This article was downloaded by: [University Degli Studi Dell'Insurbria], [Mauro Ferrari]
On: 14 April 2012, At: 01:24
Publisher: Taylor & Francis
Informa Ltd Registered in England and Wales Registered Number: 1072954 Registered office: Mortimer House, 37-41 Mortimer Street, London W1T 3JH, UK

Journal of Applied Non-Classical Logics
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/tncl20

A tableau calculus for Propositional Intuitionistic Logic with a refined treatment of nested implications

Mauro Ferrari [a], Camillo Fiorentini [b] & Guido Fiorino [c]

[a] Dipartimento di Informatica e Comunicazione, Università degli Studi dell'Insubria, Via Mazzini 5, 21100, Varese, Italy

[b] Dipartimento di Scienze dell'Informazione, Università degli Studi di Milano, Via Comelico 39, 20135, Milano, Italy

[c] Dipartimento di Metodi Quantitativi per le Scienze Economiche Aziendali, Università degli Studi di Milano-Bicocca, Piazza dell'Ateneo Nuovo 1, 20126, Milano, Italy

Available online: 13 Apr 2012

To cite this article: Mauro Ferrari, Camillo Fiorentini & Guido Fiorino (2009): A tableau calculus for Propositional Intuitionistic Logic with a refined treatment of nested implications, Journal of Applied Non-Classical Logics, 19:2, 149-166

To link to this article: http://dx.doi.org/10.3166/jancl.19.149-166

PLEASE SCROLL DOWN FOR ARTICLE

Full terms and conditions of use: http://www.tandfonline.com/page/terms-and-conditions

This article may be used for research, teaching, and private study purposes. Any substantial or systematic reproduction, redistribution, reselling, loan, sub-licensing, systematic supply, or distribution in any form to anyone is expressly forbidden.

The publisher does not give any warranty express or implied or make any representation that the contents will be complete or accurate or up to date. The accuracy of any instructions, formulae, and drug doses should be independently verified with primary sources. The publisher shall not be liable for any loss, actions, claims, proceedings, demand, or costs or damages whatsoever or howsoever caused arising directly or indirectly in connection with or arising out of the use of this material.

Downloaded by [University Degli Studi Dell'Insubria], [Mauro Ferrari] at 01:24 14 April 2012

A tableau calculus for Propositional Intuitionistic Logic with a refined treatment of nested implications

Mauro Ferrari*— Camillo Fiorentini— Guido Fiorino*****

* *Dipartimento di Informatica e Comunicazione*
Università degli Studi dell'Insubria
Via Mazzini 5, 21100 Varese (Italy)
mauro.ferrari@uninsubria.it

** *Dipartimento di Scienze dell'Informazione*
Università degli Studi di Milano
Via Comelico 39, 20135 Milano (Italy)
fiorenti@dsi.unimi.it

*** *Dipartimento di Metodi Quantitativi per le Scienze Economiche Aziendali*
Università degli Studi di Milano-Bicocca
Piazza dell'Ateneo Nuovo 1, 20126 Milano (Italy)
guido.fiorino@unimib.it

ABSTRACT. *Since 1993, when Hudelmaier developed an $O(n \log n)$-space decision procedure for propositional Intuitionistic Logic, a lot of work has been done to improve the efficiency of the related proof-search algorithms. In this paper a tableau calculus using the signs $\mathbf{T}$, $\mathbf{F}$ and $\mathbf{F_c}$ with a new set of rules to treat signed formulas of the kind $\mathbf{T}((A \to B) \to C)$ is provided. The main feature of the calculus is the reduction of both the non-determinism in proof-search and the width of proofs with respect to Hudelmaier's one. These improvements have a significant influence on the performances of the implementation.*

KEYWORDS: *Intuitionistic Propositional Logic, tableau calculi, decision procedures.*

DOI:10.3166/JANCL.19.149–166 © 2009 Lavoisier, Paris

1. Introduction

In this paper we present a tableau calculus for propositional Intuitionistic Logic $\mathbf{Int}$. The main feature of the calculus is a new set of rules to treat signed formulas

Downloaded by [University Degli Studi Dell'Insurbria], [Mauro Ferrari] at 01:24 14 April 2012

of the kind $\mathbf{T}((A \to B) \to C)$. This calculus collocates itself in a long history of researches on the design of efficient decision procedures for $\mathbf{Int}$. In this context, the main concern is the treatment of "positive" implicative formulas, namely implicative formulas having sign $\mathbf{T}$ in a tableau deduction or occurring in the left-side of a sequent (Dyckhoff, 1992; Hudelmaier, 1989; Hudelmaier, 1993; Miglioli *et al.*, 1997; Vorob'ev, 1970). Differently from Classical Logic, Intuitionistic implication is the main source of inefficiency in proof search for the known calculi and this circumstance makes the decision procedures for propositional Intuitionistic Logic PSPACE-complete (Statman, 1979; Waaler *et al.*, 1999).

Gentzen's early calculi (Gentzen, 1969) for $\mathbf{Int}$ were based on the re-use of implicative formulas. The major drawback of this solution is that deductions may have infinite depth, hence some loop-checking mechanism is needed to guarantee termination. To this aim, Vorob'ev (Vorob'ev, 1970) introduced (in the context of sequent calculi) rules to treat signed formulas of the kind $\mathbf{T}(A \to B)$ according to the main connective of A. See also (Dyckhoff, 1992; Miglioli *et al.*, 1997), where calculi with analogous properties are given. In these cases, the re-use of formulas is avoided by replacing $\mathbf{T}(A \to B)$ with "simpler" formulas built up from the subformulas of $A \to B$; moreover, suitable measures on formulas are defined, which guarantee that derivations have bounded depth. But, although on the one hand decision procedures for these calculi do not need loop-checking mechanisms, on the other hand the rules to treat formulas of the kind $\mathbf{T}((A \vee B) \to C)$ and $\mathbf{T}((A \to B) \to C)$ still give rise to proofs that may be of exponential depth in the size of the formula to be proved. This problem is overcome in Hudelmaier's sequent calculi (Hudelmaier, 1993), where proofs have linear depth and the related decision procedures require $O(n \log n)$-space. Here we refer to the Hudelmaier's sequent calculus LG, whose novelties essentially regard the treatment of formulas of the kind $\mathbf{T}(A \to B)$. To save space, in some rules of LG the repetition of formulas is avoided by introducing new propositional variables. Moreover, LG provides rules to handle sets of formulas containing both $\mathbf{T}(A \to B)$ and $\mathbf{F}A$, giving a rule for every possible form of the main connective of A. We remark that in (Hudelmaier, 1993) the $O(n \log n)$-space result is proved for the calculus LE, which improves LG by providing a compact notation to represent the pairs of formulas $\mathbf{F}A, \mathbf{T}(A \to B)$.

The calculus $\mathcal{T}_{\mathbf{Int}}$ we introduce in this paper is a refinement of Hudelmaier's calculus LG (Hudelmaier, 1993). Here, we improve LG by giving rules to treat formulas of the kind $\mathbf{T}((A \to B) \to C)$, for *all* the main connectives of B, without introducing rules treating pairs of signed formulas. As discussed in the paper, even although $\mathcal{T}_{\mathbf{Int}}$ has the same computational performances of Hudelmaier's calculi, it allows us to define a "better" decision procedure due to the following facts: (i) in general $\mathcal{T}_{\mathbf{Int}}$ proofs have width which is less than that of the corresponding LG proofs; (ii) $\mathcal{T}_{\mathbf{Int}}$ rules reduce the search space. Thus, both the search space and the dimension of the proofs of $\mathcal{T}_{\mathbf{Int}}$ are narrower than LG. The new rules of $\mathcal{T}_{\mathbf{Int}}$ give rise to a calculus whose proofs have depth bounded by $3n$, where n is the size of the formula to be proved; from such a calculus an $O(n \log n)$-space decision procedure is designed. We have

Downloaded by [University Degli Studi Dell'Insurbria], [Mauro Ferrari] at 01:24 14 April 2012

implemented a decision procedure based on $\mathcal{T}_{\mathbf{Int}}$, called PITP-3F, based on the PITP theorem prover of (Avellone *et al.*, 2008). Even if the computational complexity of our decision procedure only slightly improves the one of (Avellone *et al.*, 2004; Fiorino, 2001), the experimental results show that the new rules highly improve the performances of the implementation. In particular, in the paper we compare the PITP-3F with PITP (Avellone *et al.*, 2008) and STRIP (Galmiche *et al.*, 1999).

