i, \mathtt{t} : p_i\} \subseteq s$ for $i \in \{1, 2\}$, keeping the equivalence with $\{r_\psi\}$. For each rule $\emptyset/s$ remaining in R: if s has the form (1) and it contains some $\mathtt{b} : p_2$, by Lemma 15 we remove these formulas and obtain an equivalent set of rules.

(Step iii): For each rule $\emptyset/s \in R$, we take (a) $\mathtt{b} : \bar{\star}p_1$, $\bar{\star} \in \Theta_H \setminus \{\epsilon\}, \mathtt{b} \in \{\mathtt{t}, \mathtt{f}\}$ if s is of form (1), or (b) $\mathtt{b} : \bar{\star}(p_1 \diamond p_2)$, $\bar{\star} \in \Theta_H \setminus \{\epsilon\}, \mathtt{b} \in \{\mathtt{t}, \mathtt{f}\}$ if s is of form (2). We use Lemma 14 to move all remaining labelled formulas to the premises of the rule, changing their side of the sequent, to obtain a set R_ψ of Θ_H-simple rules equivalent to $\{r_\psi\}$. Note that the rules are Θ_H-unary, in case (a) and Θ_H-binary in case (b).

Let G_H be $G^-_H \cup R_\psi$. G_H is equivalent to $G^-_H \cup \{r_\psi\}$ and, hence, equivalent to H. $\square$

Example 29. *The rule equivalent to the axiom* $(\mathbf{c})$ $\neg\neg p_1 \supset p_1$ *(see Table 6.2) is constructed as follows:*

$$\begin{array}{ll} \emptyset/\{\mathtt{t} : \neg\neg p_1 \supset p_1\} & \\ \qquad\longrightarrow^{(i)} & \emptyset/\{\mathtt{f} : \neg\neg p_1, \mathtt{t} : p_1\} \\ \qquad\longrightarrow^{(iii)} & \{\{\mathtt{f} : p_1\}\}/\{\mathtt{f} : \neg\neg p_1\} \end{array}$$

This is the rule $(\mathtt{f} : \neg\neg)$ *of Example 23.*

Example 30. *The rule equivalent to the axiom* $(\mathbf{e}\omega^{\mathbf{r}}_{\wedge\mathbf{2}})$ $(\neg\neg p_1 \wedge \neg\neg p_2) \supset \neg\neg(p_1 \wedge p_2)$ *(see Table 6.3 where* $\sim$ *is replaced with* $\neg$*) is constructed as follows:*

$$\begin{array}{rl}\emptyset/\{\mathtt{t} : (\neg\neg p_1 \wedge \neg\neg p_2) \supset \neg\neg(p_1 \wedge p_2))\} & \\ \longrightarrow^{(i)} & \emptyset/\{\mathtt{f} : \neg\neg p_1 \wedge \neg\neg p_2, \mathtt{t} : \neg\neg(p_1 \wedge p_2)\} \\ \longrightarrow^{(i)} & \emptyset/\{\mathtt{f} : \neg\neg p_1, \mathtt{f} : \neg\neg p_2, \mathtt{t} : \neg\neg(p_1 \wedge p_2)\} \\ \longrightarrow^{(iii)} & \{\{\mathtt{t} : \neg\neg p_1\}, \{\mathtt{t} : \neg\neg p_2\}\}/\{\mathtt{t} : \neg\neg(p_1 \wedge p_2)\}\end{array}$$

This is the rule $(\mathtt{t} : \neg\neg\wedge)$ *of Example 23.*

We showed that we can define sequent calculi for the Hilbert systems belonging to the family $\mathbb{H}$. But this restriction might seem artificial since the transformation procedure can easily be adapted to further extensions of $\mathbf{Cl}^+$. This is the case, e.g., for the infinitely-valued logic L_ω in [105]. The reason for this restriction is indeed not a limit of the transformation procedure, but of the automated generation of the semantics in step 2, which by now only works for Hilbert systems belonging to $\mathbb{H}$.

6.5 Step 2: Automated Extraction of Semantics

We show how to algorithmically obtain semantics for the sequent calculi generated in step 1 by using the semantic framework of *partial non-deterministic matrices* (PNmatrices). From now on, let Θ denote an arbitrary non-empty subset of $\mathcal{U}^*_{\mathcal{L}}$ that is closed under prefixes (in particular, $\epsilon \in \Theta$). We show how to extract a PNmatrix $\mathcal{M}_G$ from a Θ-simple sequent calculus G such that $\vdash_G s$ iff $\vdash_{\mathcal{M}_G} s$.

As already mentioned before, the set Θ plays an important role: it determines the *number* and the *shape* of the truth values of the PNmatrix $\mathcal{M}_G$. Let us first explain this relationship.

Usually, the truth values $\mathtt{t}, \mathtt{f}$ only give information about whether a formula φ is "true" or "false". In our case, however, we do not only want to have information about φ, but also about all the formulas of the form $\bar{\star}\varphi$ for every $\bar{\star} \in \Theta$. Hence, instead of using truth values such as $\mathtt{t}, \mathtt{f}$, we use *functions* from Θ to $\{\mathtt{t}, \mathtt{f}\}$. These functions provide us with information about the "truth" or "falsity" of $\bar{\star}\varphi$. And since Θ always contains ϵ, we also have knowledge about φ. We denote this set of functions with F_Θ:

Definition 43. Let F_Θ denote the set of functions $\Theta \to \{\mathtt{f}, \mathtt{t}\}$. For $\bar{\star} \in \Theta$ and $u \in \mathsf{F}_\Theta$, we write $u^{\bar{\star}}$ to denote $u(\bar{\star})$. We write $\langle \bar{\star}_1 : \mathtt{b}_1, \ldots, \bar{\star}_n : \mathtt{b}_n \rangle$ for the function u, such that $u^{\bar{\star}_i} = \mathtt{b}_i$ for $1 \le i \le n$ and $\Theta = \{\bar{\star}_1, \ldots, \bar{\star}_n\}$.

A function $v : \mathcal{L} \to \mathsf{F}_\Theta$ is called *consistent* if $v(\bar{\triangleright}\varphi)^{\bar{\star}} = v(\varphi)^{\bar{\star}\bar{\triangleright}}$ for every formula φ and $\bar{\star}, \bar{\star}\bar{\triangleright} \in \Theta$.

By using F_Θ as a set of truth values, the information about "truth/falsity" of a formula can occur in *several places* in the truth values assigned by a valuation. For example, when $\Theta = \{\epsilon, \neg\}$, the information whether $\neg\varphi$ is "true" occurs in $v(\neg\varphi)^\epsilon$ and in $v(\varphi)^\neg$. The consistency property in Definition 43 ensures that there are no contradictions between these places that store the same information.

We now finally turn to step 2 of our general procedure. The algorithm to extract semantics out of any given Θ-simple sequent calculus is based on the following observations:

- The Θ-unary rules of G (see Definition 42) affect certain relationships between various formulas of the form $\bar{\star}\varphi$ for $\bar{\star} \in \Theta$. Thus, we only consider those truth values from all the possible functions in F_Θ that *respect* the Θ-unary rules, see Definition 44.
- The truth tables of the unary connectives are constructed using the information contained in each of the truth values concerning each connective. These tables will guarantee that $\mathcal{M}_G$-valuations are consistent.
- The truth tables of the binary connectives are constructed using the Θ-binary rules of G.

To define $\mathcal{M}_G$, we use the following additional notions:

Definition 44. Let $\bar{\star} \in \Theta$ and $u_1, u_2 \in \mathsf{F}_\Theta$.

- u_1 *satisfies* an $\mathcal{L}$-sequent of the form $\{\mathtt{b} : \bar{\star}p_1\}$ if $u_1^{\bar{\star}} = \mathtt{b}$.
- u_1 *respects* a Θ-unary rule $Q/\{\mathtt{b} : \bar{\star}\, p_1\}$ if it satisfies $\{\mathtt{b} : \bar{\star}p_1\}$ whenever it satisfies every $q \in Q$.
- The ordered pair $\langle u_1, u_2\rangle$ *satisfies* an $\mathcal{L}$-sequent of the form $\{\mathtt{b} : \bar{\star}p_i\}$ if $u_i^{\bar{\star}} = \mathtt{b}$ for $i \in \{1, 2\}$.

Example 31. *Let $\Theta = \{\epsilon, \neg, \circ, \neg\neg\}$. Consider the following $\mathcal{L}$-sequents:*

$$\{\mathtt{f} : p_1\} \qquad (s_1)$$
$$\{\mathtt{f} : \neg p_1\} \qquad (s_2)$$
$$\{\mathtt{f} : \neg\neg p_2\} \qquad (s_3)$$
$$\{\mathtt{t} : \circ p_2\} \qquad (s_4)$$

the following functions

$$\langle \epsilon : \mathtt{f}, \neg : \mathtt{f}, \circ : \mathtt{f}, \neg\neg : \mathtt{f}\rangle \qquad (x)$$
$$\langle \epsilon : \mathtt{t}, \neg : \mathtt{f}, \circ : \mathtt{t}, \neg\neg : \mathtt{t}\rangle \qquad (y)$$

and the Θ-unary rule

$$r = \{\{\mathtt{f} : p_1\}, \{\mathtt{f} : \neg p_1\}\}/\{\mathtt{f} : \neg\neg p_1\}$$

Then x satisfies s_1, s_2, s_3 and does not satisfy s_4. y satisfies s_2, s_4 and does not satisfy s_1, s_3. x respects r as it satisfies both premises, $\{\mathtt{f} : p_1\}$ and $\{\mathtt{f} : \neg p_1\}$, and the conclusion $\{\mathtt{f} : \neg\neg p_1\}$. y also respects r because it does not satisfy the premise $\{\mathtt{f} : p_1\}$.

We can now construct the PNmatrix $\mathcal{M}_G$ corresponding to a Θ-simple sequent calculus G:

Definition 45. Let G be a Θ-simple sequent calculus. Its PNmatrix $\mathcal{M}_G$ is defined as follows:

- The set of truth values $\mathcal{V}_{\mathcal{M}_G}$ contains all functions in F_Θ that respect all Θ-unary rules of G.
- The set of designated truth values $\mathcal{D}_{\mathcal{M}_G}$ is $\{u \in \mathcal{V}_{\mathcal{M}_G} \mid u^\epsilon = \mathtt{t}\}$.
- For any unary connective $\star \in \mathcal{U}_\mathcal{L}$, the truth table for $\star$ is given by:

$$\star_{\mathcal{M}_G}(u_1) = \{u \in \mathcal{V}_{\mathcal{M}_G} \mid u^{\bar{\star}} = u_1^{\bar{\star}\star} \text{ whenever } \bar{\star}\star \in \Theta\}.$$

- For $\diamond \in \{\wedge, \vee, \supset\}$ and $u_1, u_2 \in \mathcal{V}_{\mathcal{M}_G}$, $\diamond_{\mathcal{M}_G}(u_1, u_2)$ is the set of all $u \in \mathcal{V}_{\mathcal{M}_G}$ satisfying:
  1. $u^\epsilon \in \diamond_{\mathcal{M}_{\mathbf{Cl}^+}}(u_1^\epsilon, u_2^\epsilon)$ (where $\diamond_{\mathcal{M}_{\mathbf{Cl}^+}}$ is the classical truth table of $\diamond$; see Example 24).
  2. For every Θ-binary rule of G of the form $Q/\{\mathtt{b} : \bar{\star}(p_1 \diamond p_2)\}$, if $\langle u_1, u_2\rangle$ satisfies every $q \in Q$ then $u^{\bar{\star}} = \mathtt{b}$.

Example 32. *Consider the calculus H_0 that extends HCL^+ with the two axioms from Example 29* (**c**) $\neg\neg p_1 \supset p_1$ *and Example 30* $(\mathbf{e}\omega^{\mathbf{r}}_{\wedge \mathbf{2}})$ $(\neg\neg p_1 \wedge \neg\neg p_2) \supset \neg\neg(p_1 \wedge p_2)$. *Then, $\Theta_{H_0} = \{\epsilon, \neg, \neg\neg\}$. The corresponding Θ_{H_0}-simple sequent calculus G_{H_0} has one unary rule*

$$r_u = \{\{\mathtt{f} : p_1\}\}/\{\mathtt{f} : \neg\neg p_1\}$$

and one binary rule

$$r_b = \{\{\mathtt{t} : \neg\neg p_1\}, \{\mathtt{t} : \neg\neg p_2\}\}/\{\mathtt{t} : \neg\neg(p_1 \wedge p_2)\}$$

We construct the PNmatrix $\mathcal{M} = \mathcal{M}_{G_{H_0}}$ according to Definition 45. We start by listing $\mathsf{F}_{\Theta_{H_0}}$:

$$\mathsf{F}_{\Theta_{H_0}} = \{\langle \epsilon : \mathtt{f}, \neg : \mathtt{f}, \neg\neg : \mathtt{f}\rangle, \langle \epsilon : \mathtt{f}, \neg : \mathtt{f}, \neg\neg : \mathtt{t}\rangle, \langle \epsilon : \mathtt{f}, \neg : \mathtt{t}, \neg\neg : \mathtt{f}\rangle, \langle \epsilon : \mathtt{f}, \neg : \mathtt{t}, \neg\neg : \mathtt{t}\rangle, \langle \epsilon : \mathtt{t}, \neg : \mathtt{f}, \neg\neg : \mathtt{f}\rangle, \langle \epsilon : \mathtt{t}, \neg : \mathtt{f}, \neg\neg : \mathtt{t}\rangle, \langle \epsilon : \mathtt{t}, \neg : \mathtt{t}, \neg\neg : \mathtt{f}\rangle, \langle \epsilon : \mathtt{t}, \neg : \mathtt{t}, \neg\neg : \mathtt{t}\rangle\}$$

First, we determine the set $\mathcal{V}_\mathcal{M}$ of truth values that respect the unary rules of G_{H_0}. The only relevant unary rule in this case is r_u. Since $u \in \mathcal{V}_\mathcal{M}$ respects r_u iff

$$u^{\neg\neg} = \mathtt{f} \text{ whenever } u^\epsilon = \mathtt{f}$$

we delete the values

$$\{\langle \epsilon : \mathtt{f}, \neg : \mathtt{f}, \neg\neg : \mathtt{t}\rangle, \langle \epsilon : \mathtt{f}, \neg : \mathtt{t}, \neg\neg : \mathtt{t}\rangle\}$$

and obtain:

$$\mathcal{V}_\mathcal{M} = \{\langle \epsilon : \mathtt{f}, \neg : \mathtt{f}, \neg\neg : \mathtt{f}\rangle, \langle \epsilon : \mathtt{f}, \neg : \mathtt{t}, \neg\neg : \mathtt{f}\rangle, \langle \epsilon : \mathtt{t}, \neg : \mathtt{f}, \neg\neg : \mathtt{f}\rangle, \langle \epsilon : \mathtt{t}, \neg : \mathtt{f}, \neg\neg : \mathtt{t}\rangle, \langle \epsilon : \mathtt{t}, \neg : \mathtt{t}, \neg\neg : \mathtt{f}\rangle, \langle \epsilon : \mathtt{t}, \neg : \mathtt{t}, \neg\neg : \mathtt{t}\rangle\}$$

The set of designated truth values is:

$$\mathcal{D}_\mathcal{M} = \{\langle \epsilon : \mathtt{t}, \neg : \mathtt{f}, \neg\neg : \mathtt{f}\rangle, \langle \epsilon : \mathtt{t}, \neg : \mathtt{f}, \neg\neg : \mathtt{t}\rangle, \langle \epsilon : \mathtt{t}, \neg : \mathtt{t}, \neg\neg : \mathtt{f}\rangle, \langle \epsilon : \mathtt{t}, \neg : \mathtt{t}, \neg\neg : \mathtt{t}\rangle\}$$

Next we define the truth table for $\neg$. For every $u_1 \in \mathcal{V}_\mathcal{M}$, we take all $u \in \mathcal{V}_\mathcal{M}$ that satisfy the condition

$$u^{\bar\star} = u_1^{\bar\star\neg}$$

for all $\bar\star \in \Theta_{H_0}$ such that $\bar\star\neg \in \Theta_{H_0}$. For instance, let $u_1 = \langle \epsilon : \mathtt{f}, \neg : \mathtt{t}, \neg\neg : \mathtt{f}\rangle$. We consider those elements u from $\mathcal{V}_\mathcal{M}$ in which

$$u^\epsilon = u_1^\neg = \mathtt{t} \text{ and } u^\neg = u_1^{\neg\neg} = \mathtt{f}$$

The only such elements are

$$\langle \epsilon : \mathtt{t}, \neg : \mathtt{f}, \neg\neg : \mathtt{f}\rangle \text{ and } \langle \epsilon : \mathtt{t}, \neg : \mathtt{f}, \neg\neg : \mathtt{t}\rangle$$

The truth table for $\neg$ is thus defined as follows (we write below $\langle x, y, z\rangle$ instead of $\langle \epsilon : x, \neg : y, \neg\neg : z\rangle$):

$\neg_\mathcal{M}$	
$\langle \mathtt{f},\mathtt{f},\mathtt{f}\rangle$	$\{\langle \mathtt{f},\mathtt{f},\mathtt{f}\rangle\}$
$\langle \mathtt{f},\mathtt{t},\mathtt{f}\rangle$	$\{\langle \mathtt{t},\mathtt{f},\mathtt{f}\rangle, \langle \mathtt{t},\mathtt{f},\mathtt{t}\rangle\}$
$\langle \mathtt{t},\mathtt{f},\mathtt{f}\rangle$	$\{\langle \mathtt{f},\mathtt{f},\mathtt{f}\rangle\}$
$\langle \mathtt{t},\mathtt{f},\mathtt{t}\rangle$	$\{\langle \mathtt{f},\mathtt{t},\mathtt{f}\rangle\}$
$\langle \mathtt{t},\mathtt{t},\mathtt{f}\rangle$	$\{\langle \mathtt{t},\mathtt{f},\mathtt{f}\rangle, \langle \mathtt{t},\mathtt{f},\mathtt{t}\rangle\}$
$\langle \mathtt{t},\mathtt{t},\mathtt{t}\rangle$	$\{\langle \mathtt{t},\mathtt{t},\mathtt{f}\rangle, \langle \mathtt{t},\mathtt{t},\mathtt{t}\rangle\}$

Finally, we obtain the truth tables for the binary connectives which must meet the requirements arising (1) from the classical truth tables and (2) from the binary rules of G_{H_0}. We show the case of $\wedge$. The only binary rule of G_{H_0} that involves $\wedge$ is r_b, which imposes the requirement that for every $u \in \wedge_\mathcal{M}(u_1, u_2)$ we have

$$u^{\neg\neg} = \mathtt{t} \text{ whenever } u_1^{\neg\neg} = \mathtt{t} \text{ and } u_2^{\neg\neg} = \mathtt{t}$$

In addition, u^ϵ must behave classically. Thus we obtain the following truth table (let $\mathcal{F}_\mathcal{M} = \mathcal{V}_\mathcal{M} \setminus \mathcal{D}_\mathcal{M}$):

$\wedge_\mathcal{M}$	$\langle \mathtt{f},\mathtt{f},\mathtt{f}\rangle$	$\langle \mathtt{f},\mathtt{t},\mathtt{f}\rangle$	$\langle \mathtt{t},\mathtt{f},\mathtt{f}\rangle$	$\langle \mathtt{t},\mathtt{f},\mathtt{t}\rangle$	$\langle \mathtt{t},\mathtt{t},\mathtt{f}\rangle$	$\langle \mathtt{t},\mathtt{t},\mathtt{t}\rangle$
$\langle \mathtt{f},\mathtt{f},\mathtt{f}\rangle$	$\mathcal{F}_\mathcal{M}$	$\mathcal{F}_\mathcal{M}$	$\mathcal{F}_\mathcal{M}$	$\mathcal{F}_\mathcal{M}$	$\mathcal{F}_\mathcal{M}$	$\mathcal{F}_\mathcal{M}$
$\langle \mathtt{f},\mathtt{t},\mathtt{f}\rangle$	$\mathcal{F}_\mathcal{M}$	$\mathcal{F}_\mathcal{M}$	$\mathcal{F}_\mathcal{M}$	$\mathcal{F}_\mathcal{M}$	$\mathcal{F}_\mathcal{M}$	$\mathcal{F}_\mathcal{M}$
$\langle \mathtt{t},\mathtt{f},\mathtt{f}\rangle$	$\mathcal{F}_\mathcal{M}$	$\mathcal{F}_\mathcal{M}$	$\mathcal{D}_\mathcal{M}$	$\mathcal{D}_\mathcal{M}$	$\mathcal{D}_\mathcal{M}$	$\mathcal{D}_\mathcal{M}$
$\langle \mathtt{t},\mathtt{f},\mathtt{t}\rangle$	$\mathcal{F}_\mathcal{M}$	$\mathcal{F}_\mathcal{M}$	$\mathcal{D}_\mathcal{M}$	$\{\langle \mathtt{t},\mathtt{f},\mathtt{t}\rangle, \langle \mathtt{t},\mathtt{t},\mathtt{t}\rangle\}$	$\mathcal{D}_\mathcal{M}$	$\{\langle \mathtt{t},\mathtt{f},\mathtt{t}\rangle, \langle \mathtt{t},\mathtt{t},\mathtt{t}\rangle\}$
$\langle \mathtt{t},\mathtt{t},\mathtt{f}\rangle$	$\mathcal{F}_\mathcal{M}$	$\mathcal{F}_\mathcal{M}$	$\mathcal{D}_\mathcal{M}$	$\mathcal{D}_\mathcal{M}$	$\mathcal{D}_\mathcal{M}$	$\mathcal{D}_\mathcal{M}$
$\langle \mathtt{t},\mathtt{t},\mathtt{t}\rangle$	$\mathcal{F}_\mathcal{M}$	$\mathcal{F}_\mathcal{M}$	$\mathcal{D}_\mathcal{M}$	$\{\langle \mathtt{t},\mathtt{f},\mathtt{t}\rangle, \langle \mathtt{t},\mathtt{t},\mathtt{t}\rangle\}$	$\mathcal{D}_\mathcal{M}$	$\{\langle \mathtt{t},\mathtt{f},\mathtt{t}\rangle, \langle \mathtt{t},\mathtt{t},\mathtt{t}\rangle\}$

Example 33. *Consider $HCL^+ \in \mathbb{H}$, where $\Theta_{HCL^+} = \{\epsilon\}$. By applying the method described in step 1 (Section 6.4), we obtain $G_{HCL^+} = LK^+$. The corresponding PNmatrix $\mathcal{M}_{LK^+}$ has two truth values $u_1, u_2 \in \{\epsilon\} \to \{\mathtt{t}, \mathtt{f}\}$, where $u_1(\epsilon) = \mathtt{t}$ and $u_2(\epsilon) = \mathtt{f}$. Identifying u_1, u_2 with $\mathtt{t}$ and $\mathtt{f}$ respectively leads to the classical PNmatrix $\mathcal{M}_{\mathbf{Cl}^+}$ of Example 24.*

The next lemma asserts that $\mathcal{M}_G$ valuations are always consistent.

Lemma 16. *Let G be a Θ-simple sequent calculus for a propositional language $\mathcal{L}$, and v be an $\mathcal{M}_G$-valuation for $\mathcal{L}$. Then $v(\bar{\triangleright}\varphi)^{\bar{\star}} = v(\varphi)^{\bar{\star}\bar{\triangleright}}$ for every formula φ and $\bar{\triangleright}, \bar{\star} \in \mathcal{U}^*_{\mathcal{L}}$ such that $\bar{\star}\bar{\triangleright} \in \Theta$.*

Proof. By induction on the length of $\bar{\triangleright}$. In the base case, when $\bar{\triangleright} = \epsilon$, we are done. Suppose the claim holds when $\bar{\triangleright}$ is of length n and let $\star\bar{\triangleright} \in \mathcal{U}^*_{\mathcal{L}}$ (for $\star \in \mathcal{U}_{\mathcal{L}}$). Since v is an $\mathcal{M}_G$-valuation, $v(\star\bar{\triangleright}\varphi) \in \star_{\mathcal{M}_G}(v(\bar{\triangleright}\varphi))$. By Definition 45, this implies that $v(\star\bar{\triangleright}\varphi)^{\bar{\star}} = v(\bar{\triangleright}\varphi)^{\bar{\star}\star}$ (note that $\bar{\star}\star \in \Theta$ since Θ is closed under prefixes and $\bar{\star} \star \bar{\triangleright} \in \Theta$). By the inductive hypothesis, the claim $v(\bar{\triangleright}\varphi)^{\bar{\star}\star} = v(\varphi)^{\bar{\star}\star\bar{\triangleright}}$ holds. □

We are now ready to prove soundness and completeness for G with respect to $\mathcal{M}_G$.

Theorem 15 (Soundness and completeness)**.** *Let G be a Θ-simple sequent calculus for $\mathcal{L}$ and s_0 be an $\mathcal{L}$-sequent. Then, $\vdash_G s_0$ iff $\vdash_{\mathcal{M}_G} s_0$.*

Proof. "⇒": It suffices to show that the applications of the rules of G are "sound". Consider an application of a rule $r = Q/s$ of G inferring $\sigma(s) \cup c$ from $\{\sigma(q) \cup c \mid q \in Q\}$, where σ is an $\mathcal{L}$-substitution and c is an $\mathcal{L}$-sequent. Let v be an $\mathcal{M}_G$-valuation for $\mathcal{L}$. Suppose that v satisfies $\sigma(q) \cup c$ for every $q \in Q$. We show that v satisfies $\sigma(s) \cup c$. If v satisfies c, we are done. Suppose otherwise; then v satisfies $\sigma(q)$ for all $q \in Q$. We show that v satisfies $\sigma(s)$. We only consider the case when r is a Θ-unary rule (the proofs for Θ-binary and the rules of LK^+ rules are similar). Hence, $s = \{\mathtt{b}_0 : \bar{\triangleright}\, p_1\}$ for $\bar{\triangleright} \in \Theta \setminus \{\epsilon\}$ and $\mathtt{b}_0 \in \{\mathtt{t}, \mathtt{f}\}$. Let $\psi_1 = \sigma(p_1)$. We show that $v(\psi_1)$ satisfies every $q \in Q$. Let $q = \{\mathtt{b} : \bar{\star}\, p_1\}$ be a premise in Q. Then, $\bar{\star} \in \Theta$. The fact that v satisfies $\sigma(q) = \{\mathtt{b} : \bar{\star}\psi_1\}$ implies that $v(\bar{\star}\psi_1)^{\epsilon} = \mathtt{b}$. By Lemma 16, $v(\psi_1)^{\bar{\star}} = \mathtt{b}$. Hence $v(\psi_1)$ satisfies q. Since $v(\psi_1) \in \mathcal{V}_{\mathcal{M}_G}$, it respects r, and so $v(\psi_1)^{\bar{\triangleright}} = \mathtt{b}_0$. By Lemma 16, $v(\bar{\triangleright}\psi_1)^{\epsilon} = \mathtt{b}_0$. Thus, v satisfies $\sigma(s)$.

"⇐": Suppose that $\not\vdash_G s_0$. We construct an $\mathcal{M}_G$-valuation for $\mathcal{L}$ that does not satisfy s_0. It is a standard matter to construct a "maximal" (infinite) set Ω of labelled $\mathcal{L}$-formulas, extending s_0, that satisfies the following conditions:

1. $\not\vdash_G s$ for every $\mathcal{L}$-sequent $s \subseteq \Omega$.
2. For every labelled $\mathcal{L}$-formula $\mathtt{b} : \psi \notin \Omega$, we have $\vdash_G s \cup \{\mathtt{b} : \psi\}$ for some $\mathcal{L}$-sequent $s \subseteq \Omega$.

Note that the availability of the rules (*cut*) and (*id*) implies the following two facts:

1. For every $\mathcal{L}$-formula ψ, either $\texttt{f} : \psi \in \Omega$ or $\texttt{t} : \psi \in \Omega$. Otherwise $\vdash_G s_1 \cup \{\texttt{f} : \psi\}$ and $\vdash_G s_2 \cup \{\texttt{t} : \psi\}$ with $s_1, s_2 \subseteq \Omega$. By applying (*cut*) (and possibly weakenings), we obtain $\vdash_G s_1 \cup s_2$. Since $s_1 \cup s_2 \subseteq \Omega$, this contradicts the properties of Ω.
2. For every $\mathcal{L}$-formula ψ, either $\texttt{f} : \psi \notin \Omega$ or $\texttt{t} : \psi \notin \Omega$. Otherwise $\{\texttt{f} : \psi, \texttt{t} : \psi\} \subseteq \Omega$, but $\vdash_G \{\texttt{f} : \psi, \texttt{t} : \psi\}$ using (*id*).

Let v be the function from $\mathcal{L}$ to F_Θ defined by $v(\psi)^{\bar\star} = \texttt{b}$ iff $\texttt{b} : \bar\star\psi \notin \Omega$ for every $\psi \in \mathcal{L}$ and $\bar\star \in \Theta$. By the two facts above, v is well defined. To show that v is an $\mathcal{M}_G$-valuation, we use the following properties:

(*) Let σ be an $\mathcal{L}$-substitution. If $v(\sigma(p_1))$ satisfies an $\mathcal{L}$-sequent q of the form $\{\texttt{b} : \bar\star p_1\}$ where $\bar\star \in \Theta$ then $\vdash_G s \cup \sigma(q)$ for some $\mathcal{L}$-sequent $s \subseteq \Omega$.
Proof: Suppose that $v(\sigma(p_1))$ satisfies q. Thus $v(\sigma(p_1))^{\bar\star} = \texttt{b}$. It follows that $\texttt{b} : \bar\star\sigma(p_1) \notin \Omega$. Hence there is some $\mathcal{L}$-sequent $s \subseteq \Omega$ such that $\vdash_G s \cup \{\texttt{b} : \bar\star\sigma(p_1)\}$.

(**) Let σ be an $\mathcal{L}$-substitution. If $\langle v(\sigma(p_1)), v(\sigma(p_2))\rangle$ satisfies an $\mathcal{L}$-sequent q of the form $\{\texttt{b} : \bar\star p_i\}$ where $\bar\star \in \Theta$ and $i \in \{1, 2\}$ then $\vdash_G s \cup \sigma(q)$ for some $\mathcal{L}$-sequent $s \subseteq \Omega$.
Proof: Suppose that $\langle v(\sigma(p_1)), v(\sigma(p_2))\rangle$ satisfies q. Thus $v(\sigma(p_i))^{\bar\star} = \texttt{b}$. It follows that $\texttt{b} : \bar\star\sigma(p_i) \notin \Omega$. Hence there is some $\mathcal{L}$-sequent $s \subseteq \Omega$ such that $\vdash_G s \cup \{\texttt{b} : \bar\star\sigma(p_i)\}$.

We first prove that for every $\mathcal{L}$-formula ψ, $v(\psi)$ respects all the Θ-unary rules of G, and so $v(\psi) \in \mathcal{V}_{\mathcal{M}_G}$. Let ψ be an $\mathcal{L}$-formula, and $r = Q/\{\texttt{b} : \bar\star p_1\}$ be a Θ-unary rule of G. Suppose that $v(\psi)$ satisfies every $q \in Q$. Consider an $\mathcal{L}$-substitution σ for which $\sigma(p_1) = \psi$. By (*), for every $q \in Q$ there is some sequent $s_q \subseteq \Omega$ such that $\vdash_G s_q \cup \sigma(q)$. By applying (weakenings and) r we obtain $\vdash_G \bigcup_{q\in Q} s_q \cup \{\texttt{b} : \bar\star\psi\}$. Thus, $\{\texttt{b} : \bar\star\psi\} \notin \Omega$ and so $v(\psi)^{\bar\star} = \texttt{b}$.

