A Bottom-Up Algorithm for t-Tautologies

Simone Bova
Department of Mathematics and Computer Science
University of Siena, Pian dei Mantellini 44, 53100 Siena, Italy
bova@unisi.it

Abstract. Fuzzy logics based on residuated t-norms provide a robust mathematical formalism for logical deduction under uncertain or vague premises. In this paper, we describe a decision algorithm for the tautology problem of Basic Logic, which is the logic of continuous t-norms and their residua [Háj98, CEGT00]. Our algorithm is a refinement of the semantic method of Baaz, Hájek, Montagna, and Veith [BHMV02].

1 Introduction

Imagine designing a family of propositional logics that satisfies the following list of requirements:

(i) the propositional variables: $p_1, p_2, \ldots,$ are interpreted over the real unit interval $[0, 1]$, linearly ordered by $\leq$ in the usual way (*fuzzyness*);

(ii) the logical symbols: $\perp$ (*falsum*), $\odot$ (fuzzy conjunction), and $\to$ (fuzzy implication), are respectively interpreted over the constant 0 and the binary functions $f_\odot$ and $f_\to$ on $[0, 1]$ (*truth functionality*);

(iii) $f_\odot$ is associative, commutative, monotone and continuous;

(iv) $f_\odot(x, 1) = x$ and $f_\to(x, y) = 1$ if and only if $x \leq y$, so that the restrictions of $f_\odot$ and $f_\to$ to $\{0, 1\}^2$ behave like Boolean conjunction and implication;

(v) the fuzzy modus ponens rule, $A \odot (A \to B) \vdash B$, is sound.

In this scenario, the pairs of operations known as *t-norms* and *residua* provide suitable interpretations for fuzzy conjunction and implication. Indeed, a continuous t-norm $*$ is a continuous binary function on $[0, 1]$ that is associative, commutative, monotone ($x \leq y$ implies $x * z \leq y * z$) and has 1 as unit ($x * 1 = x$). Given a continuous t-norm $*$, the associated

Proceedings of the Twelfth ESSLLI Student Session
Ville Nurmi and Dmitry Sustretov (editors)

residuum is the binary function on $[0,1]$ uniquely determined by the condition $x \rightarrow^* y = max\{z : x * z \leq y\}$. Notice that $x \leq y$ is equivalent to $x \rightarrow^* y = 1$, and implies $y \rightarrow^* z \leq x \rightarrow^* z$ and $z \rightarrow^* x \leq z \rightarrow^* y$. Notice also that the fuzzy modus ponens is sound, since by definition $x * (x \rightarrow^* y) \leq y$, and *powerful*, in the sense that the value of y is lower bounded by the maximal value of $x * (x \rightarrow^* y)$ which preserves the requirement of soundness.

Hence, a t-norm $*$ naturally determines a propositional fuzzy logic $\mathbf{L}^*$ satisfying requirements (i)-(v) above. Formally, let $[0,1]_* = ([0,1], *, \rightarrow^*, 0)$ be the algebra over $[0,1]$ equipped with the t-norm $*$ and its residuum $\rightarrow^*$. We call $[0,1]_*$ the *t-algebra* of $*$. Then, $\mathbf{L}^*$ is the propositional logic on the connectives $\odot, \rightarrow$ and the constant $\bot$ respectively interpreted on $[0,1]_*$ as $*$, $\rightarrow^*$ and 0 (over this basis, $\neg A$ and $\top$ are defineable via $A \rightarrow \bot$ and $\neg\bot$, respectively). The tautologies of $\mathbf{L}^*$ are the formulas evaluating to 1 on $[0,1]_*$ under any valuation of the variables in $[0,1]$.

Interestingly, the Hilbert calculus **BL** (*Hájek's Basic Logic*) given by the axioms:

(A1) $(A \rightarrow B) \rightarrow ((B \rightarrow C) \rightarrow (A \rightarrow C))$

(A2) $(A \odot B) \rightarrow A$

(A3) $(A \odot B) \rightarrow (B \odot A)$

(A4) $(A \odot (A \rightarrow B)) \rightarrow (B \odot (B \rightarrow A))$

(A5a) $((A \rightarrow (B \rightarrow C)) \rightarrow ((A \odot B) \rightarrow C))$

(A5b) $((A \odot B) \rightarrow C)) \rightarrow ((A \rightarrow (B \rightarrow C))$

(A6) $((A \rightarrow B) \rightarrow C) \rightarrow (((B \rightarrow A) \rightarrow C) \rightarrow C)$

(A7) $\bot \rightarrow A$

and the rule $A \odot (A \rightarrow B) \vdash B$, turns out to be the logic of *all* continuous t-norms and their residua. That is, $\mathbf{BL} \vdash A$ if and only if, for all continuous t-norms $*$, A is a tautology of $\mathbf{L}^*$ [Háj98, CEGT00]. In this light, we formalize the *t-tautology* problem as follows:

Problem: t-TAUT $= \{\langle A \rangle : \mathbf{BL} \vdash A\} \subseteq \{0,1\}^*$
Input: $\langle A \rangle \in \{0,1\}^*$
Output: 1 if and only if $\langle A \rangle \in t$-TAUT

where $\langle A \rangle \in \{0,1\}^*$ is a binary *encoding* of A of length polynomial in the *complexity* of A, $size(A)$, which is the number of connectives occurring in A. For technical reasons, we put $size(\top) = 0$.

