

Taming Implications in Dummett Logic

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. This paper discusses a new strategy to decide Dummett logic. The strategy relies on a tableau calculus whose distinguishing features are the rules for implicative formulas. The strategy has been implemented and the experimental results are reported.

Key words: Dummett Logic, Tableau Calculi, Automated Theorem Proving

1 Introduction

The aim of this paper is to provide some ideas to reduce the search space of proofs in Dummett logic. Our results apply when implicative formulas have to be handled. The results are provided in the framework of tableau proof systems and they are explained on the basis of the Kripke semantics for Dummett 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 ([8]). In paper [4] 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 $(p \rightarrow q) \vee (q \rightarrow p)$.

Dummett logic has been extensively studied also in recent years for its relationships with computer science ([2]) and fuzzy logics ([9]). To perform automated deduction both tableau and sequent calculi have been proposed. Paper [1] provides tableau calculi having the distinguishing feature that a multiple premise rule for implicative formulas signed with **F** is provided. We recall that the sign **F** comes from Smullyan ([14, 7]) and labels those formulas that in a sequent calculus occur in the right-hand side of \Rightarrow (as it is explained in Section 2, the sign **F** has a meaning also in terms of Kripke semantics). A tableau calculus derived from those of [1] is provided in paper [5]. 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] characterizing Dummett logic by means 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 [3] 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 in [10,11], and in paper [12] a graph-theoretic decision procedure is described and implemented. The approach introduced in [3] has also disadvantages with respect to the multiple premise rule proposed in [1] and these disadvantages have been considered in [6], 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 [6] also provides an implementation that outperforms the one of [12], 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 [3] 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 [3].

The calculi quoted above have the same kind of rule to treat formulas of the kind $\mathbf{T}((A \rightarrow B) \rightarrow C)$, that is formulas that in a sequent calculus would appear in the left-hand side of \Rightarrow :

$$\frac{S, \mathbf{T}((A \rightarrow B) \rightarrow C)}{S, \mathbf{F}(A \rightarrow B), \mathbf{T}(B \rightarrow C) | S, \mathbf{TC}} \quad \frac{\Gamma, B \rightarrow C \Rightarrow \Delta, A \rightarrow B \quad \Gamma, C \Rightarrow \Delta}{\Gamma, (A \rightarrow B) \rightarrow C \Rightarrow \Delta}$$

Whatever system is used, it is not considered that in the (sub)deduction starting from $S, \mathbf{F}(A \rightarrow B), \mathbf{T}(B \rightarrow C)$, respectively starting from the premise $\Gamma, B \rightarrow C \Rightarrow \Delta, A \rightarrow B$, if $\mathbf{F}(A \rightarrow B)$ occurs in the set, respectively $A \rightarrow B$ occurs in the right-hand side of \Rightarrow , then the completeness is preserved also if no rule is applied to $\mathbf{T}(B \rightarrow C)$, respectively to $B \rightarrow C$ in the left-hand side of \Rightarrow . If B is an implicative formula this strategy avoids to introduce new branches. An analogous remark applies to the case of a set containing $\mathbf{F}B$ and $\mathbf{T}(B \rightarrow C)$, respectively to a sequent of the kind $\Gamma, B \rightarrow C \Rightarrow B, \Delta$. In this paper these remarks are developed and a tableau calculus is provided. A complete strategy is presented and the experimental results of the prolog implementation are compared with the prolog implementation of [6].

2 Basic Definitions

We consider the propositional language based on a denumerable set of propositional variables \mathcal{PV} and the logical connectives $\neg, \wedge, \vee, \rightarrow$. 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 $P \times \mathcal{PV}$ such that if $\alpha \Vdash p$ and $\alpha \leq \beta$, then $\beta \Vdash p$.

Hereafter we denote the members of P , also called *worlds* or *states*, by means of lowercase letters of the Greek alphabet.

The forcing relation is extended in a standard way to arbitrary formulas of the 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 $\beta \Vdash A$. An element $\beta \in P$ is *immediate successor* of $\alpha \in P$ iff if $\alpha \leq \gamma \leq \beta$ holds, then $\alpha = \gamma$ or $\beta = \gamma$ holds. A formula A is *valid in a model* $\underline{K} = \langle P, \leq, \rho, \Vdash \rangle$ iff $\rho \Vdash A$. It is well-known that **Dum** coincides with the set of formulas valid in all models.