We point out that our reasoning is based on semantic tools, whereas (Hudelmaier, 1993) uses syntactic techniques; to prove the equivalence between LG and Gentzen calculus, the author has to introduce some auxiliary calculi and prove their equivalence. As a by-product, our decision procedure allows us to build a counter-model for A whenever A is not intuitionistically valid.

The paper is structured as follows: in the next section we introduce notations and the preliminary definitions. In Section 3 we describe $\mathcal{T}_{\mathbf{Int}}$ and we discuss the main differences with respect to Hudelmaier's calculus LG. In Sections 4 and 5 we prove that $\mathcal{T}_{\mathbf{Int}}$ is sound and complete and we discuss the computational complexity of the related decision procedure. Finally, in Section 6 we discuss the performances of the PITP-3F implementation of $\mathcal{T}_{\mathbf{Int}}$.

2. Notation and preliminaries

We consider the propositional language $\mathcal{L}$ based on a denumerable set of propositional variables (atoms) $\mathcal{PV}$, the logical connectives $\neg$, $\wedge$, $\vee$, $\rightarrow$, the parenthesis '(' and ')'. We write $A \in \mathcal{L}$ to mean that A is a formula of $\mathcal{L}$. To avoid unessential parenthesis, we assume that $\neg$ binds stronger than $\wedge$ and $\vee$; moreover, $\wedge$ and $\vee$ bind stronger than $\rightarrow$.

Kripke models are the main tool to semantically characterize propositional Intuitionistic Logic $\mathbf{Int}$, see e.g. (Chagrov *et al.*, 1997; Fitting, 1969) for the details. A Kripke model for $\mathcal{L}$ is a structure $\underline{K} = \langle P, \leq, \rho, \Vdash \rangle$, where $\langle P, \leq, \rho \rangle$ is a finite poset with minimum element ρ and $\Vdash$ is the *forcing relation*, namely a binary relation on $P \times \mathcal{PV}$ satisfying the *monotonicity condition*: $\alpha \Vdash p$ and $\alpha \leq \beta$ implies $\beta \Vdash p$. The forcing relation is extended to arbitrary formulas of $\mathcal{L}$ as follows:

1) $\alpha \Vdash A \wedge B$ iff $\alpha \Vdash A$ and $\alpha \Vdash B$;
2) $\alpha \Vdash A \vee B$ iff $\alpha \Vdash A$ or $\alpha \Vdash B$;
3) $\alpha \Vdash A \rightarrow B$ iff, for every $\beta \in P$ such that $\alpha \leq \beta$, $\beta \Vdash A$ implies $\beta \Vdash B$;
4) $\alpha \Vdash \neg A$ iff, for every $\beta \in P$ such that $\alpha \leq \beta$, $\beta \Vdash A$ does not hold.

We write $\alpha \nVdash A$ to mean that $\alpha \Vdash A$ does not hold. It is easy to check that the monotonicity property holds for arbitrary formulas, i.e., for every formula $A \in \mathcal{L}$, $\alpha \Vdash A$ and $\alpha \leq \beta$ implies $\beta \Vdash A$. A formula A *is valid in a Kripke model* $\underline{K} = \langle P, \leq, \rho, \Vdash \rangle$ iff $\rho \Vdash A$ (by monotonicity property, this means that $\alpha \Vdash A$ for every $\alpha \in P$). It is well-known (Chagrov *et al.*, 1997; Fitting, 1969) that propositional Intuitionistic Logic $\mathbf{Int}$ coincides with the set of formulas valid in all Kripke models.

Downloaded by [University Degli Studi Dell'Insurbria], [Mauro Ferrari] at 01:24 14 April 2012

3. The tableau calculus

The tableau calculus $\mathcal{T}_{\mathbf{Int}}$ we present in this section, is a refinement of the one introduced in (Fiorino, 2001; Miglioli *et al.*, 1997). It works on *signed formulas*, namely expressions of the kind $\mathbf{T}A$, $\mathbf{F}A$ or $\mathbf{F_c}A$, where $A \in \mathcal{L}$. Signed formulas have a natural interpretation in Kripke semantics. Given a Kripke model $\underline{K} = \langle P, \leq, \rho, \Vdash\rangle$, an element $\alpha \in P$ and a signed formula H, α *realizes* H *in* $\underline{K}$, and we write $\underline{K}, \alpha \rhd H$, iff:

- $H = \mathbf{T}A$ and $\alpha \Vdash A$;
- $H = \mathbf{F}A$ and $\alpha \nVdash A$;
- $H = \mathbf{F_c}A$ and $\alpha \Vdash \neg A$.

$\underline{K}, \alpha \ntriangleright H$ means that $\underline{K}, \alpha \rhd H$ does not hold. Given a set S of signed formulas, $\underline{K}, \alpha \rhd S$ iff $\underline{K}, \alpha \rhd H$ for every $H \in S$; we say that S *is realizable* if $\underline{K}, \alpha \rhd S$ for some $\underline{K}$ and α. We call the *certain* part of S the set

$$S_c \;=\; \{\mathbf{T}A \mid \mathbf{T}A \in S\} \cup \{\mathbf{F_c}A \mid \mathbf{F_c}A \in S\}$$

We remark that, by the monotonicity property, $\underline{K}, \alpha \rhd S$ and $\alpha \leq \beta$ imply $\underline{K}, \beta \rhd S_c$.

Table 1. *The* $\mathcal{T}_{\mathbf{Int}}$ *calculus*

$\dfrac{S, \mathbf{T}(A \wedge B)}{S, \mathbf{T}A, \mathbf{T}B}\,\mathbf{T}\wedge$	$\dfrac{S, \mathbf{F}(A \wedge B)}{S, \mathbf{F}A \mid S, \mathbf{F}B}\,\mathbf{F}\wedge$	$\dfrac{S, \mathbf{F_c}(A \wedge B)}{S_c, \mathbf{F_c}A \mid S_c, \mathbf{F_c}B}\,\mathbf{F_c}\wedge$
$\dfrac{S, \mathbf{T}(A \vee B)}{S, \mathbf{T}A \mid S, \mathbf{T}B}\,\mathbf{T}\vee$	$\dfrac{S, \mathbf{F}(A \vee B)}{S, \mathbf{F}A, \mathbf{F}B}\,\mathbf{F}\vee$	$\dfrac{S, \mathbf{F_c}(A \vee B)}{S, \mathbf{F_c}A, \mathbf{F_c}B}\,\mathbf{F_c}\vee$
Tables 2 and 3	$\dfrac{S, \mathbf{F}(A \to B)}{S_c, \mathbf{T}A, \mathbf{F}B}\,\mathbf{F}\to$	$\dfrac{S, \mathbf{F_c}(A \to B)}{S_c, \mathbf{T}A, \mathbf{F_c}B}\,\mathbf{F_c}\to$
$\dfrac{S, \mathbf{T}(\neg A)}{S, \mathbf{F_c}A}\,\mathbf{T}\neg$	$\dfrac{S, \mathbf{F}(\neg A)}{S_c, \mathbf{T}A}\,\mathbf{F}\neg$	$\dfrac{S, \mathbf{F_c}(\neg A)}{S_c, \mathbf{T}A}\,\mathbf{F_c}\neg$

where $S_c = \{\mathbf{T}A \mid \mathbf{T}A \in S\} \cup \{\mathbf{F_c}A \mid \mathbf{F_c}A \in S\}$

The rules of the tableau calculus $\mathcal{T}_{\mathbf{Int}}$ are shown in Tables 1–3. In the rules we write S, H as a shorthand for $S \cup \{H\}$. Every rule applies to a set of signed formulas, but only acts on the signed formula H explicitly indicated in the premise; we call H the *major premise* of the rule, whereas we call all the other signed formulas *minor premises* of the rule.

