

Tableau Calculus for Dummett Logic Based on Present and Next State of Knowledge

Guido Fiorino

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

Abstract

In this paper we use the Kripke semantics characterization of Dummett logic to introduce a new way of handling non-forced formulas in tableau proof systems. We pursue the aim of reducing the search space by strictly increasing the number of forced propositional variables after the application of non-invertible rules. The focus of the paper is on a new tableau system for Dummett logic, for which we have an implementation. The ideas presented can be extended to intuitionistic logic as well.

1 Introduction

In this paper we present a tableau calculus and theorem prover for propositional Dummett logic. By exploiting the linearly ordered Kripke semantics of Dummett logic, we devise a tableau calculus working on two semantical levels: the present and the next state of knowledge. In this way we can guarantee that as the construction of the tableau proceeds, moving from a state of knowledge to another the known information strictly increases. Moreover, the calculus can be equipped with specialized rules to reduce the branching. As a result, the decision procedure is speeded up. In Section 6 we discuss how to apply our ideas to a tableau calculus exploiting information about the next state of knowledge in the case of Intuitionistic deduction.

Dummett logic has been extensively investigated both by people working in computer science and in logic. The history of this logic starts with Gödel, who studied the family of logics semantically characterizable by a sequence of n -valued ($n > 2$) matrices ([12]). In paper [7] Dummett studied the logic semantically characterized by an infinite valued matrix which is included in the family of logics studied by Gödel and proved that such a logic is axiomatizable by adding to any Hilbert system for propositional intuitionistic logic the axiom scheme $(p \rightarrow q) \vee (q \rightarrow p)$. Moreover, it is well-known that such a logic is semantically characterizable by linearly ordered Kripke models. Dummett logic has been extensively studied also in recent years for its relationships with computer science ([4]) and fuzzy logics ([14]). For a survey in proof theory in Gödel-Dummett logics we quote [6].

To perform automated deduction both tableau and sequent calculi have been proposed. Paper [1] provides tableau calculi whose distinguishing feature is a multiple premise rule for implicative formulas that are not-forced. The syntactical way to express this semantical meaning is by the sign \mathbf{F} . We recall that the sign \mathbf{F} comes from Smullyan ([18, 11]) and labels those formulas that in a sequent calculus occur in the right-hand side of \Rightarrow . A tableau calculus derived from those of [1] is provided in paper [9]. Its main feature is that the depth of every deduction is linearly bounded in the length of the formula to be proved.

The approach of [1], based on characterizing Dummett logic by means of the multiple premise rule, has been criticized because, from the worst case analysis perspective, there are simple examples of sets of formulas giving rise to a factorial number of branches in the number of formulas in the set. Paper [5] shows how to get rid of the multiple premise rule. New rules are provided whose correctness is strictly related to the semantics of Dummett logic. These ideas have been further developed and in paper [16] a graph-theoretic decision procedure is described

and implemented. The approach introduced in [5] has also disadvantages with respect to the multiple premise rule proposed in [1] and these disadvantages have been considered in [10], where also a new version of the multiple premise rule is proposed. This version, from a practical point of view, can reduce the branching when compared with the original one. Paper [10] also provides an implementation that outperforms the one of [16], thus proving that the approach based on the multiple premise rule of [1] deserves attention also from the practical point of view. As a matter of fact, on the one hand the rules of [5] give rise to two branches at most, on the other hand there are cases of formulas that multiple premise calculi decide with a number of steps lower than the calculi based on [5].

In this paper we continue our investigation around multiple premise calculi for Dummett logic. The tool we use to prove our results is the characterization of Dummett logic via linearly ordered Kripke models, whose elements are considered *worlds* or *states of knowledge* ordered w.r.t. \leq and $\alpha \leq \beta$ means that, roughly speaking, β is in a subsequent point of the time w.r.t. α .

With the aim to reduce the size of the proofs, we present a calculus whose main feature is the way the non-forced formulas are handled. In tableau calculi proofs start by supposing that in the present state of knowledge A is not forced. A proof of our calculus starts by adding a further semantical constraint, namely, by supposing that the present state of knowledge is the last where A is not forced. This further constraint implies that in the present we have information about a different semantical status of A in the future: if we conclude that there exists the next state of knowledge, then we deduce that in such a future state of knowledge A is forced (note that this implies that starting from such a future state of knowledge A is equivalent to \top). Such information can be handled to draw deductions about the semantical status of the formulas both in the present and in the future state of knowledge. As a consequence, our calculus handles two kinds of signed formulas: the first kind describes the semantical status of a formula at the present state of knowledge, the second kind describes the semantical status of a formula to the next state of knowledge. The introduction of the signs related to the next state of knowledge allows to introduce specialized rules that exploit the knowledge about the future to draw deductions about the present. In particular, we refer to the introduction of specialized and single conclusion rules to handle the formulas of the kind $\mathbf{F}(A \rightarrow B)$ (to be read “at the present state of knowledge, the fact $A \rightarrow B$ is not known”), thus reducing the branching generated by the multiple premise rule.

The semantical constraint used in calculus has also the advantage to change the effect of the application of the non-invertible rules. Roughly speaking, non-invertible rules draw conclusions about future states of knowledge having as premise facts about the present. The application of these rules has an effect in the proof-search, usually requiring the introduction of backtracking to guarantee the completeness of the proof-search procedure. The semantical effect is that in the construction of the counter model a new state of knowledge is added, corresponding to the facts derived by the non-invertible rule. In the known sequent/tableau calculi for intermediate logics there is no guarantee that in the conclusion of the non-invertible rules at least one fact that in the premise was explicitly unknown becomes explicitly known in the conclusion. In the semantical construction this means that there is no guarantee that between two subsequent worlds, knowledge increases. The two semantical levels used in our calculus allow us to provide non-invertible rules such that there is at least one fact that in the premise is explicitly signed as not known and in the conclusion is explicitly signed as known. This implies that moving from the present to the next state of knowledge there is at least one explicitly unknown fact of the present that becomes explicitly known in the future. This has also a correspondence in the construction of the Kripke counter model, where there are no two elements exactly forcing

the same propositional variables. Such an increment of known information, that is formulas equivalent to \top , is exploited by the *replacement rule*, which replaces all the occurrences of a formula proved to be equivalent to the \top with \top . Such a replacement reduces the size of the set to be decided. Further reductions are possible by applying replacements based on the truth table of the connectives. Rules based on the replacement have been proved effective to dramatically reduce the search space both in classical and intuitionistic logic ([17, 2]), thus we have developed a calculus whose rules are designed to conclude as much as possible about formulas equivalent to \top . We remark that by using the appropriate data structures to represent formulas and sets of formulas, the application of the replacement rule and the reduction rules based on the truth table of the connectives can be performed in constant time. The results of our prolog prototype show that the calculus we provide is suitable for an implementation outperforming those cited above.

2 Basic definitions, the calculus and general considerations

We consider the propositional language based on a denumerable set of propositional variables \mathcal{PV} , the boolean constants \top and \perp and the logical connectives $\neg, \wedge, \vee, \rightarrow$. We call *atoms* the elements of $\mathcal{PV} \cup \{\top, \perp\}$. In the following, formulas (respectively set of formulas and propositional variables) are denoted by letters $A, B, C \dots$ (respectively S, T, U, \dots and p, q, r, \dots) possibly with subscripts or superscripts.

From the introduction we recall that *Dummett Logic (Dum)* can be axiomatized by adding to any axiom system for propositional intuitionistic logic the axiom scheme $(p \rightarrow q) \vee (q \rightarrow p)$ and a well-known semantical characterization of **Dum** is by *linearly ordered Kripke models*. In the paper *model* means a linearly ordered Kripke model, namely a structure $\underline{K} = \langle P, \leq, \rho, \Vdash \rangle$, where $\langle P, \leq, \rho \rangle$ is a linearly ordered set with ρ minimum with respect to \leq and \Vdash is the *forcing relation*, a binary relation on $P \times (\mathcal{PV} \cup \{\top, \perp\})$ such that: (i) if $\alpha \Vdash p$ and $\alpha \leq \beta$, then $\beta \Vdash p$; (ii) for every $\alpha \in P$, $\alpha \Vdash \top$ holds and $\alpha \Vdash \perp$ does not hold. Hereafter we denote the members of P by means of lowercase letters of the Greek alphabet.

The forcing relation is extended in a standard way to arbitrary formulas of our language 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 $\alpha \not\Vdash A$ when $\alpha \Vdash A$ does not hold. It is easy to prove that for every formula A the *persistence* property holds: If $\alpha \Vdash A$ and $\alpha \leq \beta$, then $\beta \Vdash A$. We say that β is *immediate successor* of α iff $\alpha < \beta$ and there is no $\gamma \in P$ such that $\alpha < \gamma < \beta$. A formula A is *valid in a model* $\underline{K} = \langle P, \leq, \rho, \Vdash \rangle$ if and only if $\rho \Vdash A$. It is well-known that **Dum** coincides with the set of formulas valid in all models.

