Jefferson de Barros Santos

# Systems for Provability and Countermodel Generation in Propositional Minimal Implicational Logic

**Tese de Doutorado**

Thesis presented to the Programa de Pós-Graduação em Informática of PUC-Rio in partial fulfillment of the requirements for the degree of Doutor em Ciências – Informática.

Advisor : Prof. Edward Hermann Haeusler
Co–Advisor: Prof. Gilles Dowek

Rio de Janeiro
May 2017

Pontifícia Universidade Católica
do Rio de Janeiro

Jefferson de Barros Santos

# Systems for Provability and Countermodel Generation in Propositional Minimal Implicational Logic

Thesis presented to the Programa de Pós-Graduação em Informática of PUC-Rio in partial fulfillment of the requirements for the degree of Doutor em Ciências – Informática. Approved by the undersigned Examination Committee.

Rio de Janeiro, May 18th, 2017

**Jefferson de Barros Santos**

Doctoral student in Computer Science at Pontifícia Universidade Católica of Rio de Janeiro (PUC-Rio). He has got his M.Sc. and B.Sc. at the same university. Nowadays, he is IT coordinator at Brazilian School of Public and Business Administration of the Getulio Vargas Foundation (EBAPE/FGV). He is also an undergraduate professor of Computer Science subjects in the same institution. He also taught the Introduction to Logic course at PUC-Rio for Engineering and Computer Science students. He was a student visitor in INRIA, France from August to December 2016.

Bibliographic data

Santos, Jefferson de Barros

Systems for Provability and Countermodel Generation in Propositional Minimal Implicational Logic / Jefferson de Barros Santos ; advisor: Edward Hermann Haeusler; co–advisor: Gilles Dowek. — 2017.
82 f. : il. color. ; 30 cm

Tese (doutorado)-Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática, 2017.
Inclui bibliografia

1. Informática – Teses. 2. Lógica. 3. Lógica Proposicional Minimal Implicacional. 4. Cálculo de Sequentes. 5. Busca de Provas. 6. Geração de Contra-modelo. I. Haeusler, Edward Hermann. II. Dowek, Gilles. III. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. IV. Título.

CDD: 004

To my parents, Fatima and Helio, thank you for all your love!

## Acknowledgments

I thank so much my wife, Paula, for all the support during this long journey. I know that even with all my efforts it was not easy to put up with an absent husband in so many important moments. Thank you for staying with me. This period was the richest learning time for us as individuals and as a couple. Thank you for making life always have a motivation, always have something to learn, always have something to change... It is enriching to live beside you. I love you!

I thank my daughter Lara and my son Caio that also suffer from an absent dad. Lara was born when I finished my graduation, Caio, at the end of my master. For so many years, both of you have needed to share your dad's attention with his study. You are the most important achievement of my life.

I am thankful to my advisor and friend, Professor Edward Hermann Haeusler. Thank you for the availability to help me all the time I needed, for the patience to answer even the most fool questions. For a lot of coffee and talks much beyond our research interest. Thank you for believing I could reach this point even when I do not.

I also thank Professor Gilles Dowek, co-advisor of this work, and all Deducteam at INRIA, France. Thank you all for welcoming me into your team, during my visiting period there. Particularly, I would like to thank Frédéric Gilbert for the hospitality and all the research support that you gave me.

I thank all the members of the Examination Committee, Professors Mauricio Ayala-Rincon, Mario Benevides, Bruno Lopes, Luiz Carlos Pereira and Eduardo Laber. Thank you for the review of my work and all the comments and contributions that help improve so much my final document.

I am thankful to the Brazilian School of Public and Business Administration of the Getulio Vargas Foundation (FGV/EBAPE). Especially to Professor Flavio Vasconcelos, dean of the school, and professor Alvaro Cyrino, the school's vice-dean, for supporting me during all this time, allowing me to share my time between my functions in the school and the research work.

I also would like to thank Professor Roy Dyckhoff for the all the valuable revisions, critics, and suggestions of previous results of my research. They helped me a lot to improve my work and to find new paths to follow.

To all my colleagues of the TecMF lab at PUC-Rio. Thank you for watching my presentations so many time, to propose changes and improvements. Thank you all for the talks, and all the relaxing time together.

I thank all my family and friends that have always encouraged and supported my choices.

Finally, I also thank CNPq and CAPES for support this research.

## Abstract

Santos, Jefferson de Barros; Haeusler, Edward Hermann (advisor); Dowek, Gilles (co-advisor). **Systems for Provability and Countermodel Generation in Propositional Minimal Implicational Logic**. Rio de Janeiro, 2017. 82p. Tese de Doutorado — Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

This thesis presents a new sequent calculus called $\mathbf{LMT}^{\rightarrow}$ that has the properties to be terminating, sound and complete for Propositional Implicational Minimal Logic ($\mathbf{M}^{\rightarrow}$). $\mathbf{LMT}^{\rightarrow}$ is aimed to be used for proof search in $\mathbf{M}^{\rightarrow}$, in a bottom-up approach. *Termination* of the calculus is guaranteed by a strategy of rule application that forces an ordered way to search for proofs such that all possible combinations are stressed. For an initial formula $\alpha$, proofs in $\mathbf{LMT}^{\rightarrow}$ has an upper bound of $|\alpha| \cdot 2^{|\alpha|+1+2\cdot log_2|\alpha|}$, which together with the system strategy ensure decidability. System rules are conceived to deal with the necessity of hypothesis repetition and the context-splitting nature of $\rightarrow$-left, avoiding the occurrence of loops and the usage of backtracking. Therefore, $\mathbf{LMT}^{\rightarrow}$ steers the proof search always in a forward, *deterministic* manner. $\mathbf{LMT}^{\rightarrow}$ has the property to allow extractability of counter-models from failed proof searches (*bicompleteness*), i.e., the attempt proof tree of an expanded branch produces a Kripke model that falsifies the initial formula. Counter-model generation (using Kripke semantics) is achieved as a consequence of the completeness of the system. $\mathbf{LMT}^{\rightarrow}$ is implemented as an interactive theorem prover based on the calculus proposed here. We compare our calculus with other known deductive systems for $\mathbf{M}^{\rightarrow}$, especially with Fitting's Tableaux, a method that also has the bicompleteness property. We also proposed here a translation of $\mathbf{LMT}^{\rightarrow}$ to the Dedukti proof checker as a way to evaluate the correctness of the implementation regarding the system specification and to make our system easier to compare to others.

## Keywords

Logic; Propositional Minimal Implicational Logic; Sequent Calculus; Proof Search; Counter-model Generation;

## Resumo

Santos, Jefferson de Barros; Haeusler, Edward Hermann; Dowek, Gilles. **Sistemas de prova e geração de contra exemplo para Lógica Proposicional Minimal Implicacional**. Rio de Janeiro, 2017. 82p. Tese de Doutorado — Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Esta tese apresenta um novo cálculo de sequente, correto e completo para a Lógica Proposicional Minimal Implicacional ($\mathbf{M}^{\rightarrow}$). $\mathbf{LMT}^{\rightarrow}$ destina-se a ser usado para a busca de provas em $\mathbf{M}^{\rightarrow}$, em uma abordagem *bottom-up*. A *Terminação* do cálculo é garantida por uma estratégia de aplicação de regras que força uma maneira ordenada no procedimento de busca de provas de tal forma que todas as combinações possíveis são exploradas. Para uma fórmula inicial $\alpha$, as provas em $\mathbf{LMT}^{\rightarrow}$ têm um limite superior de $|\alpha| \cdot 2^{|\alpha|+1+2 \cdot log_2|\alpha|}$, que juntamente com a estratégia do sistema, garantem a decidibilidade do mesmo. As regras do sistema são concebidas para lidar com a necessidade de repetição de hipóteses e a natureza de perda de contexto da regra $\rightarrow$-esquerda , evitando a ocorrência de loops e o uso de *backtracking*. Portanto, a busca de prova em $\mathbf{LMT}^{\rightarrow}$ é *determinística*, sempre executando buscas no sentido *forward*. $\mathbf{LMT}^{\rightarrow}$ tem a propriedade de permitir a extração de contra-modelos a partir de buscas de prova que falharam (*bicompletude*), isto é, a árvore de tentativa de prova de um ramo totalmente expandido produz um modelo de Kripke que falsifica a fórmula inicial. A geração de contra-modelo (usando a semântica Kripke) é obtida como consequência da completude do sistema. $\mathbf{LMT}^{\rightarrow}$ é implementado como um provador de teoremas interativo baseado no cálculo proposto aqui. Comparamos nosso cálculo com outros sistemas dedutivos conhecidos para $\mathbf{M}^{\rightarrow}$, especialmente com Tableaux no estilo Fitting, um método que também tem a propriedade de ser bicompleto. Também propomos aqui uma tradução de $\mathbf{LMT}^{\rightarrow}$ para o verificador de prova Dedukti como uma forma de avaliar a correção da implementação que desenvolvemos, no que diz respeito à especificação do sistema, além de torná-lo mais fácil de comparar com outros sistemas existentes.

## Palavras-chave

Lógica; Lógica Proposicional Minimal Implicacional; Cálculo de Sequentes; Busca de Provas; Geração de Contra-modelo;

# Contents

# List of Figures

*Either mathematics is too big for the human mind, or the human mind is more than a machine.*

**Kurt Gödel**, *as quoted in Topoi: The Categorial Analysis of Logic (1979) by Robert Goldblatt, p. 13.*

# 1 Introduction

## 1.1 Motivation

Propositional Minimal Implicational Logic ($\mathbf{M}^{\rightarrow}$) is the fragment of the Propositional Minimal Logic containing only the logical connective $\rightarrow$.

The $TAUT$ problem for $\mathbf{M}^{\rightarrow}$, the general problem of deciding if a formula $\alpha \in \mathbf{M}^{\rightarrow}$ is always true, is a PSPACE-Complete problem as stated by Statman (1974), who also shows that this logic polynomially simulates Propositional Intuitionistic Logic. Statman's simulation can also be used to polynomially simulate Propositional Classical Logic. Furthermore, Haeusler (2015*b*) shows that $\mathbf{M}^{\rightarrow}$ can polynomially simulate not only Propositional Classical and Intuitionistic Logic but also the full Propositional Minimal Logic and any other decidable propositional logic with a Natural Deduction system where the Subformula Principle holds (see Prawitz, 2006).

Moreover, $\mathbf{M}^{\rightarrow}$ has a strong relation with open questions about the Computational Complexity Hierarchy as we can see from the statements below.

- If $CoNP \neq NP$ then $NP \neq P$.
- If $PSPACE = NP$ then $CoNP = NP$.
- $CoNP = NP$, iff $\forall\alpha \in TAUT_{Cla}$, $\exists DS$, a deductive system and $\Pi$, a proof of $\alpha$ in $DS$, $size(\Pi_{DS}) \leq Poly(|\alpha|)$.
- $PSPACE = NP$ iff $\forall\alpha \in TAUT_{\mathbf{M}^{\rightarrow}}$, $\exists DS$, a deductive system and $\Pi$, a proof of $\alpha$ in $DS$, $size(\Pi_{DS}) \leq Poly(|\alpha|)$.

Those characteristics show us that $\mathbf{M}^{\rightarrow}$ is as hard to implement as the most popular propositional logics. This fact, together with its very simple language (only one logical connective), makes $\mathbf{M}^{\rightarrow}$ an important research object that can provide us with many insights about the aforementioned relations of complexity classes and about the complexity of many other logics. Moreover, the problem of conducting a proof search in a deductive system for $\mathbf{M}^{\rightarrow}$has the complexity of $TAUT$ as a lower bound. Thus, the size of propositional proofs may be huge and *automated theorem provers* (ATP) should take care of super-polynomially sized proofs. Therefore, the study of deductive systems for $\mathbf{M}^{\rightarrow}$can directly influence in techniques to improve the way provers manage such proofs.

Our main contribution here is to present a sound and complete *sequent calculus* for $\mathbf{M}^{\rightarrow}$ called $\mathbf{LMT}^{\rightarrow}$, which allows the definition of a unified procedure for provability and counter-model generation in this logic. $\mathbf{LMT}^{\rightarrow}$ aims to be used for proof search in $\mathbf{M}^{\rightarrow}$, in a bottom-up approach. The system is based on a set of rules and in a general strategy for application of the rules in such a way that we can avoid the usage of *loop checkers* and mechanisms for *backtracking*. $\mathbf{LMT}^{\rightarrow}$ also avoids the necessity of working with different systems for provability and refutation, a very common approach to deal with this problem described in the literature. Counter-model generation (using Kripke semantics) is achieved as a consequence of the features of the system and the way the attempt proof tree (produced by a failed proof search) is constructed during a proof search process. We also present here a comparison of our calculus with other known deductive systems for $\mathbf{M}^{\rightarrow}$, especially with Fitting's Tableaux, a method that also has the bicompleteness property.

We implemented $\mathbf{LMT}^{\rightarrow}$ as an interactive theorem prover in Lua. Its source code can be found at `https://github.com/jeffsantos/GraphProver`. We also proposed here a translation of $\mathbf{LMT}^{\rightarrow}$ to the Dedukti proof checker as a way to evaluate the correctness of the implementation regards the system specification and to make our system easier to compare to others.

## 1.2 How this thesis is organized

Chapter 2 presents some initial definitions, establish the vocabulary and present the main problems in the field of proof search for $\mathbf{M}^{\rightarrow}$. Chapter 3 shows a study about the size of proofs in $\mathbf{M}^{\rightarrow}$ to establish a bound for proof search that can be used as a limit in the termination procedure of $\mathbf{LMT}^{\rightarrow}$. Chapter 4 presents $\mathbf{LMT}^{\rightarrow}$ itself and its main features: termination, soundness, and completeness (with the counter-model generation as a corollary). Chapter 5 compares our approach with the Tableaux System. Chapter 6 presents a translation of proofs generated in $\mathbf{LMT}^{\rightarrow}$ to the Dedukti proof checker. Chapter 7 concludes this thesis and describes some possible future work.

# 2
# Propositional Minimal Implicational Logic ($\mathrm{M}^{\rightarrow}$)

## 2.1
## Propositional Logics

Propositional logic is a *language* used to express propositions, i.e., declarative sentences (assertions) that can be evaluated as holding or not in some context. To refer to a propositional language, we use the symbol $\mathcal{L}$ and assertions are represented as formulas that belong to $\mathcal{L}$. We can formally define those concepts as follows.

**Definition 1** *The* alphabet *of a propositional logic $\mathcal{L}$ consists of:*

- *An enumerable set of propositional symbols, called atoms.*
- *The bottom symbol ($\bot$).*
- *The usual connectives (or logical operators) for negation ($\neg$), conjunction ($\wedge$), disjunction ($\vee$) and implication ($\rightarrow$). The first one being an unary operator, all the others being binary.*
- *parentheses: "(" and ")".*

**Definition 2** *We can define the general notion of a* formula *in $\mathcal{L}$ inductively:*

- *Every propositional symbol is a formula in $\mathcal{L}$. We call them atomic formulas.*
- *$\bot$ is an atomic formula.*
- *If $A$ is a formula in $\mathcal{L}$ then $\neg A$ is also a formula.*
- *If $A$ and $B$ are formulas in $\mathcal{L}$ then $(A \wedge B)$, $(A \vee B)$ and $(A \rightarrow B)$ are also.*

As usual, parentheses are used for disambiguation. We use the following conventions for the text:

– We use upper case letters to represent atomic formulas: $A$, $B$, $C$, ...

– We use lower Greek letters to represent generic formulas: $\alpha$, $\beta$, ...

– Upper case Greek letters are used to represent sets of formulas. For example $\Delta$, $\Gamma$, ...

– If parentheses are omitted, we interpret conjunctions and disjunctions nested left. Implications are nested right.

## 2.2 Logical Consequence

To be able to reason about the truth of a sentence written in a logical language, we need some mechanism to assign meaning to these sentences. When we write formulas in a logical language, the allowed logical symbols (the bottom symbol, logical connectives, and parentheses) follow their intrinsic meaning, but a form must be defined to interpret nonlogical symbols (atoms). We call *interpretation* the attribution of meaning to the nonlogical symbols of a logical language. In propositional logic, this is done by establishing when propositional letters in a formula hold or not. Then, the meaning of these atomic parts combined determines the value of the full formula that they are a part.

We say a formula is *valid* when it is true in all interpretations. For propositional logic a valid formula is called *tautology.* A formula is *valid in an interpretation* if it is always true in that interpretation. A formula $\alpha$ is said to be *satisfiable* if there is any interpretation where $\alpha$ is true. A formula $\alpha$ is said to be *falsifiable* if there is any interpretation where $\alpha$ is false. Finally, a formula $\alpha$ is said to be *unsatisfiable* if $\alpha$ is always false for any given interpretation. If $\alpha$ is unsatisfiable, $\neg\alpha$ is valid and vice versa.

The concept of *logical consequence* or *entailment* establishes a relation of consequence between a set of formulas and a formula. That is, the logical consequence determines when a formula logically follows from a set of formulas. Formally, being $\Delta$ a set of formulas and $\alpha$ a formula, we have:

$$\Delta \models \alpha$$

if and only if every interpretation satisfying all formulas of $\Delta$ also satisfies $\alpha$. A valid formula $\alpha$ is represented as $\models \alpha$.

We can have different ways to establish interpretations (and, thus, the logical consequence notion) for a propositional logic. Classical, Intuitionistic and Minimal Propositional Logics are very known propositional languages with different notions of interpretations.

## 2.3 Deductive Systems

A *deductive system* or *deductive calculus* consists of a set (possibly empty) of axioms and a set (not empty) of inference rules used as a syntactic mechanism aimed to prove *theorems* or to conclude, by a sequence of rule applications, that an assertion can be derived from a set of initial assertions (*set of hypotheses*).

Using $DS$ to represent a deductive system, to indicate that a formula $\alpha$ follows in $DS$ from a set of hypotheses ($\Delta$) we use:

$$\Delta \vdash \alpha$$

Therefore, a deductive system can be seen as a binary relation where the first component is a set of formulas and the second one is a formula. The sequence of rule applications from the formulas in the set of hypotheses $\Delta$ to the formula $\alpha$ is a *demonstration* that $\alpha$ follows from $\Delta$. If the initial set of hypotheses is empty, we call the sequence of rule applications that ends in $\alpha$ a *proof*, and $\alpha$ is a *theorem*.

A propositional logic can have many different deductive systems. In this thesis we are particularly interested in the systems of *Natural Deduction*, following the works of Jaśkowski (1934), Gentzen (1935) and Prawitz (2006), *Sequent Calculus*, originally conceived by Gentzen (1935), but considering adaptations proposed in Takeuti (2013) and the *Tableaux System*, as presented by Fitting (1969). For Natural Deduction and Sequent Calculus, we still are very influenced by notations and approaches presented in Seldin (1998).

We will study some relations between those systems when used to construct proofs in the language of the Propositional Minimal Implicational Logic.

## 2.4 Classical, Intuitionistic and Minimal Logics

One way to present the difference between Classical, Intuitionistic and Minimal Logics is comparing their respective systems of Natural Deduction. Here, we are restricted to propositional versions of those systems. In Natural Deduction, rules capture the meaning of the

*Propositional Minimal Logic* (**Min**) obtained from a Natural Deduction system with no axiom and a set of inference rules consisting of an introduction rule and an elimination rule for each logical connective of **Min**. Figure 2.1 shows the rules of such a system.

$$\frac{A \land B}{A}\,\land_1\text{-e} \qquad \frac{A \land B}{B}\,\land_2\text{-e} \qquad\qquad \frac{A \quad B}{A \land B}\,\land\text{-i}$$

$$\frac{A \lor B \quad \begin{matrix}[A]\\ \vdots \\ C\end{matrix} \quad \begin{matrix}[B]\\ \vdots \\ C\end{matrix}}{C}\,\lor\text{-e} \qquad\qquad \frac{A}{A \lor B}\,\lor_1\text{-i} \qquad \frac{B}{A \lor B}\,\lor_2\text{-i}$$

$$\frac{A \quad \neg A}{\bot}\,\neg\text{-e} \qquad\qquad \frac{\begin{matrix}[A]\\ \vdots \\ \bot\end{matrix}}{\neg A}\,\neg\text{-i}$$

$$\frac{A \quad A \to B}{B}\,\to\text{-e} \qquad\qquad \frac{\begin{matrix}[A]\\ \vdots \\ B\end{matrix}}{A \to B}\,\to\text{-i}$$

Figure 2.1: Natural Deduction System for **Min**

We obtain the *Propositional Intuitionistic Logic* (**Int**) when we add to those set of rules a new one, specifically aimed to deal with the intuitionistic meaning of $\bot$.

$$\textbf{Min} \quad + \quad \boxed{\frac{\bot}{A}\,\bot\text{-Int}}$$

*Propositional Classical Logic* (**Cla**) is the result of either to add the rule $\bot$-Cla to the set of rules of **Min** or to the set of rules of **Int**.

$$\textbf{Min} \quad + \quad \boxed{\begin{matrix}[\neg A]\\ \vdots \\ \dfrac{\bot}{A}\,\bot\text{-Cla}\end{matrix}}$$

or ....

$$\textbf{Int} \quad + \quad \boxed{\begin{matrix}[\neg A]\\ \vdots \\ \dfrac{\bot}{A}\,\bot\text{-Cla}\end{matrix}}$$

Seldin (1989) discusses the different logics that emerge from changes in the set of allowed rules for a Natural Deduction System.

Prawitz (2006) presents a comprehensible description of Natural Deduction Systems. His main result, the Normalization Theorem, states that arbitrary proofs in those logics can be transformed into an equivalent one where detours (applications of an introduction rule to the main formula followed by an elimination rule of the same main formula) are not present. These normal form proofs have a particularly interesting feature, known as the *Subformula Principle* which states that every formula occurrence in a normal deduction proof of a formula $\alpha$ from a set of hypotheses $\Gamma$ is a subformula of $\alpha$ or some formula of $\Gamma$. Subformula Principle together with the decidability of the logic allows the definition of procedures for proof search in the logic. In other words, procedures that starting from the set $\Gamma$ and using the rules of the calculus can produce the formula $\alpha$ (*top-down search*) or that from the target formula $\alpha$ using the rules of the system in a reverse sense (*bottom-up search*) ends in the set of hypotheses $\Gamma$.

Prawitz's normal proofs are equivalent to Gentzen's cut-free proofs for Sequent Calculus. These systems were shown equivalents by Gentzen (1935). We will talk in more details about Sequent Calculus in the next sections.

As previously stated, Propositional Minimal Implicational Logic ($\mathbf{M}^{\rightarrow}$) is the fragment of the Propositional Minimal Logic containing only the logical connective $\rightarrow$. The Natural Deduction system restricted to the introduction and elimination rules for the $\rightarrow$ connective (2-1) is correct and complete for this logic.

$$\frac{A \quad A \rightarrow B}{B} \rightarrow\text{-e} \qquad \frac{\begin{array}{c}[A]\\ \vdots \\ B\end{array}}{A \rightarrow B} \rightarrow\text{-i} \tag{2-1}$$

As stated in Chapter 1, this logic is the object of study of this research. We propose here a sequent calculus for bottom-up proof search in $\mathbf{M}^{\rightarrow}$. Before we start to present our system, we show in the next sections more details about $\mathbf{M}^{\rightarrow}$ and proof search procedures in its context.

## 2.5 Semantics for $\mathrm{M}^{\rightarrow}$

The semantics of $\mathbf{M}^{\rightarrow}$ is the intuitionistic semantics restricted to $\rightarrow$ only. Thus, given a propositional language $\mathcal{L}$, a $\mathbf{M}^{\rightarrow}$ model is a structure $\langle U, \preceq, \mathcal{V}\rangle$, where $U$ is a non-empty set (worlds), $\preceq$ is a partial order relation on $U$ and $\mathcal{V}$ is a function from $U$ into the power set of $\mathcal{L}$, such that if $i, j \in U$ and $i \preceq j$ then $\mathcal{V}(i) \subseteq \mathcal{V}(j)$. Given a model, the satisfaction relationship $\models$ between worlds in models and formulas is defined as in Intuitionistic Logic, namely:

- $\langle U, \preceq, \mathcal{V}\rangle \models_i p$, $p \in \mathcal{L}$, iff, $p \in \mathcal{V}(i)$
- $\langle U, \preceq, \mathcal{V}\rangle \models_i \alpha_1 \rightarrow \alpha_2$, iff, for every $j \in U$, such that $i \preceq j$, if $\langle U, \preceq, \mathcal{V}\rangle \models_j \alpha_1$ then $\langle U, \preceq, \mathcal{V}\rangle \models_j \alpha_2$.

