Translating Tableaux into Natural Deduction: Applications to Theorem Proving

Alessandro Avellone Marco Benini

Dipartimento di Scienze dell'Informazione
Università degli Studi di Milano
via Comelico 39, 20135 Milano–Italy
E-mail: {avellone,benini}@dsi.unimi.it

Ugo Moscato

Dipartimento di Metodi Quantitativi per l'Economia
Università degli Studi di Milano–Bicocca
Piazza dell'Ateneo Nuovo 1, 20126 Milano–Italy
E-mail: moscato@dsi.unimi.it

October 27, 1999

Abstract

In this paper we show a series of results on intuitionistic tableau, when applied to automatic theorem proving. In particular we will show that it is possible to directly translate a tableau proof into a natural deduction proof. We will apply this result to validate the answers coming from an automatic proving procedure which adopts a non-sound variant of the tableau calculus for intuitionistic logic.

The distinctive character of our approach lies in the usage we make of mappings between different calculi for the same logics. In this paper, we show and discuss the intuitionistic case, where a non-sound variant of the tableau calculus produces certified answers. There are classes of formulas, usually considered as difficult to prove, which are derived in a surprisingly small amount of time by our prover. We conclude our work with a partial introduction to the classes of formulas which can be proven in an efficient way by our technique.

1 Introduction

In this paper we present a method to convert tableau proofs into natural-like derivations. We will show an abstract algorithm to translate closed tableaus for intuitionistic first-order predicate logic into Prawitz's natural deduction calculus [23] for the same logic. Originally, this problem was posed in [32] by L. Wos for resolution-based theorem provers. As P. Andrews said in [1], "substantial work has been done not only on such translations, but also on improving the structure of natural deduction proofs and translating them into natural language". In fact, there are many other approaches which could resemble ours, but they are different in the choice of the starting calculus, as described in [4, 5, 12, 22, 25] and in [26, 27, 28] where the authors have developed an uniform procedure for transforming classical and non classical matrix proofs into sequent style proofs. Moreover, a closely related topic is the comparison between tableau systems and resolution-based methods, see, e.g., [18].

The distinctive character of our approach lies in the way we use the translation algorithm. Essentially, we do not require the starting tableau calculus be sound. This allows to use the translation algorithm to certify the answers from an automatic tableau prover; where the answer 'no' of the prover, when the input is a theorem, can occur either because the implementation of the calculus is incorrect or because, to improve performances, the calculus is not sound.

Let us suppose to have an environment E where we can develop proofs step by step in natural deduction, and let us suppose that this environment is correct. Now, let P be an automatic theorem prover, which, given a formula ϕ, returns "False" if it is not able to prove ϕ, and "True" plus a tableau T for ϕ if it is able to construct T. We do not assume that P is correct, i.e., that whenever P returns "True" and T, ϕ is a theorem and T is a proof for ϕ. If $\textsc{Tr}$ is a translation algorithm, because it is formally correct, $\textsc{Tr}(T)$ is a proof in E if and only if T is correct, i.e., if ϕ is a theorem. In this way the system $E + P + \textsc{Tr}$ provides reliable answers, even if P may be non-correct. Later, we will show that it makes sense to use provers which are not correct, so we can avoid significant computational costs without restricting the set of provable formulas.

In practice, we developed a package for the ISABELLE logical framework [20], where an automatic proving procedure for intuitionistic first-order logic is used to (partially) decide the validity of goals; an implementation of the translation algorithm is coupled to that prover in order to guarantee that every positive answer is trustworthy. The whole package is available in beta version at **ftp://dotto.usr.dsi.unimi.it/~logic/tableau.tar.gz**.

In the real implementation, the translation algorithm constructs a *tactic*, i.e., a function which sequentially applies elementary inference rules; in this way, even a non-closed tableau can be converted in a partial proof. The correctness of this variant of the translation algorithm follows from the propositions we will derive in Section 3.

The approach we are proposing, i.e., to use an external prover and, then, to certify the result it provides inside the ISABELLE framework, seems to be less effective than to embed the tableau calculus into ISABELLE's logic. While the second way has been successfully employed in many case [9, 10], we believe that it will not be the case with our particular tableau calculus. There are many reason for our belief: since the **IL**-T calculus uses signed formulas, it is difficult to restate its inference rules in natural deduction avoiding applicability clashes, which result in unnecessary backtracking; moreover, ISABELLE's representation of formulas is more complex than the one adopted inside the external prover, causing an unnecessary waste of space. As a result, we use as much as needed of the system resources when reconstructing the proof in ISABELLE, while we consume resources when searching for proof, but keeping this usage as low as possible, according to the nature of the **IL**-T calculus and the search strategy the tableau prover adopts.

We choose the tableau calculus developed in [16]. Our choice is motivated by the problem of simplifying the search for proofs by reducing the amount of *duplications*, where a duplication occur in a proof of a tableau calculus whenever a formula already used by an inference rule is used again by the same rule (for a comprehensive discussion about duplications, see [16]). A quite similar problem has been taken into account on the side of sequent calculi, where the counterpart of duplication is the *elimination of contractions*. There, in the intuitionistic framework important results have been obtained by Dyckhoff [6] and, independently, by Hudelmaier [11], who have exhibited cut-free and *contraction* free sequent calculi for Intuitionistic Propositional Logic, where a sequent calculus is contraction-free if no formula occurring in the lower sequent of an inference rule (i.e., the sequent obtained by applying the rule) can occur in some of the upper sequents (i.e., the sequents to which the rule is applied); in [6] also a contraction-free natural calculus is given, while in [11] it is shown that the involved calculus gives rise to an $O(n \log n)$-SPACE decision procedure for Intuitionistic Propositional Logic. The tableau calculus of [16] (which is a refinement of those in [14, 15]) has improved Fitting's tableau calculus for Intuitionistic Predicate Logic [8] by reducing the amount of duplications involved in its proofs. To do so, in [16] the $\mathbf{F_c}$-signed formulas have been introduced near the **F**-signed and the **T**-signed formulas of Fitting's calculus (see the discussion in [14]), and the rules for implication have been refined, taking into account the ideas of Dyckhoff's paper. Thus, the tableau calculus for Intuitionistic Predicate Logic in [16], is nearly optimal from the point of view of the elimination of duplications (such a calculus completely avoids duplications in the propositional framework, thus providing, for tableau calculi, a result comparable with Dyckhoff's and Hudelmaier's one for sequent calculi; on the other hand, at the predicate level, not all duplications, or *mutatis mutandis*, not all contractions, can be cut off).

Moreover, we have modified the tableau calculus in [16] in a way that the tableau rule connected with the quantifiers are not sound, since they do not handle in the proper way the binding of variables (it is a problem very close to Prolog implementations which avoid the occur-check [13]). This variation of our tableau calculus is more efficient, since it does not require so many duplications as the sound calculus, but, of course, it may produce closed tableaus which are not valid **IL**-T proof tables. Our approach is to validate a closed tableau by translating it into an **IL** proof. If this is possible, we have been able to prove in an efficient way a, possibly difficult, theorem; if this is not possible, it means that we applied in a non-sound way an expansion rule, so we are not able to judge whether the goal is a theorem.

The paper has the following structure: the next section contains the set of definitions and results we need in order to introduce our translation algorithm. Section 3 shows the translation algorithm and proves its correctness. Section 4 describes a non sound variation for **IL**-T and discusses its usage in theorem proving. In the last part we draw some conclusions, expanding the previous discussion on the correctness of a theorem prover.

2 Preliminary Definitions

The set of *well formed formulas* (*wff's* for short) is defined in the usual way, starting from the propositional connectives $\neg$, $\wedge$, $\vee$, $\rightarrow$, the quantifiers $\forall$ and $\exists$, a denumerable set $\mathcal{V}$ of individual variables, and, for any natural number n, a denumerable set $\mathcal{P}^n$ of n-ary predicate variables.