The rules of our calculus \mathbb{D} for **Dum** are in Figures 1 and 2. \mathbb{D} works on signed formulas, that is well-formed formulas prefixed with one of the *signs* $\{\mathbf{T}, \mathbf{F}, \mathbf{F}_c, \mathbf{T}_{cl}, \tilde{\mathbf{T}}, \hat{\mathbf{T}}\}$, and on sets of signed formulas (hereafter we omit the word “signed” in front of “formula” in all the contexts where no confusion arises). Before to give the intuition behind the rules of the calculus 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. $\alpha \triangleright \tilde{\mathbf{T}}A$ iff $\alpha \triangleright \mathbf{T}A$ and if $A \equiv (B \rightarrow C)$, then $\alpha \not\Vdash B$;
3. $\alpha \triangleright \hat{\mathbf{T}}A$ iff $\alpha \triangleright \mathbf{T}A$ and if $A \equiv B \rightarrow C$, then there exists $\beta \in P$ such that $\alpha < \beta$ and $\beta \not\Vdash B$;
4. $\alpha \triangleright \mathbf{F}A$ iff $\alpha \not\Vdash A$;
5. $\alpha \triangleright \mathbf{F}_c A$ iff $\alpha \Vdash \neg A$;
6. $\alpha \triangleright \mathbf{T}_{cl} A$ iff $\alpha \Vdash \neg \neg A$;
7. $\alpha \triangleright S$ iff α realizes every formula in S ;

By the semantical meaning of the signs it follows that \mathbf{F}_c and \mathbf{T}_{cl} are synonyms respectively of $\mathbf{T}\neg$ and $\mathbf{T}\neg\neg$, thus \mathbf{F}_c and \mathbf{T}_{cl} -rules are the rules to treat respectively negated and double negated forced formulas. The \mathbf{F}_c and \mathbf{T}_{cl} -rules are designed taking into account that **Dum** is characterized by linearly ordered Kripke models. The signs $\tilde{\mathbf{T}}$ and $\hat{\mathbf{T}}$ are a specialization of the sign \mathbf{T} for implicative formulas. The sign $\tilde{\mathbf{T}}$ in front of $B \rightarrow C$ conveys the information that at the present state of knowledge the formulas $B \rightarrow C$ and B are respectively forced and not forced. The sign $\hat{\mathbf{T}}$ in front of $B \rightarrow C$ conveys both the information that at the present state of knowledge $B \rightarrow C$ is forced and the information that there exists a future state of knowledge where B is not forced. This information is available in the conclusion $S, \mathbf{F}(A \rightarrow B), \mathbf{T}(B \rightarrow C)$

of the tableau rule
$$\frac{S, \mathbf{T}((A \rightarrow B) \rightarrow C)}{S, \mathbf{F}(A \rightarrow B), \mathbf{T}(B \rightarrow C) | S, \mathbf{TC}}$$
 of the calculi [1, 5, 6], but it

is not exploited. Analogously for the sequent calculi of [3, 12], where the same information is available in the premise $\Gamma, B \rightarrow C \Rightarrow \Delta, A \rightarrow B$ of the rule
$$\frac{\Gamma, B \rightarrow C \Rightarrow \Delta, A \rightarrow B \quad \Gamma, C \Rightarrow \Delta}{\Gamma, (A \rightarrow B) \rightarrow C \Rightarrow \Delta}$$
. Hereafter we consider tableau calculi.

The same remarks apply to sequent calculi. None of the above quoted calculi has rules taking into account that if $A \rightarrow B$ is not forced, then B is not forced and thus $B \rightarrow C$ is (locally) forced. The formula $B \rightarrow C$ needs to be treated only when disappear the information about the non-forcing of B . Thus, roughly speaking, the main idea of the calculus we are presenting can be summarized as follows: *a formula of the kind $\mathbf{T}(A \rightarrow B)$ does not need to be treated if there is the information that A is not forced*. In this case, if A is of the kind $C \rightarrow D$ a branch is avoided. From a syntactical point of view, the sign $\widehat{\mathbf{T}}$ in front of $A \rightarrow B$ means that in the set at hand A occurs as a consequent of an $\mathbf{F} \rightarrow$ -formula. The sign $\widetilde{\mathbf{T}}$ in front of $A \rightarrow B$ means that in the set at hand the formula $\mathbf{F}A$ occurs. The rules are designed to guarantee that the presence of a formula $\widehat{\mathbf{T}}(B \rightarrow C)$ in a set S implies that S also contains the formula $\mathbf{F}(A \rightarrow B)$. Note that $\mathbf{T} \rightarrow$ is the only rule of \mathbb{D} to introduce $\widehat{\mathbf{T}}$ -formulas. The presence of the formula $\widetilde{\mathbf{T}}(A \rightarrow B)$ in a set S which is a conclusion of the rule $\mathbf{F} \rightarrow$, implies that also $\mathbf{F}B$ is in the set. The rules of the calculus behaves on $\widetilde{\mathbf{T}}(A \rightarrow B)$ as they were two premise rules on the formulas $\mathbf{F}A, \mathbf{T}(A \rightarrow B)$. The calculus has not rules to treat the $\widehat{\mathbf{T}}$ -formulas. These formulas are treated by the rule $\mathbf{F} \rightarrow$ and they can be left unchanged or turned into $\widetilde{\mathbf{T}}$ -formulas. It is remarked that $\widetilde{\mathbf{T}}$ and $\widehat{\mathbf{T}}$ are in front of implicative formulas only.