The rules of our calculus \mathbb{D} are in Figures 1-4, where rules in Figures 1 and 2 form the logical apparatus necessary to decide **Dum**, whereas rules in Figures 3 and 4 are optimization rules, introduced to speed-up the deduction. The calculus \mathbb{D} works on signed formulas, that is well-formed formulas prefixed with one of the *signs* **T** (with **TA** to be read “the fact A is known

at the present state of knowledge”), \mathbf{F} (with $\mathbf{F}A$ to be read “the fact A is not known at the present state of knowledge”), \mathbf{F}_1 (with \mathbf{F}_1A to be read “this is the last state of knowledge where A is not known”), \mathbf{F}_n (with \mathbf{F}_nA to be read “ A is not known in the next state of knowledge”) and \mathbf{T}_n (with \mathbf{T}_nA to be read “ A will be known in the next state of knowledge”), and on sets of signed formulas (hereafter we omit the word “signed” in front of “formula” in all the contexts where no confusion arises). Formally, the meaning of the signs is provided by the relation *realizability* (\triangleright) defined as follows: Let $\underline{K} = \langle P, \leq, \rho, \Vdash \rangle$ be a model, let $\alpha \in P$, let H be a signed formula and let S be a set of signed formulas. We say that α *realizes* H (respectively α *realizes* S and \underline{K} *realizes* S), and we write $\alpha \triangleright H$ (respectively $\alpha \triangleright S$ and $\underline{K} \triangleright S$), if the following conditions hold:

1. $\alpha \triangleright \mathbf{T}A$ iff $\alpha \Vdash A$;
2. $\alpha \triangleright \mathbf{F}A$ iff $\alpha \not\Vdash A$;
3. $\alpha \triangleright \mathbf{F}_nA$ iff there exists $\beta > \alpha$, $\beta \triangleright \mathbf{F}A$;
4. $\alpha \triangleright \mathbf{T}_nA$ iff for every $\beta > \alpha$, $\beta \triangleright \mathbf{T}A$;
5. $\alpha \triangleright \mathbf{F}_1A$ iff $\alpha \triangleright \mathbf{F}A$ and $\alpha \triangleright \mathbf{T}_nA$;
6. $\alpha \triangleright S$ iff α realizes every formula in S ;

Before to enter into technical details, let us justify the introduction of the signs \mathbf{F}_1 , \mathbf{F}_n and \mathbf{T}_n in the object language by means of a motivating case. Let us suppose that a world α of a model \underline{K} realizes $S_{\mathbf{F} \rightarrow} = \{\mathbf{F}(A_1 \rightarrow B_1), \dots, \mathbf{F}(A_n \rightarrow B_n)\}$. Then there exists the last element $\beta \geq \alpha$ such that $\beta \triangleright \mathbf{T}A_j, \mathbf{F}B_j, S_{\mathbf{F} \rightarrow} \setminus \{\mathbf{F}(A_j \rightarrow B_j)\}$. Without loss of generality we let $j = 1$. Analogously, there exists an element $\gamma \geq \beta$ such that $\gamma \triangleright \mathbf{T}A_i, \mathbf{F}B_i, \mathbf{T}A_1, S_{\mathbf{F} \rightarrow} \setminus \{\mathbf{F}(A_1 \rightarrow B_1), \mathbf{F}(A_i \rightarrow B_i)\}$. Without loss of generality we let $i = 2$. Now, $\beta < \gamma$ or $\beta = \gamma$. If $\beta < \gamma$, then, since β is last element where $\mathbf{F}B_1$ holds, it follows that $\gamma \triangleright \mathbf{T}B_1$ holds. If $\beta = \gamma$, then β is the last element where both B_1 and B_2 are not forced. The example shows that we can give a rule ($\mathbf{F} \rightarrow$) taking into account that if $\alpha \triangleright \mathbf{F}(A_i \rightarrow B_i)$, for $i = 1, \dots, n$, then the set $\{\mathbf{T}A_i, \mathbf{F}B_i\}$ is realized in $\theta \in \{\alpha, \beta\}$ and for every world $\gamma > \theta$, $\gamma \triangleright \mathbf{T}B_i$ holds. The sign \mathbf{F}_1 aims to codify such a semantical property of B_i .

Generalizing the case given above, let $S_{\mathbf{F} \rightarrow} = \{\mathbf{F}(A_1 \rightarrow B_1), \dots, \mathbf{F}(A_n \rightarrow B_n)\}$ and let S be a set of formulas. Let us suppose that $\alpha \triangleright S, S_{\mathbf{F} \rightarrow}$. Then, for $i = 1, \dots, n$, let us consider the element β_i such that $\beta_i \triangleright \mathbf{T}A_i, \mathbf{F}B_i$ and for every $\gamma > \beta_i$, $\gamma \triangleright \mathbf{T}A_i, \mathbf{T}B_i$. Thus β_i is the maximum element such that $\beta_i \triangleright \mathbf{T}A_i, \mathbf{F}B_i$. Since $\alpha \triangleright \mathbf{F}(A_i \rightarrow B_i)$, such an element β_i exists. Let $\beta_j = \min\{\beta_1, \dots, \beta_n\}$, with $j \in \{1, \dots, n\}$. There are two cases: (i) $\beta_j = \alpha$, then we conclude that β realizes the set $S, \mathbf{T}A_j, \mathbf{F}_1B_j, S_{\mathbf{F} \rightarrow} \setminus \{\mathbf{F}(A_j \rightarrow B_j)\}$; (ii) $\beta_j > \alpha$, then β_j realizes the set $S_c, \mathbf{T}A_j, \mathbf{F}_1B_j, S_{\mathbf{F} \rightarrow} \setminus \{\mathbf{F}(A_j \rightarrow B_j)\}$, where, by the meaning of the signs, S_c consists of: (i) the formulas of S signed with \mathbf{T} ; (ii) the formulas $\mathbf{T}A$ such that $\mathbf{F}_1A \in S$.

With the aim of reducing the size of the proofs, we add further considerations justifying the introduction of the sign \mathbf{F}_n . Let us suppose that beside $\alpha \triangleright S, S_{\mathbf{F} \rightarrow}$ we also know that for every model $\beta_1, \dots, \beta_{j-1} > \beta_j$ holds. This rules out that $\beta_j \triangleright \mathbf{T}A_i, \mathbf{F}_1B_i$, for $i = 1, \dots, j-1$, and implies that there exists an element γ such that $\beta_j < \gamma$ and $\gamma \triangleright \mathbf{F}(A_i \rightarrow B_i)$. The sign \mathbf{F}_n aims to codify that the formulas $\mathbf{F}(A_i \rightarrow B_i)$ have the following semantical property: there exists an element γ such that $\beta_j < \gamma$ and $\gamma \triangleright \mathbf{F}(A_i \rightarrow B_i)$, thus the element β_j is not the maximum element realizing $\mathbf{F}(A_i \rightarrow B_i)$. By the meaning of \mathbf{F}_n it follows that $\beta_j \triangleright S_c, \mathbf{F}_n(A_1 \rightarrow B_1), \dots, \mathbf{F}_n(A_{j-1} \rightarrow B_{j-1}), \mathbf{T}A_j, \mathbf{F}B_j, S_{\mathbf{F} \rightarrow} \setminus \{\mathbf{F}(A_1 \rightarrow B_1), \dots, \mathbf{F}(A_{j-1} \rightarrow B_{j-1})\}$, where by the meaning of the signs, S_c consists of: (i) the formulas of S signed with \mathbf{T} ; (ii) the formulas