Next, we show that v respects the truth tables of $\mathcal{M}_G$:

1. Let $\star \in \mathcal{U}_\mathcal{L}$ and $\psi \in \mathcal{L}$. We show that $v(\star\psi) \in \star_{\mathcal{M}_G}(v(\psi))$. By the definition of $\mathcal{M}_G$, it suffices to show that $v(\star\psi)^{\bar\star} = v(\psi)^{\bar\star\star}$ whenever $\bar\star\star \in \Theta$. This follows directly from the definition of v.
2. Let $\diamond \in \{\wedge, \vee, \supset\}$ and $\psi_1, \psi_2 \in \mathcal{L}$. We show that $v(\psi_1 \diamond \psi_2) \in \diamond_{\mathcal{M}_G}(v(\psi_1), v(\psi_2))$ by showing (*i*) $v(\psi_1 \diamond \psi_2)^\epsilon \in \diamond_{\mathcal{M}_{\mathbf{Cl}^+}}(v(\psi_1)^\epsilon, v(\psi_2)^\epsilon)$ and (*ii*) $v(\psi_1 \diamond \psi_2)^{\bar\star} = \texttt{b}$ for every Θ-binary rule $r = Q/\{\texttt{b} : \bar\star(p_1 \diamond p_2)\}$ of G for which $\langle v(\psi_1), v(\psi_2)\rangle$ satisfies every $q \in Q$.
   - (*i*) We prove (*i*) for the specific case when $\diamond = \wedge$ and $v(\psi_1)^\epsilon = v(\psi_2)^\epsilon = \texttt{t}$. All other cases are handled similarly. We show that $v(\psi_1 \wedge \psi_2)^\epsilon = \texttt{t}$. The definition of v ensures that both $\texttt{t} : \psi_1$ and $\texttt{t} : \psi_2$ do not occur in Ω. Thus $\vdash_G s_1 \cup \{\texttt{t} : \psi_1\}$ and $\vdash_G s_2 \cup \{\texttt{t} : \psi_2\}$ for some $s_1, s_2 \subseteq \Omega$. By applying (weakenings and) the rule $(\texttt{t} : \wedge)$ of LK^+, we obtain that $\vdash_G s_1 \cup s_2 \cup \{\texttt{t} : \psi_1 \wedge \psi_2\}$. Hence we have $t : \psi_1 \wedge \psi_2 \notin \Omega$. It follows that $v(\psi_1 \wedge \psi_2)^\epsilon = \texttt{t}$.

(*ii*) Let $r = Q/\{b : \bar{\star}(p_1 \diamond p_2)\}$ be a Θ-binary rule of G, such that $\langle v(\psi_1), v(\psi_2)\rangle$ satisfies every $q \in Q$. Consider an $\mathcal{L}$-substitution σ for which $\sigma(p_1) = \psi_1$ and $\sigma(p_2) = \psi_2$. By (**), for every $q \in Q$ there is some sequent $s_q \subseteq \Omega$ such that $\vdash_G s_q \cup \sigma(q)$. By applying (weakenings and) r we obtain $\vdash_G \bigcup_{q \in Q} s_q \cup \{\mathtt{b} : \bar{\star}(\psi_1 \diamond \psi_2)\}$. Thus, $\{\mathtt{b} : \bar{\star}(\psi_1 \diamond \psi_2)\} \notin \Omega$ and so $v(\psi_1 \diamond \psi_2)^{\bar{\star}} = \mathtt{b}$.

Finally, note that v does not satisfy s_0. Indeed, every $\mathtt{b} : \psi \in s_0$ is also an element of Ω, and hence $v(\psi)^{\epsilon} \neq \mathtt{b}$. Thus we have $v \not\models_{\mathcal{M}_G} \psi$ for every $\mathtt{t} : \psi \in s$, and $v \models_{\mathcal{M}_G} \psi$ for every $\mathtt{f} : \psi \in s$. □

Combining the previous theorem with Theorem 14, we get:

Corollary 5. *Let $H \in \mathbb{H}$ be a Hilbert calculus for $\mathcal{L}$. For every finite set $\Gamma \cup \{\varphi\}$ of $\mathcal{L}$-formulas, $\Gamma \vdash_H \varphi$ iff $\Gamma \vdash_{\mathcal{M}_{G_H}} \varphi$.*

Proof. Suppose that $\Gamma \vdash_H \varphi$. By Theorem 14, we have $\vdash_{G_H} \Gamma \Rightarrow \varphi$. Theorem 15 implies that $\vdash_{\mathcal{M}_{G_H}} \Gamma \Rightarrow \varphi$. By definition, it follows that $\Gamma \vdash_{\mathcal{M}_{G_H}} \varphi$. The converse is similar. □

Using Proposition 3, we also obtain a general decidability result:

Corollary 6 (Decidability)**.** *Given a Hilbert system $H \in \mathbb{H}$ and a finite set $\Gamma \cup \{\varphi\}$ of formulas, it is decidable whether $\Gamma \vdash_H \varphi$ or not.*

Proof. Follows by Corollary 5 and Proposition 3. □

Exploiting Step 1 and Step 2: Analyticity

After having established the connection between a PNmatrix $\mathcal{M}_G$ and the sequent calculus G, we can exploit this relationship to reason about the analyticity of G.

As already mentioned in Chapter 2, calculi with the (global) subformula property are often referred to as analytic (see, e.g. [144]). This means that a sequent calculus is analytic if, whenever a sequent s is provable in it, it can also be proven using only the "syntactic material available within s", where by "material" we mean all subformulas occurring in s (denoted by $sub[s]$). However, weaker variants of the subformula property have also been considered in the literature, especially in paraconsistent and modal logics, e.g., [13, 116, 38, 39]. Here we use the following notion of Θ-analyticity:

Definition 46. Let $\Theta \subseteq \mathcal{U}^*_{\mathcal{L}}$ and $\mathcal{W}$ be a set of $\mathcal{L}$-formulas. Then

$$\Theta(\mathcal{W}) = \{\bar{\star}\psi \mid \bar{\star} \in \Theta, \psi \in \mathcal{W}\}$$

Let G be a sequent calculus for a propositional language $\mathcal{L}$.

1. Given an $\mathcal{L}$-sequent s and a set $\mathcal{W}$ of $\mathcal{L}$-formulas, we write $\vdash^{\mathcal{W}}_G s$ if there exists a proof of s in G consisting only of ($\mathcal{L}$-sequents that consist of) formulas from $\mathcal{W}$.
2. Given a set $\Theta \subseteq \mathcal{U}^*_{\mathcal{L}}$, G is called Θ-*analytic* if $\vdash_G s$ implies $\vdash^{\Theta(sub[s])}_G s$ for every sequent s.

We call a sequent calculus G thus Θ-analytic when for every provable sequent s, we can find a proof where the "material" occurring in its proof consists only of subformulas and their possible extensions with sequences of unary connectives contained in Θ. In this regard, our notion of analyticity is closely related to the notion of *bounded proof property* (defined in [38, 39] in the context of modal logic). This property states that the complexity of formulas appearing in s determines the bound on the complexity of formulas constructed from the subformulas of s that are allowed to appear in the proof.

The reasoning about the Θ_H-analyticity of the sequent calculus G_H amounts to the check of a decidable sufficient condition of the associated PNmatrix $\mathcal{M}_{G_H}$, namely whether the PNmatrix $\mathcal{M}_{G_H}$ is in fact an Nmatrix, i.e., if its set of truth values $\mathcal{V}_{\mathcal{M}}$ is non-empty and $\heartsuit_{\mathcal{M}}(x_1,\ldots,x_n) \neq \emptyset$ for every n-ary connective $\heartsuit$ of $\mathcal{L}$ and $x_1,\ldots,x_n \in \mathcal{V}_{\mathcal{M}}$.

Theorem 16. *Let G be a Θ-simple sequent calculus. If $\mathcal{M}_G$ is an Nmatrix then G is Θ-analytic.*

Proof. Suppose that $\mathcal{M}_G$ is an Nmatrix and $\nvdash_G^{\Theta(sub[s_0])} s_0$ for some $\mathcal{L}$-sequent s_0. We show that $\nvdash_G s_0$. By Theorem 15, it suffices to show that there exists an $\mathcal{M}_G$-valuation for $\mathcal{L}$ that does not satisfy s_0. Let $\mathcal{W} = \Theta(sub[s_0])$. It is a standard matter to extend s_0 into a "maximal" $\mathcal{L}$-sequent s^* that satisfies the following conditions:

1. s^* consists of labelled $\mathcal{L}$-formulas of the form $\mathtt{b} : \psi$ and $\psi \in \mathcal{W}$.
2. $\nvdash_G^{\mathcal{W}} s^*$.
3. For every labelled $\mathcal{L}$-formula $\mathtt{b} : \psi$ with $\psi \in \mathcal{W}$, if $\mathtt{b} : \psi \notin s^*$ then $\vdash_G^{\mathcal{W}} s^* \cup \{\mathtt{b} : \psi\}$.

As in the proof of Theorem 15, the availability of the rules (cut) and (id) implies that for every $\psi \in \mathcal{W}$ there is a unique $\mathtt{b} \in \{\mathtt{t}, \mathtt{f}\}$ such that $\mathtt{b} : \psi \in s^*$. Next, we define a function $v : \mathcal{L} \to \mathsf{F}_\Theta$ by induction on the structure of formulas. Suppose that $v(\varphi)$ is defined for every proper subformula φ of an $\mathcal{L}$-formula ψ. We define $v(\psi)$ as follows. First, if $\psi \in sub[s_0]$ then for every $\bar{\star} \in \Theta$: $v(\psi)^{\bar{\star}} = \mathtt{b}$ iff $\mathtt{b} : \bar{\star}\psi \notin s^*$. Otherwise, if ψ is an atomic formula, $v(\psi)$ is arbitrarily chosen to be one of the truth values in $\mathcal{V}_{\mathcal{M}_G}$. Otherwise, $\psi = \heartsuit(\psi_1,\ldots,\psi_n)$ is a compound formula, and in this case $v(\psi)$ is arbitrarily chosen to be one of the truth values in $\heartsuit_{\mathcal{M}_G}(v(\psi_1),\ldots,v(\psi_n))$. Note that the fact that $\mathcal{M}_G$ is an Nmatrix guarantees that these arbitrary choices are always possible. To show that v is an $\mathcal{M}_G$-valuation, we use the following properties:

(*) Let σ be an $\mathcal{L}$-substitution such that $\sigma(p_1) \in sub[s_0]$. If $v(\sigma(p_1))$ satisfies an $\mathcal{L}$-sequent of the form $\{\mathtt{b} : \bar{\star}p_1\}$ where $\bar{\star} \in \Theta$ then $\vdash_G^{\mathcal{W}} s^* \cup \{\mathtt{b} : \bar{\star}\sigma(p_1)\}$.
Proof: Suppose that $v(\sigma(p_1))$ satisfies $\{\mathtt{b} : \bar{\star}p_1\}$. Thus $v(\sigma(p_1))^{\bar{\star}} = \mathtt{b}$. Since $\sigma(p_1) \in sub[s_0]$, we have that $\mathtt{b} : \bar{\star}\sigma(p_1) \notin s^*$. The maximality of s^* ensures that $\vdash_G^{\mathcal{W}} s^* \cup \{\mathtt{b} : \bar{\star}\sigma(p_1)\}$.

(**) Let σ be an $\mathcal{L}$-substitution such that $\{\sigma(p_1), \sigma(p_2)\} \subseteq sub[s_0]$. If $\langle v(\sigma(p_1)), v(\sigma(p_2))\rangle$ satisfies an $\mathcal{L}$-sequent q of the form $\{\mathtt{b} : \bar{\star}p_i\}$ where $\bar{\star} \in \Theta$ and $i \in \{1,2\}$ then $\vdash_G^{\mathcal{W}} s^* \cup \sigma(q)$.
Proof: Suppose that $\langle v(\sigma(p_1)), v(\sigma(p_2))\rangle$ satisfies q. Thus $v(\sigma(p_i))^{\bar{\star}} = \mathtt{b}$. Since

$\sigma(p_i) \in sub[s_0]$, we have that $\mathtt{b} : \bar{\star}\sigma(p_i) \notin s^*$. The maximality of s^* ensures that $\vdash_G^{\mathcal{W}} s^* \cup \sigma(q)$.

We first prove that for every $\mathcal{L}$-formula ψ, we have $v(\psi) \in \mathcal{V}_{\mathcal{M}_G}$. If $\psi \notin sub[s_0]$, this holds by definition. Suppose that $\psi \in sub[s_0]$. We show that $v(\psi)$ respects all the Θ-unary rules of G. Let $r = Q/\{\mathtt{b} : \bar{\star}p_1\}$ be such a Θ-unary rule of G. Suppose that $v(\psi)$ satisfies every $q \in Q$. Let σ be any $\mathcal{L}$-substitution assigning ψ to p_1. By (*), $\vdash_G^{\mathcal{W}} s^* \cup \sigma(q)$ for every $q \in Q$. By applying r we obtain $\vdash_G^{\mathcal{W}} s^* \cup \{\mathtt{b} : \bar{\star}\psi\}$. Hence $v(\psi)$ satisfies $\{\mathtt{b} : \bar{\star}p_1\}$. Thus, $\{\mathtt{b} : \bar{\star}\psi\} \notin s^*$ and so $v(\psi)^{\bar{\star}} = \mathtt{b}$. Next, we show that v respects the truth tables of $\mathcal{M}_G$.

1. Let $\star \in \mathcal{U}_{\mathcal{L}}$ and $\psi \in \mathcal{L}$. We show that $v(\star\psi) \in \star_{\mathcal{M}_G}(v(\psi))$. This holds by definition when $\star\psi \notin sub[s_0]$. Suppose now that $\star\psi \in sub[s_0]$ (and so $\psi \in sub[s_0]$ as well). By the definition of $\mathcal{M}_G$, it suffices to show that $v(\star\psi)^{\bar{\star}} = v(\psi)^{\bar{\star}\star}$ whenever $\bar{\star}\star \in \Theta$. This follows directly from the definition of v.
2. Let $\diamond \in \{\wedge, \vee, \supset\}$ and $\psi_1, \psi_2 \in \mathcal{L}$. We show that $v(\psi_1 \diamond \psi_2) \in \diamond_{\mathcal{M}_G}(v(\psi_1), v(\psi_2))$. This holds by definition when $\psi_1 \diamond \psi_2 \notin sub[s_0]$. Suppose now that $\psi_1 \diamond \psi_2 \in sub[s_0]$ (and so ψ_1 and ψ_2 are in $sub[s_0]$ as well). We prove (i) $v(\psi_1 \diamond \psi_2)^{\epsilon} \in \diamond_{\mathcal{M}_{\mathbf{Cl}^+}}(v(\psi_1)^{\epsilon}, v(\psi_2)^{\epsilon})$ and (ii) $v(\psi_1 \diamond \psi_2)^{\bar{\star}} = \mathtt{b}$ for every Θ-binary rule $r = Q/\{\mathtt{b} : \bar{\star}(p_1 \diamond p_2)\}$ of G for which $\langle v(\psi_1), v(\psi_2)\rangle$ satisfies every $q \in Q$. (i) is straightforward. For (ii), let $r = Q/\{\mathtt{b} : \bar{\star}(p_1 \diamond p_2)\}$ be a Θ-binary rule of G such that $\langle v(\psi_1), v(\psi_2)\rangle$ satisfies every $q \in Q$. Consider an $\mathcal{L}$-substitution σ for which $\sigma(p_1) = \psi_1$ and $\sigma(p_2) = \psi_2$. By (**), for every $q \in Q$ we have $\vdash_G^{\mathcal{W}} s^* \cup \sigma(q)$. By applying r we obtain $\vdash_G^{\mathcal{W}} s^* \cup \{\mathtt{b} : \bar{\star}(\psi_1 \diamond \psi_2)\}$. Thus, $\{\mathtt{b} : \bar{\star}(\psi_1 \diamond \psi_2)\} \notin s^*$ and so $v(\psi_1 \diamond \psi_2)^{\bar{\star}} = \mathtt{b}$.

Finally, it is immediate to see that v does not satisfy s_0. Indeed, every $\mathtt{b} : \psi \in s_0$ occurs also in s^*, and thus $v(\psi)^{\epsilon} \neq \mathtt{b}$. □

Note that the calculus H_0 from Example 32 is Θ-analytic, because its associated PNmatrix is an Nmatrix. In the following example, we show a Θ_H-simple calculus whose extracted semantics is not an Nmatrix:

Example 34. *Let H_1 be the calculus obtained by extending HCL^+ by the axioms below (see Table 6.2) and G_{H_1} be the sequent calculus obtained by adding to LK^+ their corresponding rules:*

$(\mathbf{n_1})$	$p_1 \vee \neg p_1$	$\{\{\mathtt{f} : p_1\}\}/\{\mathtt{t} : \neg p_1\}$
$(\mathbf{b})$	$p_1 \supset (\neg p_1 \supset (\circ p_1 \supset p_2))$	$\{\{\mathtt{t} : p_1\}, \{\mathtt{t} : \neg p_1\}\}/\{\mathtt{f} : \circ p_1\}$
$(\mathbf{k})$	$\circ p_1 \vee (p_1 \wedge \neg p_1)$	$\{\{\mathtt{f} : p_1\}\}/\{\mathtt{t} : \circ p_1\}, \{\{\mathtt{f} : \neg p_1\}\}/\{\mathtt{t} : \circ p_1\}$
$(\mathbf{c})$	$\neg\neg p_1 \supset p_1$	$\{\{\mathtt{f} : \neg\neg p_1\}\}/\{\mathtt{f} : p_1\}$
$(\mathbf{o^1_\wedge})$	$\circ p_1 \supset \circ(p_1 \wedge p_2)$	$\{\{\mathtt{t} : \circ p_1\}\}/\{\mathtt{t} : \circ(p_1 \wedge p_2)\}$
$(\mathbf{o^r_\wedge})$	$(\neg p_1 \vee \neg p_2) \supset \neg(p_1 \wedge p_2)$	$\{\{\mathtt{t} : \neg p_1\}\}/\{\mathtt{t} : \neg(p_1 \wedge p_2)\}$, $\{\{\mathtt{t} : \neg p_2\}\}/\{\mathtt{t} : \neg(p_1 \wedge p_2)\}$

We construct the PNmatrix $\mathcal{M} = \mathcal{M}_{G_{H_1}}$. The set of truth values for $\mathcal{M}$ is:

$$\mathcal{V}_\mathcal{M} = \{\langle \epsilon : \mathtt{f}, \neg : \mathtt{t}, \circ : \mathtt{t}, \neg\neg : \mathtt{f}\rangle, \langle \epsilon : \mathtt{f}, \neg : \mathtt{t}, \circ : \mathtt{t}, \neg\neg : \mathtt{t}\rangle, \langle \epsilon : \mathtt{t}, \neg : \mathtt{f}, \circ : \mathtt{t}, \neg\neg : \mathtt{t}\rangle,$$
$$\langle \epsilon : \mathtt{t}, \neg : \mathtt{t}, \circ : \mathtt{f}, \neg\neg : \mathtt{t}\rangle\}$$

and the set of designated truth values is:

$$\mathcal{D}_\mathcal{M} = \{\langle \epsilon : \mathtt{t}, \neg : \mathtt{f}, \circ : \mathtt{t}, \neg\neg : \mathtt{t}\rangle, \langle \epsilon : \mathtt{t}, \neg : \mathtt{t}, \circ : \mathtt{f}, \neg\neg : \mathtt{t}\rangle\}$$

We only show the truth tables for the unary connective $\neg$ and the binary connective $\wedge$ (we write below $\langle u, v, w, x\rangle$ instead of $\langle \epsilon : u, \neg : v, \circ : w, \neg\neg : x\rangle$; $\mathcal{F}_\mathcal{M} = \mathcal{V}_\mathcal{M} \setminus \mathcal{D}_\mathcal{M}$):

$\neg_\mathcal{M}$	
$\langle \mathtt{f},\mathtt{t},\mathtt{t},\mathtt{f}\rangle$	$\{\langle \mathtt{t},\mathtt{f},\mathtt{t},\mathtt{t}\rangle\}$
$\langle \mathtt{f},\mathtt{t},\mathtt{t},\mathtt{t}\rangle$	$\{\langle \mathtt{t},\mathtt{t},\mathtt{f},\mathtt{t}\rangle\}$
$\langle \mathtt{t},\mathtt{f},\mathtt{t},\mathtt{t}\rangle$	$\{\langle \mathtt{f},\mathtt{t},\mathtt{t},\mathtt{f}\rangle, \langle \mathtt{f},\mathtt{t},\mathtt{t},\mathtt{t}\rangle\}$
$\langle \mathtt{t},\mathtt{t},\mathtt{f},\mathtt{t}\rangle$	$\{\langle \mathtt{t},\mathtt{f},\mathtt{t},\mathtt{t}\rangle\}$

$\wedge_\mathcal{M}$	$\langle \mathtt{f},\mathtt{t},\mathtt{t},\mathtt{f}\rangle$	$\langle \mathtt{f},\mathtt{t},\mathtt{t},\mathtt{t}\rangle$	$\langle \mathtt{t},\mathtt{f},\mathtt{t},\mathtt{t}\rangle$	$\langle \mathtt{t},\mathtt{t},\mathtt{f},\mathtt{t}\rangle$
$\langle \mathtt{f},\mathtt{t},\mathtt{t},\mathtt{f}\rangle$	$\mathcal{F}_\mathcal{M}$	$\mathcal{F}_\mathcal{M}$	$\mathcal{F}_\mathcal{M}$	$\mathcal{F}_\mathcal{M}$
$\langle \mathtt{f},\mathtt{t},\mathtt{t},\mathtt{t}\rangle$	$\mathcal{F}_\mathcal{M}$	$\mathcal{F}_\mathcal{M}$	$\mathcal{F}_\mathcal{M}$	$\mathcal{F}_\mathcal{M}$
$\langle \mathtt{t},\mathtt{f},\mathtt{t},\mathtt{t}\rangle$	$\mathcal{F}_\mathcal{M}$	$\mathcal{F}_\mathcal{M}$	$\{\langle \mathtt{t},\mathtt{f},\mathtt{t},\mathtt{t}\rangle\}$	$\emptyset$
$\langle \mathtt{t},\mathtt{t},\mathtt{f},\mathtt{t}\rangle$	$\mathcal{F}_\mathcal{M}$	$\mathcal{F}_\mathcal{M}$	$\{\langle \mathtt{t},\mathtt{t},\mathtt{f},\mathtt{t}\rangle\}$	$\{\langle \mathtt{t},\mathtt{t},\mathtt{f},\mathtt{t}\rangle\}$

Note that the truth table for $\wedge$ contains an empty spot in the last row. Thus, $\mathcal{M}$ is not an Nmatrix.

Theorem 16 does not apply to Θ-simple calculi G whose associated matrix $\mathcal{M}_G$ is not an Nmatrix. However, we can recover analyticity for such calculi: first, we transform $\mathcal{M}_G$ into a finite family of Nmatrices satisfying the condition of a *simple refinement* (see definition below). Then, for every PNmatrix of this family, we can use the algorithm of [14] to construct an analytic sequent calculus, hence obtaining a *family* of analytic calculi equivalent to G.

Definition 47 ([23])**.** Let $\mathcal{M}$ and $\mathcal{N}$ be PNmatrices for $\mathcal{L}$. We say that $\mathcal{N}$ is a *simple refinement* of $\mathcal{M}$ if $\mathcal{V}_\mathcal{N} \subseteq \mathcal{V}_\mathcal{M}$, $\mathcal{D}_\mathcal{N} = \mathcal{D}_\mathcal{M} \cap \mathcal{V}_\mathcal{N}$, and $\heartsuit_\mathcal{N}(x_1, \ldots, x_n) \subseteq \heartsuit_\mathcal{M}(x_1, \ldots, x_n)$ for every n-ary connective $\heartsuit$ of $\mathcal{L}$ and $x_1, \ldots, x_n \in \mathcal{V}_\mathcal{N}$.

Theorem 17. *For every finite PNmatrix $\mathcal{M}$ for $\mathcal{L}$, there is an algorithm for constructing $\mathcal{M}_1 \ldots \mathcal{M}_n$, such that:*

1. *$\mathcal{M}_1 \ldots \mathcal{M}_n$ are finite simple refinements of $\mathcal{M}$ that are Nmatrices, and*
2. *$\vdash_\mathcal{M} = \bigcap_{i=1,\ldots,n} \vdash_{\mathcal{M}_i}$.*

Proof. Let $\mathcal{M}$ be a PNmatrix for $\mathcal{L}$. Choose $\mathcal{M}_1, \ldots, \mathcal{M}_n$ to be all simple refinements of $\mathcal{M}$ which are Nmatrices. Based on the results in [23], we show that $\vdash_\mathcal{M} = \bigcap_{i=1,\ldots,n} \vdash_{\mathcal{M}_i}$.

"$\Rightarrow$": By Proposition 4.8 in [23], $\vdash_\mathcal{M} \subseteq \vdash_\mathcal{N}$ for every simple refinement $\mathcal{N}$ of $\mathcal{M}$. Therefore, $\vdash_\mathcal{M} \subseteq \bigcap_{i=1,\ldots,n} \vdash_{\mathcal{M}_i}$.

"$\Leftarrow$": Suppose that $\not\vdash_{\mathcal{M}} s$. Thus $v \not\models_{\mathcal{M}} s$ for some $\mathcal{M}$-valuation v for $\mathcal{L}$. Theorem 4.12 in [23] ensures that there exists some $\mathcal{M}_i$, such that v is an $\mathcal{M}_i$-valuation. The fact that $v \not\models_{\mathcal{M}} s$ easily entails that $v \not\models_{\mathcal{M}_i} s$, and so $\not\vdash_{\mathcal{M}_i} s$. □

After constructing a finite family of Nmatrices, the method of [14] can be applied to produce a cut-free sequent calculus $G_{\mathcal{M}}$. This method works for any Nmatrix $\mathcal{M}$ whose set of designated truth values $\mathcal{D}_{\mathcal{M}}$ is a non-empty proper subset of the set of its truth values $\mathcal{V}_{\mathcal{M}}$, provided that its language satisfies the following (slightly reformulated) condition:

Definition 48. Let $\mathcal{M}$ be an Nmatrix for $\mathcal{L}$. We say that $\mathcal{L}$ is *sufficiently expressive* for $\mathcal{M}$ if for any $x \in \mathcal{V}_{\mathcal{M}}$, there exists a set $\mathcal{S}_x$ of $\mathcal{L}$-sequents, each of which has the form $\{\mathtt{b} : \psi\}$, for some $\mathtt{b} \in \{\mathtt{f}, \mathtt{t}\}$ and $\psi \in \mathcal{L}$ in which p_1 is the only atomic variable, such that the following condition holds:

> For any $\mathcal{M}$-valuation v for $\mathcal{L}$ and $\mathcal{L}$-substitution σ, $v(\sigma(p_1)) = x$ iff v satisfies every $\mathcal{L}$-sequent in $\sigma(\mathcal{S}_x)$ with respect to $\mathcal{M}$.

Corollary 7. *For any $H \in \mathbb{H}$, there is an algorithm for constructing a family of Θ_H-analytic sequent calculi $\mathbb{F}_H$, such that for every finite set $\Gamma \cup \{\varphi\}$ of formulas: $\Gamma \vdash_H \varphi$ iff $\vdash_G \Gamma \Rightarrow \varphi$ for every $G \in \mathbb{F}_H$.*

Proof. We start by constructing $\mathcal{M}_{G_H}$. If $\mathcal{D}_{\mathcal{M}_{G_H}} = \emptyset$ or $\mathcal{D}_{\mathcal{M}_{G_H}} = \mathcal{V}_{\mathcal{M}_{G_H}}$, then $\mathcal{M}_{G_H}$ has a trivial corresponding Θ_H-analytic calculus. For the rest of the cases, we can apply Theorem 17 to obtain an equivalent family of Nmatrices. Next we show that $\mathcal{L}$ is sufficiently expressive for any simple refinement of $\mathcal{M}_{G_H}$. Indeed, for $x \in \mathcal{V}_{\mathcal{M}_{G_H}}$, define $\mathcal{S}_x = \{x^{\epsilon} : p_1\} \cup \{x^{\bar{\star}} : \bar{\star} p_1 \mid \bar{\star} \in \Theta_H\}$. Let $\mathcal{M}$ be a simple refinement of $\mathcal{M}_{G_H}$ and let v be an $\mathcal{M}$-valuation for $\mathcal{L}$. The required condition is met by the fact that for every $\bar{\star} \in \Theta_H$ and $\psi \in \mathcal{L}$, $v(\bar{\star}\psi)^{\epsilon} = v(\psi)^{\bar{\star}}$. By the method of [14], we obtain a family of corresponding cut-free calculi. The forms of the $\mathcal{S}_x$-s guarantee that the rules of the obtained calculi are Θ_H-simple, which together with their cut-admissiblity implies Θ_H-analyticity. □

6.6 A Special Case

The procedure described in Section 6.4 and Section 6.5 constructs a sequent calculus and a PNmatrix for any Hilbert calculus $H \in \mathbb{H}$. In this section we consider a subclass $\mathbb{H}_R \subseteq \mathbb{H}$ obtained by restricting the *sequences* of the unary connectives that can occur in an axiom, see the grammar given in Definition 49 below. The subclass contains infinitely many logics, including classical propositional logic or paraconsistent logics belonging to the family of C-systems.

The restriction of the grammar does not affect the algorithm that transforms any $H \in \mathbb{H}_R$ into an equivalent sequent calculus. But step 2 of our general procedure changes, since the construction of the associated PNmatrix will be slightly different. The restriction has the following advantages to its generalized version:

- The PNmatrices constructed for the sequent calculi generated in step 1 will be *smaller* and, thus, *more readable.*
- In the restricted case, we are able to prove that the PNmatrix is an Nmatrix *iff* its associated sequent calculus is "analytic" (see Theorem 19).

Section 6.6.1 contains an *implementation* of this special case for Step 1 and Step 2 and a description of the tool *Paralyzer.*

Let us begin with the definition of the restricted grammar defining the class H_R. Note that we only allow two new unary connectives in $\mathcal{U}_\mathcal{L}$:

Definition 49. Let $\mathcal{U}_\mathcal{L} = \{\star_1, \star_2\}$. $\mathbf{Ax}^R_\mathcal{L}$ is the set of $\mathcal{L}$-formulas that:

1. are generated by the following grammar (I is the initial variable):

$$\begin{array}{ll} I = R_1 \mid R_2 \mid R_3 & \text{for } \diamond \in \{\wedge, \vee, \supset\}, \star, \triangleright \in \mathcal{U}_\mathcal{L} \\ R_1 = (R_1 \diamond P_1) \mid (P_1 \diamond R_1) \mid \star p_1 & P_1 = (P_1 \diamond P_1) \mid \star p_1 \mid p_1 \mid p_2 \\ R_2 = (R_2 \diamond P_2) \mid (P_2 \diamond R_2) \mid \star(p_1 \diamond p_2) & P_2 = (P_2 \diamond P_2) \mid \star p_1 \mid p_1 \mid \star p_2 \mid p_2 \\ R_3 = (R_3 \diamond P_1) \mid (P_1 \diamond R_3) \mid \star \triangleright p_1 & \end{array}$$

2. and satisfy the following conditions: for some subformula $\varphi = \star p_1$ of an $\mathcal{L}$-formula arising from the start symbol R_1 (for the subformula $\varphi = \star(p_1 \diamond p_2)$ of an $\mathcal{L}$-formula arising from R_2 and for the subformula $\varphi = \star \triangleright p_1$ of an $\mathcal{L}$-formula arising from R_3, resp.): φ must not be contained
(a) in a positively occurring (sub)formula of the form $\psi_1 \wedge \psi_2$, and
(b) in a negatively occurring (sub)formula of the form $\psi_1 \vee \psi_2$ or $\psi_1 \supset \psi_2$.

The grammar above is that in Definition 39 with the restriction that there is no nesting of unary connectives in the premises, and the maximum nesting of unary connectives is *one* when binding a formula of the form $(p_1 \diamond p_2)$ and *two*, otherwise, see also the following example.

Example 35. *Consider the axioms from Example 26.*

$$\begin{array}{rl} p_1 \vee \neg p_1 & (\mathbf{n_1}) \\ \circ p_1 \supset \circ \neg p_1 & (\mathbf{a}_\neg) \\ \neg \circ p_1 \supset (p_1 \wedge \neg p_1) & (\mathbf{i}) \\ \neg(p_1 \vee p_2) \supset (\neg p_1 \wedge \neg p_2) & (\mathbf{o}^\mathbf{l}_\vee) \\ (\neg \circ p_1 \wedge \neg p_2) \vee (\neg \circ p_2 \wedge \neg p_1)) \supset \neg \circ (p_1 \vee p_2) & (\mathbf{i}^\mathbf{r}_\vee) \end{array}$$

The axioms $(\mathbf{n_1})$, $(\mathbf{a}_\neg)$, $(\mathbf{i})$ *and* $(\mathbf{o}^\mathbf{l}_\vee)$ *are within* $\boldsymbol{Ax}^R_\mathcal{L}$. *The axiom* $(\mathbf{i}^\mathbf{r}_\vee)$ *is not covered by the restricted grammar (it is within* $\boldsymbol{Ax}_\mathcal{L}$*, though).*

As a consequence, we do not need the set Θ anymore and instead distinguish between *three types of rules* that are generated by our algorithm in Theorem 14 (the definition is contained in Table 6.6).