As usual a formula $\alpha$ is valid in a model $\mathcal{M}$, namely $\mathcal{M} \models \alpha$, if and only if, it is satisfiable in every world $i$ of the model, namely $\forall i \in U, \mathcal{M} \models_i \alpha$. A formula is a $\mathbf{M}^{\rightarrow}$ tautology, if and only if, it is valid in every model.

## 2.6 Proof Search and Counter-Model Generation in $\mathrm{M}^{\rightarrow}$

It is known that Prawitz's Natural Deduction System for Propositional Minimal Logic with only the $\rightarrow$-rules ($\rightarrow$-Elim and $\rightarrow$-Intro) is sound and complete for the $\mathbf{M}^{\rightarrow}$ regarding Kripke semantics. As a consequence of this, Gentzen's **LJ** system (Gentzen, 1935) containing only right and left $\rightarrow$-rules is also sound and complete.

Gentzen (1935) also proved the decision problem for **Int** (which includes the case of **Min** and $\mathbf{M}^{\rightarrow}$). However, Gentzen's approach was not conceived to be a bottom-up proof search procedure. Figure 2.2 shows structural and logic rules of an adapted Gentzen's sequent calculus for $\mathbf{M}^{\rightarrow}$, called $\mathbf{LJ}^{\rightarrow}$. We restrict the right side of a sequent to one and only one formula (we are in $\mathbf{M}^{\rightarrow}$ thus sequents with empty right side does not make sense). This restriction implies that structural rules can only be considered for main formulas on the left side of a sequent. $\mathbf{LJ}^{\rightarrow}$ also incorporates contraction in the $\rightarrow$-left, with the repetition of the main formula of the conclusion on the premises. We use those adaptations to facilitate in explaining the difficulties in using sequent calculus systems for proof search in **Int**, **Min** and $\mathbf{M}^{\rightarrow}$. Dyckhoff (2016) describes in details the evolution of those adaptations over Gentzen's original **LJ**system in attempts to establishes better bottom-up proof search mechanisms for **Int**.

A central aspect when considering mechanisms for proof search in $\mathbf{M}^{\rightarrow}$ (and also for **Int**) is the application of the $\rightarrow$-left rule. The **LK** system proposed by Gentzen (1935), the sequent calculus for Classical Logic, with some adaptations (e.g. Seldin (1998)) can ensure that each rule, when applied in a bottom-up manner in the proof search, reduces the degree (the number of atomic symbols occurrences and connectives in a formula) of the main formula of the sequent (the formula to which the rule is applied) implying the termination of the system. However, the case for **Int** is more complicated. First, we have the "context-splitting" (using an expression from Dyckhoff (2016)) nature of $\rightarrow$-left, i.e., the formula on the right side of the conclusion sequent is lost in the left premise of the rule application. Second, as we can reuse a hypothesis in different parts of a proof, the main formula of the conclusion must be available to be used again by the generated premises. Thus, the $\rightarrow$-left rule has the repetition of the main formula in the premises, a scenario that allows the occurrence of loops in automatic procedures.

$$\frac{}{\Delta, \gamma \Rightarrow \gamma}\ \text{axiom}$$

$$\frac{\Delta \Rightarrow \gamma}{\alpha, \Delta \Rightarrow \gamma}\ \text{weakening (w)} \qquad \frac{\alpha, \alpha, \Delta \Rightarrow \gamma}{\alpha, \Delta \Rightarrow \gamma}\ \text{contraction (c)}$$

$$\frac{\Gamma, \alpha, \beta, \Delta \Rightarrow \gamma}{\Gamma, \beta, \alpha, \Delta \Rightarrow \gamma}\ \text{exchange (e)} \qquad \frac{\Delta \Rightarrow \alpha \qquad \alpha, \Gamma \Rightarrow \gamma}{\Delta, \Gamma \Rightarrow \gamma}\ \text{cut}$$

$$\frac{\Delta, \alpha \Rightarrow \beta}{\Delta \Rightarrow \alpha \rightarrow \beta}\ \rightarrow\text{-right } (\rightarrow\text{-r})$$

$$\frac{\Delta, \alpha \rightarrow \beta \Rightarrow \alpha \qquad \Delta, \alpha \rightarrow \beta, \beta \Rightarrow \gamma}{\Delta, \alpha \rightarrow \beta \Rightarrow \gamma}\ \rightarrow\text{-left } (\rightarrow\text{-l})$$

Figure 2.2: Rules of Gentzen's **LJ**

Since Gentzen, many others have explored solutions to deal with the challenges aforementioned proposing new calculi (sets of rules), strategies and proof search procedures to allow more automated treatment to the problem.

Unfortunately, the majority of these results are focused in **Int**, with very few work specifically dedicated to **Min** or $\mathbf{M}^{\rightarrow}$. Thus, we needed to concentrate our literature review in the **Int** case, adjusting the found results to the $\mathbf{M}^{\rightarrow}$ context by ourselves.

A crucial source of information was the work of Dyckhoff (2016), properly entitled *"Intuitionistic decision procedures since Gentzen"* that summarizes in chronological order the main results of this field. By means of personal communication, we had access to a preprint version of the document, which helps us a lot in conducting the research. In the next paragraphs, we highlight the most important of those results presented in Dyckhoff (2016).

A common way to control the proof search procedure in $\mathbf{M}^{\rightarrow}$ (and in **Int**) is by the definition of routines for loop verification as proposed in Underwood (1990). Loop checkers are very expensive procedures, although they are effective to guarantee termination in automatic provers for $\mathbf{M}^{\rightarrow}$ (and other logics with the same characteristic). The work in Heuerding et al. (1996)

and Howe (1997) are examples of techniques that can be used to minimize the performance problems that can arise with the usage of such procedures.

To avoid the use of loop checkers, Dyckhoff (1992) proposed a terminating contraction-free sequent calculus for **Int**, named **LJT**, using a technique based on the work of Vorob'ev (1970) in the 50s. Pinto & Dyckhoff (1995) extended this work showing a method to generate counter-examples in this system. They proposed two calculi, one for proof search and another for counter-model generation, forming a way to decide about the validity or not of formulas in **Int**. A characteristic of their systems is that the subformula property does not hold on them. In Ferrari et al. (2013), a similar approach is presented using systems where the subformula property holds. They also proposed a single decision procedure for **Int** which guarantee minimal depth counter-model.

Focused sequent calculi appeared initially in the Andreoli's work on linear logic (Andreoli, 1992). The author identified a subset of proofs from Gentzen-style sequent calculus, which are complete and tractable. Liang & Miller (2007) proposed the focused sequent calculi **LJF** where they used a mapping of **Int** into linear logic and adapted the Andreoli's system to work with the image. Dyckhoff & Lengrand (2006) presented the focused system **LJQ** that work direct in **Int**. Focusing is used in their system as a way to implement restrictions in the $\rightarrow$-left rule as proposed by Vorob'ev (1970) and Hudelmaier (1993). The work of Dyckhoff & Lengrand (2006) follows from the calculus with the same name presented in Herbelin (1995).

Dyckhoff (2016) also identify a list of features particularly of interest when evaluating mechanisms for proof search in **Int** that we will follow when comparing our solution to the other existent ones. They are: **termination** (proof search procedure stops both for theorems and non-theorem formulas), **bicompleteness** (extractability of models from failed proof searches), **avoidance of backtracking** (which is a very immediate approach to deal with the context split in $\rightarrow$-left, but it is also a complex and expensive procedure to implement), **simplicity** (allows easier reasoning about systems).

# 3 The Size of Proofs in $\mathrm{M}^{\to}$

## 3.1 Measuring the Size of Proofs

Hirokawa has presented an upper bound for the size of normal form Natural Deduction proofs of implicational formulas in **Int** (that correspond to $\mathbf{M}^{\to}$ formulas). Hirokawa (1991) showed that, for a formula $\alpha \in \mathbf{M}^{\to}$, this limit is $|\alpha| \cdot 2^{|\alpha|+1}$. As the Hirokawa result concerns about normal proofs in Natural Deduction we present now a translation of this system to a cut-free sequent calculus, following the $\mathbf{LJ}^{\to}$ rules presented in Section 2.6, thus we can establish the limit for proof search in $\mathbf{LJ}^{\to}$ too.

## 3.2 Translating Natural Deduction into Sequent Calculus

Figure 3.1 presents a recursively defined function[1] to translate Natural Deduction normal proofs of $\mathbf{M}^{\to}$ formulas into $\mathbf{LJ}^{\to}$ proofs (in a version of the system without the cut rule).

In this definition, $c$ is a function that returns the conclusion (last sequent) of a $\mathbf{LJ}^{\to}$ demonstration as showed in (3-1). Also, $\to -lc$ is a function that receives two $\mathbf{LJ}^{\to}$ sequents and a formula to construct the conclusion of a $\to -left$ rule application, as defined in (3-2).

$$c\left(\begin{array}{c}\Pi\\ \Gamma \Rightarrow \gamma\end{array}\right) = \Gamma \Rightarrow \gamma \tag{3-1}$$

$$\to -lc(\Gamma \Rightarrow \alpha; \beta, \Gamma \Rightarrow \gamma; \alpha \to \beta) = \Gamma, \alpha \to \beta \Rightarrow \gamma \tag{3-2}$$

[1]We use a semicolon to separate arguments of functions (in function definitions and function calls) instead of the most common approach to using commas. This change in convention aims to avoid confusion with the commas used to separate formulas and sets of formulas in sequent notation.

**Axioms:**

$$F(\alpha;\Gamma) = \Gamma, \alpha \Rightarrow \alpha$$

**Case of →Introduction:**

$$F\left(\begin{array}{c}[\alpha]^1\\ \prod\\ \hline \beta \\ \hline \alpha\to\beta\end{array}\to\text{-I}^1 \quad ;\Gamma\right) = \dfrac{F\left(\begin{array}{c}\alpha\\ \prod \\ \hline \beta\end{array};\{\alpha\}\cup\Gamma\right)}{\Gamma\Rightarrow\alpha\to\beta}\to\text{r}$$

**Case of →Elimination:**

$$F\left(\begin{array}{c}\begin{array}{cc}\prod_1 & \\ \alpha & \alpha\to\beta\end{array}\\ \hline \beta\\ \prod_2\\ C\end{array};\Gamma\right)=$$

$$\dfrac{F\left(\begin{array}{c}\prod_1\\ \alpha\end{array};\{\alpha\to\beta\}\cup\Gamma\right) \qquad\qquad F\left(\begin{array}{c}\beta\\ \prod_2\\ C\end{array};\{\alpha\to\beta\}\cup\Gamma\right)}{\to -lc\left(c\left(F\left(\begin{array}{c}\prod_1\\ \alpha\end{array};\{\alpha\to\beta\}\cup\Gamma\right)\right);c\left(F\left(\begin{array}{c}\beta\\ \prod_2\\ C\end{array};\{\alpha\to\beta\}\cup\Gamma\right)\right);\alpha\to\beta\right)}\to\text{l}$$

Figure 3.1: A recursively defined function to translate Natural Deduction proofs into $\mathbf{LJ}^{\to}$

As an example of the translation produced by the function of Figure 3.1, we show below each step of the translation of a Natural Deduction proof (3-3) into an $\mathbf{LJ}^{\to}$ proof (3-4). To shorten the size of the proofs we collapsed repeated occurrences of formulas when passed as the hypothesis argument of the recursive function call.

$$\dfrac{\dfrac{\dfrac{\dfrac{[B]^2 \qquad \dfrac{[A]^1 \qquad [A\to(B\to C)]^3}{B\to C}\to\text{-E}}{C}\to\text{-E}}{A\to C}\to\text{-I}^1}{B\to(A\to C)}\to\text{-I}^2}{(A\to(B\to C))\to(B\to(A\to C))}\to\text{-I}^3 \tag{3-3}$$

$$\nabla$$

$$F\left(\dfrac{\begin{array}{c}[A\to(B\to C)]^3\\ \prod_1\\ B\to(A\to C)\end{array}}{(A\to(B\to C))\to(B\to(A\to C))}\to\text{-I}^3\;;\emptyset\right)$$

$$\nabla$$

$$\dfrac{F\left(\begin{array}{c}[B]^2, A\to(B\to C)\\ \prod_2\\ \dfrac{A\to C}{B\to(A\to C)}\to\text{-I}^2\end{array}\;;\{A\to(B\to C)\}\right)}{\Rightarrow(A\to(B\to C))\to(B\to(A\to C))}\to\text{r}$$

$$\nabla$$

$$\dfrac{\dfrac{F\left(\begin{array}{c}[A]^1,B,A\to(B\to C)\\ \prod_3\\ \dfrac{C}{A\to C}\to\text{-I}^1\end{array}\;;\{B,A\to(B\to C)\}\right)}{A\to(B\to C)\Rightarrow B\to(A\to C)}\to\text{r}}{\Rightarrow(A\to(B\to C))\to(B\to(A\to C))}\to\text{r}$$

$$\nabla$$

$$\dfrac{\dfrac{\dfrac{F\left(\dfrac{B\qquad\dfrac{A\qquad A\to(B\to C)}{B\to C}\to\text{-E}}{C}\to\text{-E}\;;\{A,B,A\to(B\to C)\}\right)}{B,A\to(B\to C)\Rightarrow A\to C}\to\text{r}}{A\to(B\to C)\Rightarrow B\to(A\to C)}\to\text{r}}{\Rightarrow(A\to(B\to C))\to(B\to(A\to C))}\to\text{r}$$

In the following steps consider that $\Gamma=\{A,B,A\to(B\to C)\}$.

$$\nabla$$

$$\dfrac{\dfrac{\dfrac{\dfrac{F(A;\Gamma)\qquad\qquad F\left(\dfrac{B\qquad B\to C}{C}\to\text{-E};\Gamma\right)}{\to -lc\left(c\left(F(A;\Gamma)\right);c\left(F\left(\dfrac{B\qquad B\to C}{C}\to\text{-E};\Gamma\right)\right);A\to(B\to C)\right)}\to\text{l}}{B,A\to(B\to C)\Rightarrow A\to C}\to\text{r}}{A\to(B\to C)\Rightarrow B\to(A\to C)}\to\text{r}}{\Rightarrow(A\to(B\to C))\to(B\to(A\to C))}\to\text{r}$$

$$\nabla$$

$$
\begin{prooftree}
\AxiomC{$\Gamma \Rightarrow A$}
\AxiomC{$F(B;\{\Gamma, B \to C\})$}
\AxiomC{$F(C;\{\Gamma, B \to C\})$}
\RightLabel{$\to$ l}
\BinaryInfC{$\to -lc(c(F(B;\{\Gamma, B \to C\}));c(F(C;\{\Gamma, B \to C\}));B \to C)$}
\RightLabel{$\to$ l}
\BinaryInfC{$\Gamma \Rightarrow C$}
\RightLabel{$\to$ r}
\UnaryInfC{$B, A \to (B \to C) \Rightarrow A \to C$}
\RightLabel{$\to$ r}
\UnaryInfC{$A \to (B \to C) \Rightarrow B \to (A \to C)$}
\RightLabel{$\to$ r}
\UnaryInfC{$\Rightarrow (A \to (B \to C)) \to (B \to (A \to C))$}
\end{prooftree}
$$

$$\nabla$$

$$
\begin{prooftree}
\AxiomC{$\Gamma \Rightarrow A$}
\AxiomC{$\Gamma, B \to C \Rightarrow B$}
\AxiomC{$\Gamma, B \to C, C \Rightarrow C$}
\RightLabel{$\to$ l}
\BinaryInfC{$\Gamma, B \to C \Rightarrow C$}
\RightLabel{$\to$ l}
\BinaryInfC{$\Gamma \Rightarrow C$}
\RightLabel{$\to$ r}
\UnaryInfC{$B, A \to (B \to C) \Rightarrow A \to C$}
\RightLabel{$\to$ r}
\UnaryInfC{$A \to (B \to C) \Rightarrow B \to (A \to C)$}
\RightLabel{$\to$ r}
\UnaryInfC{$\Rightarrow (A \to (B \to C)) \to (B \to (A \to C))$}
\end{prooftree}
\tag{3-4}
$$

## 3.3 Dealing with Super-Polynomial Proofs

**Theorem 3** *The size of proofs in* $\mathbf{LJ}^{\to}$ *considering only implicational tautologies is the same of that in Natural Deduction, i.e. for an implicational formula* $\alpha$*, a proof in* $\mathbf{LJ}^{\to}$ *has maximum height of* $|\alpha| \cdot 2^{|\alpha|+1}$.

*Proof*: This proof follows directly from the translation function aforementioned as each step in the Natural Deduction proof is translated into exactly one step in the $\mathbf{LJ}^{\to}$ resultant proof. ■

# 4
# The Sequent Calculus $\text{LMT}^{\rightarrow}$

## 4.1
## Initial Definitions

In this chapter, we present a sound and complete *sequent calculus* for $\mathbf{M}^{\rightarrow}$. We call this system $\mathbf{LMT}^{\rightarrow}$. We can prove for each rule that if all premises are valid, then the conclusion is also valid and if at least one premise is invalid, then the conclusion also is. Besides that, this proof is constructive in the sense that for any sequent we have an effective way to produce either a proof or a counter-model of it.

We start defining the concept of *sequent* used in the proposed calculus. A sequent in our system has the following general form:

$$\{\Delta'\}, \Upsilon_1^{p_1}, \Upsilon_2^{p_2}, ..., \Upsilon_n^{p_n}, \Delta \Rightarrow [p_1, p_2, ..., p_n], \varphi \tag{4-1}$$

where $\varphi$ is a formula in $\mathcal{L}$ and $\Delta$, $\Upsilon_1^{p_1}, \Upsilon_2^{p_2}, ..., \Upsilon_n^{p_n}$ are bags[1] of formulas. Each $\Upsilon_i^{p_i}$ represents formulas associated with an atomic formula $p_i$.

A sequent has two *focus areas*, one in the left side (curly bracket)[2] and another on the right (square bracket). Curly brackets are used to control the application of the $\rightarrow$-left rule and square brackets are used to keep control of formulas that are related to a particular counter-model definition. $\Delta'$ is a set of formulas and $p_1, p_2, ..., p_n$ is a sequence that does not allow repetition. We call *context* of the sequent a pair $(\alpha, q)$, where $\alpha \in \Delta'$ and $\varphi = q$, where $q$ is an atomic formula on the right side of the sequent.

The axioms and rules of $\mathbf{LMT}^{\rightarrow}$ are presented in Figure 4.1. In each rule, $\Delta' \subseteq \Delta$.

Rules are inspired by their backward application. In a $\rightarrow$-left rule application, the atomic formula, $q$, on the right side of the conclusion goes to the []-area in the left premise. $\Delta$ formulas in the conclusion are copied to the left premise and marked with a label relating each of them with $q$. The left premise

[1]A bag (or a multiset) is a generalization of the concept of a set that, unlike a set, takes repetitions into account: a bag {A, A, B} is not the same as the bag {A, B}.

[2]Note that the symbols { and } here do not represent a set. They are used as an annotation in the sequent to determine the left side focused area. Therefore, $\Delta'$ instead is a set of formulas in the focused area.

also has a copy of $\Delta$ formulas without the $q$-label. This mechanism keeps track of proving attempts. The form of the restart rule is better understood in the completeness proof on Section 4.5. A forward reading of rules can be achieved considering the notion of validity, as described in Section 4.4.

**Axiom:**

$$\frac{}{\{\Delta', q\}, \Upsilon_1^{p_1}, \Upsilon_2^{p_2}, \ldots, \Upsilon_n^{p_n}, \Delta \Rightarrow [p_1, p_2, \ldots, p_n], q}\ \mathit{axiom}$$

**Focus:**

$$\frac{\{\Delta', \alpha\}, \Upsilon_1^{p_1}, \Upsilon_2^{p_2}, \ldots, \Upsilon_n^{p_n}, \Delta, \alpha \Rightarrow [p_1, p_2, \ldots, p_n], \beta}{\{\Delta'\}, \Upsilon_1^{p_1}, \Upsilon_2^{p_2}, \ldots, \Upsilon_n^{p_n}, \Delta, \alpha \Rightarrow [p_1, p_2, \ldots, p_n], \beta}\ \mathrm{f}_\alpha$$

**Restart:**

$$\frac{\{\}, \Upsilon_1, \Upsilon_2, \ldots, \Upsilon_i, \Upsilon_{i+1}^{p_{i+1}}, \ldots, \Upsilon_n^{p_n}, \Delta^q \Rightarrow [p_1, p_2, \ldots, p_{i+1}, \ldots, p_n, q], p_i}{\{\Delta'\}, \Upsilon_1^{p_1}, \Upsilon_2^{p_2}, \ldots, \Upsilon_i^{p_i}, \Upsilon_{i+1}^{p_{i+1}}, \ldots, \Upsilon_n^{p_n}, \Delta \Rightarrow [p_1, p_2, \ldots, p_i, p_{i+1}, \ldots, p_n], q}\ \mathrm{r}_{p_i}$$

**$\rightarrow$-Right**

$$\frac{\{\Delta'\}, \Upsilon_1^{p_1}, \Upsilon_2^{p_2}, \ldots, \Upsilon_n^{p_n}, \Delta, \alpha \Rightarrow [p_1, p_2, \ldots, p_n], \beta}{\{\Delta'\}, \Upsilon_1^{p_1}, \Upsilon_2^{p_2}, \ldots, \Upsilon_n^{p_n}, \Delta \Rightarrow [p_1, p_2, \ldots, p_n], \alpha \rightarrow \beta}\ \rightarrow\text{-r}_{\alpha\rightarrow\beta}$$

**$\rightarrow$-Left**

Considering $\overline{\Upsilon} = \bigcup_{i=1}^{n} \Upsilon_i^{p_i}$ and $\bar{p} = p_1, p_2, \ldots, p_n$, we have:

$$\frac{\{\alpha \rightarrow \beta, \Delta'\}, \overline{\Upsilon}, \Delta^q, \Delta \Rightarrow [\bar{p}, q], \alpha \qquad \{\alpha \rightarrow \beta, \Delta'\}, \overline{\Upsilon}, \Delta, \beta \Rightarrow [\bar{p}], q}{\{\alpha \rightarrow \beta, \Delta'\}\overline{\Upsilon}, \Delta \Rightarrow [\bar{p}], q}\ \rightarrow\text{-l}_{(\alpha\rightarrow\beta, q)}$$

Figure 4.1: Rules of $\mathbf{LMT}^{\rightarrow}$

## 4.2 A Proof Search Strategy

The following is a general strategy to be applied with the rules of $\mathbf{LMT}^{\rightarrow}$ to generate proofs from an input sequent (a sequent that is a candidate to be the conclusion of a proof), which is based on a bottom-up application of the rules. From the proposed strategy, we can then state a proposition about the termination of the proving process.

A *goal sequent* is a new sequent in the form of (4-1). It is a premise of one of the system's rules, generated by the application of this rule on an open branch during the proving process. If the goal sequent is an axiom, the branch where it is will stop. Otherwise, apply the first applicable rule in the following order:

1. Apply $\rightarrow$-right rule if it is possible, i.e., if the formula on the right side of the sequent, outside the []-area, is not atomic. The premise generated by this application is the new goal of this branch.

2. Choose one formula on the left side of the sequent, not labeled yet, i.e., a formula $\alpha \in \Delta$ that is not occurring in $\Delta'$, then apply the focus rule. The premise generated by this application is the new goal of this branch.

3. If all formulas on the left side have already been focused, choose the first formula $\alpha \in \Delta'$ such that the context $(\alpha, q)$ was not yet tried since the last application of a restart rule. We say that a context $(\alpha, q)$ is already tried when a formula $\alpha$ on the left was expanded (by the application of $\rightarrow$-left rule) with $q$ as the formula outside the []-area on the right side of the sequent. The premises generated by this application are new goals of the respective new branches.