Downloaded by [University Degli Studi Dell'Insurbria], [Mauro Ferrari] at 01:24 14 April 2012

Table 2. *Rules for* $\mathbf{T} \to$

$$\frac{S, \mathbf{T}A, \mathbf{T}(A \to B)}{S, \mathbf{T}A, \mathbf{T}B}\,MP$$

$$\frac{S, \mathbf{T}(A \to B)}{S, \mathbf{F_c}A \mid S, \mathbf{T}B}\,\mathbf{T}{\to}certain \quad \text{if } S = S_c$$

$$\frac{S, \mathbf{T}((A \wedge B) \to C)}{S, \mathbf{T}(A \to (B \to C))}\,\mathbf{T}{\to}\wedge$$

$$\frac{S, \mathbf{T}(\neg A \to B)}{S_c, \mathbf{T}A \mid S, \mathbf{T}B}\,\mathbf{T}{\to}\neg$$

$$\frac{S, \mathbf{T}((A \vee B) \to C)}{S, \mathbf{T}(A \to q), \mathbf{T}(B \to q), \mathbf{T}(q \to C)}\,\mathbf{T}{\to}\vee \quad \text{with } q \text{ a new atom}$$

The sets in the consequence are obtained by decomposing in some way the major premise of the rule and either copying all the minor premises (see, e.g., the rule $\mathbf{T}\wedge$ of Table 1) or only copying the certain part of the minor premises (see, e.g., the rule $\mathbf{F} \to$ of Table 1). When the conclusion of a rule R contains two sets, we separate them with the *splitting symbol* $\mid$ and we call R a *splitting rule*.

Some rules require additional conditions in order to be applied. The rule $\mathbf{T} \to certain$ of Table 2 can be applied only if $S = S_c$, namely the set S of minor premises does not contain $\mathbf{F}$-signed formulas. The rule MP (*modus ponens*) of Table 2, having $\mathbf{T}(A \to B)$ as major premise, requires the presence of $\mathbf{T}A$ among the minor premises. We point out that in (Hudelmaier, 1993) this rule is restricted to the case where A is a propositional variable. Finally, we notice that some rules of Tables 2 and 3 require the introduction of a new atom q, namely a propositional variable q not occurring in the premises of the rule. This expedient goes back to (Hudelmaier, 1993) and avoids repetitions of subformulas of the major premise in the conclusion of a rule. For instance, without the introduction of q, the consequence of $\mathbf{T} \to \vee$ should be $S, \mathbf{T}(A \to C), \mathbf{T}(B \to C)$, where C occurs twice, and this double occurrence prevents the definition of a linear complexity measure on sets of signed formulas.

A set S of signed formulas is *contradictory* if $\{\mathbf{T}A, \mathbf{F}A\} \subseteq S$ or $\{\mathbf{T}A, \mathbf{F_c}A\} \subseteq S$, for some formula A. Clearly, contradictory sets are not realizable. A *proof table* (or proof tree) for S is a finite tree τ with S as root and such that all the children of a node S' of τ are the sets in the consequence of a rule applied to S'. If all the leaves of τ are contradictory sets, we say that τ is a *closed proof table* for S and we say that

S is *provable in* $\mathcal{T}_{\mathbf{Int}}$. A set of signed formulas S is *consistent* iff S is not provable in $\mathcal{T}_{\mathbf{Int}}$. As stated in Theorem 10 of Section 5, $\mathcal{T}_{\mathbf{Int}}$ is a *complete calculus* for $\mathbf{Int}$, namely: for every finite set of signed formulas S, S is consistent if and only if S is realizable. In particular, let us say that a formula A *is provable in* $\mathcal{T}_{\mathbf{Int}}$ iff $\{\mathbf{F}A\}$ is provable. Since $A \in \mathbf{Int}$ if and only if the set $\{\mathbf{F}A\}$ is not realizable, as a corollary of the completeness of $\mathcal{T}_{\mathbf{Int}}$ we get:

COROLLARY 1 (). — *A is provable in $\mathcal{T}_{\mathbf{Int}}$ iff $A \in \mathbf{Int}$.*

Table 3. *Rules for* $\mathbf{T} \to\to$

$$\frac{S, \mathbf{T}((A \to p) \to C)}{S_c, \mathbf{T}A, \mathbf{F}p, \mathbf{T}(p \to C) \mid S, \mathbf{T}C}\,\mathbf{T}{\to}{\to}Atom \quad \text{where } p \in \mathcal{PV}$$

$$\frac{S, \mathbf{T}((A \to \neg B) \to C)}{S_c, \mathbf{T}A, \mathbf{T}B \mid S, \mathbf{T}C}\,\mathbf{T}{\to}{\to}\neg$$

$$\frac{S, \mathbf{T}((A \to (X \wedge Y)) \to C)}{S_c, \mathbf{T}A, \mathbf{F}q, \mathbf{T}(X \to (Y \to q)), \mathbf{T}(q \to C) \mid S, \mathbf{T}C}\,\mathbf{T}{\to}{\to}\wedge \quad \text{with } q \text{ a new atom}$$

$$\frac{S, \mathbf{T}((A \to (X \vee Y)) \to C)}{S_c, \mathbf{T}A, \mathbf{F}q, \mathbf{T}(X \to q), \mathbf{T}(Y \to q), \mathbf{T}(q \to C) \mid S, \mathbf{T}C}\,\mathbf{T}{\to}{\to}\vee \quad \text{with } q \text{ a new atom}$$

$$\frac{S, \mathbf{T}((A \to (X \to Y)) \to C)}{S_c, \mathbf{T}A, \mathbf{T}X, \mathbf{F}q, \mathbf{T}(Y \to q), \mathbf{T}(q \to C) \mid S, \mathbf{T}C}\,\mathbf{T}{\to}{\to}{\to} \quad \text{with } q \text{ a new atom}$$

Given a set S, in general we can apply to S more than one rule, according to the choice of the major premise $H \in S$. Suppose that, after having applied the rule R, no closed proof table for S is found. If R is an invertible rule, we can conclude that S is consistent, otherwise we have to backtrack and choose in S another major premise. Invertible rules of $\mathcal{T}_{\mathbf{Int}}$ can be semantically characterized as follows [1]. Let R be a rule with premise S and consequence $S_1|\ldots|S_n$; R is *invertible* iff, for every $1 \leq k \leq n$, S_k realizable implies S realizable. Suppose that, after having applied R to S, the proof search for S fails. This means that there is a set S_k in the consequence of R such that S_k is consistent. By the completeness of $\mathcal{T}_{\mathbf{Int}}$, S_k is realizable hence, R being invertible, S is realizable as well. By the completeness of $\mathcal{T}_{\mathbf{Int}}$, we conclude that S is consistent, thus there is no way to build a closed proof table for S.

1. The discussion holds for any complete calculus for $\mathbf{Int}$.

Downloaded by [University Degli Studi Dell'Insubria], [Mauro Ferrari] at 01:24 14 April 2012

One can easily check that the rules $\mathbf{T}\wedge$, $\mathbf{F}\wedge$, $\mathbf{T}\vee$, $\mathbf{F}\vee$, $\mathbf{F_c}\vee$, $\mathbf{T}\neg$ of Table 1 and the rules MP, $\mathbf{T} \to certain$, $\mathbf{T} \to \wedge$, $\mathbf{T} \to \vee$ of Table 2 are invertible. All the other rules are not invertible, since the set S in the premise is reduced to S_c. For instance, let us consider the rule $\mathbf{F_c}\wedge$ of Table 1. If $S_c, \mathbf{F_c}A$ is realizable, then $S_c, \mathbf{F_c}(A \wedge B)$ is realizable, but we cannot conclude anything about the realizability of the signed formulas in $S \setminus S_c$.

To conclude this section we discuss the main novelties of our calculus; in particular we consider the differences among $\mathcal{T}_{\mathbf{Int}}$ and the tableau calculi of (Fiorino, 2001; Miglioli *et al.*, 1997) and the sequent calculi introduced in (Hudelmaier, 1993). For sequent calculi we present the rules adopting the standard translation into tableau rules.