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:

$$\begin{aligned} &-\{\mathbf{T}A, \mathbf{F}A\} \subseteq S; & &-\{\widetilde{\mathbf{T}}A, \mathbf{F}A\} \subseteq S; & &-\{\widetilde{\mathbf{T}}A, \mathbf{F}_cA\} \subseteq S; \\ &-\{\mathbf{T}A, \mathbf{F}_cA\} \subseteq S; & &-\{\widehat{\mathbf{T}}A, \mathbf{F}A\} \subseteq S; & &-\{\widehat{\mathbf{T}}A, \mathbf{F}_cA\} \subseteq S; \\ &-\{\mathbf{T}A, \widetilde{\mathbf{T}}(A \rightarrow B)\} \subseteq S; & &-\{\mathbf{F}_cA, \mathbf{T}_{cl}A\} \subseteq S. \end{aligned}$$

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 $\alpha \in P$, $\alpha \not\vdash S$.*

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 $\{\mathbf{F}A\}$.

$$\begin{array}{c}
\frac{S, \mathbf{T}(A \wedge B)}{S, \mathbf{TA}, \mathbf{TB}} \mathbf{T}^\wedge \quad \frac{S, \mathbf{F}(A \wedge B)}{S, \mathbf{FA}|S, \mathbf{FB}} \mathbf{F}^\wedge \quad \frac{S, \mathbf{F}_c(A \wedge B)}{S, \mathbf{F}_cA|S, \mathbf{F}_cB} \mathbf{F}_{c^\wedge} \quad \frac{S, \mathbf{T}_{cl}(A \wedge B)}{S, \mathbf{T}_{cl}A, \mathbf{T}_{cl}B} \mathbf{T}_{cl}^\wedge \\
\frac{S, \mathbf{T}(A \vee B)}{S, \mathbf{TA}|S, \mathbf{TB}} \mathbf{T}^\vee \quad \frac{S, \mathbf{F}(A \vee B)}{S, \mathbf{FA}, \mathbf{FB}} \mathbf{F}^\vee \quad \frac{S, \mathbf{F}_c(A \vee B)}{S, \mathbf{F}_cA, \mathbf{F}_cB} \mathbf{F}_{c^\vee} \quad \frac{S, \mathbf{T}_{cl}(A \vee B)}{S, \mathbf{T}_{cl}A|S, \mathbf{T}_{cl}B} \mathbf{T}_{cl}^\vee \\
\frac{S, \mathbf{T}(\neg A)}{S, \mathbf{F}_cA} \mathbf{T}^\neg \quad \frac{S, \mathbf{F}(\neg A)}{S, \mathbf{T}_{cl}A} \mathbf{F}^\neg \quad \frac{S, \mathbf{F}_c(\neg A)}{S, \mathbf{T}_{cl}A} \mathbf{F}_{c^\neg} \quad \frac{S, \mathbf{T}_{cl}(\neg A)}{S, \mathbf{F}_cA} \mathbf{T}_{cl}^\neg \\
\frac{S, \mathbf{Tp}, \mathbf{T}(p \rightarrow B)}{S, \mathbf{Tp}, \mathbf{TB}} \mathbf{T}^\rightarrow \quad \frac{S, \mathbf{F}_c(A \rightarrow B)}{S, \mathbf{T}_{cl}A, \mathbf{F}_cB} \mathbf{F}_{c^\rightarrow} \quad \frac{S, \mathbf{T}_{cl}(A \rightarrow B)}{S, \mathbf{F}_cA|S, \mathbf{T}_{cl}B} \mathbf{T}_{cl}^\rightarrow \\
\frac{S, \mathbf{T}((A \wedge B) \rightarrow C)}{S, \mathbf{T}(A \rightarrow (B \rightarrow C))} \mathbf{T}^{\rightarrow\wedge} \quad \frac{S, \mathbf{T}(\neg A \rightarrow B)}{S, \mathbf{T}_{cl}A|S, \mathbf{TB}} \mathbf{T}^{\rightarrow\neg} \quad \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 B), \widehat{\mathbf{T}}(B \rightarrow C), |S, \mathbf{TC}} \mathbf{T}^{\rightarrow\rightarrow} \\
\frac{S, \widehat{\mathbf{T}}((A \wedge B) \rightarrow C)}{S, \widehat{\mathbf{T}}(A \rightarrow (B \rightarrow C))|S, \widehat{\mathbf{T}}(B \rightarrow (A \rightarrow C))} \widetilde{\mathbf{T}}^{\rightarrow\wedge} \\
\frac{S, \widetilde{\mathbf{T}}(\neg A \rightarrow B)}{S, \mathbf{T}_{cl}A} \widetilde{\mathbf{T}}^{\rightarrow\neg} \quad \frac{S, \widetilde{\mathbf{T}}((A \vee B) \rightarrow C)}{S, \widetilde{\mathbf{T}}(A \rightarrow C), \widetilde{\mathbf{T}}(B \rightarrow C)} \widetilde{\mathbf{T}}^{\rightarrow\vee} \\
\frac{S, \widetilde{\mathbf{T}}((A \rightarrow B) \rightarrow C)}{S, \mathbf{F}(A \rightarrow B), \widehat{\mathbf{T}}(B \rightarrow C)} \widetilde{\mathbf{T}}^{\rightarrow\rightarrow}
\end{array}$$