The theory **IL** denotes the set of all intuitionistically valid wff's. Our choice for a natural deduction system is the calculus **Ni** as shown in [31]; for reference, we report in Table 1 the complete set of inference rules. Since in [30], the **Ni** calculus is proven to be sound and complete, we will refer to it as **IL**.

$$\frac{A \quad B}{A \wedge B}\,\wedge\mathrm{I} \qquad \frac{A \wedge B}{A}\,\wedge\mathrm{E}_l \qquad \frac{A \wedge B}{B}\,\wedge\mathrm{E}_r$$

$$\frac{A}{A \vee B}\,\vee\mathrm{I}_l \qquad \frac{B}{A \vee B}\,\vee\mathrm{I}_r \qquad \frac{A \vee B \quad \begin{array}{c}[A]\\ \prod \\ C\end{array} \quad \begin{array}{c}[B]\\ \prod \\ C\end{array}}{C}\,\vee\mathrm{E}$$

$$\frac{\begin{array}{c}[A]\\ \prod \\ B\end{array}}{A \rightarrow B}\,\rightarrow\mathrm{I} \qquad \frac{A \quad A \rightarrow B}{B}\,\rightarrow\mathrm{E} \qquad \frac{\bot}{A}\,\bot\mathrm{E}$$

$$\frac{A(p)}{\forall\, x.\, A(x)}\,\forall\mathrm{I}\ (*) \qquad \frac{\forall\, x.\, A(x)}{A(t)}\,\forall\mathrm{E}$$

$$\frac{A(t)}{\exists\, x.\, A(x)}\,\exists\mathrm{I} \qquad \frac{\exists\, x.\, A(x) \quad \begin{array}{c}[A(p)]\\ \prod \\ B\end{array}}{B}\,\exists\mathrm{E}\ (**)$$

where, in $(*)$ and $(**)$, p is an eigenvariable.

Table 1: Inference rules for **IL**

A *(predicate) Kripke model* is a quadruple $\underline{K} = \langle P, \leq, \Vdash, \mathcal{D} \rangle$, where $\underline{P} = \langle P, \leq \rangle$ is a partial ordered set, called the *frame*; $\mathcal{D}$ is the *domain function*, associating, to any $\alpha \in P$, a domain $\mathcal{D}(\alpha)$ such that, for any $\alpha, \beta \in P$, if $\alpha \leq \beta$ then $\mathcal{D}(\alpha) \subseteq \mathcal{D}(\beta)$; $\Vdash$ is the *forcing relation*, defined between elements of P and atomic wff's with parameters denoting the individuals in the domains, and such that, for any atomic wff $p(x_1, \ldots, x_n)$, for any $\alpha, \beta \in P$ and any $a_1, \ldots, a_n \in \mathcal{D}(\alpha)$, if $\alpha \leq \beta$ and $\alpha \Vdash p(a_1, \ldots, a_n)$ then $\beta \Vdash p(a_1, \ldots, a_n)$. The forcing relation is extended to arbitrary closed wff's (with parameters denoting elements of the domains) in the usual way (see [7, 29]).

The tableau calculus for Intuitionistic Predicate Logic we are going to use is defined in [14, 15, 16] and it uses the three signs $\mathbf{T}$, $\mathbf{F}$ and $\mathbf{F_c}$. Given a wff A, a *signed well formed formula* (*swff* for short) will be every expression of the kind $\mathcal{S}A$, where $\mathcal{S} \in \{\mathbf{T}, \mathbf{F}, \mathbf{F_c}\}$.

The meaning of the signs $\mathbf{T}$, $\mathbf{F}$ and $\mathbf{F_c}$ is explained in terms of *realizability* as follows: given a model $\underline{K} = \langle P, \leq, \Vdash, \mathcal{D} \rangle$, an element $\alpha \in P$, and a swff H whose parameters denote elements of $\mathcal{D}(\alpha)$, we say that α *realizes* H *(in* $\underline{K}$*)*, and we write $\alpha \rhd H$ (in $\underline{K}$), if the following conditions hold:

1. If $H \equiv \mathbf{T}A$, then $\alpha \Vdash A$;
2. If $H \equiv \mathbf{F}A$, then $\alpha \nVdash A$;
3. If $H \equiv \mathbf{F_c}A$, then $\alpha \Vdash \neg A$.

We say that α *realizes a set of swff's* S (and we write $\alpha \rhd S$) iff α realizes every swff in S. A set of swff's S is *realizable* iff there is some element α of a Kripke model $\underline{K}$ such that $\alpha \rhd S$. By a *configuration* we mean a finite sequence $S_1|S_2|\ldots|S_n$ (with $n \geq 1$), where every S_j is a set of swff's. A set of swff's of a configuration is called *node.* A *configuration is realizable* iff at least an S_j is realizable.

The intuitionistic tableau calculus **IL**-T is given by the rules in Table 2, where S_c is the *certain part of* S, formally,

$$S_c = \{\mathbf{T}X \mid \mathbf{T}X \in S\} \cup \{\mathbf{F_c}X \mid \mathbf{F_c}X \in S\} \ .$$

Definition 1 *Let S be a set of swff's, a* tableau *(or* **IL**-proof-table*) for S is a finite sequence of configurations $C_1, \ldots, C_n$ such that*

- *$C_1 = \{S\}$, and*
- *$C_{i+1} = \{S_1, \ldots, S_n\} \cup \{N_1, \ldots, N_k\}$, where $C_i = \{S_1, \ldots, S_n\} \cup \{M\}$ and*

$$\frac{M}{N_1 \mid \ldots \mid N_k}$$

  is an instance of an expansion rule as in Table 2.

The active *wff in a node is the swff an expansion rule is applied to, the other swff's in that node are called the* context. *A node in the last configuration (the* terminal configuration*) of a tableau is called a* terminal node.

$\mathbf{T}$ – rules	$\mathbf{F}$ – rules	$\mathbf{F_c}$ – rules
$\dfrac{S, \mathbf{T}(A \wedge B)}{S, \mathbf{T}A, \mathbf{T}B}\ \mathbf{T}\wedge$	$\dfrac{S, \mathbf{F}(A \wedge B)}{S, \mathbf{F}A \mid S, \mathbf{F}B}\ \mathbf{F}\wedge$	$\dfrac{S, \mathbf{F_c}(A \wedge B)}{S_c, \mathbf{F_c}A \mid S_c, \mathbf{F_c}B}\ \mathbf{F_c}\wedge$
$\dfrac{S, \mathbf{T}(A \vee B)}{S, \mathbf{T}A \mid S, \mathbf{T}B}\ \mathbf{T}\vee$	$\dfrac{S, \mathbf{F}(A \vee B)}{S, \mathbf{F}A, \mathbf{F}B}\ \mathbf{F}\vee$	$\dfrac{S, \mathbf{F_c}(A \vee B)}{S, \mathbf{F_c}A, \mathbf{F_c}B}\ \mathbf{F_c}\vee$
See special table for $\mathbf{T}\to$	$\dfrac{S, \mathbf{F}(A \to B)}{S_c, \mathbf{T}A, \mathbf{F}B}\ \mathbf{F}\to$	$\dfrac{S, \mathbf{F_c}(A \to B)}{S_c, \mathbf{T}A, \mathbf{F_c}B}\ \mathbf{F_c}\to$
$\dfrac{S, \mathbf{T}(\neg A)}{S, \mathbf{F_c}A}\ \mathbf{T}\neg$	$\dfrac{S, \mathbf{F}(\neg A)}{S_c, \mathbf{T}A}\ \mathbf{F}\neg$	$\dfrac{S, \mathbf{F_c}(\neg A)}{S_c, \mathbf{T}A}\ \mathbf{F_c}\neg$
$\dfrac{S, \mathbf{T}(\forall\, x.\, A(x))}{S, \mathbf{T}A(a), \mathbf{T}(\forall\, x.\, A(x))}\ \mathbf{T}\forall$	$\dfrac{S, \mathbf{F}(\forall\, x.\, A(x))}{S_c, \mathbf{F}A(a)}\ \mathbf{F}\forall$ with a new	$\dfrac{S, \mathbf{F_c}(\forall\, x.\, A(x))}{S_c, \mathbf{F_c}A(a), \mathbf{F_c}(\forall\, x.\, A(x))}\ \mathbf{F_c}\forall$ with a new
$\dfrac{S, \mathbf{T}(\exists\, x.\, A(x))}{S, \mathbf{T}A(a)}\ \mathbf{T}\exists$ with a new	$\dfrac{S, \mathbf{F}(\exists\, x.\, A(x))}{S, \mathbf{F}A(a)}\ \mathbf{F}\exists$	$\dfrac{S, \mathbf{F_c}(\exists\, x.\, A(x))}{S, \mathbf{F_c}A(a), \mathbf{F_c}(\exists\, x.\, A(x))}\ \mathbf{F_c}\exists$

