A *DPLL* Procedure for the Propositional Product Logic

Dušan Guller

Department of Applied Informatics, Comenius University, Mlynská dolina, 842 48 Bratislava, Slovakia

Keywords: Product Logic, *DPLL* Procedure, Many-valued Logics, Automated Deduction.

Abstract: In the paper, we investigate the deduction problem of a formula from a finite theory in the propositional Product logic from a perspective of automated deduction. Our approach is based on translation of a formula to an equivalent satisfiable finite order clausal theory, consisting of order clauses. An order clause is a finite set of order literals of the form $\varepsilon_1 \diamond \varepsilon_2$ where ε_i is either a conjunction of propositional atoms or the propositional constant *0* (false) or *1* (true), and $\diamond$ is a connective either $\eqcirc$ or $\prec$. $\eqcirc$ and $\prec$ are interpreted by the equality and standard strict linear order on $[0,1]$, respectively. A variant of the *DPLL* procedure, operating over order clausal theories, is proposed. The *DPLL* procedure is proved to be refutation sound and complete for finite order clausal theories.

1 INTRODUCTION

A considerable effort has been made in development of *SAT* solvers for the problem of Boolean satisfiability, especially in the last decade. *SAT* solvers may exploit either complete solution methods (called complete or systematic *SAT* solvers) or incomplete or hybrid ones. Complete *SAT* solvers are mostly based on the Davis-Putnam-Logemann-Loveland procedure (*DPLL*) (Davis and Putnam, 1960; Davis et al., 1962) improved by various features. One of the latest overviews of development of *SAT* solvers may be found in (Biere et al., 2009). Research in many-valued logics mainly concerns finitely-valued ones. Thank to finiteness of truth value sets of these logics, almost straightforward extensions of results achieved in classical logic are feasible. The *DPLL* procedure has been firstly generalised for regular clauses over a linearly ordered truth value set (Hähnle, 1996). In (Manyà et al., 1998), it is described an implementation of this regular *DPLL* procedure with the extended two-sided Jeroslow-Wang literal selection rule defined in (Hähnle, 1996). A signed *DPLL* procedure over a finite truth value set is introduced in (Beckert et al., 2000). It is based on a branching rule forming branches for every truth value. So, the branching factor equals the cardinality of the truth value set. The branching factor can be decreased by a quotient of the truth value set wrt. a suitable equivalence. A slight modification of that equivalence enables a generalisation to an infinite truth value set as well (Guller, 2009). Another signed variant of the *DPLL* procedure for a countable clausal theory over an arbitrary truth value set is proposed in (Guller, 2009). In some sense, the *DPLL* procedure may be viewed like ”anti-resolution”. Thus, its branching rule, with finite branching factor, may be considered as if a ”signed anti-hyperresolution rule”. The procedure is refutation complete if the finitary disjunction condition for the set of signs occurring in the input countable clausal theory is satisfied. Infinitely-valued logics have not yet been explored so widely as finitely-valued ones. It is not known any general approach as signed logic one in the finitely-valued case. A solution of the *SAT* and *VAL* problems strongly varies on a chosen infinitely-valued logic. The same holds for translation of a formula to clause form, the existence of which is not guaranteed in general. Results in this area have been achieved in several ways, since infinite truth value sets form distinct algebraic structures. One approach may be based on reduction from the infinitely-valued case to the finitely-valued one, as it has been done e.g. for the *VAL* problem in the propositional infinitely-valued Łukasiewicz logic in (Mundici, 1987; Aguzzoli and Ciabattoni, 2000). Another approach exploits reduction of the *SAT* problem to mixed integer programming (*MIP*) (Hähnle, 1994a; Hähnle, 1997). In (Guller, 2010), we have devised a variant of the *DPLL* procedure with clause form translation for finite theories in the propositional Gödel logic. The results have been generalised to the

Partially supported by VEGA Grant 1/0979/12.

Guller D..
A DPLL Procedure for the Propositional Product Logic.
DOI: 10.5220/0004557402130224
In *Proceedings of the 5th International Joint Conference on Computational Intelligence* (FCTA-2013), pages 213-224
ISBN: 978-989-8565-77-8

countable case in (Guller, 2012).

Product logic (Hájek et al., 1996; Metcalfe et al., 2004; Savický et al., 2006) is one of the fundamental fuzzy logics, based on the product t-norm. It has been discovered much later than Gödel and Łukasiewicz logics, known before the beginning of research on fuzzy theory. In the paper, we investigate the deduction problem of a formula from a finite theory in the propositional Product logic from a perspective of automated deduction. Our approach is based on translation of a formula to an equivalent satisfiable finite order clausal theory, consisting of order clauses, Lemma 3.1, Theorem 3.2, Section 3. An order clause is a finite set of order literals of the form $\varepsilon_1 \diamond \varepsilon_2$ where ε_i is either a conjunction of propositional atoms or the propositional constant *0* (false) or *1* (true), and $\diamond$ is a connective either $\eqcirc$ or $\prec$. $\eqcirc$ and $\prec$ are interpreted by the equality and standard strict order on $[0,1]$, respectively. The trichotomy over order literals: either $\varepsilon_1 \prec \varepsilon_2$ or $\varepsilon_1 \eqcirc \varepsilon_2$ or $\varepsilon_2 \prec \varepsilon_1$, naturally invokes proposing a variant of the *DPLL* procedure with a trichotomy branching rule as an algorithm for deciding the satisfiability of a finite order clausal theory. The *DPLL* procedure with its basic rules is proved to be refutation sound and complete in the finite case, Theorem 4.2, Section 4. The set of basic rules may be augmented by some admissible ones, which are suitable for practical computing and considerably shorten *DPLL* trees. For solving the deduction problem, we exploit the fact that a formula ϕ is a propositional consequence of a finite theory T in Product logic if and only if their translation to a finite order clausal theory S_T^ϕ is unsatisfiable, and the *DPLL* procedure produces a closed *DPLL* tree with the root S_T^ϕ in this case, Corollary 4.3, Section 4.

The paper is organised as follows. Section 2 gives the basic notions, notation, and useful properties concerning the propositional Product logic. Section 3 deals with clause form translation. In Section 4, we propose a variant of the *DPLL* procedure with a trichotomy branching rule and prove its refutational soundness, completeness. Section 5 brings conclusions.

2 PROPOSITIONAL PRODUCT LOGIC

Throughout the paper, we shall use the common notions of propositional many-valued logics. The set of propositional atoms of Product logic will be denoted as *PropAtom*. By *PropForm* we designate the set of all propositional formulae of Product logic built up from *PropAtom* using the propositional constants *0*, false, *1*, true, and the connectives: $\neg$, negation, $\wedge$, conjunction, $\vee$, disjunction, &, strong conjunction, $\rightarrow$, implication. In addition, we introduce new binary connectives $\eqcirc$, equality, and $\prec$, strict order. By *OrdPropForm* we designate the set of all so-called order propositional formulae of Product logic built up from *PropAtom* using the propositional constants *0*, *1*, and the connectives: $\neg$, $\wedge$, $\vee$, &, $\rightarrow$, $\eqcirc$, $\prec$.[1] In the paper, we shall assume that *PropAtom* is a countable set. Let ε_i, $1 \le i \le n$, be either an expression or a set of expressions or a set of sets of expressions, in general. By $atoms(\varepsilon_1,\dots,\varepsilon_m) \subseteq PropAtom$ we denote the set of all propositional atoms of Product logic occurring in $\varepsilon_1,\dots,\varepsilon_m$.

Let X, Y, Z be sets, $Z \subseteq X$; $f : X \longrightarrow Y$ be a mapping. By $\|X\|$ we denote the set-theoretic cardinality of X. X being a finite subset of Y is denoted as $X \subseteq_{\mathcal{F}} Y$. We designate $f[Z] = \{f(z)\,|\,z \in Z\}$; $f[Z]$ is the image of Z under f; and $f|_Z = \{(z, f(z))\,|\,z \in Z\}$; $f|_Z$ is the restriction of f onto Z. Let $\gamma \le \omega$. A sequence δ of X is a bijection $\delta : \gamma \longrightarrow X$. X is countable if and only if there exists a sequence of X. $\mathbb{N}$ | $\mathbb{R}$ designates the set of natural | real numbers and $\le$, $<$ the standard, standard strict order on $\mathbb{N}$ | $\mathbb{R}$, respectively. We denote $\mathbb{R}_0^+ = \{c\,|\,0 \le c \in \mathbb{R}\}$, $\mathbb{R}^+ = \{c\,|\,0 < c \in \mathbb{R}\}$, $[0,1] = \{c\,|\,0 \le c \le 1, c \in \mathbb{R}\}$; $[0,1]$ is the unit interval. Let $c \in \mathbb{R}^+$. $\log c$ denotes the binary logarithm of c. Let $f, g : \mathbb{N} \longrightarrow \mathbb{R}_0^+$. f is of the order of g, in symbols $f \in O(g)$, iff there exist $n_0 \in \mathbb{N}$ and $c^* \in \mathbb{R}_0^+$ such that for all $n \ge n_0$, $f(n) \le c^* \cdot g(n)$. Let $\phi \in OrdPropForm$ and $T \subseteq_{\mathcal{F}} OrdPropForm$. The size of ϕ, in symbols $|\phi| > 0$, is defined as the number of nodes of its standard tree representation. We define the size of T as $|T| = \sum_{\phi \in T} |\phi|$.

Product logic is interpreted by the standard Π-algebra augmented by binary operators $\eqcirc$ and $\prec$ for $\eqcirc$ and $\prec$, respectively.

$$\Pi = ([0,1], \le, \vee, \wedge, \cdot, \Rightarrow, \overline{}, \eqcirc, \prec, 0, 1)$$

where $\vee$ | $\wedge$ denotes the supremum | infimum operator on $[0,1]$;

$$a \Rightarrow b = \begin{cases} 1 & \textit{if } a \le b, \\ \frac{b}{a} & \textit{else}; \end{cases} \qquad \overline{a} = \begin{cases} 1 & \textit{if } a = 0, \\ 0 & \textit{else}; \end{cases}$$

$$a \eqcirc b = \begin{cases} 1 & \textit{if } a = b, \\ 0 & \textit{else}; \end{cases} \qquad a \prec b = \begin{cases} 1 & \textit{if } a < b, \\ 0 & \textit{else}. \end{cases}$$

We recall that Π is a complete linearly ordered lattice algebra; $\vee$ | $\wedge$ is commutative, associative, idempotent, monotone; 0 | 1 is its neutral element; $\cdot$ is com-

[1] We assume a decreasing connective precedence: $\neg$, &, $\wedge$, $\rightarrow$, $\eqcirc$, $\prec$, $\vee$.

mutative, associative, monotone; 1 is its neutral element; the residuum operator $\Rightarrow$ of $\cdot$ satisfies the condition of residuation:

$$\text{for all } a,b,c \in \Pi, a \cdot b \leq c \Longleftrightarrow a \leq b \Rightarrow c; \quad (1)$$

Product (Gödel) negation $\overline{}$ satisfies the condition:

$$\text{for all } a \in \Pi, \overline{a} = a \Rightarrow 0; \quad (2)$$

the following properties, which will be exploited later, hold:[2]

for all $a,b,c \in \Pi$,

$$a \vee b \wedge c = (a \vee b) \wedge (a \vee c), \quad \text{(distributivity of } \vee \text{ over } \wedge\text{)} \ (3)$$

$$a \wedge (b \vee c) = a \wedge b \vee a \wedge c, \quad \text{(distributivity of } \wedge \text{ over } \vee\text{)} \ (4)$$

$$a \cdot (b \vee c) = a \cdot b \vee a \cdot c, \quad \text{(distributivity of } \cdot \text{ over } \vee\text{)} \ (5)$$

$$a \Rightarrow (b \vee c) = a \Rightarrow b \vee a \Rightarrow c, \quad (6)$$

$$a \Rightarrow b \wedge c = (a \Rightarrow b) \wedge (a \Rightarrow c), \quad (7)$$

$$(a \vee b) \Rightarrow c = (a \Rightarrow c) \wedge (b \Rightarrow c), \quad (8)$$

$$a \wedge b \Rightarrow c = a \Rightarrow c \vee b \Rightarrow c, \quad (9)$$

$$a \Rightarrow (b \Rightarrow c) = a \cdot b \Rightarrow c, \quad (10)$$

$$((a \Rightarrow b) \Rightarrow b) \Rightarrow b = a \Rightarrow b. \quad (11)$$