$\frac{S, \mathbf{T}(A \wedge B)}{S, \mathbf{TA}, \mathbf{TB}} \mathbf{T}\wedge$	$\frac{S, \mathbf{F}(A \wedge B)}{S, \mathbf{FA} S, \mathbf{FB}} \mathbf{F}\wedge$	$\frac{S, \mathbf{T}\neg(A \wedge B)}{S, \mathbf{T}\neg A S, \mathbf{T}\neg B} \mathbf{T}\neg\wedge$	$\frac{S, \mathbf{T}\neg\neg(A \wedge B)}{S, \mathbf{T}\neg\neg A, \mathbf{T}\neg\neg B} \mathbf{T}\neg\neg\wedge$
$\frac{S, \mathbf{T}(A \vee B)}{S, \mathbf{TA} S, \mathbf{TB}} \mathbf{T}\vee$	$\frac{S, \mathbf{F}(A \vee B)}{S, \mathbf{FA}, \mathbf{FB}} \mathbf{F}\vee$	$\frac{S, \mathbf{T}\neg(A \vee B)}{S, \mathbf{T}\neg A, \mathbf{T}\neg B} \mathbf{T}\neg\vee$	$\frac{S, \mathbf{T}\neg\neg(A \vee B)}{S, \mathbf{T}\neg\neg A S, \mathbf{T}\neg\neg B} \mathbf{T}\neg\neg\vee$
$\frac{S, \mathbf{F}\neg A}{S, \mathbf{T}\neg\neg A} \mathbf{F}\neg$	$\frac{S, \mathbf{T}\neg\neg\neg A}{S, \mathbf{T}\neg A} \mathbf{T}\neg\neg\neg$	$\frac{S, \mathbf{T}\neg(A \rightarrow B)}{S, \mathbf{T}\neg\neg A, \mathbf{T}\neg B} \mathbf{T}\neg\rightarrow$	$\frac{S, \mathbf{T}\neg\neg(A \rightarrow B)}{S, \mathbf{T}\neg A S, \mathbf{T}\neg\neg B} \mathbf{T}\neg\neg\rightarrow$
$\frac{S, \mathbf{T}((A \wedge B) \rightarrow C)}{S, \mathbf{T}(A \rightarrow (B \rightarrow C))} \mathbf{T}\rightarrow\wedge$	$\frac{S, \mathbf{T}(\neg A \rightarrow B)}{S, \mathbf{T}\neg\neg A S, \mathbf{TB}} \mathbf{T}\rightarrow\neg$	$\frac{S, \mathbf{T}((A \vee B) \rightarrow C)}{S, \mathbf{T}(A \rightarrow C), \mathbf{T}(B \rightarrow C)} \mathbf{T}\rightarrow\vee$	
$\frac{S, \mathbf{T}((A \rightarrow B) \rightarrow C)}{S, \mathbf{F}(A \rightarrow p), \mathbf{T}(p \rightarrow C), \mathbf{T}(B \rightarrow p) S, \mathbf{TC}} \mathbf{T}\rightarrow\rightarrow$ with p a new atom			
$\frac{S, \mathbf{F}_1(A \vee B)}{S, \mathbf{FA}, \mathbf{F}_1B, \mathbf{T}_nB S, \mathbf{FB}, \mathbf{F}_1A, \mathbf{T}_nA} \mathbf{F}_1\vee$		$\frac{S, \mathbf{F}(A \rightarrow B)}{S, \mathbf{TA}, \mathbf{F}_1B, \mathbf{T}_nB S, \mathbf{F}_n(A \rightarrow B)} \mathbf{F}\rightarrow$	
$\frac{S, \mathbf{T}_n(A \wedge B)}{S, \mathbf{T}_nA, \mathbf{T}_nB} \mathbf{T}_n\wedge$		$\frac{S, \mathbf{T}_n(\neg A)}{S, \mathbf{T}\neg A} \mathbf{T}_n\neg$	
$\frac{S, \mathbf{F}(A \rightarrow B)}{S, \mathbf{TA}, \mathbf{F}_1B} \mathbf{F}\rightarrow_1$, provided $\mathbf{T}_nB \in S$			
$\frac{S, \mathbf{F}_1(A \rightarrow B)}{S, \mathbf{TA}, \mathbf{F}_1B, \mathbf{T}_nB} \mathbf{F}_1\rightarrow$		$\frac{S, \mathbf{F}_1(A \wedge B)}{S, \mathbf{F}_1A, \mathbf{T}_nA, \mathbf{T}_nB S, \mathbf{TA}, \mathbf{T}_nB, \mathbf{F}_1B} \mathbf{F}_1\wedge$	
$\frac{S, \mathbf{T}(A \rightarrow B)}{S_{cl}, \mathbf{T}\neg A S_{cl}, \mathbf{TB}} \mathbf{T}\rightarrow\text{-cl}$		$\frac{S, \mathbf{T}\neg\neg A}{S_{cl}, \mathbf{TA}} \mathbf{T}\neg\neg\text{-cl}$	
provided S only contains \mathbf{T} and \mathbf{T}_n -formulas			
$\frac{S, \mathbf{F}_1(\neg A)}{S_{cl}, \mathbf{TA}} \mathbf{F}_1\neg$	$\frac{S, \mathbf{F}_1\perp}{S_{cl}} \mathbf{F}_1\perp$,	$\frac{S, \mathbf{T}_n\perp}{S_{cl}} \mathbf{T}_n\perp$,	
where $S_{cl} = \{\mathbf{TA} \mathbf{TA} \in S\} \cup \{\mathbf{T}\neg A \mathbf{FA} \in S\} \cup \{\mathbf{T}\neg A \mathbf{F}_1A \in S\}$			

Figure 1: The invertible rules of \mathbb{D} .

\mathbf{TA} such that $\mathbf{F}_1A \in S$. These are the intuitions justifying the rules \mathbf{F}_n and $\mathbf{F}\rightarrow$. In particular the aim of $\mathbf{F}\rightarrow$ is to distinguish if the last element realizing $\mathbf{F}(A \rightarrow B)$ is the same element α realizing the premise or an element β such that $\alpha < \beta$. The rule \mathbf{F}_n handles the formulas of the kind \mathbf{F}_nA . For every formula there exists a world β such that $\alpha < \beta$ and β is the last element such that $\beta \not\# A$. The minimum of such β does not force any formula in evidence in the premise.

The intuition behind the rule \mathbf{F}_n can be explained as follows. Let us suppose that $\alpha \triangleright S, \mathbf{F}_nA_1, \dots, \mathbf{F}_nA_u$. Thus there exists β_i such that $\alpha < \beta_i$ and $\beta_i \triangleright \mathbf{F}_1A_i$, for $i = 1, \dots, u$. We notice that β_i realizes all the \mathbf{T} formulas in S and $\beta_i \triangleright \mathbf{TC}$ if $\mathbf{F}_1C \in S$. Moreover, if $\beta_1 = \min\{\beta_1, \dots, \beta_u\}$, then $\beta_1 \triangleright \mathbf{F}_1A_1, \mathbf{FA}_2, \dots, \mathbf{FA}_u$. If β_2 is the minimum and we know that there is no model realizing $\alpha \triangleright S, \mathbf{F}_nA_1, \dots, \mathbf{F}_nA_u$ having β_1 as the minimum, then, since $\beta_2 < \beta_1$ holds, we conclude that $\beta_2 \triangleright \mathbf{F}_nA_1, \mathbf{F}_1A_2, \mathbf{FA}_3, \dots, \mathbf{FA}_u$. Analogously, if β_i is the minimum and we know that there is no model realizing $\alpha \triangleright S, \mathbf{F}_nA_1, \dots, \mathbf{F}_nA_u$ having β_1, \dots or β_{i-1} as the minimum, then, since $\beta_i < \beta_1, \dots, \beta_{i-1}$ holds, we conclude that $\beta_i \triangleright$

$$\boxed{
\begin{array}{c}
\frac{S, \mathbf{T}\neg\neg A}{S_{cl}, \mathbf{TA}|S_\phi, \mathbf{TA}} \mathbf{T}\neg\neg\text{-Atom, provided } S \text{ does not contain } \mathbf{F}_n\text{-formulas.} \\
\text{where} \\
S_\phi = \{\mathbf{TA}|\mathbf{TA} \in S\} \cup \{\mathbf{TA}|\mathbf{T}_n A \in S\} \cup \{\mathbf{TA}|\mathbf{F}_1 A \in S\} \text{ and} \\
S_{cl} = \{\mathbf{TA}|\mathbf{TA} \in S\} \cup \{\mathbf{T}\neg A|\mathbf{FA} \in S\} \cup \{\mathbf{T}\neg A|\mathbf{F}_1 A \in S\}; \\
\\
\frac{S, \mathbf{F}_n A_1, \dots, \mathbf{F}_n A_u}{V_1 | \dots | V_j | \dots | V_u} \mathbf{F}_n \\
\text{where:} \\
\text{for } j = 1, \dots, u, V_j = S_c \cup \{\mathbf{F}_n A_1, \dots, \mathbf{F}_n A_{j-1}, \mathbf{F}_1 A_j, \mathbf{F}_1 A_{j+1}, \dots, \mathbf{F}_1 A_u\}; \\
S_c = \{\mathbf{TA}|\mathbf{TA} \in S\} \cup \{\mathbf{TA}|\mathbf{F}_1 A \in S\} \cup \{\mathbf{TA}|\mathbf{T}_n A \in S\};
\end{array}
}$$

Figure 2: The non-invertible rules of \mathbb{D} .

$$\boxed{
\begin{array}{cc}
\frac{S, \mathbf{TA}}{S[A/\top], \mathbf{TA}} \text{Replace } \mathbf{T} & \frac{S, \mathbf{T}\neg A}{S[A/\perp], \mathbf{T}\neg A} \text{Replace } \mathbf{T}\neg \\
\\
\frac{S, \mathbf{T}\neg\neg A}{S[\neg A/\perp], \mathbf{T}\neg\neg A} \text{Replace } \mathbf{T}\neg\neg & \frac{S, \mathbf{F}_n A, \mathbf{F}_1 B}{S, \mathbf{F}_n A[B/\top], \mathbf{F}_1 B} \text{Replace } \mathbf{F}_1 \\
\\
\frac{S, SA, \mathbf{T}_n B}{S, SA[B/\top], \mathbf{T}_n B} \text{Replace } \mathbf{T}_n, \text{ with } S \in \{\mathbf{T}_n, \mathbf{F}_n\}
\end{array}
}$$

Figure 3: The Replacement rules

$$\boxed{
\begin{array}{cccc}
\frac{S}{S[A \wedge \perp/\perp]} \text{Simp } \wedge \perp & \frac{S}{S[\perp \wedge A/\perp]} \text{Simp } \perp \wedge & \frac{S}{S[A \wedge \top/A]} \text{Simp } \wedge \top & \frac{S}{S[\top \wedge A/A]} \text{Simp } \top \wedge \\
\frac{S}{S[A \vee \perp/A]} \text{Simp } \vee \perp & \frac{S}{S[\perp \vee A/A]} \text{Simp } \perp \vee & \frac{S}{S[A \vee \top/\top]} \text{Simp } \vee \top & \frac{S}{S[\top \vee A/\top]} \text{Simp } \top \vee \\
\frac{S}{S[\perp \rightarrow A/\top]} \text{Simp } \perp \rightarrow & \frac{S}{S[A \rightarrow \perp/\neg A]} \text{Simp } \rightarrow \perp & \frac{S}{S[\top \rightarrow A/A]} \text{Simp } \top \rightarrow & \frac{S}{S[A \rightarrow \top/\top]} \text{Simp } \rightarrow \top \\
\frac{S}{S[\neg \top/\perp]} \text{Simp } \neg \top & \frac{S}{S[\neg \perp/\top]} \text{Simp } \neg \perp & &
\end{array}
}$$

Figure 4: The Simplification rules

$\mathbf{F}_n A_1, \dots, \mathbf{F}_n A_{i-1}, \mathbf{F}_1 A_i, \mathbf{F}_1 A_{i+1}, \dots, \mathbf{F}_1 A_u$.