First of all we notice that the rules of Tables 1 and 2 essentially coincide with those described in (Miglioli *et al.*, 1997), where the sign $\mathbf{F_c}$ is introduced to characterize Intuitionistic negation. The rules of Table 3 replace the rule

$$\frac{S, \mathbf{T}((A \to B) \to C)}{S_c, \mathbf{T}A, \mathbf{F}B, \mathbf{T}(B \to C) \mid S, \mathbf{T}C}\,\mathbf{T}{\to}{\to}$$

of (Miglioli *et al.*, 1997), that goes back to (Dyckhoff, 1992) and (Vorob'ev, 1970) (given in a sequent calculus style), and Fiorino's rule $Fio \to\to$ (Fiorino, 2001) shown at the end of this section. The aim of the rule $\mathbf{T} \to\to$ is to avoid loop-checking in the decision procedure. On the other hand, the double occurrence of the formula B in the leftmost conclusion of $\mathbf{T} \to\to$ gives rise to deductions that may be of exponential depth in the length of the formula to be proved, see (Galmiche *et al.*, 1999; Hudelmaier, 1993) for a detailed discussion. In (Hudelmaier, 1993) the problem is solved by introducing, beside the rule $\mathbf{T} \to\to$, some rules to treat the leftmost conclusion of $\mathbf{T} \to\to$, according to the main connective of B. Moreover, the calculus LG (Hudelmaier, 1993) provides rules to handle the pairs of formulas $\mathbf{F}B, \mathbf{T}(B \to C)$, according to the main connective of B. The tableau rules corresponding to the rules of LG for $B = X \vee Y$ are:

$$\frac{S, \mathbf{F}(X \vee Y), \mathbf{T}(X \vee Y \to C)}{S, \mathbf{T}(Y \to q), \mathbf{T}(q \to C), \mathbf{F}X, \mathbf{T}(X \to q)}\,Hud{\to}\vee_1$$

$$\frac{S, \mathbf{F}(X \vee Y), \mathbf{T}(X \vee Y \to C)}{S, \mathbf{T}(X \to q), \mathbf{T}(q \to C), \mathbf{F}Y, \mathbf{T}(Y \to q)}\,Hud{\to}\vee_2$$

where q is a propositional variable not occurring in the premises. We remark that both the rules are required to get completeness. Indeed, to build a proof for $S, \mathbf{T}((A \to X \vee Y) \to C)$ (working on the signed formula $\mathbf{T}((A \to X \vee Y) \to C)$), in LG we firstly have to apply the rule $\mathbf{T} \to\to$:

$$\frac{S, \mathbf{T}((A \to X \vee Y) \to C)}{S_c, \mathbf{T}A, \mathbf{F}(X \vee Y), \mathbf{T}(X \vee Y \to C) \mid S, \mathbf{T}C}\,\mathbf{T}{\to}{\to}$$

Downloaded by [University Degli Studi Dell'Insubria], [Mauro Ferrari] at 01:24 14 April 2012

Downloaded by [University Degli Studi Dell'Insurbria], [Mauro Ferrari] at 01:24 14 April 2012

At this point we have to non-deterministically choose which rule to apply between $Hud \to \vee_1$ and $Hud \to \vee_2$. In the former case we get

$$S_c, \mathbf{T}A, \mathbf{T}(Y \to q), \mathbf{T}(q \to C), \mathbf{F}X, \mathbf{T}(X \to q) \mid S, \mathbf{T}C$$

in the latter

$$S_c, \mathbf{T}A, \mathbf{T}(X \to q), \mathbf{T}(q \to C), \mathbf{F}Y, \mathbf{T}(Y \to q) \mid S, \mathbf{T}C$$

Obviously, to build up a closed proof table it may be necessary to try both rules. In contrast, in $\mathcal{T}_{\mathbf{Int}}$ only the application of the rule $\mathbf{T} \to\to \vee$ is required:

$$\frac{S, \mathbf{T}((A \to X \vee Y) \to C)}{S_c, \mathbf{T}A, \mathbf{F}q, \mathbf{T}(X \to q), \mathbf{T}(Y \to q), \mathbf{T}(q \to C) \mid S, \mathbf{T}C}\,\mathbf{T}\!\to\to\vee$$

Hence our rule decreases the non-determinism in proof-search.

Now, let us consider the rule of LG for the case $B = X \wedge Y$

$$\frac{S, \mathbf{F}(X \wedge Y), \mathbf{T}(X \wedge Y \to C)}{S, \mathbf{F}X, \mathbf{T}(X \to (Y \to C)) \mid S, \mathbf{F}Y, \mathbf{T}(Y \to (X \to C))}\,Hud\!\to\!\wedge$$

and let us consider the tableau

$$\frac{\dfrac{S, \mathbf{T}((A \to X \wedge Y) \to C)}{S_c, \mathbf{T}A, \mathbf{F}(X \wedge Y), \mathbf{T}(X \wedge Y \to C) \mid S, \mathbf{T}C}\,\mathbf{T}\!\to\to}{S_c, \mathbf{T}A, \mathbf{F}X, \mathbf{T}(X \to (Y \to C)) \mid S_c, \mathbf{T}A, \mathbf{F}Y, \mathbf{T}(Y \to (X \to C)) \mid S, \mathbf{T}C}\,Hud\!\to\!\wedge$$

In our calculus, for the same initial premise we get:

$$\frac{S, \mathbf{T}((A \to X \wedge Y) \to C)}{S_c, \mathbf{T}A, \mathbf{F}q, \mathbf{T}(X \to (Y \to q)), \mathbf{T}(q \to C) \mid S, \mathbf{T}C}\,\mathbf{T}\!\to\to\wedge$$

where q is a new propositional variable. Our rule decreases the width of the proof tree. Indeed, to decide the realizability of the initial set, with our calculus $\mathcal{T}_{\mathbf{Int}}$ two sets have to be decided, instead of three sets as in LG.

Finally, let us consider the LG rule for the case $B = X \to Y$

$$\frac{S, \mathbf{F}(X \to Y), \mathbf{T}((X \to Y) \to C)}{S_c, \mathbf{T}X, \mathbf{F}Y, \mathbf{T}(Y \to C)}\,Hud\!\to\to$$

and let us consider the tableau

$$\frac{\dfrac{S, \mathbf{T}((A \to (X \to Y)) \to C)}{S_c, \mathbf{T}A, \mathbf{F}(X \to Y), \mathbf{T}((X \to Y) \to C) \mid S, \mathbf{T}C}\,\mathbf{T}\!\to\to}{S_c, \mathbf{T}A, \mathbf{T}X, \mathbf{F}Y, \mathbf{T}(Y \to C) \mid S, \mathbf{T}C}\,Hud\!\to\to$$

Downloaded by [University Degli Studi Dell'Insurbria], [Mauro Ferrari] at 01:24 14 April 2012

In our calculus the corresponding tableau is

$$\frac{S, \mathbf{T}((A \to (X \to Y)) \to C)}{S_c, \mathbf{T}A, \mathbf{T}X, \mathbf{F}q, \mathbf{T}(Y \to q), \mathbf{T}(q \to C) \mid S, \mathbf{T}C}\mathbf{T}{\to}{\to}{\to}$$

with q a new propositional variable. Hence, while we apply one non-invertible rule, in the previous proof tree two non-invertible rules are required. A deeper discussion about the proof-search strategy is given after the proof of the Completeness Theorem in Section 5.

We emphasize that the rules of Table 3 are a refinement of the rule

$$\frac{S, \mathbf{T}((A \to B) \to C)}{S_c, \mathbf{T}A, \mathbf{F}q, \mathbf{T}(B \to q), \mathbf{T}(q \to C) \mid S, \mathbf{T}C}Fio{\to}{\to} \qquad \text{with } q \text{ a new atom}$$

introduced in (Fiorino, 2001). The calculus (Fiorino, 2001) gives rise to proof trees having depth bounded by $6n$, where n is the length of the formula to be proved, and this yields an $O(n \log n)$-space decision procedure for $\mathbf{Int}$. Rules of Table 3 are obtained by specializing rule $Fio \to\to$ according to the main connective of B. As we discuss in Section 5, the new rules allow us to get proof trees having depth $3n$ at most (see Theorem 11).

4. Soundness

In order to prove the soundness of $\mathcal{T}_{\mathbf{Int}}$, we prove that its rules preserve realizability, namely: if the set in the premise of a rule R is realizable, then one of the sets in the consequence of R is realizable as well.

The following lemma is helpful to treat the rules of Table 3.

LEMMA 2. — *Let $\underline{K} = \langle P, \leq, \rho, \Vdash \rangle$ be a Kripke model and let $\alpha \in P$ such that*

$$\underline{K}, \alpha \rhd S, \mathbf{T}((A \to B) \to C) \quad \textit{and} \quad \underline{K}, \alpha \not\rhd \mathbf{T}C$$

Let $\mathcal{V}$ be the set of propositional variables occurring in $S \cup \{\mathbf{T}((A \to B) \to C)\}$ and let q be a propositional variable such that $q \notin \mathcal{V}$. Then, there exists a Kripke model $\underline{K}' = \langle P', \leq', \rho', \Vdash' \rangle$ and $\alpha' \in P'$ such that

$$\underline{K}', \alpha' \rhd S_c, \mathbf{T}A, \mathbf{F}q, \mathbf{T}(B \to q), \mathbf{T}(q \to B), \mathbf{T}(q \to C)$$

PROOF 3. — Let $\underline{K}' = \langle P, \leq, \rho, \Vdash' \rangle$ be the Kripke model based on the poset $\langle P, \rho, \leq \rangle$ with $\Vdash'$ defined as follows:

- if $p \in \mathcal{V}$ then, for every $\gamma \in P$, $\gamma \Vdash' p$ iff $\gamma \Vdash p$;
- for every $\gamma \in P$, $\gamma \Vdash' q$ iff $\gamma \Vdash B$;

- if $p \notin \mathcal{V} \cup \{q\}$ then, for every $\gamma \in P$, $\gamma \nVdash' p$.