4. Choose the leftmost formula inside the []-area that was not chosen before in this branch and apply the restart rule. The premise generated by this application is the new goal of the branch.

Figure 4.2 shows a general attempt proof tree generated by the application of the aforementioned strategy.

**Observation 4** *From the proof strategy we can make the following observations about a tree generated during a proving process:*

*(i)* *A* top sequent *is the highest sequent of a branch in the tree.*

*(ii)* *In a top sequent of a branch on the form of sequent* (4-1)*, if $\varphi \in \Delta$ then the top sequent is an axiom and the branch is called a* closed *branch. Otherwise, we say that the branch is* open *and $\varphi$ is an atomic formula.*

*(iii)* *In every sequent of the tree, $\Delta' \subseteq \Delta$.*

*(iv)* *For $i = 1, \ldots n$, $\Upsilon_{i-1}^{p_{i-1}} \subseteq \Upsilon_i^{p_i}$.*

$$\{\Delta'\}, \Upsilon_1^{p_1}, \Upsilon_2^{p_2}, \Upsilon_{i-1}^{p_{i-1}}, \ldots, \Upsilon_i^{p_i}, \Delta \Rightarrow [p_1, p_2, \ldots, p_{i-1}, p_i], p_k \quad (\textit{where } k = 1,\ 2,\ \ldots,\ i)$$

$$\vdots$$

$$\dfrac{\textit{a sequence of focus, } \to\textit{-left and } \to\textit{-right}}{\{\}, \Upsilon_1^{p_1}, \Upsilon_2^{p_2}, \ldots, \Upsilon_{i-1}^{p_{i-1}}, \Delta \Rightarrow [p_1, p_2, \ldots, p_{i-1}], p_i}$$

$$\vdots$$

$$\dfrac{\dfrac{\textit{a sequence of focus, } \to\textit{-left, } \to\textit{-right and restart (for each atomic formula in the []-area)}}{\{\varphi \to \psi\}, \Upsilon_2^{p_2}, \ldots, \Upsilon_{i-1}^{p_{i-1}}, \Delta^{p_i}, \Upsilon_1^{p_1}, \Upsilon_1, \varphi_1, \ldots, \varphi_n \Rightarrow [p_2, \ldots, p_{i-1}, p_i, p_1], p_2}}{\vdots}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\{\varphi \to \psi\}, \Upsilon_2^{p_2}, \ldots, \Upsilon_{i-1}^{p_{i-1}}, \Delta^{p_i}, \Upsilon_1^{p_1}, \Upsilon_1 \Rightarrow [p_2, \ldots, p_{i-1}, p_i, p_1], \varphi} \to\text{-right} \qquad \vdots}{\{\varphi \to \psi\}, \Upsilon_2^{p_2}, \ldots, \Upsilon_{i-1}^{p_{i-1}}, \Delta^{p_i}, \Upsilon_1 \Rightarrow [p_2, \ldots, p_{i-1}, p_i], p_1} \to\text{-left}}{\{\}, \Upsilon_2^{p_2}, \ldots, \Upsilon_{i-1}^{p_{i-1}}, \Delta^{p_i}, \Upsilon_1 \Rightarrow [p_2, \ldots, p_{i-1}, p_i], p_1} \text{focus}}{\{\Delta'\}, \Upsilon_1^{p_1}, \Upsilon_2^{p_2}, \ldots, \Upsilon_{i-1}^{p_{i-1}}, \Delta \Rightarrow [p_1, p_2, \ldots, p_{i-1}], p_i} \text{restart-}p_1}{}$$

$$\vdots$$

$$\dfrac{\textit{a sequence of focus, } \to\textit{-left and } \to\textit{-right}}{\{\varphi \to \psi\}, \Upsilon_1^{p_1}, \Upsilon_1, \varphi_1, \ldots, \varphi_n \Rightarrow [p_1], p_2}$$

$$\vdots$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\textit{a sequence } \to\textit{-right}}{\{\varphi \to \psi\}, \Upsilon_1^{p_1}, \Upsilon_1 \Rightarrow [p_1], \varphi} \qquad \dfrac{\dfrac{\vdots}{\{\varphi \to \psi, \psi\}, \psi, \Upsilon_1 \Rightarrow [], p_1}}{\{\varphi \to \psi\}, \psi, \Upsilon_1 \Rightarrow [], p_1} \text{focus}}{\{\varphi \to \psi\}, \Upsilon_1 \Rightarrow [], p_1} \to\text{-left}}{\{\}, \varphi \to \psi, \gamma_1, \ldots, \gamma_m \Rightarrow [], p_1} \text{focus}}{\vdots}$$

$$\dfrac{\{\}, \varphi \to \psi \Rightarrow [], \gamma}{\{\} \Rightarrow [], (\varphi \to \psi) \to \gamma} \to\text{-right}$$

Figure 4.2: General attempt proof tree

## 4.3 An Upper Bound for the Proof Search in $\mathbf{LMT}^{\to}$

Using the same approach applied in Chapter 3, we now propose a translation from $\mathbf{LJ}^{\to}$ proofs into the system $\mathbf{LMT}^{\to}$. The translation function needs to adapt a sequent in $\mathbf{LJ}^{\to}$ form to a sequent in $\mathbf{LMT}^{\to}$ form. Figure 4.3 presents the definition of the translation function[3].

**Axioms:**

$$F'(\Gamma,\alpha\Rightarrow\alpha;\Delta;\Upsilon;\Sigma;\Pi) = \begin{matrix}\{\Delta\},\Upsilon,\Gamma,\alpha\Rightarrow[\Sigma],\alpha\\ \Pi\end{matrix}$$

**Last rule is $\to$-right:**

$$F'\left(\dfrac{\begin{matrix}\mathcal{D}\\ \Gamma,\alpha\Rightarrow\beta\end{matrix}}{\Gamma\Rightarrow\alpha\to\beta}\,\to\text{-r};\Delta;\Upsilon;\Sigma;\Pi\right)=$$

$$\dfrac{F'\left(\begin{matrix}\mathcal{D}\\ \Gamma,\alpha\Rightarrow\beta\end{matrix};\Delta;\Upsilon;\Sigma;\begin{matrix}\{\Delta\},\Upsilon,\Gamma\Rightarrow[\Sigma],\alpha\to\beta\\ \Pi\end{matrix}\right)}{\begin{matrix}\{\Delta\},\Upsilon,\Gamma\Rightarrow[\Sigma],\alpha\to\beta\\ \Pi\end{matrix}}\to\text{r}$$

**Last rule is $\to$-left:**

$$F'\left(\dfrac{\begin{matrix}\mathcal{D}_1\\ \Gamma,\alpha\to\beta\Rightarrow\alpha\end{matrix}\qquad\begin{matrix}\mathcal{D}_2\\ \Gamma,\beta\Rightarrow q\end{matrix}}{\Gamma,\alpha\to\beta\Rightarrow q}\,\to\text{-l};\Delta;\Upsilon;\Sigma;\Pi\right)=$$

$$\dfrac{\mathcal{D}_1'\qquad\qquad\mathcal{D}_2'}{\mathit{PROOFUNTIL}\left(\mathit{FOCUS}\left(\begin{matrix}\{\Delta\},\Upsilon,\Gamma,\alpha\to\beta\Rightarrow[\Sigma],q\\ \Pi\end{matrix}\right)\right)}\to\text{l}$$

Figure 4.3: A recursive function to translate $\mathbf{LJ}^{\to}$ into $\mathbf{LMT}^{\to}$

We use some abbreviations to shorten the function definition of Figure 4.3. We present them below.

$$\mathcal{D}_1' = F'\left(\begin{matrix}\mathcal{D}_1\\ \Gamma,\alpha\to\beta\Rightarrow\alpha\end{matrix};\Delta\cup\{\alpha\to\beta\};\Upsilon\cup\Gamma^q\cup\{(\alpha\to\beta)^q\};\Sigma\cup\{q\};\Pi'\right)$$

[3]As in Chapter 3, we use semicolon to separate function arguments here

$$\mathcal{D}_2' = \quad F'\left(\begin{array}{c}\mathcal{D}_2\\ \Gamma, \beta \Rightarrow q\end{array}; \Delta \cup \{\alpha \rightarrow \beta\}; \Upsilon; \Sigma; \Pi'\right)$$

$$\Pi' = \quad PROOFUNTIL\left(FOCUS\left(\begin{array}{c}\{\Delta\}, \Upsilon, \Gamma, \alpha \rightarrow \beta \Rightarrow [\Sigma], q\\ \Pi\end{array}\right)\right)$$

$\Gamma^q =$ means that all formulas of the set $\Gamma$ are labeled with a reference to the atomic formula $q$

$(\alpha \rightarrow \beta)^q \quad =$ means the same for the individual formula $\alpha \rightarrow \beta$.

The complex case occurs when the function $F'$ is applied to a proof fragment in which the last $\mathbf{LJ}^{\rightarrow}$ rule applied is an $\rightarrow$-left. In this case, $F'$ needs to inspect the proof fragment constructed until that point to identify whether the context $(\alpha \rightarrow \beta, q)$ was already used or not. This inspection has to be done since $\mathbf{LMT}^{\rightarrow}$ does not allow two or more applications of the same context between two applications of the *restart* rule. To deal with this, we use some auxiliary functions described below.

*FOCUS* is a function that receives a fragment of proof in $\mathbf{LMT}^{\rightarrow}$ form and builds one application of the focus rule on the top of the proof fragment received in the case that the main formula of the rule is not already focused. The main formula is also an argument of the function. In the function definition (4-2), we have the constraint that $\alpha \in \Gamma$.

$$FOCUS\left(\begin{array}{c}\{\Delta\}, \Upsilon, \Gamma \Rightarrow [\Sigma], \beta\\ \prod\\ \{\} \Rightarrow [], \gamma\end{array}; \alpha\right) =$$

$$\left\{\begin{array}{ll}\begin{array}{c}\dfrac{\{\Delta, \alpha\}, \Upsilon, \Gamma \Rightarrow [\Sigma], \beta}{\{\Delta\}, \Upsilon, \Gamma \Rightarrow [\Sigma], \beta}\text{focus}\\ \prod\\ \{\} \Rightarrow [], \gamma\end{array} & \text{if } \alpha \notin \Delta\\ \\ \begin{array}{c}\{\Delta\}, \Upsilon, \Gamma \Rightarrow [\Sigma], \beta\\ \prod\\ \{\} \Rightarrow [], \gamma\end{array} & \text{otherwise}\end{array}\right\} \tag{4-2}$$

The function $PROOFUNTIL$ also receives a fragment of a proof in $\mathbf{LMT}^{\rightarrow}$ form where $(\alpha \rightarrow \beta, q)$ is one of the available contexts, applies the restart rule with an atomic formula $p$ such that $p \in \Sigma$ in the top of this fragment of proof and, then, conducts a sequence of $\mathbf{LMT}^{\rightarrow}$ rule applications following the strategy aforementioned until the point that the context $(\alpha \rightarrow \beta, q)$ is available again. This mechanism has to be done in the case that the context $(\alpha \rightarrow \beta, q)$ is already applied in an $\rightarrow$-left application, some point after the last restart rule application in the proof fragment received as the argument $\Pi$. Otherwise, the proof fragment is returned unaltered. Function $PROOFUNTIL$ is described in the function definition (4-3).

$$PROOFUNTIL\begin{pmatrix}\{\Delta, \alpha \rightarrow \beta\}, \Upsilon, \Gamma \Rightarrow [\Sigma], q\\ \prod \\ \{\} \Rightarrow [], \gamma\end{pmatrix} =$$

$$\left\{\begin{array}{ll}
\begin{array}{c}
\dfrac{\dfrac{\dfrac{\{\Delta'', \alpha \rightarrow \beta\}, \Upsilon'', \Gamma' \Rightarrow [\Sigma''], q}{\vdots}}{\{\}, \Upsilon', \Gamma^{q}, \Gamma \Rightarrow [\Sigma'], p}}{\{\Delta, \alpha \rightarrow \beta\}, \Upsilon, \Gamma \Rightarrow [\Sigma], q}\ \text{restart}{-p}\\
\prod\\
\{\} \Rightarrow [], \gamma
\end{array} & \rightarrow\text{-left}(\alpha \rightarrow \beta, q) \in \prod \\
 & \\
\begin{array}{c}
\{\Delta, \alpha \rightarrow \beta\}, \Upsilon, \Gamma \Rightarrow [\Sigma], q\\
\prod\\
\{\} \Rightarrow [], \gamma
\end{array} & \text{otherwise}
\end{array}\right\} \tag{4-3}$$

As an example of this translation, we use here the formula $((((A \rightarrow B) \rightarrow A) \rightarrow A) \rightarrow B) \rightarrow B$. We know from Dowek & Jiang (2006) that this formula needs two repetitions of the hypothesis $(((A \rightarrow B) \rightarrow A) \rightarrow A) \rightarrow B$ to be proved in $\mathbf{M}^{\rightarrow}$. To shorten the proof tree we use the following abbreviation: $((A \rightarrow B) \rightarrow A) \rightarrow A = \epsilon$. Thus, its normal proof in Natural Deduction can be represented as shown in Proof (4-4).

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[A]^1}{\epsilon}\to \mathrm{I}^4 \quad [\epsilon\to B]^3}{B}\to \mathrm{E}}{(A\to B)}\to \mathrm{I}^1 \quad [(A\to B)\to A]^2}{A}\to \mathrm{E}}{\epsilon}\to \mathrm{E}^2 \quad [\epsilon\to B]^3}{B}\to \mathrm{E}}{(\epsilon\to B)\to B}\to \mathrm{I}^3 \tag{4-4}$$

Using the translation presented in Figure 3.1, we achieve the following **LJ**$^{\to}$ cut-free proof. To shorten the proof we numbered each subformula of the initial formula that we want to prove and use these numbers to refer to those subformulas all over the proof. Let us call this **LJ**$^{\to}$ version of the proof $\mathcal{D}$ (Proof (4-5)).

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\to_3,\to_1,A,\to_1\Rightarrow A}{\to_3,\to_1,A\Rightarrow\to_2}\to\text{-R} \quad \to_3,\to_1,A,B\Rightarrow B}{\to_3,\to_1,A\Rightarrow B}\to\text{-L}}{\to_3,\to_1\Rightarrow\to_0}\to\text{-R} \quad \to_3,\to_1,A\Rightarrow A}{\to_3,\to_1\Rightarrow A}\to\text{-L}}{\to_3\Rightarrow\to_2}\to\text{-R} \quad \to_3,B\Rightarrow B}{\to_3\Rightarrow B}\to\text{-L}}{\Rightarrow ((((A\to_0 B)\to_1 A)\to_2 A)\to_3 B)\to_4 B}\to\text{-R} \tag{4-5}$$

The translation to **LMT**$^{\to}$ starts by applying the function $F'$ to the full proof $\mathcal{D}$.

$$F'\left(\mathcal{D};\begin{array}{rcl}\Gamma & = & \emptyset;\\ \Delta & = & \emptyset;\\ \Upsilon & = & \emptyset;\\ \Sigma & = & \emptyset;\\ \Pi & = & \mathit{nil}\end{array}\right)$$

This first call of $F'$ produces the end sequent of the proof in **LMT**$^{\to}$ and calls the function $F'$ recursively to the rest of the original proof in **LJ**$^{\to}$. This **LJ**$^{\to}$ fragment has now, as its last rule application, an $\to$-left.

$$\Pi \quad = \quad \{\}\Rightarrow [\,],((((A\to_0 B)\to_1 A)\to_2 A)\to_3 B)\to_4 B$$

$$\dfrac{F'\left(\dfrac{\begin{array}{c}\mathcal{D}_1\\ \to_3\Rightarrow\to_2\end{array} \quad \to_3,B\Rightarrow B}{\to_3\Rightarrow B}\to\text{-l}\;;\begin{array}{rcl}\Gamma & = & \emptyset;\\ \Delta & = & \emptyset;\\ \Upsilon & = & \emptyset;\\ \Sigma & = & \emptyset;\\ \Pi & & \end{array}\right)}{\Pi}\to \mathrm{r}$$

Then, as the main formula $(((A \to_0 B) \to_1 A) \to_2 A) \to_3 B$ (in the proof represented only by $\to_3$) is not focused yet, the call of function $F'$ first constructs an application of the focus rule on the top of the $\Pi$ fragment received as a function argument. Also, the context $(\to_3, B)$ was not expanded yet. Thus, the recursive step can proceed directly without the need of a restart (this is controlled by the $PROOFUNTIL$ function as shown in $F'$ definition in Figure 4.3).

$$\Pi \;=\; \cfrac{\cfrac{\{\to_3\}, \to_3 \Rightarrow [], B}{\{\}, \to_3 \Rightarrow [], B}\;\to \text{focus}}{\{\} \Rightarrow [], ((((A \to_0 B) \to_1 A) \to_2 A) \to_3 B) \to_4 B}\;\to \text{r}$$

$$\cfrac{F'\left(\begin{array}{c} \mathcal{D}_2 \\ \to_3 \Rightarrow \to_2 \end{array} \;;\; \begin{array}{rcl} \Gamma &=& \{\to_3\}; \\ \Delta &=& \{\to_3\}; \\ \Upsilon &=& \{\to_3^B\}; \\ \Sigma &=& \{B\}; \\ \Pi && \end{array}\right) \quad F'\left(\to_3, B \Rightarrow B \;;\; \begin{array}{rcl} \Gamma &=& \{\to_3\}; \\ \Delta &=& \{\to_3\}; \\ \Upsilon &=& \emptyset; \\ \Sigma &=& \emptyset; \\ \Pi && \end{array}\right)}{\Pi}\;\to \text{l}$$

The call of $F'$ on the right premise constructs an axiom. Thus this branch in the **LMT**$^{\to}$ proof translation being built is closed. The next recursive call "pastes" on the top of the right branch of the new version of $\Pi$ as follows.

$$\Pi \;=\; \cfrac{\cfrac{\cfrac{\{\to_3\}, \to_3^B, \to_3 \Rightarrow [B], \to_2 \qquad \{\to_3\}, \to_3, B \Rightarrow [], B}{\{\to_3\}, \to_3 \Rightarrow [], B}\;\to \text{l}}{\{\}, \to_3 \Rightarrow [], B}\;\to \text{focus}}{\{\} \Rightarrow [], ((((A \to_0 B) \to_1 A) \to_2 A) \to_3 B) \to_4 B}\;\to \text{r}$$

$$\cfrac{F'\left(\cfrac{\begin{array}{c} \mathcal{D}_3 \\ \to_3, \to_1 \Rightarrow \to_0 \end{array} \qquad \to_3, A \Rightarrow A}{\to_3, \to_1 \Rightarrow A}\;\to\text{-L} \;;\; \begin{array}{rcl} \Gamma &=& \{\to_3\}; \\ \Delta &=& \{\to_3\}; \\ \Upsilon &=& \{\to_3^B\}; \\ \Sigma &=& \{B\}; \\ \Pi && \end{array}\right)}{\Pi}\;\to \text{r}$$

This process goes until a point where the context $(\to_3, B)$ is again found, and the translation needs to deal with a restart in the **LMT**$^{\to}$ translated proof. This situation happens when the recursion of $F'$ reaches the point below.

$$F'\left(\begin{array}{cc} \dfrac{\overset{\mathcal{D}_4}{\to_3,\to_1,A\Rightarrow\to_2} \qquad \to_3,\to_1,A,B\Rightarrow B}{\to_3,\to_1,A\Rightarrow B}\to\text{-L} & ; \begin{array}{rcl} \Gamma & = & \{\to_1,A\}; \\ \Delta & = & \{\to_3,\to_1\}; \\ \Upsilon & = & \{\to_3^B,\to_3^A,\to_1^A\}; \\ \Sigma & = & \{B,A\}; \\ \Pi & & \end{array} \end{array}\right)$$

The $\Pi$ fragment constructed in this step of the recursion is presented below.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\overset{\Pi'}{\{\to_3,\to_1\},\to_3^B,\to_3^A,\to_1^A,\to_3,\to_1\Rightarrow[B,A],\to_0} \quad \{\to_3,\to_1\},\to_3^B,\to_3,\to_1,A\Rightarrow[B],A}{\{\to_3,\to_1\},\to_3^B,\to_3,\to_1\Rightarrow[B],A}\to\text{l}}{\{\to_3\},\to_3^B,\to_3,\to_1\Rightarrow[B],A}\to\text{focus}}{\{\to_3\},\to_3^B,\to_3\Rightarrow[B],\to_2}\to\text{r} \qquad \{\to_3\},\to_3,B\Rightarrow[],B}{\{\to_3\},\to_3\Rightarrow[],B}\to\text{l}}{\{\},\to_3\Rightarrow[],B}\to\text{focus}}{\{\}\Rightarrow[],((((A\to_0 B)\to_1 A)\to_2 A)\to_3 B)\to_4 B}\to\text{r}$$

The $\Pi'$ fragment in $\Pi$ is built as the result of a proof search from the top sequent of the leftmost branch of $\Pi$ until the point that there is a repetition of the context $(\to_3, B)$ and the $\to$-left rule can be applied again without offending the $\mathbf{LMT}^{\to}$ strategy. This result is produced by the $PROOFUNTIL$ call when applying $F'$ to an $\mathbf{LJ}^{\to}$ fragment that end with a $\to$-left rule application.

$$\dfrac{\dfrac{\dfrac{\Pi'_1 \qquad \Pi'_2}{\{\to_3\},\to_3^B,\to_3,\to_1,\to_1^B,A^B,\to_3^B\Rightarrow[B],A}\to\text{-l}}{\{\},\to_3^B,\to_3,\to_1,\to_1^B,A^B,\to_3^B\Rightarrow[B],A}\text{focus}}{\{\to_3,\to_1\},\to_3^B,\to_3^A,\to_1^A,\to_1,A,\to_3\Rightarrow[B,A],B}\text{restart}$$

where $\Pi'_1$ is:

$$\dfrac{\dfrac{\dfrac{\dfrac{\{\to_3,\to_1\},\to_3^B,\to_3^A,\to_1^A,\to_1^B,A^B,\to_3^B,\to_3,\to_1,A\Rightarrow[B,A],B}{\{\to_3,\to_1\},\to_3^B,\to_3^A,\to_1^A,\to_1^B,A^B,\to_3^B,\to_3,\to_1\Rightarrow[B,A],\to_0}\to\text{-r} \qquad \{\to_3\},\to_3^B,\to_3^A,\to_1^A,\to_1^B,A^B,\to_3^B,\to_3,\to_1,A\Rightarrow[B,A],A}{\{\to_3,\to_1\},\to_3^B,\to_3^A,\to_1^A,\to_1^B,A^B,\to_3^B,\to_3,\to_1\Rightarrow[B,A],A}\to\text{-l}}{\{\to_3\},\to_3^B,\to_3^A,\to_1^A,\to_1^B,A^B,\to_3^B,\to_3,\to_1\Rightarrow[B,A],A}\text{focus}}{\{\to_3\},\to_3^B,\to_3^A,\to_1^A,\to_1^B,A^B,\to_3^B,\to_3,\to_1\Rightarrow[B,A],\to_2}\to\text{-r}$$

and $\Pi'_2$ is:

$$\dfrac{\dfrac{\dfrac{\{\to_3,\to_1\},\to_3^B,\to_3^A,\to_1^A,\to_1^B,A^B,\to_3^B,B^A,\to_3,\to_1,B,A\Rightarrow[B,A],B}{\{\to_3,\to_1\},\to_3^B,\to_3^A,\to_1^A,\to_1^B,A^B,\to_3^B,B^A,\to_3,\to_1,B\Rightarrow[B,A],\to_0}\to\text{-r} \qquad \{\to_3,to_1\},\to_3^B,\to_3,\to_1,\to_1^B,A^B,\to_3^B,B,A\Rightarrow[B],A}{\{\to_3,\to_1\},\to_3^B,\to_3,\to_1,\to_1^B,A^B,\to_3^B,B\Rightarrow[B],A}\to\text{-l}}{\{\to_3\},\to_3^B,\to_3,\to_1,\to_1^B,A^B,\to_3^B,B\Rightarrow[B],A}\text{focus}$$