Rules for $\mathbf{T}\to$	
$\dfrac{S, \mathbf{T}(A \to B)}{S, \mathbf{F}A \mid S, \mathbf{T}B}\ \mathbf{T}{\to}a\neg$ with A atomic or negated	$\dfrac{S, \mathbf{T}((A \vee B) \to C)}{S, \mathbf{T}(A \to C), \mathbf{T}(B \to C)}\ \mathbf{T}{\to}\vee$
$\dfrac{S, \mathbf{T}((A \wedge B) \to C)}{S, \mathbf{T}(A \to (B \to C))}\ \mathbf{T}{\to}\wedge$	$\dfrac{S, \mathbf{T}((A \to B) \to C)}{S, \mathbf{F}(A \to B), \mathbf{T}(B \to C) \mid S, \mathbf{T}C}\ \mathbf{T}{\to}{\to}$
$\dfrac{S, \mathbf{T}((\exists\, x.\, A(x)) \to B)}{S, \mathbf{T}(\forall\, x.\, (A(x) \to B))}\ \mathbf{T}{\to}\exists$	$\dfrac{S, \mathbf{T}((\forall\, x.\, A(x)) \to B)}{S, \mathbf{F}(\forall\, x.\, A(x)), \mathbf{T}((\forall\, x.\, A(x)) \to B) \mid S, \mathbf{T}B}\ \mathbf{T}{\to}\forall$

Table 2: Expansion rules for intuitionistic tableaus

A tableau is *closed* iff all the sets S_j of its final configuration are contradictory, where a set S is *contradictory* if (for some A) either $\{\mathbf{T}A, \mathbf{F}A\} \subseteq S$ or $\{\mathbf{T}A, \mathbf{F_c}A\} \subseteq S$. Accordingly, we call *complementary pair* any set of swff's of the form $\{\mathbf{T}A, \mathbf{F}A\}$ or of the form $\{\mathbf{T}A, \mathbf{F_c}A\}$. It is immediate to verify that if S is a set of swff's and contains a complementary pair, then S is not realizable. A *proof* of a wff B in **IL** is a closed tableau whose initial node is $\{\mathbf{F}B\}$. Finally, we say that a set of swff's S is **IL**-*consistent* iff no tableau starting from S is closed.

Every rule of the **IL**-T tableau calculus is applied to a swff of a set S_i occurring in a configuration $S_1 \mid \cdots \mid S_i \mid \ldots$; the notation $S, \mathbf{T}(A \wedge B)$ points out that the rule $\mathbf{T}\wedge$ is applied to the swff $\mathbf{T}(A \wedge B)$ of the set $S \cup \{\mathbf{T}(A \wedge B)\}$, where S is possibly empty. We remark that all the rules of the **IL**-T calculus, excepting $\mathbf{T}\forall$, $\mathbf{F}\exists$, $\mathbf{F_c}\exists$, $\mathbf{F_c}\forall$ and $\mathbf{T}\rightarrow\forall$, are duplication-free, in the sense explained in [2, 3, 6, 14, 15, 16].

In [14, 15, 16], it is proved that **IL**-T is sound and complete according to the standard intuitionistic semantics based on Kripke models.

3 Translating Tableau Proofs into Natural Deduction

In this section, we introduce the algorithm which translates tableaus into natural deduction proofs, and we prove its correctness. To do so, we define the concept of *proof with gaps*, and we use this notion to define a function TR, which maps each tableau into a proof which may contain non-specified parts, the gaps. Finally we show how we can remove these gaps when the original tableau is closed.

Definition 2 *The* intuitionistic natural deduction calculus with gaps **ILG** *is defined from the inference rules in Table 1 plus*

$$\frac{\gamma_1 \cdots \gamma_n}{\alpha}\,\mathrm{G} \ .$$

A proof in this system is a generalization of proofs in **IL** calculus; for this reason we will refer to deductions in this system as *proofs with gaps* (PG) in **IL**.

To indicate gaps into a PG-proof, we use the notation

$$\Xi(G_1, \ldots, G_n)$$

where $G_1, \ldots, G_n$ are the application of the G rule (the gaps, for short) we want to mark. When we write $\Xi(\Pi_1, \ldots, \Pi_n)$ we intend that the gaps $G_1, \ldots, G_n$ are substituted with the PG-proofs $\Pi_1, \ldots, \Pi_n$.

The intuitive meaning of a proof with gaps is that it is a "partial" proof, i.e., a proof where some parts (the gaps) are not yet developed, but their assumptions and conclusions are specified. We use a gap in a proof as a placeholder for another proof, which should be developed according to our translation algorithm.

Definition 3 *The natural deduction calculus* **ILGD** *is defined from* **ILG** *plus the rules in Appendix A.*

In Proposition 2, we will prove that the rules in Appendix A can be derived in **ILG**; hence **ILG** and **ILGD** are equivalent, because they prove the same set of wff's (but, from a practical point of view, following [21], **ILGD** is, by far, more efficient then **ILG**).

Definition 4 *The translation function* $\textsc{Tr}$ *maps a tableau* T *into a proof* $\textsc{Tr}(T)$ *in* **ILGD**; *the definition is given by induction on the structure of the tableau* T:

- *base case*

  Let $T = \{S\}$ *be a tableau consisting of the initial configuration; then*

$$\textsc{Tr}(T) = \frac{\{\phi \mid \mathbf{T}\,\phi \in S\} \cup \{\neg\phi \mid \mathbf{F_c}\,\phi \in S\}}{\bigvee\{\phi \mid \mathbf{F}\,\phi \in S\}}\,\textsc{g}$$

- *induction case*

  Let $C = \{S_1, \ldots, S_n\}$ *be the last configuration of* T, *obtained from a tableau* T' *and its terminal configuration* $C' = \{S'_1, \ldots, S'_m\}$ *by applying a tableau rule to the swff* α *in the node* $S'_j = \Gamma \cup \{\alpha\}$, *and generating as new nodes* $N_1, \ldots, N_k$.

  Let $\textsc{Tr}(T') = \Xi(G_{S'_j})$, *where* $G_{S'_j}$ *is the gap corresponding to* S'_j. *Then* $\textsc{Tr}(T) = \Xi(\textsc{Tr}(G_{S'_j}))$ *and* $\textsc{Tr}(G_{S'_j})$ *is an instance of a rule in Appendix A, according to the principal connective, the sign of* α, *and the number of* $\mathbf{F}$*-wff's in* S'_j.

We observe that every node in the terminal configuration of T has a corresponding gap in $\textsc{Tr}(T)$, and this correspondence defines what we intend for gap associated with a node.

The translation function $\textsc{Tr}$ is the way how we construct the tactics in ISABELLE, in the sense that $\textsc{Tr}(T)$ can be regarded as a sequence of applications of inference rules, and the gaps as intermediate subgoals to be proven. The next propositions and theorem will prove that if T is a closed tableau then we can combine those tactics to construct a proof in **IL**.

Proposition 1 *Let* T *be a tableau; for any gap* G *in* $\textsc{Tr}(T)$, *there is a node* S *in the terminal configuration of* T *such that* G *is associated with* S *and*

$$G \equiv \frac{\{\phi \mid \mathbf{T}\,\phi \in S\} \cup \{\neg\phi \mid \mathbf{F_c}\,\phi \in S\}}{\bigvee\{\phi \mid \mathbf{F}\,\phi \in S\}}\,\textsc{g}\ .$$

Proof: By induction on the structure of the tableau T. The base case, $T = \{S\}$, is trivial. So let's suppose T is derived from T' by expanding a terminal node S' according to the rules in Table 2 and let $\textsc{Tr}(T') = \Xi(G_{S'})$.