A propositional theory is a set of propositional formulae of Product logic. An order propositional theory is a set of order propositional formulae of Product logic. A valuation $\mathcal{V}$ is a mapping $\mathcal{V} : \mathit{PropAtom} \longrightarrow [0,1]$. A partial valuation $\mathcal{V}$ with the domain $\mathit{dom}(\mathcal{V}) \subseteq \mathit{PropAtom}$, is a mapping $\mathcal{V} : \mathit{dom}(\mathcal{V}) \longrightarrow [0,1]$. Let $\mathcal{V}$ be a (partial) valuation; $\phi, \phi' \in \mathit{OrdPropForm}$, $T \subseteq \mathit{OrdPropForm}$. Let $\mathit{atoms}(\phi), \mathit{atoms}(T) \subseteq \mathit{dom}(\mathcal{V})$ in case of $\mathcal{V}$ being a partial valuation. The truth value of ϕ in $\mathcal{V}$, in symbols $\|\phi\|^{\mathcal{V}}$, is defined by the standard way; the propositional constants *0*, *1* are interpreted by 0, 1, respectively, and the connectives by the respective operators on Π. $\mathcal{V}$ is a (partial) propositional model of ϕ, in symbols $\mathcal{V} \models \phi$, iff $\|\phi\|^{\mathcal{V}} = 1$. $\mathcal{V}$ is a (partial) propositional model of T, in symbols $\mathcal{V} \models T$, iff, for all $\phi \in T$, $\mathcal{V} \models \phi$. ϕ is a tautology iff, for every valuation $\mathcal{V}$, $\mathcal{V} \models \phi$. ϕ is equivalent to ϕ', in symbols $\phi \equiv \phi'$, iff, for every valuation $\mathcal{V}$, $\|\phi\|^{\mathcal{V}} = \|\phi'\|^{\mathcal{V}}$.

[2] We assume a decreasing operator precedence: $\overline{}$, $\cdot$, $\wedge$, $\Rightarrow$, $\eqdot$, $\prec$, $\vee$.

3 TRANSLATION TO ORDER CLAUSAL FORM

We now describe some translation of a formula to a finite order clausal theory. To have the output theory of polynomial size, our translation exploits interpolation using new atoms. The output theory will be of linearithmic size at the cost of being only equivalent satisfiable to the input formula. A similar approach exploiting the renaming subformulae technique can be found in (Plaisted and Greenbaum, 1986; de la Tour, 1992; Hähnle, 1994b; Nonnengart et al., 1998; Sheridan, 2004; Guller, 2010). At first, we introduce notions of a to the power of n and of conjunction of propositional atoms. Let $a \in \mathit{PropAtom}$ and $n > 0$. a to the power of n is the pair (a,n), written as a^n. The power a^1 is denoted as a; if it does not cause the ambiguity with the denotation of the single propositional atom a in given context. We define the size of a^n as $|a^n| = n > 0$. A conjunction Cn of propositional atoms is a non-empty finite set of powers such that for all $a^m, b^n \in \mathit{Cn}$, $a \neq b$. A conjunction $\{a_0^{m_0}, \dots, a_n^{m_n}\}$ of propositional atoms is written in the form $a_0^{m_0} \,\&\, \cdots \,\&\, a_n^{m_n}$. A conjunction $\{p\}$ of propositional atoms is called a unit conjunction of propositional atoms and denoted as p; if it does not cause the ambiguity with the denotation of the single power p in given context. The set of all conjunctions of propositional atoms is designated as *PropConj*. Let $\mathcal{V}$ be a (partial) valuation; p be a power, $\mathit{Cn} \in \mathit{PropConj}$, $\mathit{Cn}_1, \mathit{Cn}_2 \in \mathit{PropConj} \cup \{\emptyset\}$. Let $\mathit{atoms}(\mathit{Cn}) \subseteq \mathit{dom}(\mathcal{V})$ in case of $\mathcal{V}$ being a partial valuation. The truth value of $\mathit{Cn} = a_0^{m_0} \,\&\, \cdots \,\&\, a_n^{m_n}$ in $\mathcal{V}$ is defined by

$$\|\mathit{Cn}\|^{\mathcal{V}} = \underbrace{\|a_0\|^{\mathcal{V}} \cdot \dots \cdot \|a_0\|^{\mathcal{V}}}_{m_0} \cdot \dots \cdot \underbrace{\|a_n\|^{\mathcal{V}} \cdot \dots \cdot \|a_n\|^{\mathcal{V}}}_{m_n}.$$

We define the size of Cn as $|\mathit{Cn}| = \sum_{p \in \mathit{Cn}} |p| > 0$. By $p \,\&\, \mathit{Cn}$ we denote $\{p\} \cup \mathit{Cn}$ where $p \notin \mathit{Cn}$. Cn_1 is a subconjunction of Cn_2, in symbols $\mathit{Cn}_1 \sqsubseteq \mathit{Cn}_2$, iff, for all $a^m \in \mathit{Cn}_1$, there exists $a^n \in \mathit{Cn}_2$ and $m \leq n$. We define $\mathit{Cn}_1 \sqcap \mathit{Cn}_2 = \{a^{\min(m,n)} \mid a^m \in \mathit{Cn}_1, a^n \in \mathit{Cn}_2\} \in \mathit{PropConj} \cup \{\emptyset\}$. Cn_1 and Cn_2 are disjoint iff $\mathit{Cn}_1 \sqcap \mathit{Cn}_2 = \emptyset$. We finally introduce order clauses in Product logic. l is an order literal of Product logic iff $l = \varepsilon_1 \diamond \varepsilon_2$ where either $\varepsilon_1 \in \mathit{PropAtom} \cup \{\mathit{0}, \mathit{1}\}$, $\varepsilon_2 \in \{\mathit{0}, \mathit{1}\}$, or $\varepsilon_1 \in \{\mathit{0}, \mathit{1}\}$, $\varepsilon_2 \in \mathit{PropAtom} \cup \{\mathit{0}, \mathit{1}\}$, or $\varepsilon_i \in \mathit{PropConj}$, $\varepsilon_1 \sqcap \varepsilon_2 = \emptyset$, $\diamond \in \{\eqdot, \prec\}$. The set of all order literals of Product logic is designated as *OrdLit*. Let $l = \varepsilon_1 \diamond \varepsilon_2 \in \mathit{OrdLit}$. We define the size of l as $|l| = 1 + |\varepsilon_1| + |\varepsilon_2| > 0$. An order clause of Product logic is a finite set of order literals of Product logic; since $=$ is commutative, we identify the order literals $\varepsilon_1 \eqdot \varepsilon_2$ and $\varepsilon_2 \eqdot \varepsilon_1$. An order clause

$\{l_1,\dots,l_n\}$ is written in the form $l_1 \vee \cdots \vee l_n$. The order clause $\emptyset$ is called the empty order clause and denoted as $\square$. An order clause $\{l\}$ is called a unit order clause and denoted as l; if it does not cause the ambiguity with the denotation of the single order literal l in given context. We designate the set of all order clauses of Product logic as *OrdCl*. Let $l, l_0, \dots, l_n \in OrdLit$ and $C, C' \in OrdCl_{\mathcal{L}}$. We define the size of C as $|C| = \sum_{l \in C} |l|$. By $l \vee C$ we denote $\{l\} \cup C$ where $l \notin C$. Analogously, by $l_0 \vee \cdots \vee l_n \vee C$ we denote $\{l_0\} \cup \cdots \cup \{l_n\} \cup C$ where, for all $i, i' \leq n$, $i \neq i'$, $l_i \notin C$ and $l_i \neq l_{i'}$. By $C \vee C'$ we denote $C \cup C'$. C is a subclause of C', in symbols $C \sqsubseteq C'$, iff $C \subseteq C'$. An order clausal theory is a set of order clauses. A unit order clausal theory is a set of unit order clauses.

Let $\phi, \phi' \in PropOrdForm$, $T, T' \subseteq PropOrdForm$, $S, S' \subseteq OrdCl$; $\mathcal{V}$ be a (partial) valuation. Let $atoms(l), atoms(C), atoms(S) \subseteq dom(\mathcal{V})$ in case of $\mathcal{V}$ being a partial valuation. Note that $\mathcal{V} \models l$ if and only if either $l = \varepsilon_1 \eqcirc \varepsilon_2$, $\|\varepsilon_1 \eqcirc \varepsilon_2\|^{\mathcal{V}} = 1$, $\|\varepsilon_1\|^{\mathcal{V}} = \|\varepsilon_2\|^{\mathcal{V}}$; or $l = \varepsilon_1 \prec \varepsilon_2$, $\|\varepsilon_1 \prec \varepsilon_2\|^{\mathcal{V}} = 1$, $\|\varepsilon_1\|^{\mathcal{V}} < \|\varepsilon_2\|^{\mathcal{V}}$. $\mathcal{V}$ is a (partial) propositional model of C, in symbols $\mathcal{V} \models C$, iff there exists $l^* \in C$ such that $\mathcal{V} \models l^*$. $\mathcal{V}$ is a (partial) propositional model of S, in symbols $\mathcal{V} \models S$, iff, for all $C \in S$, $\mathcal{V} \models C$. $\phi' \mid T' \mid C' \mid S'$ is a propositional consequence of $\phi \mid T \mid C \mid S$, in symbols $\phi \mid T \mid C \mid S \models_P \phi' \mid T' \mid C' \mid S'$, iff, for every propositional model $\mathcal{V}$ of $\phi \mid T \mid C \mid S$, $\mathcal{V} \models \phi' \mid T' \mid C' \mid S'$. $\phi \mid T \mid C \mid S$ is satisfiable iff there exists a propositional model of $\phi \mid T \mid C \mid S$. Note that both $\square$ and $\square \in S$ are unsatisfiable. $\phi \mid T \mid C \mid S$ is equisatisfiable to $\phi' \mid T' \mid C' \mid S'$ iff $\phi \mid T \mid C \mid S$ is satisfiable if and only if $\phi' \mid T' \mid C' \mid S'$ is satisfiable. Let $S \subseteq_{\mathcal{F}} OrdCl$. We define the size of S as $|S| = \sum_{C \in S} |C|$. Let $l \in OrdLit$. l is a simplified order literal of Product logic iff if $l = \varepsilon_1 \diamond \varepsilon_2$, $\varepsilon_i \in PropConj$, then either $\varepsilon_1 = a$, $\varepsilon_2 = b$, or $\varepsilon_1 = a$, $\varepsilon_2 = b \,\&\, c$, or $\varepsilon_1 = a \,\&\, b$, $\varepsilon_2 = c$. The set of all simplified order literals of Product logic is designated as *SimOrdLit*. We denote $SimOrdCl = \{C \mid C \in OrdCl, C \subseteq SimOrdLit\}$. Let $\mathbb{I} = \mathbb{N} \times \mathbb{N}$; $\mathbb{I}$ is an infinite countable set of indices. Let $\tilde{\mathbb{A}} = \{\tilde{a}_{\mathbb{i}} \mid \mathbb{i} \in \mathbb{I}\} \subseteq PropAtom$; $\tilde{\mathbb{A}}$ is an infinite countable set of new propositional atoms. Let $A \subseteq \tilde{\mathbb{A}}$. We denote $\mathscr{E}_A = \{\varepsilon \mid \varepsilon \in \mathscr{E}, atoms(\varepsilon) \cap \tilde{\mathbb{A}} \subseteq A\}$, $\mathscr{E} = PropForm \mid \mathscr{E} = PropConj \mid \mathscr{E} = OrdLit \mid \mathscr{E} = OrdCl \mid \mathscr{E} = SimOrdLit \mid \mathscr{E} = SimOrdCl$. From a computational point of view, the worst case time and space complexity will be estimated using the logarithmic cost measurement. Let $\mathcal{A}$ be an algorithm. $\#O_{\mathcal{A}}$ denotes the number of all elementary operations executed by $\mathcal{A}$. The translation to order clausal form is based on the following lemma.