Fig. 1. The invertible rules of \mathbb{D} .

$$\begin{array}{c}
\frac{S, \mathbf{T}_{cl}A}{S_c, \mathbf{TA}} \mathbf{T}_{cl}\text{-Atom} \\
S_c = \{\mathbf{TA}|\mathbf{TA} \in S\} \cup \{\mathbf{F}_cA|\mathbf{F}_cA \in S\} \cup \{\mathbf{TA}|\mathbf{T}_{cl}A \in S\} \cup \\
\cup \{\mathbf{TA}|\widetilde{\mathbf{T}}A \in S\} \cup \{\mathbf{TA}|\widehat{\mathbf{T}}A \in S\}; \\
\frac{S, \mathbf{F}(A_1 \rightarrow B_1), \dots, \mathbf{F}(A_n \rightarrow B_n)}{S_c^1, \mathbf{TA}_1, \mathbf{FB}_1, S_{\mathbf{F}^\rightarrow}^1 | S_c^2, S_{\mathbf{F}^\rightarrow}^2, \mathbf{TA}_2, \mathbf{FB}_2 | \dots | S_c^n, S_{\mathbf{F}^\rightarrow}^n, \mathbf{TA}_n, \mathbf{FB}_n} \mathbf{F}^\rightarrow \\
S_c^i = \{\mathbf{TA}|\mathbf{TA} \in S\} \cup \{\mathbf{F}_cA|\mathbf{F}_cA \in S\} \cup \{\mathbf{T}_{cl}A|\mathbf{T}_{cl}A \in S\} \cup \\
\cup \{\mathbf{TA}|\widehat{\mathbf{T}}A \in S\} \cup \{\widehat{\mathbf{T}}(C \rightarrow D) | \widehat{\mathbf{T}}(C \rightarrow D) \in S \text{ and } C = B_i\} \cup \\
\cup \{\widehat{\mathbf{T}}(C \rightarrow D) | \widehat{\mathbf{T}}(C \rightarrow D) \in S \text{ with } C \neq B_i\}; \\
S_{\mathbf{F}^\rightarrow} = \{\mathbf{F}(A_1 \rightarrow B_1), \dots, \mathbf{F}(A_n \rightarrow B_n)\}, \\
S_{\mathbf{F}^\rightarrow}^i = S_{\mathbf{F}^\rightarrow} \setminus \{\mathbf{F}(A_i \rightarrow B_i)\}, \text{ for } i = 1, \dots, n.
\end{array}$$

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

The set $U = \{\mathbf{T}(B \wedge C), \mathbf{T}(A \wedge C), \mathbf{F}(A \vee B)\}$ put in evidence that both the rule \mathbf{T}^\wedge (taking $S = \{\mathbf{T}(A \wedge C), \mathbf{F}(A \vee B)\}$ or $S = \{\mathbf{T}(B \wedge C), \mathbf{F}(A \vee B)\}$) and the rule \mathbf{F}^\vee apply to U . This gives rise to three choices to go on with the proof. After the choice is done, if there is no way to prove the conclusion of the

application of the rule, then the question is if another choice had given a proof. If the rule is *invertible*, then there is no need to backtrack on another rule: A rule is *invertible* iff if there exists a proof for the premise, then there exists a proof for the conclusion. The notion of invertible rule is also definable via semantics: a rule is invertible iff if one of the sets in the conclusion is realizable by a model \underline{K} , then the premise is realizable by \underline{K} . It is well-known that the invertibility of the rules of the calculus is a desirable property, since it implies that every choice is deterministic. The calculus \mathbb{D} has two non-invertible rules, namely $\mathbf{F} \rightarrow$ and $\mathbf{T}_{cl}\text{-Atom}$. In Section 4 we present a complete strategy such that every choice is deterministic. The strategy relies on respecting a particular sequence in the application of the rules: $\mathbf{T}_{cl}\text{-Atom}$ is applied if no other rule is applicable and $\mathbf{F} \rightarrow$ is applied if no other rule but $\mathbf{T}_{cl}\text{-Atom}$ is applicable.

3 Correctness

The following proposition states that the rules in Tables 1 and 2 preserve the realizability. This is the main step towards to prove the correctness of \mathbb{D} .