By induction hypothesis on T', for every gap G in $\textsc{Tr}(T')$, there exists S, terminal node in T' associated with G, such that

$$G \equiv \frac{\{\phi \mid \mathbf{T}\,\phi \in S\} \cup \{\neg\phi \mid \mathbf{F_c}\,\phi \in S\}}{\bigvee\{\phi \mid \mathbf{F}\,\phi \in S\}}\,\textsc{g}\ .$$

By definition, $\textsc{Tr}(T) = \Xi(\textsc{Tr}(G_{S'}))$, i.e., $\textsc{Tr}(T)$ is obtained from $\Xi(G_{S'})$, by replacing the G-rule corresponding to the gap $G_{S'}$ according to the rules in Appendix A. Thus it suffices to prove that the number of gaps introduced by $\textsc{Tr}(\textsc{Tr}(G_{S'}))$ is exactly equal to the splittings introduced by the tableau rule applied to S', and that the new gaps have the required shape.

The proof is by cases according to the rules in Appendix A. For the sake of simplicity we only treat the cases corresponding to the translation rules $\mathbf{T}\wedge$, $\mathbf{F_c}\wedge$, and $\mathbf{F_c}\wedge\bot$; the other cases are similar.

• Let $S' = \{\mathbf{T}A_1,\ldots,\mathbf{T}A_n, \mathbf{F}B_1,\ldots,\mathbf{F}B_m, \mathbf{F_c}C_1,\ldots,\mathbf{F_c}C_k, \mathbf{T}(H\wedge E)\}$ where the last configuration of T is obtained by expanding the swff $\mathbf{T}(H\wedge E)$ and let $G_{S'}$ be equal to

$$\frac{A_1,\ldots,A_n,\neg C_1,\ldots\neg C_k,(H\wedge E)}{B_1\vee\ldots\vee B_m}\,G_{S'}\ .$$

Then $\textsc{Tr}(G_{S'})$, corresponding to the $\mathbf{T}\wedge$ translation rule, is the following:

$$\dfrac{H\wedge E \qquad \dfrac{A_1,\ldots,A_n,\neg C_1,\ldots\neg C_k,H,E}{B_1\vee\ldots\vee B_m}\,\textsc{g}^*}{B_1\vee\ldots\vee B_m}\,\mathbf{T}\wedge$$

On the other hand, the tableau rule $\mathbf{T}\wedge$ transform the set S' into the set $S = \{\mathbf{T}A_1,\ldots,\mathbf{T}A_n,\ \mathbf{F}B_1,\ldots,\mathbf{F}B_m,\ \mathbf{F_c}C_1,\ldots,\mathbf{F_c}C_k, \mathbf{T}H, \mathbf{T}E\}$. Hence, the gap $\textsc{G}^*$ is associated with the set of swff's S and it has the required shape.

• Let $S' = \{\mathbf{T}A_1,\ldots,\mathbf{T}A_n,\ \mathbf{F}B_1,\ldots,\mathbf{F}B_m,\ \mathbf{F_c}C_1,\ldots,\mathbf{F_c}C_k, \mathbf{F_c}(H\wedge E)\}$, $\Gamma = \{A_1,\ldots,A_n\}$, $\Theta = \{\neg C_1,\ldots,\neg C_k\}$, and $D \equiv B_1\vee\ldots\vee B_m$ where last configuration of T is obtained by expanding the swff $\mathbf{F_c}(H\wedge E)$. Moreover, let $G_{S'}$ be equal to

$$\frac{\Gamma\cup\Theta\cup\{\neg(H\wedge E)\}}{D}\,G_{S'}$$

There are two cases, depending if D differs from $\bot$ or not; then $\textsc{Tr}(G_{S'})$, corresponding to the $\mathbf{F_c}\wedge$ translation rule or to the $\mathbf{F_c}\wedge\bot$ translation rule, respectively, are the following:

$$\cfrac{\neg(H \wedge E) \qquad \cfrac{\Gamma \cup \Theta, [H]}{\bot}\,{\scriptstyle G_1} \qquad \cfrac{\Gamma \cup \Theta, [E]}{\bot}\,{\scriptstyle G_2}}{D}\;{\scriptstyle \mathbf{F_c}\wedge} \qquad (\text{if } m \neq 0)$$

$$\cfrac{\neg(H \wedge E) \qquad \cfrac{\Gamma \cup \Theta, [H]}{\bot}\,{\scriptstyle G_1} \qquad \cfrac{\Gamma \cup \Theta, [E]}{\bot}\,{\scriptstyle G_2}}{\bot}\;{\scriptstyle \mathbf{F_c}\wedge\bot} \qquad (\text{if } m = 0)$$

On the other hand, $\mathbf{F_c}\wedge$-rule transform the set S' into $S_1 = \{\mathbf{T}A_1, \dots, \mathbf{T}A_n, \mathbf{F}B_1, \dots, \mathbf{F}B_m, \mathbf{F_c}C_1, \dots, \mathbf{F_c}C_k, \mathbf{F_c}H\}$ and into the set $S_2 = \{\mathbf{T}A_1, \dots, \mathbf{T}A_n, \mathbf{F}B_1, \dots, \mathbf{F}B_m, \mathbf{F_c}C_1, \dots, \mathbf{F_c}C_k, \mathbf{F_c}E\}$. Hence, the gaps G_1 and G_2 are associated with the sets of swff's S_1 and S_2, respectively, and their shapes are as required. □

Proposition 2 *The rules in Appendix A can be derived in* **ILG**.

Proof: For the sake of simplicity we treat only the cases corresponding to the translation rules $\mathbf{T}\wedge$, $\mathbf{F_c}\wedge$, and $\mathbf{F_c} \wedge \bot$; the other cases are similar. Hence,

$$\cfrac{A \wedge B \qquad \cfrac{\Gamma, [A], [B]}{D}\,{\scriptstyle G}}{D}\;{\scriptstyle \mathbf{T}\wedge}$$

becomes

$$\cfrac{\Gamma \quad \cfrac{A \wedge B}{A}\,{\scriptstyle \wedge E} \qquad \cfrac{A \wedge B}{B}\,{\scriptstyle \wedge E}}{D}\;{\scriptstyle G}$$

Similarly,

$$\cfrac{\cfrac{\Gamma}{D \vee A}\,{\scriptstyle G_1} \qquad \cfrac{\Gamma}{D \vee B}\,{\scriptstyle G_2}}{D \vee (A \wedge B)}\;{\scriptstyle \mathbf{F}\wedge}$$

is expanded into

$$\cfrac{\cfrac{\Gamma}{D \vee A}\,{\scriptstyle G_1} \qquad \cfrac{[D]_2}{D \vee (A \wedge B)}\,{\scriptstyle \vee I} \qquad \cfrac{\cfrac{\Gamma}{D \vee B}\,{\scriptstyle G_2} \qquad \cfrac{[D]_1}{D \vee (A \wedge B)}\,{\scriptstyle \vee I} \qquad \cfrac{\cfrac{[A]_2 \quad [B]_1}{A \wedge B}\,{\scriptstyle \wedge I}}{D \vee (A \wedge B)}\,{\scriptstyle \vee I}}{D \vee (A \wedge B)}\,{\scriptstyle \vee E_1}}{D \vee (A \wedge B)}\;{\scriptstyle \vee E_2}$$

The following rule

$$\dfrac{\neg(A \wedge B) \qquad \dfrac{\Gamma, [\neg A]}{\bot}\,{\scriptstyle G_1} \qquad \dfrac{\Gamma, [\neg B]}{\bot}\,{\scriptstyle G_2}}{D}\;{\scriptstyle \mathbf{F_c}\wedge}$$

is translated as

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[A]_1 \quad [B]_2}{A \wedge B}\,{\scriptstyle \wedge I} \qquad \neg(A \wedge B)}{\bot}\,{\scriptstyle \to E}}{\neg B}\,{\scriptstyle \to I_2}}{\bot}\,{\scriptstyle G_2}}{\neg A}\,{\scriptstyle \to I_1}}{\bot}\,{\scriptstyle G_1}}{D}\,{\scriptstyle \bot E}$$

Finally, if the conclusion is $\bot$, as in the following rule

$$\dfrac{\neg(A \wedge B) \qquad \dfrac{\Gamma, [\neg A]}{\bot}\,{\scriptstyle G_1} \qquad \dfrac{\Gamma, [\neg B]}{\bot}\,{\scriptstyle G_2}}{\bot}\;{\scriptstyle \mathbf{F_c}\wedge\bot}$$

we can expand it into a more compact form

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[D]_1 \quad [E]_2}{D \wedge E}\,{\scriptstyle \wedge I} \qquad \neg(A \wedge B)}{\bot}\,{\scriptstyle \to E}}{\neg B}\,{\scriptstyle \to I_2}}{\bot}\,{\scriptstyle G_2}}{\neg A}\,{\scriptstyle \to I_1}}{\bot}\,{\scriptstyle G_1}$$

□

Using Propositions 1 and 2 we obtain immediately the following corollary.