The top sequent of the fragment $\Pi'_1$ is the point where we can apply the $\rightarrow$-left rule to the context $(\rightarrow_3, B)$ again. Thus the next recursion call becomes:

$$\frac{\Pi'_3 \quad \Pi'_4}{\Pi} \rightarrow 1$$

where

$$\Pi'_3 = F' \left( \frac{\rightarrow_3, \rightarrow_1, A \Rightarrow A}{\rightarrow_3, \rightarrow_1, A \Rightarrow \rightarrow_2} \; ; \; \begin{array}{ccc} \Gamma & = & \{\rightarrow_3, \rightarrow_1, A\}; \\ \Delta & = & \{\rightarrow_3, \rightarrow_1\}; \\ \Upsilon & = & \{\rightarrow_3^A, \rightarrow_1^A, \rightarrow_3^B, \rightarrow_1^B, A^B\}; \\ \Sigma & = & \{B, A\}; \\ \Pi & & \end{array} \right)$$

and

$$\Pi'_4 = F' \left( \rightarrow_3, \rightarrow_1, A, B \Rightarrow B \; ; \; \begin{array}{ccc} \Gamma & = & \{\rightarrow_3, \rightarrow_1, A\}; \\ \Delta & = & \{\rightarrow_3, \rightarrow_1\}; \\ \Upsilon & = & \{\rightarrow_3^A, \rightarrow_1^A, \rightarrow_3^B, \rightarrow_1^B, A^B\}; \\ \Sigma & = & \{B, A\}; \\ \Pi & & \end{array} \right)$$

Finally, after finish the translation process we obtained the translated proof of (4-6).

$$
\dfrac{
  \dfrac{
    \dfrac{
      \dfrac{
        \begin{array}{c}
          \dfrac{\{\to_3, \to_1\}, \to_3^B, \to_3^A, \to_1^A, \to_1^B, A^B, \to_3^B, \to_3, \to_1, A \Rightarrow [B, A], A}{\{\to_3, \to_1\}, \to_3^B, \to_3^A, \to_1^A, \to_1^B, A^B, \to_3^B, \to_3, \to_1, A \Rightarrow [B, A], \to_2}\;\to \text{r}
          \qquad
          \{\to_3, \to_1\}, \to_3^B, \to_3^A, \to_1^A, \to_1^B, A^B, \to_3^B, \to_3, \to_1, A \Rightarrow [B, A], A \\
          \Pi' \\
          \dfrac{\{\to_3, \to_1\}, \to_3^B, \to_3^A, \to_1^A, \to_3, \to_1 \Rightarrow [B, A], \to_0}{\{\to_3, \to_1\}, \to_3^B, \to_3, \to_1 \Rightarrow [B], A}\;\to \text{l}
        \end{array}
        \qquad
        \{\to_3, \to_1\}, \to_3^B, \to_3, \to_1, A \Rightarrow [B], A
      }{
        \dfrac{\{\to_3\}, \to_3^B, \to_3, \to_1 \Rightarrow [B], A}{\{\to_3\}, \to_3^B, \to_3 \Rightarrow [B], \to_2}\;\to \text{r}
      }\;\to \text{focus}
      \qquad
      \{\to_3\}, \to_3, B \Rightarrow [], B
    }{
      \{\to_3\}, \to_3 \Rightarrow [], B
    }\;\to \text{l}
  }{
    \{\}, \to_3 \Rightarrow [], B
  }\;\to \text{focus}
}{
  \{\} \Rightarrow [], ((((A \to_0 B) \to_1 A) \to_2 A) \to_3 B) \to_4 B
}\;\to \text{r}
\tag{4-6}
$$

### 4.3.1 Termination

To control the end of the proof search procedure of $\mathbf{LMT}^{\rightarrow}$ our approach is to define an upper bound limit to the size of its proof search tree. Then, we need to show that the $\mathbf{LMT}^{\rightarrow}$ strategy here proposed allows exploring all the possible ways to expand the proof tree until it reaches this size.

From Theorem 3, we know that the upper bound for cut-free proofs based on $\mathbf{LJ}^{\rightarrow}$ is $|\alpha| \cdot 2^{|\alpha|+1}$, where $\alpha$ is the initial formula that we want to prove. We use the translation presented in Figure 4.3 on the previous Section to find a similar limit for $\mathbf{LMT}^{\rightarrow}$proofs. We have to analyze three cases to establish an upper bound for $\mathbf{LMT}^{\rightarrow}$.

The cases are described bellow and are summarized in Table 4.1.

(i) Axioms of $\mathbf{LJ}^{\rightarrow}$ maps one to one with axioms of $\mathbf{LMT}^{\rightarrow}$;

(ii) $\rightarrow$-right applications of $\mathbf{LJ}^{\rightarrow}$ maps one to one with $\rightarrow$-right applications of $\mathbf{LMT}^{\rightarrow}$;

(iii) $\rightarrow$-left applications of $\mathbf{LJ}^{\rightarrow}$ maps to $\mathbf{LMT}^{\rightarrow}$ in three different possible sub-cases, according to the context $(\alpha \rightarrow \beta, q)$ in which the rule is being applied in $\mathbf{LJ}^{\rightarrow}$. We have to consider the fragment of $\mathbf{LMT}^{\rightarrow}$ already translated to decide the appropriate case.

over the proof fragment of $\mathbf{LMT}^{\rightarrow}$ translated until this point of the recursion.

- If the context is **not yet focused neither expanded**

  Then, one application of $\rightarrow$-left in $\mathbf{LJ}^{\rightarrow}$ maps to two applications of rules in $\mathbf{LMT}^{\rightarrow}$: first, a focus application, then an $\rightarrow$-left application.

- If the context is **already focused but not yet expanded**

  Then, one application of $\rightarrow$-left in $\mathbf{LJ}^{\rightarrow}$ maps to one application of $\rightarrow$-left in $\mathbf{LMT}^{\rightarrow}$.

- If the context $(\alpha \rightarrow \beta, q)$ is **already focused and expanded**

  Then, the one application of $\rightarrow$-left in $\mathbf{LJ}^{\rightarrow}$ maps to the height of the $\mathbf{LMT}^{\rightarrow}$ proof fragment produced by the execution of the $PROOFUNTIL$ function. Let this height be called $h$.

| $\mathbf{LJ}^{\rightarrow}$ | $\mathbf{LMT}^{\rightarrow}$ | | Map | |
|---|---|---|---|---|
| axiom | axiom | | 1:1 | |
| $\rightarrow$-right | $\rightarrow$-right | | 1:1 | |
| | focused | expanded | | |
| | No | No | 1:2 | 1 focus and 1 $\rightarrow$-left |
| $\rightarrow$-left | Yes | No | 1:1 | 1 $\rightarrow$-left |
| | Yes | Yes | 1:h | 1 to the size of $PROOFUNTIL$ |

Table 4.1: Mapping the size of $\mathbf{LJ}^{\rightarrow}$ proofs into $\mathbf{LMT}^{\rightarrow}$

**Lemma 5** *The height $h$ that defines the size of the proof fragment returned by the function $PROOFUNTIL$ has a maximum limit of $2^{2log_2|\alpha|}$, where $\alpha$ is the main formula of the initial sequent of the proof in $\mathbf{LMT}^{\rightarrow}$.*

*Proof*: Consider a proof $\prod_{\mathbf{LJ}^{\rightarrow}}$ of an initial sequent in $\mathbf{LJ}^{\rightarrow}$ with the form $\Rightarrow \alpha$. The process of translating $\prod_{\mathbf{LJ}^{\rightarrow}}$ to $\mathbf{LMT}^{\rightarrow}$ produces a proof $\prod_{\mathbf{LMT}^{\rightarrow}}$ with the initial sequent in the form $\{\} \Rightarrow [], \alpha$. Consider that $\alpha$ has the form $\alpha_1 \rightarrow \alpha_2$. In some point of the translation to $\mathbf{LMT}^{\rightarrow}$, we reach a point where a context $(\psi \rightarrow \varphi, q)$ is already focused and expanded in the already translated part of the proof $\prod_{\mathbf{LMT}^{\rightarrow}}$. At this point, the function $PROOFUNTIL$ generates a fragment of the proof $\prod_{\mathbf{LMT}^{\rightarrow}}$, call it $\Sigma$ of size $h$. The height $h$ is bound by the number of applications of $\rightarrow$-left rules in $\Sigma$. This can be determined by the multiplication of the degree of the formula $\alpha_1$ (that bounds the number of possible implicational formulas in the left side of a sequent in $\mathbf{LMT}^{\rightarrow}$) by the maximum number of atomic formulas ($n$) inside the []-area in the highest branch of $\Sigma$ (each $p_i$ inside the []-area allows one application of the restart rule). Thus we can formalize this in the following manner:

$$\begin{aligned} h &= n \times |\alpha_1| \\ h &= |\alpha| \times |\alpha| \\ h &= |\alpha|^2 \end{aligned}$$

Since

$$|\alpha|^2 = 2^{2 \cdot log_2|\alpha|}$$

Then, we have that

$$h = 2^{2 \cdot log_2|\alpha|}$$

■

**Theorem 6** *The size of a proof for a formula $\alpha \in \mathbf{M}^{\rightarrow}$ in $\mathbf{LMT}^{\rightarrow}$ has an upper bound of $|\alpha| \cdot 2^{|\alpha|+1+2\cdot log_2|\alpha|}$.*

*Proof*: Considering the size of proofs for a formula $\alpha$ using $\mathbf{LJ}^{\rightarrow}$, the mapping in Table 4.1 and the Lemma 5, the proof follows directly. ■

**Theorem 7** $\mathbf{LMT}^{\rightarrow}$ *eventually stops.*

*Proof*: To guarantee termination, we use the upper bound presented in Theorem 6 to limit the height of opened branches during the $\mathbf{LMT}^{\rightarrow}$ proof search process. The strategy presented in Section 4.2 forces an ordered application of rules that produces all possible combination of formulas to be expanded in the right and left sides of generated sequents. In other words, when the upper bound is reached the proof search had, for sure, stressed all possible expansions until that point.

– $\rightarrow$-right rule is applied until we obtain an atomic formula on the right side.

– focus rule is applied until every non-labeled formula become focused. The same formula can not be focused twice unless a restart rule is applied.

– $\rightarrow$-left rule can not be applied more than once to the same context unless a restart rule is applied.

– between two applications of the restart rule in a branch there is only one possible application of a $\rightarrow$-left rule for a context $(\alpha, q)$. $\alpha$ and $q$ are always subformulas of the initial formula.

– restart rule is applied for each atomic formula that appears on the right side of sequents in a branch in the order of its appearance in the []-area, which means that proof search will apply the restart rule for each $p_i, i = 1, \ldots n$ until the branch reaches the defined limit.

■

The proof of completeness of $\mathbf{LMT}^{\rightarrow}$ is closely related with this strategy and with the way the proof tree is labeled during the proving process. Section 4.4 presents the soundness proof of $\mathbf{LMT}^{\rightarrow}$ and Section 4.5, the completeness proof.

## 4.4 Soundness

In this section, we prove the soundness of $\mathbf{LMT}^{\rightarrow}$. A few basic facts and definitions used in the proof follow.

**Definition 8** *A sequent* $\{\Delta'\}, \Upsilon_1^{p_1}, \Upsilon_2^{p_2}, \ldots, \Upsilon_n^{p_n}, \Delta \;\Rightarrow\; [p_1, p_2, \ldots, p_n], \varphi$ *is* valid*, if and only if,* $\Delta', \Delta \models \varphi$ *or* $\exists i(\bigcup_{k=1}^{i} \Upsilon_k^{p_k}) \models p_i$*, for* $i = 1, \ldots n$*.*

**Definition 9** *We say that a rule is* sound*, if and only if, in the case of the premises of the sequent are valid sequents, then its conclusion also is.*

A calculus is sound, if and only if, each of its rules is sound. We prove the soundness of $\mathbf{LMT}^{\rightarrow}$ by showing that this is the case for each one of its rules.

**Proposition 10** *Considering validity of a sequent as defined in Definition 8,* $\mathbf{LMT}^{\rightarrow}$ *is sound.*

*Proof*: We show that supposing that premises of a rule are valid then, the validity of the conclusion follows. In the sequel, we analyze each rule of $\mathbf{LMT}^{\rightarrow}$.

**$\rightarrow$-left** We need to analyze both premises together. Thus we have the combinations described below.

- Supposing the left premise is valid because $\alpha \rightarrow \beta, \Delta', \Delta \models \alpha$ and the right premise is valid because $\alpha \rightarrow \beta, \Delta', \Delta, \beta \models q$. We also know that $\alpha \rightarrow \beta \in \Delta$ and $\Delta' \subseteq \Delta$. In this case, the conclusion holds:

$$\dfrac{\dfrac{\dfrac{\alpha \rightarrow \beta \;\; \Delta' \;\; \Delta}{\prod} \quad \atop \alpha \qquad\qquad \alpha \rightarrow \beta}{\beta}}{q}$$

- Supposing the left premise is valid because $\exists i(\bigcup_{k=1}^{i} \Upsilon_k^{p_k}) \models p_i$, for $i = 1, \ldots, n$, the conclusion holds as it is the same. Supposing the left premise is true because $\Delta^q \models q$, the conclusion also holds, as $\Delta^q = \Delta$.
- Supposing the right premise is valid because $\exists i(\bigcup_{k=1}^{i} \Upsilon_k^{p_k}) \models p_i$, for $i = 1, \ldots, n$, then conclusion also holds.

**restart** Here, we have three cases to evaluate.

- Supposing the premise is valid because $\Upsilon_1^{p_1}, \Upsilon_2^{p_2}, \ldots, \Upsilon_i^{p_i} \models p_i$, then $\exists i(\bigcup_{k=1}^{i} \Upsilon_k^{p_k}) \models p_i$, for $i = 1, \ldots, n$. The conclusion is also valid.
- Supposing the premise is valid because $\exists j(\bigcup_{k=i+1}^{j} \Upsilon_k^{p_k}) \models p_j$, for $j = i+1, \ldots, n$, then conclusion also holds.
- Supposing the premise is valid because $\Delta^q \models q$, then $\Delta \models q$ and, as $\Delta' \subseteq \Delta$, $\Delta', \Delta \models q$.

**$\rightarrow$-right**

- Supposing the premise is valid because $\Upsilon_1^{p_1}, \Upsilon_2^{p_2}, \ldots, \Upsilon_i^{p_i} \models p_i$, then $\exists i(\bigcup_{k=1}^{i} \Upsilon_k^{p_k}) \models p_i$, for $i = 1, \ldots, n$. This is also valid in the conclusion.
- Supposing the premise is valid because $\Delta', \Delta, \alpha \models \beta$, then every Kripke model that satisfies $\Delta'$, $\Delta$ and $\alpha$ also satisfies $\beta$. We know that $\Delta' \subseteq \Delta$. Those models also satisfies $\alpha \rightarrow \beta$ and, then, conclusion also holds.

**focus**

- Supposing the premise is valid because $\Upsilon_1^{p_1}, \Upsilon_2^{p_2}, \ldots, \Upsilon_i^{p_i} \models p_i$, then $\exists i(\bigcup_{k=1}^{i} \Upsilon_k^{p_k}) \models p_i$, for $i = 1, \ldots, n$. This is also valid in the conclusion.
- Supposing the premise is valid because $\Delta', \alpha, \Delta, \alpha \models \beta$, then the conclusion also holds as $\Delta', \Delta, \alpha \models \beta$.

■

From Proposition 10, we conclude that **LMT**$^{\rightarrow}$ only prove tautologies.

## 4.5 Completeness

By Observation 4.ii we know that a top sequent of an open branch in an attempt proof tree has the general form below, where $q$ is an atomic formula:

$$\{\Delta'\}, \Upsilon_1^{p_1}, \Upsilon_2^{p_2}, ..., \Upsilon_n^{p_n}, \Delta \Rightarrow [p_1, p_2, ..., p_n], q$$

From Definition 8 and considering that $\Delta' \in \Delta$ in any sequent of an attempt proof tree following our proposed strategy, we can define a invalid sequent as follows:

**Definition 11** *A sequent is* invalid *if and only if* $\Delta \not\models q$ *and* $\forall i(\bigcup_{k=1}^{i} \Upsilon_k^{p_k}) \not\models p_i$, *for* $i = 1, \ldots, n$.

Our proof of the completeness starts with a definition about atomic formulas in the left and right side of a top sequent.

**Definition 12** *We can construct a Kripke counter-model $\mathcal{M}$ that satisfies atomic formulas in the right side of a top sequent and that falsifies the atomic formula on the left. This construction can be done in the following way:*

1. *The model $\mathcal{M}$ has an initial world $w_0$.*

2. *By the proof strategy, we can conclude that, in any sequent of the proof tree, $\Upsilon_1^{p_1} \subseteq \Upsilon_2^{p_2} \subseteq \cdots \subseteq \Upsilon_n^{p_n} \subseteq \Delta$. We create a world in the model $\mathcal{M}$ corresponding for each one of these bags of formulas and, using the inclusion relation between them, we define a respective accessibility relation in the model $\mathcal{M}$ between such worlds. That is, we create worlds $w_{\Upsilon_1^{p_1}}, w_{\Upsilon_2^{p_2}}, \ldots, w_{\Upsilon_n^{p_n}}, w_\Delta$ related in the following form: $w_{\Upsilon_1^{p_1}} \preceq w_{\Upsilon_2^{p_2}} \preceq \cdots \preceq w_{\Upsilon_n^{p_n}} \preceq w_\Delta$. As $w_0$ is the first world of $\mathcal{M}$, it precedes $w_{\Upsilon_1^{p_1}}$, that is, $w_0 \preceq w_{\Upsilon_1^{p_1}}$ is also included in the accessibility relation. If $\Upsilon_i^{p_i} = \Upsilon_{i+1}^{p_{i+1}}$, for $i = 1, \ldots, n$, then the associated worlds that correspond to those sets have to be collapsed in a single world $w_{\Upsilon_i^{p_i} - \Upsilon_{i+1}^{p_{i+1}}}$. In this case, the previous relation $w_{\Upsilon_i^{p_i}} \preceq w_{\Upsilon_{i+1}^{p_{i+1}}}$ is removed from the $\preceq$ relation of the model $\mathcal{M}$ and the pairs $w_{\Upsilon_{i-1}^{p_{i-1}}} \preceq w_{\Upsilon_i^{p_i}}$ and $w_{\Upsilon_{i+1}^{p_{i+1}}} \preceq w_{\Upsilon_{i+2}^{p_{i+2}}}$ become respectively $w_{\Upsilon_{i-1}^{p_{i-1}}} \preceq w_{\Upsilon_i^{p_i} - \Upsilon_{i+1}^{p_{i+1}}}$ and $w_{\Upsilon_i^{p_i} - \Upsilon_{i+1}^{p_{i+1}}} \preceq w_{\Upsilon_{i+2}^{p_{i+2}}}$.*

3. *By the Definition 11 of an invalid sequent, $\Delta \nvDash q$. The world $w_\Delta$ will be used to guarantee this. We set $q$ false in $w_\Delta$, i.e, $\mathcal{M} \nvDash_{w_\Delta} q$. We also set every atomic formula that is in $\Delta$ as true, i.e., $\forall p, p \in \Delta, \mathcal{M} \vDash_{w_\Delta} p$.*

4. *By the Definition 11 of an invalid sequent, we also need that $\forall i (\bigcup_{k=1}^{i} \Upsilon_k^{p_k}) \nvDash p_i$, for $i = 1, \ldots n$. Thus, for each $i, i = 1, \ldots, n$ we set $\mathcal{M} \nvDash_{w_{\Upsilon_i^{p_i}}} p_i$ and $\forall p, p \in \Upsilon_i^{p_i}$, being $p$ an atomic formula, $\mathcal{M} \vDash_{w_{\Upsilon_i^{p_i}}} p$. In the case of collapsed worlds, we keep the satisfaction relation of the previous individual worlds in the collapsed one.*

5. *In $w_0$ set every atomic formula inside the []-area (all of them are atomic) as false. That is, $\mathcal{M} \nvDash_{w_0} p_i$, for $i = 1, \ldots, n$. We also set the atomic formula outside the []-area false in this world: $\mathcal{M} \nvDash_{w_0} q$. Those definitions make $w_0$ consistent with the $\preceq$ relation of $\mathcal{M}$.*

The Figure 4.4 shows the general shape of counter-models following the steps enumerated above. This procedure to construct counter-model allows us to state the following lemma:

$$
\begin{array}{c}
\mathbf{w_0} \\
\not\models p_1, \ldots, p_n \\
\not\models q \\
| \\
\mathbf{w_{\Upsilon_1^{p_1}}} \\
\forall p, p \in \Upsilon_1^{p_1}\ , p \text{ atomic}, \models p \\
\not\models p_1 \\
| \\
\vdots \\
| \\
\mathbf{w_{\Upsilon_n^{p_n}}} \\
\forall p, p \in \Upsilon_n^{p_n}\ , p \text{ atomic}, \models p \\
\not\models p_n \\
| \\
\mathbf{w_\Delta} \\
\forall p, p \in \Delta\ , p \text{ atomic}, \models p \\
\not\models q
\end{array}
$$

Figure 4.4: General schema of counter-models

**Lemma 13** *Let $S$ be a top sequent of an open branch in an attempt proof tree generated by the strategy presented in Section 4.2. Then we can construct a Kripke model $\mathcal{M}$ with a world $u$ where $\mathcal{M} \not\models_u S$, using the aforementioned counter-model generation procedure.*

*Proof*: We can prove this by induction on the degree of formulas in $\Delta$. From Definition 12, items 3 and 4 we know the value of each atomic formula in the worlds $w_\Delta$ and in each world $w_{\Upsilon_i^{p_i}}$. The inductive hypothesis is that every formula in $\Delta$ is true in $w_\Delta$. Thus, as $\Upsilon_1^{p_1} \subseteq \Upsilon_2^{p_2} \subseteq \cdots \subseteq \Upsilon_n^{p_n} \subseteq \Delta$, every formula in $\Upsilon_i^{p_i}$ is true in $w_{\Upsilon_i^{p_i}}$, for $i = 1, \ldots, n$.

Thus, we have two cases to consider:

1. The top sequent is in the rightmost branch of the proof tree ([]-area is empty).
   Let $\alpha \rightarrow \beta$ be a formula in $\mathbf{M}^{\rightarrow}$ that is in $\Delta$. We show that $\mathcal{M} \models_{w_\Delta} \alpha \rightarrow \beta$. In this case, by the proof strategy, $\beta \equiv (\beta_1 \rightarrow (\beta_2 \rightarrow \cdots \rightarrow (\beta_m \rightarrow p)))$, where $p$ is an atomic formula. By Definition 12.3 $\models_{w_\Delta} p$. As $w_\Delta$ has no accessible world from it (except for itself), $\models_{w_\Delta} \beta$. By the proof strategy, $\beta_m \rightarrow p, \beta_{m-1} \rightarrow \beta_m \rightarrow p, \ldots, \beta_2 \rightarrow \cdots \rightarrow \beta_{m-1} \rightarrow \beta_m \rightarrow p, \beta_1 \rightarrow \beta_2 \rightarrow \cdots \rightarrow \beta_{m-1} \rightarrow \beta_m \rightarrow p$ also are in $\Delta$. The degree of each of these formulas is less than the degree of $\alpha \rightarrow \beta$ and, by the induction hypothesis, all of them are true in $w_\Delta$. Thus $\models_{w_\Delta} \beta$ and $\models_{w_\Delta} \alpha \rightarrow \beta$.
   As the []-area is empty, the sets $\Upsilon_i^{p_i}$ are also empty. The counter-model only has two words, $w_0$ and $w_\Delta$, following the properties described in Definition 12.