Lemma 1. *For every rule of \mathbb{D} , if a world α of a model $\underline{K} = \langle P, \leq, \rho, \Vdash \rangle$ realizes the premise, then there exists a world of a possibly different model realizing at least one of the conclusions.*

Proof. The proof proceeds by taking into account every rule of \mathbb{D} . Here the proof of the correctness of rule $\mathbf{T} \rightarrow \rightarrow$ is provided. Rule $\mathbf{T} \rightarrow \rightarrow$: by hypothesis $\alpha \triangleright S, \mathbf{T}((A \rightarrow B) \rightarrow C)$. By definition of forcing of implication we have two cases: (i) $\alpha \Vdash C$, thus $\alpha \triangleright S, \mathbf{TC}$; (ii) $\alpha \not\Vdash (A \rightarrow B)$. This implies that $\alpha \Vdash B \rightarrow C$ and there exists a world $\beta \in P$ such that $\alpha \leq \beta$, $\beta \Vdash A$ and $\beta \not\Vdash B$. If $\alpha < \beta$, then immediately we get $\alpha \triangleright \mathbf{F}(A \rightarrow B), \widehat{\mathbf{T}}(B \rightarrow C)$. If $\alpha = \beta$ let us consider the model $\underline{K}' = \langle P \cup \{\alpha'\}, \leq', \rho, \Vdash' \rangle$ defined as follows:

$$\begin{aligned} \leq' &= \leq \cup \{(\gamma, \alpha') \mid \gamma \in P \text{ and } \gamma \leq \alpha\} \cup \{(\alpha', \gamma) \mid \gamma \in P \text{ and } \alpha < \gamma\}; \\ \Vdash' &= \Vdash \cup \{(\alpha', p) \mid (\alpha, p) \in \Vdash\}. \end{aligned}$$

The model \underline{K}' is obtained from \underline{K} by adding a new world α' as immediate successor of α and defining the forcing in α' as the forcing in α . By structural induction it is easy to prove that in \underline{K}' the worlds α and α' force the same formulas. Moreover $\alpha \Vdash' A$ holds iff $\alpha \Vdash A$ holds. Thus the world α of \underline{K}' realizes the premise of the rule $\mathbf{T} \rightarrow \rightarrow$. Finally, since $\alpha' \not\Vdash' B$ holds, we get that $\alpha \triangleright \widehat{\mathbf{T}}(B \rightarrow C)$ holds. \square

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

4 Complete Strategy to Decide Dummett Logic with \mathbb{D}

In the following we sketch the recursive procedure $\text{DUM}(S)$ using \mathbb{D} to decide S : Given a set S of signed 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 (respectively β -rules) the rules of Figure 1 with one conclusion (respectively with two conclusions). The α -formulas (respectively β -formulas) are the kind of the non-atomic signed formulas in evidence in the premise of the α -rules (respectively β -rules). Let S be a set of formulas, let $H \in S$ be an α or β -formula. With $Rule(H)$ 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 $Rule(H)$. If Tab_1 and Tab_2 are closed proof tables for S_1 and S_2 respectively, then $\frac{S}{Tab_1}Rule(H)$ or $\frac{S}{Tab_1 \mid Tab_2}Rule(H)$ denote the closed proof table for S defined in the obvious way. Moreover, $\mathcal{R}_i(H)$ ($i = 1, 2$) denotes the set containing the formulas of S_i which replaces H . For instance:

$$\mathcal{R}_1(\mathbf{T}(A \wedge B)) = \{\mathbf{TA}, \mathbf{TB}\},$$

$$\mathcal{R}_1(\mathbf{T}(A \vee B)) = \{\mathbf{TA}\}, \mathcal{R}_2(\mathbf{T}(A \vee B)) = \{\mathbf{TB}\}.$$

In the case of $\mathbf{F} \rightarrow$ the notation above generalizes. Let $S_{\mathbf{F} \rightarrow}$ be the set of all the $\mathbf{F} \rightarrow$ -formulas of S . Let $S_1 \mid \dots \mid S_n$ the nodes of the proof tree obtained by applying to S the rule $\mathbf{F} \rightarrow$. If Tab_1, \dots, Tab_n are closed proof tables respectively for S_1, \dots, S_n , then $\frac{S}{Tab_1 \mid \dots \mid Tab_n} \mathbf{F} \rightarrow$ is the closed proof table for S . $\mathcal{R}_i(S_{\mathbf{F} \rightarrow})$ denotes the set of formulas that replace the set $S_{\mathbf{F} \rightarrow}$ in the i -th conclusion of $\mathbf{F} \rightarrow$. For example, given $S_{\mathbf{F} \rightarrow} = \{\mathbf{F}(A_1 \rightarrow B_1), \mathbf{F}(A_2 \rightarrow B_2), \mathbf{F}(A_3 \rightarrow B_3)\}$, $\mathcal{R}_2(S_{\mathbf{F} \rightarrow}) = \{\mathbf{F}(A_1 \rightarrow B_1), \mathbf{TA}_2, \mathbf{FB}_2, \mathbf{F}(A_3 \rightarrow B_3)\}$.