As a stronger result, Aglianò and Montagna [AM03] have shown that $A \in t$-TAUT if and only if A is a tautology with respect to the interpretation of the propositional language into a special t-algebra, defined as follows.

Definition 1. *The* algebra $\omega[0,1]_{\text{Ł}}$ *is the algebra on the support* $[0,+\infty]$ *equipped with the operations* $*$, $\rightarrow^*$ *and the constants* 0, $+\infty$, *where* $*$ *and* $\rightarrow^*$ *are respectively defined by:*

$$x * y = \begin{cases} min\{x, y\} & \text{if } \lfloor x \rfloor \neq \lfloor y \rfloor \\ max\{\lfloor x \rfloor, x + y - \lfloor x \rfloor - 1\} & \text{if } \lfloor x \rfloor = \lfloor y \rfloor < +\infty \\ +\infty & \text{if } x = y = +\infty \end{cases}$$

$$x \rightarrow^* y = \begin{cases} y & \text{if } \lfloor y \rfloor < \lfloor x \rfloor \\ \lfloor x \rfloor + 1 - x + y & \text{if } \lfloor x \rfloor = \lfloor y \rfloor \text{ and } y < x \\ +\infty & \text{if } x \leq y \end{cases}$$

where $\lfloor x \rfloor$ *is the integer part of* x *and* $\lfloor +\infty \rfloor = +\infty$.

A valuation *(of the propositional language into* $\omega[0,1]_{\text{Ł}}$*) is a function* v *such that* $v(\bot) = 0$, $v(p_i) \in [0,+\infty]$ *for all* $i \in \mathbf{N}$, $v(A \odot B) = v(A) * v(B)$ *and* $v(A \rightarrow B) = v(A) \rightarrow^* v(B)$.

Theorem 1 (Aglianò and Montagna, 2003)**.** $A \in$ *t-TAUT if and only if* $v(A) = v(\top)$ *for every valuation* v.

The interpretation $\omega[0,1]_{\text{Ł}}$ allows to show that the complement of *t*-TAUT is (complete for) $\mathcal{NP}$, and, as a consequence, that *t*-TAUT is decidable [BHMV02]. Hence, it is natural to investigate decision algorithms for *t*-TAUT. In this paper, we present an algorithm, called Bottom-Up-BL, which is a refinement of the semantic method of Baaz, Hájek, Montagna, and Veith (BHMV-BL, in the sequel).

2 A Bottom-Up Algorithm for *t*-Tautologies

The present section introduces Bottom-Up-BL. After presenting the basic idea, patterned after BHMV-BL (Subsection 2.1), we describe in detail how Bottom-Up-BL works (Subsection 2.2), and we provide an example (Subsection 2.3). The main result of the paper is that Bottom-Up-BL is correct for *t*-TAUT (Subsection 2.4). Not surprisingly, the worst case running time of the algorithm is $\exp(n^{O(1)})$, where $O(n)$ bounds above the size of the input. For background on algorithms, we refer the reader to [CLRS01].

2.1 Idea

Any valuation v determines a total *order* $\leq_A$ over the subformulas of A (plus $\top$), stipulating that $B_1 \leq_A B_2$ if and only if $v(B_1) \leq v(B_2)$, for B_1, B_2 subformulas of A. Such an order satisfies either $A <_A \top$ or $A =_A \top$. However, there exist total orders of the subformulas of A (plus $\top$) not corresponding to any valuation v. We call the former orders *consistent*, and the latter