	Rule	Application form
$\mathcal{U}_{\mathcal{L}}$-unary-one	$\mathcal{P}/\{\mathtt{t} : \star p_1\}$	$\dfrac{\Gamma, \odot_1\varphi \Rightarrow \Delta \ \dots \ \Gamma, \odot_n\varphi \Rightarrow \Delta \ \Gamma \Rightarrow \bullet_1\varphi, \Delta \ \dots \ \Gamma \Rightarrow \bullet_m\varphi, \Delta}{\Gamma \Rightarrow \star\varphi, \Delta}$
	$\mathcal{P}/\{\mathtt{f} : \star p_1\}$	$\dfrac{\Gamma, \odot_1\varphi \Rightarrow \Delta \ \dots \ \Gamma, \odot_n\varphi \Rightarrow \Delta \ \Gamma \Rightarrow \bullet_1\varphi, \Delta \ \dots \ \Gamma \Rightarrow \bullet_m\varphi, \Delta}{\Gamma, \star\varphi \Rightarrow \Delta}$
	where $\mathcal{P} = \{\{\mathtt{f} : \odot_1 p_1\}, \dots, \{\mathtt{f} : \odot_n p_1\}, \{\mathtt{t} : \bullet_1 p_1\}, \dots, \{\mathtt{t} : \bullet_m p_1\}\}$	
$\mathcal{U}_{\mathcal{L}}$-unary-two	$\mathcal{P}/\{\mathtt{t} : \star \rhd p_1\}$	$\dfrac{\Gamma, \odot_1\varphi \Rightarrow \Delta \ \dots \ \Gamma, \odot_n\varphi \Rightarrow \Delta \ \Gamma \Rightarrow \bullet_1\varphi, \Delta \ \dots \ \Gamma \Rightarrow \bullet_m\varphi, \Delta}{\Gamma \Rightarrow \star \rhd \varphi, \Delta}$
	$\mathcal{P}/\{\mathtt{f} : \star \rhd p_1\}$	$\dfrac{\Gamma, \odot_1\varphi \Rightarrow \Delta \ \dots \ \Gamma, \odot_n\varphi \Rightarrow \Delta \ \Gamma \Rightarrow \bullet_1\varphi, \Delta \ \dots \ \Gamma \Rightarrow \bullet_m\varphi, \Delta}{\Gamma, \star \rhd \varphi \Rightarrow \Delta}$
	where $\mathcal{P} = \{\{\mathtt{f} : \odot_1 p_1\}, \dots, \{\mathtt{f} : \odot_n p_1\}, \{\mathtt{t} : \bullet_1 p_1\}, \dots, \{\mathtt{t} : \bullet_m p_1\}\}$	
$\mathcal{U}_{\mathcal{L}}$-binary	$\mathcal{Q}/\{\mathtt{t} : \star(p_1 \diamond p_2)\}$	$\dfrac{\begin{array}{ccc}\Gamma, \odot_1\varphi_{i_1} \Rightarrow \Delta & \dots & \Gamma, \odot_n\varphi_{i_n} \Rightarrow \Delta \\ \Gamma \Rightarrow \bullet_1\varphi_{j_1}, \Delta & \dots & \Gamma \Rightarrow \bullet_m\varphi_{j_m}, \Delta\end{array}}{\Gamma \Rightarrow \star(\varphi_1 \diamond \varphi_2), \Delta}$
	$\mathcal{Q}/\{\mathtt{f} : \star(p_1 \diamond p_2)\}$	$\dfrac{\begin{array}{ccc}\Gamma, \odot_1\varphi_{i_1} \Rightarrow \Delta & \dots & \Gamma, \odot_n\varphi_{i_n} \Rightarrow \Delta \\ \Gamma \Rightarrow \bullet_1\varphi_{j_1}, \Delta & \dots & \Gamma \Rightarrow \bullet_m\varphi_{j_m}, \Delta\end{array}}{\Gamma, \star(\varphi_1 \diamond \varphi_2) \Rightarrow \Delta}$
	where $\mathcal{Q} = \{\{\mathtt{f} : \odot_1 p_{i_1}\}, \dots, \{\mathtt{f} : \odot_n p_{i_n}\}, \{\mathtt{t} : \bullet_1 p_{j_1}\}, \dots, \{\mathtt{t} : \bullet_m p_{j_m}\}\}$	

Table 6.6: The general form of our rules ($\mathtt{b} \in \{\mathtt{f}, \mathtt{t}\}$, $\star, \rhd \in \mathcal{U}_{\mathcal{L}}$, $\odot_i, \bullet_j \in \mathcal{U}_{\mathcal{L}} \cup \{\epsilon\}$, $\diamond \in \{\wedge, \vee, \supset\}$, $i_1, \dots, i_n, j_1, \dots, j_m \in \{1, 2\}$)

Definition 50. An $\mathcal{L}$-rule Q/s is called $\mathcal{U}_{\mathcal{L}}$-*simple* if it is either a $\mathcal{U}_{\mathcal{L}}$-unary-one, $\mathcal{U}_{\mathcal{L}}$-unary-two, or a $\mathcal{U}_{\mathcal{L}}$-binary rule. A sequent calculus for $\mathcal{L}$ is called $\mathcal{U}_{\mathcal{L}}$-*simple* if it is obtained by augmenting LK^+ with a finite set of $\mathcal{U}_{\mathcal{L}}$-simple $\mathcal{L}$-rules. We shall omit $\mathcal{U}_{\mathcal{L}}$ when it is clear from the context.

Definition 51. $\mathbb{H}_R$ is the family of Hilbert calculi obtained by extending HCL^+ with any finite set of axioms from $\mathbf{Ax}^R_{\mathcal{L}}$ for some language $\mathcal{L}$. $\mathbb{H}_R$ is properly contained in $\mathbb{H}$.

$\mathbb{H}_R$ also includes well-known Hilbert calculi, e.g.:

- the standard calculus for (propositional) classical logic **Cl** (that is obtained by adding the axioms ($\mathbf{n_1}$) and ($\mathbf{n_2}$) from Table 6.2, page 104 to HCL^+).
- the Hilbert calculi for the logics **B**, **BK**, **bC**, **Ci** or $\mathbf{C}_1$ from page 104.

- the Hilbert calculi for other C-systems that are defined by adding to HCL^+ the axioms (**b**) and ($\mathbf{n_1}$), as well as different subsets of the axioms ($\mathbf{r}_\diamond$), (**k**), (**i**), ($\mathbf{o}^{\mathbf{1}}_\diamond$), ($\mathbf{o}^{\mathbf{2}}_\diamond$), ($\mathbf{a}_\neg$), (**e**), ($\mathbf{a}_\diamond$) and (**c**) from Table 6.2.

In contrast to $\mathbb{H}$, the family $\mathbb{H}_R$ does for example *not* contain

- the Hilbert calculus for the logic **LFI1** from page 104,
- the Hilbert calculi for other C-systems that are defined by adding to HCL^+ (amongst others) the axioms ($\mathbf{i}^{\mathbf{l}}_\diamond$) or ($\mathbf{i}^{\mathbf{r}}_\diamond$) of Table 6.2.
- the Hilbert calculi for the logics $\mathbf{L2^{n+2}}$ for each $n \geq 0$ discussed in [106].

Corollary 8. *Let $H \in \mathbb{H}_R$ be a Hilbert calculus for $\mathcal{L}$. The algorithm in Theorem 14 constructs an equivalent $\mathcal{U}_\mathcal{L}$-simple sequent calculus G_H for $\mathcal{L}$.*

We now show how a PNmatrix $\mathcal{M}_G$ is extracted from a $\mathcal{U}_\mathcal{L}$-simple sequent calculus G. The main difference between the general and the restricted case is the information that is coded in a truth value: In the general case, a truth value consists of functions of all elements of Θ, while in the restricted case, we take the elements of $\mathcal{U}_\mathcal{L} \cup \{\epsilon\}$, which is actually a subset of Θ. The truth values thus do not say anything about the *sequences* of the unary connectives, which makes them smaller in size. We use $\mathsf{F}_{\mathcal{U}_\mathcal{L}}$ to denote the set of functions $\{\mathcal{U}_\mathcal{L} \cup \{\epsilon\}\} \to \{\mathtt{f}, \mathtt{t}\}$.

Example 36. *Let $H \in \mathbb{H}_R$ be the Hilbert system defined by extending HCL^+ with the axioms ($\mathbf{n_1}$) $p_1 \vee \neg p_1$, (**c**) $\neg\neg p_1 \supset p_1$ and ($\mathbf{o}^{\mathbf{1}}_\wedge$) $\neg(p_1 \wedge p_2) \supset (\neg p_1 \vee \neg p_2)$ from Table 6.2. Then $\mathcal{U}_\mathcal{L} = \{\neg\}$ and $\Theta = \{\epsilon, \neg, \neg\neg\}$.*

F_Θ consists of truth values of the form $\langle \epsilon : \mathtt{b}, \neg : \mathtt{b}, \neg\neg : \mathtt{b}\rangle$ whereas $\mathsf{F}_{\mathcal{U}_\mathcal{L}}$ consists of truth values of the form $\langle \epsilon : \mathtt{b}, \neg : \mathtt{b}\rangle$.

We now show the algorithm to extract a PNmatrix out of any given $\mathcal{U}_\mathcal{L}$-simple sequent calculus. It is slightly different from the one for the general case, based on the following observations:

- We only consider those truth values from all the possible functions in $\mathsf{F}_{\mathcal{U}_\mathcal{L}}$ that respect the $\mathcal{U}_\mathcal{L}$*-unary-one* rules (cf. Table 6.6).
- The truth tables of the unary connectives are constructed using the $\mathcal{U}_\mathcal{L}$*-unary-two* rules (cf. Table 6.6).
- $\mathcal{U}_\mathcal{L}$*-binary* rules (cf. Table 6.6) determine the truth tables of the binary connectives.

Note that the notions for satisfaction of an $\mathcal{L}$-sequent and respecting a Θ-unary rule from Definition 44 are the same for the restricted case.

Definition 52. Given a $\mathcal{U}_\mathcal{L}$-simple sequent calculus G, the PNmatrix $\mathcal{M}_G$ is defined as follows:

- The set of truth values $\mathcal{V}_{\mathcal{M}_G}$ contains all functions in $\mathsf{F}_{\mathcal{U}_\mathcal{L}}$ that respect all $\mathcal{U}_\mathcal{L}$-unary-one rules of G.
- The set of designated truth values $\mathcal{D}_{\mathcal{M}_G}$ is $\{u \in \mathcal{V}_{\mathcal{M}_G} \mid u^\epsilon = \mathtt{t}\}$.

- For any unary connective $\star \in \mathcal{U}_\mathcal{L}$, the truth table for $\star$ is given by:
  1. $\star_{\mathcal{M}_G}(u_1) = \{u \in \mathcal{V}_{\mathcal{M}_G} \mid u^\epsilon = u_1^\star\}$ and
  2. for every $\mathcal{U}_\mathcal{L}$-unary-two rule of G of the form $Q/\{\mathtt{b} : \triangleright \star p_1\}$, if u_1 satisfies every $q \in Q$ then $u^\triangleright = \mathtt{b}$.
- For $\diamond \in \{\wedge, \vee, \supset\}$ and $u_1, u_2 \in \mathcal{V}_{\mathcal{M}_G}$, $\diamond_{\mathcal{M}_G}(u_1, u_2)$ is the set of all $u \in \mathcal{V}_{\mathcal{M}_G}$ satisfying:
  1. $u^\epsilon \in \diamond_{\mathcal{M}_{\mathbf{Cl}^+}}(u_1^\epsilon, u_2^\epsilon)$ (where $\diamond_{\mathcal{M}_{\mathbf{Cl}^+}}$ is the classical truth table of $\diamond$; see Example 24).
  2. For every $\mathcal{U}_\mathcal{L}$-binary rule of G of the form $Q/\{\mathtt{b} : \star(p_1 \diamond p_2)\}$, if $\langle u_1, u_2\rangle$ satisfies every $q \in Q$ then $u^\star = \mathtt{b}$.

Example 37. *Consider the calculus H_R that extends HCL^+ with the three axioms from Example 36, with $\mathcal{U}_\mathcal{L} = \{\neg\}$. The corresponding $\mathcal{U}_\mathcal{L}$-simple sequent calculus G_{H_R} has one unary-one rule*

$$r_{u_1} = \{\{\mathtt{f} : p_1\}\}/\{\mathtt{t} : \neg p_1\}$$

one unary-two rule

$$r_{u_2} = \{\{\mathtt{f} : p_1\}\}/\{\mathtt{f} : \neg\neg p_1\}$$

and one binary rule

$$r_b = \{\{\mathtt{f} : \neg p_1\}, \{\mathtt{f} : \neg p_2\}\}/\{\mathtt{f} : \neg(p_1 \wedge p_2)\}$$

We construct the PNmatrix $\mathcal{M} = \mathcal{M}_{G_{H_R}}$ according to Definition 52 and start by listing $\mathsf{F}_{\mathcal{U}_\mathcal{L}}$:

$$\mathsf{F}_{\mathcal{U}_\mathcal{L}} = \{\langle \epsilon : \mathtt{f}, \neg : \mathtt{f}\rangle, \langle \epsilon : \mathtt{f}, \neg : \mathtt{t}\rangle, \langle \epsilon : \mathtt{t}, \neg : \mathtt{f}\rangle, \langle \epsilon : \mathtt{t}, \neg : \mathtt{t}\rangle\}$$

We first determine the set $\mathcal{V}_\mathcal{M}$ of truth values that respect the unary-one rules of G_{H_R}. The only relevant rule in this case is r_{u_1}. Since $u \in \mathcal{V}_\mathcal{M}$ respects r_{u_1} iff

$$u^\neg = \mathtt{t} \text{ whenever } u^\epsilon = \mathtt{f}$$

we delete the value $\langle \epsilon : \mathtt{f}, \neg : \mathtt{f}\rangle$ and obtain:

$$\mathcal{V}_\mathcal{M} = \{\langle \epsilon : \mathtt{f}, \neg : \mathtt{t}\rangle, \langle \epsilon : \mathtt{t}, \neg : \mathtt{f}\rangle, \langle \epsilon : \mathtt{t}, \neg : \mathtt{t}\rangle\}$$

The set of designated truth values is:

$$\mathcal{D}_\mathcal{M} = \{\langle \epsilon : \mathtt{t}, \neg : \mathtt{f}\rangle, \langle \epsilon : \mathtt{t}, \neg : \mathtt{t}\rangle\}$$

Next we define the truth table for $\neg$. For every $u_1 \in \mathcal{V}_\mathcal{M}$, we take all $u \in \mathcal{V}_\mathcal{M}$ that (1) satisfy the condition

$$u^\epsilon = u_1^\neg$$

and (2) respect the unary-two rule r_{u_2}. For instance, let $u_1 = \langle \epsilon : \mathtt{f}, \neg : \mathtt{t}\rangle$. The only elements satisfying condition (1) are

$$\{\langle \epsilon : \mathtt{t}, \neg : \mathtt{f}\rangle, \langle \epsilon : \mathtt{t}, \neg : \mathtt{t}\rangle\}$$

where $\langle \epsilon : \mathtt{t}, \neg : \mathtt{t}\rangle$ *does not respect* r_{u_2} *and is hence removed. The truth table for* $\neg$ *is then constructed as follows (we write below* $\langle x, y\rangle$ *instead of* $\langle \epsilon : x, \neg : y\rangle$*):*

$\neg_\mathcal{M}$	
$\langle \mathtt{f}, \mathtt{t}\rangle$	$\{\langle \mathtt{t}, \mathtt{f}\rangle, \langle \mathtt{t}, \mathtt{t}\rangle\}$
$\langle \mathtt{t}, \mathtt{f}\rangle$	$\{\langle \mathtt{f}, \mathtt{t}\rangle\}$
$\langle \mathtt{t}, \mathtt{t}\rangle$	$\{\langle \mathtt{t}, \mathtt{f}\rangle, \langle \mathtt{t}, \mathtt{t}\rangle\}$

$\longrightarrow_{(r_{u_2})}$

$\neg_\mathcal{M}$	
$\langle \mathtt{f}, \mathtt{t}\rangle$	$\{\langle \mathtt{t}, \mathtt{f}\rangle\}$
$\langle \mathtt{t}, \mathtt{f}\rangle$	$\{\langle \mathtt{f}, \mathtt{t}\rangle\}$
$\langle \mathtt{t}, \mathtt{t}\rangle$	$\{\langle \mathtt{t}, \mathtt{f}\rangle, \langle \mathtt{t}, \mathtt{t}\rangle\}$

Finally, we obtain the truth tables for the binary connectives which must meet the requirements arising (1) from the classical truth tables and (2) from the binary rules of G_{H_R}. *We show the case of* $\wedge$*, as* r_b *involves* $\wedge$*. Condition (1) gives us the following truth table for* $\wedge$*:*

$\wedge_\mathcal{M}$	$\langle \mathtt{f}, \mathtt{t}\rangle$	$\langle \mathtt{t}, \mathtt{f}\rangle$	$\langle \mathtt{t}, \mathtt{t}\rangle$
$\langle \mathtt{f}, \mathtt{t}\rangle$	$\{\langle \mathtt{f}, \mathtt{t}\rangle\}$	$\{\langle \mathtt{f}, \mathtt{t}\rangle\}$	$\{\langle \mathtt{f}, \mathtt{t}\rangle\}$
$\langle \mathtt{t}, \mathtt{f}\rangle$	$\{\langle \mathtt{f}, \mathtt{t}\rangle\}$	$\{\langle \mathtt{t}, \mathtt{f}\rangle, \langle \mathtt{t}, \mathtt{t}\rangle\}$	$\{\langle \mathtt{t}, \mathtt{f}\rangle, \langle \mathtt{t}, \mathtt{t}\rangle\}$
$\langle \mathtt{t}, \mathtt{t}\rangle$	$\{\langle \mathtt{f}, \mathtt{t}\rangle\}$	$\{\langle \mathtt{t}, \mathtt{f}\rangle, \langle \mathtt{t}, \mathtt{t}\rangle\}$	$\{\langle \mathtt{t}, \mathtt{f}\rangle, \langle \mathtt{t}, \mathtt{t}\rangle\}$

By condition (2), r_b *imposes the requirement that for every* $u \in \wedge_\mathcal{M}(u_1, u_2)$ *we have*

$$u^\neg = \mathtt{f} \text{ whenever } u_1^\neg = \mathtt{f} \text{ and } u_2^\neg = \mathtt{f}$$

Thus we obtain the following truth table for $\wedge$*:*

$\wedge_\mathcal{M}$	$\langle \mathtt{f}, \mathtt{t}\rangle$	$\langle \mathtt{t}, \mathtt{f}\rangle$	$\langle \mathtt{t}, \mathtt{t}\rangle$
$\langle \mathtt{f}, \mathtt{t}\rangle$	$\{\langle \mathtt{f}, \mathtt{t}\rangle\}$	$\{\langle \mathtt{f}, \mathtt{t}\rangle\}$	$\{\langle \mathtt{f}, \mathtt{t}\rangle\}$
$\langle \mathtt{t}, \mathtt{f}\rangle$	$\{\langle \mathtt{f}, \mathtt{t}\rangle\}$	$\{\langle \mathtt{t}, \mathtt{f}\rangle\}$	$\{\langle \mathtt{t}, \mathtt{f}\rangle, \langle \mathtt{t}, \mathtt{t}\rangle\}$
$\langle \mathtt{t}, \mathtt{t}\rangle$	$\{\langle \mathtt{f}, \mathtt{t}\rangle\}$	$\{\langle \mathtt{t}, \mathtt{f}\rangle, \langle \mathtt{t}, \mathtt{t}\rangle\}$	$\{\langle \mathtt{t}, \mathtt{f}\rangle, \langle \mathtt{t}, \mathtt{t}\rangle\}$

As comparison to the matrix that is produced in the restricted case, in the following example we show the PNmatrix that is extracted for the same calculus by the more general procedure of Section 6.5:

Example 38. *Let* $\Theta = \{\epsilon, \neg, \neg\neg\}$*. We show the PNmatrix* $\mathcal{M}_\Theta$ *for the sequent calculus* G_{H_R} *according to Definition 37.*

The set of truth values (respecting the Θ*-unary rules* r_{u_1} *and* r_{u_2}*) is*

$$\mathcal{V}_{\mathcal{M}_\Theta} = \{\langle \epsilon : \mathtt{f}, \neg : \mathtt{t}, \neg\neg : \mathtt{f}\rangle, \langle \epsilon : \mathtt{t}, \neg : \mathtt{f}, \neg\neg : \mathtt{f}\rangle, \langle \epsilon : \mathtt{t}, \neg : \mathtt{f}, \neg\neg : \mathtt{t}\rangle, \langle \epsilon : \mathtt{t}, \neg : \mathtt{t}, \neg\neg : \mathtt{f}\rangle, \langle \epsilon : \mathtt{t}, \neg : \mathtt{t}, \neg\neg : \mathtt{t}\rangle\}$$

The set of designated truth values is thus:

$$\mathcal{D}_{\mathcal{M}_\Theta} = \{\langle \epsilon : \mathtt{t}, \neg : \mathtt{f}, \neg\neg : \mathtt{f}\rangle, \langle \epsilon : \mathtt{t}, \neg : \mathtt{f}, \neg\neg : \mathtt{t}\rangle, \langle \epsilon : \mathtt{t}, \neg : \mathtt{t}, \neg\neg : \mathtt{f}\rangle, \langle \epsilon : \mathtt{t}, \neg : \mathtt{t}, \neg\neg : \mathtt{t}\rangle\}$$

The truth table for $\neg$ *is as follows (we write below* $\langle x, y, z\rangle$ *instead of* $\langle \epsilon : x, \neg : y, \neg\neg : z\rangle$*):*

$\neg_{\mathcal{M}_\Theta}$	
$\langle \mathtt{f}, \mathtt{t}, \mathtt{f} \rangle$	$\{\langle \mathtt{t}, \mathtt{f}, \mathtt{f} \rangle, \langle \mathtt{t}, \mathtt{f}, \mathtt{t} \rangle\}$
$\langle \mathtt{t}, \mathtt{f}, \mathtt{f} \rangle$	$\emptyset$
$\langle \mathtt{t}, \mathtt{f}, \mathtt{t} \rangle$	$\{\langle \mathtt{f}, \mathtt{t}, \mathtt{f} \rangle\}$
$\langle \mathtt{t}, \mathtt{t}, \mathtt{f} \rangle$	$\{\langle \mathtt{t}, \mathtt{f}, \mathtt{f} \rangle, \langle \mathtt{t}, \mathtt{f}, \mathtt{t} \rangle\}$
$\langle \mathtt{t}, \mathtt{t}, \mathtt{t} \rangle$	$\{\langle \mathtt{t}, \mathtt{t}, \mathtt{f} \rangle, \langle \mathtt{t}, \mathtt{t}, \mathtt{t} \rangle\}$

and the truth table for $\wedge$ (respecting the Θ-binary rule r_b):

$\wedge_{\mathcal{M}_\Theta}$	$\langle \mathtt{f}, \mathtt{t}, \mathtt{f} \rangle$	$\langle \mathtt{t}, \mathtt{f}, \mathtt{f} \rangle$	$\langle \mathtt{t}, \mathtt{f}, \mathtt{t} \rangle$	$\langle \mathtt{t}, \mathtt{t}, \mathtt{f} \rangle$	$\langle \mathtt{t}, \mathtt{t}, \mathtt{t} \rangle$
$\langle \mathtt{f}, \mathtt{t}, \mathtt{f} \rangle$	$\langle \mathtt{f}, \mathtt{t}, \mathtt{f} \rangle$	$\langle \mathtt{f}, \mathtt{t}, \mathtt{f} \rangle$	$\langle \mathtt{f}, \mathtt{t}, \mathtt{f} \rangle$	$\langle \mathtt{f}, \mathtt{t}, \mathtt{f} \rangle$	$\langle \mathtt{f}, \mathtt{t}, \mathtt{f} \rangle$
$\langle \mathtt{t}, \mathtt{f}, \mathtt{f} \rangle$	$\langle \mathtt{f}, \mathtt{t}, \mathtt{f} \rangle$	$\{\langle \mathtt{t}, \mathtt{f}, \mathtt{f} \rangle, \langle \mathtt{t}, \mathtt{f}, \mathtt{t} \rangle\}$	$\{\langle \mathtt{t}, \mathtt{f}, \mathtt{f} \rangle, \langle \mathtt{t}, \mathtt{f}, \mathtt{t} \rangle\}$	$\mathcal{D}_\Theta$	$\mathcal{D}_\Theta$
$\langle \mathtt{t}, \mathtt{f}, \mathtt{t} \rangle$	$\langle \mathtt{f}, \mathtt{t}, \mathtt{f} \rangle$	$\{\langle \mathtt{t}, \mathtt{f}, \mathtt{f} \rangle, \langle \mathtt{t}, \mathtt{f}, \mathtt{t} \rangle\}$	$\{\langle \mathtt{t}, \mathtt{f}, \mathtt{f} \rangle, \langle \mathtt{t}, \mathtt{f}, \mathtt{t} \rangle\}$	$\mathcal{D}_\Theta$	$\mathcal{D}_\Theta$
$\langle \mathtt{t}, \mathtt{t}, \mathtt{f} \rangle$	$\langle \mathtt{f}, \mathtt{t}, \mathtt{f} \rangle$	$\mathcal{D}_\Theta$	$\mathcal{D}_\Theta$	$\mathcal{D}_\Theta$	$\mathcal{D}_\Theta$
$\langle \mathtt{t}, \mathtt{t}, \mathtt{t} \rangle$	$\langle \mathtt{f}, \mathtt{t}, \mathtt{f} \rangle$	$\mathcal{D}_\Theta$	$\mathcal{D}_\Theta$	$\mathcal{D}_\Theta$	$\mathcal{D}_\Theta$

It can easily be seen that the PNmatrix $\mathcal{M}$ from Example 37 is easier to read and smaller than the PNmatrix $\mathcal{M}_\Theta$. Note that, while $\mathcal{M}$ is in fact an Nmatrix, the PNmatrix $\mathcal{M}_\Theta$ contains an empty spot in the truth table of the connective $\neg$.

The soundness and completeness proof in the restricted case is similar to the soundness and completeness proof of the general case. Below we only give the statement of the theorem; the proof can be found in Appendix B.

The main difference from the general to the restricted case is that in the latter we can prove that a PNmatrix is an Nmatrix *iff* it is $\mathcal{U}_\mathcal{L}$-analytic (recall Theorem 16, where we only prove that *if* the PNmatrix is an Nmatrix, it is Θ-analytic). The proof for analyticity in the restricted case requires a more complicated definition of ($\mathcal{U}_\mathcal{L}$-)analyticity, which is reflected in the introduction of the two sets $\mathcal{U}^-_\mathcal{L}(\mathcal{W})$ and $\mathcal{U}^+_\mathcal{L}(\mathcal{W})$ and a weaker notion of "satisfaction".

We now denote by $\mathcal{W}$ an arbitrary set of $\mathcal{L}_\mathcal{U}$-formulas closed under subformulas. Moreover, an $\mathcal{L}$-sequent s is called a $\mathcal{W}$*-sequent* if $sub[s] \subseteq \mathcal{W}$. The set $\mathcal{U}^-_\mathcal{L}(\mathcal{W})$ contains all formulas of $\mathcal{W}$ *except* formulas with an outermost unary connective that are not a proper subformula of another formula in $\mathcal{W}$. The set $\mathcal{U}^+_\mathcal{L}(\mathcal{W})$ contains the formulas of $\mathcal{W}$ and all formulas of $\mathcal{U}^-_\mathcal{L}(\mathcal{W})$ extended with the unary connectives of $\mathcal{U}_\mathcal{L}$. $\mathcal{U}^+_\mathcal{L}(\mathcal{W})$ hence contains the "extended material" we will use to define our notion of $\mathcal{U}_\mathcal{L}$-analyticity.

Definition 53. The sets $\mathcal{U}^+_\mathcal{L}(\mathcal{W})$ and $\mathcal{U}^-_\mathcal{L}(\mathcal{W})$ are defined as follows:

$\mathcal{U}^-_\mathcal{L}(\mathcal{W}) = \mathcal{W} \setminus \{\star\psi \in \mathcal{W} \mid \star \in \mathcal{U}_\mathcal{L}, \star\psi \text{ is not a proper subformula of a formula in } \mathcal{W}\}$
$\mathcal{U}^+_\mathcal{L}(\mathcal{W}) = \mathcal{W} \cup \{\star\psi \mid \star \in \mathcal{U}_\mathcal{L}, \psi \in \mathcal{U}^-_\mathcal{L}(\mathcal{W})\}$

A $\mathcal{U}^-_\mathcal{L}(\mathcal{W})$-valuation $v : \mathcal{U}^-_\mathcal{L}(\mathcal{W}) \to \mathsf{F}_{\mathcal{U}_\mathcal{L}}$ *w-satisfies* a $\mathcal{U}^+_\mathcal{L}(\mathcal{W})$-sequent s if there exists some labelled formula $\mathtt{b} : \psi \in s$, such that either

(i) ψ does not have the form $\star\varphi$ and $v(\psi)^\epsilon = \mathtt{b}$, or
(ii) $\psi = \star\varphi$ (for some $\star \in \mathcal{U}_\mathcal{L}$ and $\varphi \in \mathcal{U}^-_\mathcal{L}(\mathcal{W})$) and $v(\varphi)^\star = \mathtt{b}$.

Note that $\psi \in \mathcal{U}_{\mathcal{L}}^{-}(\mathcal{W})$ whenever $\star\psi \in \mathcal{U}_{\mathcal{L}}^{+}(\mathcal{W})$ for some $\star \in \mathcal{U}_{\mathcal{L}}$.

Example 39. *For $\mathcal{U}_{\mathcal{L}} = \{\neg\}$ and $\mathcal{W} = \{p_1, p_2, \neg p_1, \neg p_2, p_1 \vee p_2, \neg p_1 \vee p_2, \neg(p_1 \vee p_2)\}$, we have*

- $\mathcal{U}_{\mathcal{L}}^{-}(\mathcal{W}) = \{p_1, p_2, \neg p_1, p_1 \vee p_2, \neg p_1 \vee p_2\}$, *and*
- $\mathcal{U}_{\mathcal{L}}^{+}(\mathcal{W}) = \mathcal{W} \cup \{\neg\neg p_1, \neg(\neg p_1 \vee p_2)\}$.

We can now prove soundness and completeness of our calculi (the proofs can be found in Appendix B).

Theorem 18 (Soundness and Completeness)**.** *Let s_0 be a $\mathcal{W}$-sequent and G be a $\mathcal{U}_{\mathcal{L}}$-simple sequent calculus for $\mathcal{L}$. Then, $\vdash_G^{\mathcal{U}_{\mathcal{L}}^{+}(\mathcal{W})} s_0$ (cf. Definition 46) iff every $\mathcal{M}_G$-$\mathcal{U}_{\mathcal{L}}^{-}(\mathcal{W})$-valuation w-satisfies s_0.*

Corollary 9. *For every $\mathcal{L}$-sequent s, $\vdash_G s$ iff $\vdash_{\mathcal{M}_G} s$.*

We now define $\mathcal{U}_{\mathcal{L}}$-analyticity by using the set $\mathcal{U}_{\mathcal{L}}$: we call a sequent calculus $\mathcal{U}_{\mathcal{L}}$-analytic when for every provable sequent s, we can find a proof where the "material" occurring in its proofs consists only of subformulas of the set $\mathcal{U}_{\mathcal{L}}^{+}$:

Definition 54. A $\mathcal{U}_{\mathcal{L}}$-simple calculus G is *$\mathcal{U}_{\mathcal{L}}$-analytic* if for every $\mathcal{L}$-sequent s: $\vdash_G s$ implies that $\vdash_G^{\mathcal{U}_{\mathcal{L}}^{+}(sub[s])} s$.