It is easy to check that $\Vdash'$ satisfies the monotonicity condition. Moreover, if H is a formula whose propositional variables belong to $\mathcal{V}$ and $\gamma \in P$, then $\gamma \Vdash H$ iff $\gamma \Vdash' H$. In particular, by the assumptions $\alpha \Vdash (A \to B) \to C$ and $\alpha \nVdash C$, we get $\alpha \Vdash' (A \to B) \to C$ and $\alpha \nVdash' C$. This implies $\alpha \nVdash' A \to B$, therefore there exists $\beta \in P$ such that $\alpha \leq \beta$, $\beta \Vdash' A$ and $\beta \nVdash' B$. We get:

1) $\beta \Vdash' B \to q$ and $\beta \Vdash' q \to B$ (by definition of $\Vdash'$ on q);
2) $\beta \nVdash' q$ (by (1) and by the fact that $\beta \nVdash' B$);
3) $\beta \Vdash' q \to C$ (indeed, $\alpha \Vdash' (A \to B) \to C$, $\alpha \leq \beta$ and $\beta \Vdash' q \to B$).

Summarizing, we conclude

$$\underline{K}', \beta \rhd S_c, \mathbf{T}A, \mathbf{F}q, \mathbf{T}(B \to q), \mathbf{T}(q \to B), \mathbf{T}(q \to C)$$

which proves the assertion. ■

Now we prove that the rules of $\mathcal{T}_{\mathbf{Int}}$ preserve the realizability.

LEMMA 4. — *Let S be a set of signed formulas, let $\underline{K} = \langle P, \leq, \rho, \Vdash \rangle$ be a Kripke model and let $\alpha \in P$ such that $\underline{K}, \alpha \rhd S$, and let R be a rule of $\mathcal{T}_{\mathbf{Int}}$ applicable to S. Then, there exist a set S' in the consequence of the rule R, a Kripke model $\underline{K}' = \langle P', \leq', \rho', \Vdash' \rangle$ and $\alpha' \in P'$ such that $\underline{K}', \alpha' \rhd S'$.*

PROOF 5. — By case distinction on R. We only discuss the most relevant cases of Tables 2 and 3.

Rule $\mathbf{T} \to certain$: let us assume $\underline{K}, \alpha \rhd S_c, \mathbf{T}(A \to B)$. By finiteness of P, there is $\phi \in P$ such that $\alpha \leq \phi$ and ϕ is a maximal element of $\underline{K}$ (that is, for every $\psi \in P$, $\phi \leq \psi$ implies $\phi = \psi$). By the monotonicity property, $\underline{K}, \phi \rhd S_c, \mathbf{T}(A \to B)$. If $\phi \Vdash B$, we immediately get $\underline{K}, \phi \rhd S_c, \mathbf{T}B$; otherwise $\phi \nVdash A$ and, being ϕ a maximal element, this implies $\phi \Vdash \neg A$, hence $\underline{K}, \phi \rhd S_c, \mathbf{F_c}A$.

Rule $\mathbf{T} \to\to Atom$: if $\underline{K}, \alpha \rhd S, \mathbf{T}((A \to p) \to C)$, then $\alpha \Vdash (A \to p) \to C$, thus $\alpha \Vdash C$ or $\alpha \nVdash A \to p$. In the first case, we immediately deduce that $\underline{K}, \alpha \rhd S, \mathbf{T}C$. In the second case, there exists $\beta \in P$ such that $\alpha \leq \beta$, $\beta \Vdash A$ and $\beta \nVdash p$. Moreover, since $\beta \Vdash (A \to p) \to C$, we also have $\beta \Vdash p \to C$. We conclude that $\underline{K}, \beta \rhd S_c, \mathbf{T}A, \mathbf{F}p, \mathbf{T}(p \to C)$.

Rule $\mathbf{T} \to\to \vee$: if $\underline{K}, \alpha \rhd S, \mathbf{T}((A \to (X \vee Y)) \to C)$, then $\alpha \Vdash (A \to (X \vee Y)) \to C$. If $\alpha \Vdash C$, we immediately get $\underline{K}, \alpha \rhd S, \mathbf{T}C$. Otherwise, by Lemma 2 there exist a Kripke model $\underline{K}' = \langle P', \leq', \rho', \Vdash' \rangle$, $\alpha' \in P'$ and q such that

$$\underline{K}', \alpha' \rhd S_c, \mathbf{T}A, \mathbf{F}q, \mathbf{T}((X \vee Y) \to q), \mathbf{T}(q \to (X \vee Y)), \mathbf{T}(q \to C)\ .$$

Since $\alpha' \Vdash' (X \vee Y) \to q$ implies both $\alpha' \Vdash' X \to q$ and $\alpha' \Vdash' Y \to q$, we get $\underline{K}', \alpha' \rhd S_c, \mathbf{T}A, \mathbf{F}q, \mathbf{T}(X \to q), \mathbf{T}(Y \to q), \mathbf{T}(q \to C)$.

Downloaded by [University Degli Studi Dell'Insubria], [Mauro Ferrari] at 01:24 14 April 2012

Downloaded by [University Degli Studi Dell'Insubria], [Mauro Ferrari] at 01:24 14 April 2012

Rule $\mathbf{T}\rightarrow\rightarrow\rightarrow$: if $\underline{K}, \alpha \rhd S, \mathbf{T}((A \rightarrow (X \rightarrow Y)) \rightarrow C)$, then $\alpha \Vdash (A \rightarrow (X \rightarrow Y)) \rightarrow C$. If $\alpha \Vdash C$, we immediately get $\underline{K}, \alpha \rhd S, \mathbf{T}C$. Otherwise, by Lemma 2 there exist a Kripke model $\underline{K}' = \langle P', \leq', \rho', \Vdash' \rangle$, $\alpha' \in P'$ and q such that

$$\underline{K}', \alpha' \rhd S_c, \mathbf{T}A, \mathbf{F}q, \mathbf{T}((X \rightarrow Y) \rightarrow q), \mathbf{T}(q \rightarrow (X \rightarrow Y)), \mathbf{T}(q \rightarrow C)\ .$$

Since $\alpha' \Vdash' (X \rightarrow Y) \rightarrow q$ and $\alpha' \nVdash' q$, there exists $\beta' \in P'$ such that $\alpha' \leq' \beta'$, $\beta' \Vdash' X$ and $\beta' \nVdash' Y$. Since $\beta' \Vdash' q \rightarrow (X \rightarrow Y)$, we have $\beta' \nVdash' q$. Moreover, since $\beta' \Vdash' (X \rightarrow Y) \rightarrow q$, it holds that $\beta' \Vdash' Y \rightarrow q$. Summarizing, we get

$$\underline{K}', \beta' \rhd S_c, \mathbf{T}A, \mathbf{F}q, \mathbf{T}X, \mathbf{T}(Y \rightarrow q), \mathbf{T}(q \rightarrow C)$$

and this concludes the proof.

The other cases are similar. In particular, in theses cases the consequence of a rule is realized in the same model $\underline{K}$ (or even at the same element α). ■

As a consequence we get:

THEOREM 6 (SOUNDNESS). — *Let S be a set of signed formulas. If S is realizable, then S is consistent.*

PROOF 7. — Suppose that S is not consistent and let τ be a closed proof table for S. If, by absurd, S is realizable, by the previous lemma there must be a leaf S_f of τ such that S_f is realizable, a contradiction (recall that S_f is a contradictory set). Thus, S is not realizable, and this concludes the proof. ■

5. Completeness

To prove the completeness of $\mathcal{T}_{\mathbf{Int}}$ we introduce the complexity measure $\deg$ on formulas:

- if p is a propositional variable, then $\deg(p) = 0$;
- $\deg(A \wedge B) = \deg(A) + \deg(B) + 2$;
- $\deg(A \vee B) = \deg(A) + \deg(B) + 3$;
- $\deg(A \rightarrow B) = \deg(A) + \deg(B) + 1$;
- $\deg(\neg A) = \deg(A) + 1$.