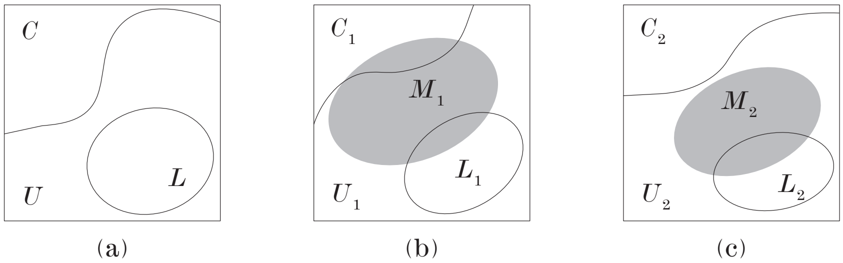

Figure 1.1: (a) depicts the partition (given a formula A) of the set of orders into consistent orders, C, inconsistent orders, U, and locally inconsistent orders, $L \subseteq U$. By definition, $C, U \neq \emptyset$. BHMV-BL searches $C \cup U$, while BOTTOM-UP-BL searches $(C \cup U) \setminus L$. (b) and (c) depict the search spaces of BOTTOM-UP-BL, given in input (distinct) formulas A_1 and A_2. The gray regions, M_1 and M_2, are the set of the orders where $A_1 <_{A_1} \top$ and $A_2 <_{A_2} \top$ respectively. In the first case the output is 0 ($M_1 \cap C_1 \neq \emptyset$), in the second case the output is 1 ($M_2 \cap C_2 = \emptyset$.)

orders *inconsistent*, respectively, sets C and U in Figure 1.1(a). As an example, if $B_1, B_2, B_1 \to B_2, B_1 \odot B_2$ are subformulas of A, any order where $B_1 \to B_2 <_A B_2$ or $B_1 <_A B_1 \odot B_2$ is inconsistent, observed that any valuation v satisfies both $v(B_2) \leq v(B_1 \to B_2)$ and $v(B_1 \odot B_2) \leq v(B_1)$. Now, the important consequence of Theorem 1 is that the semantic consistency of a given order is computable in polynomial time. Hence, since $C \cup U$ is finite, the algorithm BHMV-BL can check exhaustively *all* the orders for semantic consistency; the output will be 1 if and only if all consistent orders satisfy $A =_A \top$ (equivalently, all the orders satisfying $A <_A \top$ are inconsistent). Our simple observation is that BHMV-BL approach allows for the following refinement: if we construct the orders inductively on the complexity of the subformulas of A, starting from all the orders of the *variables* of A, we can *immediately* detect some inconsistencies (applying Fact 1, see below), and therefore we can avoid the computation of a certain number of inconsistent orders (the set $L \subseteq U$ of *locally* inconsistent orders in Figure 1.1(a)), improving the effectiveness of the computation (we guess that L is large).

More precisely, let A be a formula and S be the set of the subformulas of A. Any valuation v determines a partition of $S \cup \{\top\}$ into $h = |H|$ blocks, where $H = \{\lfloor v(B) \rfloor : B \in S \cup \{\top\}\}$. Let $b_1 < \cdots < b_{h-1} < +\infty$ be the natural total order over H, let $I = \{\bot_1, \ldots, \bot_h\}$ be a set of fresh constant symbols (*idempotents*, in the sequel) and put $v(\bot_j) = b_j$ for all $1 \leq j < h$ and $v(\bot_h) = +\infty$. Now, v determines a total order $\leq_A$ over $S \cup \{\top\} \cup I$, stipulating that, for every pair B_1, B_2 of formulas in $S \cup \{\top\} \cup I$, $B_1 \leq_A B_2$ if and only if $v(B_1) \leq v(B_2)$.

Notation 1. *Let $B_1, B_2 \in S$. In the sequel, $B_1 =_A B_2$ is for $B_1 \leq_A B_2$ and $B_2 \leq_A B_1$, and $B_1 <_A B_2$ is for $B_1 \leq_A B_2$ and $B_1 \neq_A B_2$. Also, if there exists $j \leq h$ such that $B_1 <_A \bot_j \leq_A B_2$, we write $B_1 \ll B_2$; if there exists*

$j < h$ such that $\bot_j \leq_A B_1 <_A B_2 <_A \bot_{j+1}$, we write $B_1 \prec B_2$; if there exists $j < h$ such that $\bot_j \leq_A B_1 \leq_A B_2 <_A \bot_{j+1}$, we write $B_1 \preccurlyeq B_2$.

As before, we say that an order $\leq_A$ over $S \cup \{\top\} \cup I$ is *consistent* if and only if it corresponds to a valuation. Some inconsistencies follow immediately from Definition 1.

Fact 1. *Let A, S and I as above, and let $\leq_A$ be any total order over $S \cup \{\top\} \cup I$. Then, $\leq_A$ is consistent only if it satisfies all the following statements ($B_1, B_2, B_1 \odot B_2, B_1 \to B_2, C_1, C_2, C_1 \odot C_2, C_1 \to C_2 \in S$):*

(i) *If $B_1 \ll B_2$ or $\bot_j =_A B_1 \leq_A B_2$ ($j \leq h$), then $B_1 \odot B_2 =_A B_1$.*