Lemma 3.1. *Let $\phi \in PropForm_\emptyset$, $T \subseteq_{\mathcal{F}} PropForm_\emptyset$; $F \subseteq \mathbb{I}$ such that there exists n_0 and $F \cap \{(i,j) \mid i \geq n_0\} = \emptyset$; $n_\phi \geq n_0$.*

(i) *There exist either $J_\phi = \emptyset$ or $J_\phi = \{(n_\phi, j) \mid j \leq n_{J_\phi}\}$, $J_\phi \subseteq \{(i,j) \mid i \geq n_0\}$, $J_\phi \cap F = \emptyset$, and $S_\phi \subseteq_{\mathcal{F}} SimOrdCl_{\{\tilde{a}_{\mathbb{j}} \mid \mathbb{j} \in J_\phi\}}$ such that*

  (a) $\|J_\phi\| \leq 2 \cdot |\phi|$;

  (b) *either $J_\phi = \emptyset$, $S_\phi = \{\square\}$ or $J_\phi = S_\phi = \emptyset$ or $J_\phi \neq \emptyset$, $\square \notin S_\phi \neq \emptyset$;*

  (c) *there exists a partial valuation $\mathcal{V}$, $dom(\mathcal{V}) = atoms(\phi)$, and $\mathcal{V} \models \phi$ if and only if there exists a partial valuation $\mathcal{V}'$, $dom(\mathcal{V}') = atoms(\phi) \cup \{\tilde{a}_{\mathbb{j}} \mid \mathbb{j} \in J_\phi\}$, and $\mathcal{V}' \models S_\phi$, satisfying $\mathcal{V} = \mathcal{V}'|_{atoms(\phi)}$;*

  (d) *$|S_\phi| \in O(|\phi|)$; the number of all elementary operations of the translation of ϕ to S_ϕ, is in $O(|\phi|)$; the time and space complexity of the translation of ϕ to S_ϕ, is in $O(|\phi| \cdot \log |\phi|)$;*

  (e) *if $S_\phi \neq \emptyset$ and $S_\phi \neq \{\square\}$, then $J_\phi \neq \emptyset$; for all $C \in S_\phi$, $\emptyset \neq atoms(C) \cap \tilde{\mathbb{A}} \subseteq \{\tilde{a}_{\mathbb{j}} \mid \mathbb{j} \in J_\phi\}$.*

(ii) *There exist $J_T \subseteq_{\mathcal{F}} \{(i,j) \mid i \geq n_0\}$, $J_T \cap F = \emptyset$, and $S_T \subseteq_{\mathcal{F}} SimOrdCl_{\{\tilde{a}_{\mathbb{j}} \mid \mathbb{j} \in J_T\}}$ such that*

  (a) $\|J_T\| \leq 2 \cdot |T|$;

  (b) *either $J_T = \emptyset$, $S_T = \{\square\}$ or $J_T = S_T = \emptyset$ or $J_T \neq \emptyset$, $\square \notin S_T \neq \emptyset$;*

  (c) *there exists a partial valuation $\mathcal{V}$, $dom(\mathcal{V}) = atoms(T)$, and $\mathcal{V} \models T$ if and only if there exists a partial valuation $\mathcal{V}'$, $dom(\mathcal{V}') = atoms(T) \cup \{\tilde{a}_{\mathbb{j}} \mid \mathbb{j} \in J_T\}$, and $\mathcal{V}' \models S_T$, satisfying $\mathcal{V} = \mathcal{V}'|_{atoms(T)}$;*

  (d) *$|S_T| \in O(|T|)$; the number of all elementary operations of the translation of T to S_T, is in $O(|T|)$; the time and space complexity of the translation of T to S_T, is in $O(|T| \cdot \log(1 + |T|))$;*

  (e) *if $S_T \neq \emptyset$ and $S_T \neq \{\square\}$, then $J_T \neq \emptyset$; for all $C \in S_T$, $\emptyset \neq atoms(C) \cap \tilde{\mathbb{A}} \subseteq \{\tilde{a}_{\mathbb{j}} \mid \mathbb{j} \in J_T\}$.*

Proof. Technical using interpolation.

Let $\theta \in PropForm_\emptyset$. There exists $\theta' \in PropForm_\emptyset$ such that (12)

(a) $\theta' \equiv \theta$;

(b) $|\theta'| \leq 2 \cdot |\theta|$; θ' can be built up via a postorder traversal of θ with $\#O \in O(|\theta|)$, the time and space complexity in $O(|\theta| \cdot \log |\theta|)$;

(c) θ' does not contain $\neg$;

(d) either $\theta' = 0$, or 0 is a subformula of θ' if and only if 0 is a subformula of a subformula of θ' of the form $\vartheta \rightarrow 0$, $\vartheta \neq 0$;

(e) either $\theta' = 1$ or 1 is not a subformula of θ'.

The proof is by induction on the structure of θ.

Let $\theta \in \mathit{PropForm}_{\emptyset} - \{0, 1\}$; (12c–e) hold for θ; $G \subseteq \mathbb{I}$ such that there exists n_1 and $G \cap \{(i, j) \,|\, i \geq n_1\} = \emptyset$; $n_\theta \geq n_1$; $\mathbb{i} = (n_\theta, j_{\mathbb{i}}) \in \{(i, j) \,|\, i \geq n_1\}$, $\tilde{a}_{\mathbb{i}} \in \tilde{\mathbb{A}}$, $\mathbb{i} \notin G$. There exist $J = \{(n_\theta, j) \,|\, j_{\mathbb{i}} + 1 \leq j \leq n_J\} \subseteq \{(i, j) \,|\, i \geq n_1\}$, $j_{\mathbb{i}} \leq n_J$, $J \cap (G \cup \{\mathbb{i}\}) = \emptyset$, and $S^s \subseteq_{\mathcal{F}} \mathit{SimOrdCl}_{\{\tilde{a}_{\mathbb{i}}\} \cup \{\tilde{a}_{\mathbb{j}} \,|\, \mathbb{j} \in J\}}$, $s = +, -$, such that for both s, (13)

(a) $\|J\| \leq |\theta| - 1$;

(b) there exists a partial valuation $\mathcal{V}$, $\mathit{dom}(\mathcal{V}) = \mathit{atoms}(\theta) \cup \{\tilde{a}_{\mathbb{i}}\}$, and $\mathcal{V} \models \tilde{a}_{\mathbb{i}} \rightarrow \theta \in \mathit{PropForm}_{\{\tilde{a}_{\mathbb{i}}\}}$ if and only if there exists a partial valuation $\mathcal{V}'$, $\mathit{dom}(\mathcal{V}') = \mathit{atoms}(\theta) \cup \{\tilde{a}_{\mathbb{i}}\} \cup \{\tilde{a}_{\mathbb{j}} \,|\, \mathbb{j} \in J\}$, and $\mathcal{V}' \models S^+$, satisfying $\mathcal{V} = \mathcal{V}'|_{\mathit{atoms}(\theta) \cup \{\tilde{a}_{\mathbb{i}}\}}$;

(c) there exists a partial valuation $\mathcal{V}$, $\mathit{dom}(\mathcal{V}) = \mathit{atoms}(\theta) \cup \{\tilde{a}_{\mathbb{i}}\}$, and $\mathcal{V} \models \theta \rightarrow \tilde{a}_{\mathbb{i}} \in \mathit{PropForm}_{\{\tilde{a}_{\mathbb{i}}\}}$ if and only if there exists a partial valuation $\mathcal{V}'$, $\mathit{dom}(\mathcal{V}') = \mathit{atoms}(\theta) \cup \{\tilde{a}_{\mathbb{i}}\} \cup \{\tilde{a}_{\mathbb{j}} \,|\, \mathbb{j} \in J\}$, and $\mathcal{V}' \models S^-$, satisfying $\mathcal{V} = \mathcal{V}'|_{\mathit{atoms}(\theta) \cup \{\tilde{a}_{\mathbb{i}}\}}$;

(d) $|S^s| \leq 20 \cdot |\theta|$, S^s can be built up from θ via a preorder traversal of θ with $\#O \in O(|\theta|)$;

(e) for all $C \in S^s$, $\emptyset \neq \mathit{atoms}(C) \cap \tilde{\mathbb{A}} \subseteq \{\tilde{a}_{\mathbb{i}}\} \cup \{\tilde{a}_{\mathbb{j}} \,|\, \mathbb{j} \in J\}$; $\tilde{a}_{\mathbb{i}} \eqcirc 1, \tilde{a}_{\mathbb{i}} \prec 1 \notin S^s$.

The proof is by induction on the structure of θ using the interpolation rules in Table 1.

(i) By (12) for ϕ, there exists $\phi' \in \mathit{PropForm}_{\emptyset}$ such that (12a–e) hold for ϕ'. We then distinguish three cases for ϕ'.

Case 1: $\phi' = 0$. We put $J_\phi = \emptyset \subseteq \{(i, j) \,|\, i \geq n_0\}$, $J_\phi \cap F = \emptyset$, and $S_\phi = \{\square\} \subseteq_{\mathcal{F}} \mathit{SimOrdCl}_{\emptyset}$.

Case 2: $\phi' = 1$. We put $J_\phi = \emptyset \subseteq \{(i, j) \,|\, i \geq n_0\}$, $J_\phi \cap F = \emptyset$, and $S_\phi = \emptyset \subseteq_{\mathcal{F}} \mathit{SimOrdCl}_{\emptyset}$.

Case 3: $\phi' \neq 0, 1$. We have $n_\phi \geq n_0$. We put $\mathbb{i} = (n_\phi, 0) \in \{(i, j) \,|\, i \geq n_0\}$. Then $\tilde{a}_{\mathbb{i}} \in \tilde{\mathbb{A}}$. We get by (13) for ϕ', F, n_0, n_ϕ, $\mathbb{i}$, $\tilde{a}_{\mathbb{i}}$ that there exist $J = \{(n_\phi, j) \,|\, 1 \leq j \leq n_J\} \subseteq \{(i, j) \,|\, i \geq n_0\}$, $J \cap (F \cup \{\mathbb{i}\}) = \emptyset$, $S^+ \subseteq_{\mathcal{F}} \mathit{SimOrdCl}_{\{\tilde{a}_{\mathbb{i}}\} \cup \{\tilde{a}_{\mathbb{j}} \,|\, \mathbb{j} \in J\}}$, and (13a–e) hold for ϕ', $\tilde{a}_{\mathbb{i}}$, J, S^+. We put $n_{J_\phi} = n_J$, $J_\phi = \{\mathbb{i}\} \cup J \subseteq \{(i, j) \,|\, i \geq n_0\}$, $J_\phi \cap F = \emptyset$, and $S_\phi = \{\tilde{a}_{\mathbb{i}} \eqcirc 1\} \cup S^+ \subseteq_{\mathcal{F}} \mathit{SimOrdCl}_{\{\tilde{a}_{\mathbb{j}} \,|\, \mathbb{j} \in J_\phi\}}$.