We extend the function $\deg$ to signed formulas as follows:

- For a signed formula $\mathcal{S}A$ ($\mathcal{S} \in \{\mathbf{T}, \mathbf{F}, \mathbf{F_c}\}$), $\deg(\mathcal{S}A) = \deg(A)$.
- For a finite set S of signed formulas, $\deg(S) = \sum_{H \in S} \deg(H)$.

The definition of $\deg$ is motivated by the fact that, if S' is a set in the consequence of a rule of $\mathcal{T}_{\mathbf{Int}}$ applied to a finite set of signed formulas S, then $\deg(S') < \deg(S)$.

To describe our proof search strategy, we introduce the notion of *rule related to S and H*, where S is a set of signed formulas and H a signed formula.

- If H has not the form $\mathbf{T}(A \rightarrow B)$, the rule related to S and H is the only rule of Table 1 having H as major premise and $S \setminus \{H\}$ as set of minor premises.

- If $H = \mathbf{T}(A \to B)$ and $\mathbf{T}A \in S$, the rule related to S and H is the rule MP of Table 2 having H as major premise and $S \setminus \{H\}$ as set of minor premises.

- If $H = \mathbf{T}(A \to B)$, $\mathbf{T}A \notin S$ and $S = S_c$, the rule related to S and H is the rule $\mathbf{T} \to certain$ of Table 2 having H as major premise and $S \setminus \{H\}$ as set of minor premises.

- If $H = \mathbf{T}(A \to B)$, $\mathbf{T}A \notin S$ and $S \neq S_c$, the rule related to S and H is one of the rules of Table 2 and 3 having H as major premise and $S \setminus \{H\}$ as set of minor premises (there exists only one applicable rule).

Notice that given S and H there exists at most one rule R of $\mathcal{T}_{\mathbf{Int}}$ related to S and H. If R is a splitting rule, we denote with $\mathcal{R}^1_{S,H}$ and $\mathcal{R}^2_{S,H}$ the leftmost set and the rightmost set in the consequence of R respectively; for non-splitting rules we denote with $\mathcal{R}^1_{S,H}$ the only set in the consequence of R ($\mathcal{R}^2_{S,H}$ is not defined). The main lemma to prove the completeness of $\mathcal{T}_{\mathbf{Int}}$ is:

LEMMA 8. — *Let S be a finite set of signed formulas. If S is consistent, then S is realizable.*

PROOF 9. — By complete induction on $\deg(S)$. Assume that the assertion holds for all S' such that $\deg(S') < \deg(S)$; we prove it for S. Let $S_0 \subseteq S$ be the set of signed formulas H of S satisfying one of the following conditions:

(i) $H = \mathbf{T}(A \wedge B)$ or $H = \mathbf{F}(A \wedge B)$ or $H = \mathbf{T}(A \vee B)$ or $H = \mathbf{F}(A \vee B)$ or $H = \mathbf{F}_c(A \vee B)$ or $H = \mathbf{T}(\neg A)$ or $H = \mathbf{T}((A \wedge B) \to C)$ or $H = \mathbf{T}((A \vee B) \to C)$.

(ii) $H = \mathbf{T}(A \to B)$ and ($\mathbf{T}A \in S$ or $S = S_c$).

(iii) $H = \mathbf{T}(\neg A \to C)$ or $H = \mathbf{T}((A \to p) \to C)$ or $H = \mathbf{T}((A \to \neg B) \to C)$ or $H = \mathbf{T}((A \to (X \wedge Y)) \to C)$ or $H = \mathbf{T}((A \to (X \vee Y)) \to C)$ or $H = \mathbf{T}((A \to (X \to Y)) \to C)$, and $\mathcal{R}^2_{S,H}$ is consistent.

Firstly, let us assume that $S_0 \neq \emptyset$ and let H be any formula of S_0. Since S is consistent, there exists $k \in \{1, 2\}$ such that the set $S' = \mathcal{R}^k_{S,H}$ is consistent; in particular, if H is one of the signed formulas in case (iii), we take $S' = \mathcal{R}^2_{S,H}$, where we recall that $\mathcal{R}^2_{S,H} = (S \setminus \{H\}) \cup \{\mathbf{T}C\}$. Since S' is consistent and $\deg(S') < \deg(S)$, by induction hypothesis there exists a Kripke model $\underline{K} = \langle P, \leq, \rho, \Vdash \rangle$ such that $\underline{K}, \rho \rhd S'$. It is easy to check that $\underline{K}, \rho \rhd S$, and this proves the assertion.

Secondly, let us assume that $S_0 = \emptyset$. Let $S_1 \subseteq S$ be the set of formulas $H \in S$ satisfying one of the following conditions:

1) $H = \mathbf{T}p$ or $H = \mathbf{F_c}p$ or $H = \mathbf{F}p$, with p a propositional variable.
2) $H = \mathbf{T}(p \to B)$, with p a propositional variable and $\mathbf{T}p \notin S$.

Let $S_2 \subseteq S$ be the set of formulas $H \in S$ satisfying one of the following conditions:

3) $H = \mathbf{F_c}(A \wedge B)$ or $H = \mathbf{F}(A \to B)$ or $H = \mathbf{F_c}(A \to B)$ or $H = \mathbf{F}(\neg A)$ or $H = \mathbf{F_c}(\neg A)$.
4) $H = \mathbf{T}(\neg A \to C)$ or $H = \mathbf{T}((A \to Z) \to C)$, and $\mathcal{R}^1_{S,H}$ is consistent.

Since S is consistent and S_0 is empty, we have $S_1 \cup S_2 = S$. If $S_2 = \emptyset$, then $S = S_1$ can be realized in the Kripke model $\underline{K} = \langle P, \leq, \rho, \Vdash \rangle$ where $P = \{\rho\}$ and,

Downloaded by [University Degli Studi Dell'Insurbria], [Mauro Ferrari] at 01:24 14 April 2012

Downloaded by [University Degli Studi Dell'Insurbria], [Mauro Ferrari] at 01:24 14 April 2012

for every propositional variable p, $\rho \Vdash p$ iff $\mathbf{T}p \in S$ (note that $\underline{K}$ can be seen as a classical model). Otherwise $S_2 \neq \emptyset$. Let us assume $S_2 = \{H_1, \ldots, H_n\}$. By the choice of S_2, for every $j \in \{1, \ldots, n\}$ there is $k \in \{1, 2\}$ such that the set $T_j = \mathcal{R}^k_{S,H_j}$ is consistent (if $H_j \neq \mathbf{F_c}(A \wedge B)$, we take $k = 1$). Since $\deg(T_j) < \deg(S)$ and T_j is consistent, by induction hypothesis there exists a Kripke model $\underline{K}_j = \langle P_j, \leq_j, \rho_j, \Vdash_j \rangle$ such that $\underline{K}_j, \rho_j \rhd T_j$. Without loss of generality, we assume that the P_j's are pairwise disjoint. We build the Kripke model $\underline{K} = \langle P, \leq, \rho, \Vdash \rangle$ where ρ is a new element ($\rho \notin \bigcup_{1 \leq j \leq n} P_j$) and the immediate successors of ρ are the elements $\rho_1, \ldots, \rho_n$; formally:

$$P \;=\; \bigcup_{1 \leq j \leq n} P_j \;\cup\; \{\rho\} \qquad\qquad \leq \;=\; \bigcup_{1 \leq j \leq n} \leq_j \;\cup \{(\rho, \alpha) \;\mid\; \alpha \in P\}$$

Finally, for every $\alpha \in P$ and every propositional variable p, $\alpha \Vdash p$ iff one of the following conditions holds:

- there is $j \in \{1, \ldots, n\}$ such that $\alpha \in P_j$ and $\alpha \Vdash_j p$;
- $\alpha = \rho$ and $\mathbf{T}p \in S$.

One can easily prove that $\Vdash'$ satisfies the monotonicity condition. Moreover, for every $\alpha \in P_j$ and every formula H, $\alpha \Vdash H$ iff $\alpha \Vdash_j H$; in particular, $\underline{K}, \rho_j \rhd T_j$ for every $1 \leq j \leq n$.

We prove that $\underline{K}, \rho \rhd H$ for every $H \in S$ (recall that $S = S_1 \cup S_2$). The proof bases on a case distinction on the conditions (1)–(4).