FUNCTION DUM (S)

1. If S is an inconsistent set, then DUM returns the proof S ;
2. If an α -rule applies to S , then let H be a α -formula of S . If $DUM((S \setminus \{H\}) \cup \mathcal{R}_1(H))$ returns a proof π , then DUM returns the proof $\frac{S}{\pi}Rule(H)$, otherwise DUM returns NULL;
3. If a β -rule applies to S , then let H be a β -formula of S . Let $\pi_1 = DUM((S \setminus \{H\}) \cup \mathcal{R}_1(H))$ and $\pi_2 = DUM((S \setminus \{H\}) \cup \mathcal{R}_2(H))$. If π_1 or π_2 is NULL, then DUM returns NULL, otherwise DUM returns $\frac{S}{\pi_1 \mid \pi_2}Rule(H)$;
4. If the rule $\mathbf{F} \rightarrow$ applies to S , then let $S_{\mathbf{F} \rightarrow} = \{\mathbf{F}(A \rightarrow B) \in S\}$ and let n be the number of formulas in $S_{\mathbf{F} \rightarrow}$. If there exists $i \in \{1, \dots, n\}$, such that $\pi_i = DUM((S \setminus S_{\mathbf{F} \rightarrow})_c \cup \mathcal{R}_i(S_{\mathbf{F} \rightarrow}))$ is NULL, then DUM returns NULL. Otherwise π_1, \dots, π_n are proofs and DUM returns $\frac{S}{\pi_1 \mid \dots \mid \pi_n} \mathbf{F} \rightarrow$;
5. If the rule \mathbf{T}_{c1} -Atom applies to S , then let H be a \mathbf{T}_{c1} -Atom formula of S . If $DUM((S \setminus \{H\}) \cup \mathcal{R}_1(H))$ returns a proof π , then DUM returns the proof $\frac{S}{\pi} \mathbf{T}_{c1}$ -Atom, otherwise DUM returns NULL;
6. If none of the previous points apply, then DUM returns NULL.

END FUNCTION DUM.

It is useful to remark the following facts: (i) when Step 4 is performed, S contains atomic formulas, implicative formulas signed with \mathbf{F} or $\widehat{\mathbf{T}}$ and implicative formulas of the kind $\mathcal{S}(p \rightarrow B)$, with $\mathcal{S} \in \{\mathbf{F}, \widehat{\mathbf{T}}\}$. Note that if $\mathcal{S}(p \rightarrow B) \in S$, then $\mathbf{T}p \notin S$ holds. As a matter of fact, if $\{\widehat{\mathbf{T}}(p \rightarrow B), \mathbf{T}p\} \subseteq S$, then S is incon-

sistent and this case is handled in Step 1; if $\{\mathbf{T}(p \rightarrow B), \mathbf{T}p\} \subseteq S$, then Step 2 is applicable; (ii) when Step 5 is applied the formulas of the kind $\mathcal{S}(p \rightarrow A)$ with $\mathcal{S} \in \{\mathbf{T}, \tilde{\mathbf{T}}\}$ are the only kind of non-atomic formulas in S and $\mathbf{T}p \notin S$; (iii) when Step 6 is applied there is no formula signed with \mathbf{T}_{c1} and formulas of the kind $\mathcal{S}(p \rightarrow B)$, with $\mathcal{S} \in \{\mathbf{T}, \tilde{\mathbf{T}}\}$, are the only non-atomic formulas in S and $\mathbf{T}p \notin S$.

The termination of FUNCTION DUM is based on the fact that the rules of \mathbb{D} replace the formulas in evidence in the premise with *simpler formulas*, where *simpler* is based on a measure complexity function. In order to get the completeness of FUNCTION DUM, in the following it is proved that given a set of formulas S , if the call $\text{DUM}(S)$ fails to return a proof for S , then from the non-closed tableau there is enough information to build a model $\underline{K} = \langle P, \leq, \rho, \Vdash \rangle$ such that $\rho \triangleright S$.

Lemma 2 (Completeness). *Let S be a set of formulas and suppose that $\text{DUM}(S)$ returns the NULL value. 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 6 has been performed and this implies that S is not inconsistent (otherwise Step 1 would have been performed) and S only contains atomic formulas signed with \mathbf{T}, \mathbf{F} and \mathbf{F}_c , formulas of the kind $\mathcal{S}(p \rightarrow A)$ with $\mathcal{S} \in \{\mathbf{T}, \tilde{\mathbf{T}}\}$, and $\mathbf{T}p \notin S$. Let $\underline{K} = \langle P, \leq, \rho, \Vdash \rangle$, where $P = \{\rho\}$, $\rho \leq \rho$ and $\rho \Vdash p$ iff $\mathbf{T}p \in S$. \underline{K} is a model. By considering every possible kind of formula in S , it is easy to prove that ρ realizes S .