Moreover, we can now show that we can use $\mathcal{M}_G$ to check whether G is $\mathcal{U}_{\mathcal{L}}$-analytic (and vice versa; for the proof see Appendix B):

Theorem 19. *A $\mathcal{U}_{\mathcal{L}}$-simple calculus G is $\mathcal{U}_{\mathcal{L}}$-analytic iff $\mathcal{M}_G$ is an Nmatrix.*

Recall the calculus G_{H_R} and its PNmatrix $\mathcal{M}$ from Example 37. Since the PNmatrix $\mathcal{M}$ does not contain an empty set in its truth tables for the connectives (and is an Nmatrix), G_{H_R} is $\mathcal{U}_{\mathcal{L}}$-analytic by Theorem 19. The PNmatrix $\mathcal{M}_\Theta$ for G_{H_R} extracted by the procedure in the general case (see Example 38), however contains an empty set in the truth table for the connective $\neg$. The calculus G_{H_R} is thus an example of an analytic calculus for which the general procedure does *not* extract an Nmatrix. Note that this does not contradict our result for the general case, since we show in Theorem 16 only that *if* the extracted PNmatrix is an Nmatrix, the corresponding sequent calculus is Θ-analytic. We leave the further analysis of this example and its implication as a question for future research.

Finally, we show the PNmatrix generated in the more restricted case for the calculus H_1 of Example 34:

Example 40. *Let H_1 be the calculus obtained by extending HCL^+ with the axioms of Example 34, namely $(\mathbf{n_1})$, $(\mathbf{b})$, $(\mathbf{k})$, $(\mathbf{c})$, $(\mathbf{o_\wedge^1})$ and $(\mathbf{o_\wedge^r})$. Let G_{H_1} be the sequent calculus obtained by extending LK^+ with the following rules corresponding to the axioms:*

$(\mathbf{n_1})$	$\{\{\mathtt{f}:p_1\}\}/\{\mathtt{t}:\neg p_1\}$
$(\mathbf{b})$	$\{\{\mathtt{t}:p_1\},\{\mathtt{t}:\neg p_1\}\}/\{\mathtt{f}:\circ p_1\}$
$(\mathbf{k})$	$\{\{\mathtt{f}:p_1\}\}/\{\mathtt{t}:\circ p_1\},\{\{\mathtt{f}:\neg p_1\}\}/\{\mathtt{t}:\circ p_1\}$
$(\mathbf{c})$	$\{\{\mathtt{f}:\neg\neg p_1\}\}/\{\mathtt{f}:p_1\}$
$(\mathbf{o}^{\mathbf{1}}_{\wedge})$	$\{\{\mathtt{t}:\circ p_1\}\}/\{\mathtt{t}:\circ(p_1\wedge p_2)\}$
$(\mathbf{o}^{\mathbf{r}}_{\wedge})$	$\{\{\mathtt{t}:\neg p_1\}\}/\{\mathtt{t}:\neg(p_1\wedge p_2)\},\{\{\mathtt{t}:\neg p_2\}\}/\{\mathtt{t}:\neg(p_1\wedge p_2)\}$

Let $\mathcal{M}' = \mathcal{M}'_{G_{H_1}}$ be the PNmatrix obtained according to Definition 52. The set of truth values for $\mathcal{M}'$ is

$$\mathcal{V}_{\mathcal{M}} = \{\langle\epsilon:\mathtt{f},\neg:\mathtt{t},\circ:\mathtt{t}\rangle,\langle\epsilon:\mathtt{t},\neg:\mathtt{f},\circ:\mathtt{t}\rangle,\langle\epsilon:\mathtt{t},\neg:\mathtt{t},\circ:\mathtt{f}\rangle\}$$

and the set of designated truth values:

$$\mathcal{D}_{\mathcal{M}} = \{\langle\epsilon:\mathtt{t},\neg:\mathtt{f},\circ:\mathtt{t}\rangle,\langle\epsilon:\mathtt{t},\neg:\mathtt{t},\circ:\mathtt{f}\rangle\}$$

Note that in comparison to the PNmatrix $\mathcal{M}$ of Example 34, each truth value is smaller in size and the set of truth values contains less elements. We now only show the truth table for the unary connective $\neg$ and the binary connective $\wedge$ for $\mathcal{M}'$:

$\neg_{\mathcal{M}'}$	
$\langle\epsilon:\mathtt{f},\neg:\mathtt{t},\circ:\mathtt{t}\rangle$	$\{\langle\epsilon:\mathtt{t},\neg:\mathtt{f},\circ:\mathtt{t}\rangle,\langle\epsilon:\mathtt{t},\neg:\mathtt{t},\circ:\mathtt{f}\rangle\}$
$\langle\epsilon:\mathtt{t},\neg:\mathtt{f},\circ:\mathtt{t}\rangle$	$\{\langle\epsilon:\mathtt{f},\neg:\mathtt{t},\circ:\mathtt{t}\rangle\}$
$\langle\epsilon:\mathtt{t},\neg:\mathtt{t},\circ:\mathtt{f}\rangle$	$\{\langle\epsilon:\mathtt{t},\neg:\mathtt{f},\circ:\mathtt{t}\rangle,\langle\epsilon:\mathtt{t},\neg:\mathtt{t},\circ:\mathtt{f}\rangle\}$

$\wedge_{\mathcal{M}'}$	$\langle\epsilon:\mathtt{f},\neg:\mathtt{t},\circ:\mathtt{t}\rangle$	$\langle\epsilon:\mathtt{t},\neg:\mathtt{f},\circ:\mathtt{t}\rangle$	$\langle\epsilon:\mathtt{t},\neg:\mathtt{t},\circ:\mathtt{f}\rangle$
$\langle\epsilon:\mathtt{f},\neg:\mathtt{t},\circ:\mathtt{t}\rangle$	$\{\langle\epsilon:\mathtt{f},\neg:\mathtt{t},\circ:\mathtt{t}\rangle\}$	$\{\langle\epsilon:\mathtt{f},\neg:\mathtt{t},\circ:\mathtt{t}\rangle\}$	$\{\langle\epsilon:\mathtt{f},\neg:\mathtt{t},\circ:\mathtt{t}\rangle\}$
$\langle\epsilon:\mathtt{t},\neg:\mathtt{f},\circ:\mathtt{t}\rangle$	$\{\langle\epsilon:\mathtt{f},\neg:\mathtt{t},\circ:\mathtt{t}\rangle\}$	$\{\langle\epsilon:\mathtt{t},\neg:\mathtt{f},\circ:\mathtt{t}\rangle\}$	$\emptyset$
$\langle\epsilon:\mathtt{t},\neg:\mathtt{t},\circ:\mathtt{f}\rangle$	$\{\langle\epsilon:\mathtt{f},\neg:\mathtt{t},\circ:\mathtt{t}\rangle\}$	$\{\langle\epsilon:\mathtt{t},\neg:\mathtt{t},\circ:\mathtt{t}\rangle\}$	$\{\langle\epsilon:\mathtt{t},\neg:\mathtt{t},\circ:\mathtt{f}\rangle\}$

Note that the truth table for the binary connective $\wedge$ contains an empty set. Hence, by Theorem 19, the sequent calculus H_1 is not $\mathcal{U}_{\mathcal{L}}$-analytic.

We show an example of a sequent that can only be derived by a cut on a formula that is not contained in $\mathcal{U}^+_{\mathcal{L}}$. Consider the following $\mathcal{L}$-sequent

$$s = \{\mathtt{f}:\circ p_2,\mathtt{t}:\neg p_2,\mathtt{t}:\circ p_1\}$$

and the sets $\mathcal{U}^-_{\mathcal{L}}(sub[s]) = \{p_1,p_2\}$ and $\mathcal{U}^+_{\mathcal{L}}(sub[s]) = \{p_1,p_2,\circ p_1,\circ p_2,\neg p_1,\neg p_2\}$. We show that s can only be proven with an application of (cut) on a formula that is not contained in $\mathcal{U}^+_{\mathcal{L}}(sub[s])$. A derivation with (cut) is as follows:

$$
\dfrac{
  \dfrac{
    \{\texttt{f}:p_2,\texttt{t}:\circ p_1,\texttt{t}:p_2\}
    \quad
    \dfrac{\{\texttt{f}:p_1,\texttt{f}:p_2,\texttt{t}:p_1\}}{\{\texttt{f}:p_2,\texttt{t}:\circ p_1,\texttt{t}:p_1\}}\,(\texttt{t}:\circ)
  }{\{\texttt{f}:p_2,\texttt{t}:\circ p_1,\texttt{t}:\psi\wedge p_1\}}\,(\texttt{t}:\wedge)
  \quad
  \dfrac{
    \dfrac{\{\texttt{f}:p_2,\texttt{f}:\neg p_1,\texttt{t}:\neg p_1\}}{\{\texttt{f}:p_2,\texttt{t}:\circ p_1,\texttt{t}:\neg p_1\}}\,(\texttt{t}:\circ)
  }{\{\texttt{f}:p_2,\texttt{t}:\circ p_1,\texttt{t}:\neg(\psi\wedge p_1)\}}\,(\texttt{t}:\neg\wedge)
}{(*)\{\texttt{f}:\circ(p_2\wedge p_1),\texttt{f}:p_2,\texttt{t}:\circ p_1\}}\,(\texttt{f}:\circ)
$$

$$
\dfrac{
  \dfrac{
    \{\texttt{f}:p_2,\texttt{t}:p_2,\texttt{t}:\circ p_1\}
    \quad
    \dfrac{
      \dfrac{
        \dfrac{\{\texttt{f}:\neg p_2,\texttt{t}:\neg p_2\}}{\{\texttt{t}:\neg p_2,\texttt{t}:\circ p_2\}}\,(\texttt{t}:\circ)
      }{\{\texttt{t}:\neg p_2,\texttt{t}:\circ(p_2\wedge p_1)\}}\,(\texttt{t}:\circ\wedge)
      \quad
      \dfrac{\vdots\,(*)}{\{\texttt{f}:\circ(p_2\wedge p_1),\texttt{f}:p_2,\texttt{t}:\circ p_1\}}
    }{\{\texttt{f}:p_2,\texttt{t}:\neg p_2,\texttt{t}:\circ p_1\}}\,(cut)
  }{\{\texttt{f}:\circ p_2,\texttt{f}:p_2,\texttt{t}:\circ p_1\}}\,(\texttt{f}:\circ)
}{\{\texttt{f}:\circ p_2,\texttt{t}:\neg p_2,\texttt{t}:\circ p_1\}}\,(\texttt{t}:\neg)
$$

Note that the cut formula $\circ(p_2 \wedge p_1)$ *is not contained in* $\mathcal{U}^+_{\mathcal{L}}(sub[s])$.

6.6.1 Tool: *Paralyzer*

Paralyzer (PARAconsistent logics anaLYZER) implements the procedure for the restricted case introduced in Section 6.6. It takes as input Hilbert axioms given by the grammar in Definition 49 that are specified in the language of the positive fragment of classical propositional logic extended by two new unary connectives. The axioms are transformed into equivalent logical sequent rules and a finite-valued, non-deterministic semantics is generated for the calculus obtained by adding these new rules to LK^+. Moreover, an encoding of the new calculus in *Isabelle* is produced, which can be used to perform interactive proof search.

Paralyzer can handle the logics that are described by Hilbert calculi which belong to the family $\mathbb{H}_R$. Examples are classical propositional logic **Cl**, but also paraconsistent logics known as C-systems, e.g. **B**, **BK**, **bC**.

The system is available at

`http://www.logic.at/tinc/webparalyzer/`

Example

The main page of the tool is illustrated in Figure 6.1. The user enters (1) a set of axioms and (2) the base calculus, i.e., the calculus that will be extended with the generated rules. The default option for the base calculus is LK^+, see Table 6.1. A second option is the calculus for **BK**, see Table 6.4.

After the user has provided (1) and (2), the results (the sequent rules equivalent to the input axioms, the associated semantics, and a link to the LaTeX-paper containing the *Isabelle*-encoding) are presented in a pop-up window, see Figure 6.2. Regarding the text representation, note that `G` and `D` in the rules stand for multisets of formulas Γ and Δ. In the PNmatrix, `V_M` indicates the possible set of truth values for the PNmatrix `M`. A tuple $\langle \varphi : x, \star_1\varphi : y, \star_2\varphi : z\rangle$ with $x, y, z \in \{0, 1\}$ is abbreviated as `xyz`. Note that we use `0` for `f` and `1` for `t`. Moreover, we only consider truth values with three elements in the

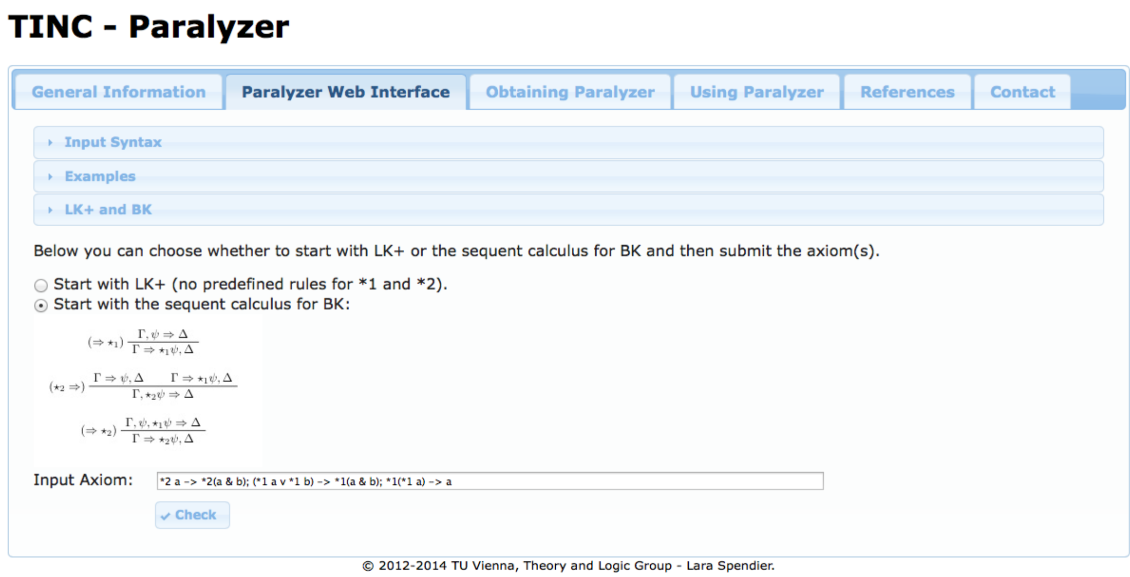


Figure 6.1: Main screen of *Paralyzer*

Paralyzer output — i.e., even if we extend our language only by one connective $\star_1$, we have a second "dummy" connective $\star_2$ in our Nmatrix. The generation of the (smaller) Nmatrix with only one connective is in fact also implemented, but was abandoned due to code maintenance reasons at a later stage.

When the program is started via the command-line, the user types `compute`, enters the axiom and the base calculus (`0` for LK^+, `2` for G_k). The following is an example output:

```
?- compute.
|: '*2' a -> '*2' (a & b); ('*1' a v '*1' b) -> '*1'(a & b);
   '*1'('*1' a) -> a.
|: 2.

Equivalent Logical Rule(s):


G,*1 a =>D      G =>a,D      G =>b,D
-----------------------------
G,*1 (a&b) =>D



G =>*1 a,D
-----------------------------
G =>*1 (a&b),D
```

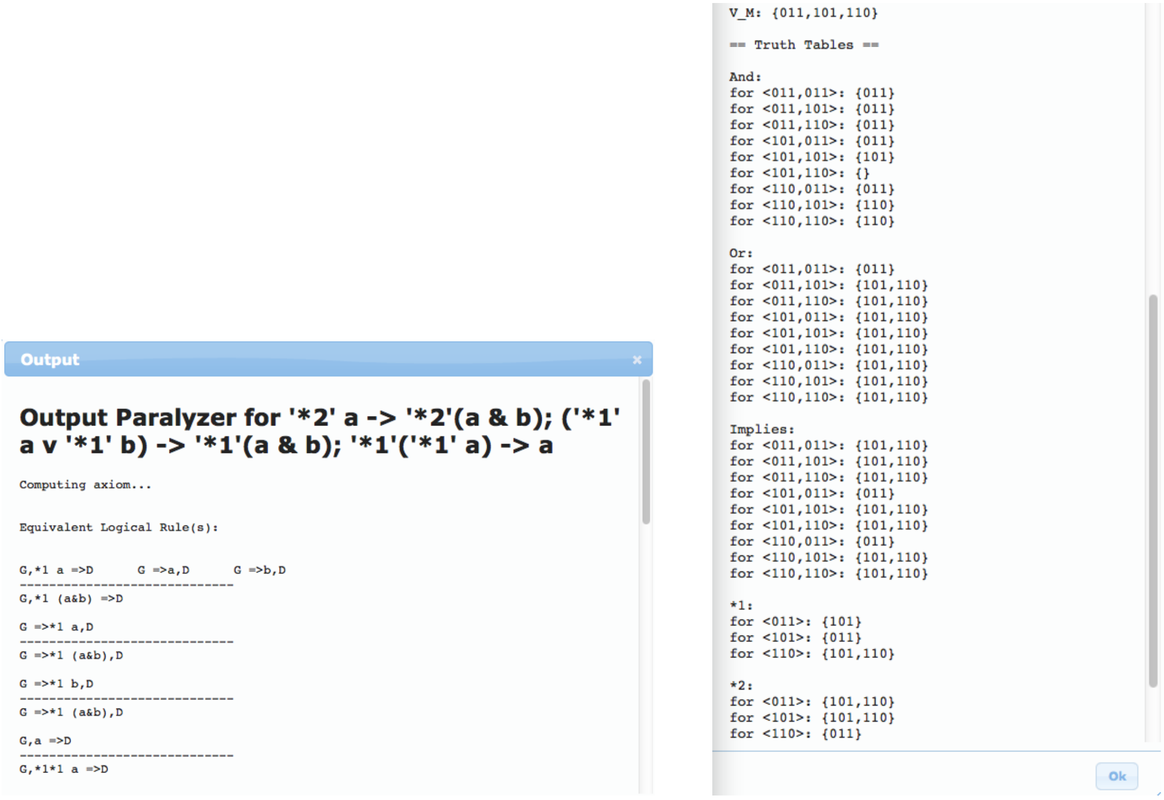


Figure 6.2: Dialog box – left: set of logical rules; right: semantics in form of a PNmatrix

```
G =>*1 b,D
-----------------------------
G =>*1 (a&b),D

G,a =>D
-----------------------------
G,*1*1 a =>D

V_M: {011,101,110}
== Truth Tables ==

And:                    Or:                        Implies:
for <011,011>: {011}    for <011,011>: {011}       for <011,011>: {101,110}
for <011,101>: {011}    for <011,101>: {101,110}   for <011,101>: {101,110}
for <011,110>: {011}    for <011,110>: {101,110}   for <011,110>: {101,110}
for <101,011>: {011}    for <101,011>: {101,110}   for <101,011>: {011}
for <101,101>: {101}    for <101,101>: {101,110}   for <101,101>: {101,110}
for <101,110>: {}       for <101,110>: {101,110}   for <101,110>: {101,110}
for <110,011>: {011}    for <110,011>: {101,110}   for <110,011>: {011}
for <110,101>: {110}    for <110,101>: {101,110}   for <110,101>: {101,110}
```

```
for <110,110>: {110}   for <110,110>: {101,110}   for <110,110>: {101,110}

*1:                    *2:
for <011>: {101}       for <011>: {101,110}
for <101>: {011}       for <101>: {101,110}
for <110>: {101,110}   for <110>: {011}
```

Implementation Details

Paralyzer is implemented in Prolog. The implementation consists of 13 files and roughly 2700 lines of code (including documentation) and follows the general **TINC**-structure described in Chapter 3 (recall Figure 3.5). The instantiation of the general **TINC**-structure for *Paralyzer* is depicted in Figure 6.3.

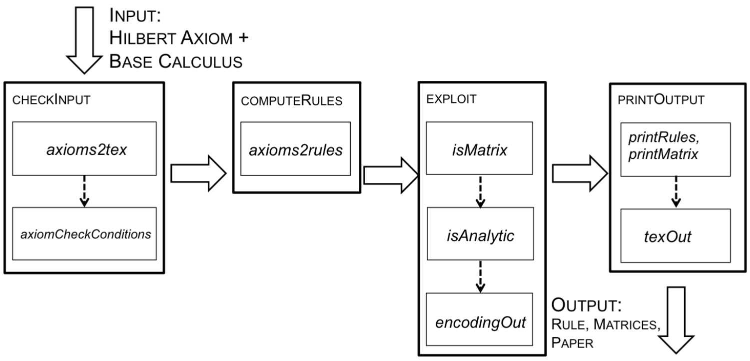


Figure 6.3: Implementation details *Paralyzer*

Input and CHECKINPUT. The input for *Paralyzer* is a Hilbert axiom and the base calculus, which are provided as parameters to the first component, CHECKINPUT.

The user can select the base calculus for the algorithm out of two options: the default option LK^+ (see Table 6.1) or the calculus G_k for **BK**, which is depicted in Table 6.4. Note that if G_k is selected as base calculus, also the invertibility of the rules ($\circ \Rightarrow$) and ($\Rightarrow \circ$) is exploited when computing the equivalent logical rules.

The syntax of the Hilbert axiom given as input is as follows:

- the letters `a, b` for (atomic) formulas
- logical connectives: `&` (and), `v` (or), `->` (implication) and `*1, *2` (unary connectives)
- a semicolon `;` to concatenate axioms

`axioms2tex` implements a syntactic check by using a definite clause grammar to determine whether the input formula only contains axioms that fall into the grammar in Definition 49.1. `axiomCheckConditions` determines whether the axioms obey the conditions regarding positively and negatively occurring subformulas in Definition 49.2.

computeRules and exploit. If the input formula provided by the user passes all syntactic checks it can be processed by the second component computeRules, which contains the implementation of the algorithm introduced in Theorem 14. `axioms2rules` transforms the axioms given as input into equivalent sequent rules. Its core is the method `axiom2logical`:

Code Example 5. *In Paralyzer, the central point to start the transformation procedure is* `axiom2logical`*, which calls* `is_logical_rule` *and* `is_rule_completion`*:*

```
%% axiom2logical(+Axiom, -Logical, +Predefined)
%%  + ... parameter given as input, - ... return value
%% creates the logical rule from an axiom
%% Axiom  ... list of axioms provided by the user
%% Logical ... Logical rule after applying the algorithm
%% Predefined ... flag=1 if *2 is an invertible connective
axiom2logical(Axiom, Logical1, Predefined) :-
      %start with the axiom in the succedent of the conclusion
    Conclusion = [[[],[Axiom]]],
      %is_logical_rule decomposes the axiom in the conclusion
    is_logical_rule(Conclusion, ConFinal, Predefined),
      %if the conclusion after the decomposition contains axioms,
      %they are removed
    removeAxioms(ConFinal, ConFinal1),
      %is_rule_completion implements the Ackermann lemma
    is_rule_completion(ConFinal1, Logical),
      %premises containing formulas that do not occur in the
      %conclusion are removed
    removeNoSubformulaPremises(Logical, Logical1).

%% is_logical_rule(+Axiom, -ConFinal,  +Predefined)
%% decomposes the axiom by repeatedly applying the invertible rules
%%  Axiom ... list of axioms provided by the user
%%  ConFinal ... conclusion of the rule
%% Predefined ... flag=1 if *2 is an invertible connective
is_logical_rule([],[],_).
is_logical_rule([H|T], ConFinal,Predefined) :-
      %we start to decomponse the first formula in the list
    apply_invertibility([H], ConFinal1,Predefined),
      %recursive call of the method
    is_logical_rule(T, ConFinal2,Predefined),
      %concatenation of the decomposed formulas into one list (return value)
    append(ConFinal1, ConFinal2, ConFinal).
```

`apply_invertibility` *then implements the recursive application of invertible rules to decompose the axiom, while* `is_rule_completion` *contains the implementation of the Ackermann lemma: it determines the formula that will be introduced in the conclusion of the rule and shifts the other formulas to the premise(s).*

We show parts of the implementation of `apply_invertibility`*:*

```
%% apply_invertibility(+Axiom, -ConFinal, +Predefined)
%%  starts application of invertible rules to the axiom
%%  checks if the outermost connective is invertible (is_invertible) and
%%  applies either invertible-right-rules or invertible-left-rules
%%  Axiom ... axiom provided by the user
%%  ConFinal ... Conclusion of the form [Antecedent,Succedent]
%%  Predefined ...flag=1 if *2 is an invertible connective
apply_invertibility([],[],_).
apply_invertibility([[A,S]|T], ConFinal,Predefined) :-
      %checks if outermost connective of succedent is invertible
    is_invertible(S,Predefined),
      %decompose succedent if connective is invertible right
    is_invertible_right([A,S], F,Predefined),
      % concatenate 'new' [A,S] with rest of the list
    append(F, T, CF1),
      % recursive call for decomposition
    apply_invertibility(CF1, ConFinal,Predefined).

%% is_invertible_right(+Conclusion, -Final, +Predefined),
%%  applies the invertible rules (->,r),(v,r),(&,r)
%%   (and (*2,r) if Predefined=1)
%% Conclusion  ... conclusion of the form [Antecedent, Succedent]
%% Final       ... list of conclusions of the form [[A,S],[A,S],...]
%%              see e.g. (&,r) which creates two rules!
%%  Predefined ... flag=1 if *2 is an invertible connective

%% (->, r): => A -> B .. A => B
is_invertible_right([A, S], F,_) :-
    member(Ax1 -> Ax2, S), %checks if Ax1->Ax2 is contained in S
    remove(Ax1 -> Ax2, S, S1), %if yes, remove Ax1->Ax2
    append([Ax2], S1, S2), %concatenate Ax2 to new succedent
    append([Ax1], A, A1),  %concatenate Ax1 to new antecedent
    F = [[A1, S2]].        %return value
%% the other rules are omitted
```

The third component EXPLOIT contains an implementation of the following things:

- `isMatrix` extracts a PNmatrix out of the newly generated calculus.
- `isAnalytic` uses the PNmatrix generated in the previous step to check the analyticity of the calculus.
- `encodingOut` constructs a formalization of the calculus in the language of the generic proof assistant *Isabelle* [171] that allows to perform interactive proof search. The encoding is a shallow embedding (recall the explanation of a shallow versus a deep embedding in Section 3.3). It is created by using the existing *Isabelle*-encoding of the propositional sequent calculus *LK* (without the rules for negation) by Lawrence Paulson, and by translating the newly constructed rules into a corresponding encoding as follows: `~` (`+`, resp.) is used to denote $\star_1$ ($\star_2$, resp.). Upper-case letters denote single formulas, while upper-case letters preceded by `$` denote (possibly empty) sequences of formulas. Rule premises are encoded left of `==>` (within brackets `[|`, `|]` and semicolon-separated (`;`) in case of multiple premises) while the conclusion is right of `==>`. See Figure 6.4 for examples of rules and their corresponding *Isabelle*-encodings as created by *Paralyzer*.

	Rules	**Encoded rules**
$(\Rightarrow \star_1)$	$\dfrac{\Gamma, \varphi \Rightarrow \Delta}{\Gamma \Rightarrow \star_1\varphi, \Delta}$	`$H,P |- $E,$F ==> $H |- $E,~P,$F`
$(\Rightarrow \star_2)$	$\dfrac{\Gamma, \varphi, \star_1\varphi \Rightarrow \Delta}{\Gamma \Rightarrow \star_2\varphi, \Delta}$	`$H,P,~P |- $E,$F ==> $H |- $E,+P,$F`
$(\star_2 \Rightarrow)$	$\dfrac{\Gamma \Rightarrow \varphi, \Delta \quad \Gamma \Rightarrow \star_1\varphi, \Delta}{\Gamma, \star_2\varphi \Rightarrow \Delta}$	`[| $H,$G |- $E,P; $H,$G |- $E,~P |] ==> $H,+P,$G |- $E`

Figure 6.4: Sequent rules for $\star_1 = \neg$ and $\star_2 = \circ$ of **BK** and their *Isabelle*-encoding

Code Example 6. *In the following code example, we show the implementation of* `isAnalytic`*. We check for each connective of the PNmatrix (*`And`*,*`Or`*,*`Implies`*,* `Star1`*,* `Star2`*) if its truth table contains an empty spot (*`hasEmptySpot`*):*

```
%% isAnalytic(+PNmatrix,-EmptySpot)
%%  + ... parameter given as input, - ... return value
%%  PNmatrix   ... contains the PNmatrix
%%  EmptySpot ... Flag; analytic = 1, non-analytic = 0
isAnalytic([_, [], _, _, _, _, _], 1).
isAnalytic([_, _, And, Or, Implies, Star1, Star2], EmptySpot) :-
    hasEmptySpot(And, ESA),
    ( ESA \= 1 -> hasEmptySpot(Or, ESO),
```

```
      ( ES0 \= 1 -> hasEmptySpot(Implies, ESI),
        ( ESI \= 1 -> hasEmptySpot(Star1, ES1),
          ( ES1 \= 1 -> hasEmptySpot(Star2, EmptySpot)
                      ; EmptySpot = 1 )
                    ; EmptySpot = 1 )
                  ; EmptySpot = 1 )
                ; EmptySpot = 1 ).

%% hasEmptySpot(+TruthTable,-EmptySpot)
%%    TruthTable... truth table for a connective
%%    EmptySpot ... Flag; empty spot = 1
hasEmptySpot([], 0).
hasEmptySpot([[_, Values]|T], EmptySpot) :-
    ( Values = [] -> % if there is an empty spot
      EmptySpot = 1  % flag is set to 1 (return value)
    ; hasEmptySpot(T, EmptySpot) % else, we check the other values
    ).
```

Output and PRINTOUTPUT. The last component PRINTOUTPUT contains the methods for creating a text representation of the calculus and its semantics on the commandline or web interface (`printRules`); moreover, the generated LaTeX-paper contains the resulting calculus (and information, whether it is analytic or not), its PNmatrix and the *Isabelle*-encoding (`texOut`).

CHAPTER 7

Conclusion

7.1 Summary

Motivated by the desire to provide results and tools for the automated investigation of substructural, intermediate and paraconsistent propositional logics, the core of our work is a general method to construct analytic calculi in various formalisms. This method, which is a generalized version of the systematic procedure in [52], is depicted in Figure 7.1 (cf. Chapter 3). Our approach works for logics that are described by adding properties in the form of Hilbert axioms or frame conditions to a suitable base logic. The properties are then translated into rules by a central transformation procedure, which relies on (a) the invertibility of the logical rules of the calculus and (b) the Ackermann lemma. Our calculi are obtained by adding these rules to the calculus for the base logic.

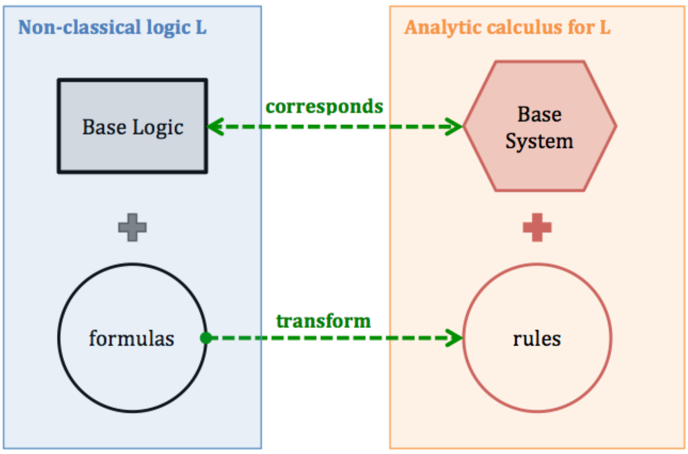


Figure 7.1: Our approach to define analytic calculi

We instantiated this general method for certain classes of intermediate and paraconsistent logics to automatically generate analytic calculi for them (for substructural logics, this is done in [52]) and use the obtained calculi to establish interesting properties for the formalized logics. We introduced the framework **TINC** (*Tools for the Investigation of Non-Classical logics*), where (most of) our procedures are implemented.

For **substructural logics**, we considered Monoidal t-norm logic **MTL** and the axiomatic extensions with axioms within the class $\mathcal{P}_3$. The procedure in [52] automatically generates analytic hypersequent calculi for these logics, transforming the axioms into equivalent structural hypersequent rules. We investigated the resulting calculi and identified a large class of rules (called *convergent*), whose corresponding logics are standard complete.