If $H = \mathbf{T}p$, by definition $\rho \Vdash p$. If $H = \mathbf{F}p$ then, by consistency of S, $\mathbf{T}p \notin S$, hence $\rho \nVdash p$. If $H = \mathbf{F_c}p$, then $\mathbf{F_c}p \in T_j$ for every $1 \leq j \leq n$ (indeed, $\mathbf{F_c}p \in S_c$ and $S_c \subseteq T_j$). It follows that $\rho_j \Vdash \neg p$ for every $1 \leq j \leq n$. Moreover, by consistency of S, $\mathbf{T}p \notin S$. We conclude $\rho \Vdash \neg p$.

Let $H = \mathbf{T}(p \to B)$ and let $\alpha \in P$ such that $\alpha \Vdash p$. Since $\mathbf{T}p \notin S$ (by Condition (2) in the definition of S_1), we have $\rho \nVdash p$, hence $\alpha \neq \rho$. Let i be such that $\alpha \in P_i$. Since $\rho_i \Vdash p \to B$ and $\rho_i \leq \alpha$, it follows that $\alpha \Vdash B$.

Let $H = \mathbf{F}(A \to B)$. There exists an m such that $T_m = (S_c \setminus \{H\}) \cup \{\mathbf{T}A, \mathbf{F}B\}$ and $\underline{K}, \rho_m \rhd T_m$. It follows that $\rho_m \Vdash A$ and $\rho_m \nVdash B$, hence $\rho \nVdash A \to B$.

Let $H = \mathbf{T}((A \to (X \wedge Y)) \to C)$. There exists m such that

$$T_m \;=\; (\, S_c \setminus \{H\}\,) \;\cup\; \{\, \mathbf{T}A,\; \mathbf{F}p,\; \mathbf{T}(X \to (Y \to p)),\; \mathbf{T}(p \to C)\,\}$$

and $\underline{K}, \rho_m \rhd T_m$. Let $\alpha \in P$ such that $\alpha \Vdash A \to (X \wedge Y)$. Since $\rho \leq \rho_m$, $\rho_m \Vdash A$ and $\rho_m \nVdash X \wedge Y$ (otherwise, $\rho_m \Vdash p$ would follow), $\alpha \neq \rho$. Let j be such that $\alpha \in P_j$. If $j = m$, we have $\rho_m \leq \alpha$, which implies $\alpha \Vdash C$. Let $j \neq m$. In this case, $H \in T_j$. By the fact that $\underline{K}, \rho_j \rhd T_j$, $H \in T_j$ and $\rho_j \leq \alpha$, we get $\alpha \Vdash C$. The remaining cases are similar. ■

By the previous lemma and the Soundness Theorem (Theorem 6), we conclude that $\mathcal{T}_{\mathbf{Int}}$ is a complete calculus for $\mathbf{Int}$:

Downloaded by [University Degli Studi Dell'Insubria], [Mauro Ferrari] at 01:24 14 April 2012

THEOREM 10 (COMPLETENESS). — *Let S be a finite set of signed formulas. Then, S is consistent if and only if S is realizable.*

The proof of Lemma 8 implicitly defines a decision procedure for Intuitionistic Logic; indeed, starting from a finite set S of signed formulas, either a closed proof table or a counter-model for S can be built. In the following, we sketch the strategy we apply in the decision procedure.

In our decision procedure, cases (i)–(ii) in the definition of S_0 correspond to the application of invertible rules. As usual, applying invertible rules before non-invertible ones reduces the search-space. Accordingly, if there exists $H \in S$ satisfying one of cases (i)–(ii), we firstly apply the rule related to S and H; if the search for a closed proof table fails, we conclude that S is not provable (as discussed in Section 3, there is no need to backtrack and try the application of another rule to S). Otherwise, let us assume that no formula $H \in S$ satisfies cases (i)–(ii) and that there exists an $H = \mathbf{T}(A \to B)$ in S. Under that assumption, we try to build first a proof table for the "invertible consequence" $\mathcal{R}^2_{S,H} = (S \setminus \{H\}) \cup \{\mathbf{T}C\}$; if such a proof does not exist, we get a counter-model for S and hence S is not provable. On the other hand, if we find a proof for $\mathcal{R}^2_{S,H}$ but $\mathcal{R}^1_{S,H}$ is not provable, one of the cases (3) and (4) in the definition of S_2 holds: neither a proof table nor any counter model can be constructed. We have to try the application of another rule to S because the counter model for S relies on the counter model of $\mathcal{R}^1_{S,H_j}$, for all $H_j \in S$, as a whole. In all the other cases, either non-invertible rules are applicable to S or no rules at all.

Finally, we remark that a proof table for a set S not containing **F**-signed formulas is a classical derivation. Indeed, in the proof we can always apply one of the rules of Table 1 or the rules MP and $\mathbf{T} \to certain$ of Table 2, which are classical rules and do not generate **F**-signed formulas.

We conclude this section discussing the complexity of our calculus. Given a formula A, $|A|$ denotes the number symbols occurring in A; similarly, if S is a set of signed formulas, $|S|$ is the number of symbols occurring in S.

THEOREM 11 (). — *Let S be a finite set of signed formulas. Then, the depth of every proof table for S is at most $3|S|$.*

PROOF 12. — Let us consider the complexity measure $\deg$ defined at the beginning of Section 5. By induction hypothesis on the structure of a formula A, one can prove that $\deg(A) \leq 3|A|$. This implies that $\deg(S) \leq 3|S|$. By inspecting the rules of the calculus and how they are used to build proof tables, it follows that the complexity w.r.t. $\deg$ of every set of signed formulas in a proof tree is higher than all its immediate successors, and this proves the proposition. ■

An inspection of the rules of $\mathcal{T}_{\mathbf{Int}}$ yields that the increase of symbols in any consequence compared to its premise is bounded by a constant. As a consequence, see (Hudelmaier, 1993), a depth-first decision procedure for S requires at most $O(n \log n)$ bits to store the required data structures.

Downloaded by [University Degli Studi Dell'Insurbria], [Mauro Ferrari] at 01:24 14 April 2012

6. Experimental results

We devote this section to discuss the improvements obtained by implementing rules of Table 3. We have implemented the new rules by modifying PITP (Avellone *et al.*, 2008). No further modification has been done. We call PITP-3F the new version of the theorem prover[2]. We remark that PITP implements the tableau calculus of (Avellone *et al.*, 2004; Fiorino, 2001) and it is at present the fastest available theorem prover for propositional Intuitionistic Logic on the formulas of the ILTP library, see (Avellone *et al.*, 2008) for a detailed comparison with other provers.

Experiments have been carried out along the lines of (Raths *et al.*, 2007) and the results are summarized in Tables 4–6[3]. In particular, Table 4 and Table 5 refer to simulations over randomly generated formulas with a time bound of 10 minutes, the former considering formulas with 2000 connectives and 100 variables, the latter referring to formulas with 5000 connectives and 100 variables. Table 6 summarizes the results obtained with the same formulas considered in Table 5 with the time bound extended to 50 minutes. In every entry we indicate the number of formulas decided in the specified time range and between brackets we put the total time required to decide them; "k(n.a.)" in the last column means that k formulas have not been decided within the indicated time bound. The last row indicates the gain in number of formulas of PITP-3F over PITP.

Table 4. *Randomly generated formulas with* 2000 *connectives and* 100 *variables, time limit* 10 *minutes*

	0-1s	1-10s	10-100s	100-600s	>600s
PITP	1905(20s)	20(71s)	14(508s)	11(2576s)	48(n.a.)
PITP-3F	1910(21s)	19(67s)	13(368s)	12(2901s)	44(n.a.)
Total improvement	+5	+4	+3	+4	

A deeper analysis of the execution times on randomly generated formulas with 2000 connectives and 100 variables shows that PITP requires 3175 seconds to solve the 1950 formulas decided in 10 minutes. To decide these 1950 formulas, PITP-3F takes 1913 seconds: this gives an improvement of about 40%. If we consider also the four formulas decided by PITP-3F and not decided by PITP in 10 minutes, we have that PITP-3F requires 3357 seconds, whereas PITP requires 6876 seconds with an advantage of about 51%. We run also STRIP on the same formulas. During the experiments we observed that on the first 782 formulas, STRIP took more than 10 minutes on 341 of them.