Step: Let us assume by induction hypothesis that the proposition holds for all sets S' such that $\text{DUM}(S')$ requires less than n recursive calls. The proposition is proved to hold for a set S requiring n recursive calls. All the possible cases where the procedure returns the NULL value have to be inspected. Here we provide the case related to the NULL instruction performed at Step 4. Since the NULL instruction in Step 4 has been performed, at least a π_i is NULL. By induction hypothesis there is a model $\underline{K}' = \langle P, \leq', \rho', \Vdash' \rangle$ realizing $(S \setminus S_{\mathbf{F}_-})_c \cup \mathcal{R}_i(S_{\mathbf{F}_-})$. We define a model $\underline{K} = \langle P \cup \{\rho\}, \leq, \rho, \Vdash \rangle$ as follows:

$$P \cap \{\rho\} = \emptyset; \leq = \leq' \cup \{(\rho, \alpha) \mid \alpha \in P\}; \Vdash = \Vdash' \cup \{(\rho, p) \mid \mathbf{T}p \in S\}.$$

Since $\langle P, \leq', \rho' \rangle$ is a linear order, then, by construction, $\langle P, \leq, \rho \rangle$ is a linear order (note that ρ' is the only immediate successor of ρ). The forcing relation is preserved since the formulas of the kind $\mathbf{T}p \in S$ are in $(S \setminus S_{\mathbf{F}_-})_c$ and by hypothesis the minimum ρ' of \underline{K}' realizes $(S \setminus S_{\mathbf{F}_-})_c$. Since the world ρ' of \underline{K}' realizes $\mathcal{R}_i(S_{\mathbf{F}_-})$ it follows that the world ρ of \underline{K} realizes $S_{\mathbf{F}_-}$. To prove that ρ realizes S the main task is to prove that $\hat{\mathbf{T}}$ and $\tilde{\mathbf{T}}$ -formulas are realized. If a formula of the kind $\hat{\mathbf{T}}(B_i \rightarrow C) \in S$, then $\hat{\mathbf{T}}(B_i \rightarrow C) \in (S \setminus S_{\mathbf{F}_-})_c$. By induction hypothesis $\rho' \triangleright \hat{\mathbf{T}}(B_i \rightarrow C)$, thus $\rho' \Vdash A \rightarrow B_i$ and $\rho' \not\Vdash B_i$, and this implies $\rho \triangleright \hat{\mathbf{T}}(B_i \rightarrow C)$. If a formula of the kind $\hat{\mathbf{T}}(B_j \rightarrow C) \in S$, with $i \neq j$, then $\hat{\mathbf{T}}(B_j \rightarrow C) \in (S \setminus S_{\mathbf{F}_-})_c$. By induction hypothesis $\rho' \triangleright \hat{\mathbf{T}}(B_j \rightarrow C)$,

thus $\rho' \Vdash B_j \rightarrow C$ and $\rho' \not\Vdash B_j$ hold. By the semantical meaning of $\widehat{\mathbf{T}}$ it follows $\rho \Vdash \widehat{\mathbf{T}}(B_j \rightarrow C)$. If $\widetilde{\mathbf{T}}(A \rightarrow B) \in S$, then $\mathbf{T}(A \rightarrow B) \in (S \setminus S_{\mathbf{F}_\rightarrow})_c$ with A atomic and $\mathbf{T}A \notin S$. By construction of \underline{K} , $\rho \not\Vdash A$. Since by induction hypothesis $\rho' \Vdash A \rightarrow B$ we have $\rho \Vdash A \rightarrow B$ and by the meaning of $\widetilde{\mathbf{T}}$ we conclude $\rho \triangleright \widetilde{\mathbf{T}}(A \rightarrow B)$ holds. \square

Theorem 2 (Completeness). *If A is valid in every model, then $\text{DUM}(\{\mathbf{F}A\})$ returns a proof.*