2. The top sequent is in any other branch that is not the rightmost one ([]-area is not empty).
   Let $\alpha \to \beta$ be a formula in $\mathbf{M}^{\to}$ that is in $\Delta$. We show that $\mathcal{M} \vDash_{w_\Delta} \alpha \to \beta$. In this case, by the proof strategy, $\alpha \equiv (\alpha_1 \to (\alpha_2 \to \cdots \to (\alpha_m \to q)))$, where $q$ is the atomic formula in the right side of the sequent, out of the []-area. By Definition 12.3 $\nvDash_{w_\Delta} q$. By the proof strategy, $\alpha_1, \alpha_2, \ldots, \alpha_m$ also are in $\Delta$. The degree of each of these formulas is less than the degree of $\alpha \to \beta$ and, by the induction hypothesis, all of them are true in $w_\Delta$. This ensures $\nvDash_{w_\Delta} \alpha$ and $\vDash_{w_\Delta} \alpha \to \beta$.
   Considering now a formula $\alpha \to \beta$ from $\mathbf{M}^{\to}$ that is in $\Upsilon_i^{p_i}$. By Definition 12.2, $\alpha \to \beta$ also belongs to $\Delta$. From the last paragraph, we show that, for any formula $\alpha \to \beta \in \Delta$, $\nvDash_{w_\Delta} \alpha$. As $\nvDash_{w_\Delta} \alpha$, by the accessibility relation of the Kripke model, $\nvDash_{w_{\Upsilon_i^{p_i}}} \alpha$, for each $i = 1, \ldots, n$. Thus, the value of $\alpha \to \beta$ is defined in any of these worlds by the value of $\alpha \to \beta$ in $w_\Delta$, that we showed to be true. Thus, $\vDash_{w_{\Upsilon_i^{p_i}}} \alpha \to \beta$.

As stated in Definition 12.2, $\Upsilon_1^{p_1} \subseteq \Upsilon_2^{p_2} \subseteq \cdots \subseteq \Upsilon_n^{p_n} \subseteq \Delta$ and following the accessibility relation rule of the $\mathbf{M}^{\to}$ semantic (relations are reflexive and transitive) we conclude that:

$$\begin{aligned}
\mathcal{M} &\vDash_{w_0} \Upsilon_1^{p_1}, \nvDash_{w_0} p_1\\
&\vDash_{w_0} \Upsilon_2^{p_2}, \nvDash_{w_0} p_2\\
&\vdots\\
&\vDash_{w_0} \Upsilon_n^{p_n}, \nvDash_{w_0} p_n\\
&\vDash_{w_0} \Delta, \nvDash_{w_0} q
\end{aligned}$$

Proving Lemma 13. ■

**Definition 14** *A rule is said* invertible *or* double-sound *iff the validity of its conclusion implies the validity of its premises.*

In other words, by Definition 14 we know that a counter-model for a top sequent of a proof tree which can not be expanded anymore can be used to construct a counter-model to every sequent in the same branch of the tree until the conclusion (root sequent). In the case of the →-right rule in our system, not just if the premise of the rule has a counter-model then so does the conclusion, but the same counter-model will do. Weich (1998) called rules with this property *preserving counter model.* Dyckhoff (personal communication, 2015) proposed to call this kind of rules of *strongly invertible* rules. In the case of the $\rightarrow$-left rule, this is the same when one of the premises is valid, but, considering the case that both premises are not valid, we need to mix the counter-models of both sides to construct the counter-model for the conclusion of the rule. This way to produce counter-models is what we call a *weakly invertible* rule.

**Lemma 15** *The rules of* $\mathbf{LMT}^{\rightarrow}$ *are invertible.*

*Proof*: We show that the rules of $\mathbf{LMT}^{\rightarrow}$ are invertible when considering a proof tree labeled in the schema presented in Section 4.2. We prove that for the structural rules (focus and restart) and $\rightarrow$-right, from the existence of a Kripke model that makes the premise of the rule invalid follows that the conclusion is also invalid. For the $\rightarrow$-left rule, from the Kripke models of the premises, we can construct a Kripke model that also makes the conclusion of the rule invalid.

**$\rightarrow$-right** If the premise is invalid, then there is a Kripke model $\mathcal{M}$ where $\Delta', \Delta, \alpha \nvDash \beta$ and $\forall i (\bigcup_{k=1}^{i} \Upsilon_k^{p_k}) \nvDash p_i$, for $i = 1, \ldots, n$ in a given world $u$ of $\mathcal{M}$. Thus, in the conclusion we have:

- By the definition of semantics of Section 2.5, there have to be a world $v$, $u \preceq v$, in the model $\mathcal{M}$ where $\Delta', \Delta, \alpha$ are satisfied and where $\beta$ is not. Thus, in $u$, $\alpha \rightarrow \beta$ can not hold.
- By the model $\mathcal{M}$, for each $i$, exists a world $v_i$, $u \preceq v_i$, where $\vDash_{v_i} \Upsilon_i^{p_i}$ and $\nvDash_{v_i} p_i$.
- Thus, the conclusion is also invalid.

**$\rightarrow$-left** Considering that one of the premises of $\rightarrow$-left is not valid, the conclusion also is. We have to evaluate three cases:

1. **The right premise is invalid but the left premise is valid**. Then there is a Kripke model $\mathcal{M}$ where $\alpha \rightarrow \beta, \Delta', \Delta, \beta \nvDash q$ and $\forall i (\bigcup_{k=1}^{i} \Upsilon_k^{p_k}) \nvDash p_i$, for $i = 1, \ldots, n$ from a given world $u$. Thus, in the conclusion we have:
   - By the model $\mathcal{M}$, there have to be a world $v$, $u \preceq v$, in the model where $\alpha \rightarrow \beta, \Delta', \Delta, \beta$ are satisfied and where $q$ is not.
   - By the model $\mathcal{M}$, for each $i$, exists a world $v_i$, $u \preceq v_i$, where $\vDash_{v_i} \Upsilon_i^{p_i}$ and $\nvDash_{v_i} p_i$.

- Thus, the conclusion is invalid too.

2. **The left premise is invalid but the right premise is valid**. Then there is a Kripke model $\mathcal{M}$ where $\alpha \to \beta, \Delta', \Delta \not\models \alpha$ and $\forall i(\bigcup_{k=1}^{i} \Upsilon_k^{p_k}) \not\models p_i$, for $i = 1, \ldots, n$, and $\Delta^q \not\models q$ from a given world $u$. Thus, in the conclusion we have:

   - By the model $\mathcal{M}$, there have to be a world $v$, $u \preceq v$, in the model where $\alpha \to \beta, \Delta', \Delta$ are satisfied and where $\alpha$ is not.
   - By the model $\mathcal{M}$, for each $i$, exists a world $v_i$, $u \preceq v_i$, where $\models_{v_i} \Upsilon_i^{p_i}$ and $\not\models_{v_i} p_i$.
   - We also know by $\mathcal{M}$ that there is a world $v_{\Delta^q}$, $u \preceq v_{\Delta^q}$, where $\models_{v_{\Delta^q}} \Delta^q$ and $\not\models_{v_{\Delta^q}} q$. We also have that $\Delta^q = \Delta$ and that $\alpha \to \beta \in \Delta$. Therefore, $\models_{v_{\Delta^q}} \Delta'$ and $\models_{v_{\Delta^q}} \alpha \to \beta$.
   - Thus, the conclusion can not be valid.

3. **Both left and right premises are invalid**. Then there are two models $\mathcal{M}_1$ and $\mathcal{M}_2$, from the right and left premises respectively. In $\mathcal{M}_1$ there is a world $u_1$ that makes the right sequent invalid as described in item 1. In $\mathcal{M}_2$ there is a world $u_2$ that makes the sequent of the left premise invalid as described in item 2. Considering the way Kripke models are constructed based on Lemma 13, we know that $u_1$ and $u_2$ are root worlds of their respective counter-models. Thus, converting the two models into one, $\mathcal{M}_3$, by mixing $u_1$ and $u_2$ in the root of $\mathcal{M}_3$, called $u_3$, we have that in $u_3$:

   - $\alpha \to \beta, \Delta', \Delta$ are satisfied and $\alpha$ is not.
   - for $i = 1, \ldots, n$, we have that $\models_{u_3} \Upsilon_i^{p_i}$ and $\not\models_{u_3} p_i$.
   - $\not\models_{u_3} q$
   - Thus, the conclusion is also invalid.

**focus** If we have a model that invalidates the premise, this model also invalidates the conclusion as the sequents in the premise and in the conclusion are the same despite the repetition of the focused formula $\alpha$.

**restart** If the restart premise is invalid, then there is a Kripke model $\mathcal{M}$ and a world $u$ from which $\Upsilon_1, \Upsilon_2, \ldots, \Upsilon_i \not\models p_i$ and $\forall j(\bigcup_{k=1}^{j} \Upsilon_k^{p_k}) \models p_j$, for $j = i+1, \ldots, n$, and $\Delta^q \not\models q$. Thus, in the conclusion we have:

- By the model $\mathcal{M}$, there have to be a world $v$, $u \preceq v$, in the model where $\Upsilon_1, \Upsilon_2, \ldots, \Upsilon_i$ are satisfied and where $p_i$ is not. Each $\Upsilon_k$ has the same formulas as $\Upsilon_k^{p_k}$ and, by the restart condition, we know that $\not\models p_k$, for $k = 1, \ldots, i$.

– By the model $\mathcal{M}$, for each $j$, exists a world $v_j$, $u \preceq v_j$, where $\vDash_{v_j} \Upsilon_j^{p_j}$ and $\nvDash_{v_j} p_j$.

– We also know by $\mathcal{M}$ that there is a world $v_q$, $u \preceq v_q$, where $\vDash_{v_q} \Delta^q$ and $\nvDash_{v_q} q$. We also have that $\Delta' \subseteq \Delta$. Therefore, $\vDash_{v_q} \Delta'$.

– Thus, the conclusion is invalid.

■

Now we can state a proposition about completeness of $\mathbf{LMT}^{\rightarrow}$:

**Proposition 16** $\mathbf{LMT}^{\rightarrow}$ *is complete regarding the proof strategy presented in Section 4.2*

*Proof*: It follows direct from Proposition 7 ($\mathbf{LMT}^{\rightarrow}$ terminates) and Lemma 13 (we can construct a counter-model for a top sequent in a terminated open branch of $\mathbf{LMT}^{\rightarrow}$) and Lemma 15 (the rules of $\mathbf{LMT}^{\rightarrow}$ are invertible). ■

### 4.5.1 Examples

**Example 17** *As an example, consider the Peirce formula,* $((A \to B) \to A) \to A$*, which is very known to be a Classic tautology, but not a tautology in Intuitionistic neither in Minimal Logic (which includes* $\mathbf{M}^{\to}$*). Proof* (4-7) *below shows the open branch of an attempt proof tree for this formula. Following our termination criteria, this branch should be higher than the fragment below, but, to help improve understanding, we stop that branch in the point from which proof search just produces repetition (similar to the use of a loop checker).*

$$
\begin{prooftree}
\AxiomC{$\vdots$}
\AxiomC{$\vdots$}
\AxiomC{$\vdots$}
\AxiomC{$\to_1, A, (\to_1)^A, \{\ \to_1, A\}\Rightarrow_{28}[A,B], B$}
\RightLabel{$\to \mathit{left}\!- \to_1$}
\BinaryInfC{$\to_1, A, (\to_1)^A, \{\ \to_1, A\}\Rightarrow_{26}[A,B], B$}
\RightLabel{$\mathit{focus} - A$}
\UnaryInfC{$\to_1, A, (\to_1)^A, \{\ \to_1\ \}\Rightarrow_{25}[A,B], B$}
\RightLabel{$\mathit{focus}\!- \to_1$}
\UnaryInfC{$\to_1, A, (\to_1)^A, \{\}\Rightarrow_{22}[A,B], B$}
\RightLabel{$\mathit{restart} - B$}
\UnaryInfC{$(\to_1)^B, \to_1, (A)^B, (\to_1)^A, A, \{\ \to_1, A\}\Rightarrow_{21}[B,A], B$}
\RightLabel{$\to \mathit{left}\!- \to_1$}
\BinaryInfC{$(\to_1)^B, \to_1, (A)^B, (\to_1)^A, A, \{\ \to_1, A\}\Rightarrow_{19}[B,A], B$}
\RightLabel{$\mathit{focus} - A$}
\UnaryInfC{$(\to_1)^B, \to_1, (A)^B, \{\ \to_1\ \}, (\to_1)^A, A\Rightarrow_{16}[B,A], B$}
\RightLabel{$\to \mathit{right}\!- \to_0$}
\UnaryInfC{$(\to_1)^B, \to_1, (A)^B, \{\ \to_1\ \}, (\to_1)^A\Rightarrow_{13}[B,A], \to_0$}
\AxiomC{$\vdots$}
\RightLabel{$\to \mathit{left}\!- \to_1$}
\BinaryInfC{$(\to_1)^B, \to_1, (A)^B, \{\ \to_1\ \}\Rightarrow_{12}[B], A$}
\RightLabel{$\mathit{focus}\!- \to_1$}
\UnaryInfC{$(\to_1)^B, \to_1, (A)^B, \{\}\Rightarrow_{11}[B], A$}
\RightLabel{$\mathit{restart} - A$}
\UnaryInfC{$\to_1, (\to_1)^A, A, \{\ \to_1, A\}\Rightarrow_{9}[A], B$}
\RightLabel{$\to \mathit{left}\!- \to_1$}
\BinaryInfC{$\to_1, (\to_1)^A, A, \{\ \to_1, A\}\Rightarrow_{6}[A], B$}
\RightLabel{$\mathit{focus} - A$}
\UnaryInfC{$\to_1, \{\ \to_1\ \}, (\to_1)^A, A\Rightarrow_{5}[A], B$}
\RightLabel{$\to \mathit{right}\!- \to_0$}
\UnaryInfC{$\to_1, \{\ \to_1\ \}, (\to_1)^A\Rightarrow_{3}[A], \to_0$}
\AxiomC{$\vdots$}
\RightLabel{$\to \mathit{left}\!- \to_1$}
\BinaryInfC{$\to_1, \{\ \to_1\ \}\Rightarrow_{2}[], A$}
\RightLabel{$\mathit{focus}\!- \to_1$}
\UnaryInfC{$\{\}, \to_1\ \Rightarrow_{1}[], A$}
\RightLabel{$\to \mathit{right}\!- \to_2$}
\UnaryInfC{$\{\}\Rightarrow_{0}(((A \to_0 B) \to_1 A) \to_2 A), []$}
\end{prooftree}
\tag{4-7}
$$

*From the top sequent of the open branch* ($\Rightarrow_{28}$) *we can generalize the sequent as:*

$$
\begin{array}{cccccc}
\Delta' & \Upsilon_1^{p_1} & \Delta & \Rightarrow_{28} & [p_1, p_2] & q \\
\{\to 1, A\} & \to_1^A & \to_1, A & \Rightarrow_{28} & [A, B], & B
\end{array}
$$

*Thus, following the method described in Section 4.5 we can extract the counter-model below that falsifies the sequent:*

$$\begin{array}{c}
\mathbf{w_0} \\
\not\models A \\
\not\models B \\
\not\models A \rightarrow_0 B \\
\models (A \rightarrow_0 B) \rightarrow_1 A \\
| \\
\mathbf{w_{\Upsilon^A}} \\
\not\models A \\
\not\models B \\
\not\models A \rightarrow_0 B \\
\models (A \rightarrow_0 B) \rightarrow_1 A \\
| \\
\mathbf{w_{\Delta}} \\
\models A \\
\not\models B \\
\not\models A \rightarrow_0 B \\
\models (A \rightarrow_0 B) \rightarrow_1 A
\end{array} \tag{4-8}$$

*From Lemma 15 we can extend this counter-model to falsify the initial sequent* ($\Rightarrow_0$)*, showing that the Pierce rule does not hold on* $\mathbf{M}^{\rightarrow}$.

**Example 18** *As another example, we can consider the Dummett formula: $(A \rightarrow B) \vee (B \rightarrow A)$. It is known that a Kripke counter-model that falsifies this formula need at least two branches in* **Int** *and* **Min***, so it is also in* $\mathbf{M}^{\rightarrow}$*. This example allows us to understand how to mix the right and left premises counter-models to falsifies a conclusion sequent of a $\rightarrow$-left rule.*

*As we want to use* $\mathbf{LMT}^{\rightarrow}$*, we need to convert the Dummett formula from the form above to its implicational version. We use here the general translation presented in Haeusler (2015*b*). Thus, the translated version is a formula $\alpha$ as follows:*

$$\alpha \equiv (((A \rightarrow B) \rightarrow A) \rightarrow (((B \rightarrow A) \rightarrow A) \rightarrow (C \rightarrow A))) \rightarrow ((((A \rightarrow B) \rightarrow B) \rightarrow (((B \rightarrow A) \rightarrow B) \rightarrow (C \rightarrow B))) \rightarrow C)$$

*To shorten the presentation, consider the following abbreviations:*

$$\begin{aligned}\alpha_1 &= ((A \rightarrow B) \rightarrow A) \rightarrow (((B \rightarrow A) \rightarrow A) \rightarrow (C \rightarrow A))\\ \alpha_2 &= ((A \rightarrow B) \rightarrow B) \rightarrow (((B \rightarrow A) \rightarrow B) \rightarrow (C \rightarrow B))\end{aligned}$$

*The tree* (4-9)*, presents a shortened version of a totally expanded attempt proof tree in* $\mathbf{LMT}^{\rightarrow}$*for the Dummet formula in implicational formula. We concentrate in show the application of the operational rules ($\rightarrow$-left and $\rightarrow$-right). We removed from the tree the focus area on the left of each sequent and the applications of structural rules. We also exclude from the tree the labeled formulas, just showing the necessary formulas for the specific point in the tree. The list of sequents above the boxed sequents in the tree represent top sequents of each branch after the total expansion of $\alpha_2$.*

$$
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\begin{array}{c}
\ldots \Rightarrow [B,C],A \\
\cancel{\ldots,B \Rightarrow [C],B} \\
\ldots \Rightarrow [C,B],B \\
\ldots,A \Rightarrow [C],B \\
\ldots \Rightarrow [C],C \\
\ldots,B \Rightarrow [],C \\
| \\
\boxed{\ldots \Rightarrow [C,A],A\ (M_1)}
\end{array}
\quad
\begin{array}{c}
\ldots,B \Rightarrow [B,C],A \\
\cancel{\ldots,B \Rightarrow [C],B} \\
\cancel{\ldots,B \Rightarrow [C,B],B} \\
\cancel{\ldots,B,A \Rightarrow [C],B} \\
\ldots,B \Rightarrow [C],C \\
\ldots,B \Rightarrow [],C \\
| \\
\boxed{\ldots,B \Rightarrow [C],A\ (M_2)}
\end{array}
}{\ldots,A \to B \Rightarrow [C],A}
}{\ldots, \Rightarrow [C],(A \to B) \to A}
\quad
\dfrac{
\dfrac{
\dfrac{
\begin{array}{c}
\ldots \Rightarrow [B,C],A \\
\cancel{\ldots,B \Rightarrow [C],B} \\
\ldots \Rightarrow [C,B],B \\
\ldots,A \Rightarrow [C],B \\
\ldots \Rightarrow [C],C \\
\ldots,B \Rightarrow [],C \\
| \\
\boxed{\ldots \Rightarrow [C,A],q\ (M_1)}
\end{array}
\quad
\cancel{\ldots,A \Rightarrow [C],A}
}{\ldots,B \to A \Rightarrow [C],A}
}{\ldots, \Rightarrow [C],(B \to A) \to A}
\quad
\dfrac{
\begin{array}{c}
\ldots \Rightarrow [B,C],A \\
\cancel{\ldots,B \Rightarrow [C],B} \\
\ldots \Rightarrow [C,B],B \\
\ldots,A \Rightarrow [C],B \\
\ldots \Rightarrow [C],C \\
\ldots,B \Rightarrow [],C \\
| \\
\boxed{\ldots, \Rightarrow [C],C\ (M_1)}
\end{array}
\quad
\begin{array}{c}
\cancel{\ldots,A \Rightarrow [B,C],A} \\
\cancel{\ldots,A,B \Rightarrow [C],B} \\
\ldots,A \Rightarrow [C,B],B \\
\ldots,A \Rightarrow [C],B \\
\ldots,A \Rightarrow [C],C \\
\ldots,A,B \Rightarrow [],C \\
| \\
\boxed{\ldots,A \Rightarrow [],C\ (M_3)}
\end{array}
}{\ldots,C \to A \Rightarrow [],C}
}{\ldots,((B \to A) \to A) \to (C \to A) \Rightarrow [],C}
}{((A \to B) \to A) \to (((B \to A) \to A) \to (C \to A)), \alpha_2 \Rightarrow [],C}
\to -left_{(\alpha_1,C)}
}{\alpha_1,\alpha_2 \Rightarrow [],C}
}{\Rightarrow [],\alpha_1 \to (\alpha_2 \to C)}
}{\Rightarrow [],\alpha}
\qquad (4\text{-}9)
$$

*From the open branches of the tree* (4-9) *we extract the following three models. Models $M_1$ repeats in some branches. Thus, we only represent it once here. We indicated in tree* (4-9)*, in the top sequents of each branch, the corresponding model generated on it, following the Definition 12.*

$M_1$

$$\begin{tikzcd}
 & & \mathbf{w_0} \arrow[dll, dash] \arrow[dl, dash] \arrow[d, dash] \arrow[dr, dash] \arrow[drr, dash] & & \\
\begin{matrix}\mathbf{w_1}\\ \nvDash C\end{matrix} \arrow[d, dash] & \begin{matrix}\mathbf{w_2}\\ \nvDash C\end{matrix} \arrow[d, dash] & \begin{matrix}\mathbf{w_3}\\ \nvDash C\end{matrix} \arrow[d, dash] & \begin{matrix}\mathbf{w_4}\\ \nvDash C\end{matrix} & \begin{matrix}\mathbf{w_5}\\ \nvDash C\\ \vDash B\end{matrix} \\
\begin{matrix}\mathbf{w_{11}}\\ \nvDash B\end{matrix} \arrow[d, dash] & \begin{matrix}\mathbf{w_{21}}\\ \nvDash B\end{matrix} \arrow[d, dash] & \begin{matrix}\mathbf{w_{31}}\\ \vDash A\\ \nvDash B\end{matrix} & & \\
\begin{matrix}\mathbf{w_{12}}\\ \vDash A\end{matrix} & \begin{matrix}\mathbf{w_{22}}\\ \vDash B\end{matrix} & & &
\end{tikzcd}$$

$M_2$

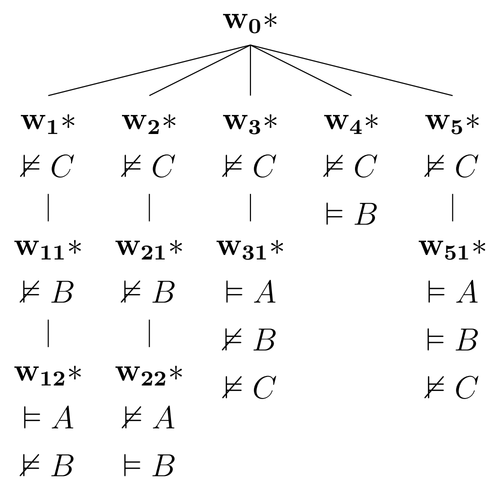
$$\begin{tikzcd}
 & \mathbf{w'_0} \arrow[dl, dash] \arrow[dr, dash] & \\
\begin{matrix}\mathbf{w'_1}\\ \nvDash C\end{matrix} \arrow[d, dash] & & \begin{matrix}\mathbf{w'_2}\\ \nvDash C\end{matrix} \arrow[d, dash] \\
\begin{matrix}\mathbf{w'_{11}}\\ \nvDash B\end{matrix} \arrow[d, dash] & & \begin{matrix}\mathbf{w'_{21}}\\ \vDash B\\ \nvDash C\end{matrix} \\
\begin{matrix}\mathbf{w'_{12}}\\ \vDash B\\ \nvDash A\end{matrix} & &
\end{tikzcd}$$

$M_3$

$$\begin{tikzcd}
 & & \mathbf{w''_0} \arrow[dll, dash] \arrow[dl, dash] \arrow[d, dash] \arrow[dr, dash] & \\
\begin{matrix}\mathbf{w''_1}\\ \nvDash C\end{matrix} \arrow[d, dash] & \begin{matrix}\mathbf{w''_2}\\ \nvDash C\end{matrix} \arrow[d, dash] & \begin{matrix}\mathbf{w''_3}\\ \nvDash C\end{matrix} \arrow[d, dash] & \begin{matrix}\mathbf{w''_4}\\ \nvDash C\end{matrix} \arrow[d, dash] \\
\begin{matrix}\mathbf{w''_{11}}\\ \nvDash B\end{matrix} \arrow[d, dash] & \begin{matrix}\mathbf{w''_{21}}\\ \nvDash B\end{matrix} \arrow[d, dash] & \begin{matrix}\mathbf{w''_{31}}\\ \vDash A\\ \nvDash C\end{matrix} & \begin{matrix}\mathbf{w''_{41}}\\ \vDash A\\ \nvDash B\\ \nvDash C\end{matrix} \\
\begin{matrix}\mathbf{w''_{12}}\\ \vDash A\\ \nvDash B\end{matrix} & \begin{matrix}\mathbf{w''_{22}}\\ \vDash A\end{matrix} & &
\end{tikzcd}$$