As for Table 5, if we consider the formulas decided by both provers in 10 minutes we get that PITP-3F requires 3093 seconds whereas PITP requires 4651 seconds, with

2. Available at `http://www.dimequant.unimib.it/~guidofiorino/pitp.jsp`
3. Experiments have been carried out on a 3.00GHz Intel Xeon CPU computer with 2MB cache size and 2GB RAM.

Table 5. *Randomly generated formulas with* 5000 *connectives and* 100 *variables, time limit* 10 *minutes*

	0-1s	1-10s	10-100s	100-600s	>600s
PITP	1810(43s)	41(142s)	22(844s)	14(3622s)	113(n.a.)
PITP-3F	1810(42s)	44(140s)	22(678s)	23(5904s)	101(n.a.)
Total improvement	0	+3	+3	+12	

a time reduction of about 34%. If we also consider the twelve formulas decided by PITP-3F and not decided by PITP in 10 minutes, we have that PITP-3F requires 6765 seconds, whereas PITP requires 45320 seconds and the improvement is about 85%. If we extend the time bound to 50 minutes (Table 6) we see that PITP-3F requires 16950 seconds whereas PITP requires 30156 seconds, with an advantage of about 44%. Finally, if we also consider the seven formulas decided by PITP-3F and not decided by PITP in 50 minutes we have 24893 seconds vs. 128349 seconds with an improvement of about 80%.

Table 6. *Randomly generated formulas with* 5000 *connectives and* 100 *variables, time limit* 50 *minutes*

	0-1s	1-10s	10-100s	100-3000s	>3000s
PITP	1810(43s)	41(142s)	22(844s)	32(29127s)	95(n.a.)
PITP-3F	1810(42s)	44(140s)	22(678s)	36(24032s)	88(n.a.)
Total improvement	0	+3	+3	+7	

As another experiment we run both provers on 2000 randomly generated formulas containing 750 connectives and 50 variables without time limit. PITP-3F solved all the formulas in 77848 seconds whereas PITP took 166573 seconds.

To conclude this discussion, we remark that we have not reported experiments over the formulas in ILTP Library (Raths *et al.*, 2007). On these formulas PITP-3F weakly improves PITP, but this essentially depends on implementation features; indeed, on the formulas of ILTP Library the significant features of our calculus are not exploited since they contains only "trivial" cases of nested implications and only the $\mathbf{T} \rightarrow\rightarrow Atom$ rule is required.

In Table 7 we report the results related to some formulas of ILTP library modified by substituting every propositional variable p_i occurring in them with the formula $(r_i \rightarrow s_i) \rightarrow t_i$. By this substitution we obtain formulas with nested implications. The execution times show that PITP-3F is faster than PITP; in particular, on the family formulas obtained from SYJ207+1 and SYJ211+1 the running time of PITP grows faster than PITP-3F. This is a further clue that the rules introduced in this paper improve the performances. For the sake of completeness we also run STRIP. The results show that STRIP outperforms PITP-3F on two families of formulas. We remark that the growing ratio on the family SYJ203 is approximately four for both provers. On

Downloaded by [University Degli Studi Dell'Insubria], [Mauro Ferrari] at 01:24 14 April 2012

Downloaded by [University Degli Studi Dell'Insurbria], [Mauro Ferrari] at 01:24 14 April 2012

the other families PITP-3F is faster than STRIP. We also remark that on such a families PITP-3F has a lower growing ratio than STRIP.

Table 7. PITP *and* PITP-3F *compared on formulas of ILTP library modified by substituting* p_i *with* $(r_i \rightarrow s_i) \rightarrow t_i$ *(times in seconds)*

Formula SYJ	PITP	PITP-3F	STRIP
201.3	0.12	0.11	*0*
201.4	1.46	1.17	*0.004*
201.5	15.92	12.21	*0.008*
201.6	165.78	126.23	*0.02*
203.8	0.290	0.210	*0.056*
203.9	1.06	0.8	*0.2*
203.10	4.12	3.02	*0.85*
203.11	15.95	11.71	*3.10*
207+1.2	0.02	*0.001*	> 600
207+1.3	0.78	*0.170*	> 600
207+1.4	151.83	*8.5*	> 600

Formula SYJ	PITP	PITP-3F	STRIP
209+1.6	0.18	*0.13*	9.56
209+1.7	1.48	*1.02*	254
209+1.8	13.99	*9.28*	> 600
209+1.9	141.95	*92.99*	> 600
211+1.6	0.66	*0.04*	43
211+1.7	2.7	*0.05*	> 600
211+1.8	11.24	*0.12*	> 600
211+1.9	46.22	*0.28*	> 600

7. Conclusions

This paper describes the tableau calculus for propositional Intuitionistic Logic $\mathcal{T}_{\mathbf{Int}}$. On the one hand $\mathcal{T}_{\mathbf{Int}}$ has the same computational properties of the calculus LG presented in paper (Hudelmaier, 1993), on the other hand $\mathcal{T}_{\mathbf{Int}}$ has some features that LG lacks and deserves to be considered. In particular, both the proof search-space and the size of the proof-tree of $\mathcal{T}_{\mathbf{Int}}$ are narrower than LG and this can reduce the running time. $\mathcal{T}_{\mathbf{Int}}$ is also an improvement of the calculus of (Avellone *et al.*, 2004) on which PITP is based. At present PITP is the fastest prover among those of ILTP library. Our comparisons between PITP and PITP-3F, that is the implementation of PITP in which the new rules are inserted, confirm that in the practice $\mathcal{T}_{\mathbf{Int}}$ gives advantages.

8. References

Avellone A., Fiorino G., Moscato U., “A new O($n \log n$)-SPACE decision procedure for propositional intuitionistic logic”, *in* M. Baaz, J. Makowsky, A. Voronkov (eds), *LPAR 2002: Short Contributions, CSL 2003: Extended Posters*, vol. VIII of *Kurt Gödel Society, Collegium Logicum*, pp. 17-33, 2004.

Avellone A., Fiorino G., Moscato U., “Optimization techniques for propositional intuitionistic logic and their implementation”, *Theoretical Computer Science*, vol. 409, num. 1, pp. 41–58, 2008.

Chagrov A., Zakharyaschev M., *Modal Logic*, Oxford University Press, 1997.

Dyckhoff R., "Contraction-free sequent calculi for intuitionistic logic", *Journal of Symbolic Logic*, vol. 57, num. 3, pp. 795-807, 1992.

Fiorino G., Decision procedures for propositional intermediate logics, PhD thesis, Dipartimento di Scienze dell'Informazione, Università degli Studi di Milano, Italy, 2001.

Fitting M., *Intuitionistic Logic, Model Theory and Forcing*, North-Holland, 1969.

Galmiche D., Larchey-Wendling D., "Structural Sharing and Efficient Proof-Search in Propositional Intuitionistic Logic", *ASIAN'99*, vol. 1742 of *LNCS*, pp. 101-112, 1999.

Gentzen G., "Investigations into logical deduction", *in* M. Szabo (ed.), *The Collected Works of Gerhard Gentzen*, North-Holland, pp. 68-131, 1969.

Hudelmaier J., Bounds for cut elimination in intuitionistic propositional logic, PhD thesis, Universität Tubingen, 1989.

Hudelmaier J., "An $O(n \log n)$-SPACE decision procedure for intuitionistic propositional logic", *Journal of Logic and Computation*, vol. 3, num. 1, pp. 63-75, 1993.

Miglioli P., Moscato U., Ornaghi M., "Avoiding duplications in tableau systems for intuitionistic logic and Kuroda logic", *Logic Journal of the IGPL*, vol. 5, num. 1, pp. 145–167, 1997.

Raths T., Otten J., Kreitz C., "The ILTP Problem Library for Intuitionistic Logic", *JAR*, vol. 31, pp. 261-271, 2007.

Statman R., "Intuitionistic logic is polynomial-space complete", *Theoretical Computer Science*, vol. 9, num. 1, pp. 67-72, 1979.

Vorob'ev N. N., "A new algorithm of derivability in a constructive calculus of statements", *Sixteen papers on logic and algebra*, vol. 94 of *American Mathematical Society Translations, Series 2*, American Mathematical Society, Providence, R.I., pp. 37-71, 1970.

Waaler A., Wallen L., "Tableaux for Intuitionistic Logics", *in* M. D'Agostino, D. Gabbay, R. Hähnle, J. Posegga (eds), *Handbook of tableau methods*, Kluwer academic publisher, pp. 255-296, 1999.

Downloaded by [University Degli Studi Dell'Insurbria], [Mauro Ferrari] at 01:24 14 April 2012