(ii) *If $B_1 \preccurlyeq B_2$, then $B_1 \odot B_2 \prec B_1$.*

(iii) *If $B_1 \leq_A C_1$ and $B_2 \leq_A C_2$, then $B_1 \odot B_2 \leq_A C_1 \odot C_2$. If in addition $\bot_j <_A C_1 \odot C_2$ and $B_1 <_A C_1$ or $B_2 <_A C_2$, then $B_1 \odot B_2 <_A C_1 \odot C_2$.*

(iv) *If $B_1 \leq_A B_2$, then $B_1 \to B_2 =_A \top$.*

(v) *If $B_2 \ll B_1$, then $B_1 \to B_2 =_A B_2$.*

(vi) *If $B_2 \prec B_1$, then $B_2 \prec B_1 \to B_2$.*

(vii) *If $B_1 \leq_A C_1$ and $C_2 \leq_A B_2$, then $C_1 \to C_2 \leq_A B_1 \to B_2$. If in addition $B_1 <_A C_1$ or $C_2 <_A B_2$, then $C_1 \to C_2 <_A B_1 \to B_2$.*

We insist that the condition above is necessary, but not sufficient (in general the inclusion $L \subseteq U$ in Figure 1.1(a) is strict). The idea beyond Bottom-Up-BL is to exploit systematically Fact 1 to reduce the search space, avoiding the computation of locally inconsistent orders (compare the description of the iteration step given in Subsection 2.2).

2.2 Algorithm

We describe in detail the algorithm Bottom-Up-BL, commenting on the pseudocode below. The input to the algorithm is a formula A, where $a_1, \ldots, a_k$ are the *atoms* (subformulas of complexity 0) of A and $size(A) = n$. Notice that, if $size(A) = n$, then the variables of A are at most $n + 1$.

Bottom-Up-BL($\langle A \rangle$)
for $h \leftarrow 2$ **to** $k + 1$
    $o_A \leftarrow \bot_1 <_A \cdots <_A \bot_h =_A \top$
    $w_A(\bot_j)_{1 \leq j < h} = 0$, $w_A(\bot_h) = w_A(\top) = 1$
    $p_A \leftarrow \emptyset$
    $ORD_h \leftarrow \{(o_A, w_A, p_A)\}$
    for $i \leftarrow 0$ **to** n
        $S \leftarrow \{E : E \text{ subformula of } A, size(E) = i\}$

```
8        ORD_i ← extensions of ORD_{i−1} to S not excluded by Fact 1
9      ▷ Let ORD_n = {(o_{A,1}, w_{A,1}, p_{A,1}), ..., (o_{A,l}, w_{A,l}, p_{A,l})}.
10     if (∃1 ≤ m ≤ l) A <_{A,m} ⊤ holds and p_{A,m} is feasible
11        output 0
12   output 1
```

Let $2 \leq h \leq k+1$, and let S, I be as above. Let $S_{-1} = \{\top\} \cup I$ and $S_i = \{E : E \in S \cup \{\top\} \cup I, size(E) \leq i\} \subseteq S$, for $i = 0, \ldots, n$. In the sequel, for all $-1 \leq i \leq n$, ORD_i is a finite set of triples of the form (o_A, w_A, p_A), where:

(i) o_A (*order*, in the sequel) is the union of a relation $E_1 <_A E_2$ satisfying irreflexivity, antisymmetry and transitivity over S_i, and a relation $E_1 =_A E_2$ satisfying reflexivity, symmetry and transitivity over S_i, with the (technical) exception to the symmetry of $=_A$ that there are not $E \in S_i \setminus I$ and $\bot_j \in I$ satisfying $E =_A \bot_j$. Moreover, o_A satisfies the chain $E_1 \triangleleft_1 \cdots \triangleleft_{|S_i|-1} E_{|S_i|}$, where $S_i = \bigcup_{p=1}^{|S_i|} E_p$ and $\triangleleft_p \in \{<_A, =_A\}$ for all $p = 1, \ldots, |S_i| - 1$. In the sequel, given an order o_A, $\leq_A$ is for $<_A$ or $=_A$ and, if B_1 and B_2 are subformulas of A, $min_A\{B_1, B_2\}$ is B_1 if $B_1 \leq_A B_2$ and B_2 otherwise.

(ii) w_A is a linear function over S_i.

(iii) p_A is a set of linear equality and inequality constraints with integer coefficients over unknowns (among) $x_1, \ldots, x_n, x_{n+1}$.

In lines 2-5 the algorithm settles ORD_h ($2 \leq h \leq k+1$) to the triple (o_A, w_A, p_A), where $o_A = \{\bot_1 <_A \bot_2, \ldots, \bot_{h-1} <_A \bot_h\} \cup \{\bot_h =_A \top\}$, $w_A(\bot_j) = 0$, for $1 \leq j < h$, $w_A(\bot_h) = w_A(\top) = 1$ and $p_A = \emptyset$.