The sign \mathbf{T}_n is strictly related to \mathbf{F}_1 and conveys redundant information with respect to the completeness. For this reason we give only two rules to treat the formulas of this kind and the replacement rules. This presentation of the calculus takes into account the implementation of the prolog prototype. We have decided to embed in the logical part the information conveyed from the \mathbf{F}_1 -formulas. The explicit introduction of \mathbf{T}_n emphasizes how to handle the information of the \mathbf{F}_1 -formulas and its computational cost. An alternative presentation could avoid to insert

the sign \mathbf{T}_n . In this case the rule $\mathbf{F} \rightarrow_1$ becomes

$$\frac{S, \mathbf{F}(A \rightarrow B), \mathbf{F}_1 B}{S, \mathbf{T}_n A, \mathbf{F}_1 B}$$

Also the replacement rules for \mathbf{T}_n can be restated as rules involving \mathbf{F}_1 . The rules $\mathbf{T}_n \wedge$ and $\mathbf{T}_n \neg$ deserve some considerations. From the point of view of the presentation of the logical calculus they are not relevant and can be deleted from Figure 1. From the practical point of view, the application of these rules is useful because they break down the information deriving from \mathbf{F}_1 formulas and can be used, as an example, to turn a $\mathbf{F} \rightarrow$ -formula into a $\mathbf{F}_n \rightarrow$ -formula. Thus one wonders how to insert them in the decision procedure. Our implementation treats explicitly the information derived from \mathbf{F}_1 -formulas by means of the sign \mathbf{T}_n and of the logical rules. A different choice could be to use the \mathbf{F}_1 -formulas and implicitly break down the semantical information of the \mathbf{F}_1 -formulas to decide, as an example, if a given $\mathbf{F} \rightarrow$ -formula can be turned into a $\mathbf{F}_1 \rightarrow$ -formula, or if $\mathbf{F}_n A$ is equivalent to (the inconsistent) formula $\mathbf{F}_n \top$. Our choice is to insert all the machinery in the logical apparatus of the calculus. Finally, the sign \mathbf{T}_n permits to introduce many other special rules allowing a reduction of the branching. We postpone to Section 6 the discussion of these special rules.

From the meaning of the signs we get the conditions that make a set of formulas inconsistent. A set S is *inconsistent* if one of the following conditions holds:

$$\begin{array}{llll} -\mathbf{T} \perp \in S; & -\mathbf{F} \top \in S; & -\mathbf{F}_1 \top \in S; & -\mathbf{F}_n \top \in S; \\ -\{\mathbf{F}_n A, \mathbf{F}_1 \neg B\} \subseteq S; & -\{\mathbf{F}_n A, \mathbf{T}_n \perp\} \subseteq S; & -\{\mathbf{F}_n A, \mathbf{F}_1 \perp\} \subseteq S; & \end{array}$$

We emphasize that inconsistency conditions of the last line are related to the existence of a future state of knowledge α and in such a α the other formula of the pair is not realizable. It is easy to prove the following

Proposition 1. *If a set of formulas S is inconsistent, then for every Kripke model $\underline{K} = \langle P, \leq, \rho, \Vdash \rangle$ and for every $\alpha \in P$, $\alpha \not\Vdash S$.*

Proof. We only consider some significant the case $\{\mathbf{F}_n A, \mathbf{F}_1 \neg B\} \subseteq S$. By absurd, let us suppose that $\alpha \triangleright S$, then $\alpha \triangleright \mathbf{F}_1 \neg B$ and $\alpha \triangleright \mathbf{F}_n A$. Since $\alpha \triangleright \mathbf{F}_n A$, there exists $\beta \in P$ such that $\alpha < \beta$. By definition of \mathbf{F}_1 , it follows that $\alpha \not\Vdash \neg B$ and for every $\beta \in P$, if $\alpha < \beta$, then $\beta \Vdash \neg B$. Note that such a β exists. Since \underline{K} is a linearly ordered Kripke model, by definition of negation it follows that $\alpha \Vdash \neg B$. Thus we have that $\alpha \not\Vdash \neg B$ and $\alpha \Vdash \neg B$, absurd. The other cases are easy to prove. □

A proof table (or proof tree) for S is a tree, rooted in S and obtained by the subsequent instantiation of the rules of the calculus. The premise of the rules are instantiated in a duplication-free style: in the application of the rules we always consider that the formulas in evidence in the premise are not in S . We say that a rule \mathcal{R} applies to a set U when it is possible to instantiate the premise of \mathcal{R} with the set U and we say that a rule \mathcal{R} applies to a formula $H \in U$ (respectively the set $\{H_1, \dots, H_n\} \subseteq U$) to mean that it is possible to instantiate the premise of \mathcal{R} taking S as $U \setminus \{H\}$ (respectively $U \setminus \{H_1, \dots, H_n\}$).

A *closed proof table* is a proof table whose leaves are all inconsistent sets. A closed proof table is a proof of the calculus and a formula A is provable iff there exists a closed proof table for $\{\mathbf{F}_1 A\}$. We refer to [13] for full details on tableaux systems.

The calculus contains two non-invertible rules. In Section 4 we prove that it is possible to devise a complete strategy that relies on respecting a particular sequence in the application of

the rules: $\mathbf{T}\neg\neg$ -Atom is applied if no other rule is applicable and \mathbf{F}_n is applied if no other rule but $\mathbf{T}\neg\neg$ -Atom is applicable.

3 Correctness

To prove the correctness of \mathbb{D} with respect to Dummett logic we need to prove that, if there exists a closed proof table for $\{\mathbf{F}_1A\}$, then A is a valid formula in Dummett logic. The main step is to prove that the rules of the calculus preserve realizability:

Proposition 2. *For every rule of \mathbb{D} , if a model realizes the premise, then there exists a model realizing at least one of the conclusions.*

Proof. Let α be an element of $\underline{K} = \langle P, \leq, \rho, \Vdash \rangle$. We analyze the correctness of some rules of \mathbb{D} . Rule $\mathbf{F}_1 \rightarrow$: if $\alpha \triangleright S, \mathbf{F}_1(A \rightarrow B)$, then, by definition of \mathbf{F}_1 , $\alpha \not\Vdash A \rightarrow B$ and for every $\beta \in P$, if $\alpha < \beta$, then $\beta \Vdash A \rightarrow B$. This implies, by definition of forcing of an implicative formula, that $\alpha \Vdash A$ and $\alpha \not\Vdash B$ hold. By the preservation of forcing we get that, for every $\beta \in P$, if $\alpha < \beta$, then $\beta \Vdash A$, thus $\beta \Vdash B$. Therefore $\alpha \triangleright \mathbf{F}_1B$.

Rule $\mathbf{F}_1 \wedge$: if $\alpha \triangleright S, \mathbf{F}_1(A \wedge B)$ then, by definition of \mathbf{F}_1 , $\alpha \not\Vdash A \wedge B$, thus $\alpha \not\Vdash A$ or $\alpha \not\Vdash B$. Moreover, for every $\beta \in P$, if $\alpha < \beta$, then $\beta \Vdash A \wedge B$, thus $\beta \Vdash A$ and $\beta \Vdash B$. We have two possible cases: (i) $\alpha \not\Vdash A$, thus $\alpha \triangleright \mathbf{F}_1A$, $\alpha \triangleright \mathbf{T}_nA$ and $\alpha \triangleright \mathbf{T}_nB$; (ii) $\alpha \not\Vdash B$, thus $\alpha \triangleright \mathbf{T}_nA$, $\alpha \triangleright \mathbf{F}_1B$ and $\alpha \triangleright \mathbf{T}_nB$.

Rule $\mathbf{F}_1 \neg$: if $\alpha \triangleright S, \mathbf{F}_1(\neg A)$, then we notice that no \mathbf{F}_n -formula is in S . By definition of \mathbf{F}_1 , $\alpha \not\Vdash \neg A$, thus there exists $\gamma \in P$ such that $\alpha \leq \gamma$ and $\gamma \Vdash A$. Moreover, for every $\beta \in P$, if $\alpha < \beta$, then $\beta \Vdash \neg A$. By definition of forcing, an element cannot force a formula and its negation, thus the only possibility is that α is the final element of \underline{K} , thus $\gamma = \alpha$. Since α is the final element of \underline{K} , for every formula B , $\alpha \not\Vdash B$ implies $\alpha \Vdash \neg B$ and $\alpha \Vdash \neg\neg B$ implies $\alpha \Vdash B$. Thus $\alpha \triangleright S_{cl}$.

Rule $\mathbf{T}\neg\neg$ -Atom: if $\alpha \triangleright S, \mathbf{T}\neg\neg A$, then we have two cases: (i) α is not the final element ϕ . By the semantical meaning of the signs, it follows that ϕ realizes all the \mathbf{T} -formulas in S and if $\mathbf{F}_1B \in S$, then $\phi \triangleright \mathbf{T}B$; (ii) if α is the final element, then it is immediate to check that α realizes the leftmost conclusion of $\mathbf{T}\neg\neg$ -Atom.