*Therefore, at the point in the tree where the rule $\to$-left is applied to the context $(\alpha_1, C)$ (the single labeled rule application in the tree (4-9)) we have the join of these models. Counter-model $M_4$ below represent this unification (to legibility we remove repeated branches in $M_4$). Thus, $M_4 \nvDash \alpha$:*

$M_4$


$\mathbf{w_0}*$
$\mathbf{w_1}*$ $\nvDash C$
$\mathbf{w_2}*$ $\nvDash C$
$\mathbf{w_3}*$ $\nvDash C$
$\mathbf{w_4}*$ $\nvDash C$ $\vDash B$
$\mathbf{w_5}*$ $\nvDash C$
$\mathbf{w_{11}}*$ $\nvDash B$
$\mathbf{w_{21}}*$ $\nvDash B$
$\mathbf{w_{31}}*$ $\vDash A$ $\nvDash B$ $\nvDash C$
$\mathbf{w_{51}}*$ $\vDash A$ $\vDash B$ $\nvDash C$
$\mathbf{w_{12}}*$ $\vDash A$ $\nvDash B$
$\mathbf{w_{22}}*$ $\nvDash A$ $\vDash B$

# 5 Comparing $\mathrm{LMT}^{\rightarrow}$ and Tableaux

## 5.1 Fitting's Tableaux Systems

In this chapter, we explore the relationship between the system $\mathbf{LMT}^{\rightarrow}$ presented in the previous chapter and the Tableaux System in the way it was proposed in Fitting (1969). We will use the name Tableaux to refer generally to the family of systems that follow the same technique. To talk about an individual Tableaux implementation for a specific logic we use the name Tableaux together with the initials of the logic in subscript. For example, Figure 5.1 presents the full set of rules of Fitting's Tableaux for Intuitionistic Propositional Logic (**Int**) that we will just refer to as $\mathbf{Tableaux_{Int}}$.

$$\begin{array}{llll}
T\wedge & \dfrac{S, T(\alpha \wedge \beta)}{S, T\alpha, T\beta} & F\rightarrow & \dfrac{S, F(\alpha \wedge \beta)}{S, F\alpha | S, F\beta} \\[2ex]
T\vee & \dfrac{S, T(\alpha \vee \beta)}{S, T\alpha | S, T\beta} & F\vee & \dfrac{S, F(\alpha \vee \beta)}{S, F\alpha, F\beta} \\[2ex]
T\neg & \dfrac{S, T(\neg\alpha)}{S, F\alpha} & F\neg & \dfrac{S, F(\neg\alpha)}{S_T, T\alpha} \\[2ex]
T\rightarrow & \dfrac{S, T(\alpha \rightarrow \beta)}{S, F\alpha | S, T\beta} & F\rightarrow & \dfrac{S, F(\alpha \rightarrow \beta)}{S_T, T\alpha, F\beta}
\end{array}$$

Figure 5.1: Fitting's Tableaux Rules

Each rule of $\mathbf{Tableaux_{Int}}$ has a main signed formula in the premise and the rule application produces the decomposition of this formula in newly signed formulas on the conclusion. By a *signed formula* we mean $T\alpha$ or $F\alpha$, where $\alpha$ is a formula in **Int**. If S is a set of signed formulas and $\varphi$ is a single signed formula, we will write $S \cup \{\varphi\}$ simply as $S, \varphi$, following Fitting's notation style. In Figure 5.1, $S$ is any set (possibly empty) of signed formulas and $\alpha$ and $\beta$ are any formulas. The set $S_T$ that appears in rules $F\neg$ and $F\rightarrow$ means $\{T\alpha | T\alpha \in S\}$.

Fitting (1969) describes the way to apply the rules in a proof search procedure and establishes the common vocabulary to use in this process. We summarize here the main important concepts to uniformise the descriptions in the following sections. We call a *configuration*, a finite collection of $S_1, S_2, \ldots, S_n$ of sets of signed formulas. By an *application of a rule $\mathcal{R}$ to a configuration* $S_1, S_2, \ldots, S_n$ we mean the replacement of this configuration for a new one, just changing some $S_i, i = 1, \ldots n$ in the original configuration by the result of applying rule $\mathcal{R}$ to $S_i$. A *tableau* is finite sequence of configurations $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_n$, where each configuration (except the first) is the result of applying one of the **Tableaux$_{\mathbf{Int}}$** rules to the preceding configuration. A set $S$ of signed formulas is said *closed* if it contains both $T\alpha$ and $F\alpha$. A configuration is closed if each $S_i$ in it is closed. A tableau is closed if some of its configurations is closed. We call a tableau for a set $S$ a tableau $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_n$ in which $\mathcal{C}_1$ is $\{S\}$. $S$ is said *inconsistent* if some tableau for $S$ is closed, otherwise $S$ is *consistent.* $\alpha$ is a *theorem* if $\{F\alpha\}$ is inconsistent, and a closed tableaux for $F\alpha$ is a *proof* of $\alpha$.

If we change $S_T$ for $S$ in rules $F\neg$ and $F\rightarrow$, we get the Tableaux System for Classical Propositional Logic (**Cla**). We call this system **Tableaux$_{\mathbf{Cla}}$**.

To get a system for the Full Propositional Minimal Logic (**Min**) we use the same set of rules from Figure 5.1, but, in this case, the $\bot$ has no meaning and using the usual form $\neg\alpha = \alpha \rightarrow \bot$ we get the new version of $\neg$-rules below where $\bot$ is a simple atomic formula in the logic and, thus, $T\bot$ is not a contradiction. This system is called **Tableaux$_{\mathbf{Min}}$**. For $\mathbf{M}^{\rightarrow}$, we just need to restrict ourselves to the $\rightarrow$-rules of Figure 5.1.

$$T\neg \quad \frac{S, T(\alpha \rightarrow \bot)}{S, F\alpha | S, T\bot} \qquad\qquad F\neg \quad \frac{S, F(\alpha \rightarrow \bot)}{S_T, T\alpha, F\bot}$$

Fitting (1969) also makes the link between these systems and the appropriate semantics of the respective logic. In the classical system, $T\alpha$ and $F\alpha$ mean $\alpha$ is true, and $\alpha$ is false respectively. That is, if the situation above the line of a rule holds, the situation below the line also holds. Proof search is a refutation procedure: we start by supposing $\alpha$ is not true ($F\alpha$), then, reaching a configuration in which $\alpha$ is both true and false makes us conclude that $\alpha$ has to be true since this situation is not possible. In the intuitionistic and minimal cases, $T\alpha$ means $\alpha$ is proven, and $F\alpha$ means $\alpha$ has not been proved. We can then read the rules as: if the situation above the line is the case, the situation below it is possible, i.e. compatible with our present knowledge. As an example let us consider the case of $F \rightarrow$-rule, that is important for us here interested in $\mathbf{M}^{\rightarrow}$: if $\alpha \rightarrow \beta$ has not been proven, it is possible to prove $\alpha$ without proving $\beta$. Otherwise, a proof of $\beta$ would be directly derived from a proof of $\alpha$, which constitutes a proof of $\alpha \rightarrow \beta$. And we have $S_T$ in the bottom of the rule because in proving $\alpha$ we can review some additional unproven formulas. As the classical version, the system for **Int** and **Min** is also a refutation procedure, since a formula can not be both proven and not proven at the same time.

Thus, following what was described in the aforementioned paragraphs, we have a procedure for proof search using Tableaux Systems for the three propositional logics described here. Let us consider some examples, the first one being in **Cla**. To prove that the formula $\neg\neg(A \vee \neg A)$ is a classical tautology we start with a configuration $\mathcal{C}_1 = \{\{F\neg\neg(A \vee \neg A)\}\}$ and after a successive application of rules we stop with a closed tableau.

$$\begin{gathered}
\{\{F\neg\neg(A \vee \neg A)\}\} \\
\{\{T\neg(A \vee \neg A)\}\} \\
\{\{F(A \vee \neg A)\}\} \\
\{\{FA, F\neg A\}\} \\
\{\{FA, TA\}\}
\end{gathered} \tag{5-1}$$

We call this style of visualization of tableau configurations of *tableaux-as-sets*. To facilitate our description of soundness and completeness of the system, this is the default view that we use in this text. However, the known *tableaux-as-trees* view can also be used. This view is very well described also by Fitting in D'Agostino et al. (2013). From the case of the proof 5-1 we have:

$$\begin{array}{lc}
(1) & F\neg\neg(A \vee \neg A) \\
(2) & T\neg(A \vee \neg A) \\
(3) & F(A \vee \neg A) \\
(4) & FA \\
(5) & F\neg A \\
(6) & TA \\
 & \otimes
\end{array} \tag{5-2}$$

In proof 5-2 the only one branch is closed by steps (4) and (6).

The translation of a tableau for **Cla** to the sequent calculus **LK** is immediate:

- the set inside configuration $\mathcal{C}_1$ of the tableau becomes the root sequent in **LK**, i.e., **LK** proofs follow the tableau configurations, but upside down.
- a configuration $\mathcal{C}_i$ with one set inside becomes the conclusion sequent of an **LK** application of a rule with the sets of $\mathcal{C}_{i+1}$ becoming the premises sequents (one or two) of this rule application.
- inside a set of a configuration, formulas signed with $T$ become left formulas of the translated sequent in **LK** and formulas signed with $F$, the right ones.

This way, for the case of the proof 5-1 we can construct the following **LK** proof:

$$\dfrac{\dfrac{\dfrac{\dfrac{A \Rightarrow A}{\Rightarrow A, \neg A}\,\neg\text{-r}}{\Rightarrow (A \vee \neg A)}\,\vee\text{-r}}{\neg(A \vee \neg A) \Rightarrow}\,\neg\text{-l}}{\Rightarrow \neg\neg(A \vee \neg A)}\,\neg\text{-r}$$

But the formula $\neg\neg(A \vee \neg A)$ is also an intuitionistic tautology. Thus, using the rules of Figure 5.1 we can construct an intuitionistic proof for it. One thing that is not clear in the first look to the rules of Fitting's Tableaux for **Int** or **Min** is that, in the cases of $T\neg$ and $T \rightarrow$ we must repeat the main formula above the line of the rule application in the part below the line. This repetition of hypotheses is necessary to achieve completeness in some logics (as presented in Chapter 2). Therefore, a more precise way to represent those rules would be:

$$T\neg \quad \frac{S, T\alpha \rightarrow \beta}{S, T\alpha \rightarrow \beta, F\alpha | S, T\alpha \rightarrow \beta, T\beta} \qquad\qquad T\neg \quad \frac{S, T\neg\alpha}{S, T\neg\alpha, F\alpha}$$

Thus, the intuitionistic proof of $\neg\neg(A \vee \neg A)$ is presented in the proof 5-3 below, based on the $\textbf{Tableaux}_{\textbf{Int}}$ rules:

$$\begin{gathered}
\{\{F\neg\neg(A \vee \neg A)\}\} \\
\{\{T\neg(A \vee \neg A)\}\} \\
\{\{T\neg(A \vee \neg A), F(A \vee \neg A)\}\} \\
\{\{T\neg(A \vee \neg A), FA, F\neg A\}\} \\
\{\{T\neg(A \vee \neg A), TA\}\} \\
\{\{T\neg(A \vee \neg A), TA, F(A \vee \neg A)\}\} \\
\{\{T\neg(A \vee \neg A), TA, FA, F\neg A\}\}
\end{gathered} \tag{5-3}$$

The translation from $\mathbf{Tableaux_{Int}}$ to **LJ** is very close to the classical translation of $\mathbf{Tableaux_{Cla}}$ to **LK** that we explained above. But there are some changes in the presented approach that we need to do. Because the application of a $T$-rule does not remove $F$-formulas and **LJ** proofs just allow one formula on the right side, we need to decide which of the $F$-formulas in one of the configuration sets to use as the right formula of the translated sequent. Therefore, we have three cases to consider:

– in the case of a $T\neg$ rule applied to a formula $\alpha$, we use $\alpha$ as the right formula of the premise sequent in the translation:

$$T\neg \quad \frac{S, T\neg\alpha}{S, T\neg\alpha, F\alpha} \qquad\qquad \neg - l \quad \frac{\Gamma, \neg\alpha \Rightarrow \alpha}{\Gamma, \neg\alpha \Rightarrow \gamma}$$

where $S = \Gamma \cup \{F\gamma\}$

– in the case of a $T \rightarrow$ rule applied to a formula $\alpha \rightarrow \beta$, we use $\alpha$ as the right formula of the left premise sequent in the translation and the last inserted $F$-formula in $S$ ($\gamma$, in the schema below) in the right premise.

$$T \rightarrow \qquad \frac{S, T\alpha \rightarrow \beta}{S, T\alpha \rightarrow \beta, F\alpha | S, T\alpha \rightarrow \beta, T\beta}$$

$$\rightarrow -l \quad \frac{\Gamma, \alpha \rightarrow \beta \Rightarrow \alpha \qquad \Gamma, \alpha \rightarrow \beta, \beta \Rightarrow \gamma}{\Gamma, \alpha \rightarrow \beta \Rightarrow \gamma}$$

where $S = \Gamma \cup \{F\gamma\}$

– in the case of a $F\vee$ rule applied to a formula $\alpha \vee \beta$, we have to choose which one of the cases to keep in the premise sequent in the translation:

$$T\neg \quad \frac{S, F\alpha \vee \beta}{S, F\alpha, F\beta} \qquad\qquad \vee - r \quad \frac{\Gamma \Rightarrow \alpha}{\Gamma \Rightarrow \alpha \vee \beta}$$

$$\vee - r \quad \frac{\Gamma \Rightarrow \beta}{\Gamma \Rightarrow \alpha \vee \beta}$$

where $S = \Gamma$

Thus, the translated **LJ** proof for the tableau proof 5-3 is:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\neg(A \vee \neg A), A \Rightarrow A}{\neg(A \vee \neg A), A \Rightarrow A \vee \neg A}\ \vee\text{-r}}{\neg(A \vee \neg A), A \Rightarrow}\ \neg\text{-l}}{\neg(A \vee \neg A) \Rightarrow \neg A}\ \neg\text{-r}}{\neg(A \vee \neg A) \Rightarrow A \vee \neg A}\ \vee\text{-r}}{\neg(A \vee \neg A) \Rightarrow}\ \neg\text{-l}}{\Rightarrow \neg\neg(A \vee \neg A)}\ \neg\text{-r}$$

The *tableaux-as-tree* view of the Proof 5-3 is constructed in a slightly different manner than in the classical version. In **Cla** every time a formula is expanded, it can be checked as already used since it will not be expanded anymore. In the case of an Intuitionistic Logic (and also for Minimal Logic) tableaux, trees can grow as well as shrink. Every time an $F$-rule is applied we need to erase the previous $F$-formulas at the same branch, following the rules of Figure 5.1. In the tree view below this happens when expanding the $F$-formula in line (5). At this time, we need to erase the only $F$-formula previously generated (the one in line (4)). Moreover, in tree view for **Int** and **Min** , $T$-formulas are kept unchecked, meaning that they can be used again since it is necessary to the completeness of the system.

$$\begin{array}{lc}
(1) & F\neg\neg(A \vee \neg A) \checkmark \\
(2) & T\neg(A \vee \neg A) \\
(3) & F(A \vee \neg A) \checkmark \\
(4) & \cancel{FA} \\
(5) & F\neg A \checkmark \\
(6) & TA \\
(7) & F(A \vee \neg A) \checkmark \\
(8) & FA \\
(9) & F\neg A \\
 & \otimes
\end{array} \tag{5-4}$$

As already mentioned the case of **Min** is the same of **Int**, only considering that $\perp$ has no meaning, and so, a signed formula $T\perp$ is allowed. The case for $\mathbf{M}^{\rightarrow}$ is the sub-case of **Min** by restricting ourselves to $T \rightarrow$ and $F \rightarrow$ rule.

## 5.2 Soundness and Completeness of Fitting's Tableaux

In this section, we will show the Soundness and Completeness Proofs of $\mathbf{Tableaux_{M^{\rightarrow}}}$. From now we will keep the focus on $\mathbf{M}^{\rightarrow}$ as it is our main objective of study in this text. We will follow the proofs presented in Fitting (1969) for $\mathbf{Tableaux_{Int}}$, adapting them to the specific case of Tableaux for $\mathbf{M}^{\rightarrow}$.

### 5.2.1 Soundness

**Definition 19** *A set $\{T\alpha_1, \ldots, T\alpha_n, F\beta_1, \ldots, F\beta_m\}$ is realizable if there is some model $\langle U, \preceq, \mathcal{V}\rangle$ and some world $w \in U$ such that $w \models \alpha_1, \ldots, w \not\models \alpha_n, w \not\models \beta_1, \ldots, w \not\models \beta_m$. In this case, $w$* realizes *the set. For a configuration $\mathcal{C} = S_1, S_2, \ldots, S_n$, $\mathcal{C}$ is realizable if some $S_i$ is realizable.*

**Theorem 20** *If a configuration $\mathcal{C}_i$ of a tableau $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_n$ is realizable, then the configuration $\mathcal{C}_{i+1}$ is also realizable.*

*Proof*: We have two cases, depending on the rule that produced $\mathcal{C}_{i+1}$:

**Case (a)** $\mathcal{C}_i$ is $\{\ldots, \{S, T(\alpha \to \beta)\}, \ldots\}$ and $\mathcal{C}_{i+1}$ is $\{\ldots, \{S, T(\alpha \to \beta), F\alpha\}, \{S, T(\alpha \to \beta), T\beta\}, \ldots\}$. Supposing $\mathcal{C}_i$ is realizable, then, some $S_j \in \mathcal{C}_i$ is realizable. If $S_j$ is not $\{S, T(\alpha \to \beta)\}$ thus $S_j$ is still realizable in $\mathcal{C}_{i+1}$. If that element is $\{S, T(\alpha \to \beta)\}$, then for some model $\mathcal{M} = \langle U, \preceq, \mathcal{V}\rangle$ and some world $w \in U$, $w$ realizes $\{S, T(\alpha \to \beta)\}$. That is, $w$ realizes $S$ and $w \models \alpha \to \beta$ which means that, for all $v \in U$ such that $w \preceq v$, $v \not\models \alpha$ or $v \models \beta$, so either $w$ realizes $\{S, T(\alpha \to \beta), F\alpha\}$ or $\{S, T(\alpha \to \beta), T\beta\}$.

**Case (b)** $\mathcal{C}_i$ is $\{\ldots, \{S, F(\alpha \to \beta)\}, \ldots\}$ and $\mathcal{C}_{i+1}$ is $\{\ldots, \{S, T(\alpha \to \beta), T\alpha, F\beta\}, \ldots\}$. Supposing $\mathcal{C}_i$ is realizable, for the same reason of case (a) we just need to evaluate the case that $\{S, F(\alpha \to \beta)\}$ is realizable. Then for some model $\mathcal{M} = \langle U, \preceq, \mathcal{V}\rangle$ and some world $w \in U$, $w$ realizes $S$ and $w \not\models \alpha \to \beta$ which means that, there is a $v \in U$ such that $w \preceq v$, $v \models \alpha$ and $v \not\models \beta$. By the Semantics of Kripke models for $\mathbf{M}^{\to}$ presented in Chapter 2 we know that if $w$ realizes $S$ then $v$ also realizes $S_T$. Hence, $v$ realizes $\{S_T, T\alpha, F\beta\}$ and $C_{i+1}$ is realizable.

■

**Corollary 21** $\textbf{Tableaux}_{\mathbf{M}^{\to}}$ *is sound regarding the Semantics of $\mathbf{M}^{\to}$, that is, if $\vdash_{\textbf{Tableaux}_{\mathbf{M}^{\to}}} \alpha$ then $\alpha$ is valid.*

*Proof*: We prove by contrapositive. Suppose $\alpha$ is not valid, so there is a model $\mathcal{M} = \langle U, \preceq, \mathcal{V}\rangle$ and some world $w \in U$ such that $\mathcal{M} \not\models_w \alpha$. This means that $\{F\alpha\}$ is realizable. But a proof of $\alpha$ would be a closed tableau $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_n$ where $\mathcal{C}_1$ is $\{\{F\alpha\}\}$. But, by Theorem 20 we know that if $\mathcal{C}_1$ is realizable, each $\mathcal{C}_i$ is also. A realizable configuration can not be closed, showing the contradiction. Hence, $\nvdash_{\textbf{Tableaux}_{\mathbf{M}^{\to}}} \alpha$. ■

### 5.2.2 Hintikka collections

Here, we are going to present some important definitions to be used in the completeness proof of $\textbf{Tableaux}_{\mathbf{M}^{\rightarrow}}$. The central concept is the definition of a *Hintikka collection* for $\mathbf{M}^{\rightarrow}$. This concept is adapted here from Fitting (1969) that presented Hintikka collections in terms of **Int**. Both cases are inspired by the notion of *Hintikka sets* originally conceived by Hintikka and very developed in Smullyan (1995).

**Definition 22** *Let $\mathcal{U}$ be a collection of consistent sets of signed formulas in $\mathbf{M}^{\rightarrow}$. $\mathcal{U}$ is a* Hintikka collection *for $\mathbf{M}^{\rightarrow}$ if for any set $\Gamma \in \mathcal{U}$ we have:*

- $T(\alpha \rightarrow \beta) \in \Gamma \Rightarrow F\alpha \in \Gamma$ *or* $T\beta \in \Gamma$,
- $F(\alpha \rightarrow \beta) \in \Gamma \Rightarrow$ *for some* $\Delta \in \mathcal{U}$, $\Gamma_T \subseteq \Delta$, $T(\alpha) \in \Delta$, $F(\beta) \in \Delta$.

**Definition 23** *Let $\mathcal{U}$ be a Hintikka collection for $\mathbf{M}^{\rightarrow}$. Let $\mathcal{M} = \langle U, \preceq, \mathcal{V} \rangle$ be a model according with the semantics presented in Chapter 2. $\mathcal{M}$ is a model for $\mathcal{U}$ if:*

1. $\Gamma_T \subseteq \Delta \Rightarrow$ *there are* $w_\Gamma$, $w_\Delta \in U$ *such that* $w_\Gamma \preceq w_\Delta$
2. $T(\alpha) \in \Gamma \Rightarrow w_\Gamma \models \alpha$
   $F(\alpha) \in \Gamma \Rightarrow w_\Gamma \not\models \alpha$.