Now, let ORD_h be fixed. The *main loop* of Bottom-Up-BL spans lines 6-8. The ith iteration of the loop ($0 \leq i \leq n$) is aimed to extend the triples in ORD_{i-1} to all subformulas of A of complexity less than or equal to i (stipulate that ORD_{-1} is ORD_h). In the description below, we consider several possible ways of extending each triple in ORD_{i-1}, and we assume that the algorithm put every extension considered in ORD_i.

Initialization (Step $i = 0$). For each $(o_A, w_A, p_A) \in ORD_h$, the order o_A is extended to the atoms $a_1, \ldots, a_k$ of A in such a way that: (i) If $\bot$ is an atom of A, then $\bot_1 =_A \bot$ holds in the extended order. (ii) For each $1 \leq j < h$, there exists an atom a of A such that $\bot_j \leq_A a <_A \bot_{j+1}$ holds in the extended order. (iii) There not exists a variable p of A such that $p <_A \bot_1$ or $\bot_h <_A p$ holds in the extended order. Notice that $\bot_j =_A p$ can hold in the extended order, but $p =_A \bot_j$ can not hold because of the previous stipulation on $=_A$, for $1 \leq j \leq h$. There are several possible ways of extending o_A to the atoms. For each of such choices, w_A and p_A are extended as follows. As regards to w_A: if a is $\bot$, then $w_A(a) = 0$, otherwise, if a is a variable p_i, then $w_A(a) = x_i$ ($i \geq 1$). As regards to p_A: (*i*) For each pair p_i, p_j such that $p_i =_A p_j$, the constraint $x_i = x_j$ is added to p_A. (*ii*) For each pair p_i, p_j such that $p_i <_A p_j$, the constraint $x_i < x_j$ is added to p_A. (*iii*) For each p_i such that $\bot_j =_A p_i$, the constraint $x_i = 0$ is added to p_A if $j < h$, otherwise the constraint $x_i = 1$ is added to p_A. (*iv*) For each p_i such that $\bot_j <_A p_i$, the constraint $0 < x_i$ is added to p_A. (*v*) For each p_i such that $p_i <_A \bot_j$, the constraint $x_i < 1$ is added to p_A.

Iteration (Step $i+1$, $i \geq 0$). Let S_i, S_{i+1}, ORD_i and ORD_{i+1} be determined as above. At iteration $i+1$, the algorithm computes ORD_{i+1}, given ORD_i. Each triple in ORD_{i+1} is the result of the extension of a triple in ORD_i to all the subformulas of A of complexity $i+1$. There are several possibilities to extend an order $o_{A,i}$ over S_i to an order $o_{A,i+1}$ over S_{i+1}. Among all, the algorithm computes only the extensions considered below ($w_{A,i}$ and $p_{A,i}$ are extended accordingly). For a fixed $(o_A, w_A, p_A) \in ORD_i$ and a fixed subformula E of A of complexity $i+1$, Bottom-Up-BL works as follows. If E has the form $B_1 \odot B_2$, then:

($\odot_1$) If $B_1 \ll B_2$ or $\bot_j =_A B_1 \leq_A B_2$ for some $j \leq h$, then o_A is extended to $B_1 \odot B_2$ by adding $B_1 \odot B_2 =_A B_1$. Also, $w_A(B_1 \odot B_2) = w_A(B_1)$ is settled, and no constraint is added to p_A.

($\odot_2$) If $B_2 \ll B_1$ or $\bot_j =_A B_2 \leq_A B_1$ for some $j \leq h$, then o_A is extended to $B_1 \odot B_2$ by adding $B_1 \odot B_2 =_A B_2$. Also, $w_A(B_1 \odot B_2) = w_A(B_2)$ is settled, and no constraint is added to p_A.