Rule $\mathbf{T}_n \perp$: if $\alpha \triangleright S, \mathbf{T}_n \perp$, then for every $\beta \in P$, if $\alpha < \beta$, then $\beta \Vdash \perp$. Since no world forces \perp the only possibility is that $\alpha < \beta$ does not hold, thus α is the final element of \underline{K} . Since α behaves as a classical model with respect to the semantics of the connectives, we get that $\alpha \triangleright S_{cl}$. □

Theorem 1. *If there exists a proof table for A , then A is valid in **Dum**.*

Proof. By hypothesis there exists a proof table starting from $\{\mathbf{F}_1A\}$ whose leaves are all inconsistent sets. An inconsistent set is not realizable and the rules of \mathbb{D} preserve the realizability, thus \mathbf{F}_1A is not realizable. By absurd, let us suppose that there exist a model $\underline{K} = \langle P, \leq, \rho, \Vdash \rangle$ and a world $\alpha \in P$ such that $\alpha \not\Vdash A$. Then, there exists $\beta \in P$ s.t. $\alpha \leq \beta$, $\beta \not\Vdash A$ and for every $\gamma \in P$ s.t. $\beta < \gamma$, $\gamma \Vdash A$ holds. Thus $\beta \triangleright \mathbf{F}_1A$ and, since the rules of \mathbb{D} preserve the realizability, the leaves of the proof table are realized, which is impossible. We conclude that for every model $\underline{K} = \langle P, \leq, \rho, \Vdash \rangle$ and every $\alpha \in P$, $\alpha \Vdash A$, that is A is valid in **Dum**. □

The aim of the sign \mathbf{F}_1 is to label those formulas that after an application of \mathbf{F}_n become equivalent to \top . After an application of the rule \mathbf{F}_n , the formulas signed with \mathbf{F}_1 become signed with \mathbf{T} , thus they are equivalent to \top . Hence the rule Replace \mathbf{T} of Fig. 3 are applicable. The occurrences of \top are removed by using the simplification rules of Fig. 4. We refer to the whole machinery of Fig. 3 and 4 as *reduction rules*. It is an easy task to check that the rules are invertible. We emphasize that Replace $\mathbf{T}\neg$ and Replace $\mathbf{T}\neg\neg$ are a specialization of Replace \mathbf{T} . The rule Replace \mathbf{F}_1 exploits the meaning of \mathbf{F}_1 and \mathbf{F}_n to perform a replacement: if $\alpha \triangleright \mathbf{F}_n A, \mathbf{F}_1 B$ holds, then we conclude $\beta \triangleright \mathbf{F}A, \mathbf{T}B$, with β immediate successor of α . Thus $\beta \triangleright \mathbf{F}A[B/\top]$ and $\alpha \triangleright \mathbf{F}A[B/\top]$ hold.

It can be noticed that in the conclusion of the \mathbf{F}_1 -rules there is a duplication of a subformula of the premise. This duplication, that occurs with sign \mathbf{T}_n , is correct but not necessary to the completeness and is related to the logical treatment of the information under sign \mathbf{F}_1 . It has to be noticed that this duplication does not explode in an exponential number of formulas since the presence of the rule Replace \mathbf{T}_n allows the \mathbf{T}_n -formulas generated by duplication from \mathbf{F}_1 -formulas to be treated only once. As an example, consider $\mathbf{F}_1 \rightarrow$ and let us suppose that B occurs as subformula of $\mathbf{T}_n C$ in S . By applying Replace \mathbf{T}_n the occurrence of B in C is replaced by \top , thus the rules of the calculus handle the connectives of B twice: once when $\mathbf{F}_1 B$ is handled and once when $\mathbf{T}_n B$ is handled. We also remark that there are only two rules for \mathbf{T}_n -formulas. This implies that the overhead necessary to manage by \mathbf{T}_n -formulas the information conveyed by \mathbf{F}_1 -formulas does not implies the generation of branches and requires at most a number of applications of rules of \mathbb{D} which is linear in the length of the \mathbf{F}_1 -formulas.

Compared with the calculus in paper [10], introducing the sign \mathbf{F}_1 has a price: proof tables can be wider, because of the combined action of $\mathbf{F} \rightarrow$ and \mathbf{F}_n . To reduce this problem there is the rule $\mathbf{F} \rightarrow_1$, which is not necessary to the completeness and represents an example of exploitation of \mathbf{T}_n -formulas. Moreover, since the application of rules in Fig.2 turns the \mathbf{F}_1 -formulas into \mathbf{T} -formulas, that is formulas equivalent to the logical value \top , we investigate if this feature can reduce the depth of the proofs. We have experienced that the reduction rules reduce the size of the proofs, for this reason we are devising a calculus that introduces as much as possible formulas equivalent to the logical value \top .

4 A Strategy to Decide Dummett Logic and Its Completeness

In the following we sketch the recursive procedure $\text{DUM}(S)$. Given a set S of formulas, $\text{DUM}(S)$ returns either a closed proof table for S or NULL (if there exists a model realizing S). To describe DUM we use the following definitions and notations. We call α -rules and β -rules the rules of Figure 1 with one conclusion and with two conclusions, respectively. The α -formulas and β -formulas are the kind of the signed formulas in evidence in the premise of the α -rules and β -rules, respectively (e.g. $\mathbf{T}(A \wedge B)$ is an α -formula and $\mathbf{T}(A \vee B)$ is a β -formula). Let S be a set of formulas, let $H \in S$ be an α or β -formula. With $\text{Rule}(H)$ we denote the rule corresponding to H in Figure 1. Let S_1 or $S_1 \mid S_2$ be the nodes of the proof tree obtained by applying to S the rule $\text{Rule}(H)$. If Tab_1 and Tab_2 are closed proof tables for S_1 and S_2 respectively, then $\frac{S}{\text{Tab}_1} \text{Rule}(H)$ or $\frac{S}{\text{Tab}_1 \mid \text{Tab}_2} \text{Rule}(H)$ denote the closed proof table for S defined in the obvious way. Moreover, for H different from $\mathbf{F}_1 \perp$ and $\mathbf{T}_n \perp$, $\mathcal{R}_i(H)$ ($i = 1, 2$) denotes the set containing the formulas of S_i which replaces H . For instance:

$$\begin{aligned} \mathcal{R}_1(\mathbf{T}(A \wedge B)) &= \{\mathbf{T}A, \mathbf{T}B\}, \\ \mathcal{R}_1(\mathbf{T}(A \vee B)) &= \{\mathbf{T}A\}, \mathcal{R}_2(\mathbf{T}(A \vee B)) = \{\mathbf{T}B\}, \end{aligned}$$

In the case of \mathbf{F}_n we generalize the above notation. Let $S_{\mathbf{F}_n}$ be the set of all the \mathbf{F}_n -formulas of S . Let $S_1 | \dots | S_n$ be the nodes of the proof tree obtained by applying to S the rule \mathbf{F}_n . If $Tab_1 \dots, Tab_n$ are closed proof tables for S_1, \dots, S_n , respectively, then $\frac{S}{Tab_1 | \dots | Tab_n} \mathbf{F}_n$ is the closed proof table for S . With $\mathcal{R}_i(S_{\mathbf{F}_n})$ we denote the set of formulas that replaces the set $S_{\mathbf{F}_n}$ in the i -th conclusion of \mathbf{F}_n . For example, given $S_{\mathbf{F}_n} = \{\mathbf{F}_n A_1, \mathbf{F}_n A_2, \mathbf{F}_n A_3\}$, $\mathcal{R}_2(S_{\mathbf{F}_n \rightarrow}) = \{\mathbf{F}_n A_1, \mathbf{F}_n A_2, \mathbf{F}_n A_3\}$.

We consider SIMPLIFICATION as function that applies to its parameters the reduction rules as long as possible.

FUNCTION DUM (S)

1. If S is an inconsistent set, then DUM returns the proof S ;
2. Let $S' = \text{SIMPLIFICATION}(S)$. If $S' \neq S$, then let $\pi = \text{DUM}(S')$. If π is a proof, then DUM returns $\frac{S}{\pi} \text{SIMP}$, otherwise DUM returns NULL;
3. If the rule $\mathbf{F}_1 \perp$ or $\mathbf{T}_n \perp$ applies to S , then let $H \in S$ be the formula $\mathbf{T}_n \perp$ or $\mathbf{F}_1 \perp$. If $\text{DUM}((S \setminus \{H\})_{cl})$ returns a proof π , then DUM returns the proof $\frac{S}{\pi} \text{Rule}(H)$, otherwise DUM returns NULL;
4. If the rule $\mathbf{T} \neg \neg \text{-cl}$ or $\mathbf{F}_1 \neg$ applies to S , then let $H \in S$ be a formula of the kind $\mathbf{T} \neg \neg A$ or $\mathbf{F}_1 \neg$. If $\text{DUM}((S \setminus \{H\})_{cl} \cup \{\mathbf{T} A\})$ returns a proof π , then DUM returns the proof $\frac{S}{\pi} \text{Rule}(H)$, otherwise DUM returns NULL;
5. If an α -rule applies to S , then let H be a α -formula of S . If $\text{DUM}((S \setminus \{H\}) \cup \mathcal{R}_1(H))$ returns a proof π , then DUM returns the proof $\frac{S}{\pi} \text{Rule}(H)$, otherwise DUM returns NULL;
6. If the rule $\mathbf{T} \rightarrow \text{-cl}$ applies to S , then let $H \in S$ be a formula of the kind $\mathbf{T}(A \rightarrow B)$. Let $\pi_1 = \text{DUM}((S \setminus \{H\})_{cl} \cup \{\mathbf{T} \neg A\})$ and $\pi_2 = \text{DUM}((S \setminus \{H\})_{cl} \cup \{\mathbf{T} B\})$. If π_1 or π_2 is NULL, then DUM returns NULL, otherwise DUM returns $\frac{S}{\pi_1 | \pi_2} \mathbf{T} \rightarrow \text{-cl}$;
7. If a β -rule applies to S , then let H be a β -formula of S . Let $\pi_1 = \text{DUM}((S \setminus \{H\}) \cup \mathcal{R}_1(H))$ and $\pi_2 = \text{DUM}((S \setminus \{H\}) \cup \mathcal{R}_2(H))$. If π_1 or π_2 is NULL, then DUM returns NULL, otherwise DUM returns $\frac{S}{\pi_1 | \pi_2} \text{Rule}(H)$;
8. If the rule \mathbf{F}_n applies to S , then let $S_{\mathbf{F}_n} = \{\mathbf{F}_n A \in S\}$ and $n = |S_{\mathbf{F}_n}|$. If there exists $i \in \{1, \dots, n\}$, such that $\pi_i = \text{DUM}((S \setminus S_{\mathbf{F}_n})_c \cup \mathcal{R}_i(S_{\mathbf{F}_n}))$ is NULL, then DUM returns NULL. Otherwise π_1, \dots, π_n are proofs and DUM returns $\frac{S}{\pi_1 | \dots | \pi_n} \mathbf{F}_n$;
9. If the rule $\mathbf{T} \neg \neg \text{-Atom}$ applies to S , then let H be a $\mathbf{T} \neg \neg \text{-Atom}$ formula of S . Let $\pi_1 = \text{DUM}((S \setminus \{H\})_{cl} \cup \mathcal{R}_1(H))$ and $\pi_2 = \text{DUM}((S \setminus \{H\})_{\phi} \cup \mathcal{R}_2(H))$. If π_1 or π_2 is NULL, then DUM returns NULL, otherwise DUM returns $\frac{S}{\pi_1 | \pi_2} \text{Rule}(H)$;
10. If none of the previous points apply, then DUM returns NULL.