**Theorem 24** *There is a model for any Hintikka collection for $\mathbf{M}^{\rightarrow}$.*

*Proof*: Let $\mathcal{U}$ be a Hintikka collection. Let $\mathcal{M} = \langle U, \preceq, \mathcal{V} \rangle$ be a model. Let $\Gamma, \Delta \in \mathcal{U}$ and $\Gamma_T \subseteq \Delta$ then define $w_\Gamma, w_\Delta \in U$ and the relation $\preceq$ as $w_\Gamma \preceq w_\Delta$. If $Tp \in \Gamma$, where $p$ is an atomic formula, let $w_\Gamma \models p$. To ensure that item (2) of last definition holds for any formula $\alpha \rightarrow \beta$, we use an induction on the degree of the formula. Thus we have two cases:

$$\begin{aligned} T(\alpha \rightarrow \beta) \in \Gamma &\Rightarrow \forall \Delta \in \mathcal{U}(\Gamma_T \subseteq \Delta \Rightarrow T(\alpha \rightarrow \beta) \in \Delta) \\ &\Rightarrow \forall \Delta \in \mathcal{U}(\Gamma_T \subseteq \Delta \Rightarrow F(\alpha) \in \Delta \text{ or } T(\beta) \in \Delta) \\ &\Rightarrow \forall \Delta \in \mathcal{U}(w_\Gamma \preceq w_\Delta \Rightarrow \Delta \not\models \alpha \text{ or } \Delta \models \beta) \\ &\Rightarrow w_\Gamma \models \alpha \rightarrow \beta, \end{aligned} \tag{5-5}$$

$$\begin{aligned} F(\alpha \to \beta) \in \Gamma \;&\Rightarrow\; \exists \Delta \in \mathcal{U}(\Gamma_T \subseteq \Delta \text{ and } T(\alpha) \in \Delta \text{ and } F(\beta) \in \Delta) \\ &\Rightarrow\; \exists \Delta \in \mathcal{U}(w_\Gamma \preceq w_\Delta \text{ and } \Delta \models \alpha \text{ and } \Delta \nvDash \beta) \qquad (5\text{-}6) \\ &\Rightarrow\; w_\Gamma \nvDash \alpha \to \beta. \end{aligned}$$

■

Then, to show the completeness of $\mathbf{Tableaux_{M^{\to}}}$ we just need to prove that: for an arbitrary formula $\alpha \in \mathbf{M}^{\to}$, if $\nvdash_{\mathbf{Tableaux_{M^{\to}}}} \alpha$ then there is a Hintikka collection $\mathcal{U}$ such that for some $\Gamma \in \mathcal{U}$, $F\alpha \in \Gamma$. That is, there is a model with a world $w_\Gamma$ such that $\Gamma \nvDash \alpha$.

## 5.2.3 Completeness

Here we adapt the original completeness proof for $\mathbf{Tableaux_{Int}}$ presented in Fitting (1969) to the specific case of $\mathbf{M}^{\to}$. We follow the definitions and steps of the proof very closed to the Fitting's version.

**Definition 25** *A* reduced set *$S'$ for a finite, consistent set $S$ of signed formulas of $\mathbf{M}^{\to}$ is defined in the following manner: Let $S_0$ be $S$. Let $S_n$ be a finite, consistent set of signed formulas. Suppose that the rule $T \to$ apply to $S_n$, i.e., $S_n = \{V, T\alpha \to \beta\}$. As $S_n$ is consistent, so clearly either $\{V, T\alpha \to \beta, F\alpha\}$ or $\{V, T\alpha \to \beta, T\beta\}$ is consistent. Let $S_{n+1}$ be $\{V, T\alpha \to \beta, F\alpha\}$ if consistent, otherwise let $S_{n+1}$ be $\{V, T\alpha \to \beta, T\beta\}$. Thus, we define a sequence $S_0, S_1, S_2, \ldots$ with the property that for an element $S_n$ of the sequence, $S_n \subseteq S_{n+1}$. Consider that $\mathcal{P}(S)$ is the collection of all signed subformulas of formulas in $S$. Each $S_n$ is finite and consistent. Hence, $S_n \subseteq \mathcal{P}(S)$. Therefore, there is only a finite number of different possible $S_n$. This way, there must be a member of the sequence, say $S_n$ such that the application of another rule $T \to$ in some of the formulas of $S_n$ produces $S_n$ again. This $S_n$ is the reduced set $S'$ of $S$. If $S'$ is a reduced set, we know that:*

$$T(\alpha \to \beta) \in S' \Rightarrow F\alpha \in S' \text{ or } T\beta \in S'$$

**Observation 26** *Any finite, consistent set of signed formula has a finite, consistent reduced set.*

**Definition 27** *Let $S$ be a finite, consistent set of signed formulas. If $F\alpha \to \beta \in S$ then $\{S_T, T\alpha, F\beta\}$ is an* associated set *of $S$. We say that $\mathcal{A}(S)$ is the collection of all associated sets of $S$.*

**Observation 28** *Consider a set $S$ and its associated set $\mathcal{A}(S)$ according to Definition 27.*

- *Let $U \in \mathcal{A}(S)$, then $U \subseteq \mathcal{P}(S)$. Since $\mathcal{P}(S)$ is finite , $\mathcal{A}(S)$ is also finite.*
- *As $S$ is consistent, any associated set of $S$ is also consistent.*
- *$F\alpha \rightarrow \beta \in S \Rightarrow$ for some $U \in \mathcal{A}(S), S_T \subseteq U, T\alpha \in U, F\beta \in U$.*

**Theorem 29** $\mathbf{Tableaux_{M^{\rightarrow}}}$ *is complete.*

*Proof*: Suppose $\nvdash_{\mathbf{Tableaux_{M^{\rightarrow}}}} \alpha$. Then $\{F\alpha\}$ is consistent. Starting with $\{F\alpha\}$ we create the following sequence:

$S_0$ is the reduced set of $\{F\alpha\}$

then produce $\mathcal{A}(S_0) = \{U_1, U_2, \ldots, U_n\}$

$S_1$ is the reduced set of $U_1$
$S_2$ is the reduced set of $U_2$
$\vdots$
$S_n$ is the reduced set of $U_n$

then produce $\mathcal{A}(S_1) = \{U_{n+1}, U_{n+2}, \ldots, U_m\}$

$S_{n+1}$ is the reduced set of $U_{n+1}$
$S_{n+2}$ is the reduced set of $U_{n+2}$
$\vdots$
$S_m$ is the reduced set of $U_m$

then repeat the process with $S_2$ and so on.

Hence we form the sequence $S_0, S_1, S_2, \ldots$. Since each $S_i \subseteq \mathcal{P}(S)$, there are only finitely many possible different $S_i$. Thus, eventually, we reach a point $S_k$ of the sequence where any continuation repeats a previous produced member of it. Let $\mathcal{U} = \{S_0, S_1, \ldots, S_k\}$. From Definition 22 we can concluded that $\mathcal{U}$ is a Hintikka collection. Hence, there is a model that falsifies $\alpha$ in $\mathbf{M}^{\rightarrow}$.

■

**Corollary 30** *From the proof of Theorem 29 we can also infer that $\mathbf{M}^{\rightarrow}$ is decidable. All the process described above is bound by the size of $\mathcal{P}(S)$. This last is bound by the number of signed subformulas of the first configuration $\{F\alpha\}$, that is, in the end, bound by the degree of $\alpha$, i.e., $|\mathcal{P}(S)| <= 2^{|\alpha|}$.*

Fitting finished the presentation of his completeness proof for **Int** stating that we can extract from the proof a procedure for decidability in that logic. In fact, the aforementioned procedure works as a strategy for rule application in a $\mathbf{Tableaux_{M^{\to}}}$ proof search. For instance, the general method of Tableaux Systems described in the begin of this Chapter only states that with a rule application we move from one configuration to another. But as in $\mathbf{M}^{\to}$ we need to repeat hypothesis to prove some tautologies of the language, this general application of rules can lead us to loops during the proof search. The order forced by the procedure embedded in the Completeness Proof above avoids this problem.

**Example 31** *We follow now this procedure to show that $((A \to B) \to A) \to A$, the Peirce formula, is not a valid formula in $\mathbf{M}^{\to}$.*

*Suppose* $\nvdash_{\mathbf{Tableaux_{M^{\to}}}} ((A \to B) \to A) \to A$*, then* $\{F(((A \to B) \to A) \to A)\}$ *is consistent. Starting with* $\{F(((A \to B) \to A) \to A)\}$ *we need to follow this sequence:*

$S_0$ *is the reduced set of* $\{F(((A \to B) \to A) \to A)\}$

$$\mathcal{A}(S_0) = \{\{T((A \to B) \to A), FA\}\}$$

$S_1$ *is the reduced set of* $\{T((A \to B) \to A), FA\}$

$S_1 = \{T((A \to B) \to A), FA\}$

$S_1 = \{T((A \to B) \to A), FA, F(A \to B)\}$

*(since* $S_1 = \{\{T((A \to B) \to A), FA, TA)\}\}$ *is not consistent)*

$$\mathcal{A}(S_1) = \{\{T((A \to B) \to A), TA, FB\}\}$$

$S_2$ *is the reduced set of* $\{T((A \to B) \to A), TA, FB\}$

$S_2 = \{T((A \to B) \to A), TA, FB\}$

$S_2 = \{T((A \to B) \to A), TA, FB, F(A \to B)\}$

*(since* $S_1 = \{\{T((A \to B) \to A), TA, FB, FA)\}\}$ *is not consistent)*

$$\mathcal{A}(S_2) = \{\{T((A \to B) \to A), TA, FB, TA, FB\}\}$$

$S_3$ *is the reduced set of* $\{T((A \to B) \to A), TA, FB, TA, FB\}$

$S_3 = \{T((A \to B) \to A), TA, FB, TA, FB\}$

$S_3 = \{T((A \to B) \to A), TA, FB, TA, FB, F(A \to B)\}$

*(but here we have the repetition of an already produced $S_i$, thus the previous one becomes the $S_k$)*

*This sequence can be followed during the proof search of* $\mathbf{Tableaux_{M^{\to}}}$ *generating the attempt proof tree presented in* (5-7).

$$S_0$$

$$\{\ \{F((A \to B) \to A) \to A\}\ \}$$

$$\{\{T(A \to B) \to A, FA\}\}$$

$$S_1 \qquad\qquad\qquad\qquad \Gamma_1$$

$$\{\ \{T(A \to B) \to A, FA, F(A \to B)\}\ ,\ \{T(A \to B) \to A, FA, TA\}\ \}$$

$$\{\{T(A \to B) \to A, TA, FB\}, \Gamma_1\}$$

$$S_2 \qquad\qquad\qquad\qquad \Gamma_2$$

$$\{\ \{T(A \to B) \to A, TA, FB, F(A \to B)\}\ ,\ \{T(A \to B), TA, FB, TA\}\ , \Gamma_1\}$$

$$\{\{T(A \to B) \to A, TA, TA, FB\}, \Gamma_2, \Gamma_1\}$$

*Repetition of a previously generated configuration*

$$\{\{T(A \to B) \to A, TA, TA, FB, F(A \to B)\}, \{T(A \to B), TA, TA, FB, TA\}, \Gamma_2, \Gamma_1\} \tag{5-7}$$

*This way, the process stops and we construct the collection* $\mathcal{U} = \{S_0, S_1, S_2\}$ *that is a Hintikka collection. Using the Definition 24 we can generate the following model:*

$$\begin{array}{c} \mathbf{w_{S_0}} \\ \nvDash ((A \to B) \to A) \to A \\ | \\ \mathbf{w_{S_1}} \\ \nvDash A \\ | \\ \mathbf{w_{S_2}} \\ \vDash A \\ \nvDash B \end{array} \tag{5-8}$$

**Example 32** *Another interesting example is the case of the formula* $((A \to B) \vee (B \to A))$*, the Dummett formula, that is not a valid formula in* **Int** *neither in* **Min**. *Therefore, the translated version of this formula for* $\mathbf{M}^{\to}$ *is neither valid. The steps below follow the full procedure described in Fitting (1969) in the Completeness Proof of* $\mathbf{Tableaux_{Int}}$. *The importance of this example is to show a formula that needs a Kripke model with parallel worlds as a counter-model.*

*Suppose* $\nvdash_{\mathbf{Tableaux_{M\to}}} ((A \to B) \vee (B \to A))$, *then* $\{F((A \to B) \vee (B \to A))\}$ *is consistent. Starting with* $\{F((A \to B) \vee (B \to A))\}$ *we need to follow this sequence:*

$S_0$ *is the reduced set of* $\{F((A \to B) \vee (B \to A))\}$
$S_0 = \{F((A \to B) \vee (B \to A)), FA \to B, FB \to A\}$

$$\mathcal{A}(S_0) = \{\{TA, FB\}, \{TB, FB\}\}$$

$S_1$ *is the reduced set of* $\{TA, FB\}$
$S_1 = \{TA, FB\}$

$$\mathcal{A}(S_1) = \{\}$$

$S_2$ *is the reduced set of* $\{TB, FA\}$
$S_2 = \{TB, FA\}$

$$\mathcal{A}(S_2) = \{\}$$

*In this case, the proof search procedure does not produce all the sets of the sequence* $S_0, S_1, S_2$, *which is due to the fact that in the second step of the proof search we need to make a choice for which formula expand, not considering the other one. Although we lose information necessary to build the counter-model, the procedure is still able to confirm that the initial formula is not valid. We can see this in the tableau* (5-9).

$$\begin{array}{c} \{\{F((A \to B) \vee (B \to A))\}\} \\ \{\{F((A \to B) \vee (B \to A)), FA \to B, FB \to A\}\} \\ \{\{TA, FB\}\} \end{array} \tag{5-9}$$

*Using the collection* $\mathcal{U} = \{S_0, S_1, S_2\}$, *that is a Hintikka collection, we can generate the following model:*

$$\begin{tikzcd} & \begin{array}{c}\mathbf{w_{S_0}}\\ \nvDash A \to B\\ \nvDash B \to A\end{array} \arrow[dl, dash] \arrow[dr, dash] & \\ \begin{array}{c}\mathbf{w_{S_1}}\\ \vDash A\\ \nvDash B\end{array} & & \begin{array}{c}\mathbf{w_{S_2}}\\ \vDash B\\ \nvDash A\end{array} \end{tikzcd} \tag{5-10}$$

## 5.3 Comparison between $\mathrm{LMT}^{\rightarrow}$ and Fitting's Tableaux

We use the criteria proposed in Dyckhoff (2016) and discussed here in Chapter 2, Section 2.6 to compare the systems $\mathbf{LMT}^{\rightarrow}$ and $\mathbf{Tableaux}_{\mathbf{M}^{\rightarrow}}$.

### 5.3.1 Termination

$\mathbf{LMT}^{\rightarrow}$ and $\mathbf{Tableaux}_{\mathbf{M}^{\rightarrow}}$ use different approaches to guarantee termination of the proof search procedure. $\mathbf{LMT}^{\rightarrow}$ has an ordered way to generate proofs that are aimed to stress all possible combinations of rule applications (as we presented in Theorem 7 in Chapter 4).

The order established by the strategy extracted from the Completeness Proof of $\mathbf{Tableaux}_{\mathbf{M}^{\rightarrow}}$ (Theorem 29) avoids loops due to hypothesis repetition, but we still need a mechanism to identify the end of the proof search procedure. The suggestion based on the Fitting's Completeness is to check for repetition of a configuration each time an expansion is done. Thus, the direct way to implement this using a *loop checker*.

For an initial formula $\alpha$, the space for proof search in $\mathbf{LMT}^{\rightarrow}$ is bound by $|\alpha| \cdot 2^{|\alpha|+1+2log_2|\alpha|}$ which is established in Theorem 6 in Chapter 4. In the case of $\mathbf{Tableaux}_{\mathbf{M}^{\rightarrow}}$ the proof search space is limited by the size of $\mathcal{P}(\{F\alpha\})$ that is bound by $2^{|\alpha|}$ (as shown in Corollary 30).

### 5.3.2 Bicompleteness

$\mathbf{LMT}^{\rightarrow}$ uses the top sequent of a totally expanded and opened branch to construct a counter-model that, then, by inversibility of the system's rules can be extended as counter-model for the initial formula (as described in Section 4.5 in Chapter 4). The inclusion relation between sets of labeled formulas in this top sequent is used to define the accessibility relation of the counter-model. $\mathbf{Tableaux}_{\mathbf{M}^{\rightarrow}}$ uses a very similar approach: from a full expanded opened branch of a tableau, it is possible to build a Hintikka collection, and, then, we construct the counter-model (see Theorem 29). The accessibility relation in the model is derived from the inclusion relation of true formulas in sets of the Hintikka collection. The main difference between the two strategies is that $\mathbf{Tableaux}_{\mathbf{M}^{\rightarrow}}$ uses information spread through the branch to generate the associated Hintikka collection while $\mathbf{LMT}^{\rightarrow}$ accumulates this information on the top sequent of the opened branch.

Example 4-8 is the one generated for the Peirce formula from $\mathbf{LMT}^{\rightarrow}$ and Example 5-8 from $\mathbf{Tableaux}_{\mathbf{M}^{\rightarrow}}$.

### 5.3.3 Determinism

$\mathbf{LMT}^{\rightarrow}$ avoids the use of backtracking techniques keeping the proof attempt history locally in each sequent of the proof search tree. Its restart rule is used to go back to a point that was not yet fully explored but in a forward manner.

To explain the need for some backtracking mechanism in the case of $\mathbf{Tableaux}_{\mathbf{M}^{\rightarrow}}$ we use the proof fragment (5-11) that shows the first part of a tableau for the formula $((((A \rightarrow C) \rightarrow A) \rightarrow A) \rightarrow C) \rightarrow C$ that we know from Dowek & Jiang (2006) that needs to use the assumption $(((A \rightarrow C) \rightarrow A) \rightarrow A) \rightarrow C$ at least twice to be proved in $\mathbf{M}^{\rightarrow}$. As presented in Haeusler (2015*a*) this formula is used as the base to define a family of formulas in $\mathbf{M}^{\rightarrow}$ with no bounds on the use of assumptions. We numbered each configuration of the tableau to refer to them. In step (6) we reach a configuration where the first set has two $F$-formulas in it. So, we need to create two associated sets, one for each formula. But, following a forward application of rules, the choice for one $F$-formula will eliminate the other, forcing us to mark this part of the proof to come back after full expansion of the first chosen $F$-formula.

### 5.3.4 Simplicity

According to Dyckhoff (2016), simplicity means "to allow easier reasoning about systems". In this sense, $\mathbf{LMT}^{\rightarrow}$ has a non-standard sequent definition (comparing with **LJ** style of sequents) while the signed formulas in $\mathbf{Tableaux}_{\mathbf{M}^{\rightarrow}}$ is a very known technique. To keep the rule applications always forward in the proof search procedure, $\mathbf{LMT}^{\rightarrow}$ needs the restart rule and its labeling system that also add complexity to reader's understanding of generated proofs. However, in contrast, with an aim in automatic reasoning, $\mathbf{LMT}^{\rightarrow}$ becomes a more practical solution as a unified procedure for both, provability and counter-model generation (although, $\mathbf{Tableaux}_{\mathbf{M}^{\rightarrow}}$ also has this feature), without the need of loop checkers (although its use can shorten the space to search for proofs).

$$
\begin{array}{ll}
(1) & \{\overset{S_0}{\{F(((((A \to C) \to A) \to A) \to C) \to C)\}}\} \\
(2) & \{\{T((((A \to C) \to A) \to A) \to C), FC\}\} \\
(3) & \{\overset{S_1}{\{((((A \to C) \to A) \to A) \to C), FC, F(((A \to C) \to A) \to A\}}, \overset{\Gamma_1}{\{T((((A \to C) \to A) \to A) \to C), FC, TC\}}\} \\
(4) & \{\{T((((A \to C) \to A) \to A) \to C), T(((A \to C) \to A), FA\}, \Gamma_1\} \\
(5) & \{T((((A \to C) \to A) \to A) \to C), T((A \to C) \to A), FA, F(A \to C)\}, \overset{\Gamma_2}{\{T((((A \to C) \to A) \to A) \to C), T((A \to C) \to A), FA, TA\}}, \Gamma_1\} \\
(6) & \{\overset{S_2}{\{T((((A \to C) \to A) \to A) \to C), T((A \to C) \to A), FA, F(A \to C), F(((A \to C) \to A) \to A\}}, \{T((((A \to C) \to A) \to A) \to C), T((A \to C) \to A), FA, F(A \to C), TC\}, \Gamma_2, \Gamma_1\} \\
 & \vdots
\end{array}
\tag{5-11}
$$

# 6
# Checking $\mathbf{LMT}^{\rightarrow}$ Proofs with Dedukti

## 6.1
## What is Dedukti

Dedukti is a "universal proof checker, based on the $\lambda\Pi$-calculus modulo", as stated in its maintenance website (`http://dedukti.gforge.inria.fr`). Assaf et al. (2016) describe both, the $\lambda\Pi$-calculus modulo theory and Dedukti as its implementation. The abstract of this paper summarize very well what

> The $\lambda\Pi$-calculus modulo theory, implemented in the Dedukti system, is a logical framework where many theories can be expressed: constructive and classical predicate logic, Simple Type Theory, programming languages, Pure type systems, the Calculus of Inductive Constructions with universes... This feature allows to use it to check large libraries of proofs coming from various proof systems: Zenon, iProver, FoCaLiZe, HOL-Light, and Matita.

$\lambda\Pi$-calculus modulo theory can be seen as a variant of Martin Löf's Intuitionistic Type Theory (Martin-Löf & Sambin, 1984) extended with *dependent types* and *rewrite*, which allows $\lambda\Pi$-calculus modulo to express so many different theories and systems. As a logical framework, $\lambda\Pi$-calculus modulo theory provides a universal definition of fundamental concepts of Logic (formulas, connectives, proofs, models, soundness, completeness, and so on) in a *once and only once* manner in the description of those theories and systems.

Dedukti, the implementation of $\lambda\Pi$-calculus modulo, is a proof checker, aimed to check proofs generated by different automated and semi-automated provers. With more systems being ported to Dedukti "we can start imagining a single library of formal proofs expressed in different theories, developed in different systems, but formulated in a single framework" (Assaf et al., 2016). Hence, we decided to use Dedukti to check proofs generated in our prover, the implementation of our calculus $\mathbf{LMT}^{\rightarrow}$.

The process to translate a system to Dedukti is done by instrumenting the system to be translated (let us call it System B) allowing it to be able to generate a version (a file) of its proofs in the language of Dedukti (i.e., in the logical framework syntax). This translation of a proof of System B works as a certificate that Dedukti will process to check correctness. Thus, the first part of this process is to design carefully the translation of System B to Dedukti syntax using $\lambda\Pi$-calculus modulo theory as the foundation.

Dedukti is developed by the Deducteam, an INRIA research team of the *Programs, Verification, and Proofs* theme, located in France.

In this section, we present the step by step translation of $\mathbf{LMT}^{\rightarrow}$ proofs into Dedukti.

## 6.2 Translation of $\mathrm{LMT}^{\rightarrow}$ Proofs into Dedukti

To allow Dedukti to be used as the proof checker for $\mathbf{LMT}^{\rightarrow}$, we need to convert $\mathbf{LMT}^{\rightarrow}$ proofs into a form that Dedukti already knows how to deal with. In Assaf et al. (2016), authors present a technique to encode a Natural Deduction (ND) system for $\mathbf{M}^{\rightarrow}$ into Dedukti. Our approach here is to use this already known translation of ND into Dedukti. First, we need to convert $\mathbf{LMT}^{\rightarrow}$ proofs into this ND system. Then, we can use the encoding of ND into Dedukti to allow Dedukti to check $\mathbf{LMT}^{\rightarrow}$ proofs. In the next section, we present a strategy [1] for translations of proofs produced by $\mathbf{LMT}^{\rightarrow}$ into ND.

Before start presenting the translation, we need to highlight some characteristics of ND and Dedukti that need to be considered during the definition of the translation strategy:

– Standard ND does not handle the bracket's areas of $\mathbf{LMT}^{\rightarrow}$.

– Standard ND for $\mathbf{M}^{\rightarrow}$ just allows one formula on the right side of a sequent.

– A sequent in $\mathbf{LMT}^{\rightarrow}$ is implemented as a bag having multiples bags on the left and a sequence of atomic formulas (the []-area) and a formula on the right side. The ND encoded in Dedukti uses the structure of Dedukti sequents, that is a bag of lists on the left and a single formula on the right.

[1]This strategy was first presented by Frédéric Gilbert from INRIA in the translation of Zenon theorem prover for Dedukti and proposed by him to the translation of $\mathbf{LMT}^{\rightarrow}$ to Dedukti.

## Steps

1. **Dealing with bag of lists**

   Considering the general form of the sequent in $\mathbf{LMT}^{\rightarrow}$, presented in Chapter 4 and reproduced below:

$$\{\Delta'\}, \Upsilon_1^{p_1}, \Upsilon_2^{p_2}, ..., \Upsilon_n^{p_n}, \Delta \Rightarrow [p_1, p_2, ..., p_n], \varphi$$

   We change the structure of the sequent in the following manner:

   – $\Delta, \Delta', \Upsilon_1^{p_1}, \Upsilon_2^{p_2}, ..., \Upsilon_n^{p_n}$ become lists of formulas in the form $\alpha_n, \alpha_{n-1}, \ldots, \alpha_1$, where $\alpha_1$ is the head of the list.
   – In $\Delta'$, we control the list to avoid repetition, thus it acts as a sequence that does not allow repetition.
   – $p_1, p_2, ..., p_n$ becomes a queue where $p_1$ is the front and $p_n$ is the back.