Corollary 1 *Let T be a tableau; then* $\text{TR}(T)$ *can be translated in a* PG*-proof.*

In the previous proposition we proved that every rule in Appendix A is derivable in **ILG**. We note that some rules (e.g., $\mathbf{F_c}\wedge\bot$) are instances of others (e.g., $\mathbf{F_c}\wedge$); the reason behind this apparent duplication of rules is that they have a shorter, more "natural" proof. Since the proof of a derived rule becomes part of the proof we generate through TR, it makes sense to have simpler and more natural proofs, so to enhance their readability.

We observe that the way of composing gaps in the expansion of a translation rule may not respect the shape of the rule itself, as in the correctness proof of $\mathbf{F_c}\wedge$. The translation rules are designed in such a way that their applications reflect the content of the corresponding node in the tableau.

Since **ILG** and **ILGD** are essentially the same calculi, then, without loss of generality we can suppose that TR is a function mapping every tableau T into a proof in **ILG**.

Definition 5 *Let T be a tableau and let $\mathrm{TR}(T) = \Xi(G_{S_1}, \ldots, G_{S_n})$ where all the gaps are put into evidence. The* Gap closure *of $\mathrm{TR}(T)$, $\Pi_c(\mathrm{TR}(T))$ is the PG-proof obtained from $\mathrm{TR}(T)$ by substituting the rules in Appendix A with their derivations according to Proposition 2 and by replacing the gaps $G_{S_1}, \ldots, G_{S_n}$ with PG-proofs according the following conventions:*

1. *if S_i is $\mathbf{F}$-closed, then*

$$G_{S_i} \equiv \frac{A_1, \ldots, A_n, \neg C_1, \ldots \neg C_k, D}{B_1 \vee \ldots \vee B_m \vee D}\,\mathrm{G}$$

is replaced with

$$\cfrac{\cfrac{\cfrac{D}{B_1 \vee D}\,\vee\mathrm{I}}{B_1 \vee B_2 \vee D}\,\vee\mathrm{I}}{\begin{array}{c}\vdots\,\vee\mathrm{I}\\ B_1 \vee \ldots \vee B_m \vee D\end{array}} \quad \textit{(if } m \neq 0\textit{)} \qquad\qquad D \quad \textit{(if } m = 0\textit{)}$$

2. *if S_i is $\mathbf{F_c}$-closed, then*

$$G_{S_i} \equiv \frac{A_1, \ldots, A_n, \neg C_1, \ldots \neg C_k, D, \neg D}{B_1 \vee \ldots \vee B_m}\,\mathrm{G}$$

is replaced with

$$\cfrac{\cfrac{D \quad \neg D}{\bot}\,\to\mathrm{E}}{B_1 \vee \ldots \vee B_m}\,\bot\mathrm{E} \quad \textit{(if } m \neq 0\textit{)} \qquad \cfrac{D \quad \neg D}{\bot}\,\to\mathrm{E} \quad \textit{(if } m = 0\textit{)}$$

3. *If S_i is not closed, then the corresponding gap is left unchanged.*

We observe that closing a gap G in the sense of the previous definition, may cancel antecedents of G which are not essential to complete the proof. It is obvious that Π_c is a function from PG-proofs to PG-proofs. Now, we have to prove that, if T is a closed tableau and P is the result of translating it into **ILG**, then $\Pi_c(P)$ is, indeed, a proof in the **IL** system, i.e., it contains no gaps.

Theorem 1 *Let T be a closed tableau starting from a set S of swff's. Then $\Pi_c(\text{Tr}(T))$ is an* **IL***-proof of*

$$\{\phi \mid \mathbf{T}\,\phi \in S\} \cup \{\neg\phi \mid \mathbf{F_c}\,\phi \in S\} \vdash_{\mathbf{IL}} \bigvee\{\phi \mid \mathbf{F}\,\phi \in S\}$$

Proof: By Propositions 1 and 2 , the gaps belonging to the PG-proof $\text{Tr}(T)$ are those associated to the terminal nodes of T. Since T is closed, all gaps in $\text{Tr}(T)$ are replaced with proofs without gaps in $\Pi_c(\text{Tr}(T))$, hence, $\Pi_c(\text{Tr}(T))$ is a PG-proof without gaps, that is, an **IL**-proof. □

4 Non-Sound Tableau

In the previous sections we introduced **IL**-T, a tableau calculus which is sound and complete, and an algorithm, Tr, which maps **IL**-T proofs to natural deduction derivations.

Our claim is that, considering an implementation P of a prover for **IL**-T, we can certify its answers using an implementation of Tr. Actually, we coded such a system as an external prover (oracle) for the Isabelle logical framework. The key idea is that we do not need to prove that the prover P and that the translation procedure are correctly implemented, i.e., no formal verification of the system is needed since we can guarantee that its answers are as reliable as the core Isabelle inference engine.

A natural question which arises at this point is: if our prover is not correct on purpose, because it implements a non-sound variant of **IL**-T, what happens? Does it makes sense at all to study such non-sound variants?

We think that it can be very interesting to code a non-sound theorem prover and to use our translation algorithm to discriminate between good and bad answers. In this respect, we are going to describe a variant of **IL**-T which uses *dummy* variables [17, 24]; this modification of the original **IL**-T calculus is not sound in the usual sense, i.e., although not everything is provable, some "proofs" do not respect the usual semantics of logical operators: for an example, see subsection 4.1.

Let Ξ be a denumerable set of symbols, called *dummy variables*, and consider the tableau calculus **IL**-TDummy given by the expansion rules in Table 2 where $\mathbf{T}\forall$, $\mathbf{F}\exists$ and $\mathbf{F_c}\exists$ are replaced by the following:

$$\frac{S, \mathbf{T}(\forall\, x.\, A(x))}{S, \mathbf{T}A(\alpha), \mathbf{T}(\forall\, x.\, A(x))}\ \mathbf{T}\forall\text{Dummy}$$

$$\frac{S, \mathbf{F}(\exists\, x.\, A(x))}{S, \mathbf{F}A(\alpha)}\ \mathbf{F}\exists\text{Dummy}$$

$$\frac{S, \mathbf{F_c}(\exists\, x.\, A(x))}{S, \mathbf{F_c}A(\alpha), \mathbf{F_c}(\exists\, x.\, A(x))}\ \mathbf{F_c}\exists\text{Dummy}$$

where $\alpha \in \Xi$. The rule which detects if a node is closed is: a node N is a tableau T is closed if there is a formula ϕ and a formula ψ such that

- $\mathbf{F}\phi \in N$ or $\mathbf{F_c}\phi \in N$,
- $\mathbf{T}\psi \in N$,
- for a proper substitution σ of dummy variables, $\sigma\,\phi = \sigma\,\psi$.

So a strategy for developing a tableau in **IL**-TDummy would be to iterate the following steps:

1. choose an open node N in the tableau T; if there is no open node, return T (proof discovered);
2. check if there is an unifiable complementary pair in N;
3. if so, mark N as closed, and apply the unifier over the whole tableau T;
4. if not, expand N using a rule;
5. if this is not possible, by backtracking try alternatives; if there are no alternatives, return a failure.

From a very practical point of view, this expansion strategy has some advantages; in fact, we delay the choice for a witness term in the expansion rules which require one. Sometimes (most of the time, from empirical measures) this fact compensate the disadvantage of running an unification algorithm, because we have the "right" witness in the "right" moment instead of generating mostly useless instances, as required in the **IL**-T calculus. In fact, the usage of dummy variables is very similar to the use of metavariables in ISABELLE [19], although we do not take care to distinguish between free and bound occurrences.