We presented two approaches to introduce analytic calculi for **intermediate logics**, differing in their starting points: the first transforms Hilbert axioms – the syntactic specification of the logic – into hypersequent rules, the latter defines labelled calculi by turning frame conditions – the semantic characterization of the logic – into labelled rules. Both methods are instantiations of the general approach. For the first method, we defined a general pattern that combines an algorithm with a heuristic to find logical hypersequent rules for axioms of a certain form. As a case study, we applied this procedure to the logic $\mathbf{Bd}_2$, hence introducing the first cut-free hypersequent calculus for it. The second method, which is inspired by [76], transforms frame conditions of a certain shape into equivalent labelled rules. Moreover, we provided uniform proofs of soundness, completeness and cut elimination for the resulting calculi.

We then considered **paraconsistent logics** that are described by means of Hilbert systems of a specific form. These Hilbert systems are defined by (i) extending the language of classical propositional logic with finitely many unary connectives and (ii) adding to the positive fragment of classical propositional logic axioms of a certain shape. Examples of Hilbert systems described this way are the standard system for classical propositional logic, or systems for logics of formal inconsistency (in particular, C-systems). We introduced a systematic two-step-procedure to generate sequent calculi and semantics for these logics. The first step is an instantiation of the general method: we transform Hilbert axioms describing the logic into equivalent logical rules, i.e. rules mentioning at least one unary connective. Based on the obtained calculi, in the second step the procedure extracts semantics in the form of PNmatrices. We used the extracted semantics to reason about the decidability of the logics and analyticity of the obtained calculi.

Last but not least, we created the framework **TINC**, which stands for *Tools for the Investigation of Non-Classical logics*. **TINC** currently contains three tools which implement (some of) the procedures to generate analytic calculi: *AxiomCalc* implements the method from [52] to obtain analytic calculi for substructural logics and checks whether the corresponding logic is standard complete. *Framinator* implements the transformation procedure for intermediate logics to translate frame conditions into equivalent labelled sequent calculus rules. Finally, the tool *Paralyzer* implements the two-step procedure for transforming (a restricted class of) Hilbert axioms into equivalent logical rules and extracting the semantics automatically.

7.2 Some Open Questions

The results developed in this thesis provide first steps towards a systematic investigation of non-classical logics. Naturally, these results raise further questions. In the following, we discuss some of these questions and shed light on future research directions.

Q1: Can our method capture more/other non-classical logics?

Yes. In fact, this is ongoing research and substructural, intermediate and paraconsistent logics were just among the first classes of non-classical logics that we have considered. Therefore it would be very interesting to extend our method(s) to capture

(i) more complicated formulas describing substructural or intermediate logics, and/or
(ii) other non-classical logics, like e.g. (first-order) modal logics, temporal logics or conditional logics (note that some of these logics are covered in [59]).

Regarding (i), recall that a classification of Hilbert axioms describing substructural logics (called substructural hierarchy) has been introduced in [52]. Any Hilbert axiom within the classes $\mathcal{N}_2$ and $\mathcal{P}_3$ of the substructural hierarchy is transformed into equivalent (hyper)sequent rules by the procedure in [52]. A similar classification of frame conditions specifying intermediate logics has been presented in Chapter 5. Our method transforms any frame condition within the class Π_2 of this classification into an equivalent labelled sequent rule. The respective classes of these hierarchies account for the difficulty to deal with the corresponding axioms or frame conditions proof-theoretically. However, as already mentioned in Section 4.2 (and Section 5.4), not all Hilbert axioms (and frame conditions) are formulas within the class $\mathcal{P}_3$ (Π_2) that can be transformed with existing procedures. Examples are the Hilbert axiom describing Łukasiewicz logic or the frame condition specifying the Kreisel-Putnam logic **KP**, which are beyond $\mathcal{P}_3$ (Π_2). Therefore it would be highly desirable to capture more "complicated" axioms or frame conditions, i.e., axioms beyond the levels $\mathcal{N}_2$ and $\mathcal{P}_3$, or frame conditions beyond Π_2.

In general, there are two main factors in our method that can be changed to capture other (or larger) classes of logics:

The first factor is the *formalism* for the base calculus. Most of the results so far are established for the framework of sequent or hypersequent calculus (see e.g. [52, 62]), but this is not the only possibility. For example, in [59] an analogous method has been used for the framework of display calculus. Further formalisms could e.g. be nested sequent calculus [42, 82], or the calculus of structures [96]. The fundamental criteria for the selection of the base calculus are: (a) a separation of logical and structural rules in the calculus and the possibility to assign polarity to the connectives (or quantifiers) of the logic, and (b) the existence of identity axioms and the cut rule. Based on that, we can determine the invertibility of the logical rules and formulate a suitable version of the Ackermann lemma, which are both required for the transformation procedure.

Second, following the spirit of our results in Chapters 5 and 6, we can change the transformation procedure to generate *logical* instead of structural rules. The addition of logical rules to the base calculus however requires some additional investigation. Since new logical rules interact with all the existing rules of the calculus mentioning the same connectives, it is more difficult to provide a uniform proof of cut elimination. A possibility is to find sufficient conditions of the calculi for cut elimination, as has been done e.g. in [145, 118]. Alternatively, an ad-hoc syntactic proof (as introduced in Section 5.3) or suitable semantic methods (as in [112] or by using tools like PNmatrices in Chapter 6) are needed to get a view on the obtained calculus as a whole.

Of course our method can also be extended in other ways to cover more logics. Possible inspirations could be other (semi-)automated procedures, such as the methods introduced in [112, 117, 115, 124] where formulas (Hilbert axioms or frame conditions) are translated into structural or logical rules, or the method in [133], which transforms frame conditions into systems of labelled rules.

*Q2: How can we combine **TINC** with existing systems?*

TINC and its tools implement (most of) the theoretical results established in this PhD thesis. One interesting research direction is thus to think of possible applications of the (encoded) calculi that are automatically generated by *AxiomCalc*, *Framinator* or *Paralyzer*. The calculi produced by *Paralyzer* are already encoded in *Isabelle* [171] to allow semi-automated proof search. Naturally, encodings of the calculi for *COQ* [37], *TWELF* [147] or other reasoners could be generated in an analogous way.

Two interesting tools that could be used in combination with **TINC** are *TATU* [136] and *MetTeL*^{2} [164] (see also Section 3.3). The investigative tool *TATU* takes as input a calculus encoded in SELLF [135] and then checks whether the specified proof system admits cut elimination. *TATU* could also be used to perform proof search in the encoded calculi. This could be very useful in combination with the generation of logical rules (see Q1). Another interesting research direction is the encoding of our calculi for the tableau prover *MetTeL*^{2} to perform automated proof search.

Q3: Can we show standard completeness for other logics?

In Chapter 4, we introduced the notion of convergent rules for the hypersequent rules that are generated from axioms that are within the class $\mathcal{P}_3$ of the substructural hierarchy. We showed standard completeness for any propositional logic extending **MTL** with a set of axioms having equivalent convergent rules. In [27], the notion of convergent rules was extended to calculi for first-order substructural logics. The results led to standard completeness proofs for axiomatic extensions of first-order **MTL**. As shown in [27], convergency of rules is however not a necessary condition for density elimination (and, hence, for standard completeness). An interesting research direction would be to extend the notion of convergency to obtain a necessary and sufficient condition for density elimination.

Note that the condition of convergent rules is also too weak to ensure density elimination for (propositional or first-order) hypersequent calculi that do *not* contain the rules for weakening, i.e. (w, l) and (w, r). These calculi characterize extensions of Uninorm Logic

UL [127], i.e., **MTL** without weakening. Until recently, mostly calculi-tailored proofs of density elimination (and standard completeness) have been available for weakening-free logics, see [127, 168, 25]. A first general density elimination proof has been introduced in [26], where standard completeness is proved for a large class of extensions of **UL** with many axioms within $\mathcal{N}_2$. Note however, that axioms within the class $\mathcal{P}_3$ still cannot be covered by this method (and neither all the axioms of the class $\mathcal{N}_2$). An interesting question here would indeed be to identify a class of $\mathcal{P}_3$ axioms that, when added to **UL**, lead to standard complete logics.

Q4: Can we provide a uniform proof of analyticity for the hypersequent calculi introduced in Chapter 5?

In the first part of Chapter 5, we transformed Hilbert axioms of a specific form into equivalent *logical* hypersequent rules using a heuristics. Since the addition of logical rules to the base calculus might destroy cut elimination, as mentioned above, we had to provide an ad-hoc proof of cut elimination for the newly obtained calculus. Therefore, it is an interesting question if we can provide a fully automated procedure for the introduction of logical hypersequent rules that do not harm the analyticity of the resulting calculus.

One possibility is to find strong enough conditions for the preservation of analyticity that can be checked, similar e.g. to the convergency condition for standard completeness. It would also be interesting to use the results from [115] and see to which extent they can be used for our calculi. An alternative would be to implement the generation of the calculi in **TINC** and use tools such as *TATU* to reason about the analyticity of the calculi.

Q5: Can we define analytic calculi and non-deterministic semantics for a larger class of paraconsistent logics?

In fact, step 1 of the procedure in Chapter 6 (the systematic introduction of calculi) could be easily adapted to capture e.g. paraconsistent logics extending intuitionistic logic, substructural paraconsistent logics or first-order logics. However, we currently have no uniform method to show that the generated calculi are indeed analytic. The construction of the corresponding PNmatrices (step 2) would also require a deeper investigation, and this is currently the main theoretical problem for extending our work. For the time being there is indeed no theory of PNmatrices for first-order logics, intuitionistic logics or substructural logics (that, in fact, also lack a theory of Nmatrices).

Moreover, step 2 of our procedure works only for logics characterized by finitely-valued partial non-deterministic matrices. Some paraconsistent logics can however only be characterized by *infinitely-valued* (partial) non-deterministic matrices, e.g. logics defined by Hilbert systems that include the axioms **(l)** $\neg(\varphi \wedge \neg\varphi) \supset \circ\varphi$ or **(d)** $\neg(\neg\varphi \wedge \varphi) \supset \circ\varphi$ (cf. Table 6.2, Chapter 6). For such logics, step 1 of our procedure still works, but we cannot extract semantics in form of PNmatrices out of it. A similar problem was addressed in [17], where the modular procedure from [16, 18] has been extended to construct infinitely-valued Nmatrices and equivalent sequent calculi for logics including these "problematic" axioms. An interesting open question is if we can – and if yes, how to – similarly extend our procedure to these logics.

APPENDIX A

Substructural Logics

The proofs in this appendix belong to Section 4.3. We prove that *HMTL* (the hypersequent calculus for **MTL**, see Table 4.4) extended with any set of convergent rules (see Definition 24) admits density elimination.

Lemma 3. *Let R be any set of convergent rules extending the calculus HMTL and let H be the calculus defined by HMTL+R.*

1. *Any derivation d of H can be transformed into a derivation of $H[{}^{p}/_{\alpha}]^{l}[{}^{p}/_{\Rightarrow\alpha}]^{r}$, for any formula α and propositional variable p.*
2. *Let d' and d_1 be derivations of $G' \mid \Sigma, p \Rightarrow \Pi \mid \Lambda \Rightarrow p$ (where $p \notin G', \Sigma, \Pi, \Lambda$) and $G' \mid \Theta, \Delta \Rightarrow \Psi$. We can find a derivation of $G' \mid \Theta, \Lambda \Rightarrow \Psi \mid \Sigma, \Delta \Rightarrow \Pi$.*

Proof. 1. Let d_α be the derivation d where p is replaced with α. By induction on $|d_\alpha|$. When d_α ends in an initial sequent, we are done. Otherwise, consider the last inference rule r' in d_α. Let r' be any logical or structural rule of *HMTL* or $r' \in R$. The claim holds since all rules of *HMTL* and convergent rules (that are completed rules) are substitutive (recall Definition 17).

2. By 1. and d' we have a derivation d_2 of $G' \mid \Sigma, \odot\Delta \Rightarrow \Pi \mid \Lambda \Rightarrow \odot\Delta$ where $\odot\Delta$ stands for the multiplicative conjunction $\cdot$ of the formulas in Δ (note that $p \notin G', \Sigma, \Pi, \Lambda$). The desired derivation follows by applying (cut) to $G' \mid \Theta, \Lambda \Rightarrow \Psi \mid \Delta \Rightarrow \odot\Delta$ and the end hypersequent of

$$\dfrac{\begin{array}{c}\vdots\, d_2 \\ G' \mid \Sigma, \odot\Delta \Rightarrow \Pi \mid \Lambda \Rightarrow \odot\Delta\end{array} \qquad \dfrac{\begin{array}{c}\vdots\, d_1 \\ G' \mid \Theta, \Delta \Rightarrow \Psi\end{array}}{G' \mid \Theta, \odot\Delta \Rightarrow \Psi \mid \Sigma, \odot\Delta \Rightarrow \Pi}\,(\cdot, l), (ew)}{G' \mid \Theta, \Lambda \Rightarrow \Psi \mid \Sigma, \odot\Delta \Rightarrow \Pi}\,(cut)$$

□

In the following proof, we denote by S_i^* the sequent $S_i[{}^p/_\Lambda]^l[{}^p/_{\Sigma\Rightarrow\Pi}]^r$, and by G^*,H^*, the hypersequents G,H, where the same substitution is applied to each one of their components.

Theorem 4 (Density elimination). *HMTL extended with any set R of convergent rules admits density elimination.*

Proof. To perform density elimination, it is sufficient to repeatedly remove topmost applications of (D). Let d be a derivation in $HMTL + (D) + R$ ending in an application of (D), with d' the (D)-free derivation ending in $G' \mid \Sigma, p \Rightarrow \Pi \mid \Lambda \Rightarrow p$, i.e.:

$$\dfrac{\begin{array}{c}\vdots\, d' \\ G' \mid \Sigma, p \Rightarrow \Pi \mid \Lambda \Rightarrow p\end{array}}{G' \mid \Sigma, \Lambda \Rightarrow \Pi}\ (D)$$

Since convergent rules are completed rules of a particular form, they preserve cut elimination when added to *HMTL*. Hence we assume that d' is cut-free.

Claim: For each hypersequent H in d' that is not a p-axiom, one can find a (D)-free derivation of $G' \mid H^*$.

The result on density elimination follows from this claim as follows: Let H be $G' \mid \Lambda \Rightarrow p \mid \Sigma, p \Rightarrow \Pi$. From the claim we get that $G' \mid G' \mid \Lambda, \Sigma \Rightarrow \Pi \mid \Lambda, \Sigma \Rightarrow \Pi$ is derivable (note that $(G')^* = G'$ by the eigenvariable condition on p). The desired proof of $G' \mid \Lambda, \Sigma \Rightarrow \Pi$ follows by multiple applications of (ec).

The proof of the claim proceeds by induction on the height of the cut-free subderivation d_H of H in $HMTL + R$. We distinguish cases according to the last rule r applied in d_H. When $|d_H| = 0$, or when r is (ec) or (ew), we are done after an application to the induction hypothesis. Otherwise, consider the following cases:

Suppose that r is any rule except for (com), (ec), (ew) or a convergent rule:

$$\dfrac{G \mid S_1 \quad \dots \quad G \mid S_m}{G \mid S}\ r$$

and $G \mid S$ does not contain a p-axiom. Hence, also no $G \mid S_i$ for $i \in \{1, \dots, m\}$ contains a p-axiom. Then by inductive hypothesis, we get a derivation of $G' \mid G^* \mid S_i^*$ for any premise $G \mid S_i$. The claim follows by application of r and (ec), if needed.

Suppose that r is (com). We have to distinguish two cases: First, suppose that none of the premises contains a p-axiom. Then the claim follows by applications of the inductive hypothesis and (com), e.g.:

$$\dfrac{\begin{array}{c}\vdots d_1 \\ G \mid \Gamma_1, \Gamma_2 \Rightarrow p\end{array} \qquad \begin{array}{c}\vdots d_2 \\ G \mid \Delta_1, \Delta_2, p^k \Rightarrow \Psi\end{array}}{G \mid \Gamma_1, \Delta_1, p^k \Rightarrow \Psi \mid \Gamma_2, \Delta_2 \Rightarrow p}\ (com)$$

Then by inductive hypothesis we have:

$$\vdash_{HMTL} G' \mid G^* \mid \Gamma_1, \Gamma_2, \Sigma \Rightarrow \Pi \quad \text{and } \vdash_{HMTL} G' \mid G^* \mid \Delta_1, \Delta_2, \Lambda^k \Rightarrow \Psi$$

Hence by (com), $\vdash_{HMTL} G' \mid G^* \mid \Gamma_1, \Delta_1, \Lambda^k \Rightarrow \Psi \mid \Gamma_2, \Delta_2, \Sigma \Rightarrow \Pi$.

For the second case, suppose that one premise contains a p-axiom, e.g.

$$\frac{G \mid \Gamma_1, \Gamma_2, p^l \Rightarrow p \qquad G \mid \Delta_1, p^{(k-l)}, \Delta_2 \Rightarrow \Psi}{G \mid \Gamma_1, \Delta_1, p^k \Rightarrow \Psi \mid \Gamma_2, \Delta_2 \Rightarrow p} (com)$$

We show how to obtain a (D)-free derivation of

$$G' \mid G^* \mid \Gamma_1, \Delta_1, \Lambda^k \Rightarrow \Psi \mid \Gamma_2, \Delta_2, \Sigma \Rightarrow \Pi$$

By the inductive hypothesis, we get a derivation $d_1 \vdash_{HMTL} G' \mid G^* \mid \Delta_1, \Lambda^{(k-l)}, \Delta_2 \Rightarrow \Psi$. By applying Lemma 3.2 to d_1 and to $d' \vdash_{HMTL} G' \mid \Sigma, p \Rightarrow \Pi \mid \Lambda \Rightarrow p$, we get a derivation of $G' \mid G^* \mid \Delta_1, \Lambda^{k-l+1} \Rightarrow \Psi \mid \Delta_2, \Sigma \Rightarrow \Pi$.

The desired derivation of $G' \mid G^* \mid \Gamma_1, \Delta_1, \Lambda^k \Rightarrow \Psi \mid \Gamma_2, \Delta_2, \Sigma \Rightarrow \Pi$ is then obtained by applications of (w, l).

Finally, suppose that r is a convergent rule of the form

$$\frac{G \ \mid \ S_1 \quad \dots \quad G \ \mid \ S_m}{G \ \mid \ C_1 \ \mid \ \dots \ \mid \ C_q} \ r$$

and that the conclusion of r contains no p-axiom. We show how to find a derivation of

$$G' \mid G^* \mid C_1^* \mid \cdots \mid C_q^*.$$

Take a premise $G \mid S_i$. If $G \mid S_i$ is not a p-axiom, the inductive hypothesis gives us a derivation of $G' \mid G^* \mid S_i^*$. Note that this is always the case when $R(S_i) = \emptyset$, and when $G \mid S_i$ is a 0-pivot as in the latter case the metavariables instantiated to obtain S_i are all included in one component of the conclusion. Thus, if $G \mid S_i$ was a p-axiom, the conclusion would be a p-axiom as well, contradicting the assumption.

Assume now that $G \mid S_i$ is a p-axiom. We show below that we can always obtain a (D)-free derivation of

$$G' \mid G^* \mid S_i^* \mid C_s^*$$

for some $s \in \{1, \dots, q\}$. As r is convergent, there is an n-pivot premise $G \mid S_j$ for $G \mid S_i$. We show how to use $G \mid S_j$ to obtain the required derivation with the following case distinction:

- $n = 1$: There is a 1-pivot premise $G \mid S_j$ for $G \mid S_i$, i.e., (the metasequent leading to) S_i differs only in one metavariable from (that of) S_j. By Definition 23, $G \mid S_j$ is also a 0-pivot and hence it is not a p-axiom. Let $G \mid S_i$ and $G \mid S_j$ be obtained as instantiations of the following premises[1] of r:

$$S_i \quad \text{is obtained from} \quad \overline{\Theta}, \overline{\Gamma}^k, \overline{\Delta}^l \Rightarrow \overline{\Pi} \quad \text{and} \quad S_j \quad \text{from} \quad \overline{\Theta}, \overline{\Delta}^{k+l} \Rightarrow \overline{\Pi}$$

[1]To simplify the notation $\overline{\Theta}$ stands for all the metavariables S_i and S_j have in common except $\overline{\Delta}$.

As $G \mid S_i$ is a p-axiom and $G \mid S_j$ is not, only $\overline{\Gamma}$ can be instantiated with a propositional variable p. The most general case is when $\overline{\Gamma}$ is instantiated by Γ, p^n, the metavariable $\overline{\Delta}$ by the multiset Δ, $\overline{\Theta}$ by Θ and $\overline{\Pi}$ by p. Hence

$$G \mid S_i \quad \text{is} \quad G \mid \Theta, \Gamma^k, \Delta^l, p^{mk} \Rightarrow p \quad \text{and} \quad G \mid S_j \quad \text{is} \quad G \mid \Theta, \Delta^{k+l} \Rightarrow p.$$

As $G \mid S_j$ is not a p-axiom, by the inductive hypothesis we have a derivation for $G' \mid G^* \mid \Theta, \Delta^{k+l}, \Sigma \Rightarrow \Pi$. Using the derivation d' of $G' \mid \Sigma, p \Rightarrow \Pi \mid \Lambda \Rightarrow p$, by k applications of Lemma 3.2 with (ew) and (ec) we get

$$G' \mid G^* \mid \Theta, \Lambda^k, \Delta^l, \Sigma \Rightarrow \Pi \mid \Sigma, \Delta \Rightarrow \Pi$$

Now, by multiple applications of internal weakenings (w, l), we have

$$G' \mid G^* \mid \Theta, \Gamma^k, \Lambda^{mk}, \Delta^l, \Sigma \Rightarrow \Pi \mid \Sigma, \Delta \Rightarrow \Pi$$

From further repeated applications of (w, l) on the fourth component, we finally obtain $G' \mid G^* \mid S_i^* \mid C_s^*$, where C_s stands for the component of the conclusion to which all the metavariables in (the metasequent leading to) S_j belong.

- $n > 1$: Let $G \mid S_j$ be an n-pivot premise for $G \mid S_i$. By Definition 23 (the metasequent leading to) $G \mid S_i$ differs from (that of) $G \mid S_j$ by n metavariables and there exist n other premises for which $G \mid S_j$ is an $(n-1)$-pivot. As in the previous case, let S_i and S_j be obtained respectively as instantiations of the following premises of r

$$\overline{\Theta}, \overline{\Gamma}_1^{k_1}, \ldots, \overline{\Gamma}_n^{k_n}, \overline{\Delta}_1^{l_1}, \ldots, \overline{\Delta}_n^{l_n} \Rightarrow \overline{\Pi} \quad \text{and} \quad \overline{\Theta}, \overline{\Delta}_1^{k_1+l_1}, \ldots, \overline{\Delta}_n^{k_n+l_n} \Rightarrow \overline{\Pi}$$

Assume w.l.o.g. that $G \mid S_j$ is:

$$G \mid \Theta, \Delta_1^{k_1+l_1}, \ldots, \Delta_n^{k_n+l_n} \Rightarrow p$$

Two cases have to be considered, according to the possible instantiations of the metavariables $\overline{\Gamma_i}$ with the propositional variable p in $G \mid S_i$:

(i) In $G \mid S_i$ all the metavariables $\overline{\Gamma_i}$ are instantiated with a multiset Γ_i together with at least one occurrence of p. Then we repeatedly apply Lemma 3.2 to d' and $G \mid S_j$ together with (ew) and (ec) to replace $\Delta_1, \ldots, \Delta_n$ with Λ in S_j, respectively $k_1, \ldots k_n$ times. This way we get

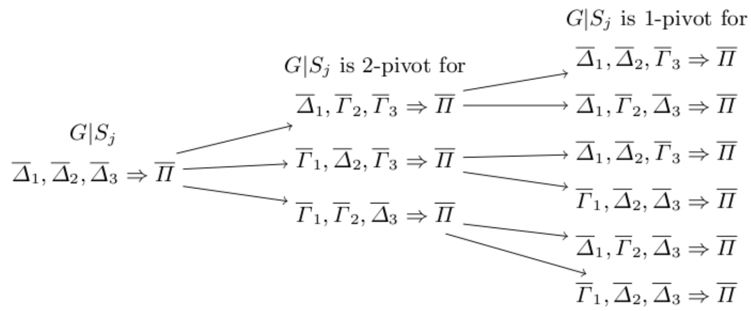
$$G \mid \Theta, \Lambda^{k_1}, \ldots, \Lambda^{k_n}, \Delta_1^{l_1}, \ldots, \Delta_n^{l_n}, \Sigma \Rightarrow \Pi \mid \Delta_1, \Sigma \Rightarrow \Pi \mid \ldots \mid \Delta_n, \Sigma \Rightarrow \Pi$$

The desired hypersequent $G' \mid G^* \mid S_i^* \mid C_s^*$ follows by suitable applications of (w, l) and (ec) (as in the 1-pivot case, C_s stands for the component of the conclusion to which all the metavariables in S_j belong).

(ii) In $G \mid S_i$ all the metavariables $\overline{\Gamma_i}$ are instantiated with Γ_i, p^{m_i} and $m_i > 0$ only for t of them ($1 \leq t < n$). (Note that in this case Lemma 3.2 would replace each metavariable Δ_i with Λ, leading to at least n occurrences of Λ, and $n > r$.) The idea is to find another premise of r which is not a p-axiom and suitably apply Lemma 3.2. The existence of (at least one) such a premise is guaranteed by the notion of n-pivot.

We first illustrate the way we proceed with an example for $n = 3$.

Assume that S_i arises as an instantiation of $\overline{\Gamma}_1, \overline{\Gamma}_2, \overline{\Gamma}_3 \Rightarrow \overline{\Pi}$ and S_j as an instantiation of $\overline{\Delta}_1, \overline{\Delta}_2, \overline{\Delta}_3 \Rightarrow \overline{\Pi}$ ($G \mid S_j$ is a 3-pivot for $G \mid S_i$). By definition of 3-pivot, there exist 3 premises in r for which $G \mid S_j$ is a 2-pivot. For each of these premises, there exist 2 premises in r for which $G \mid S_j$ is a 1-pivot. In the figure below we show how all these premises are related w.r.t the metavariables they instantiate.



(Case $r = 1$) If only 1 metavariable, say $\overline{\Gamma}_1$, is instantiated in S_i with Γ_1, p we need to find a corresponding premise which will not contain a p-axiom, i.e., that does not contain $\overline{\Gamma}_1$. The first occurrence of such a premise is among the premises that have $G \mid S_j$ as a 2-pivot, that is $\overline{\Delta}_1, \overline{\Gamma}_2, \overline{\Gamma}_3 \Rightarrow \overline{\Pi}$.

(Case $r = 2$) Assume now that 2 metavariables, say $\overline{\Gamma}_1, \overline{\Gamma}_2$, are instantiated with Γ_1, p and Γ_2, p, respectively. Again, we need to find a corresponding premise that is not a p-axiom. In this case, the set of premises that have $G \mid S_j$ as a 2-pivot does not suffice because each of them contains either $\overline{\Gamma}_1$ or $\overline{\Gamma}_2$. The first occurrence of a premise that is not a p-axiom is among the premises that have $G \mid S_j$ as a 1-pivot, i.e., $\overline{\Delta}_1, \overline{\Delta}_2, \overline{\Gamma}_3 \Rightarrow \overline{\Pi}$.

In general, we can eventually find a premise that is not a p-axiom among those that have $G \mid S_j$ as $(n - t)$-pivot. Assume for the general case that the occurrences of p in $G \mid S_i$ are related to the instantiation of t different metavariables, w.l.o.g $\overline{\Gamma_1}, \ldots, \overline{\Gamma_t}$, i.e., S_i is

$$\Theta, \Gamma_1^{k_1}, \ldots, \Gamma_n^{k_n}, \Delta_1^{l_1}, \ldots, \Delta_n^{l_n}, p^{m_1 k_1 + \ldots + m_t k_t} \Rightarrow p \quad \text{with } m_1, \ldots, m_t > 0$$

Then we can find premises for which $G \mid S_j$ is an $(n - t)$-pivot; (the metasequents leading to) those premises differ from (that of) $G \mid S_i$ in t metavariables. (At least) one of these premises will not be a p-axiom, and hence it will have the form:

$$G \mid \Theta, \Delta_1^{k_1+l_1}, \ldots, \Delta_t^{k_t+l_t}, \Gamma_{t+1}^{k_{t+1}}, \Delta_{t+1}^{l_{t+1}}, \ldots, \Gamma_n^{k_n}, \Delta_n^{l_n} \Rightarrow p$$

By the inductive hypothesis on the depth of the derivation we have a derivation of:

$$G' \mid G^* \ \mid \ \Theta, \Delta_1^{k_1+l_1}, \ldots, \Delta_t^{k_t+l_t}, \Gamma_{t+1}^{k_{t+1}}, \Delta_{t+1}^{l_{t+1}}, \ldots, \Gamma_n^{k_n}, \Delta_n^{l_n}, \Sigma \Rightarrow \Pi$$

Then we repeatedly apply Lemma 3.2 together with (ew) and (ec) to replace $\Delta_1, \ldots, \Delta_t$ with Λ, respectively $k_1, \ldots, k_t$ times. After suitable applications of (w, l) and (ec), we finally get

$$G' \mid G^* \ \mid \ S_i^* \mid C_s^*$$

Summing up, when the last rule in d_H is convergent, for each premise $G \mid S_i$ we have:

- If $G \mid S_i$ does not contain any p-axiom, $G' \mid G^* \mid S_i^*$ is (D)-free derivable.
- If $G \mid S_i$ contains a p-axiom, then $G' \mid G^* \mid S_i^* \mid C_s^*$ is (D)-free derivable.

The required derivation of $G' \mid G^* \mid {C_1}^* \mid \ldots \mid {C_q}^*$ follows by r and subsequent applications of (ec), if needed. This completes the proof of the main claim. □

APPENDIX B

Paraconsistent Logics

The proofs in this appendix belong to Section 6.6, which contains our procedure for the specific subclass of paraconsistent logics defined by the Hilbert calculi $\mathbb{H}_R$.

For any Hilbert calculus $H \in \mathbb{H}_R$, we construct a $\mathcal{U}_\mathcal{L}$-simple sequent calculus G and a PNmatrix $\mathcal{M}_G$ by Corollary 8 and Definition 52. We first present the soundness and completeness proof of the $\mathcal{U}_\mathcal{L}$-simple sequent calculus G w.r.t. its PNmatrix $\mathcal{M}_G$. In Theorem 19, we prove that a $\mathcal{U}_\mathcal{L}$-simple sequent calculus is $\mathcal{U}_\mathcal{L}$-analytic iff its PNmatrix is an Nmatrix (i.e., it does not have an empty set in its truth tables).

Theorem 18 (Soundness and Completeness). *Let s_0 be a $\mathcal{W}$-sequent and G be a $\mathcal{U}_\mathcal{L}$-simple sequent calculus for $\mathcal{L}$. Then, $\vdash_G^{\mathcal{U}_\mathcal{L}^+(\mathcal{W})} s_0$ iff every $\mathcal{M}_G$-$\mathcal{U}_\mathcal{L}^-(\mathcal{W})$-valuation w-satisfies s_0.*

Proof. "$\Rightarrow$": It suffices to show that whenever an $\mathcal{M}_G$-$\mathcal{U}_\mathcal{L}^-(\mathcal{W})$-valuation w-satisfies the premises of some application of a rule $r = Q/s$ of G consisting solely of formulas from $\mathcal{U}_\mathcal{L}^+(\mathcal{W})$, it also w-satisfies its conclusion. Consider such an application of r inferring $\sigma(s) \cup c$ from the set $\{\sigma(q) \cup c \mid q \in Q\}$, where c is an $\mathcal{L}$-sequent, and σ is an $\mathcal{L}$-substitution. Let v be an $\mathcal{M}_G$-$\mathcal{U}_\mathcal{L}^-(\mathcal{W})$-valuation and suppose that v w-satisfies $\sigma(q) \cup c$ for every $q \in Q$. We prove that v w-satisfies $\sigma(s) \cup c$. If v w-satisfies c, we are done. Suppose otherwise. Then v w-satisfies $\sigma(q)$ for every $q \in Q$. We show that v w-satisfies $\sigma(s)$ (and, thus, it w-satisfies $\sigma(s) \cup c$). Let $\sigma(p_1) = \psi_1$ and $\sigma(p_2) = \psi_2$. We consider the case when r is an $\mathcal{U}_\mathcal{L}$-simple rule (the proofs for the rules of LK^+ are similar). The following three cases can occur:

- Suppose that $r = Q/\{\mathtt{b} : \triangleright p_1\}$ is a unary-one rule. Note that since we only consider applications of r consisting solely of formulas from $\mathcal{U}_\mathcal{L}^+(\mathcal{W})$, we have that $\triangleright\psi_1 \in \mathcal{U}_\mathcal{L}^+(\mathcal{W})$ and so $\psi_1 \in \mathcal{U}_\mathcal{L}^-(\mathcal{W})$. The fact v w-satisfies $\sigma(q)$ for every $q \in Q$ implies that $v(\psi_1)$ satisfies every $q \in Q$. To see this, consider the following cases:
  - $q = \{\mathtt{b} : p_1\}$, and ψ_1 does not have the form $\star\varphi$: Since v w-satisfies $\sigma(q)$, $v(\psi_1)^\epsilon = \mathtt{b}$.