Figure 6.1 represents the adapted set of rules of $\mathbf{LMT}^{\rightarrow}$ considering those changes.

$$\dfrac{}{\{\Delta'_1, q, \Delta'_2\}, \Upsilon_1^{p_1}, \Upsilon_2^{p_2}, \ldots, \Upsilon_n^{p_n}, \Delta \Rightarrow [p_1, p_2, \ldots, p_n], q}\ axiom$$

$$\dfrac{\{\Delta', \alpha\}, \Upsilon_1^{p_1}, \Upsilon_2^{p_2}, \ldots, \Upsilon_n^{p_n}, \Delta, \alpha \Rightarrow [p_1, p_2, \ldots, p_n], \beta}{\{\Delta'\}, \Upsilon_1^{p_1}, \Upsilon_2^{p_2}, \ldots, \Upsilon_n^{p_n}, \Delta, \alpha \Rightarrow [p_1, p_2, \ldots, p_n], \beta}\ \text{focus}_{\alpha}$$

$$\dfrac{\{\}, \Upsilon_1, \Upsilon_2^{p_2}, \ldots, \Upsilon_n^{p_n}, \Delta^q \Rightarrow [p_2, \ldots, p_n, q], p_1}{\{\Delta'\}, \Upsilon_1^{p_1}, \Upsilon_2^{p_2}, \ldots, \Upsilon_n^{p_n}, \Delta \Rightarrow [p_1, p_2, \ldots, p_n], q}\ \text{restart}_{p_i}$$

$$\dfrac{\{\Delta'\}, \Upsilon_1^{p_1}, \Upsilon_2^{p_2}, \ldots, \Upsilon_n^{p_n}, \Delta, \alpha \Rightarrow [p_1, p_2, \ldots, p_n], \beta}{\{\Delta'\}, \Upsilon_1^{p_1}, \Upsilon_2^{p_2}, \ldots, \Upsilon_n^{p_n}, \Delta \Rightarrow [p_1, p_2, \ldots, p_n], \alpha \rightarrow \beta}\ \rightarrow\text{-right}_{\alpha\rightarrow\beta}$$

$$\dfrac{\{\Delta'_1, \alpha \rightarrow \beta, \Delta'_2\}, \overline{\Upsilon}, \Delta^q, \Delta \Rightarrow [\bar{p}, q], \alpha \qquad \{\Delta'_1, \alpha \rightarrow \beta, \Delta'_2\}, \overline{\Upsilon}, \Delta, \beta \Rightarrow [\bar{p}], q}{\{\Delta'_1, \alpha \rightarrow \beta, \Delta'_2\}\overline{\Upsilon}, \Delta \Rightarrow [\bar{p}], q}\ \rightarrow\text{-left}_{\alpha\rightarrow\beta}$$

$$\text{where } \overline{\Upsilon} = \bigcup_{i=1}^{n} \Upsilon_i^{p_i} \text{ and } \bar{p} = p_1, p_2, \ldots, p_n.$$

Figure 6.1: Rules of $\mathbf{LMT}^{\rightarrow}$ adapted to Dedukti data structure

2. **Mixing syntax and semantic to avoid the usage of brackets**
   In Chapter 4 we described some properties of the sequent of $\mathbf{LMT}^{\rightarrow}$:

   – A sequent of $\mathbf{LMT}^{\rightarrow}$ in the aforementioned general form is *valid* if and only if, $\Delta', \Delta \models \varphi$ or $\exists i(\bigcup_{k=1}^{i} \Upsilon_k^{p_k}) \models p_i$, for $i = 1, \ldots n$.

   – And from Observations 4.iii and 4.iv we have:

     – In every sequent of the tree, $\Delta' \subseteq \Delta$.
     – For $i = 1, \ldots n$, $\Upsilon_{i-1}^{p_{i-1}} \subseteq \Upsilon_i^{p_i}$.

   Based on these concepts we produce an interpretation of the general sequent of $\mathbf{LMT}^{\rightarrow}$ indexed by a formula $G$, say it, $\Big\| \;\; \Big\|_G$:

$$\Big\| \{\Delta'\}, \overline{\Upsilon_i^{p_i}}, \Delta \Rightarrow_{LMT} [\overline{p_i}], \varphi \Big\|_G = (\Delta \rightarrow \varphi) \rightarrow G, \Big((\Upsilon_i^{p_i} \rightarrow p_i) \rightarrow G\Big)_i^n \Rightarrow_{ND} G$$

   where a formula in the notation $\Gamma \rightarrow \beta$ is an abbreviation for $\gamma_1, \gamma_2, \ldots, \gamma_k \rightarrow \beta$, for some $k >= 0$, that means that $\gamma_1 \rightarrow (\gamma_2 \rightarrow (\ldots \rightarrow (\gamma_k \rightarrow \beta)))$.

3. **Translating rules**
   Provide an interpretation of rules under $\Big\| \;\; \Big\|_G$ in Natural Deduction. That is, for each rule $R$ of $\mathbf{LMT}^{\rightarrow}$, construct a proof $\Pi$ in Natural Deduction such that it has the premises and the conclusion of $R$ translated to $\Big\| \;\; \Big\|_G$ as hypotheses and conclusion, respectively.

$$\frac{S_1 \quad \ldots \quad S_n}{S}\,R \qquad \rhd \qquad \begin{array}{c} \Big\| S_1 \Big\|_G \quad \ldots \quad \Big\| S_n \Big\|_G \\ \prod \; ND \text{ tree for } R \\ \Big\| S \Big\|_G \end{array}$$

   – Premises and conclusions of focus, restart and $\rightarrow$-right rules are, basically, identities in this translation.
   – Axiom and $\rightarrow$-left are analyzed in more details in the next section.

4. **Translating a proof**
   Translate of $\overset{\mathcal{D}}{\{\} \Rightarrow [\,], \varphi}$, a proof in $\mathbf{LMT}^{\rightarrow}$ to ND:

   – By induction, we can translate a proof of a formula $\varphi$ in $\mathbf{LMT}^{\rightarrow}$ by a proof in ND of the sequent $\varphi \rightarrow G \Rightarrow G$

– Using $\varphi$ as $G$ we have the sequent $\varphi \to \varphi \Rightarrow \varphi$, that can be proved in ND using the following schema:

$$\dfrac{\dfrac{\varphi \to \varphi \Rightarrow \varphi}{\Rightarrow (\varphi \to \varphi) \to \varphi}{\to}\mathrm{I} \qquad \dfrac{\varphi \Rightarrow \varphi}{\Rightarrow \varphi \to \varphi}{\to}\mathrm{I}}{\Rightarrow \varphi}{\to}\mathrm{E}$$

## 6.3 Translating Rules of $\mathrm{LMT}^{\to}$ to Dedukti

Following the translation strategy presented in the last section, we will translate the axiom and each rule of $\mathbf{LMT}^{\to}$ to the Natural Deduction implementation of Dedukti.

First, consider the following abbreviations:

$$F_i = \Upsilon_i^{p_i} \to p_i$$
$$\overrightarrow{F} = \Big((\Upsilon_i^{p_i} \to p_i) \to G\Big)_i^n, \text{ for } i = 1, \ldots, n$$

**Axiom**

The translated version of the $\mathbf{LMT}^{\to}$ axiom in the ND system of Dedukti using the interpretation $\Big\| \ \Big\|_G$ is a $\prod_{axiom}$ proof as follows:

$$\begin{array}{c} \prod_{axiom} \\ (\Delta_1', q, \Delta_2') \to q \to G, \overrightarrow{F} \Rightarrow G \end{array}$$

To facilitate readability of the proof, consider the following abbreviation of formulas:

$$A \equiv (\Delta_1', q, \Delta_2') \to q,$$
$$\Delta_1' \equiv \alpha_1, \ldots, \alpha_n$$
$$\Delta_2' \equiv \beta_1, \ldots, \beta_m$$

The ND proof of the translated axiom becomes:

$$\dfrac{\dfrac{}{A \to G, \overrightarrow{F} \Rightarrow A \to G} \qquad \dfrac{\begin{array}{c}\dfrac{}{A \to G, \overrightarrow{F}, \alpha_1, \ldots, \alpha_n, q, \beta_1, \ldots, \beta_m \Rightarrow q} \\ \vdots \\ A \to G, \overrightarrow{F} \Rightarrow (\alpha_1 \to (\ldots \to (\alpha_n \to (q \to (\beta_1 \to (\ldots \to (\beta_m \to q)))))))\end{array}}{A \to G, \overrightarrow{F} \Rightarrow (\Delta_1, q, \Delta_2) \to q}}{A \to G, \overrightarrow{F} \Rightarrow G}$$

The cases of focus, restart and →-right rules are trivial, as premises and conclusions of each of these rules are exactly the same. The application of these rules only changes the form of the sequent:

**Focus**

$$\frac{((\Delta,\alpha)\to\beta)\to G,\overrightarrow{F}\Rightarrow G}{((\Delta,\alpha)\to\beta)\to G,\overrightarrow{F}\Rightarrow G}\,{\scriptstyle focus}$$

**Restart**

$$\frac{(\Upsilon_1^{p_1}\to p_1)\to G,\Big(F_i\to G\Big)_{i=2}^{n},\Delta^q\to(q\to G)\Rightarrow G}{\Delta\to(q\to G),\overrightarrow{F}\Rightarrow G}\,{\scriptstyle restart}$$

**→-Right**

$$\frac{((\Delta,\alpha)\to\beta)\to G,\overrightarrow{F}\Rightarrow G}{(\Delta\to(\alpha\to\beta))\to G,\overrightarrow{F}\Rightarrow G}\,{\scriptstyle \to-right}$$

**→-Left**

Consider the following abbreviations:

$$\begin{aligned}C &\equiv (\Delta_1,\alpha\to\beta,\Delta_2)\to q\\ D &\equiv (\Delta_1,\alpha\to\beta,\Delta_2)\to\alpha\\ E &\equiv (\Delta_1,\alpha\to\beta,\Delta_2,\beta)\to q\\ \Gamma &\equiv \overrightarrow{F}\end{aligned}$$

The translated version of the $\mathbf{LMT}^{\to}$→-left rule in the ND system of Dedukti using the interpretation $\Big\|\ \Big\|_G$ becomes:

$$\begin{array}{c} D\to G,\Gamma,C\to G\Rightarrow G \quad E\to G,\Gamma\Rightarrow G\\ \prod{}_{\to-left}\\ C\to G,\Gamma\Rightarrow G\end{array}$$

$\prod_{\to-left}$ can be expanded as follows:

$$\dfrac{\dfrac{C \to G, \Gamma, D \to G \Rightarrow G}{C \to G, \Gamma \Rightarrow (D \to G) \to G} \qquad \dfrac{\dfrac{\dfrac{\dfrac{\Gamma, E \to G \Rightarrow G}{D, C \to G, \Gamma, E \to G \Rightarrow G}}{D, C \to G, \Gamma \Rightarrow (E \to G) \to G} \qquad \dfrac{\dfrac{\begin{array}{c}\prod_1 \\ D, C \to G, \Gamma, E \Rightarrow C\end{array} \quad \dfrac{}{D, C \to G, \Gamma, E \Rightarrow C \to G}}{D, C \to G, \Gamma, E \Rightarrow G}}{D, C \to G, \Gamma \Rightarrow E \to G}}{D, C \to G, \Gamma \Rightarrow G}}{C \to G, \Gamma \Rightarrow D \to G}}{C \to G, \Gamma \Rightarrow G}$$

The fragment $\prod_1$ of $\prod_{\to-left}$ can be expanded as follows, considering that $\Delta_1 \equiv \alpha_1, \ldots, \alpha_n$ and $\Delta_2 \equiv \beta_1, \ldots, \beta_m$ and $\Delta \equiv \Delta_1, \alpha \to \beta, \Delta_2$:

$$\dfrac{\dfrac{\begin{array}{c}\dfrac{\dfrac{}{D, \Delta \Rightarrow \beta_m} \qquad \dfrac{\dfrac{}{D, \Delta \Rightarrow \beta_{m-1}} \quad \begin{array}{c} D, \Delta \Rightarrow D \\ \vdots \\ D, \Delta \Rightarrow \beta_{m-1} \to (\beta_m \to \alpha)\end{array}}{D, \Delta \Rightarrow \beta_m \to \alpha}}{D, \Delta \Rightarrow \alpha} \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{E, \Delta, \alpha \Rightarrow \alpha} \quad \dfrac{}{E, \Delta, \alpha \Rightarrow \alpha \to \beta}}{E, \Delta, \alpha \Rightarrow \beta} \qquad \dfrac{\dfrac{}{E, \Delta, \alpha \Rightarrow \beta_m} \quad \begin{array}{c} E, \Delta, \alpha \Rightarrow E \\ \vdots \\ E, \Delta, \alpha \Rightarrow \beta_m \to (\beta \to q)\end{array}}{E, \Delta, \alpha \Rightarrow \beta \to q}}{E, \Delta, \alpha \Rightarrow q}}{E, \Delta \Rightarrow \alpha \to q} \\ \hline D, E, \Delta \Rightarrow q \\ \hline D, E, \Delta_1, \alpha \to \beta, \Delta_2 \Rightarrow q \\ \hline\hline D, E \Rightarrow (\Delta_1, \alpha \to \beta, \Delta_2) \to q \end{array}}{D, E \Rightarrow C}}{D, C \to G, \Gamma, E \Rightarrow C}$$

## 6.4 A Translation Example

As an example of the application of this strategy, we can see the proof of the tautology $(q \to_0 p) \to_2 (q \to_1 p)$ in $\mathbf{LMT}^{\to}$ and its correspondent translated proof in the Natural Deduction system supported by Dedukti. We numbered the subformulas to facilitate understanding of the translation steps.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\{(q \to_0 p), q\}, ((q \to_0 p))^p, (q)^p, (q \to_0 p), q \Rightarrow_{LMT} [p], q \qquad \dfrac{\{(q \to_0 p), q, p\}, (q \to_0 p), q, p \Rightarrow [\,], p}{\{(q \to_0 p), q\}, (q \to_0 p), q, p \Rightarrow_{LMT} [\,], p}\,\text{focus}_p}{\{(q \to_0 p), q\}, (q \to_0 p), q \Rightarrow_{LMT} [\,], p}\,\to \text{left}_{\to_0}}{\{(q \to_0 p)\}, (q \to_0 p), q \Rightarrow_{LMT} [\,], p}\,\text{focus}_q}{\{\}, (q \to_0 p), q \Rightarrow_{LMT} [\,], p}\,\text{focus}_{\to_0}}{\{\}, (q \to_0 p) \Rightarrow_{LMT} [\,], (q \to_1 p)}\,\to \text{right}_{\to_1}$$

$$\dfrac{\{\}, (q \to_0 p) \Rightarrow_{LMT} [\,], (q \to_1 p)}{\{\} \Rightarrow_{LMT}, [\,], ((q \to_0 p) \to_2 (q \to_1 p))}\,\to \text{right}_{\to_2}$$

$$\nabla$$

$$\overset{\prod \text{axiom}_q}{((q \to_0 p, q) \to q) \to G, ((q \to_0 p, q) \to p) \to G \Rightarrow_{ND} G} \quad \overset{\prod \text{axiom}_p}{((q \to_0 p, q, p) \to p) \to G \Rightarrow_{ND} G}$$

$$\prod \to \text{left}_{\to_0}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{((q \to_0 p, q) \to p) \to G \Rightarrow_{ND} G}{((q \to_0 p, q) \to p) \to G \Rightarrow_{ND} G}\,\text{focus}-_q}{((q \to_0 p, q) \to p) \to G \Rightarrow_{ND} G}\,\text{focus}-_{\to_0}}{((q \to_0 p) \to (q \to_1 p)) \to G \Rightarrow_{ND} G}\,\to \text{right}_{\to_1}}{((q \to_0 p) \to_2 (q \to_1 p)) \to G \Rightarrow_{ND} G}\,\to \text{right}_{\to_2}$$

$\prod_{\to - left_0}$, $\prod_{axiom_p}$ and $\prod_{axiom_q}$ are obtained following the translations maps presented in Section 6.3. To complete the proof we need to perform the final step shown in Section 6.2.

# 7 Conclusion and Future Work

We presented here the definition of a sequent calculus for proof search in the context of the Propositional Minimal Implicational Logic ($\mathbf{M}^{\rightarrow}$). Our calculus, called $\mathbf{LMT}^{\rightarrow}$, aims to proceed the proof search of $\mathbf{M}^{\rightarrow}$ formulas in a bottom-up, forward-always approach. Termination of the proof search is achieved without using loop checkers during the process. $\mathbf{LMT}^{\rightarrow}$ is a deterministic process, which means that the system does not need an explicit backtracking mechanism to be complete. In this sense, $\mathbf{LMT}^{\rightarrow}$ is a bicomplete process, generating Kripke counter-models from search trees produced by unsuccess proving processes.

In the definition of the calculus, we also presented some translations between deductive systems for $\mathbf{M}^{\rightarrow}$: ND to $\mathbf{LJ}^{\rightarrow}$ and $\mathbf{LJ}^{\rightarrow}$ to $\mathbf{LMT}^{\rightarrow}$. We also established a relation between $\mathbf{LMT}^{\rightarrow}$ and Fitting's Tableaux Systems for $\mathbf{M}^{\rightarrow}$regards the counter-model generation in those systems.

We keep the development of a theorem prover, built in Lua (`https://github.com/jeffsantos/GraphProver`), and translated to the Dedukti proof checker, in such a way that proofs generated by the system can be checked in a very standard tool.

As future work, we can enumerate some important features to be developed or extended in the system as well as some new research topics that can be initialized.

- **Precise upper bound for termination** The upper bound used here for achieving termination in $\mathbf{LMT}^{\rightarrow}$ is a very high bound. Many non-theorems can be identified in a small number of steps. We can still explore options to shorten the size of the proof search tree. Even in the case of theorems, our labeling mechanism in conjunction with the usage of the restart rule produces many repetitions in the proof tree.
- **Compression and sharing** Following the techniques proposed by Gordeev & Haeusler (2016) we can explore new ways to shorten the size of proofs generated by $\mathbf{LMT}^{\rightarrow}$.
- **Minimal counter-models** The size of the generated counter-model in $\mathbf{LMT}^{\rightarrow}$ still takes into account every possible combination of subformu-

las, yielding Kripke models with quite a lot worlds. There are still work to be done in order to produce smaller models. Stoughton (1996) presents an implementation of the systems in Dyckhoff (1992) and in Pinto & Dyckhoff (1995) with the property of “minimally sized, normal natural deduction of the sequent, or it finds a ”small”, tree-based Kripke countermodel of the sequent” using the words of the author. This reference can be a good start point to improve $\mathbf{LMT}^{\rightarrow}$ counter-model generation.

– **Dedukti translation implementation** The translation to Dedukti needs to be fully implemented and incorporated into the prover implementation code.

# Bibliography

Andreoli, J.-M. (1992), 'Logic programming with focusing proofs in linear logic', *Journal of Logic and Computation* **2**(3), 297–347.

Assaf, A., Burel, G., Cauderlier, R., Delahaye, D., Dowek, G., Dubois, C., Gilbert, F., Halmagrand, P., Hermant, O. & Saillard, R. (2016), 'Expressing theories in the $\lambda\pi$-calculus modulo theory and in the dedukti system'.

D'Agostino, M., Gabbay, D. M., Hähnle, R. & Posegga, J. (2013), *Handbook of tableau methods*, Springer Science & Business Media.

Dowek, G. & Jiang, Y. (2006), 'Eigenvariables, bracketing and the decidability of positive minimal predicate logic', *Theoretical Computer Science* **360**(1), 193–208.

Dyckhoff, R. (1992), 'Contraction-free sequent calculi for intuitionistic logic', *The Journal of Symbolic Logic* **57**(03), 795–807.

Dyckhoff, R. (2016), Intuitionistic decision procedures since gentzen, *in* 'Advances in Proof Theory', Springer, pp. 245–267.

Dyckhoff, R. & Lengrand, S. (2006), LJQ: a strongly focused calculus for intuitionistic logic, *in* 'Logical Approaches to Computational Barriers', Springer, pp. 173–185.

Ferrari, M., Fiorentini, C. & Fiorino, G. (2013), 'Contraction-free linear depth sequent calculi for intuitionistic propositional logic with the subformula property and minimal depth counter-models', *Journal of automated reasoning* **51**(2), 129–149.

Fitting, M. C. (1969), *Intuitionistic logic, model theory and forcing*, North-Holland Amsterdam.

Gentzen, G. (1935), 'Untersuchungen über das logische schließen. i', *Mathematische zeitschrift* **39**(1), 176–210.

Gordeev, L. & Haeusler, E. H. (2016), 'NP vs PSPACE', *arXiv preprint arXiv:1609.09562.*

Haeusler, E. H. (2015*a*), 'How many times do we need an assumption to prove a tautology in minimal logic? examples on the compression power of classical reasoning', *Electronic Notes in Theoretical Computer Science* **315**, 31–46.

Haeusler, E. H. (2015*b*), 'Propositional logics complexity and the sub-formula property', *Electronic Proceedings in Theoretical Computer Science* **179**, 1–16. Proceedings of DCM2014, Vienna, 2014.

Herbelin, H. (1995), A $\lambda$-calculus structure isomorphic to Gentzen-style sequent calculus structure, *in* 'Computer Science Logic', Springer, pp. 61–75.

Heuerding, A., Seyfried, M. & Zimmermann, H. (1996), Efficient loop-check for backward proof search in some non-classical propositional logics, *in* 'Theorem Proving with Analytic Tableaux and Related Methods', Springer, pp. 210–225.

Hirokawa, S. (1991), 'Number of proofs for implicational formulas', *Introduction to Mathematical Analysis (in Japanese)* **772**, 72–74.

Howe, J. M. (1997), Two loop detection mechanisms: a comparison, *in* 'Automated Reasoning with Analytic Tableaux and Related Methods', Springer, pp. 188–200.

Hudelmaier, J. (1993), 'An O(n log n)-space decision procedure for intuitionistic propositional logic', *Journal of Logic and Computation* **3**(1), 63–75.

Jaśkowski, S. (1934), *On the rules of suppositions in formal logic*, Seminarjum Filozoficzne. Wydz. Matematyczno-Przyrodniczy UW.

Liang, C. & Miller, D. (2007), Focusing and polarization in intuitionistic logic, *in* 'Computer Science Logic', Springer, pp. 451–465.

Martin-Löf, P. & Sambin, G. (1984), *Intuitionistic type theory*, Vol. 9, Bibliopolis Napoli.

Pinto, L. & Dyckhoff, R. (1995), Loop-free construction of counter-models for intuitionistic propositional logic, *in* 'Symposia Gaussiana, Conf A', Walter de Gruyter & Co (Berlin), pp. 225–232.

Prawitz, D. (2006), *Natural deduction: A proof-theoretical study*, Courier Dover Publications.

Seldin, J. P. (1989), ‘Normalization and excluded middle. i’, *Studia Logica* **48**(2), 193–217.

Seldin, J. P. (1998), ‘Manipulating proofs’.

Smullyan, R. M. (1995), *First-order logic*, Courier Corporation.

Statman, R. (1974), Structural Complexity of Proofs, PhD thesis, Stanford University.

Stoughton, A. (1996), Porgi: a proof-or-refutation generator for intuitionistic propositional logic, *in* ‘CADE Workshop on Proof-search in Type-theoretic Languages’, pp. 109–116.

Takeuti, G. (2013), *Proof theory*, Vol. 81, Courier Corporation.

Underwood, J. (1990), A constructive completeness proof for intuitionistic propositional calculus, Technical report, Cornell University.

Vorob’ev, N. N. (1970), ‘A new algorithm for derivability in the constructive propositional calculus’, *American Mathematical Society Translations* **94**(2), 37–71.

Weich, K. (1998), Decision procedures for intuitionistic propositional logic by program extraction, *in* ‘Automated Reasoning with Analytic Tableaux and Related Methods’, Springer, pp. 292–306.