END FUNCTION DUM.

We emphasize that function DUM respects a particular sequence in the application of the rules: $\mathbf{T} \neg \neg \text{-Atom}$ is applied if no other rule is applicable and \mathbf{F}_n is applied if no other rule but $\mathbf{T} \neg \neg \text{-Atom}$ is applicable. As a result no backtracking step is necessary.

By inspecting the rules of the calculus it follows that the procedure terminates. In particular the rightmost conclusion of the rule $\mathbf{F} \rightarrow$ only change the sign of the formula in evidence in the premise, but we remark that the rule \mathbf{F}_n handles these kind of formulas, thus the combined action of $\mathbf{F} \rightarrow$ and \mathbf{F}_n allows to prove that there is no an infinite loop handling $\mathbf{F} \rightarrow$ -formulas.

In order to get the completeness of DUM, in the following it is proved that given a set of

formulas S , if the call of $\text{DUM}(S)$ returns NULL , then there is enough information to build a model $\underline{K} = \langle P, \leq, \rho, \Vdash \rangle$ such that $\rho \triangleright S$.

Lemma 1 (Completeness). *Let S be a set of formulas and suppose that $\text{DUM}(S)$ returns NULL . Then, there exists a Kripke model $\underline{K} = \langle P, \leq, \rho, \Vdash \rangle$ such that $\rho \triangleright S$.*

Proof. By induction on the number of nested recursive calls.

Basis: There are no recursive calls. Then Step 10 has been performed. We notice that S is not inconsistent (otherwise Step 1 would have been performed) and S does not contain neither \mathbf{F}_n -formulas nor $\mathbf{T}\neg\neg$ -formulas (otherwise Step 8 or Step 9 would have been performed). Indeed, S only contains atomic formulas signed with $\mathbf{T}, \mathbf{F}, \mathbf{T}_n, \mathbf{F}_1$, formulas of the kind $\mathbf{T}\neg p$, with p atomic, and formulas of the kind $\mathbf{T}(p \rightarrow A)$ with p atomic and $\mathbf{T}p \notin S$ (otherwise, if $\mathbf{T}p \in S$ holds, then Step 2 applies and at least a recursive call to DUM would have been performed). It is easy to prove that the model $\underline{K} = \langle P, \leq, \rho, \Vdash \rangle$, where $P = \{\rho\}$, $\rho \leq \rho$ and $\rho \Vdash p$ iff $\mathbf{T}p \in S$, realizes S .

Step: By induction hypothesis we assume that the proposition holds for all sets S' such that $\text{DUM}(S')$ requires less than n recursive calls. We prove the proposition holds for a set S such that $\text{DUM}(S)$ requires n recursive calls by inspecting all the possible cases where the procedure returns the NULL value.

NULL value returned performing Step 4 with H of the kind $\mathbf{F}_1(\neg A)$. By induction hypothesis there exists a model \underline{K} realizing $S_{cl}, \mathbf{T}A$. Notice that S does not contain \mathbf{F}_n -formulas, otherwise S would be inconsistent, and S_{cl} contains only \mathbf{T} -formulas. This implies that all the formulas occurring in the subsequent recursive calls are signed with \mathbf{T} . Thus no step related to the rules of Fig. 2 is employed. We conclude that ρ is the only element of \underline{K} and $\rho \triangleright S_{cl}$ implies $\rho \triangleright S$. Moreover, ρ maximal element of \underline{K} and $\rho \triangleright \mathbf{T}A$ imply $\rho \triangleright \mathbf{F}_1(\neg A)$.

NULL value returned performing Step 8. We notice that S is consistent, thus no formula of the kind $\mathbf{F}_1(\neg A)$ belongs to S . The set S contains atomic formulas and formulas of the kind $\mathbf{T}(p \rightarrow A)$, $\mathbf{F}_n(A \rightarrow B)$, $\mathbf{F}_n(A \wedge B)$, $\mathbf{F}_n(A \vee B)$. Since the NULL instruction in Step 8 has been performed, at least a π_i is NULL , then by induction hypothesis there is a model $\underline{K}' = \langle P, \leq', \rho', \Vdash' \rangle$ realizing $(S \setminus S_{\mathbf{F}_n})_c \cup \mathcal{R}_i(S_{\mathbf{F}_n})$. We define a Kripke model $\underline{K} = \langle P \cup \{\rho\}, \leq, \rho, \Vdash \rangle$, where

$$\begin{aligned} P \cap \{\rho\} &= \emptyset, \\ \leq &= \leq' \cup \{(\rho, \alpha) \mid \alpha \in P\} \\ \Vdash &= \Vdash' \cup \{(\rho, p) \mid \mathbf{T}p \in S\} \end{aligned}$$

Since \underline{K}' is a Dummett model realizing $(S \setminus S_{\mathbf{F}_n})_c$, it follows that \underline{K} is a Dummett model. As a matter of fact, ρ' is the immediate successor of ρ and $\mathbf{T}A \in S$ implies $\mathbf{T}A \in (S \setminus S_{\mathbf{F}_n})_c$, thus the forcing relation is preserved. Moreover, $\mathbf{F}_1A \in S$ implies $\mathbf{T}A \in (S \setminus S_{\mathbf{F}_n})_c$ and the consistency of S implies $\mathbf{T}A \notin S$. Thus we have proved that $\underline{K}' \triangleright (S \setminus S_{\mathbf{F}_n})_c \cup \mathcal{R}_i(S_{\mathbf{F}_n})$. It is an easy task to check that \underline{K} realizes S .

NULL value returned performing Step 9. We notice that when Step 9 is reached S does not contain any \mathbf{F}_n -formula. Thus if a $\mathbf{T}\neg\neg$ -formula belongs to S the recursive call is performed and it is a correct application of the rule $\mathbf{T}\neg\neg\text{-Atom}$. Since Step 9 returns NULL one between π_1 or π_2 is NULL . If π_1 is NULL , then by induction hypothesis there is a model $\underline{K} = \langle P, \leq, \rho, \Vdash \rangle$ such that $\rho \triangleright (S \setminus \{\mathbf{T}\neg\neg A\})_{cl} \cup \mathbf{T}A$. By the meaning of $(S \setminus \{\mathbf{T}\neg\neg A\})_{cl}$ immediately we

get that $\rho \triangleright S$. If π_2 is NULL, then, by induction hypothesis, the recursive call returns a model $\underline{K}' = \langle P, \leq', \rho', \Vdash' \rangle$ realizing $(S \setminus \{\mathbf{T}\neg\neg A\})_\phi \cup \{\mathbf{T}A\}$. We define a model $\underline{K} = \langle P \cup \{\rho\}, \leq, \rho, \Vdash \rangle$ as in the above point. Now, $\rho' \triangleright \mathbf{T}A$ implies $\rho \triangleright \mathbf{T}\neg\neg A$. Moreover, by definition of $(S \setminus \{\mathbf{T}\neg\neg A\})_\phi$, it is immediate to prove that $\rho \triangleright (S \setminus \{\mathbf{T}\neg\neg A\})$ holds. \square

We conclude the section with a sketch of deduction with our calculus. We consider the provable formula SYJ201.001 of ILTP library:

$$H \equiv (((p0 \rightarrow p1) \wedge (p1 \rightarrow p0)) \rightarrow C) \wedge (((p1 \rightarrow p2) \wedge (p2 \rightarrow p1)) \rightarrow C) \wedge \\ (((p2 \rightarrow p0) \wedge (p0 \rightarrow p2)) \rightarrow C) \rightarrow C$$

where $C \equiv (p0 \wedge (p1 \wedge p2))$.