($\odot_3$) Otherwise, let $j < h$ be maximal such that $\bot_j <_A B_1 <_A \bot_{j+1}$ and $\bot_j <_A B_2 <_A \bot_{j+1}$. Then, o_A is extended to $B_1 \odot B_2$ in such a way that: (*i*) $\bot_j \leq_A B_1 \odot B_2 <_A \min_A\{B_1, B_2\} <_A \bot_{j+1}$ holds in the extended order. (*ii*) For any pair C_1, C_2 of formulas, if $B_1 \leq_A C_1$, $B_2 \leq_A C_2$ and $C_1 \odot C_2$ has already been added in o_A, then $B_1 \odot B_2 \leq_A C_1 \odot C_2$ holds in the extended order. Moreover, if $\bot_j <_A C_1 \odot C_2$ ($j < h$), $B_1 \leq_A C_1$, $B_2 \leq_A C_2$ and at least one of the last two inequalities is strict, then $B_1 \odot B_2 <_A C_1 \odot C_2$ holds in the extended order. (*iii*) For any pair C_1, C_2 of formulas, if $C_1 \leq_A B_1$, $C_2 \leq_A B_2$ and $C_1 \odot C_2$ has already been added in o_A, then $C_1 \odot C_2 \leq_A B_1 \odot B_2$ holds in the extended order. Moreover, if $\bot_j <_A C_1 \odot C_2$ ($j < h$), $C_1 \leq_A B_1$, $C_2 \leq_A B_2$ and at least one of the last two inequalities is strict, then $C_1 \odot C_2 <_A B_1 \odot B_2$ holds in the extended order. There are several possible ways of extending o_A to $B_1 \odot B_2$ satisfying the conditions above. For each choice, w_A and p_A are extended accordingly, as follows. As regards to w_A, if $\bot_j =_A B_1 \odot B_2$ for some $j < h$, then $w_A(B_1 \odot B_2) = 0$ is settled, otherwise $w_A(B_1 \odot B_2) = w_A(B_1) + w_A(B_2) - 1$ is settled. As regards to p_A: (*i*) If $\bot_j =_A B_1 \odot B_2$ for some $j < h$, then the constraint $w_A(B_1)+w_A(B_2) \leq 1$ is added to p_A. *(ii)* If $\bot_j <_A B_1 \odot B_2$ for some $j < h$, then the constraint $1 < w_A(B_1) + w_A(B_2)$ is added to p_A. Also, if $B_1 \odot B_2 =_A D$ for some formula D already added to o_A, the constraint $w_A(B_1 \odot B_2) = w_A(D)$ is added to p_A; otherwise, if the formulas D_1, D_2, already added to o_A, are respectively maximal such that $D_1 <_A B_1 \odot B_2$ and minimal such that $B_1 \odot B_2 <_A D_2$, the constraints $w_A(D_1) < w_A(B_1 \odot B_2)$ and $w_A(B_1 \odot B_2) < w_A(D_2)$ are added to p_A.

If E has the form $B_1 \to B_2$, then:

($\to_1$) If $B_1 \leq_A B_2$, then o_A is extended to $B_1 \to B_2$ by adding $\bot_h =_A \top =_A B_1 \to B_2$. Also, $w_A(B_1 \to B_2) = 1$ is settled, and no constraint is added to p_A.

($\to_2$) If $B_2 \ll B_1$, then then o_A is extended to $B_1 \to B_2$ by adding $B_1 \to B_2 =_A B_2$. Also, $w_A(B_1 \to B_2) = w_A(B_2)$ is settled, and no constraint is added to p_A.

($\to_3$) If $B_2 \prec B_1$, then let $j < h$ be maximal such that $\bot_j \leq_A B_2$. Then, o_A is extended to $B_1 \to B_2$ in such a way that: *(i)* $\bot_j \leq_A B_2 <_A B_1 \to B_2 <_A \bot_{j+1}$ holds in the extended order. *(ii)* For any pair C_1, C_2 of formulas, if $B_1 \leq_A C_1$, $C_2 \leq_A B_2$ and $C_1 \to C_2$ has already been added in o_A, then $C_1 \to C_2 \leq_A B_1 \to B_2$ holds in the extended order. Moreover, if at least one of the above two inequalities is strict, then $C_1 \to C_2 <_A B_1 \to B_2$ holds in the extended order. *(iii)* For any pair C_1, C_2 of formulas, if $C_1 \leq_A B_1$, $B_2 \leq_A C_2$ and $C_1 \to C_2$ has already been added in o_A, then $B_1 \to B_2 \leq_A C_1 \to C_2$ holds in the extended order. Moreover, if at least one of the above two inequalities is strict, then $B_1 \to B_2 <_A C_1 \to C_2$ holds in the extended order. Again, there are several possible ways of extending o_A to $B_1 \odot B_2$ satisfying the conditions above. For each choice, w_A and p_A are extended accordingly, as follows. As regards to w_A, $w_A(B_1 \to B_2) = w_A(B_2)+1-w_A(B_1)$ is settled. As regards to p_A: *(i)* If $B_1 \to B_2 =_A D$ for some formula D already added to o_A, the constraint $w_A(B_1 \to B_2) = w_A(D)$ is added to p_A. *(ii)* Otherwise, let the formulas D_1, D_2, already added to o_A, be respectively maximal such that $D_1 <_A B_1 \to B_2$ and minimal such that $B_1 \to B_2 <_A D_2$. Then, if $D_2 <_A \bot_{j+1}$, the constraint $w_A(D_1) < w_A(B_1 \odot B_2)$ and $w_A(B_1 \odot B_2) < w_A(D_2)$ are added to p_A; otherwise, if $\bot_{j+1} =_A D_2$, only the constraint $w_A(D_1) < w_A(B_1 \odot B_2)$ is added to p_A.