(ii) straightforwardly follows from (i). □

We conclude this section by the following theorem.

Theorem 3.2. *Let $\phi \in \mathit{PropForm}_{\emptyset}$, $T \subseteq_{\mathcal{F}} \mathit{PropForm}_{\emptyset}$; $F \subseteq \mathbb{I}$ such that there exists n_0 and $F \cap \{(i, j) \,|\, i \geq n_0\} = \emptyset$. There exist $J_T^\phi \subseteq_{\mathcal{F}} \{(i, j) \,|\, i \geq n_0\}$, $J_T^\phi \cap F = \emptyset$, and $S_T^\phi \subseteq_{\mathcal{F}} \mathit{SimOrdCl}_{\{\tilde{a}_{\mathbb{j}} \,|\, \mathbb{j} \in J_T^\phi\}}$ such that*

(i) *$T \models_P \phi$ if and only if S_T^ϕ is unsatisfiable;*

(ii) *$\|J_T^\phi\| \in O(|T| + |\phi|)$; $|S_T^\phi| \in O(|T| + |\phi|)$; the number of all elementary operations of the translation of T and ϕ to S_T^ϕ, is in $O(|T| + |\phi|)$; the time and space complexity of the translation of T and ϕ to S_T^ϕ, is in $O((|T| + |\phi|) \cdot \log(|T| + |\phi|))$.*

Proof. (i) We put $J_{n_0} = \{(n_0, j) \,|\,\} \subseteq \{(i, j) \,|\, i \geq n_0\}$ and $G = F \cup J_{n_0} \subseteq \mathbb{I}$. We get by Lemma 3.1(ii) for T, G, $n_0 + 1$ that there exist $J_T \subseteq_{\mathcal{F}} \{(i, j) \,|\, i \geq n_0 + 1\}$, $J_T \cap G = \emptyset$, $S_T \subseteq_{\mathcal{F}} \mathit{SimOrdCl}_{\{\tilde{a}_{\mathbb{j}} \,|\, \mathbb{j} \in J_T\}}$, and 3.1(ii a–e) hold for T, J_T, S_T. By (12) for ϕ, there exists $\phi' \in \mathit{PropForm}_{\emptyset}$ such that (12a–e) hold for ϕ'. We then distinguish three cases for ϕ'.

Case 1: $\phi' = 0$. We put $J_T^\phi = J_T \subseteq_{\mathcal{F}} \{(i, j) \,|\, i \geq n_0\}$, $J_T^\phi \cap F = \emptyset$, and $S_T^\phi = S_T \subseteq_{\mathcal{F}} \mathit{SimOrdCl}_{\{\tilde{a}_{\mathbb{j}} \,|\, \mathbb{j} \in J_T^\phi\}}$.

Case 2: $\phi' = 1$. We put $J_T^\phi = \emptyset \subseteq_{\mathcal{F}} \{(i, j) \,|\, i \geq n_0\}$, $J_T^\phi \cap F = \emptyset$, and $S_T^\phi = \{\square\} \subseteq_{\mathcal{F}} \mathit{SimOrdCl}_{\emptyset}$.

Case 3: $\phi' \neq 0, 1$. We put $\mathbb{i} = (n_0, 0) \in \{(i, j) \,|\, i \geq n_0\}$. Then $\tilde{a}_{\mathbb{i}} \in \tilde{\mathbb{A}}$. We get by (13) for ϕ', F, n_0, n_0, $\mathbb{i}$, $\tilde{a}_{\mathbb{i}}$ that there exist $J = \{(n_0, j) \,|\, 1 \leq j \leq n_J\} \subseteq \{(i, j) \,|\, i \geq n_0\}$, $J \cap (F \cup \{\mathbb{i}\}) = \emptyset$, $S^- \subseteq_{\mathcal{F}} \mathit{SimOrdCl}_{\{\tilde{a}_{\mathbb{i}}\} \cup \{\tilde{a}_{\mathbb{j}} \,|\, \mathbb{j} \in J\}}$, and (13a–e) hold for ϕ', $\tilde{a}_{\mathbb{i}}$, J, S^-. We put $J_T^\phi = J_T \cup \{\mathbb{i}\} \cup J \subseteq_{\mathcal{F}} \{(i, j) \,|\, i \geq n_0\}$, $J_T^\phi \cap F = \emptyset$, and $S_T^\phi = S_T \cup \{\tilde{a}_{\mathbb{i}} \prec 1\} \cup S^- \subseteq_{\mathcal{F}} \mathit{SimOrdCl}_{\{\tilde{a}_{\mathbb{j}} \,|\, \mathbb{j} \in J_T^\phi\}}$.

(ii) straightforwardly follows. The theorem is proved. □

4 *DPLL* PROCEDURE

We devise a variant of the *DPLL* procedure over finite order clausal theories. Let $a, \dots, f \in \mathit{PropAtom}$, $\mathit{Cn}, \mathit{Cn}_1, \dots, \mathit{Cn}_4 \in \mathit{PropConj}$, $\diamond_1, \diamond_2 \in \{\eqcirc, \prec\}$, $l, l_1, l_2, l_3 \in \mathit{OrdLit}$, $C \in \mathit{OrdCl}$, $T \subseteq \mathit{OrdCl}$. l is a contradiction iff either $l = 0 \eqcirc 1$ or $l = 0 \prec 0$ or $l = 1 \prec 0$ or $l = 1 \prec 1$ or $l = a \prec 0$ or $l = 1 \prec a$ or $l = \mathit{Cn} \prec \mathit{Cn}$. l is a tautology iff either $l = 0 \eqcirc 0$ or $l = 1 \eqcirc 1$ or $l = 0 \prec 1$ or $l = \mathit{Cn} \eqcirc \mathit{Cn}$. $0 \eqcirc a \vee 0 \prec a$ is a 0-dichotomy. $a \prec 1 \vee a \eqcirc 1$ is a 1-dichotomy. $\mathit{Cn}_1 \prec \mathit{Cn}_2 \vee \mathit{Cn}_1 \eqcirc \mathit{Cn}_2 \vee \mathit{Cn}_2 \prec \mathit{Cn}_1$ is a trichotomy. Some auxiliary operations are defined in Table 2. We define a transitivity operation in Table 3. For example,

$$(a \& b \prec c \& e) \blacktriangleright (c \& d \eqcirc a \& f) =$$

$$(a \& b \& d \prec c \& d \& e) \blacktriangleright (c \& d \eqcirc a \& f) =$$

$$a \& b \& d \prec a \& e \& f =$$

$$b \& d \prec e \& f.$$

Table 1: Interpolation rules for $\wedge$, $\vee$, &, $\to$.