5 The Implementation and the Performances

The ideas presented in this paper have been integrated in the implementation for EPDL of [6], the result is a new prototype prover for Dummett logic called *Dummett Logic Solver for Implications* (DLSI)¹. Cause lack of space, in previous sections the focus has been given to the main idea. There are simple improvements that can be applied to the presentation. Note that in the leftmost conclusion of the rule $\mathbf{T} \rightarrow \rightarrow$ the subformula B occurs twice. This is a source of inefficiency since there can be deduction of exponential depth. Using the well-known indexing technique consisting in replacing a formula with new propositional variable (adopted also in [5, 6, 12]) the result is a calculus whose deductions have depth linearly bounded in the size of the formula to be proved. In [6] a sequence of optimizations is described. Among them a new version of the multiple premise rule of [1] is provided. To simplify the presentation the multiple premise rule of [1] is adopted. DLSI and EPDL differ for the logical rules, neither new optimizations nor code optimization has been performed. To compare EPDL and DLSI some experiments have been carried out. The formulas of ILTP library of paper [13] have been considered. The goal of the calculus is to treat efficiently the case of B non-atomic when formulas of the kind $\mathbf{T}((A \rightarrow B) \rightarrow C)$ occur in the proofs. This never happen with the formulas of ILTP library. Thus the substitution consisting in replacing every propositional variable p_i with $q_i \rightarrow (r_i \rightarrow t_i)$ has been applied to every formula of the ILTP library. Experiments have been performed on the formulas resulting by applying this substitution. Figure 3 shows those family formulas on which the performances of EPDL and DLSI differ². The results show that DLSI outperforms EPDL. Moreover on every family, the timings of DLSI increase of a lower factor than EPDL. On the missing family formulas the timings of EPDL and DLSI are comparable. Finally, Figure 4 gives an account of the comparison between EPDL and DLSI on 10000 randomly generated formulas. It is reported the number of formulas solved respectively within 1, 10, 100, 600 and requiring more than 600 seconds and in parenthesis the seconds required to decide all the formulas (i.e. EPDL solves 9823 formulas within 1 second and the time to decide these 9823 formulas is altogether 204 seconds). Experiments show that the ideas on which the calculus presented in this paper relies improve the known proofs strategies.

¹ Downloadable from <http://www.dimequant.unimib.it/~guidofiorino/dlsi.jsp>

² Timings in seconds, experiments performed on Intel(R) Xeon(TM) 3.00GHz

ILTP Formula	EPDL	DLSI	ILTP Formula	EPDL	DLSI	ILTP Formula	EPDL	DLSI
SYJ201.1	50.67	0.10	SYJ205.2	13689	17.30	SYJ210.5	1627	0.03
SYJ201.2	51852	3.05	SYJ207.1	0.01	0.01	SYJ210.6	119803	0.05
SYJ203.7	26.15	0.17	SYJ207.2	10724	1.18	SYJ211.1	1.67	0.02
SYJ203.8	69.97	0.20	SYJ209.7	18.05	0.18	SYJ211.2	523	0.03
SYJ205.1	123	0.01	SYJ209.8	47.85	0.25			

Fig. 3. Timings on ILTP formulas modified with substitution $X \rightarrow (Y \rightarrow Z)$.

Prover	0-1secs.	1-10secs.	10-100secs.	100-600secs.	>600
EPDL	9823(204s.)	134(491s.)	35(1123s.)	5(1424s.)	3(11007s.)
DLSI	9843(216s.)	116(387s.)	35(943s.)	4(1004s.)	2(6561s.)

Fig. 4. Timings on randomly generated formulas.

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. Avron. Simple consequence relations. *Journal of Information and Computation*, 92:276–294, 1991.
3. A. Avron and B. Konikowska. Decomposition proof systems for Gödel-Dummett logics. *Studia Logica*, 69(2):197–219, 2001.
4. M. Dummett. A propositional calculus with a denumerable matrix. *Journal of Symbolic Logic*, 24:96–107, 1959.
5. G. Fiorino. An $O(n \log n)$ -SPACE decision procedure for the propositional Dummett Logic. *Journal of Automated Reasoning*, 27(3):297–311, 2001.
6. G. Fiorino. Fast decision procedure for propositional Dummett logic based on a multiple premise tableau calculus. T.R.164, Dip. di Metodi Quantitativi per le Scienze Economiche ed Aziendali, Università degli Studi di Milano-Bicocca, 2008.
7. M.C. Fitting. *Intuitionistic Logic, Model Theory and Forcing*. North-Holland, 1969.
8. K. Gödel. On the intuitionistic propositional calculus. In S. Feferman et al, editor, *Collected Works*, volume 1. Oxford University Press, 1986.
9. P. Hajek. *Metamathematics of Fuzzy Logic*. Kluwer, 1998.
10. D. Larchey-Wendling. Combining proof-search and counter-model construction for deciding Gödel-Dummett logic. In A. Voronkov, editor, *CADE*, vol. 2392 of *LNCS*, pag. 94–110. Springer, 2002.
11. D. Larchey-Wendling. Counter-model search in Gödel-Dummett logics. In D.A. Basin and M. Rusinowitch, editors, *IJCAR*, vol. 3097 of *LNCS*, pag. 274–288. Springer, 2004.
12. D. Larchey-Wendling. Graph-based decision for Gödel-Dummett logics. *Journal of Automated Reasoning*, 38(1-3):201–225, 2007.
13. T. Raths, J. Otten, and C. Kreitz. The ILTP problem library for intuitionistic logic. *Journal of Automated Reasoning*, 38(1-3):261–271, 2007.
14. R.M. Smullyan. *First-Order Logic*. Springer, Berlin, 1968.