Termination (Step $i = n$). The number of orders of the form $B_1 \leq_A \cdots \leq_A B_{(k+n)+1+h}$, where each B_i is a distinct formula in $S \cup \{\top\} \cup I$ is clearly finite. The main loop of the algorithm computes a (proper) subset of these orders for every fixed h, so it terminates for every formula A. Let ORD_n be the set computed at termination of the main loop, for some $2 \leq h \leq k+1$. For each triple $(o_A, w_A, p_A) \in ORD_n$, the order o_A contains $\top$ and all the subformulas of A, including A itself. If there exists a triple $(o_A, w_A, p_A) \in ORD_n$ such that $A <_A \top$ holds in o_A and p_A is feasible (line 10), then Bottom-Up-BL breaks the external loop and outputs 0 (line 11). Otherwise, Bottom-Up-BL iterates the external loop if $h \leq k$ (line 1), or outputs 1 if $h > k$ (line 12).

2.3 Example

Let A be $((p_1 \to \bot) \to \bot) \to p_1$. The atoms of A are p_1 and $\bot$ and the subformulas of A excluding atoms, ordered by increasing complexity, are $p_1 \to \bot$, $(p_1 \to \bot) \to \bot$ and A. For $h = 2$, we have $ORD_h = \{(o_A, w_A, p_A)\}$ where $o_A \rightleftharpoons \bot_1 <_{A,1} \bot_2 =_A \top$. At step $i = 0$, we have $ORD_0 = \{(o_{A,1}, w_{A,1}, p_{A,1}), (o_{A,2}, w_{A,2}, p_{A,2})\}$ where $o_{A,1} \rightleftharpoons \bot_1 =_{A,1} \bot =_{A,1} p_1 <_{A,1} \bot_2 =_{A,1} \top$, and $o_{A,2} \rightleftharpoons \bot_1 =_{A,2} \bot <_{A,2} \bot_2 =_{A,2} \top =_{A,2} p_1$. For $h = 3$, we have $ORD_h = \{(o_A, w_A, p_A)\}$ where $o_A \rightleftharpoons \bot_1 <_A \bot_2 <_A \bot_3 =_A \top$. At step $i = 0$, we have $ORD_0 = \{(o_{A,3}, w_{A,3}, p_{A,3}), (o_{A,4}, w_{A,4}, p_{A,4})\}$ where $o_{A,3} \rightleftharpoons \bot_1 =_{A,3} \bot <_{A,3} \bot_2 =_{A,3} p_1 <_{A,3} \bot_3 =_{A,3} \top$, and $o_{A,4} \rightleftharpoons \bot_1 =_{A,4} \bot <_{A,4} \bot_2 <_{A,4} p_1 <_{A,4} \bot_3 =_{A,4} \top$. Now, for $h = 3$ and $i = 1, 2, 3$, Bottom-Up-BL computes, among the other possibilities, the following extension of $(o_{A,4}, w_{A,4}, p_{A,4})$ above, where $w_{A,4}(\bot) = w_{A,4}(\bot_1) = w_{A,4}(\bot_2) = 0$, $w_{A,4}(p_1) = x_1$, $w_{A,4}(\bot_3) = w_{A,4}(\top) = 1$, and $p_{A,4} = \{0 < x_1, x_1 < 1\}$. By $(\to_2)$, subformula $p_1 \to \bot$ adds $p_1 \to \bot =_{A,4} \bot$ and settles $w_{A,4}(p_1 \to \bot) = w_{A,4}(\bot) = 0$ ($p_{A,4}$ is unchanged). By $(\to_1)$, subformula $(p_1 \to \bot) \to \bot$ adds $\top =_{A,4} (p_1 \to \bot) \to \bot$ and settles $w_{A,4}((p_1 \to \bot) \to \bot) = w_{A,4}(\top) = 1$ ($p_{A,4}$ is unchanged). By $(\to_2)$, A adds $A =_{A,4} p_1 <_{A,4} \top$ and settles $w_{A,4}(A) = w_{A,4}(p_1) = x_1$ ($p_{A,4}$ is unchanged). Hence, at termination, $A <_{A,4} \top$ holds in $o_{A,4}$ and $p_{A,4}$ is feasible (any real number in $(0, 1)$ is a solution to $p_{A,4}$), and Bottom-Up-BL outputs 0.

2.4 Correctness

Bottom-Up-BL is *sound* and *complete* for t-TAUT. Formally,

Theorem 2. *Let A be a formula. Then, $\langle A \rangle \in t$-TAUT if and only if* Bottom-Up-BL *outputs* 1 *on input* $\langle A \rangle$.