Case:	Laws	
$\theta = \theta_1 \wedge \theta_2$		
Positive interpolation $\dfrac{\tilde{a}_{\mathrm{i}} \to \theta_1 \wedge \theta_2}{\left\{\tilde{a}_{\mathrm{i}} \prec \tilde{a}_{\mathrm{i}_1} \vee \tilde{a}_{\mathrm{i}} = \tilde{a}_{\mathrm{i}_1}, \tilde{a}_{\mathrm{i}} \prec \tilde{a}_{\mathrm{i}_2} \vee \tilde{a}_{\mathrm{i}} = \tilde{a}_{\mathrm{i}_2}, \tilde{a}_{\mathrm{i}_1} \to \theta_1, \tilde{a}_{\mathrm{i}_2} \to \theta_2\right\}}$	(7)	(14)
$\lvert \text{Consequent} \rvert = 12 + \lvert \tilde{a}_{\mathrm{i}_1} \to \theta_1 \rvert + \lvert \tilde{a}_{\mathrm{i}_2} \to \theta_2 \rvert \leq 20 + \lvert \tilde{a}_{\mathrm{i}_1} \to \theta_1 \rvert + \lvert \tilde{a}_{\mathrm{i}_2} \to \theta_2 \rvert$		
Negative interpolation $\dfrac{\theta_1 \wedge \theta_2 \to \tilde{a}_{\mathrm{i}}}{\left\{\tilde{a}_{\mathrm{i}_1} \prec \tilde{a}_{\mathrm{i}} \vee \tilde{a}_{\mathrm{i}_1} = \tilde{a}_{\mathrm{i}} \vee \tilde{a}_{\mathrm{i}_2} \prec \tilde{a}_{\mathrm{i}} \vee \tilde{a}_{\mathrm{i}_2} = \tilde{a}_{\mathrm{i}}, \theta_1 \to \tilde{a}_{\mathrm{i}_1}, \theta_2 \to \tilde{a}_{\mathrm{i}_2}\right\}}$	(9)	(15)
$\lvert \text{Consequent} \rvert = 12 + \lvert \theta_1 \to \tilde{a}_{\mathrm{i}_1} \rvert + \lvert \theta_2 \to \tilde{a}_{\mathrm{i}_2} \rvert \leq 20 + \lvert \theta_1 \to \tilde{a}_{\mathrm{i}_1} \rvert + \lvert \theta_2 \to \tilde{a}_{\mathrm{i}_2} \rvert$		
$\theta = \theta_1 \vee \theta_2$		
Positive interpolation $\dfrac{\tilde{a}_{\mathrm{i}} \to (\theta_1 \vee \theta_2)}{\left\{\tilde{a}_{\mathrm{i}} \prec \tilde{a}_{\mathrm{i}_1} \vee \tilde{a}_{\mathrm{i}} = \tilde{a}_{\mathrm{i}_1} \vee \tilde{a}_{\mathrm{i}} \prec \tilde{a}_{\mathrm{i}_2} \vee \tilde{a}_{\mathrm{i}} = \tilde{a}_{\mathrm{i}_2}, \tilde{a}_{\mathrm{i}_1} \to \theta_1, \tilde{a}_{\mathrm{i}_2} \to \theta_2\right\}}$	(6)	(16)
$\lvert \text{Consequent} \rvert = 12 + \lvert \tilde{a}_{\mathrm{i}_1} \to \theta_1 \rvert + \lvert \tilde{p}_{\mathrm{i}_2} \to \theta_2 \rvert \leq 20 + \lvert \tilde{a}_{\mathrm{i}_1} \to \theta_1 \rvert + \lvert \tilde{p}_{\mathrm{i}_2} \to \theta_2 \rvert$		
Negative interpolation $\dfrac{(\theta_1 \vee \theta_2) \to \tilde{a}_{\mathrm{i}}}{\left\{\tilde{a}_{\mathrm{i}_1} \prec \tilde{a}_{\mathrm{i}} \vee \tilde{a}_{\mathrm{i}_1} = \tilde{a}_{\mathrm{i}}, \tilde{a}_{\mathrm{i}_2} \prec \tilde{a}_{\mathrm{i}} \vee \tilde{a}_{\mathrm{i}_2} = \tilde{a}_{\mathrm{i}}, \theta_1 \to \tilde{a}_{\mathrm{i}_1}, \theta_2 \to \tilde{a}_{\mathrm{i}_2}\right\}}$	(8)	(17)
$\lvert \text{Consequent} \rvert = 12 + \lvert \theta_1 \to \tilde{a}_{\mathrm{i}_1} \rvert + \lvert \theta_2 \to \tilde{a}_{\mathrm{i}_2} \rvert \leq 20 + \lvert \theta_1 \to \tilde{a}_{\mathrm{i}_1} \rvert + \lvert \theta_2 \to \tilde{a}_{\mathrm{i}_2} \rvert$		
$\theta = \theta_1 \,\&\, \theta_2$		
Positive interpolation $\dfrac{\tilde{a}_{\mathrm{i}} \to \theta_1 \,\&\, \theta_2}{\left\{\tilde{a}_{\mathrm{i}} \prec \tilde{a}_{\mathrm{i}_1} \,\&\, \tilde{a}_{\mathrm{i}_2} \vee \tilde{a}_{\mathrm{i}} = \tilde{a}_{\mathrm{i}_1} \,\&\, \tilde{a}_{\mathrm{i}_2}, \tilde{a}_{\mathrm{i}_1} \to \theta_1, \tilde{a}_{\mathrm{i}_2} \to \theta_2\right\}}$		(18)
$\lvert \text{Consequent} \rvert = 8 + \lvert \tilde{a}_{\mathrm{i}_1} \to \theta_1 \rvert + \lvert \tilde{a}_{\mathrm{i}_2} \to \theta_2 \rvert \leq 20 + \lvert \tilde{a}_{\mathrm{i}_1} \to \theta_1 \rvert + \lvert \tilde{a}_{\mathrm{i}_2} \to \theta_2 \rvert$		
Negative interpolation $\dfrac{\theta_1 \wedge \theta_2 \to \tilde{a}_{\mathrm{i}}}{\left\{\tilde{a}_{\mathrm{i}_1} \,\&\, \tilde{a}_{\mathrm{i}_2} \prec \tilde{a}_{\mathrm{i}} \vee \tilde{a}_{\mathrm{i}_1} \,\&\, \tilde{a}_{\mathrm{i}_2} = \tilde{a}_{\mathrm{i}}, \theta_1 \to \tilde{a}_{\mathrm{i}_1}, \theta_2 \to \tilde{a}_{\mathrm{i}_2}\right\}}$		(19)
$\lvert \text{Consequent} \rvert = 8 + \lvert \theta_1 \to \tilde{a}_{\mathrm{i}_1} \rvert + \lvert \theta_2 \to \tilde{a}_{\mathrm{i}_2} \rvert \leq 20 + \lvert \theta_1 \to \tilde{a}_{\mathrm{i}_1} \rvert + \lvert \theta_2 \to \tilde{a}_{\mathrm{i}_2} \rvert$		
$\theta = \theta_1 \to 0$		
Positive interpolation $\dfrac{\tilde{a}_{\mathrm{i}} \to (\theta_1 \to 0)}{\left\{\tilde{a}_{\mathrm{i}} = 0 \vee \tilde{a}_{\mathrm{i}_1} = 0, \theta_1 \to \tilde{a}_{\mathrm{i}_1}\right\}}$	(10)	(20)
$\lvert \text{Consequent} \rvert = 6 + \lvert \theta_1 \to \tilde{a}_{\mathrm{i}_1} \rvert \leq 20 + \lvert \theta_1 \to \tilde{a}_{\mathrm{i}_1} \rvert$		
Negative interpolation $\dfrac{(\theta_1 \to 0) \to \tilde{a}_{\mathrm{i}}}{\left\{0 \prec \tilde{a}_{\mathrm{i}_1} \vee \tilde{a}_{\mathrm{i}} = 1, \tilde{a}_{\mathrm{i}_1} \to \theta_1\right\}}$		(21)
$\lvert \text{Consequent} \rvert = 6 + \lvert \tilde{a}_{\mathrm{i}_1} \to \theta_1 \rvert \leq 20 + \lvert \tilde{a}_{\mathrm{i}_1} \to \theta_1 \rvert$		
$\theta = \theta_1 \to \theta_2, \theta_2 \neq 0$		
Positive interpolation $\dfrac{\tilde{a}_{\mathrm{i}} \to (\theta_1 \to \theta_2)}{\left\{\tilde{a}_{\mathrm{i}} \,\&\, \tilde{a}_{\mathrm{i}_1} \prec \tilde{a}_{\mathrm{i}_2} \vee \tilde{a}_{\mathrm{i}} \,\&\, \tilde{a}_{\mathrm{i}_1} = \tilde{a}_{\mathrm{i}_2}, \theta_1 \to \tilde{a}_{\mathrm{i}_1}, \tilde{a}_{\mathrm{i}_2} \to \theta_2\right\}}$	(10)	(22)
$\lvert \text{Consequent} \rvert = 8 + \lvert \theta_1 \to \tilde{a}_{\mathrm{i}_1} \rvert + \lvert \tilde{a}_{\mathrm{i}_2} \to \theta_2 \rvert \leq 20 + \lvert \theta_1 \to \tilde{a}_{\mathrm{i}_1} \rvert + \lvert \tilde{a}_{\mathrm{i}_2} \to \theta_2 \rvert$		
Negative interpolation $\dfrac{(\theta_1 \to \theta_2) \to \tilde{a}_{\mathrm{i}}}{\left\{\tilde{a}_{\mathrm{i}_1} \prec \tilde{a}_{\mathrm{i}_2} \vee \tilde{a}_{\mathrm{i}_1} = \tilde{a}_{\mathrm{i}_2} \vee \tilde{a}_{\mathrm{i}_2} \prec \tilde{a}_{\mathrm{i}_1} \,\&\, \tilde{a}_{\mathrm{i}} \vee \tilde{a}_{\mathrm{i}_2} = \tilde{a}_{\mathrm{i}_1} \,\&\, \tilde{a}_{\mathrm{i}}, \tilde{a}_{\mathrm{i}_2} \prec \tilde{a}_{\mathrm{i}_1} \vee \tilde{a}_{\mathrm{i}} = 1, \tilde{a}_{\mathrm{i}_1} \to \theta_1, \theta_2 \to \tilde{a}_{\mathrm{i}_2}\right\}}$		(23)
$\lvert \text{Consequent} \rvert = 20 + \lvert \tilde{a}_{\mathrm{i}_1} \to \theta_1 \rvert + \lvert \theta_2 \to \tilde{a}_{\mathrm{i}_2} \rvert \leq 20 + \lvert \tilde{a}_{\mathrm{i}_1} \to \theta_1 \rvert + \lvert \theta_2 \to \tilde{a}_{\mathrm{i}_2} \rvert$		

Table 2: Auxiliary operations.

$$Cn_1 \odot Cn_2 = \{a^{m+n} \,|\, a^m \in Cn_1, a^n \in Cn_2\} \cup \{a^m \,|\, a^m \in Cn_1, a \notin atoms(Cn_2)\} \cup \{a^n \,|\, a^n \in Cn_2, a \notin atoms(Cn_1)\} \in PropConj \cup \{\emptyset\},$$

$$Cn_1 \Downarrow Cn_2 = \{a^{m-n} \,|\, a^m \in Cn_1, a^n \in Cn_2, m > n\} \cup \{a^m \,|\, a^m \in Cn_1, a \notin atoms(Cn_2)\} \in PropConj \cup \{\emptyset\} \text{ if } Cn_2 \sqsubseteq Cn_1,$$

$$Cn_1 \rhd Cn_2 = \{a^{n-m} \,|\, a^m \in Cn_1, a^n \in Cn_2, n > m\} \cup \{a^n \,|\, a^n \in Cn_2, a \notin atoms(Cn_1)\} \in PropConj \cup \{\emptyset\}$$

$Cn_1, Cn_2 \in PropConj \cup \{\emptyset\}$.

Table 3: Transitivity operation.

$$(Cn_1 \diamond_1 Cn_2) \blacktriangleright (Cn_3 \diamond_2 Cn_4) = \begin{cases} 1 \diamond 1 & \text{if } Cn_7 = Cn_8 = \emptyset, \\ \square & \text{if } Cn_7 = \emptyset, Cn_8 \neq \emptyset, \\ \square & \text{if } Cn_7 \neq \emptyset, Cn_8 = \emptyset, \diamond = \eqcirc, \\ 1 \eqcirc 1 & \text{if } Cn_7 \neq \emptyset, Cn_8 = \emptyset, \diamond = \prec, \\ Cn_7 \diamond Cn_8 & \text{if } Cn_7 \neq \emptyset, Cn_8 \neq \emptyset, \end{cases}$$

$$Cn_5 = (Cn_1 \odot (Cn_2 \rhd Cn_3)),$$

$$Cn_6 = (((Cn_2 \odot (Cn_2 \rhd Cn_3)) \Downarrow Cn_3) \odot Cn_4),$$

$$Cn_7 = (Cn_5 \Downarrow (Cn_5 \sqcap Cn_6)),$$

$$Cn_8 = (Cn_6 \Downarrow (Cn_5 \sqcap Cn_6)),$$

$$\diamond = \begin{cases} \eqcirc \text{ if } \diamond_1 = \diamond_2 = \eqcirc, \\ \prec \text{ else}, \end{cases}$$

$$(Cn_1 \diamond_1 Cn_2) \blacktriangleright (Cn_3 \diamond_2 Cn_4) \in OrdCl$$

$Cn_1, \dots, Cn_4 \in PropConj$, $\diamond_1, \diamond_2 \in \{\eqcirc, \prec\}$.

An auxiliary simplification function is defined in Table 4. Basic rules are defined as follows:

(*Contradiction simplification rule*) (24)

$$\frac{T}{T - \{l \vee C\} \cup \{C\}}$$

$l \vee C \in T$, l *is a contradiction.*

(*One literal 0-simplification rule*) (25)

$$\frac{T}{T - \{l \vee C\} \cup simpl(a \eqcirc 0, l \vee C)}$$

$a \eqcirc 0, l \vee C \in T, a \in atoms(l)$.

(*One literal 1-simplification rule*) (26)

$$\frac{T}{T - \{l \vee C\} \cup simpl(a \eqcirc 1, l \vee C)}$$

$a \eqcirc 1, l \vee C \in T, a \in atoms(l)$.

(*0-dichotomy branching rule*) (27)

$$\frac{T}{T \cup \{l_1\} \mid T \cup \{l_2\}}$$

$l_1 \vee l_2$ *is a 0-dichotomy*, $atoms(l_1 \vee l_2) \subseteq atoms(T)$.

(*1-dichotomy branching rule*) (28)

$$\frac{T}{T \cup \{l_1\} \mid T \cup \{l_2\}}$$

$l_1 \vee l_2$ *is a 1-dichotomy*, $atoms(l_1 \vee l_2) \subseteq atoms(T)$.

Table 4: Auxiliary simplification function.