The actual implementation of our prover for **IL**-TDummy uses an adaption of the standard Prolog unification algorithm [13], which, as it is well known, is non correct since it avoids occur-checks. This fact plus that we do not take care to respect eigenvariables during the unification process, lead us to a non sound but reasonably efficient theorem prover, which seems to be able to prove very quickly a great amount of not too complex goals. Being an instrument of a logical framework, the **IL**-TDummy prover is perfectly adequate for an interactive usage [21].

An important remark about using a non-sound calculus regards the translation function TR. Since an **IL**-TDummy proof table which respects eigenvariables is a closed **IL**-T tableau, we are allowed to use TR as described before to translate tableaus into tactics.

If we generate a new variant V for a tableau calculus C whose V-proof tables cannot be interpreted as those of the original calculus C, we are forced to define a new translation function τ. In this case we must be able to prove that the τ function is correct, that is, it maps a correct tableau for C into **IL**. Using the intermediate result in this proof, one the first side, we can instruct the ISABELLE system about the facts which are needed to translate proof in the C calculus, and, on the other side, we have a general design schema for the code of a translation function. Then, we can code a theorem prover for the V calculus, and, applying to its results the implemented τ function, we get answers which enjoy the same properties as for **IL**-TDummy versus **IL**.

4.1 An Illustrating Example

The formula $\neg\neg\exists\, y.\,\forall\, x.\, P(y) \to P(x)$ is a theorem in **IL**. Its proof in our tableau system require one duplication.

When we try to prove this goal with our non-sound prover, which uses dummy variables, it generates an incorrectly closed tableau. Denoting dummy variables with lowercase greek letters, and deleting unnecessary duplicated formulas, the wrong tableau is as follows:

$$
\begin{array}{c}
\mathbf{F}\neg\neg\exists\, y.\,\forall\, x.\, P(y) \to P(x) \\ \hline
\mathbf{T}\neg\exists\, y.\,\forall\, x.\, P(y) \to P(x) \\ \hline
\mathbf{F_c}\exists\, y.\,\forall\, x.\, P(y) \to P(x) \\ \hline
\mathbf{F_c}\forall\, x.\, P(\alpha) \to P(x) \\ \hline
\mathbf{F}P(\alpha) \to P(t) \\ \hline
\mathbf{T}P(\alpha), \mathbf{F}P(t) \\ \hline\hline
\text{closed with } \alpha \equiv t.
\end{array}
$$

It is immediate to check that this tableau could not be generated in the original calculus **IL**-T.

When we apply our translation algorithm to this tableau, we obtain an ISABELLE tactic Tac, that means a sequence of inference steps. Applying Tac to the original goal produces a failure. In order to show why ISABELLE rejects this tactic, we can write down the natural-like proof which would be the result of applying Tac to the goal:

$$
\dfrac{\dfrac{[\neg\exists\, y.\,\forall\, x.\, P(y) \to P(x)] \qquad \dfrac{[\neg\forall\, x.\, P(t) \to P(x)] \quad \dfrac{[P(t)]}{P(t) \to P(t)}\,\mathbf{F}\!\to}{\bot}\,\mathbf{F_c}\forall}{\bot}\,\mathbf{F_c}\exists}{\neg\neg\exists\, y.\,\forall\, x.\, P(y) \to P(x)}\,\mathbf{F}\neg
$$

This *proof* violates the conditions on eigenvariables in the application of the $\mathbf{F_c}\forall$ rule: in fact the second occurrence of t in $P(t) \to P(t)$ should be an eigenvariable, so ISABELLE rejects this step, and the (wrong) tableau could not be certified.

For the sake of completeness, we present also the right tableau, which is generated by the version of our prover, which does not make use of dummy variables; as usual, we omit duplicated formulas when not needed:

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\mathbf{F}\neg\neg\exists\, y.\,\forall\, x.\, P(y) \to P(x)}{\mathbf{T}\neg\exists\, y.\,\forall\, x.\, P(y) \to P(x)}}{\mathbf{F_c}\exists\, y.\,\forall\, x.\, P(y) \to P(x)}}{\mathbf{F_c}\forall\, x.\, P(a) \to P(x), \mathbf{F_c}\exists\, y.\,\forall\, x.\, P(y) \to P(x)}}{\mathbf{F}P(a) \to P(b), \mathbf{F_c}\exists\, y.\,\forall\, x.\, P(y) \to P(x)}}{\mathbf{T}P(a), \mathbf{F}P(b), \mathbf{F_c}\exists\, y.\,\forall\, x.\, P(y) \to P(x)}}{\mathbf{F_c}\forall\, x.\, P(b) \to P(x), \mathbf{T}P(a), \mathbf{F}P(b)}}{\mathbf{F}P(b) \to P(c), \mathbf{T}P(a), \mathbf{F}P(b)}}{\mathbf{T}P(b), \mathbf{F}P(c), \mathbf{T}P(a), \mathbf{F}P(b)}}{\text{closed by } P(b).}
$$

Translating this tableau into a natural-like proof is straightforward (we leave it to the reader), and Isabelle has no problem to certify its correctness.

5 Conclusions

Summarizing, in this paper we have shown that there is a correct algorithm which uniformly maps tableaus into natural-like deductions.

The first consequence of this result, is that, in order to trust the answers of an automatic theorem prover which uses our tableau calculus, we just need to prove that the implementation of the function Tr we developed, is correct. Of course, in this way we get a compromise: we break down the complexity of a formal correctness proof, but we are not able to guarantee the completeness of the prover, i.e., the ability to ensure that every theorem will get proven.

Following this line, we can refine our result: in fact, there is no real need to prove that the implementation of Tr is correct; admitting the possibility that a properly closed tableau could be discarded by a wrongly constructed tactic (that amounts to potentially enlarge the incompleteness of the prover), we reduce our trust to the correctness of the Isabelle core engine, which is responsible for the evaluation of the tactics Tr constructs.

In some sense, we cannot ask for more: in order to do some kind of theorem proving, we need an environment which provides the basic functionalities; our choice was Isabelle, but any other logical framework would have been the same. Then we have to develop strategies and/or provers for constructing proofs in this framework. The maximum we can achieve is to limit our *faith* to the core of the framework, which, eventually, would be proven formally correct.

The second consequence is more subtle; what Proposition 1 says, is that the translation rules respect the tableau structure. If we change the starting tableau calculus, it is possible to generate a set of translation rules which permits to prove Proposition 1 in a similar way as we did. In order to prove Theorem 1, it is essential to establish Proposition 2, but this is possible only if the tableau calculus is sound. So, even if the tableau calculus is not sound, we have a translation function which may leave some gaps into the natural-like deduction it generates. It may appear odd to use a non-sound calculus, but it is not. In fact, if we let the α parameter in the rules $\mathbf{T}\forall$, $\mathbf{F}\exists$ and $\mathbf{F_c}\exists$ to be a dummy variable which may get instantiated by unification during the process of constructing a tableau, we get a calculus which is not sound, since it is does not handle in the proper way the binding of variables (it is a problem very close to Prolog implementations which avoid the occur-check [13]). This variation of our tableau calculus is much more efficient, since it does not require so many duplications as the sound calculus, but, of course, it may produce closed tableaus which are not **IL**-T proof tables. Our approach is to validate a closed tableau by translating it into an **IL** proof. If this is possible, we have been able to prove in an efficient way a, possibly difficult, theorem; if this is not possible, it means that we applied in a non-sound way an expansion rule, so we are not able to judge whether the goal is a theorem.

The third consequence of our approach depends on the particular shape of the translation rule: since we took care of avoiding unnecessary detours, our natural-like proofs are inspectable. This is important when we want to analyze the strategies the prover adopts.

Concluding, we would like to remark that our technique can be extended to many other tableau calculi for a great varieties of logic. Moreover, we believe that "validating by translation" could be a winning approach in many situations, where it is important to have only correct answer, but completeness, i.e., the ability to generate every possible answer, is not a concern.

Acknowledgements

We would like to thank Professor Mario Ornaghi for the many useful hints and comments, and for the hard work of proof-reading this paper.

References

[1] P. B. Andrews. More on the problem of finding a mapping between clause representation and natural deduction representation. *Journal of Automated Reasoning*, 7:285–286, 1991.