The proof stems from the following correspondence between classes of valuations and triples (o_A, w_A, p_A) with feasible p_A's computed by Bottom-Up-BL. On the one hand, let (o_A, w_A, p_A) be a triple computed by Bottom-Up-BL, where p_A is feasible. Let $\mathbf{b} = b_1 < \cdots < b_j < \cdots < b_{h-1}$ be any linear order of $h - 1$ nonnegative integers and let $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_n, \mathbf{x}_{n+1})$ be any solution to p_A (there are several possible choices). Then, the valuation

v corresponding to (o_A, w_A, p_A) under $\mathbf{b}$ and $\mathbf{x}$ is such that, for $1 \le i \le n+1$: if $\bot_h =_A p_i$, then $v(p_i) = +\infty$; otherwise, if $j < h$ is maximal such that $\bot_j \le_A p_i <_A \bot_{j+1}$, then $\lfloor v(p_i) \rfloor = b_j$ and $v(p_i) - \lfloor v(p_i) \rfloor = \mathbf{x}_i$. For definiteness, put $v(p_i) = 0$ for all $i > n+1$. On the other hand, let v be a valuation and let $v_0, \dots, v_n$ be the restrictions of v to the subformulas of A of complexity $\le 0, \dots, \le n$ respectively. Also, let $H = \{\lfloor v(a_i) \rfloor : 1 \le i \le k\}$, $h = |H| + 1$ and $b_1 < \cdots < b_j < \cdots < b_{h-1}$ be the natural total order of H. Now, extend v to $\bot_1, \dots, \bot_h, \top$ by putting $v(\bot_h) = v(\top) = +\infty$ and $v(\bot_j) = b_j$ for $1 \le j \le h-1$. Then, for $i = 0, \dots, n$ the triple $(o_{A,i}, w_{A,i}, p_{A,i})$ corresponding to v_i can be computed mimicking iterations from 0 to i of Bottom-Up-BL main loop with h settled as above (the case $i = n$ gives the triple corresponding to the valuation v): $o_{A,i}$ is settled to the order $\le_{A,i}$ determined by valuation v_i; $w_{A,i}$ and $p_{A,i}$ are settled in such a way that clauses $(\odot_1)$, $(\odot_2)$, $(\odot_3)$ and $(\to_1)$, $(\to_2)$, $(\to_3)$ are satisfied, with respect to the order $o_{A,i}$. Such a correspondence owns the following key property.

Fact 2. *If a valuation v corresponds to a triple (o_A, w_A, p_A) computed by* Bottom-Up-BL *such that $A <_A \top$ holds in o_A and p_A is feasible, then $v(A) < v(\top)$. Conversely, if a triple (o_A, w_A, p_A) computed by* Bottom-Up-BL *corresponds to a valuation v such that $v(A) < v(\top)$, then $A <_A \top$ holds in o_A and p_A is feasible.*

3 Conclusion

In this paper, we refined the decision algorithm for t-tautologies of Baaz, Hájek, Montagna, and Veith [BHMV02]. Specifically, we exploited an inductive construction to avoid the brute force computation of all the orders of the subformulas of the input formula. We mention two natural developments of the present work.

From the complexity point of view, it would be interesting to investigate the existence of a class $\mathcal{F}$ of formulas such that the set of locally inconsistent orders of any $A \in \mathcal{F}$ is provably *large*. Indeed, any $A \in \mathcal{F}$ would be *easy* for Bottom-Up-BL, but still *hard* for BHMV-BL. From the algorithmic point of view, it would be interesting to formalize a *top-down* refinement of BHMV-BL, patterned after the logical calculus presented in [BM07], and to compare its performances against those of the bottom-up refinement presented in this paper. In particular, [BM07] implies that the bound $\exp(n^{O(1)})$ can be improved to $\exp(3n/2)$, where $O(n)$ bounds above the size of the input.

Acknowledgments. The author thanks his supervisor Franco Montagna for stimulating discussions on the subject, and the anonymous referees for careful comments on the first version of this paper.

Bibliography

[AM03] P. Aglianò and F. Montagna. Varieties of BL-Algebras I: General Properties. *Journal of Pure and Applied Algebra*, 181:105–129, 2003.

[BHMV02] M. Baaz, P. Hájek, F. Montagna, and H. Veith. Complexity of *t*-Tautologies. *Annals of Pure and Applied Logic*, 113:3–11, 2002.

[BM07] S. Bova and F. Montagna. Proof Search in Hájek's Basic Logic. *ACM Transactions on Computational Logic*, 2007. To appear.

[CEGT00] R. Cignoli, F. Esteva, L. Godo, and A. Torrens. Basic Fuzzy Logic is the Logic of Continuous t-Norms and their Residua. *Soft Computing*, 4(2):106–112, 2000.

[CLRS01] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press, Cambridge, MA, 2001.

[Háj98] P. Hájek. *Metamathematics of Fuzzy Logic*. Kluwer Academic Publishers, Dordrecht, 1998.