The proof starts with $\mathbf{F}_1 H$. Since H is implicative, $\mathbf{F}_1 \rightarrow$ is applied and we get

$$S_1 = \{ \mathbf{T} ((((p0 \rightarrow p1) \wedge (p1 \rightarrow p0)) \rightarrow C) \wedge (((p1 \rightarrow p2) \wedge (p2 \rightarrow p1)) \rightarrow C) \wedge \\ \wedge (((p2 \rightarrow p0) \wedge (p0 \rightarrow p2)) \rightarrow C)), \\ \mathbf{F}_1 C, \mathbf{T}_n C \}$$

By applying two times $\mathbf{T}\wedge$, two times $\mathbf{T}_n\wedge$ and three times $\mathbf{T} \rightarrow \wedge$ we get

$$S_2 = \{ \mathbf{T}_n p1, \mathbf{T}_n p2, \mathbf{T}_n p0, \mathbf{T}((p2 \rightarrow p0) \rightarrow ((p0 \rightarrow p2) \rightarrow H)), \\ \mathbf{T}((p1 \rightarrow p2) \rightarrow ((p2 \rightarrow p1) \rightarrow H)), \\ \mathbf{T}((p0 \rightarrow p1) \rightarrow ((p1 \rightarrow p0) \rightarrow H)), \mathbf{F}_1 C \}$$

We apply to S_2 the rule $\mathbf{T} \rightarrow \rightarrow$, having $\mathbf{T}((p2 \rightarrow p0) \rightarrow ((p0 \rightarrow p2) \rightarrow H))$ as main premise. The result are two sets, we only consider the leftmost

$$S_3 = \{ \mathbf{F}(p2 \rightarrow p0), \mathbf{T}(p0 \rightarrow ((p0 \rightarrow p2) \rightarrow H)), \mathbf{T}_n p1, \mathbf{T}_n p2, \mathbf{T}_n p0, \\ \mathbf{T}((p1 \rightarrow p2) \rightarrow ((p2 \rightarrow p1) \rightarrow H)), \mathbf{T}((p0 \rightarrow p1) \rightarrow ((p1 \rightarrow p0) \rightarrow H)), \mathbf{F}_1 C \}$$

We apply to S_3 the rule $\mathbf{F} \rightarrow_1$ and SIMPLIFICATION as long as possible. The result is

$$S_4 = \{ \mathbf{T}p2, \mathbf{F}_1 p0, \mathbf{T}(p1 \rightarrow (p0 \wedge p1)), \mathbf{T}(p0 \rightarrow (p0 \wedge p1)), \mathbf{T}_n p1, \mathbf{T}_n p0, \\ \mathbf{T}((p0 \rightarrow p1) \rightarrow ((p1 \rightarrow p0) \rightarrow (p0 \wedge p1))), \mathbf{F}_1(p0 \wedge p1) \}$$

We apply to S_4 the rule $\mathbf{T} \rightarrow \rightarrow$. The results are two sets, we only consider the leftmost:

$$S_5 = \{ \mathbf{F}(p0 \rightarrow p1), \mathbf{T}(p1 \rightarrow ((p1 \rightarrow p0) \rightarrow (p0 \wedge p1))), \mathbf{T}p2, \mathbf{F}_1 p0, \mathbf{T}(p1 \rightarrow (p0 \wedge p1)), \\ \mathbf{T}(p0 \rightarrow (p0 \wedge p1)), \mathbf{T}_n p1, \mathbf{T}_n p0, \mathbf{F}_1(p0 \wedge p1) \}$$

We apply to S_5 the rule $\mathbf{F} \rightarrow_1$ and then SIMPLIFICATION as long as possible. We get

$$S_6 = \{ \mathbf{T}p0, \mathbf{T}p2, \mathbf{T}p1, \mathbf{F}_1 \top \}$$

which is contradictory. Along the same line we get a contradiction from the sets we left.

5 Implementation and Experimental Results

The prolog prototype EPDL-new¹ implements the strategy described in Section 4. The developing effort is not in the implementation of particular data structures but on the strategy to reduce the branching. As a result, the application of the reduction rules requires a quadratic number of steps in the length of the premise. By using the advanced data structures employed in [2], such a number can be reduced to be constant. Since the replacement rules have been proved useful to speed-up the intuitionistic deduction ([2]), beside the replacement rules of Figure 3, EPDL-new implements the replacement rules:

$$\frac{S, \mathbf{T}A, \mathbf{F}_1B}{S, \mathbf{T}A, \mathbf{T}_nA[B/\top], \mathbf{F}_1B} \text{Replace } \mathbf{F}_1\text{-dup} \qquad \frac{S, \mathbf{T}A, \mathbf{T}_nB}{S, \mathbf{T}A, \mathbf{T}_nA[B/\top], \mathbf{T}_nB} \text{Replace } \mathbf{T}_n\text{-dup}$$

provided a \mathbf{F}_n -formula is in S

The advantage of these two further replacement rules is that they are a mechanism to discover information that will be forced in the next state of knowledge. Since the sign \mathbf{T} subsumes \mathbf{T}_n , an alternative is to introduce a \mathbf{T}_n -formula for every \mathbf{T} -formula in the set, but this would increase the number of \mathbf{T}_n -formulas also in cases where this information cannot be exploited. With our choice a \mathbf{T}_n -formula is introduced only when a new information about the future is explicitly known. The benefit of discovering as much \mathbf{T}_n -formulas as possible is related to the rule $\mathbf{F} \rightarrow_1$, a particular case of $\mathbf{F} \rightarrow$, that has one conclusion. The proviso on the application of the rules guarantees that we do not waste time by applying the replacement rules if there is no witness that a future state exists. Such a condition is necessary to guarantee the correctness of

$$\frac{S, \mathbf{T}\neg A, \mathbf{T}_nB}{S, \mathbf{T}\neg A[B/\top], \mathbf{T}_nB} \text{Replace } \mathbf{T}_n\text{-special}$$

provided a \mathbf{F}_n -formula is in S

a version of $\text{Replace } \mathbf{T}_n\text{-dup}$ that does not require duplication of the negated \mathbf{T} -formula. Since $\text{Replace } \mathbf{T}_n\text{-dup}$ and $\text{Replace } \mathbf{F}_1\text{-dup}$ copy the premise $\mathbf{T}A$, to avoid an infinite loops in the application of the rules a special labelling on the copied formulas is implemented.

In Fig. 5 we report the comparisons with EPDL ([10]) on the formulas of ILTP library (the ILTP family formulas from SYJ201 to SYJ206 are valid in Dummett logic). On the missing families the behaviour of the implementations is similar. On the the families of formulas SYJ201, SYJ205, SYJ207 and SYJ208 EPDL-new is clearly faster than EPDL and the timings of EPDL-new grow slower than EPDL. As regard the family SYJ202 (the pigeon principle) EPDL-new is clearly slower. This is due to the overhead to manage the \mathbf{T}_n -formulas introduced in the deduction that do not give any advantage to decide this formula. The overhead to manage the \mathbf{T}_n -formulas is twofold. In the deduction occur formulas of the kind $\mathbf{T}_n(A \wedge B)$, thus the decision procedure applies the rule $\mathbf{T}_n\wedge$. Moreover, the application of the rules in Fig. 3 and 4 requires, for both EPDL and EPDL-new, quadratic time in the number of connectives and atoms in the set of formulas. Since the set managed by EPDL-new is bigger than EPDL we have such an increment in the timings. As we noticed at the beginning of this section, by employing the data structures of [2], the time required to apply the reduction rules can be lowered from quadratic to constant in the number of symbols of the set to be handled. Thus the impact of the overhead is greatly reduced. As regard the family formulas SYJ211, we notice that the different timings have to be charged to the way the reduction rules are implemented.

¹available from <http://www.dimequant.unimib.it/~guidofiorino/epdl.jsp>

Formula	EPDL	EPDL-new	Formula	EPDL	EPDL-new
SYJ201+1.002	0.21	0.03	SYJ208+1.009	6.55	1.70
SYJ201+1.003	2.73	0.07	SYJ208+1.010	14.75	3.05
SYJ201+1.004	33.63	0.14	SYJ208+1.011	30.06	5.03
SYJ201+1.005	358.07	0.26	SYJ208+1.012	71.28	8.23
SYJ202+1.004	0.12	0.23	SYJ211+1.017	0.17	1.05
SYJ202+1.005	1.02	2.24	SYJ211+1.018	0.20	1.16
SYJ202+1.006	9.32	27.21	SYJ211+1.019	0.22	1.38
SYJ202+1.007	100	335	SYJ211+1.020	0.26	1.59
SYJ205+1.009	12.04	2.18	SYJ212+1.011	0.25	0.90
SYJ205+1.010	30.44	3.00	SYJ212+1.012	0.49	1.90
SYJ205+1.011	75.57	3.84	SYJ212+1.013	1.18	4.12
SYJ205+1.012	187.17	5.06	SYJ212+1.014	2.22	8.25
SYJ207+1.003	0.27	0.04			
SYJ207+1.004	3.33	0.10			
SYJ207+1.005	37.78	0.17			
SYJ207+1.006	419.80	0.27			

Figure 5: EPDL and New-EPDL on ILTP formulas.

As a matter of fact, to decide the last formula of the family EPDL takes 127 rules and EPDL-new 147, where the difference in the number of the rules is due to the application of $\mathbf{T}_n \wedge$ -rule. We also remark that the growing ratio of the two implementations is similar. Finally EPDL decides SYJ212+1.014 by applying 54 rules whereas EPDL-new applies 81. These formulas are huge (SYJ212+1.014 is a formula in 14 variables, containing 131063 connectives and 98296 variable occurrences). Because of the duplication of the calculus, the sets of formulas managed by EPDL-new contains more symbols than those managed by EPDL and the difference in timings is mostly chargeable to the implementation of the reduction rules rather than to the difference on the number of applications of rules. Another clue to support our statement is that EPDL-new requires to apply 63, 69, 75 and 81 rules to decide respectively SYJ212+1.011, SYJ212+1.012, SYJ212+1.013 and SYJ212+1.014, whereas to decide the same formulas EPDL applies respectively 42, 46, 50 and 54 rules. We remark that the number of rules increases of a constant value for both provers, but the timings of the EPDL and EPDL-new approximately double.