- $q = \{\mathtt{b} : p_1\}$, and ψ_1 has the form $\star\varphi$: Since v w-satisfies $\sigma(q)$, $v(\varphi)^\star = \mathtt{b}$ and as v is an $\mathcal{M}_G$-valuation, $v(\star\varphi)^\epsilon = \mathtt{b}$.
- $q = \{\mathtt{b} : \star p_1\}$: Since v w-satisfies $\sigma(q)$, $v(\psi_1)^\star = \mathtt{b}$.

Hence, we obtain that $v(\psi_1)$ satisfies q. Since $v(\psi_1) \in \mathcal{V}_{\mathcal{M}_G}$, $v(\psi_1)$ respects r, and so $v(\psi_1)^\triangleright = \mathtt{b}$. Thus v w-satisfies $\{\mathtt{b} : \triangleright\psi_1\}$.

- Suppose that $r = Q/\{\mathtt{b} : \star\triangleright p_1\}$ is a unary-two rule. As in the previous case, $v(\psi_1)$ satisfies every $q \in Q$. Thus, since $v(\triangleright\psi_1) \in \triangleright_{\mathcal{M}_G}(v(\psi_1))$, we have $v(\triangleright\psi_1)^\star = \mathtt{b}$. It follows that v w-satisfies $\{\mathtt{b} : \star\triangleright\psi_1\}$.
- Suppose that $r = Q/\{\mathtt{b} : \star(p_1 \diamond p_2)\}$ is a binary rule. Similarly to the previous cases, $\langle v(\psi_1), v(\psi_2)\rangle$ satisfies every $q \in Q$. Thus, since $v(\psi_1 \diamond \psi_2) \in \diamond_{\mathcal{M}_G}(v(\psi_1), v(\psi_2))$, we have that $v(\psi_1 \diamond \psi_2)^\star = \mathtt{b}$. It follows that v w-satisfies $\{\mathtt{b} : \star(\psi_1 \diamond \psi_2)\}$.

"$\Leftarrow$": Suppose that $\nvdash_G^{\mathcal{U}_\mathcal{L}^+(\mathcal{W})} s_0$. We construct an $\mathcal{M}_G$-$\mathcal{U}_\mathcal{L}^-(\mathcal{W})$-valuation v that does not w-satisfy s_0. It is straightforward to construct a "maximal" (infinite) set Ω of labelled $\mathcal{L}$-formulas that extends s_0 and satisfies the following conditions:

1. Ω consists of labelled $\mathcal{L}$-formulas of the form $\mathtt{b} : \psi$ for $\psi \in \mathcal{U}_\mathcal{L}^+(\mathcal{W})$.
2. $\nvdash_G^{\mathcal{U}_\mathcal{L}^+(\mathcal{W})} s$ for every $\mathcal{L}$-sequent $s \subseteq \Omega$.
3. For every formula $\psi \in \mathcal{U}_\mathcal{L}^+(\mathcal{W})$ and $\mathtt{b} \in \{\mathtt{f}, \mathtt{t}\}$, if $\mathtt{b} : \psi \notin \Omega$ then $\vdash_G^{\mathcal{U}_\mathcal{L}^+(\mathcal{W})} s \cup \{\mathtt{b} : \psi\}$ for some $\mathcal{L}$-sequent $s \subseteq \Omega$.

Note that the availability of the rules (*cut*) and (*id*) implies that:

1. For every $\psi \in \mathcal{U}_\mathcal{L}^+(\mathcal{W})$, either $\mathtt{f} : \psi \in \Omega$ or $\mathtt{t} : \psi \in \Omega$. Otherwise, we would have $\vdash_G^{\mathcal{U}_\mathcal{L}^+(\mathcal{W})} s_1 \cup \{\mathtt{f} : \psi\}$ and $\vdash_G^{\mathcal{U}_\mathcal{L}^+(\mathcal{W})} s_2 \cup \{\mathtt{t} : \psi\}$ for $s_1, s_2 \subseteq \Omega$. By applying (*cut*) (and weakenings) we could obtain $\vdash_G^{\mathcal{U}_\mathcal{L}^+(\mathcal{W})} s_1 \cup s_2$. Since $s_1 \cup s_2 \subseteq \Omega$, this contradicts the properties of Ω.
2. Similarly, for every $\psi \in \mathcal{U}_\mathcal{L}^+(\mathcal{W})$, either $\mathtt{f} : \psi \notin \Omega$ or $\mathtt{t} : \psi \notin \Omega$. Otherwise, $\{\mathtt{t} : \psi, \mathtt{f} : \psi\} \in \Omega$, but $\vdash_G^{\mathcal{U}_\mathcal{L}^+(\mathcal{W})} \{\mathtt{f} : \psi, \mathtt{t} : \psi\}$ by applying (*id*).

Let v be the function from $\mathcal{U}_\mathcal{L}^-(\mathcal{W})$ to $\mathsf{F}_{\mathcal{U}_\mathcal{L}}$ defined by $v(\psi)^\epsilon = \mathtt{t}$ iff $\mathtt{f} : \psi \in \Omega$, and for every $\star \in \mathcal{U}_\mathcal{L}$: $v(\psi)^\star = \mathtt{t}$ iff $\mathtt{f} : \star\psi \in \Omega$. Thus we have that for every $\psi \in \mathcal{U}_\mathcal{L}^-(\mathcal{W})$ and $\mathtt{b} \in \{\mathtt{f}, \mathtt{t}\}$, $v(\psi)^\epsilon = b$ iff $\mathtt{b} : \psi \notin \Omega$, and for every $\star \in \mathcal{U}_\mathcal{L}$ $v(\psi)^\star = \mathtt{b}$ iff $\mathtt{b} : \star\psi \notin \Omega$. We show that v does not w-satisfy s_0. Let $\mathtt{b} : \psi \in s_0$ such that ψ does not have the form $\star\varphi$. Thus $\psi \in \mathcal{U}_\mathcal{L}^-(\mathcal{W})$, and since $s_0 \subseteq \Omega$, $v(\psi)^\epsilon \neq \mathtt{b}$. Similarly, let $\mathtt{b} : \psi \in s_0$ such that ψ does have the form $\psi = \star\varphi$ (for some $\star \in \mathcal{U}$ and $\mathcal{L}_\mathcal{U}$-formula φ). Thus $\varphi \in \mathcal{U}_\mathcal{L}^-(\mathcal{W})$, and since $s_0 \subseteq \Omega$, $v(\varphi)^\star \neq \mathtt{b}$.

To show that v is an $\mathcal{M}_G$-valuation, we use the following properties:

(∗) Let σ be an $\mathcal{L}_\mathcal{U}$-substitution, such that $\sigma(p_1) \in \mathcal{U}_\mathcal{L}^-(\mathcal{W})$. If $v(\sigma(p_1))$ satisfies an $\mathcal{L}$-sequent q of the form $\{\mathtt{b} : \star p_1\}$ or $\{\mathtt{b} : p_1\}$, then $\vdash_G^{\mathcal{U}_\mathcal{L}^+(\mathcal{W})} s \cup \sigma(q)$ for some $\mathcal{L}$-sequent $s \subseteq \Omega$.

Proof. Suppose that $v(\sigma(p_1))$ satisfies q. Then one of the following holds:

- $q = \mathtt{b} : p_1$ and $v(\sigma(p_1))^{\epsilon} = \mathtt{b}$. Thus $\mathtt{b} : \sigma(p_1) \notin \Omega$, and since $\sigma(p_1) \in \mathcal{U}^{+}_{\mathcal{L}}(\mathcal{W})$, we obtain that $\vdash_G^{\mathcal{U}^{+}_{\mathcal{L}}(\mathcal{W})} s \cup \{\mathtt{b} : \sigma(p_1)\}$ for some $\mathcal{L}$-sequent $s \subseteq \Omega$.
- $q = \mathtt{b} : \star p_1$ and $v(\sigma(p_1))^{\star} = \mathtt{b}$. Thus $b : \star\sigma(p_1) \notin \Omega$, and since $\star\sigma(p_1) \in \mathcal{U}^{+}_{\mathcal{L}}(\mathcal{W})$, we obtain that $\vdash_G^{\mathcal{U}^{+}_{\mathcal{L}}(\mathcal{W})} s \cup \{\mathtt{b} : \star\sigma(p_1)\}$ for some $\mathcal{L}_{\mathcal{U}}$-sequent $s \subseteq \Omega$.

Similarly, we have the following:

(∗∗) Let q be an $\mathcal{L}$-sequent of the form q of the form $\{\mathtt{b} : \star p_i\}$ or $\{\mathtt{b} : p_i\}$ for $i \in \{1, 2\}$, and σ be an $\mathcal{L}$-substitution, such that $\sigma(p_1), \sigma(p_2) \in \mathcal{U}^{-}_{\mathcal{L}}(\mathcal{W})$. If $\langle v(\sigma(p_1)), v(\sigma(p_2))\rangle$ satisfies q, then $\vdash_G^{\mathcal{U}^{+}_{\mathcal{L}}(\mathcal{W})} s \cup \sigma(q)$ for some $\mathcal{L}$-sequent $s \subseteq \Omega$.
Proof. Suppose that $\langle v(\sigma(p_1)), v(\sigma(p_2))\rangle$ satisfies q. Then one of the following holds:

- $q = \mathtt{b} : p_i$ and $v(\sigma(p_i))^{\epsilon} = \mathtt{b}$ for $i \in \{1, 2\}$. Thus $\mathtt{b} : \sigma(p_i) \notin \Omega$, and since $\sigma(p_i) \in \mathcal{U}^{+}_{\mathcal{L}}(\mathcal{W})$, we obtain that $\vdash_G^{\mathcal{U}^{+}_{\mathcal{L}}(\mathcal{W})} s \cup \{\mathtt{b} : \sigma(p_i)\}$ for some $\mathcal{L}$-sequent $s \subseteq \Omega$.
- $q = \mathtt{b} : \star p_i$ and $v(\sigma(p_i))^{\star} = \mathtt{b}$ for $i \in \{1, 2\}$. Thus $b : \star\sigma(p_i) \notin \Omega$, and since $\star\sigma(p_i) \in \mathcal{U}^{+}_{\mathcal{L}}(\mathcal{W})$, we obtain that $\vdash_G^{\mathcal{U}^{+}_{\mathcal{L}}(\mathcal{W})} s \cup \{\mathtt{b} : \star\sigma(p_i)\}$ for some $\mathcal{L}_{\mathcal{U}}$-sequent $s \subseteq \Omega$.

We show that $\mathcal{V}_{\mathcal{M}_G}$ is the range of v. Let $\psi \in \mathcal{U}^{-}_{\mathcal{L}}(\mathcal{W})$. To prove that $v(\psi) \in \mathcal{V}_{\mathcal{M}_G}$, we show that $v(\psi)$ respects all unary-one rules of G. Consider a unary-one rule of G, $r = Q/\{\mathtt{b} : \star p_1\}$. Suppose that $v(\psi)$ satisfies every $q \in Q$. We show that $v(\psi)^{\star} = \mathtt{b}$. Let σ be any $\mathcal{L}$-substitution, assigning ψ to p_1. By (∗), for every $q \in Q$, there exists some $\mathcal{L}$-sequent $s_q \subseteq \Omega$ such that $\vdash_G^{\mathcal{U}^{+}_{\mathcal{L}}(\mathcal{W})} s_q \cup \sigma(q)$. By applying weakenings and the rule r, we obtain that $\vdash_G^{\mathcal{U}^{+}_{\mathcal{L}}(\mathcal{W})} \bigcup_{q\in Q} s_q \cup \{\mathtt{b} : \star\psi\}$ (here we use the fact that $\star\psi \in \mathcal{U}^{+}_{\mathcal{L}}(\mathcal{W})$ since $\psi \in \mathcal{U}^{-}_{\mathcal{L}}(\mathcal{W})$). Thus, $\{\mathtt{b} : \star\psi\} \notin \Omega$, and so $v(\psi)^{\star} = \mathtt{b}$.

Next, we show that v respects the truth-tables of $\mathcal{M}_G$:

1. Let $\rhd\psi \in \mathcal{U}^{-}_{\mathcal{L}}(\mathcal{W})$ (where $\rhd \in \mathcal{U}_{\mathcal{L}}$). We show that $v(\rhd\psi) \in \rhd_{\mathcal{M}_G}(v(\psi))$. By the construction of $\rhd_{\mathcal{M}_G}$, it suffices to show: (i) $v(\rhd\psi)^{\epsilon} = v(\psi)^{\rhd}$; and (ii) $v(\rhd\psi)^{\star} = \mathtt{b}$ for every unary-complex rule $r = Q/\{\mathtt{b} : \star\rhd p_1\}$ of G for which $v(\psi)$ satisfies every $q \in Q$. (i) trivially holds using the definition of v. For (ii), let $r = Q/\{\mathtt{b} : \star\rhd p_1\}$ be a unary-complex rule of G, and suppose that $v(\psi)$ satisfies every $q \in Q$. We prove that $v(\rhd\psi)^{\star} = \mathtt{b}$. Let σ be any $\mathcal{L}$-substitution, assigning ψ to p_1. By (∗) (note that $\psi \in \mathcal{U}^{-}_{\mathcal{L}}(\mathcal{W})$ since $\mathcal{U}^{-}_{\mathcal{L}}(\mathcal{W})$ is closed under subformulas), for every $q \in Q$, there exists some $\mathcal{L}$-sequent $s_q \subseteq \Omega$ such that $\vdash_G^{\mathcal{U}^{+}_{\mathcal{L}}(\mathcal{W})} s_q \cup \sigma(q)$. By applying weakenings and the rule r, we obtain that $\vdash_G^{\mathcal{U}^{+}_{\mathcal{L}}(\mathcal{W})} \bigcup_{q\in Q} s_q \cup \{\mathtt{b} : \star\rhd\psi\}$ (note that $\star\rhd\psi \in \mathcal{U}^{+}_{\mathcal{L}}(\mathcal{W})$ since $\rhd\psi \in \mathcal{U}^{-}_{\mathcal{L}}(\mathcal{W})$). This implies that $\mathtt{b} : \star\rhd\psi \notin \Omega$, and so $v(\rhd\psi)^{\star} = \mathtt{b}$.
2. Let $\psi_1 \diamond \psi_2 \in \mathcal{U}^{-}_{\mathcal{L}}(\mathcal{W})$ for $\diamond \in \{\wedge, \vee, \supset\}$. We show that $v(\psi_1 \diamond \psi_2) \in \diamond_{\mathcal{M}_G}(v(\psi_1), v(\psi_2))$. Here it suffices to show: (i) $v(\psi_1 \diamond \psi_2)^{\star} = \mathtt{b}$ for every binary rule $r = Q/\{\mathtt{b} : \star(p_1 \diamond p_2)\}$ of G for which $\langle v(\psi_1), v(\psi_2)\rangle$ satisfies every $q \in Q$; and (ii) $v(\diamond(\psi_1, \psi_2))^{\epsilon} \in$

$\diamond_{\mathcal{M}_{\mathbf{Cl}^+}}(v(\psi_1)^\epsilon, v(\psi_2)^\epsilon)$. We prove (i) ((ii) is similar). Let $r = Q/\{\mathtt{b} : \star(p_1 \diamond p_2)\}$ be a binary rule of G, and suppose that $\langle v(\psi_1), v(\psi_2)\rangle$ satisfies every $q \in Q$. We prove that $v(\psi_1 \diamond \psi_2)^\star = \mathtt{b}$. Let σ be any $\mathcal{L}$-substitution, assigning ψ_1 to p_1, and ψ_2 to p_2. By $(**)$, for every $q \in Q$, there exists some $\mathcal{L}$-sequent $s_q \subseteq \Omega$ such that $\vdash_G^{\mathcal{U}^+_{\mathcal{L}}(\mathcal{W})} s_q \cup \sigma(q)$. By applying weakenings and the rule r, we obtain that $\vdash_G^{\mathcal{U}^+_{\mathcal{L}}(\mathcal{W})} \bigcup_{q \in Q} s_q \cup \{\mathtt{b} : \star(\psi_1 \diamond \psi_2)\}$ (note that $\star(\psi_1 \diamond \psi_2) \in \mathcal{U}^+_{\mathcal{L}}(\mathcal{W})$ since $\psi_1 \diamond \psi_2 \in \mathcal{U}^-_{\mathcal{L}}(\mathcal{W})$). This implies that $\{\mathtt{b} : \star(\psi_1 \diamond \psi_2)\} \notin \Omega$, and so $v(\psi_1 \diamond \psi_2)^\star = \mathtt{b}$.

□

Corollary 9. *For every $\mathcal{L}$-sequent s, $\vdash_G s$ iff $\vdash_{\mathcal{M}_G} s$.*

Proof. The claim follows by choosing $\mathcal{W} = \mathcal{L}$ in Theorem 18 (in this case $\mathcal{U}^+_{\mathcal{L}}(\mathcal{W}) = \mathcal{U}^-_{\mathcal{L}}(\mathcal{W}) = \mathcal{W}$). Note that an $\mathcal{M}_G$-valuation v for $\mathcal{L}$ w-satisfies an $\mathcal{L}$-sequent iff $v \models_{\mathcal{M}_G} s$ (since $v(\star\psi)^\epsilon = v(\psi)^\star$ for every $\mathcal{L}$-formula $\star\psi$). □

Theorem 19. *A $\mathcal{U}_{\mathcal{L}}$-simple calculus G is $\mathcal{U}_{\mathcal{L}}$-analytic iff $\mathcal{M}_G$ is an Nmatrix.*

Proof. "⇒": Suppose that $\mathcal{M}_G$ is not an Nmatrix. First, if $\mathcal{V}_{\mathcal{M}_G}$ is empty, then $\vdash_{\mathcal{M}_G} \emptyset$, and so by Corollary 9, $\vdash_G \emptyset$. But, $\mathcal{U}^+_{\mathcal{L}}(\emptyset) = \emptyset$, and clearly there is no derivation in G that does not contain any formula. It follows that G is not $\mathcal{U}_{\mathcal{L}}$-analytic in this case. Otherwise, there exist either some $\rhd \in \mathcal{U}_{\mathcal{L}}$ and $u \in \mathcal{V}_{\mathcal{M}_G}$ such that $\rhd_{\mathcal{M}_G}(u) = \emptyset$, or some $\diamond \in \{\wedge, \vee, \supset\}$ and $u_1, u_2 \in \mathcal{V}_{\mathcal{M}_G}$ such that $\diamond_{\mathcal{M}_G}(u_1, u_2) = \emptyset$. We consider here only the first case (the second case is similar). Define the $\mathcal{L}_{\mathcal{U}}$-sequent $s = \{\overline{u^\epsilon} : p_1\} \cup \{\overline{u^\star} : \star p_1 \mid \star \in \mathcal{U}_{\mathcal{L}}\}$ (where $\overline{\mathtt{t}} = \mathtt{f}$ and $\overline{\mathtt{f}} = \mathtt{t}$). We first prove that $\vdash_G s$. By Corollary 9 it suffices to show $\vdash_{\mathcal{M}_G} s$. Suppose otherwise, and let v be an $\mathcal{M}_G$-valuation for $\mathcal{L}$ such that $v \not\models_{\mathcal{M}_G} s$. Then, $v(p_1)^\epsilon = u^\epsilon$ and $v(\star p_1)^\epsilon = u^\star$ for every $\star \in \mathcal{U}_{\mathcal{L}}$. Since v is an $\mathcal{M}_G$-valuation, we have that $v(p_1)^\star = u^\star$ for every $\star \in \mathcal{U}_{\mathcal{L}}$. It follows that $v(p_1) = u$. Moreover, we have $v(\rhd p_1) \in \rhd_{\mathcal{M}_G}(v(p_1))$. Clearly, this is not possible under the assumption that $\rhd_{\mathcal{M}_G}(u) = \emptyset$. Next we claim that $\not\vdash_G^{\mathcal{U}^+_{\mathcal{L}}(sub[s])} s$ (and so G is not $\mathcal{U}_{\mathcal{L}}$-analytic). To see this, note that the $\{p_1\}$-valuation defined by $v(p_1) = u$ is an $\mathcal{M}_G$-$\mathcal{U}^-_{\mathcal{L}}(sub[s])$-valuation that does not w-satisfy s. By Theorem 18, $\not\vdash_G^{\mathcal{U}^+(sub[s])} s$.

"⇐": Assume that $\mathcal{M}_G$ is an Nmatrix and $\not\vdash_G^{\mathcal{U}^+_{\mathcal{L}}(sub[s])} s$ for some $\mathcal{L}$-sequent s. We prove that $\not\vdash_G s$. By Theorem 18, there exists an $\mathcal{M}_G$-$\mathcal{U}^-_{\mathcal{L}}(sub[s])$-valuation v that does not w-satisfy s. As $\mathcal{M}_G$ is an Nmatrix, v can be extended to a (full) $\mathcal{M}_G$-valuation v'. Note that $v' \not\models_{\mathcal{M}_G} s$ (since $v(\star\psi)^\epsilon = v'(\psi)^\star$ for every $\mathcal{L}$-formula $\star\psi$). Corollary 9 then entails that $\not\vdash_G s$. □

APPENDIX C

TINC: Tool Output

C.1 Example LaTeX-output of *AxiomCalc*

The following pages contain the file generated by *AxiomCalc* for the example shown in Section 4.3.1.

Standard completeness for **MTL** extended with the axiom $\neg(\alpha \cdot \beta) \vee (\alpha \wedge \beta \to \alpha \cdot \beta)$

AxiomCalc*

November 16, 2014

Abstract

This paper introduces a cut-free hypersequent calculus for **MTL** extended with the (Hilbert) axiom $\neg(\alpha \cdot \beta) \vee (\alpha \wedge \beta \to \alpha \cdot \beta)$. The calculus is generated by the Prolog-program *AxiomCalc*, which implements the procedure in [2]. Moreover, it shows that the resulting logic is standard complete. This is done by checking the conditions in [1] on the generated calculus, which guarantee standard completeness for the considered logic.

1 Introduction

We introduce a cut-free hypersequent calculus for Monoidal t-norm logic **MTL** extended with the axiom $\neg(\alpha\cdot\beta)\vee(\alpha\wedge\beta \to \alpha\cdot\beta)$. The analytic calculus for this logic is obtained via a Prolog-implementation of the procedure in [2]. Moreover, we check whether the newly generated rule is convergent. This ensures standard completeness for **MTL** extended with $\neg(\alpha \cdot \beta) \vee (\alpha \wedge \beta \to \alpha \cdot \beta)$, that is, completeness of the logic with respect to algebras based on the truth values in $[0, 1]$.

2 Preliminaries

The basic system we will deal with is Monoidal t-norm logic **MTL** which is the logic of left-continuous t-norms[1]. It is obtained by adding the prelinearity axiom $(\alpha \to \beta) \vee (\beta \to \alpha)$ to intuitionistic logic without contraction, see Table 1 for the corresponding hypersequent calculus *HMTL*. **MTL** is standard complete.

Formulas of **MTL** are built from propositional variables and the constants 0 and 1 by using $\to$ (implication), $\wedge$ (additive conjunction), $\cdot$ (multiplicative conjunction), and $\vee$ (disjunction). We use $\neg\alpha$ as an abbreviation for $\alpha \to 0$.

We use $\alpha, \beta, \ldots$ to denote (metavariables for) formulas, Π stands for stoups, i.e., either a (metavariable for a) formula or the empty set, and $\Gamma, \Delta, \ldots$ denote (metavariables for) finite (possibly empty) multisets of formulas.

*`http://www.logic.at/tinc/webaxiomcalc`

[1] A *t*-norm is a commutative, associative, increasing function $* : [0,1]^2 \to [0,1]$ with identity element 1. $*$ is *left continuous* iff whenever $\{x_n\}, \{y_n\}$ $(n \in N)$ are increasing sequences in $[0, 1]$ s.t. their suprema are x and y, then $\sup\{x_n * y_n : n \in N\} = x * y$. The residuum of $*$ is a function $\to^*$ where $x \to^* y = \max\{z \mid x * z \leq y\}$.

$\dfrac{G \mid \Gamma \Rightarrow \alpha \quad G \mid \alpha, \Delta \Rightarrow \Pi}{G \mid \Gamma, \Delta \Rightarrow \Pi}\ (cut)$	$\dfrac{}{G \mid \alpha \Rightarrow \alpha}\ (init)$	$\dfrac{}{G \mid 0 \Rightarrow}\ (0l)$
$\dfrac{G \mid \Gamma \Rightarrow \alpha \quad G \mid \Delta \Rightarrow \beta}{G \mid \Gamma, \Delta \Rightarrow \alpha \cdot \beta}\ (\cdot r)$	$\dfrac{G \mid \alpha, \beta, \Gamma \Rightarrow \Pi}{G \mid \alpha \cdot \beta, \Gamma \Rightarrow \Pi}\ (\cdot l)$	$\dfrac{}{G \mid \Rightarrow 1}\ (1r)$
$\dfrac{G \mid \Gamma \Rightarrow \alpha \quad G \mid \beta, \Delta \Rightarrow \Pi}{G \mid \Gamma, \alpha \rightarrow \beta, \Delta \Rightarrow \Pi}\ (\rightarrow l)$	$\dfrac{G \mid \alpha, \Gamma \Rightarrow \beta}{G \mid \Gamma \Rightarrow \alpha \rightarrow \beta}\ (\rightarrow r)$	$\dfrac{G \mid \Gamma \Rightarrow \Pi}{G \mid \Gamma, \alpha \Rightarrow \Pi}\ (wl)$
$\dfrac{G \mid \Gamma \Rightarrow \alpha \quad G \mid \Gamma \Rightarrow \beta}{G \mid \Gamma \Rightarrow \alpha \wedge \beta}\ (\wedge r)$	$\dfrac{G \mid \alpha_i, \Gamma \Rightarrow \Pi}{G \mid \alpha_1 \wedge \alpha_2, \Gamma \Rightarrow \Pi}\ (\wedge l)$	$\dfrac{G \mid \Gamma \Rightarrow}{G \mid \Gamma \Rightarrow \Pi}\ (wr)$
$\dfrac{G \mid \alpha, \Gamma \Rightarrow \Pi \quad G \mid \beta, \Gamma \Rightarrow \Pi}{G \mid \alpha \vee \beta, \Gamma \Rightarrow \Pi}\ (\vee l)$	$\dfrac{G \mid \Gamma \Rightarrow \alpha_i}{G \mid \Gamma \Rightarrow \alpha_1 \vee \alpha_2}\ (\vee r)$	$\dfrac{G}{G \mid \Gamma \Rightarrow \Pi}\ (EW)$
$\dfrac{G \mid \Gamma \Rightarrow \Pi \mid \Gamma \Rightarrow \Pi}{G \mid \Gamma \Rightarrow \Pi}\ (EC)$	$\dfrac{G \mid \Gamma_1, \Delta_1 \Rightarrow \Pi_1 \quad G \mid \Gamma_2, \Delta_2 \Rightarrow \Pi_2}{G \mid \Gamma_1, \Gamma_2 \Rightarrow \Pi_1 \mid \Delta_1, \Delta_2 \Rightarrow \Pi_2}\ (com)$	

Table 1: Hypersequent calculus *HMTL* for **MTL**

Definition 1 *A* hypersequent *G is a multiset $S_1 \mid \cdots \mid S_n$ where each S_i for $i = 1, \ldots, n$ is a sequent, called a* component *of the hypersequent. A hypersequent is called* single-conclusion *if all its components are single-conclusion.*

The symbol "|" is intended to denote disjunction at the meta-level. In this paper, we only consider single-conclusion (hyper)sequents. Given a sequent S henceforth we will denote by $LHS(S)$ its left hand side and by $RHS(S)$ its right hand side. Let $S := \Gamma_1, \Gamma_2 \Rightarrow \Pi$, we indicate by $S[^{\Gamma_1}/_{\Sigma}]^l$ the sequent $\Sigma, \Gamma_2 \Rightarrow \Pi$.

As in the case of sequent calculus, the hypersequent calculus consists of initial axioms, logical rules, the cut-rule and structural rules. Initial axioms, logical rules and the cut-rule are essentially the same as in the sequent calculus. The only difference is that a (possibly empty) side hypersequent G may occur in hypersequents. The structural rules are divided into two groups: *internal* structural rules and *external* structural rules. The former are applied to formulas within sequents. External rules instead manipulate the components of a hypersequent and therefore increase the expressive power of hypersequent calculus with respect to sequent calculus.

The notion of proof in *HMTL* is defined as usual. Let R be a set of rules. If there is a proof in *HMTL* extended with R (*HMTL*+R, for short) of a sequent S_0 from a set of sequents $\mathcal{S}$, we say that S_0 is derivable from $\mathcal{S}$ in *HMTL*+R and write $\mathcal{S} \vdash_{HMTL+R} S_0$. We write $\vdash_{HMTL+R} \alpha$ if $\emptyset \vdash_{HMTL+R} \Rightarrow \alpha$.

Two hypersequent rules (hr_0) and (hr_1) are equivalent (in *HMTL*) if the relations $\vdash_{HMTL+(hr_0)}$ and $\vdash_{HMTL+(hr_1)}$ coincide when restricted to sequents.

2.1 Substructural Hierarchy

The substructural hierarchy is a novel classification of Hilbert axioms based on the logical connectives of **MTL**.

Definition 2 (Substructural Hierarchy) [2] *Let $\mathcal{A}$ be a set of atomic formulas. For $n \geq 0$, the sets $\mathcal{P}_n, \mathcal{N}_n$ of formulas are defined as follows:*

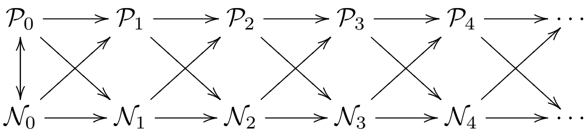


Figure 1: The substructural hierarchy [2]

$$\begin{aligned}&\mathcal{P}_0 ::= \mathcal{N}_0 ::= \mathcal{A}\\ &\mathcal{P}_{n+1} ::= \mathcal{N}_n \mid \mathcal{P}_{n+1} \cdot \mathcal{P}_{n+1} \mid \mathcal{P}_{n+1} \vee \mathcal{P}_{n+1} \mid 1\\ &\mathcal{N}_{n+1} ::= \mathcal{P}_n \mid \mathcal{P}_{n+1} \to \mathcal{N}_{n+1} \mid \mathcal{N}_{n+1} \wedge \mathcal{N}_{n+1} \mid 0\end{aligned}$$

A graphical representation of the substructural hierarchy is depicted in Figure 1. Note that the arrows $\to$ stand for inclusions $\subseteq$ of the classes.

2.2 From axioms to analytic rules

The axiom $\neg(\alpha \cdot \beta) \vee (\alpha \wedge \beta \to \alpha \cdot \beta)$ is within the class $\mathcal{P}_3$ of the substructural hierarchy [2]. Using the algorithm in [2], the axiom

$$\neg(\alpha \cdot \beta) \vee (\alpha \wedge \beta \to \alpha \cdot \beta)$$

can be transformed into the following rule to be added to the hypersequent calculus *HMTL*:

$$\frac{\begin{array}{cc} G \mid \Gamma_1, \Gamma_1, \Delta_1 \Rightarrow \Pi_1 & G \mid \Gamma_2, \Gamma_3, \Delta_1 \Rightarrow \Pi_1 \\ G \mid \Gamma_2, \Gamma_1, \Delta_1 \Rightarrow \Pi_1 & G \mid \Gamma_1, \Gamma_3, \Delta_1 \Rightarrow \Pi_1 \end{array}}{G \mid \Gamma_2, \Gamma_3 \Rightarrow \mid \Gamma_1, \Delta_1 \Rightarrow \Pi_1}$$

Theorem 3 (Soundness and Completeness.) *The axiom* $\neg(\alpha\cdot\beta)\vee(\alpha\wedge\beta \to \alpha \cdot \beta)$ *is equivalent to the newly generated rule.*

Proof. See [2].