$$
\begin{aligned}
&\mathit{simpl}(a \eqcirc 0, a \diamond \varepsilon \vee C) = \{0 \diamond \varepsilon \vee C\} \ \mathit{if}\ a \eqcirc 0 \neq a \diamond \varepsilon \vee C,\\
&\mathit{simpl}(a \eqcirc 0, \varepsilon \diamond a \vee C) = \{\varepsilon \diamond 0 \vee C\} \ \mathit{if}\ a \eqcirc 0 \neq \varepsilon \diamond a \vee C,\\
&\mathit{simpl}(a \eqcirc 0, Cn_1 \eqcirc Cn_2 \vee C) = \{ \bigvee_{b \in \mathit{atoms}(Cn_2)} b \eqcirc 0 \vee C\} \ \mathit{if}\ a \in \mathit{atoms}(Cn_1),\\
&\mathit{simpl}(a \eqcirc 0, Cn_1 \prec Cn_2 \vee C) = \{0 \prec b \vee C \,|\, b \in \mathit{atoms}(Cn_2)\} \ \mathit{if}\ a \in \mathit{atoms}(Cn_1),\\
&\mathit{simpl}(a \eqcirc 0, Cn_1 \prec Cn_2 \vee C) = \{C\} \ \mathit{if}\ a \in \mathit{atoms}(Cn_2);\\
&\mathit{simpl}(a \eqcirc 1, a \diamond \varepsilon \vee C) = \{1 \diamond \varepsilon \vee C\} \ \mathit{if}\ a \eqcirc 1 \neq a \diamond \varepsilon \vee C,\\
&\mathit{simpl}(a \eqcirc 1, \varepsilon \diamond a \vee C) = \{\varepsilon \diamond 1 \vee C\} \ \mathit{if}\ a \eqcirc 1 \neq \varepsilon \diamond a \vee C,\\
&\mathit{simpl}(a \eqcirc 1, Cn_1 \eqcirc Cn_2 \vee C) = \{(Cn_1 - \{a^n\}) \eqcirc Cn_2 \vee C\} \ \mathit{if}\ \{a\} \subset \mathit{atoms}(Cn_1), a^n \in Cn_1,\\
&\mathit{simpl}(a \eqcirc 1, Cn_1 \eqcirc Cn_2 \vee C) = \{b \eqcirc 1 \vee C \,|\, b \in \mathit{atoms}(Cn_2)\} \ \mathit{if}\ \{a\} = \mathit{atoms}(Cn_1),\\
&\mathit{simpl}(a \eqcirc 1, Cn_1 \prec Cn_2 \vee C) = \{(Cn_1 - \{a^n\}) \prec Cn_2 \vee C\} \ \mathit{if}\ \{a\} \subset \mathit{atoms}(Cn_1), a^n \in Cn_1,\\
&\mathit{simpl}(a \eqcirc 1, Cn_1 \prec Cn_2 \vee C) = \{C\} \ \mathit{if}\ \{a\} = \mathit{atoms}(Cn_1),\\
&\mathit{simpl}(a \eqcirc 1, Cn_1 \prec Cn_2 \vee C) = \{Cn_1 \prec (Cn_2 - \{a^n\}) \vee C\} \ \mathit{if}\ \{a\} \subset \mathit{atoms}(Cn_2), a^n \in Cn_2,\\
&\mathit{simpl}(a \eqcirc 1, Cn_1 \prec Cn_2 \vee C) = \{ \bigvee_{b \in \mathit{atoms}(Cn_1)} b \prec 1 \vee C\} \ \mathit{if}\ \{a\} = \mathit{atoms}(Cn_2);\\
&\mathit{simpl}(l, C) \subseteq_{\mathcal{F}} \mathit{OrdCl}
\end{aligned}
$$

$a \in \mathit{PropAtom}$, $\varepsilon \in \{0, 1\}$, $Cn_1, Cn_2 \in \mathit{PropConj}$, $l \in \{a \eqcirc 0, a \eqcirc 1\}$, $C \in \mathit{OrdCl}$.

(*One literal transitivity rule*) (29)

$$\frac{T}{T \cup \{(Cn_1 \diamond_1 Cn_2) \blacktriangleright (Cn_3 \diamond_2 Cn_4)\}}$$

T is a unit order clausal theory,
$Cn_1 \diamond_1 Cn_2, Cn_3 \diamond_2 Cn_4 \in T$,
for all $a \in \mathit{atoms}(Cn_1, \dots, Cn_4)$, $0 \prec a, a \prec 1 \in T$.

(*Trichotomy branching rule*) (30)

$$\frac{T}{T - \{l_1 \vee C\} \cup \{l_1\} \mid T - \{l_1 \vee C\} \cup \{C\} \cup \{l_2\} \mid T - \{l_1 \vee C\} \cup \{C\} \cup \{l_3\}}$$

$l_1 \vee C \in T$, $C \neq \square$, $l_1 \vee l_2 \vee l_3$ *is a trichotomy,*
for all $a \in \mathit{atoms}(l_1, l_2, l_3)$, $0 \prec a, a \prec 1 \in T$.

Rules (24)–(30) are sound in view of satisfiability. The proof is straightforward. The refutational completeness argument of the basic rules, Theorem 4.2(ii), can be provided using the excess literal technique (Anderson and Bledsoe, 1970). From this point of view, Rules (24) and (29) handle the base case: T is a unit order clausal theory; while Rule (30) handles the induction one: it subtracts the excess literal measure of T at least by 1 for the clausal theory in every branch of its consequent.

T is closed under Rules (24) and (29) iff for every application of Rules (24) and (29) of the form $\frac{T}{T'}$, $T' = T$. By $\mathit{trans}(T) \subseteq \mathit{OrdCl}$ we denote the least set such that $\mathit{trans}(T) \supseteq T$ and $\mathit{trans}(T)$ is closed under Rules (24), (29).

Using the basic rules, one can construct a finitely generated tree with the input theory as the root in the usual manner, so as the classical *DPLL* procedure does; for every parent vertex, there exists an application of Rule (24)–(30) such that the theory in its antecedent is in the parent vertex and the theories in its consequent are in the children vertices. A branch of a tree is closed iff it contains a vertex T' such that $\square \in T'$. A branch of a tree is open iff it is not closed. A tree is closed iff every branch of it is finite and closed. A closed tree is finite by König's Lemma. A tree is open iff it is not closed. A tree is linear iff it consists of only one branch, beginning in its root and ending in its only leaf.

The following lemma shows that Rules (24) and (29) are refutation complete for a special kind of (countable) unit order clausal theory, which will be exploited in the base case of Theorem 4.2(ii).

Lemma 4.1. *Let* $T = \mathit{trans}(T) \subseteq \mathit{OrdCl}$ *be a count-*

able unit order clausal theory such that for all $a \in atoms(T)$, either there exists $a \eqcirc \varepsilon \in T$, $\varepsilon \in \{0,1\}$, satisfying, for all $C \in T$ and $C \neq a \eqcirc \varepsilon$, $a \notin atoms(C)$; or $0 \prec a, a \prec 1 \in T$. There exists a partial model $\mathfrak{A}$ of T, $dom(\mathfrak{A}) = atoms(T)$.

Proof. By the lemma assumption that T is a unit order clausal theory, $\square \notin T = trans(T)$. In addition, by the lemma assumption that T is a countable set, there exist $\gamma \leq \omega$ and a sequence $\delta : \gamma \longrightarrow atoms(T)$ of $atoms(T)$. At first, we define a partial valuation $\mathcal{V}_\alpha$ by recursion on $\alpha \leq \gamma$ in Table 5. It is straightforward to prove the following statements:

For all $\alpha \leq \gamma$, $\mathcal{V}_\alpha$ is a partial valuation, $dom(\mathcal{V}_\alpha) = \delta[\alpha]$; and for all $\alpha \leq \alpha' \leq \gamma$, $\mathcal{V}_\alpha \subseteq \mathcal{V}_{\alpha'}$. (31)

The proof is by induction on $\alpha \leq \gamma$.

For all $\alpha \leq \gamma$ and $l \in T$ such that $atoms(l) \subseteq dom(\mathcal{V}_\alpha)$, $\mathcal{V}_\alpha \models l$. (32)

The proof is by induction on $\alpha \leq \gamma$.

We put $\mathfrak{A} = \mathcal{V}_\gamma$. By (31), $\mathfrak{A}$ is a partial valuation, $dom(\mathfrak{A}) \overset{(31)}{=\!=} \delta[\gamma] = atoms(T)$. Let $l \in T$. Then $atoms(l) \subseteq atoms(T) = dom(\mathfrak{A})$ and $\mathfrak{A} \overset{(32)}{\models\!\!=} l$. So, $\mathfrak{A} \models T$. We conclude that $\mathfrak{A}$ is a partial model of T, $dom(\mathfrak{A}) = atoms(T)$. $\square$

The *DPLL* procedure is refutation sound and complete.

Theorem 4.2 (Refutational Soundness and Completeness of the *DPLL* Procedure)**.** *Let $S \subseteq_{\mathcal{F}} OrdCl$.*

(i) *If there exists a closed tree Tree with the root S constructed using Rules* (24)–(30)*, then S is unsatisfiable.*

(ii) *There exists a finite tree Tree with the root S constructed using Rules* (24)–(30) *with the following properties:*

if S is unsatisfiable, then Tree is closed; (33)

if S is satisfiable, then Tree is open and there exists a partial model $\mathfrak{A}$ of S, $dom(\mathfrak{A}) = atoms(S)$, related to Tree. (34)

Proof. (i) The proof is by induction on the structure of *Tree* using Rules (24)–(30).

(ii) In the first phase, we can construct a finite tree $Tree^*$ with leaves S_i, $i \leq n$, using Rules (24)–(28) such that for all $i \leq n$, $atoms(S_i) \subseteq atoms(S)$, $S_i \models_P S$; for all $a \in atoms(S_i)$, either there exists $a \eqcirc \varepsilon \in S_i$, $\varepsilon \in \{0,1\}$, satisfying, for all $C \in S_i$ and $C \neq a \eqcirc \varepsilon$, $a \notin atoms(C)$; or $0 \prec a, a \prec 1 \in S_i$; S is satisfiable if and only if there exists $i^* \leq n$ such that S_{i^*} is satisfiable. The proof is by induction on $\|atoms(S)\|$.

In the second phase, we exploit the excess literal technique. Let $S^F \subseteq_{\mathcal{F}} OrdCl$. We define $elmeasure(S^F) = (\sum_{C \in S^F} \|C\|) - \|S^F\|$. For all $i \leq n$, there exists a finite tree $Tree_i$ with the root S_i constructed using Rules (24), (29), (30) with the following properties:

if S_i is unsatisfiable, then $Tree_i$ is closed; (35)

if S_i is satisfiable, then $Tree_i$ is open and there exists a partial model $\mathfrak{A}_i$ of S_i, $dom(\mathfrak{A}_i) = atoms(S_i)$, related to $Tree_i$. (36)

Let $i \leq n$. We proceed by induction on $elmeasure(S_i)$.

Case 1: $elmeasure(S_i) = 0$. We distinguish two cases.

Case 1.1: $\square \in S_i$. We put $Tree_i = S_i$. Then S_i is unsatisfiable; $Tree_i$ is a closed tree with the root S_i; (35) holds and (36) holds trivially.

Case 1.2: $\square \notin S_i$. Then S_i is a unit order clausal theory; there exists a finite linear tree $Tree_i$ with the root S_i and the leaf $trans(S_i)$ constructed using Rules (24) and (29). We get two cases.

Case 1.2.1: $\square \in trans(S_i)$. Then $Tree_i$ is closed; its only branch from S_i to $trans(S_i)$ is closed; by (i) for $Tree_i$, S_i is unsatisfiable; (35) holds and (36) holds trivially.

Case 1.2.2: $\square \notin trans(S_i)$. Then $Tree_i$ is open; its only branch from S_i to $trans(S_i)$ is open; $trans(S_i)$ is a unit order clausal theory; we have, for all $a \in atoms(S_i)$, either there exists $a \eqcirc \varepsilon \in S_i$, $\varepsilon \in \{0,1\}$, satisfying, for all $C \in S_i$ and $C \neq a \eqcirc \varepsilon$, $a \notin atoms(C)$; or $0 \prec a, a \prec 1 \in S_i$; for all $C \in trans(S_i) - S_i$, for all $a \in atoms(C)$, $0 \prec a, a \prec 1 \in S_i \subseteq trans(S_i)$; the proof is by induction on $\|trans(S_i) - S_i\|$ using Rule (29); for all $a \in atoms(S_i) = atoms(trans(S_i))$, either there exists $a \eqcirc \varepsilon \in S_i \subseteq trans(S_i)$, $\varepsilon \in \{0,1\}$, satisfying, for all $C \in trans(S_i)$ and $C \neq a \eqcirc \varepsilon$, $a \notin atoms(C)$; or $0 \prec a, a \prec 1 \in S_i \subseteq trans(S_i)$; by Lemma 4.1 for $trans(S_i)$, there exists a partial model $\mathfrak{A}_i$ of $trans(S_i)$, $dom(\mathfrak{A}_i) = atoms(trans(S_i))$; $\mathfrak{A}_i$, $dom(\mathfrak{A}_i) = atoms(trans(S_i)) = atoms(S_i)$, is a partial model of $S_i \subseteq trans(S_i)$ related to $Tree_i$; S_i is satisfiable; (36) holds and (35) holds trivially.