Summarizing, on some families of formulas the ideas on which the calculus is based on do not apply, as a result the deduction is slowed-down. This difference is remarkable on the family SYJ202, the pigeon principle, that, apart from the first rule, is decided by a deduction requiring rules for connectives \wedge and \vee . We point out that on the families SYJ202, SYJ211 and SYJ212 the multiple premise rule \mathbf{F}_n is not applied. We conjecture that implementing the calculus with better data structures, on the family formulas SYJ211 and SYJ212 the difference in timings between EPDL and EPDL-new can be reduced almost to zero. Finally we remark that on the other families the deduction requires some steps of the multiple premise rule \mathbf{F}_n and in this case both timings and growing ratio of EPDL-new are lower than EPDL.

6 Conclusions and future work

The main novelty of the calculus presented in this paper is the presence of rules tailored to draw deductions about the (immediate) future and rules exploiting the information about facts known in the future to draw facts about the present state of knowledge. The aim is to reduce the branching of the proof to decide a formula in Dummett logic. The presence of the signs \mathbf{F}_1 and \mathbf{T}_n allow us to introduce specialized rules in particular to handle implicative formulas.

Rule $\mathbf{F} \rightarrow_1$ is an example but other specialized rules can be introduced:
$$\frac{S, \mathbf{F}(A \rightarrow B), \mathbf{F}_1 A}{S, \mathbf{F}_1 A, \mathbf{F}_n B}$$

is a further example of single concluded rule for $\mathbf{F} \rightarrow$ -formulas. Also formulas of the kind $\mathbf{T}((A \rightarrow B) \rightarrow C)$ can be handled by means of specialized rules:

$$\frac{S, \mathbf{T}((A \rightarrow B) \rightarrow C), \mathbf{F}_1(A \rightarrow B)}{S, \mathbf{F}_1(A \rightarrow B), \mathbf{T}_n C} \quad \frac{S, \mathbf{T}((A \rightarrow B) \rightarrow C), \mathbf{F}_1 C}{S, \mathbf{F}(A \rightarrow B), \mathbf{F}_1 C}$$

$$\frac{S, \mathbf{T}((A \rightarrow B) \rightarrow C), \mathbf{T}_n(A \rightarrow B)}{S, \mathbf{F}_1(A \rightarrow B), \mathbf{T}_n C | S, \mathbf{T}_n(A \rightarrow B), \mathbf{TC}} \quad \frac{S, \mathbf{T}((A \rightarrow B) \rightarrow C), \mathbf{T}_n C}{S, \mathbf{F}(A \rightarrow B), \mathbf{T}_n C | S, \mathbf{TC}}$$

These rules are a particular version of $\mathbf{T} \rightarrow \rightarrow$ allowing to reduce branching or the size of the conclusion.

As related works, we quote the implementation LC-cmodels ([16]), compared with EPDL in [10]. We also quote the approach of [3], where propositional Dummett logic is decided via a decision procedure for propositional Intuitionistic logic. Paper [3] introduces the notion of Generalized Tableaux to decide intermediate logics. A Generalized Tableau is a tableau for propositional Intuitionistic logic plus a rule to be applied once as first rule of the deduction. The aim of this rule is to introduce formulas obtained by instantiating the axiom scheme of the logic under consideration. For the case of Dummett logic, to decide a given formula A , the special rule introduces the set of formulas obtained by instantiating in every possible way the propositional variables the axiom schemata $(p \rightarrow q) \vee (q \rightarrow p)$ with the formulas in $Rsf(A) = \{B | B \text{ is subformula of } A \text{ and } B \text{ is a propositional variable or } B \equiv C \rightarrow D \text{ or } B \equiv \neg C\}$. Since $|Rsf(A)| = O(|A|)$ and there are $|Rsf(A)|$ choices for p and q , it follows that the special rule introduces $O(|A|^2)$ formulas ($|A|$ denotes the cardinality of A). Thus the number of connectives to be handled in the deduction is $O(|A|^3)$. Paper [15] proves that propositional intuitionistic logic is decidable in $O(n \lg n)$ -SPACE, hence this technique requires $O(n^3 \lg n)$ -SPACE and the depth of the deductions is $O(|A|^3)$. Another approach is the translation of A into a formula to be decided in classical logic. One can exploit the fact that a formula with n propositional variables is satisfiable in a Kripke model having $n + 1$ worlds at most and writing a formula A' expressing that for every B and C , subformulas of A , $B \rightarrow C$ or $C \rightarrow B$ holds. Moreover also the persistence of the forcing has to be expressed. Thus the size of A' is $O(|A|^3)$.

The ideas presented in this paper for propositional Dummett logic can be applied to intuitionistic deduction too. Here we sketch how to design a calculus for the propositional implicative fragment. The sign \mathbf{F}_n is no longer necessary. Formulas of the kind $\mathbf{F}(A \rightarrow B)$ are handled

by the rule
$$\frac{S, \mathbf{F}(A \rightarrow B)}{S, \mathbf{T}_A, \mathbf{F}_1 B | S_c, \mathbf{T}_A, \mathbf{F}_1 B}$$
 and formulas of the kind $\mathbf{T}(A \rightarrow B)$ are handled by the rule
$$\frac{S, \mathbf{T}(A \rightarrow B)}{S, \mathbf{T}_B | S, \mathbf{F}_1 A, \mathbf{T}_n B | S_c, \mathbf{F}_1 A, \mathbf{T}_n B}$$
. We emphasize that these rules respect the subformula

property, thus we are on the way to design a duplication-free calculus (in the sense of [1]) respecting the subformula property, as long as the replacement rules are not employed. As regard

the decision procedure equipped with the replacement rules, we expect to improve the performances of fCube ([8]) as EPDL-new improves the performances of EPDL. Our consideration is based on the fact that, since intuitionistic logic is in PSPACE whereas Dummett logic is in NP, the overhead due to the application of the replacement and special rules is less in intuitionistic than in Dummett logic.

References

- [1] A. Avellone, M. Ferrari, and P. Miglioli. Duplication-free tableau calculi and related cut-free sequent calculi for the interpolable propositional intermediate logics. *Logic Journal of the IGPL*, 7(4):447–480, 1999.
- [2] A. Avellone, G. Fiorino, and U. Moscato. Optimization techniques for propositional intuitionistic logic and their implementation. *Theoretical Computer Science*, 409(1):41–58, 2008.
- [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 Reasoning with Analytic Tableaux and Related Methods: Tableaux '97*, volume 1227 of *LNAI*, pages 43–61. Springer-Verlag, 1997.
- [4] A. Avron. Simple consequence relations. *Journal of Information and Computation*, 92:276–294, 1991.
- [5] A. Avron and B. Konikowska. Decomposition proof systems for gödel-dummett logics. *Studia Logica*, 69(2):197–219, 2001.
- [6] M. Baaz, A. Ciabattoni, and C. G. Fermüller. Hypersequent calculi for Gödel logics – a survey. *J. of Logic and Computation*, 13(6):835–861, 2003.
- [7] M. Dummett. A propositional calculus with a denumerable matrix. *Journal of Symbolic Logic*, 24:96–107, 1959.
- [8] C. Fiorentini M. Ferrari and G. Fiorino. fcube: An efficient prover for intuitionistic propositional logic. In *LPAR 2010*.
- [9] G. Fiorino. An $O(n \log n)$ -SPACE decision procedure for the propositional Dummett Logic. *Journal of Automated Reasoning*, 27(3):297–311, 2001.
- [10] G. Fiorino. Fast decision procedure for propositional dummett logic based on a multiple premise tableau calculus. *Information Sciences*, 180(19):3633 – 3646, 2010.
- [11] M.C. Fitting. *Intuitionistic Logic, Model Theory and Forcing*. North-Holland, 1969.
- [12] K. Gödel. On the intuitionistic propositional calculus. In S. Feferman et al, editor, *Collected Works*, volume 1. Oxford University Press, 1986.
- [13] R. Hähnle. Tableaux and related methods. In John Alan Robinson and Andrei Voronkov, editors, *Handbook of Automated Reasoning*, pages 100–178. Elsevier and MIT Press, 2001.
- [14] P. Hajek. *Metamathematics of Fuzzy Logic*. Kluwer, 1998.
- [15] J. Hudelmaier. An $O(n \log n)$ -SPACE decision procedure for intuitionistic propositional logic. *Journal of Logic and Computation*, 3(1):63–75, 1993.
- [16] D. Larchey-Wendling. Graph-based decision for Gödel-Dummett logics. *J. Autom. Reasoning*, 38(1-3):201–225, 2007.
- [17] F. Massacci. Simplification: A general constraint propagation technique for propositional and modal tableaux. In Harrie de Swart, editor, *Proc. International Conference on Automated Reasoning with Analytic Tableaux and Related Methods, Oosterwijk, The Netherlands*, volume 1397 of *LNCS*, pages 217–232. Springer-Verlag, 1998.
- [18] R.M. Smullyan. *First-Order Logic*. Springer, Berlin, 1968.