[2] A. Avellone, M. Ferrari, and P. Miglioli. Duplication-free tableau calculi together with cut-free and contracti on-free sequent calculi for the interpolable propositional intermediate logics. *Journal of Logic and Computation*, 1997. To appear.

[3] A. Avellone, P. Miglioli, U. Moscato, and M. Ornaghi. Generalized tableau systems for intermediate propositional logics. In D. Galmiche, editor, *Proceedings of the 6th International Conference on automated reason ing with*

analytic tableaux and related methods: tableaux '97, volume 1227 of *LNAI*, pages 43–61. Springer–Verlag, 1997.

[4] W. Bibel, D. Korn, C. Kreitz, and S. Schmitt. Problem-oriented applications of automated theorem proving. In J. Calmet and C. Limongelli, editors, *International Symposium DISCO '96*, volume 1128 of *LNCS*, pages 1–21. Springer–Verlag, 1996.

[5] F. Buffoli. Costruzione di tableaux e deduzioni naturali normalizzate supportata da refutazioni clausali. In Stefania Costantini, editor, *Settimo convegno sulla programmazione logica*, pages 235–253, 1992. In Italian.

[6] R. Dyckhoff. Contraction–free sequent calculi for intuitionistic logic. *Journal of Symbolic Logic*, 57(3):795–807, 1992.

[7] M. Ferrari and P. Miglioli. Counting the maximal intermediate constructive logics. *Journal of Symbolic Logic*, 58(4):1365–1401, 1993.

[8] M. C. Fitting. *Intuitionistic Logic, Model Theory and Forcing.* North-Holland, 1969.

[9] J. Harrison. Binary Decision Diagrams as a HOL derived rule. *The Computer Journal*, 38:162–170, 1995.

[10] J. Harrison. Stalmarck's algorithm as a HOL derived rule. In *Proceedings of 9th International Conference on Theorem Proving in Higher Order Logics*, number 1125 in LNCS, pages 221–234. Springer Verlag, 1996.

[11] J. Hudelmaier. An $O(n \log n)$–space decision procedure for intuitionistic propositional logic. *Journal of Logic and Computation*, 3(1):63–75, 1993.

[12] C. Kreitz, J. Otten, and S. Schmitt. Guiding program development systems by connection based proof strategy. In M. Proietti, editor, *Proceedings of 5th International Workshop, LOPSTR 95*, volume 1207 of *LNCS*, pages 137–151. Springer–Verlag, 1995.

[13] J. W. Lloyd. *Foundations of Logic Programming.* Springer-Verlag, Berlin, 1994.

[14] P. Miglioli, U. Moscato, and M. Ornaghi. How to avoid duplications in a refutation system for intuitionistic logic and Kuroda logic. In K. Broda, M. D'Agostino, R. Goré, R. Johnson, and S. Reeves, editors, *Proceedings of 3rd Workshop on Theorem Proving with Analytic Tableaux and Related Methods.* Abingdon, U.K., May 4–6, 1994. Imperial College of Science, Technology and Medicine TR-94/5, 1994, pp. 169-187.

[15] P. Miglioli, U. Moscato, and M. Ornaghi. An improved refutation system for intuitionistic predicate logic. *Journal of Automated Reasoning*, 12:361–373, 1994.

[16] P. Miglioli, U. Moscato, and M. Ornaghi. Avoiding duplications in tableau systems for intuitionistic and Kuroda logics. *L.J. of the IGPL*, 5(1):145–167, 1997.

[17] W. M. J. Ophelders. *Automated Theorem Proving Based upon a Tableau-Method with Unification under Restrictions*. PhD thesis, Katholieke Universiteit Brabant, 1992.

[18] W. M. J. Ophelders and H. C. M. De Swart. Tableaux versus resolution a comparison. *Fundamenta Informaticae*, 20(2):109–127, 1993.

[19] L. C. Paulson. The foundation of a generic theorem prover. *Journal of Automated Reasoning*, 5(3):363–397, 1989.

[20] L. C. Paulson. *Isabelle: A Generic Theorem Prover*. Number 828 in LNCS. Springer-Verlag, 1994.

[21] L. C. Paulson. Generic automatic proof tools. In Robert Veroff, editor, *Automated Reasoning and its Applications*, chapter 3. MIT Press, 1997. In press. Available as Report 396, Computer Laboratory, University of Cambridge.

[22] F. Pfenning. *Proof transformations in higher-order logic*. PhD thesis, Carnegie Mellon University, 1987.

[23] D. Prawitz. *Natural Deduction. A Proof–Theoretical Study*. Almqvist–Wiksell, 1965.

[24] M. Redaelli. Sistemi di refutazione per logiche intermedie e modali. Master's thesis, Università degli Studi di Milano, Dipartimento di Scienze dell'Informazione, 1994.

[25] D. Sahlin, T. Franzén, and S. Haridi. An intuitionistic predicate logic theorem prover. *Journal of Logic and Computation*, 2(5):619–656, 1992.

[26] S. Schmitt and C. Kreitz. On transforming intuitionistic matrix proofs into standard-sequent proofs. In *Proceedings of the 4th Workshop on Theorem Proving with Analytic Tableaux and Related Methods*, volume 918 of *LNAI*, pages 106–121. Springer–Verlag, 1995.

[27] S. Schmitt and C. Kreitz. Converting non-classical matrix proofs into sequent-style systems. In *CADE-13*, volume 1104 of *LNAI*, pages 418–431. Springer–Verlag, 1996.

[28] S. Schmitt and C. Kreitz. Deleting redundancy in proof reconstruction. pages 262–276. International Conference TABLEAUX'98, Analytic Tableaux and Related Methods, 1998.

[29] C. A. Smorynski. Applications of Kripke models. In A.S. Troelstra, editor, *Metamathematical Investigation of Intuitionistic Arithmetic and Analysis*, volume 344 of *Lecture Notes in Mathematics*. Springer–Verlag, 1973.

[30] A. S. Troelstra. *Metamathematical Investigation of Intuitionistic Arithmetic and Analysis.* Lecture Notes in Mathematics 344. Springer–Verlag, 1973.

[31] A. S. Troelstra and H. Schwichtenberg. *Basic Proof Theory.* Cambridge University Press, 1996.

[32] L. Wos. *Automated Reasoning:33 Basic Research Problems.* Prentice Hall, 1988.

A Translation Rules

The algorithm (TR) which underlies the application of a translation rule is as follows:

1. If S is the context of the node to which a tableau expansion rule is applied, then $\Gamma = \{\phi \mid \mathbf{T}\phi \in S\} \cup \{\neg\phi \mid \mathbf{F_c}\phi \in S\}$ and $D = \bigvee\{\phi \mid \mathbf{F}\phi \in S\}$.
2. In general, a tableau expansion rule is mapped into the translation rule with the same name, unless
   (a) there is no corresponding translation rule (this happens for the $\mathbf{F}\vee$ and the $\mathbf{T}\neg$ rules); in this case, the proof with gaps is left unchanged;
   (b) there is a no$\vee$ or a $\bot$ variant (e.g. $\mathbf{F}\wedge$no$\vee$ and $\mathbf{F_c}\wedge\bot$) and the context of the node to which the tableau rule is applied contains no $\mathbf{F}$-wffs; in this case the variant rule is used.
3. When applying a translation rule, any antecedent which is not an instance of the G rule, is discharged, since it is an assumption of the gap we are filling (see Proposition 1).

We would like to remark that in the translation rules $\mathbf{T}\forall$, $\mathbf{F_c}\forall$, $\mathbf{F_c}\forall\bot$, $\mathbf{F_c}\exists$ and $\mathbf{F_c}\exists\bot$ the active formula is duplicated in the gap. In fact, according to our tableau calculus, the rules $\mathbf{T}\forall$, $\mathbf{F_c}\forall$ and $\mathbf{F_c}\exists$ require duplicating the active formula. In order to maintain the correspondence between terminal nodes and gaps, as shown in Proposition 1, we must duplicate the active formula in the translation rule. But, in the proof obtained when translating a closed tableau, these assumptions may disappear since, as noted in the remark after Definition 5, unneeded assumptions are deleted by the gap closure operation.