Case 2: $elmeasure(S_i) > 0$. Then there exist $l_1, l_2, l_3 \in OrdLit$, $\square \neq C \in OrdCl$, and $l_1 \vee C \in S_i$, $l_1 \vee l_2 \vee l_3$ is a trichotomy. We put $S_i^1 = (S_i - \{l_1 \vee C\}) \cup \{l_1\} \subseteq_{\mathcal{F}} OrdCl$, $S_i^2 = (S_i - \{l_1 \vee C\}) \cup \{C\} \cup \{l_2\} \subseteq_{\mathcal{F}} OrdCl$, $S_i^3 = (S_i - \{l_1 \vee C\}) \cup \{C\} \cup \{l_2\} \subseteq_{\mathcal{F}} OrdCl$. Then

$$\frac{S_i}{S_i^1 \mid S_i^2 \mid S_i^3}$$

is an application of Rule (30); for all $1 \leq j \leq 3$, $elmeasure(S_i^j) < elmeasure(S_i)$; for all $1 \leq j \leq 3$, by induction hypothesis for S_i^j, there exists a finite tree

Table 5: $\mathcal{V}_\alpha$.

$$\mathcal{V}_0 = \emptyset;$$

$$\mathcal{V}_\alpha = \mathcal{V}_{\alpha-1} \cup \{(\delta(\alpha-1), \lambda_{\alpha-1})\} \quad (1 \leq \alpha \leq \gamma \, \textit{is a successor ordinal}),$$

$$\begin{aligned}
\mathbb{E}_{\alpha-1} = {} & \left\{ \left(\frac{\|Cn_1\|^{\mathcal{V}_{\alpha-1}}}{\|Cn_2\|^{\mathcal{V}_{\alpha-1}}} \right)^{\frac{1}{n}} \,\middle|\, Cn_1 \eqcirc \delta(\alpha-1)^n \,\&\, Cn_2 \in T, Cn_1, Cn_2 \in \mathit{PropConj}, \mathit{atoms}(Cn_1, Cn_2) \subseteq \mathit{dom}(\mathcal{V}_{\alpha-1}) \right\} \cup \\
& \left\{ \left(\|Cn_1\|^{\mathcal{V}_{\alpha-1}} \right)^{\frac{1}{n}} \,\middle|\, Cn_1 \eqcirc \delta(\alpha-1)^n \in T, Cn_1 \in \mathit{PropConj}, \mathit{atoms}(Cn_1) \subseteq \mathit{dom}(\mathcal{V}_{\alpha-1}) \right\} \cup \\
& \left\{ \|\varepsilon\|^{\mathcal{V}_{\alpha-1}} \,|\, \delta(\alpha-1) \eqcirc \varepsilon \in T, \varepsilon \in \{\mathit{0}, \mathit{1}\} \right\},
\end{aligned}$$

$$\begin{aligned}
\mathbb{D}_{\alpha-1} = {} & \left\{ \left(\frac{\|Cn_1\|^{\mathcal{V}_{\alpha-1}}}{\|Cn_2\|^{\mathcal{V}_{\alpha-1}}} \right)^{\frac{1}{n}} \,\middle|\, Cn_1 \prec \delta(\alpha-1)^n \,\&\, Cn_2 \in T, Cn_1, Cn_2 \in \mathit{PropConj}, \mathit{atoms}(Cn_1, Cn_2) \subseteq \mathit{dom}(\mathcal{V}_{\alpha-1}) \right\} \cup \\
& \left\{ \left(\|Cn_1\|^{\mathcal{V}_{\alpha-1}} \right)^{\frac{1}{n}} \,\middle|\, Cn_1 \prec \delta(\alpha-1)^n \in T, Cn_1 \in \mathit{PropConj}, \mathit{atoms}(Cn_1) \subseteq \mathit{dom}(\mathcal{V}_{\alpha-1}) \right\},
\end{aligned}$$

$$\begin{aligned}
\mathbb{U}_{\alpha-1} = {} & \left\{ \left(\frac{\|Cn_1\|^{\mathcal{V}_{\alpha-1}}}{\|Cn_2\|^{\mathcal{V}_{\alpha-1}}} \right)^{\frac{1}{n}} \,\middle|\, \delta(\alpha-1)^n \,\&\, Cn_2 \prec Cn_1 \in T, Cn_1, Cn_2 \in \mathit{PropConj}, \mathit{atoms}(Cn_1, Cn_2) \subseteq \mathit{dom}(\mathcal{V}_{\alpha-1}) \right\} \cup \\
& \left\{ \left(\|Cn_1\|^{\mathcal{V}_{\alpha-1}} \right)^{\frac{1}{n}} \,\middle|\, \delta(\alpha-1)^n \prec Cn_1 \in T, Cn_1 \in \mathit{PropConj}, \mathit{atoms}(Cn_1) \subseteq \mathit{dom}(\mathcal{V}_{\alpha-1}) \right\},
\end{aligned}$$

$$\lambda_{\alpha-1} = \begin{cases} \dfrac{\bigvee \mathbb{D}_{\alpha-1} + \bigwedge \mathbb{U}_{\alpha-1}}{2} & \textit{if } \mathbb{E}_{\alpha-1} = \emptyset, \\ \bigvee \mathbb{E}_{\alpha-1} & \textit{else}; \end{cases}$$

$$\mathcal{V}_\gamma = \bigcup_{\alpha<\gamma} \mathcal{V}_\alpha \quad (\gamma \, \textit{is a limit ordinal})$$

$Tree_i^j$ with the root S_i^j constructed using Rules (24), (29), (30), and (35), (36) hold for $Tree_i^j$. We put

$$Tree_i = \frac{S_i}{Tree_i^1 \mid Tree_i^2 \mid Tree_i^3}.$$

Then $Tree_i$ is a finite tree with the root S_i constructed using Rules (24), (29), (30). We get two cases.

Case 2.1: S_i is unsatisfiable. Then, for all $1 \leq j \leq 3$, S_i^j is unsatisfiable; by (35) for $Tree_i^j$, $Tree_i^j$ is closed; $Tree_i$ is closed; (35) holds and (36) holds trivially.

Case 2.2: S_i is satisfiable. Then there exists $1 \leq j^* \leq 3$ and $S_i^{j^*}$ is satisfiable; by (36) for $Tree_i^{j^*}$, $Tree_i^{j^*}$ is open, there exists a partial model $\mathfrak{A}_i^{j^*}$ of $S_i^{j^*}$, $dom(\mathfrak{A}_i^{j^*}) = atoms(S_i^{j^*})$, related to $Tree_i^{j^*}$; $Tree_i$ is open; we have $l_1 \vee l_2 \vee l_3$ is a trichotomy; $atoms(l_1) = atoms(l_2) = atoms(l_3)$, $atoms(S_i^{j^*}) \subseteq atoms(S_i)$, $S_i^{j^*} \models_P S_i$. We put $\mathfrak{A}_i = \mathfrak{A}_i^{j^*} \cup \{(a,0) \,|\, a \in atoms(S_i) - atoms(S_i^{j^*})\}$, $dom(\mathfrak{A}_i) = atoms(S_i)$, a partial valuation. Then $\mathfrak{A}_i|_{atoms(S_i^{j^*})} = \mathfrak{A}_i^{j^*} \models S_i^{j^*}$, $\mathfrak{A}_i \models S_i$, $\mathfrak{A}_i$, $dom(\mathfrak{A}_i) = atoms(S_i)$, is a partial model of S_i, related to $Tree_i$; (36) holds and (35) holds trivially. The induction is completed.

We construct *Tree* from $Tree^*$ by replacing the leaf S_i with $Tree_i$ for every $i \leq n$. We have $Tree^*$, for all $i \leq n$, $Tree_i$ are finite. Hence, *Tree* is finite. It remains to prove (33) and (34).

Let S be unsatisfiable. We have S is satisfiable if and only if there exists $i^* \leq n$ such that S_{i^*} is satisfiable. Then, for all $i \leq n$, S_i is unsatisfiable; by (35) for $Tree_i$, $Tree_i$ is closed; *Tree* is closed; (33) holds.

Let S be satisfiable. We have S is satisfiable if and only if there exists $i^* \leq n$ such that S_{i^*} is satisfiable. Then, there exists $i^* \leq n$ and S_{i^*} is satisfiable; by (36) for $Tree_{i^*}$, $Tree_{i^*}$ is open, there exists a partial model $\mathfrak{A}_{i^*}$ of S_{i^*}, $dom(\mathfrak{A}_{i^*}) = atoms(S_{i^*})$, related to $Tree_{i^*}$; *Tree* is open; we have, for all $i \leq n$, $atoms(S_i) \subseteq atoms(S)$, $S_i \models_P S$; $atoms(S_{i^*}) \subseteq atoms(S)$, $S_{i^*} \models_P S$. We put $\mathfrak{A} = \mathfrak{A}_{i^*} \cup \{(a,0) \,|\, a \in atoms(S) - atoms(S_{i^*})\}$, $dom(\mathfrak{A}) = atoms(S)$, a partial valuation. Then $\mathfrak{A}|_{atoms(S_{i^*})} = \mathfrak{A}_{i^*} \models S_{i^*}$, $\mathfrak{A} \models S$, $\mathfrak{A}$, $dom(\mathfrak{A}) = atoms(S)$, is a partial model of S related to *Tree*; (34) holds. The theorem is proved. □

The set of basic rules has been proposed as a minimal one, which is suitable for theoretical purposes; i.e. not to get complicated soundness and completeness arguments. For practical computing, it may be

Table 6: Translation of ϕ to S^{ϕ}.

$$\phi = a \rightarrow 0 \vee (a \rightarrow a \& b) \rightarrow b$$

$$\{\tilde{a}_0 \prec 1, (\underbrace{a \rightarrow 0}_{\tilde{a}_1} \vee \underbrace{(a \rightarrow a \& b) \rightarrow b}_{\tilde{a}_2}) \rightarrow \tilde{a}_0\} \tag{17}$$

$$\{\tilde{a}_0 \prec 1, \tilde{a}_1 \prec \tilde{a}_0 \vee \tilde{a}_1 = \tilde{a}_0, \tilde{a}_2 \prec \tilde{a}_0 \vee \tilde{a}_2 = \tilde{a}_0, (\underbrace{a}_{\tilde{a}_3} \rightarrow 0) \rightarrow \tilde{a}_1, ((\underbrace{a \rightarrow a \& b}_{\tilde{a}_4}) \rightarrow \underbrace{b}_{\tilde{a}_5}) \rightarrow \tilde{a}_2\} \qquad (21),(23)$$

$$\{\tilde{a}_0 \prec 1, \tilde{a}_1 \prec \tilde{a}_0 \vee \tilde{a}_1 = \tilde{a}_0, \tilde{a}_2 \prec \tilde{a}_0 \vee \tilde{a}_2 = \tilde{a}_0, 0 \prec \tilde{a}_3 \vee \tilde{a}_1 = 1, \tilde{a}_3 \prec a \vee \tilde{a}_3 = a, \tilde{a}_4 \prec \tilde{a}_5 \vee \tilde{a}_4 = \tilde{a}_5 \vee \tilde{a}_5 \prec \tilde{a}_4 \& \tilde{a}_2 \vee \tilde{a}_5 = \tilde{a}_4 \& \tilde{a}_2, \tilde{a}_5 \prec \tilde{a}_4 \vee \tilde{a}_2 = 1,$$
$$b \prec \tilde{a}_5 \vee b = \tilde{a}_5, \tilde{a}_4 \rightarrow (\underbrace{a}_{\tilde{a}_6} \rightarrow \underbrace{a \& b}_{\tilde{a}_7})\} \tag{22}$$