Theorem 4 (Cut-Admissibility.) *The cut rule is admissible in the calculus* HMTL *extended with the newly generated rule.*

Proof. See [2].

A cut-elimination procedure can be found in [4].

3 Standard completeness for MTL+$\neg(\alpha\cdot\beta)\vee(\alpha\wedge \beta \to \alpha \cdot \beta)$

Let (r) be any hypersequent rule generated by the procedure in [2] where S_i, C_j denote sequents

$$\frac{G \mid S_1 \quad \dots \quad G \mid S_m}{G \mid C_1 \mid \dots \mid C_q}$$

Definition 5 *Let $G|S_i$ and $G|S_j$ be among the premises of (r).*

(0-pivot) *$G|S_i$ is a* 0-pivot *if there is an $s \in \{1, \ldots, q\}$ such that $RHS(S_i) = RHS(C_s)$ and the different metavariables in the $LHS(S_i)$ are contained in those of $LHS(C_s)$.*

(n-pivot) *$G|S_j$ is an* n-pivot for *$G|S_i$, for $n > 0$, if the following conditions hold:*

- *$G|S_j$ is a* 0-pivot
- *$RHS(S_i) = RHS(S_j)$*
- *$LHS(S_j) = LHS(S_i[^{\Gamma_1}/_{\Delta_1}, \ldots ^{\Gamma_n}/_{\Delta_n}]^l)$ for $\Gamma_1, \ldots \Gamma_n \in LHS(S_i)$ and $\Delta_1, \ldots \Delta_n \in LHS(S_j)$*
- *$G|S_j$ is a* (n-1)-pivot *for n premises $G|S_{j_1}$, .. ,$G|S_{j_n}$, and for $i = 1..n$ $LHS(S_j) = LHS(S_{j_i}[^{\Gamma_1}/_{\Delta_1}, \ldots, ^{\Gamma_{i-1}}/_{\Delta_{i-1}}, \ldots, ^{\Gamma_{i+1}}/_{\Delta_{i+1}}, ^{\Gamma_n}/_{\Delta_n}]^l)$*

Definition 6 *A completed hypersequent rule (r) is* convergent *if for each premise $G|S_i$ one of the following conditions holds: (1) $RHS(S_i) = \emptyset$, (2) $G|S_i$ is a 0-pivot, or (3) there is a premise $G|S_j$ which is an* n-pivot *for $G|S_i$, with $n > 0$.*

Lemma 7 *The rule equivalent to the axiom $\neg(\alpha \cdot \beta) \vee (\alpha \wedge \beta \to \alpha \cdot \beta)$ is convergent.*

Proof. Consider again the generated rule:

$$\frac{\begin{array}{cc} G \mid \Gamma_1, \Gamma_1, \Delta_1 \Rightarrow \Pi_1 & G \mid \Gamma_2, \Gamma_3, \Delta_1 \Rightarrow \Pi_1 \\ G \mid \Gamma_2, \Gamma_1, \Delta_1 \Rightarrow \Pi_1 & G \mid \Gamma_1, \Gamma_3, \Delta_1 \Rightarrow \Pi_1 \end{array}}{G \mid \Gamma_2, \Gamma_3 \Rightarrow \mid \Gamma_1, \Delta_1 \Rightarrow \Pi_1}$$

The premise(s) of the rule satisfy conditions (1)-(3) in Definition 6. Indeed:

- The following premises are a **0-pivot**:
  $G \mid \Gamma_1, \Gamma_1, \Delta_1 \Rightarrow \Pi_1$
- There exists a **1-pivot** for the following premises:
  $G \mid \Gamma_1, \Gamma_3, \Delta_1 \Rightarrow \Pi_1 \quad G \mid \Gamma_2, \Gamma_1, \Delta_1 \Rightarrow \Pi_1$
- There exists a **2-pivot** for the following premises:
  $G \mid \Gamma_2, \Gamma_3, \Delta_1 \Rightarrow \Pi_1$

Theorem 8 *The logic formalized by the calculus HMTL extended with any convergent rule is standard complete.*

Proof. See [1].

Hence, **MTL** extended with $\neg(\alpha \cdot \beta) \vee (\alpha \wedge \beta \to \alpha \cdot \beta)$ is standard complete.

References

[1] P. Baldi, A. Ciabattoni, and L. Spendier. Standard completeness for extensions of MTL: an automated approach. In *Proceedings of Int. Workshop on Logic, Language, Information and Computation (WoLLIC 2012), LNCS*, pages 154–167, 2012.

[2] A. Ciabattoni, N. Galatos, and K. Terui. From axioms to analytic rules in nonclassical logics. In *IEEE Symposium on Logic in Computer Science (LICS 08)*, pages 229–240, 2008.

[3] A. Ciabattoni and L. Spendier. Tools for the Investigation of Substructural and Paraconsistent Logics. In *Proceedings of JELIA 2014, LNAI*, pages 18–32, 2014.

[4] A. Ciabattoni, L. Straßburger, and K. Terui. Expanding the realm of systematic proof theory. In *Proceedings of Computer Science Logic (CSL 09), LNCS*, pages 163–178, 2009.

C.2 Example LaTeX-output of *Framinator*

The following pages contain the file generated by *Framinator* for the example shown in Section 5.4.1.

A labelled calculus for **G3I** extended with the frame condition $\forall x, y, z \exists w(x \leq y \wedge x \leq z \rightarrow y \leq w \wedge z \leq w)$

Framinator*

September 7, 2014

Abstract

We introduce a cut-free labelled calculus for the intermediate logic defined by extending **G3I** with the frame condition $\forall x, y, z \exists w(x \leq y \wedge x \leq z \rightarrow y \leq w \wedge z \leq w)$. The calculus is generated by the Prolog-tool *Framinator* (FRAMe condItioNs Automatically TO Rules), which implements the procedure in [1].

1 Introduction

Intermediate logics, i.e., logics between intuitionistic and classical logic, have a natural Kripke semantics defined by imposing conditions on the standard intuitionistic frame. Cut-free labelled systems [3, 5, 4] have been provided for a large class of intermediate logics in a modular way in [2]. The resulting calculi are indeed defined by adding to the base labelled calculus for intuitionistic logic extra structural rules corresponding to the frame conditions — that are formulas of first-order classical logic — characterizing the considered logic.

In this paper, we introduce a cut-free labelled calculus for the logic obtained by extending **G3I** with the frame condition $\forall x, y, z \exists w(x \leq y \wedge x \leq z \rightarrow y \leq w \wedge z \leq w)$. The calculus is obtained via a Prolog-implementation of the procedure in [1], where a classification of the frame conditions according to their quantifier alternation and an algorithm to automatically create structural rules out of them are introduced.

2 Preliminaries

The language of propositional intermediate logics consists of infinitely many propositional variables $p, q \ldots$, the connectives & (conjunction), $\vee$ (disjunction), $\supset$ (implication), and the constant $\bot$ for falsity. $\varphi, \psi, \ldots$ are formulas built from atoms by using connectives and $\bot$. Multisets of formulas are denoted by $\Gamma, \Delta, \ldots$.

An intuitionistic frame is a pair $\mathfrak{F} = \langle W, \leqslant \rangle$ where W is a non-empty set, and $\leqslant$ is a reflexive and transitive (accessibility) relation on W. An intuitionistic

*http://www.logic.at/tinc/webframinator/

$$x \leqslant y, x : p, \Gamma \Rightarrow \Delta, y : p \qquad \frac{x : \varphi, x : \psi, \Gamma \Rightarrow \Delta}{x : \varphi \& \psi, \Gamma \Rightarrow \Delta}\,{\scriptstyle L\&} \qquad \frac{\Gamma \Rightarrow \Delta, x : \varphi \quad \Gamma \Rightarrow \Delta, x : \psi}{\Gamma \Rightarrow \Delta, x : \varphi \& \psi}\,{\scriptstyle R\&}$$

$$\frac{}{x : \bot, \Gamma \Rightarrow \Delta}\,{\scriptstyle L\bot} \qquad \frac{\Gamma \Rightarrow \Delta, x : \varphi, x : \psi}{\Gamma \Rightarrow \Delta, x : \varphi \vee \psi}\,{\scriptstyle R\vee} \qquad \frac{x : \varphi, \Gamma \Rightarrow \Delta \quad x : \psi, \Gamma \Rightarrow \Delta}{x : \varphi \vee \psi, \Gamma \Rightarrow \Delta}\,{\scriptstyle L\vee}$$

$$\frac{x \leqslant x, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}\,{\scriptstyle Ref} \qquad \frac{x \leqslant y, y : \varphi, \Gamma \Rightarrow \Delta, y : \psi}{\Gamma \Rightarrow \Delta, x : \varphi \supset \psi}\,{\scriptstyle R\supset} \qquad \frac{x \leqslant z, x \leqslant y, y \leqslant z, \Gamma \Rightarrow \Delta}{x \leqslant y, y \leqslant z, \Gamma \Rightarrow \Delta}\,{\scriptstyle Trans}$$

$$\frac{x \leqslant y, x : \varphi \supset \psi, \Gamma \Rightarrow \Delta, y : \varphi \quad x \leqslant y, x : \varphi \supset \psi, y : \psi, \Gamma \Rightarrow \Delta}{x \leqslant y, x : \varphi \supset \psi, \Gamma \Rightarrow \Delta}\,{\scriptstyle L\supset}$$

Table 1: Labelled calculus **G3I** for intuitionistic logic [2]

model $\mathfrak{M} = \langle \mathfrak{F}, \Vdash \rangle$ is a frame $\mathfrak{F}$ together with a relation $\Vdash$ (called the forcing) between elements of W and atomic formulas. Intuitively, $x \Vdash p$ means that the atom p is true at x. Forcing is assumed to be monotonic w.r.t. the relation $\leqslant$, namely, if $x \leqslant y$ and $x \Vdash p$ then also $y \Vdash p$. It is defined inductively on arbitrary formulas as follows:

$(\Vdash \bot)$	$x \Vdash \bot$	for no x	
$(\Vdash \&)$	$x \Vdash \varphi \& \psi$	iff	$x \Vdash \varphi$ and $x \Vdash \psi$
$(\Vdash \vee)$	$x \Vdash \varphi \vee \psi$	iff	$x \Vdash \varphi$ or $x \Vdash \psi$
$(\Vdash \supset)$	$x \Vdash \varphi \supset \psi$	iff	$x \leqslant y$ and $y \Vdash \varphi$ implies $y \Vdash \psi$.

Intermediate logics are obtained from intuitionistic logic by imposing on intuitionistic frames additional conditions on the relation $\leqslant$. The latter conditions are usually expressed as formulas of first-order classical logic in which variables are interpreted as elements of W, and the binary predicate $\leqslant$ denotes the accessibility relation of $\mathfrak{F}$. Atomic formulas are *relational atoms* of the form $x \leqslant y$. Compound formulas are built from relational atoms using the propositional connectives $\wedge$, $\vee$, $\rightarrow$, $\neg$, and the quantifiers $\forall$ and $\exists$.

Labelled systems are a variant of sequent calculus in which the relational semantics of the formalized logics is made explicit part of the syntax [3, 5, 4]. In a labelled system, each formula φ receives a label x, indicated by $x : \varphi$. The labels are interpreted as possible worlds, and a labelled formula $x : \varphi$ corresponds to $x \Vdash \varphi$. Moreover, labels may occur also in expressions for accessibility relation (relational atoms) like, e.g., $x \leqslant y$ of intuitionistic and intermediate logics.

Definition 1 *A labelled sequent is a sequent consisting of labelled formulas and relational atoms.*

Table 1 depicts the labelled calculus **G3I** for intuitionistic logic. Note that its logical rules are obtained directly from the inductive definition of forcing. The rule $R \supset$ must satisfy the *eigenvariable* condition (y does not occur in the conclusion). The structural rules Ref and $Trans$ for relational atoms correspond to the assumptions of reflexivity and transitivity of $\leqslant$ in $\mathfrak{F}$.

3 From frame conditions to labelled rules

We introduce a classification of frame conditions that is basically the arithmetical hierarchy. W.l.o.g. we will consider formulas in prenex form. The class to which a formula belongs is determined by the alternation of universal and existential quantifiers in the prefix:

Definition 2 ([1]) *The classes* Π_k *and* Σ_k *are defined as follows:* $A \in \Sigma_0$ *and* $A \in \Pi_0$*, if* A *is quantifier-free. Otherwise:*

- *if* A *is classically equivalent to* $\exists \overline{x} B$ *where* $B \in \Pi_n$ *then* $A \in \Sigma_{n+1}$
- *if* A *is classically equivalent to* $\forall \overline{x} B$ *where* $B \in \Sigma_n$ *then* $A \in \Pi_{n+1}$

The transformation procedure introduced in [1] works for formulas that are within Π_2. Note that the *geometric* formulas introduced in [2] are formulas within Π_2, and the rules for geometric formulas presented in [2] are interderivable with the rules generated by our transformation procedure.
The frame condition

- $\forall x, y, z \exists w (x \leq y \wedge x \leq z \rightarrow y \leq w \wedge z \leq w)$

is within the class Π_2 of the hierarchy. Using the algorithm described in [1], we transform the frame condition into the following structural rule:

$$\frac{\Gamma \Rightarrow \Delta, x \leq y \qquad \Gamma \Rightarrow \Delta, x \leq z \qquad y \leq w, z \leq w, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}$$

Let **G3SI*** be the labelled calculus obtained by adding to **G3I** initial sequents of the form $x \leq y, \Gamma \Rightarrow \Delta, x \leq y$ and the rule stated above.

Theorem 3 (Soundness and Completeness) G3SI* *is sound and complete for the logic defined by imposing on the standard intuitionistic frame the above frame condition.*

Proof. See [1].

Theorem 4 (Cut elimination) G3SI* *admits cut elimination.*

Proof. See [1].

References

[1] A. Ciabattoni, P. Maffezioli, and L. Spendier. Hypersequent and Labelled Calculi for Intermediate Logics. In *Proceedings of TABLEAUX 2013*, D. Galmiche and D. Larchey-Wendling (Eds.), LNCS 8123, pp. 81–96, 2013.

[2] R. Dyckhoff and S. Negri. Proof analysis in intermediate logics. *Archive for Mathematical Logic*, 51(1-2): 71–92, 2012.

[3] D. Gabbay. *Labelled Deductive Systems: Foundations.* Oxford Univ. Press, 1996.

[4] S. Negri. Proof analysis in non-classical logics. In *Logic Coll. 2005*, pp. 107–128, 2007.

[5] L. Vigano. *Labelled Non-Classical Logics.* Kluwer, 2000.

C.3 Example LaTeX-output of *Paralyzer*

The following pages contain the file generated by *Paralyzer* for the example shown in Section 6.6.1.

A sequent calculus and effective semantics for BK $+ \star_2\alpha \to \star_2(\alpha \wedge \beta), (\star_1\alpha \vee \star_1\beta) \to \star_1(\alpha \wedge \beta), \star_1 \star_1 \alpha \to \alpha$

Paralyzer*

March 5, 2015

Abstract

We introduce a sequent calculus, its encoding in *Isabelle* and effective semantics using partial non-deterministic matrices for the Hilbert system BK extended with the axioms $\star_2\alpha \to \star_2(\alpha \wedge \beta), (\star_1\alpha \vee \star_1\beta) \to \star_1(\alpha \wedge \beta), \star_1\star_1\alpha \to \alpha$. The calculus, its encoding and the semantics are generated by the PROLOG-program *Paralyzer* (PARAconsistent logics anaLYZER), which implements the procedure in [4, 5].

1 Introduction

Non-classical logics are often introduced using Hilbert systems. The usefulness of these logics, however, strongly depends on two essential components. The first is an intuitive semantics, which can provide insights into the logic. A desirable property of such semantics is effectiveness, in the sense that it naturally induces a decision procedure for the logic. Examples of effective semantics include finite-valued matrices, and their generalizations: non-deterministic finite-valued matrices (Nmatrices) and partial Nmatrices (PNmatrices), see [1, 3]. The second component is a corresponding analytic calculus, i.e. a calculus whose proofs only consist of concepts already contained in the result. Analytic calculi are useful for establishing various properties of the formalized logics, and are also the key for developing automated reasoning methods for them.

In this paper we introduce a sequent calculus for the logic obtained by extending the Hilbert system BK with the axioms $\star_2\alpha \to \star_2(\alpha \wedge \beta), (\star_1\alpha \vee \star_1\beta) \to \star_1(\alpha \wedge \beta), \star_1 \star_1 \alpha \to \alpha$. Recall that BK is an Hilbert axiomatization of CL^+, the positive fragment of classical propositional logic, extended with the axioms $\star_1\alpha \vee \alpha$, $\star_2\alpha \supset (\alpha \wedge \star_1\alpha \supset \beta)$ and $\star_2\alpha \vee (\alpha \wedge \star_2\alpha)$. Our calculus is used to introduce a PNmatrix semantics for the considered logic. Though the resulting calculus might not be analytic in the sense of [5], the corresponding PNmatrix establishes the decidability of the considered logic [4, 5] and can be used to define a family of analytic sequent calculi for it, using the method in [2]. In addition, we provide an encoding of the introduced calculus for the automated theorem prover *Isabelle* [7]. This allows us to perform (semi-)automated proof search within the logic.

*http://www.logic.at/tinc/webparalyzer/

$$\frac{}{\alpha \Rightarrow \alpha}\ (init) \qquad \frac{\Gamma, \alpha \Rightarrow \beta}{\Gamma \Rightarrow \alpha \supset \beta, \Delta}\ (\supset, r) \qquad \frac{\Gamma \Rightarrow \alpha, \Delta \quad \Gamma, \beta \Rightarrow \Delta}{\Gamma, \alpha \supset \beta \Rightarrow \Delta}\ (\supset, l)$$

$$\frac{\Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta, \alpha}\ (w, r) \qquad \frac{\Gamma, \alpha, \beta \Rightarrow \Delta}{\Gamma, \alpha \wedge \beta \Rightarrow \Delta}\ (\wedge, l) \qquad \frac{\Gamma \Rightarrow \alpha, \Delta \quad \Gamma \Rightarrow \beta, \Delta}{\Gamma \Rightarrow \alpha \wedge \beta, \Delta}\ (\wedge, r)$$

$$\frac{\Gamma \Rightarrow \Delta}{\Gamma, \alpha \Rightarrow \Delta}\ (w, l) \qquad \frac{\Gamma \Rightarrow \alpha, \beta, \Delta}{\Gamma \Rightarrow \alpha \vee \beta, \Delta}\ (\vee, r) \qquad \frac{\Gamma, \alpha \Rightarrow \Delta \quad \Gamma, \beta \Rightarrow \Delta}{\Gamma, \alpha \vee \beta \Rightarrow \Delta}\ (\vee, l)$$

$$\frac{\Gamma, \alpha \Rightarrow \Delta}{\Gamma \Rightarrow \Delta, \star_1\alpha}\ (\star_1, r) \qquad \frac{\Gamma, \star_1\alpha, \alpha \Rightarrow \Delta}{\Gamma \Rightarrow \Delta, \star_2\alpha}\ (\star_2, r) \qquad \frac{\Gamma \Rightarrow \alpha, \Delta \quad \Gamma \Rightarrow \star_1\alpha, \Delta}{\Gamma, \star_2\alpha \Rightarrow \Delta}\ (\star_2, l)$$

$$\frac{\Gamma \Rightarrow \Delta, \alpha \quad \alpha, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}\ (cut)$$

Table 1: the sequent system for BK

2 Preliminaries

In what follows $\mathcal{L}$ is the language of CL$^+$, consisting of atomic formulas $\{p_i\}$, the binary connectives $\wedge$ (conjunction), $\vee$ (disjunction) and $\supset$ (implication) extended with the unary connectives $\{\star_1, \star_2\}$.

In this paper, metavariables $\alpha, \beta, \varphi, \ldots$ denote formulas, and $\Gamma, \Delta, \ldots$ stand for finite (possibly empty) multisets of formulas.

A sequent calculus G for $\mathcal{L}$ consists of a finite set of rules. We write $\mathcal{S} \vdash_G s$ whenever the sequent s is derivable from the set $\mathcal{S}$ of sequents in G.

Definition 1 *A sequent calculus G is equivalent to a Hilbert system H if for every finite set $\Gamma \cup \{\varphi\}$ of formulas: φ is provable in H from Γ (in symbols $\Gamma \vdash_H \varphi$) iff $\Gamma \Rightarrow \varphi$ is provable in G (in symbols $\vdash_G \Gamma \Rightarrow \varphi$).*

We denote by H the Hilbert system for $\mathcal{L}$ obtained by extending BK with the axioms $\star_2\alpha \rightarrow \star_2(\alpha \wedge \beta), (\star_1\alpha \vee \star_1\beta) \rightarrow \star_1(\alpha \wedge \beta), \star_1 \star_1 \alpha \rightarrow \alpha$. We define a sequent calculus G for $\mathcal{L}$, which is equivalent to H, by extending the sequent system for BK (see Table 1) with logical rules corresponding to the aforementioned axioms. From the obtained sequent calculus, we will extract effective semantics using partial non-deterministic matrices (PNmatrices), which are a generalization of non-deterministic finite-valued matrices (Nmatrices) [1].

Definition 2 *A* partial non-deterministic matrix *(*PNmatrix*) $\mathcal{M}$ for $\mathcal{L}$ consists of: (i) a set $\mathcal{V}_\mathcal{M}$ of truth values, (ii) a subset $\mathcal{D}_\mathcal{M} \subseteq \mathcal{V}_\mathcal{M}$ (designated truth values), and (iii) a truth-table $\diamond_\mathcal{M} : {\mathcal{V}_\mathcal{M}}^n \rightarrow P(\mathcal{V}_\mathcal{M})$ for every n-ary connective $\diamond$ of $\mathcal{L}$.*

Remark. If there are no empty spots in the PNmatrix, i.e., the set of truth values $\mathcal{V}_\mathcal{M}$ is not empty and no truth table contains a $P(\mathcal{V}_\mathcal{M}) = \emptyset$, the PNmatrix is an ordinary Nmatrix.

Definition 3 *Let $\mathcal{M}$ be a PNmatrix for $\mathcal{L}$.*

1. *An $\mathcal{M}$-*valuation *for $\mathcal{L}$ is a function $v : \mathcal{L} \rightarrow \mathcal{V}_\mathcal{M}$ that respects the truth tables of $\mathcal{M}$, i.e. $v(\diamond(\varphi_1, \ldots, \varphi_n)) \in \diamond_\mathcal{M}(v(\varphi_1), \ldots, v(\varphi_n))$ for every compound formula $\diamond(\varphi_1, \ldots, \varphi_n) \in \mathcal{L}$.*

2. *An $\mathcal{M}$-valuation v for $\mathcal{L}$* satisfies *an $\mathcal{L}$-formula φ (with respect to $\mathcal{M}$; denoted by $v \models_{\mathcal{M}} \varphi$) if $v(\varphi) \in \mathcal{D}_{\mathcal{M}}$.*
3. *Given an $\mathcal{L}$-sequent s, $\vdash_{\mathcal{M}} s$ if $v \models_{\mathcal{M}} s$ for every $\mathcal{M}$-valuation v for $\mathcal{L}$.*

3 Sequent calculus and effective semantics

3.1 From axiom to rules

Using the algorithm described in [4, 5], the axioms

$$\star_2\alpha \to \star_2(\alpha \wedge \beta), (\star_1\alpha \vee \star_1\beta) \to \star_1(\alpha \wedge \beta), \star_1 \star_1 \alpha \to \alpha$$

are transformed into the following logical rules:

$$\frac{\Gamma, \star_1\alpha \Rightarrow \Delta \qquad \Gamma \Rightarrow \alpha, \Delta \qquad \Gamma \Rightarrow \beta, \Delta}{\Gamma, \star_1(\alpha \wedge \beta) \Rightarrow \Delta} \qquad \frac{\Gamma \Rightarrow \star_1\alpha, \Delta}{\Gamma \Rightarrow \star_1(\alpha \wedge \beta), \Delta}$$

$$\frac{\Gamma \Rightarrow \star_1\beta, \Delta}{\Gamma \Rightarrow \star_1(\alpha \wedge \beta), \Delta} \qquad \frac{\Gamma, \alpha \Rightarrow \Delta}{\Gamma, \star_1 \star_1 \alpha \Rightarrow \Delta}$$

Let G be the sequent calculus obtained by extending the calculus for the sequent system for BK (see Table 1) with the above rules.

Theorem 4 *G is equivalent to H.*

Proof. See [4, 5].

3.2 From rules to semantics

The sequent calculus G is used to extract semantics for the considered logic. The PNmatrix $\mathcal{M}$ for G is defined as follows:

$\mathcal{V}_{\mathcal{M}} = \{011, 101, 110\}$
$\mathcal{D}_{\mathcal{M}} = \{101, 110\}$

The truth tables for the unary connectives are as follows:

$\star_1$	
011	$\{101\}$
101	$\{011\}$
110	$\{101, 110\}$

$\star_2$	
011	$\{101, 110\}$
101	$\{101, 110\}$
110	$\{011\}$

The truth tables for the binary connectives are as follows:

$\tilde{\wedge}$	011	101	110
011	{011}	{011}	{011}
101	{011}	{101}	∅
110	{011}	{110}	{110}

$\tilde{\vee}$	011	101	110
011	{011}	{101, 110}	{101, 110}
101	{101, 110}	{101, 110}	{101, 110}
110	{101, 110}	{101, 110}	{101, 110}

$\tilde{\supset}$	011	101	110
011	{101, 110}	{101, 110}	{101, 110}
101	{011}	{101, 110}	{101, 110}
110	{011}	{101, 110}	{101, 110}

Theorem 5 (Soundness and Completeness) $\vdash_G s$ *iff* $\vdash_{\mathcal{M}} s$.

Proof. See [4, 5].

Theorem 6 (Decidability) *Given a finite set* $\Gamma \cup \{\varphi\}$ *of formulas, it is decidable whether* $\Gamma \vdash_H \varphi$ *or not.*

Proof. See [4, 5].

Remark. There are empty spots in the truth tables of the connectives. Hence, the sequent calculus G might not be analytic. However, a family of cut-free sequent calculi for $\mathcal{L}$ can be constructed starting from the PNmatrix $\mathcal{M}$, using the method in [2], see [4, 5] for details.

3.3 An *Isabelle*-encoding of G

We provide an encoding of the sequent calculus G for the generic proof assistant *Isabelle* [7] which leads to a (semi-)automated theorem prover for the logic underlying the calculus. Note that in the encoding, a single formula is denoted by an upper-case letter (e.g., `P`) whereas (possibly empty) sequences of formulas are denoted by upper-case letters preceded with a $ sign (e.g., `$H`, `$G`, `$E,$F`). Premises are encoded left of `==>` and the conclusion of the rule is encoded right of `==>`. If there is more than one premise, the premises are within brackets `[|`, `|]` and comma-separated (`;`). We use $\sim$ to denote $\star_1$ and `+` for $\star_2$.

Encoding: ParalyzerEncoding.thy

```
(*  Title:      Sequents/ParalyzerEncoding.thy
Author:     Paralyzer, Vienna University of Technology
            based on LK0.thy by L. Paulson, University of Cambridge
Copyright   2013 Vienna University of Technology
*)

header {* Propositonal Paraconsistent Logic BK +
(*2 a-> *2 (a&b);*1 a v *1 b-> *1 (a&b);*1*1 a->a) *}

theory ParalyzerEncoding
imports Sequents
```

```
begin

consts
  Trueprop       :: "two_seqi"
  conj         :: "[o,o] => o"       (infixr "&" 35)
  disj         :: "[o,o] => o"       (infixr "|" 30)
  imp          :: "[o,o] => o"       (infixr "-->" 25)
  not          :: "o => o"           ("~ _" [40] 40)
  circ         :: "o => o"           ("+ _" [39] 39)

syntax
 "_Trueprop"    :: "two_seqe" ("((_)/ |- (_))" [6,6] 5)

parse_translation {* [(@{syntax_const "_Trueprop"},
   two_seq_tr @{const_syntax Trueprop})] *}
print_translation {* [(@{const_syntax Trueprop},
   two_seq_tr' @{syntax_const "_Trueprop"})] *}

axioms

 (*Structural rules: contraction, thinning, exchange*)

  contRS: "$H |- $E, $S, $S, $F ==> $H |- $E, $S, $F"
  contLS: "$H, $S, $S, $G |- $E ==> $H, $S, $G |- $E"

  thinRS: "$H |- $E, $F ==> $H |- $E, $S, $F"
  thinLS: "$H, $G |- $E ==> $H, $S, $G |- $E"

  exchRS: "$H |- $E, $R, $S, $F ==> $H |- $E, $S, $R, $F"
  exchLS: "$H, $R, $S, $G |- $E ==> $H, $S, $R, $G |- $E"

  cut:   "[| $H |- $E, P;  $H, P |- $E |] ==> $H |- $E"

 (*Propositional rules*)

  basic: "$H, P, $G |- $E, P, $F"

  conjR: "[| $H|- $E, P, $F;  $H|- $E, Q, $F |] ==> $H|- $E, P&Q, $F"
  conjL: "$H, P, Q, $G |- $E ==> $H, P & Q, $G |- $E"

  disjR: "$H |- $E, P, Q, $F ==> $H |- $E, P|Q, $F"
  disjL: "[| $H, P, $G |- $E;  $H, Q, $G |- $E |] ==> $H, P|Q, $G |- $E"

  impR:  "$H, P |- $E, Q, $F ==> $H |- $E, P-->Q, $F"
  impL:  "[| $H,$G |- $E,P;  $H, Q, $G |- $E |] ==> $H, P-->Q, $G |- $E"

  (** new **)
    notR:  "$H, P |- $E, $F ==> $H |- $E, ~P, $F"
  circR: "$H, P, ~P |- $E, $F ==> $H |- $E, +P, $F"
  circL: "[| $H, $G |- $E, P; $H, $G |- $E, ~P |] ==> $H, +P, $G |- $E"

ruleStar1: "[| $G,~P,$H |- $E; $G, $H |- $E, P; $G, $H |- $E, Q|] ==> $G,~(P&Q),$H |- $E"
ruleStar2: "$G |- $E, ~P,$F ==> $G |- $E, ~(P&Q),$F"
ruleStar3: "$G |- $E, ~Q,$F ==> $G |- $E, ~(P&Q),$F"
```

```
ruleStar4: "$G,P,$H |- $E ==> $G,~~P,$H |- $E"


(** Structural Rules on formulas **)

(*contraction*)

lemma contR: "$H |- $E, P, P, $F ==> $H |- $E, P, $F"
  by (rule contRS)

lemma contL: "$H, P, P, $G |- $E ==> $H, P, $G |- $E"
  by (rule contLS)

(*thinning*)

lemma thinR: "$H |- $E, $F ==> $H |- $E, P, $F"
  by (rule thinRS)

lemma thinL: "$H, $G |- $E ==> $H, P, $G |- $E"
  by (rule thinLS)

(*exchange*)

lemma exchR: "$H |- $E, Q, P, $F ==> $H |- $E, P, Q, $F"
  by (rule exchRS)

lemma exchL: "$H, Q, P, $G |- $E ==> $H, P, Q, $G |- $E"
  by (rule exchLS)

(*Tactic*)

ML {*
    val apply_tac =
    let
      val rules = @{thms basic  conjL conjR disjL disjR impL impR
       notR circR circL  ruleStar1 ruleStar2 ruleStar3 ruleStar4}
    in
      atac
      ORELSE' resolve_tac rules
    end
*}
end
```

References

[1] A. Avron and A. Zamansky. Non-deterministic semantics for logical systems - A survey. In D. Gabbay and F. Guenther, editors, *Handbook of Philosophical Logic*, volume 16, pages 227–304. Springer, 2011.

[2] A. Avron, J. Ben-Naim, and B. Konikowska. Cut-free ordinary sequent calculi for logics having generalized finite-valued semantics. *Logica Universalis*, 1:41–69, 2006.

[3] M. Baaz, O. Lahav, and A. Zamansky. Finite-valued Semantics for Canonical Labelled Calculi. *Journal of Automated Reasoning*, 51(4):401–430, 2013.

[4] A. Ciabattoni, O. Lahav, L. Spendier and A. Zamansky. Automated Support for the Investigation of Paraconsistent and Other Logics. In *Logical Foundations of Computer Science (Lecture Notes in Computer Science)*, Vol. 7734. Springer, 119–133.