$\dfrac{A \wedge B \qquad \dfrac{\Gamma, [A], [B]}{D}\,\mathrm{G}}{D}\,\mathbf{T}\wedge$	
$\dfrac{\dfrac{\Gamma}{D \vee A}\,\mathrm{G} \qquad \dfrac{\Gamma}{D \vee B}\,\mathrm{G}}{D \vee (A \wedge B)}\,\mathbf{F}\wedge$	$\dfrac{\neg(A \wedge B) \qquad \dfrac{\Gamma, [\neg A]}{\bot}\,\mathrm{G} \qquad \dfrac{\Gamma, [\neg B]}{\bot}\,\mathrm{G}}{D}\,\mathbf{F_c}\wedge$
$\dfrac{\dfrac{\Gamma}{A}\,\mathrm{G} \qquad \dfrac{\Gamma}{B}\,\mathrm{G}}{A \wedge B}\,\mathbf{F}\wedge\mathrm{no}\vee$	$\dfrac{\neg(A \wedge B) \qquad \dfrac{\Gamma, [\neg A]}{\bot}\,\mathrm{G} \qquad \dfrac{\Gamma, [\neg B]}{\bot}\,\mathrm{G}}{\bot}\,\mathbf{F_c}\wedge\bot$

$\dfrac{A \vee B \qquad \dfrac{\Gamma, [A]}{D}\,\mathrm{G} \qquad \dfrac{\Gamma, [B]}{D}\,\mathrm{G}}{D}\,\mathbf{T}\vee$
$\dfrac{\neg(A \vee B) \qquad \dfrac{\Gamma, [\neg A], [\neg B]}{D}\,\mathrm{G}}{D}\,\mathbf{F_c}\vee$

$\dfrac{\dfrac{\Gamma, [A]}{\bot}\,\mathrm{G}}{D \vee \neg A}\,\mathbf{F}\neg$	$\dfrac{\dfrac{\Gamma, [A]}{\bot}\,\mathrm{G}}{\neg A}\,\mathbf{F}\neg\mathrm{no}\vee$
$\dfrac{\neg\neg A \qquad \dfrac{\Gamma, [A]}{\bot}\,\mathrm{G}}{D}\,\mathbf{F_c}\neg$	$\dfrac{\neg\neg A \qquad \dfrac{\Gamma, [A]}{\bot}\,\mathrm{G}}{\bot}\,\mathbf{F_c}\neg\bot$

$\dfrac{\exists\, x.\, A(x) \qquad \dfrac{\Gamma, [A(p)]}{D}\,\mathrm{G}}{D}\,\mathbf{T}\exists$ with p eigenvariable	
$\dfrac{\neg\exists\, x.\, A(x) \qquad \dfrac{\Gamma, [\neg A(a)], \neg\exists\, x.\, A(x)}{D}\,\mathrm{G}}{D}\,\mathbf{F_c}\exists$	$\dfrac{\dfrac{\Gamma}{D \vee A(a)}\,\mathrm{G}}{D \vee \exists\, x.\, A(x)}\,\mathbf{F}\exists$
$\dfrac{\neg\exists\, x.\, A(x) \qquad \dfrac{\Gamma, [\neg A(a)], \neg\exists\, x.\, A(x)}{\bot}\,\mathrm{G}}{\bot}\,\mathbf{F_c}\exists\bot$	$\dfrac{\dfrac{\Gamma}{A(a)}\,\mathrm{G}}{\exists\, x.\, A(x)}\,\mathbf{F}\exists\mathrm{no}\vee$

$\dfrac{\forall\, x.\, A(x) \qquad \dfrac{\Gamma, [A(a)], \forall\, x.\, A(x)}{D}\,\mathrm{G}}{D}\,\mathbf{T}\forall$	
$\dfrac{\dfrac{\Gamma}{A(p)}\,\mathrm{G}}{D \vee \forall\, x.\, A(x)}\,\mathbf{F}\forall$ with p eigenvariable	$\dfrac{\dfrac{\Gamma}{A(p)}\,\mathrm{G}}{\forall\, x.\, A(x)}\,\mathbf{F}\forall\mathrm{no}\vee$ with p eigenvariable
$\dfrac{\neg\forall\, x.\, A(x) \qquad \dfrac{\Gamma, \neg\forall\, x.\, A(x)}{A(p)}\,\mathrm{G}}{D}\,\mathbf{F_c}\forall$ with p eigenvariable	$\dfrac{\neg\forall\, x.\, A(x) \qquad \dfrac{\Gamma, \neg\forall\, x.\, A(x)}{A(p)}\,\mathrm{G}}{\bot}\,\mathbf{F_c}\forall\bot$ with p eigenvariable

$$\dfrac{\dfrac{\Gamma, [A]}{B}\,\mathrm{G}}{D \vee (A \to B)}\ \mathbf{F}{\to}$$

$$\dfrac{\dfrac{\Gamma, [A]}{B}\,\mathrm{G}}{A \to B}\ \mathbf{F}{\to}\mathrm{no}{\vee}$$

$$\dfrac{\neg(A \to B) \qquad \dfrac{\Gamma, [A], [\neg B]}{\bot}\,\mathrm{G}}{G}\ \mathbf{F_c}{\to}$$

$$\dfrac{\neg(A \to B) \qquad \dfrac{\Gamma, [A], [\neg B]}{\bot}\,\mathrm{G}}{\bot}\ \mathbf{F_c}{\to}\bot$$

$$\dfrac{A \to B \quad \dfrac{\Gamma}{D \vee A}\,\mathrm{G} \quad \dfrac{\Gamma, [B]}{D}\,\mathrm{G}}{D}\ \mathbf{T}{\to}a\neg$$

where A is atomic or negated

$$\dfrac{A \to B \quad \dfrac{\Gamma}{A}\,\mathrm{G} \quad \dfrac{\Gamma, [B]}{\bot}\,\mathrm{G}}{\bot}\ \mathbf{T}{\to}a\neg\bot$$

where A is atomic or negated

$$\dfrac{A \vee B \to C \qquad \dfrac{\Gamma, [A \to C], [B \to C]}{D}\,\mathrm{G}}{D}\ \mathbf{T}{\to}{\vee}$$

$$\dfrac{A \wedge B \to C \qquad \dfrac{\Gamma, [A \to (B \to C)]}{D}\,\mathrm{G}}{D}\ \mathbf{T}{\to}{\wedge}$$

$$\dfrac{(A \to B) \to C \quad \dfrac{\Gamma, [B \to C]}{D \vee (A \to B)}\,\mathrm{G} \quad \dfrac{\Gamma, [C]}{D}\,\mathrm{G}}{D}\ \mathbf{T}{\to}{\to}$$

$$\dfrac{(A \to B) \to C \quad \dfrac{\Gamma, [B \to C]}{A \to B}\,\mathrm{G} \quad \dfrac{\Gamma, [C]}{\bot}\,\mathrm{G}}{\bot}\ \mathbf{T}{\to}{\to}\bot$$

$$\dfrac{(\forall\, x.\, A(x)) \to B \quad \dfrac{\Gamma, (\forall\, x.\, A(x)) \to B}{D \vee (\forall\, x.\, A(x))}\,\mathrm{G} \quad \dfrac{\Gamma, [B]}{D}\,\mathrm{G}}{D}\ \mathbf{T}{\to}\forall$$

$$\dfrac{(\forall\, x.\, A(x)) \to B \quad \dfrac{\Gamma, (\forall\, x.\, A(x)) \to B}{\forall\, x.\, A(x)}\,\mathrm{G} \quad \dfrac{\Gamma, [B]}{\bot}\,\mathrm{G}}{\bot}\ \mathbf{T}{\to}\forall\bot$$

$$\dfrac{(\exists\, x.\, A(x)) \to B \qquad \dfrac{\Gamma, [\exists\, x.\, A(x) \to B]}{D}\,\mathrm{G}}{D}\ \mathbf{T}{\to}\exists$$

$$\dfrac{(\exists\, x.\, A(x)) \to B \qquad \dfrac{\Gamma, [\exists\, x.\, A(x) \to B]}{\bot}\,\mathrm{G}}{\bot}\ \mathbf{T}{\to}\exists\bot$$