$$\{\tilde{a}_0 \prec 1, \tilde{a}_1 \prec \tilde{a}_0 \vee \tilde{a}_1 = \tilde{a}_0, \tilde{a}_2 \prec \tilde{a}_0 \vee \tilde{a}_2 = \tilde{a}_0, 0 \prec \tilde{a}_3 \vee \tilde{a}_1 = 1, \tilde{a}_3 \prec a \vee \tilde{a}_3 = a, \tilde{a}_4 \prec \tilde{a}_5 \vee \tilde{a}_4 = \tilde{a}_5 \vee \tilde{a}_5 \prec \tilde{a}_4 \& \tilde{a}_2 \vee \tilde{a}_5 = \tilde{a}_4 \& \tilde{a}_2, \tilde{a}_5 \prec \tilde{a}_4 \vee \tilde{a}_2 = 1,$$
$$b \prec \tilde{a}_5 \vee b = \tilde{a}_5, \tilde{a}_4 \& \tilde{a}_6 \prec \tilde{a}_7 \vee \tilde{a}_4 \& \tilde{a}_6 = \tilde{a}_7, a \prec \tilde{a}_6 \vee a = \tilde{a}_6, \tilde{a}_7 \rightarrow \underbrace{a}_{\tilde{a}_8} \& \underbrace{b}_{\tilde{a}_9}\} \tag{18}$$

$$\begin{array}{llllllll}
S^{\phi} = \{\tilde{a}_0 \prec 1 & [1] & \tilde{a}_1 \prec \tilde{a}_0 \vee \tilde{a}_1 = \tilde{a}_0 & [2] & \tilde{a}_2 \prec \tilde{a}_0 \vee \tilde{a}_2 = \tilde{a}_0 & [3] & 0 \prec \tilde{a}_3 \vee \tilde{a}_1 = 1 & [4] \\
\tilde{a}_3 \prec a \vee \tilde{a}_3 = a & [5] & \tilde{a}_4 \prec \tilde{a}_5 \vee \tilde{a}_4 = \tilde{a}_5 \vee \tilde{a}_5 \prec \tilde{a}_4 \& \tilde{a}_2 \vee \tilde{a}_5 = \tilde{a}_4 \& \tilde{a}_2 & [6] & \tilde{a}_5 \prec \tilde{a}_4 \vee \tilde{a}_2 = 1 & [7] & b \prec \tilde{a}_5 \vee b = \tilde{a}_5 & [8] \\
\tilde{a}_4 \& \tilde{a}_6 \prec \tilde{a}_7 \vee \tilde{a}_4 \& \tilde{a}_6 = \tilde{a}_7 & [9] & a \prec \tilde{a}_6 \vee a = \tilde{a}_6 & [10] & \tilde{a}_7 \prec \tilde{a}_8 \& \tilde{a}_9 \vee \tilde{a}_7 = \tilde{a}_8 \& \tilde{a}_9 & [11] & \tilde{a}_8 \prec a \vee \tilde{a}_8 = a & [12] \\
\tilde{a}_9 \prec b \vee \tilde{a}_9 = b & [13]\} & & & & & &
\end{array}$$

augmented by additional admissible rules, which do not change the semantics of the *DPLL* procedure. For example, we can add a rule:

(*Tautology simplification rule*) (37)

$$\frac{T}{T - \{l \vee C\}}$$

$l \vee C \in T$, *l is a tautology.*

We can strengthen Rule (29), denoted as $(29)^{\#}$, by omitting the application condition: *T is a unit order clausal theory*. Such admissible rules are obviously sound and helpful for constructing more compact *DPLL* trees in many cases, however, superfluous for the completeness argument. Concerning the deduction problem of a formula from a finite theory, we conclude.

Corollary 4.3. *Let $\phi \in PropForm_{\emptyset}$ and $T \subseteq_{\mathcal{F}} PropForm_{\emptyset}$. There exist $A_T^{\phi} \subseteq_{\mathcal{F}} \tilde{\mathbb{A}}$, $S_T^{\phi} \subseteq_{\mathcal{F}} SimOrdCl_{A_T^{\phi}}$, a finite tree Tree with the root S_T^{ϕ} constructed using Rules (24)–(30) with the following properties:*

if $T \models_P \phi$, then Tree is closed; (38)

if $T \not\models \phi$, then Tree is open and there exists a partial model $\mathfrak{A}$ of T, $dom(\mathfrak{A}) = atoms(T, \phi)$, related to Tree such that $\mathfrak{A} \not\models \phi$. (39)

Proof. An immediate consequence of Theorems 3.2 and 4.2. $\square$

Let $\phi = a \rightarrow 0 \vee (a \rightarrow a \& b) \rightarrow b \in PropForm_{\emptyset}$, $a, b \in PropAtom_{\emptyset}$. Using Corollary 4.3, we show that ϕ is a tautology. At first, we translate ϕ to $S^{\phi} \subseteq_{\mathcal{F}} SimOrdCl$ in Table 6. Before we start *DPLL* derivation, it is suitable to investigate several cases when the input atoms a, b get the truth values 0, 1. Case 1: $\|a\| = 0$. Then $\|\phi\| = 1$. Case 2: $\|a\| = 1$. Then $\|\phi\| = \|b\| \Rightarrow \|b\| = 1$. Hence, in all the cases, $\|\phi\| = 1$, and it remains to investigate whether $\|\phi\| = 1$ for the case $0 \prec a$, $a \prec 1$, $0 \prec b$, $b \prec 1$ by the *DPLL* procedure.

Case 3: We add $0 \prec a$ [14], $a \prec 1$ [15], $0 \prec b$ [16], $b \prec 1$ [17]. Primarily using Rules (27) and (28), we can derive a branch in the constructed tree such that for all $i \leq 9$, $0 \prec \tilde{a}_i$, $\tilde{a}_i \prec 1$; the other branches are closed, ending in $\square$. We then lengthen this branch by deriving

$\tilde{a}_5 \prec \tilde{a}_4$	[18]	: [7]
$\tilde{a}_5 \prec \tilde{a}_4 \& \tilde{a}_2 \vee \tilde{a}_5 = \tilde{a}_4 \& \tilde{a}_2$	[19]	: [6] [18]
$b \prec \tilde{a}_4 \& \tilde{a}_2 \vee b = \tilde{a}_4 \& \tilde{a}_2$	[20]	: [19] [8]
$\tilde{a}_7 \prec a \& b \vee \tilde{a}_7 = a \& b$	[21]	: [11] [12] [13]
$\tilde{a}_4 \& a \prec \tilde{a}_7 \vee \tilde{a}_4 \& a = \tilde{a}_7$	[22]	: [9] [10]
$\tilde{a}_4 \prec b \vee \tilde{a}_4 = b$	[23]	: [22] [21]
$\square$	[24]	: [20] [23] $(29)^{\#}$; $\tilde{a}_2 \prec 1$.

Hence, all the cases-branches of the constructed tree are closed; we have reached $\square$ in all of them. We get the constructed tree by the *DPLL* procedure is closed.

So, we have proved $\emptyset \models_P \phi$ and ϕ is a tautology.

5 CONCLUSIONS

We have investigated the deduction problem of a formula from a finite theory in the propositional Product logic. The deduction problem has been solved via translation of a formula to an equivalent satisfiable finite order clausal theory, consisting of order clauses. An order clause is a finite set of order literals of the form $\varepsilon_1 \diamond \varepsilon_2$ where ε_i is either a conjunction of propositional atoms or the propositional constant *0* (false) or *1* (true), and $\diamond$ is a connective either $\eqcirc$ or $\prec$. $\eqcirc$ and $\prec$ are interpreted by the equality and standard strict order on $[0,1]$, respectively. The trichotomy over order literals: either $\varepsilon_1 \prec \varepsilon_2$ or $\varepsilon_1 \eqcirc \varepsilon_2$ or $\varepsilon_2 \prec \varepsilon_1$, has naturally led to a variant of the *DPLL* procedure with a trichotomy branching rule, which is refutation sound and complete in the finite case.

REFERENCES

Aguzzoli, S. and Ciabattoni, A. (2000). Finiteness in infinite-valued Łukasiewicz logic. *Journal of Logic, Language and Information*, 9(1):5–29.

Anderson, R. and Bledsoe, W. W. (1970). A linear format for resolution with merging and a new technique for establishing completeness. *J. ACM*, 17(3):525–534.

Beckert, B., Hähnle, R., and Manyà, F. (2000). The SAT problem of signed CNF formulas. In Basin, D., DAgostino, M., Gabbay, D., Matthews, S., and Viganò, L., editors, *Labelled Deduction*, volume 17 of *Applied Logic Series*, pages 59–80. Springer Netherlands.

Biere, A., Heule, M. J., van Maaren, H., and Walsh, T. (2009). *Handbook of Satisfiability*, volume 185 of *Frontiers in Artificial Intelligence and Applications*. IOS Press, Amsterdam.

Davis, M., Logemann, G., and Loveland, D. (1962). A machine program for theorem-proving. *Commun. ACM*, 5(7):394–397.

Davis, M. and Putnam, H. (1960). A computing procedure for quantification theory. *J. ACM*, 7(3):201–215.

de la Tour, T. B. (1992). An optimality result for clause form translation. *J. Symb. Comput.*, 14(4):283–302.

Guller, D. (2009). On the refutational completeness of signed binary resolution and hyperresolution. *Fuzzy Sets and Systems*, 160(8):1162 – 1176. Featured Issue: Formal Methods for Fuzzy Mathematics, Approximation and Reasoning, Part II.

Guller, D. (2010). A DPLL procedure for the propositional Gödel logic. In Filipe, J. and Kacprzyk, J., editors, *IJCCI (ICFC-ICNC)*, pages 31–42. SciTePress.

Guller, D. (2012). On the satisfiability and validity problems in the propositional Gödel logic. In Madani, K., Dourado Correia, A., Rosa, A., and Filipe, J., editors, *Computational Intelligence*, volume 399 of *Studies in Computational Intelligence*, pages 211–227. Springer Berlin / Heidelberg.

Hähnle, R. (1994a). Many-valued logic and mixed integer programming. *Ann. Math. Artif. Intell.*, 12(3-4):231–263.

Hähnle, R. (1994b). Short conjunctive normal forms in finitely valued logics. *J. Log. Comput.*, 4(6):905–927.

Hähnle, R. (1996). Exploiting data dependencies in many-valued logics. *Journal of Applied Non-Classical Logics*, 6(1):49–69.

Hähnle, R. (1997). Proof theory of many-valued logic–linear optimization–logic design: connections and interactions. *Soft Comput.*, 1(3):107–119.

Hájek, P., Godo, L., and Esteva, F. (1996). A complete many-valued logic with product-conjunction. *Arch. Math. Log.*, 35(3):191–208.

Manyà, F., Béjar, R., and Escalada-Imaz, G. (1998). The satisfiability problem in regular CNF-formulas. *Soft Comput.*, 2(3):116–123.

Metcalfe, G., Olivetti, N., and Gabbay, D. M. (2004). Analytic calculi for product logics. *Arch. Math. Log.*, 43(7):859–890.

Mundici, D. (1987). Satisfiability in many-valued sentential logic is NP-complete. *Theor. Comput. Sci.*, 52:145–153.

Nonnengart, A., Rock, G., and Weidenbach, C. (1998). On generating small clause normal forms. In Kirchner, C. and Kirchner, H., editors, *CADE*, volume 1421 of *Lecture Notes in Computer Science*, pages 397–411. Springer.

Plaisted, D. A. and Greenbaum, S. (1986). A structure-preserving clause form translation. *J. Symb. Comput.*, 2(3):293–304.

Savický, P., Cignoli, R., Esteva, F., Godo, L., and Noguera, C. (2006). On Product logic with truth-constants. *J. Log. Comput.*, 16(2):205–225.

Sheridan, D. (2004). The optimality of a fast CNF conversion and its use with SAT. In *SAT*.