[5] A. Ciabattoni, O. Lahav, L. Spendier and A. Zamansky. Taming Paraconsistent (and Other) Logics: An Algorithmic Approach. Accepted for publication in ACM Transactions on Computational Logic (TOCL), 2014.

[6] A. Ciabattoni and L. Spendier. Tools for the Investigation of Substructural and Paraconsistent Logics. In *Proceedings of JELIA 2014, LNAI*, pages 18–32, 2014.

[7] M. Wenzel, L. C. Paulson and T. Nipkow. The Isabelle Framework. In *Proceedings of Theorem Proving in Higher Order Logics 2008 (Lecture Notes in Computer Science)*. 33–38, 2008.

Bibliography

[1] P. Abate and R. Goré. The Tableaux Work Bench. In *Proceedings of Automated Reasoning with Analytic Tableaux and Related Methods*, LNCS, pages 230–236, 2003.

[2] R. Alenda, N. Olivetti, and G. L. Pozzato. Nested sequent calculi for conditional logics. In L. F. del Cerro, A. Herzig, and J. Mengin, editors, *Proceedings of the European Conference on Logics in Artifical Intelligence (JELIA)*, volume 7519 of *LNCS*, pages 14–27, 2012.

[3] A. Almukdad and D. Nelson. Constructible falsity and inexact predicates. *Journal of Symbolic Logic*, 49:231–233, 1984.

[4] A. R. Anderson and N. D. Belnap. *Entailment: The Logic of Relevance and Necessity.* Princeton University Press, Princeton, 1975.

[5] J.-M. Andreoli. Logic programming with focusing proofs in linear logic. *Journal of Logic and Computation*, 2(3):297–347, 1992.

[6] O. Arieli and A. Avron. Reasoning with logical bilattices. *Journal of Logic, Language and Information*, 5(1):25–63, 1996.

[7] F. G. Asenjo. A calculus of antinomies. *Notre Dame Journal of Formal Logic*, 7(1):103–105, 1966.

[8] A. Avellone, M. Ferrari, and P. Miglioli. Duplication-free tableau calculi and related cut-free sequent calculi for the interpolable propositional intermediate logics. *Logic Journal of the IGPL*, 7(4):447–480, 1999.

[9] A. Avron. A constructive analysis of RM. *Journal of Symbolic Logic*, 52(4):939–951, 1987.

[10] A. Avron. Hypersequents, logical consequence and intermediate logics for concurrency. *Annals of Mathematics and Artificial Intelligence*, 4(3-4):225–248, 1991.

[11] A. Avron. The method of hypersequents in the proof theory of propositional non-classical logics. In W. Hodges, M. Hyland, C. Steinhorn, and J. Truss, editors, *Logic: from foundations to applications*, pages 1–32. Clarendon Press, 1996.

[12] A. Avron. Non-deterministic matrices and modular semantics of rules. *Logica Universalis*, pages 155–173, 2007.

[13] A. Avron. Non-deterministic Semantics for Families of Paraconsistent Logics. In J.-Y. Beziau, W. A. Carnielli, and D. Gabbay, editors, *Handbook of Paraconsistency*, volume 9 of *Studies in Logic*, pages 285–320. College Publications, 2007.

[14] A. Avron, J. Ben-Naim, and B. Konikowska. Cut-free ordinary sequent calculi for logics having generalized finite-valued semantics. *Logica Universalis*, 1:41–69, 2006.

[15] A. Avron and B. Konikowska. Proof systems for logics based on non-deterministic multiple-valued structures. *Logic Journal of the IGPL*, 13:365–387, 2005.

[16] A. Avron, B. Konikowska, and A. Zamansky. A systematic generation of analytic calculi for Logics of Formal Inconsistency. In J.-Y. Beziau and M. E. Coniglio, editors, *Logic without Frontiers: Festschrift for W.A. Carnielli on the occasion of his 60th Birthday*, volume 17 of *Tribute*. College Publications, 2011.

[17] A. Avron, B. Konikowska, and A. Zamansky. Modular construction of cut-free sequent calculi for paraconsistent logics. In *Logic in Computer Science (LICS), 2012 27th Annual IEEE Symposium on*, pages 85–94, 2012.

[18] A. Avron, B. Konikowska, and A. Zamansky. Cut-free sequent calculi for C-systems with generalized finite-valued semantics. *Journal of Logic and Computation*, 21(3):517–540, 2013.

[19] A. Avron and I. Lev. Canonical propositional Gentzen-type systems. In R. Goré, A. Leitsch, and T. Nipkow, editors, *Proceedings of the International Joint Conference on Automated Reasoning (IJCAR)*, volume 2083 of *LNAI*, pages 529–544, 2001.

[20] A. Avron and I. Lev. Non-deterministic multiple-valued structures. *Journal of Logic and Computation*, 15(3):241–261, 2005.

[21] M. Baaz, A. Ciabattoni, and F. Montagna. Analytic calculi for monoidal t-norm based logic. *Fundamenta Informaticae*, 59(4):315–332, 2004.

[22] M. Baaz, C. G. Fermüller, G. Salzer, and R. Zach. MUltlog 1.0: Towards an expert system for many-valued logics. In *Proceedings of the International Conference on Automated Deduction (CADE)*, LNCS, pages 226–230, 1996.

[23] M. Baaz, O. Lahav, and A. Zamansky. Finite-valued semantics for canonical labelled calculi. *Journal of Automated Reasoning*, 51(4):401–430, 2013.

[24] M. Baaz and R. Zach. Hypersequents and the proof theory of intuitionistic fuzzy logic. In *Proceedings of Computer Science Logic (CSL)*, volume 1862 of *LNCS*, pages 187–201, 2000.

[25] P. Baldi. A note on standard completeness for some extensions of uninorm logic. *Soft Computing*, 18(8):1463–1470, 2014.

[26] P. Baldi and A. Ciabattoni. Standard completeness for uninorm-based logics. In *Accepted for the IEEE International Symposium on Multiple-Valued Logics (ISMVL)*, 2015.

[27] P. Baldi and A. Ciabattoni. Uniform proofs of standard completeness for extensions of first-order MTL. *Theoretical Computer Science*, Accepted for publication.

[28] P. Baldi, A. Ciabattoni, and L. Spendier. Standard completeness for extensions of MTL: an automated approach. In L. Ong and R. Queiroz, editors, *Proceedings of Logic, Language, Information, and Computation (WoLLIC)*, volume 7456 of *LNCS*, pages 154–167. Springer, 2012.

[29] S. Barro and R. Marin, editors. *Fuzzy Logic in Medicine*, volume 83 of *Studies in Fuzziness and Soft Computing*. Springer, 2002.

[30] D. Batens. Dialectical dynamics within formal logics. *Logique et Analyse*, 114:161–173, 1986.

[31] D. Batens. Dynamic dialectical logics. In G. Priest, R. Routley, and J. Norman, editors, *Paraconsistent Logic. Essays on the Inconsistent*, pages 187–217. Philosophia Verlag, München, 1989.

[32] D. Batens. The need for adaptive logics in epistemology. In S. Rahman, J. Symons, D. Gabbay, and J. P. van Bendegem, editors, *Logic, Epistemology, and the Unity of Science*, pages 459–485. Kluwer Academic Publishers, 2004.

[33] N. D. Belnap. A useful four-valued logic. In J. M. Dunn and G. Eppstein, editors, *Modern Uses of Multiple-Valued Logic*, pages 7–37. Reidel, Dordrecht, 1977.

[34] N. D. Belnap. Display logic. *Journal of Philosophical Logic*, 11(4):375–417, 1982.

[35] C. Benzmüller and B. Woltzenlogel Paleo. Automating Gödel's ontological proof of God's existence with higher-order automated theorem provers. In T. Schaub, G. Friedrich, and B. O'Sullivan, editors, *Proceedings of 21st European Conference on Artificial Intelligence (ECAI 2014)*, pages 93–98, 2014.

[36] L. Bertossi, A. Hunter, and T. Schaub. Introduction to inconsistency tolerance. In L. Bertossi, A. Hunter, and T. Schaub, editors, *Inconsistency Tolerance*, pages 1–14. Springer, 2005.

[37] Y. Bertot and P. Castéran. *Interactive Theorem Proving and Program Development – Coq'Art: The Calculus of Inductive Constructions*. Springer, 2004.

[38] N. Bezhanishvili and S. Ghilardi. Bounded proofs and step frames. In D. Galmiche and D. Larchey-Wendling, editors, *Proceedings of Automated Reasoning with Analytic Tableaux and Related Methods*, volume 8123 of *LNCS*, pages 44–58. Springer, 2013.

[39] N. Bezhanishvili and S. Ghilardi. The bounded proof property via step algebras and step frames. *Annals of Pure and Applied Logic*, 165(12):1832–1863, 2014.

[40] J. Brotherston. Bunched logics displayed. *Studia Logica*, 100(6):1223–1254, 2012.

[41] K. Brünnler. Deep sequent systems for modal logic. *Archive for Mathematical Logic*, 48(6):551–577, 2009.

[42] K. Brünnler. Nested sequents. *habilitation thesis*, Universität Bern, 2010.

[43] S. R. Buss, editor. *Handbook Proof Theory*. Elsevier, 1998.

[44] S. R. Buss. *An introduction to proof theory*, pages 1–78. Handbook Proof Theory. Elsevier, 1998.

[45] W. A. Carnielli, M. E. Coniglio, and J. Marcos. Logics of formal inconsistency. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic*, volume 14, pages 15–107. Springer, 2007.

[46] W. A. Carnielli and J. Marcos. Tableau systems for logics of formal inconsistency. In *Proceedings of the International Conference on Artificial Intelligence (ICAI)*, pages 848–852, 2001.

[47] W. A. Carnielli and J. Marcos. A taxonomy of C-systems. In W. A. Carnielli, M. E. Coniglio, and I. D'Ottaviano, editors, *Paraconsistency: The Logical Way to the Inconsistent*, number 228 in Lecture Notes in Pure and Applied Mathematics, pages 1–94. Marcel Dekker, 2002.

[48] C. Castellini. *Automated Reasoning in Quantified Modal and Temporal Logics*. PhD thesis, University of Edinburgh, 2004.

[49] A. Chagrov and M. Zakharyaschev. *Modal Logic*. Clarendon Press, Oxford, 1997.

[50] A. Ciabattoni, F. Esteva, and L. Godo. T-norm based logics with n-contraction. *Neural Network World*, 12(5):441–452, 2002.

[51] A. Ciabattoni and M. Ferrari. Hypertableau and path-hypertableau calculi for some families of intermediate logics. In *Proceedings of Automated Reasoning with Analytic Tableaux and Related Methods*, volume 1847 of *LNAI*, pages 160–175, 2000.

[52] A. Ciabattoni, N. Galatos, and K. Terui. From axioms to analytic rules in non-classical logics. In *23rd Annual IEEE Symposium on Logic in Computer Science, 2008 (LICS '08)*, pages 229–240, 2008.

[53] A. Ciabattoni, N. Galatos, and K. Terui. MacNeille completions of FL-algebras. *Algebra Universalis*, 66(4):229–240, 2011.

[54] A. Ciabattoni, N. Galatos, and K. Terui. Algebraic proof theory for substructural logics: cut-elimination and completions. *Annals of Pure and Applied Logic*, 163(3):266–290, 2012.

[55] A. Ciabattoni, O. Lahav, L. Spendier, and A. Zamansky. Automated support for the investigation of paraconsistent and other logics. In *Proceedings of Logical Foundations of Computer Science (LFCS)*, volume 7734 of *LNCS*, pages 119–133. Springer, 2013.

[56] A. Ciabattoni, O. Lahav, L. Spendier, and A. Zamansky. Taming paraconsistent (and other) logics: An algorithmic approach. *ACM Transactions on Computational Logic*, 16(1):5:1–5:23, 2015.

[57] A. Ciabattoni, P. Maffezioli, and L. Spendier. Hypersequent and labelled calculi for intermediate logics. In D. Galmiche and D. Larchey-Wendling, editors, *Proceedings of Automated Reasoning with Analytic Tableaux and Related Methods*, volume 8123 of *LNCS*, pages 81–96. Springer, 2013.

[58] A. Ciabattoni and G. Metcalfe. Density elimination. *Theoretical Computer Science*, 403(2-3):328–346, 2008.

[59] A. Ciabattoni and R. Ramanayake. Structural extensions of display calculi: a general recipe. In L. Libkin, U. Kohlenbach, and R. Queiroz, editors, *Proceedings of Logic, Language, Information, and Computation (WoLLIC)*, volume 8071 of *LNCS*, pages 81–95. Springer, 2013.

[60] A. Ciabattoni, R. Ramanayake, and H. Wansing. Hypersequent and display calculi – a unified perspective. *Studia Logica*, 102(6):1245–1294, 2014.

[61] A. Ciabattoni and L. Spendier. Tools for the investigation of substructural and paraconsistent logics. In E. Fermé and J. Leite, editors, *Proceedings of the European Conference on Logics in Artifical Intelligence (JELIA)*, volume 8761 of *LNAI*, pages 18–32, 2014.

[62] A. Ciabattoni, L. Straßburger, and K. Terui. Expanding the realm of systematic proof theory. In E. Grädel and R. Kahle, editors, *Proceedings of Computer Science Logic (CSL)*, volume 5771 of *LNCS*, pages 163–178, 2009.

[63] N. C. A. Da Costa. On the theory of inconsistent formal systems. *Notre Dame Journal of Formal Logic*, 15:497–510, 1974.

[64] M. D'Agostino and D. Gabbay. A generalization of analytic deduction via labelled deductive systems I: Basic substructural logics. *Journal of Automated Reasoning*, 13:243–281, 1994.

[65] M. D'Agostino and M. Mondadori. The taming of the cut. classical refutations with analytic cut. *Journal of Logic and Computation*, 4(3):285–319, 1994.

[66] J. E. Dawson and R. Goré. Embedding display calculi into logical frameworks: Comparing Twelf and Isabelle. *Electronic Notes in Theoretical Computer Science*, 42:89–103, 2001.

[67] J. E. Dawson and R. Goré. Formalised cut admissibility for display logic. In V. A. C. an dCésar A. Munoz and S. Tahar, editors, *Proceedings of Theorem Proving in Higher Order Logic (TPHOL)*, volume 2410 of *LNCS*, pages 131–147, 2002.

[68] P. Deransart, A. Ed-Dbali, and L. Cervoni. *Prolog: The Standard Reference Manual*. Springer, 1996.

[69] I. D'Ottaviano and M. A. de Castro. Analytical tableaux for da costa's hierarchy of paraconsistent logics c_n, $1 \leq n < \omega$. *Journal of Applied Non-classical Logics*, 15(1):69–103, 2005.

[70] I. D'Ottaviano and M. A. de Castro. Analytical tableaux for da Costa's hierarchy of paraconsistent logics. In *Proceedings of Logic, Language, Information, and Computation (WoLLIC)*, volume 143 of *Electronic Notes in Theoretical Computer Science*, pages 27–44, 2006.

[71] M. Dummett. A propositional calculus with denumerable matrix. *Journal of Symbolic Logic*, 24(2):97–106, 1959.

[72] J. M. Dunn. Intuitive semantics for first-degree entailment and 'coupled trees'. *Philosophical Studies*, 29(3):149–168, 1976.

[73] J. M. Dunn. Relevant logic and entailment. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic*, pages 117–224. Reidel, Dordrecht, 1986.

[74] J. M. Dunn. Gaggle theory: An abstraction of Galois connections and residuations with applications to negation and various logical operations. In *Proceedings of the European workshop on Logics in Artifical Intelligence (JELIA)*, volume 478 of *LNCS*, pages 31–51. Springer, 1991.

[75] J. M. Dunn. Partial gaggles applied to logics with restricted structural rules. In K. Dosen and P. Schroeder-Heister, editors, *Substructural Logics*, pages 63–108. Oxford University Press, 1993.

[76] R. Dyckhoff and S. Negri. Proof analysis in intermediate logics. *Archive for Mathematical Logic*, 51(1-2):71–92, 2012.

[77] F. Esteva, J. Gispert, L. Godo, and F. Montagna. On the standard and rational completeness of some axiomatic extensions of the monoidal t-norm logic. *Studia Logica*, 71(2):199–226, 2002.

[78] F. Esteva and L. Godo. Monoidal t-norm based logic: towards a logic for left-continuous t-norms. *Fuzzy Sets and Systems*, 124(3):271–288, 2001.

[79] M. Ferrari and P. Miglioli. Counting the maximal intermediate constructive logics. *Journal of Symbolic Logic*, 58(4):1364–1401, 1993.

[80] M. Fitting. *Proof Methods for Modal and Intuitionistic Logics*. Reidel, Dordrecht, 1983.

[81] M. Fitting. Prefixed tableaus and nested sequents. *Annals of Pure and Applied Logic*, 163:291–313, 2012.

[82] M. Fitting. Nested sequents for intuitionistic logics. *Notre Dame Journal of Formal Logic*, 55(1):41–61, 2014.

[83] D. Gabbay. *Labelled Deductive Systems: Foundations*. Clarendon Press, Oxford, 1996.

[84] D. Gabbay and H. J. Ohlbach. Quantifier elimination in second-order predicate logic. *South African Computer Journal*, 7:35–43, 1992.

[85] N. Galatos, P. Jipsen, T. Kowalski, and H. Ono. *Residuated Lattices: An Algebraic Glimpse at Substructural Logics*, volume 151 of *Studies in Logic and the Foundations of Mathematics*. Elsevier, 2007.

[86] D. Gallin. *Intensional and Higher-Order Modal Logics*. North-Holland, Amsterdam, 1975.

[87] O. Gasquet, A. Herzig, D. Longin, and M. Sahade. LoTREC: Logical tableaux research engineering companion. In *Proceedings of Automated Reasoning with Analytic Tableaux and Related Methods*, volume 3702 of *LNCS*, pages 318–322, 2005.

[88] G. Gentzen. Untersuchungen über das Logische Schließen I und II. *Mathematische Zeitschrift*, 39:176–210, 405–431, 1935.

[89] A. J. Gil and G. Salzer. MUltseq: a generic prover for sequents and equations. In *Collegium Logicum: Annals of the Kurt-Gödel-Society*. Kurt-Gödel-Society, Vienna, 2001.

[90] J.-Y. Girard. Linear logic. *Theoretical Computer Science*, 50(1):1–102, 1987.

[91] K. Gödel. Zum intuitionistischen Aussagenkalkül. *Anzeiger Akademie der Wissenschaften Wien, Math.-naturwissensch. Klasse*, 69:65–66, 1932.

[92] R. Goré. Gaggles, Gentzen and Galois: How to display your favourite substructural logic. *Logic Journal of the IGPL*, 6(5):669–694, 1998.

[93] R. Goré. Substructural logics on display. *Logic Journal of the IGPL*, 6(3):451–504, 1998.

[94] R. Goré, L. Postniece, and A. Tiu. Cut-elimination and proof search for bi-intuitionistic tense logic. In L. Beklemishev, V. Goranko, and V. Shehtman, editors, *Proceedings of Advances in Modal Logic*, volume 8, pages 156–177, 2010.

[95] R. Goré and R. Ramanayake. Labelled tree sequents, tree hypersequents and nested (deep) sequents. In T. Bolander, T. Braüner, S. Ghilardi, and L. Moss, editors, *Proceedings of Advances in Modal Logic*, volume 9 of *College Publications*, pages 279–299, 2012.

[96] A. Guglielmi. A system of interaction and structure. *ACM Transactions on Computational Logic*, 8(1):1–64, 2007.

[97] P. Hájek. *Metamathematics of Fuzzy Logic.* Springer, 1998.

[98] C. Hewitt. Large-scale organizational computing requires unstratified reflection and strong paraconsistency. In *Coordination, Organizations, Institutions, and Norms in Agent Systems III*, volume 4870 of *LNCS*, pages 110–124, 2008.

[99] R. Horcik. Alternative proof of standard completeness theorem for mtl. *Soft Computing*, 11(2):123–129, 2007.

[100] V. A. Jankov. Calculus of the weak law of the excluded middle. *Izv. Akad. Nauk SSSR Ser. Mat.*, 32:1044–1051, 1968.

[101] S. Jaśkowski. Propositional calculus for contradictory deductive system (communicated at the meeting of march 19, 1948). *Studia Logica*, 24:143–160, 1969.

[102] S. Jenei and F. Montagna. A proof of standard completeness for Esteva and Godo's logic MTL. *Studia Logica*, 70(2):183–192, 2002.

[103] J. A. Kalman. *Automated Reasoning with OTTER.* Rinton Press, 2001.

[104] N. Kamide. A cut-free system for 16-valued reasoning. *Bulletin of the Section of Logic*, 34(4):213–225, 2005.

[105] N. Kamide. Proof systems combining classical and paraconsistent negations. *Studia Logica*, 91(2):217–238, 2009.

[106] N. Kamide. A hierarchy of weak double negations. *Studia Logica*, 101(6):1277–1297, 2013.

[107] N. Kamide and H. Wansing. Sequent calculi for some trilattice logics. *The Review of Symbolic Logic*, 2(2):374–395, 2009.

[108] N. Kamide and H. Wansing. Proof theory of Nelson's paraconsistent logic: A uniform perspective. *Theoretical Computer Science*, 415:1–38, 2011.

[109] R. Kashima. Cut-free sequent calculi for some tense logics. *Studia Logica*, 53(1):119–135, 1994.

[110] M. Kracht. Power and weakness of the modal display calculus. In H. Wansing, editor, *Proof Theory of Modal Logic*, volume 2 of *Applied Logic Series*, pages 93–121. Springer, 1996.

[111] G. Kreisel and H. Putnam. Unableitbarkeitsbeweismethode für den intuitionistischen Aussagenkalül. *Archiv für mathematische Logic und Grundlagenforschung*, 3:74–78, 1957.

[112] O. Lahav. From frame properties to hypersequent rules in modal logics. In *28th Annual IEEE/ACM Symposium on Logic in Computer Science (LICS)*, pages 408–417, 2013.

[113] J. Lambek. The mathematics of sentence structure. *American Mathematical Monthly*, 12:166–178, 1958.

[114] J. Lambek. From categorial grammar to bilinear logic. In K. Dosen and P. Schroeder-Heister, editors, *Substructural Logics*, volume 2, pages 207–237. Oxford University Press, 1993.

[115] B. Lellmann. Axioms vs. hypersequent rules with context restrictions: Theory and applications. In S. Demri, D. Kapur, and C. Weidenbach, editors, *Proceedings of the International Joint Conference on Automated Reasoning (IJCAR)*, volume 8562 of *LNCS*, pages 307–321, 2014.

[116] B. Lellmann and D. Pattinson. Cut elimination for shallow modal logics. In *Proceedings of Automated Reasoning with Analytic Tableaux and Related Methods*, LNCS, pages 211–225, 2011.

[117] B. Lellmann and D. Pattinson. Correspondence between modal Hilbert axioms and sequent rules with an application to S5. In D. Galmiche and D. Larchey-Wendling, editors, *Proceedings of Automated Reasoning with Analytic Tableaux and Related Methods*, volume 8123 of *LNCS*, pages 219–233, 2013.

[118] B. Lellmann and D. Pattinson. Constructing cut free sequent systems with context restrictions based on classical or intuitionistic logic. In K. Lodaya, editor, *Proceedings of the International Conference on Logic and Its Applications (ICLA)*, volume 7750 of *LNCS*, pages 148–160, 2014.

[119] J. Łukasiewicz. O logice tojwartosciowej. *Ruch Filozoficzny*, 5:169–171, 1920.

[120] J. Łukasiewicz and A. Tarski. Untersuchungen über den Aussagenkalkül. *Comptes Rendus des Séances de la Societé des Sciences et des Lettres de Varsovie, Classe III*, 23, 1930.

[121] W. MacCaull. Relational proof system for linear and other substructural logics. *Logic Journal of the IGPL*, 5(5):673–697, 1997.

[122] H. M. MacNeille. Partially ordered sets. *Transactions of the American Mathematical Society*, 42:416–460, 1937.

[123] L. Maksimova. Interpolation properties of superintuitionistic logics. *Studia Logica*, 38(4):419–428, 1979.

[124] S. Marin and L. Straßburger. Label-free modular systems for classical and intuitionistic modal logics. In R. Goré, B. Kooi, and A. Kurucz, editors, *Proceedings of Advances in Modal Logic*, volume 10 of *College Publications*, pages 387–406, 2014.

[125] W. McCune. Prover9 and Mace4. `http://www.cs.unm.edu/~mccune/prover9/`, 2005–2010.

[126] G. Metcalfe. Proof theory for Casari's comparative logics. *Journal of Logic and Computation*, 16(4):405–422, 2006.

[127] G. Metcalfe and F. Montagna. Substructural fuzzy logics. *Journal of Symbolic Logic*, 7(3):834–864, 2007.

[128] G. Metcalfe, N. Olivetti, and D. Gabbay. Sequent and hypersequent calculi for Lukasiewicz and Abelian logics. *ACM Transactions on Computational Logic*, 6(3):578–613, 2005.

[129] G. Metcalfe, N. Olivetti, and D. Gabbay. *Proof Theory for Fuzzy Logics*, volume 36 of *Applied Logic Series*. Springer, 2009.

[130] F. Montagna and H. Ono. Kripke semantics, undecidability and standard completeness for Esteva and Godo's logic MTL∀. *Studia Logica*, 71(2):227–245, 2002.

[131] S. Negri. Proof analysis in non-classical logics. In *Logic Colloquium 2005*, pages 107–128. Cambridge University Press, 2007.

[132] S. Negri. Kripke completeness revisited. In G. Primiero and S. Rahman, editors, *Acts of Knowledge - History, Philosophy and Logic*, pages 233–266. College Publications, 2009.

[133] S. Negri. Proof analysis beyond geometric theories: from rule systems to systems of rules. *Journal of Logic and Computation*, 2014, in press.

[134] S. Negri and J. von Plato. *Structural Proof Theory*. Cambridge University Press, 2001.

[135] V. Nigam and D. Miller. Algorithmic specifications in linear logic with subexponentials. In *Proceedings of ACM SIGPLAN Conference on Principles and Practice of Declarative Programming (PPDP)*, pages 129–140, 2009.

[136] V. Nigam, E. Pimentel, and G. Reis. An extended framework for specifying and reasoning about proof systems. *Journal of Logic and Computation*, 2014, in press.

[137] V. Nigam, G. Reis, and L. Lima. Quati: An automated tool for proving permutation lemmas. In S. Demri, D. Kapur, and C. Weidenbach, editors, *Proceedings of the International Joint Conference on Automated Reasoning (IJCAR)*, LNCS, pages 255–261, 2014.

[138] T. Nipkow, L. C. Paulson, and M. Wenzel. *Isabelle/HOL – A Proof Assistant for Higher-Order Logic*, volume 2283 of *LNCS*. Springer, 2002.

[139] H. J. Ohlbach. Logic engineering – Konstruktion von Logiken. *Künstliche Intelligenz*, 6(3):34–38, 1992.

[140] H. J. Ohlbach. Computer support for the development and investigation of logics. *Logic Journal of the IGPL*, 4(1):109–127, 1996.

[141] H. Ono. Substructural logics and residuated lattices – an introduction. In V. F. Hendricks and J. Malinowski, editors, *Trends in Logic: 50 Years of Studia Logica*, volume 21, pages 193–228. Springer, 2003.

[142] E. Orlowska. Relational proof system for relevant logics. *Journal of Symbolic Logic*, 57(4):1425–1440, 1992.

[143] M. Osorio, J. A. Navarro, and J. Arrazola. Applications of intuitionistic logic in Answer Set Programming. *Theory and Practice of Logic Programming*, 4(3):325–354, 2004.

[144] F. Paoli. *Substructural Logics: A Primer*. Kluwer Academic Publishers, Dordrecht, 2002.

[145] D. Pattinson and L. Schröder. Generic modal cut elimination applied to conditional logics. In M. Giese and A. Waaler, editors, *Proceedings of Automated Reasoning with Analytic Tableaux and Related Methods*, volume 5607 of *LNCS*, pages 280–294, 2009.

[146] D. Pearce. Stable inference as intuitionistic validity. *Journal of Logic Programming*, 38(1):79–91, 1999.

[147] F. Pfenning and C. Schürmann. System description: Twelf — a meta-logical framework for deductive systems. In *Proceedings of the International Conference on Automated Deduction (CADE)*, volume 1632 of *LNCS*, pages 202–206, 1999.

[148] N. H. Phuong and V. Kreinovich. Fuzzy logic and its applications in medicine. *International Journal of Medical Informatics*, 62(2–3):165–173, 2001.

[149] F. Poggiolesi. The method of tree-hypersequents for modal propositional logic. In D. Makinson, J. Malinowski, and H. Wansing, editors, *Towards Mathematical Philosophy*, volume 28 of *Trends in Logic*, pages 31–51, 2009.

[150] G. Pottinger. Uniform, cut-free formulations of T, S4 and S5 (abstract). *Journal of Symbolic Logic*, 48(3):900, 1983.

[151] G. Priest. The logic of paradox. *Journal of Philosophical Logic*, 8:219–241, 1979.

[152] R. Ramanayake. Embedding the hypersequent calculus in the display calculus. *Journal of Logic and Computation*, 2014, in press.

[153] N. Rescher and R. Brandom. *The Logic of Inconsistency.* Blackwell, Oxford, 1979.

[154] R. Rothenberg. *On the relationship between hypersequent calculi and labelled sequent calculi for intermediate logics with geometric Kripke semantics.* PhD thesis, University of St. Andrews, 2010.

[155] G. Sambin and S. Valentini. The modal logic of provability. the sequential approach. *Journal of Philosophical Logic*, 11:311–342, 1982.

[156] Y. Shramko and H. Wansing. Some useful 16-valued logics: how a computer network should think. *Journal of Philosophical Logic*, 34:121–153, 2005.

[157] R. Smullyan. *First-Order Logic.* Springer, Berlin, 1968.

[158] L. Sterling and E. Shapiro. *The Art of Prolog.* MIT Press, 1987.

[159] L. Straßburger. Cut elimination in nested sequents for intuitionistic modal logics. In F. Pfenning, editor, *Proceedings of the Foundations of Software Science and Computation Structures*, volume 7794 of *LNCS*, pages 209–224, 2013.

[160] G. Takeuti. *Proof Theory (2nd ed.).* North-Holland, Amsterdam, 1987.

[161] G. Takeuti and S. Titani. Intuitionistic fuzzy logic and intuitionistic fuzzy set theory. *Journal of Symbolic Logic*, 49(3):851–866, 1984.

[162] H. Tews. Formalizing cut elimination of coalgebraic logics in Coq. In *Proceedings of Automated Reasoning with Analytic Tableaux and Related Methods*, volume 8123 of *LNCS*, pages 257–272, 2013.

[163] D. Tishkovsky, R. A. Schmidt, and M. Khodadadi. MetTeL: A tableau prover with logic-independent inference engine. In *Proceedings of Automated Reasoning with Analytic Tableaux and Related Methods*, volume 6793 of *LNCS*, pages 242–247, 2011.

[164] D. Tishkovsky, R. A. Schmidt, and M. Khodadadi. The tableau prover generator MetTeL2. In *Proceedings of the European Conference on Logics in Artifical Intelligence (JELIA)*, LNCS, pages 492–495, 2012.

[165] A. S. Troelstra and H. Schwichtenberg. *Basic proof theory (2nd ed.).* Cambridge University Press, New York, 2000.

[166] T. Vetterlein and A. Ciabattoni. On the (fuzzy) logical content of CADIAG-2. *Fuzzy Sets and Systems*, 161(14):1941–1958, 2010.

[167] L. Viganò. *Labelled Non-Classical Logics*. Kluwer Academic Publishers, Dordrecht, 2000.

[168] S.-M. Wang. Uninorm logic with the n-potency axiom. *Fuzzy Sets and Systems*, 205:116–126, 2012.

[169] H. Wansing. *Displaying Modal Logic*. Trends in Logic. Kluwer Academic Publishers, Dordrecht, 1998.

[170] H. Wansing. Constructive negation, implication, and co-implication. *Journal of Applied Non-classical Logics*, 18(2–3):341–364, 2008.

[171] M. Wenzel, L. C. Paulson, and T. Nipkow. The Isabelle framework. In *Proceedings of Theorem Proving in Higher Order Logic*, LNCS, pages 33–38, 2008.