Artificial Intelligence Research and Development
I. Sanz et al. (Eds.)

doi:10.3233/FAIA230706

A Complete Tableau Calculus for the Regular MaxSAT Problem

Jordi COLL, [a] Chu-Min LI, [b,c] Felip MANYÀ [a] and Elifnaz YANGIN [a]
[a] *Artificial Intelligence Research Institute (IIIA), CSIC, Bellaterra, Spain*
[b] *Aix Marseille Univ, Université de Toulon, CNRS, LIS, Marseille, France*
[c] *MIS, Université de Picardie, Amiens, France*

Abstract. We define a tableau calculus for solving the Maximum Satisfiability problem of regular propositional logic (Regular MaxSAT). Given a multiset of regular clauses Φ, we prove that the calculus is sound in the sense that if the minimum number of contradictions derived among the branches of a completed tableau for Φ is m, then the minimum number of unsatisfied clauses in Φ is m. We also prove that it is complete in the sense that if the minimum number of unsatisfied clauses in Φ is m, then the minimum number of contradictions among the branches of any completed tableau for Φ is m. Furthermore, we describe how to extend the proposed calculus to solve Regular MaxSAT in the case where we consider weighted formulas.

Keywords. regular propositional logic, maximum satisfiability, semantic tableaux, completeness.

1. Introduction

Regular propositional logic is a multiple-valued logical formalism for knowledge representation that lies in the intersection of the areas of constraint programming, many-valued logics and annotated logic programming [6].

Regular propositional formulas are multiple-valued propositional formulas equipped with a regular sign. Given a finite truth value set N equipped with a total order $\leq$, a regular sign is a subset of N either of the form $\{j \in N \mid j \geq i\}$, denoted by $\geq i$, or $\{j \in N \mid j \leq i\}$, denoted by $\leq i$, for some $i \in N$. For simplicity, we assume that N takes rational values between 0 and 1. Given a multiple-valued propositional formula ϕ, a regular propositional formula is an expression either of the form $\geq i : \phi$ or $\leq i : \phi$. In Regular logic, a truth assignment v maps every propositional variable to a value of N. An assignment v is extended to regular propositional formulas by interpreting conjunction as the minimum function, disjunction as the maximum function and negation of a propositional formula ϕ as $1 - v(\phi)$. Then, an assignment v satisfies a regular propositional formula $\geq i : \phi$ iff $v(\phi) \geq i$, and v satisfies a regular propositional formula $\leq i : \phi$ iff $v(\phi) \leq i$.

In contrast to signed propositional formulas [6], where a sign can be any subset of N, regular propositional formulas offer advantages such as the distinction between positive and negative literals. In this paper, we assume that N is finite, but in the case that N is infinite, regular signs allow us to deal with infinite subsets of truth values.

In the area of satisfiability testing, one of the problems that has attracted more interest in recent years is the Maximum Satisfiability problem (MaxSAT) [5,15,16]. SAT is usually considered in the particular case of formulas in Conjunctive Normal Form (CNF), i.e. a problem instance consists of a conjunctions of clauses, where a clause is a disjunction of literals. Whereas SAT is the problem of deciding if there exists a truth assignment for a given Boolean CNF formula that satisfies all clauses, MaxSAT is the problem of finding a truth assignment that minimizes the number of unsatisfied clauses in a Boolean CNF formula.

In practice, SAT is used as a generic problem solving formalism for decision problems and MaxSAT for optimization problems. The development of highly competitive MaxSAT solvers (e.g. [5,14]) has allowed to apply MaxSAT to solve challenging optimization problems in various fields such as bioinformatics [22], circuit design and debugging [23], combinatorial testing [3], diagnosis [10], planning [24], scheduling [7] and team formation [21].

In this paper we focus on the MaxSAT problem for regular propositional formulas in CNF, or Regular MaxSAT. Namely, the problem is to find a truth assignment that satisfies the largest possible number of regular clauses from a given multiset. Our aim is to define a complete tableau-style proof system for Regular MaxSAT.

Our work is motivated by the fact that the logic machinery defined for Regular SAT is not valid for Regular MaxSAT. Unfortunately, the inference rules for Regular SAT are unsound in Regular MaxSAT because they preserve satisfiability but do not preserve the minimum number of unsatisfied clauses between the premises and the conclusions. As a consequence, in the Boolean case, new complete resolution and tableau-style proof systems for MaxSAT have had to be defined (see e.g. [4,8,9,11,12,13,17,18]). In the multiple-valued case, there exist also resolution and tableau-style proof systems for MaxSAT (see e.g. [1,2,19,20]). Nevertheless, specific proof systems for Regular MaxSAT have not been investigated so far.

The main contributions of the present paper can be summarized as follows:

- The definition of the first tableau-style proof system for Regular MaxSAT and the corresponding proofs of soundness and completeness.
- The extension of the proposed proof system for dealing with clauses that have an associated weight, or in other words, a sound and complete proof system for Weighted Regular MaxSAT.

The paper is structured as follows. Section 2 defines basic concepts. Section 3 defines a Regular MaxSAT tableau calculus and proves its soundness and completeness. Section 4 defines an extension of the proposed calculus to Weighted MaxSAT. Section 5 ends the paper with some concluding remarks.

2. Preliminaries

Given a countable set of propositional variables $\mathcal{V} = \{x_i\}_{i \in \mathbb{N}}$, we define a propositional clause as a set of literals connected by the disjunction ($\vee$) operator, where a literal is either a variable x_i or its negation $\neg x_i$. Let $N = \{0, \frac{1}{n-1}, \ldots, \frac{n-1}{n-1}\}$ be a finite set of truth values. The $|N|$-*valued propositional logic* considered in this paper is established by a semantics comprehending an assignment $v : \mathcal{V} \to N$, which is extended to literals and clauses as follows:

$$v(\neg x_i) = 1 - v(x_i);$$
$$v(l_1 \vee l_2 \vee \cdots \vee l_n) = \max\{v(l_1), v(l_2), \ldots, v(l_n)\}.$$

Given $N = \{0, \frac{1}{n-1}, \ldots, \frac{n-1}{n-1}\}$ equipped with the natural total order $\leq$ on rational numbers, a regular sign is a subset of N either of the form $\{j \in N \mid j \geq i\}$, denoted by $\geq i$, or $\{j \in N \mid j \leq i\}$, denoted by $\leq i$, for some $i \in N$. Given a clause C, a regular clause is an expression either of the form $\geq i\!:\!C$ or $\leq i\!:\!C$. An assignment v satisfies $\geq i\!:\!C$ iff $v(C) \geq i$, and satisfies $\leq i\!:\!C$ iff $v(C) \leq i$.

Given a multiset of regular clauses Φ, the Regular Maximum Satisfiability Problem, or Regular MaxSAT, is to find an assignment that minimizes the number of unsatisfied clauses in Φ.

A weighted regular clause is a pair (C, w), where C is a regular clause and w, its weight, is a positive number. Given a multiset of weighted regular clauses Φ, Weighted Regular MaxSAT is to find an assignment that minimizes the sum of weights of unsatisfied clauses in Φ.

3. A Regular MaxSAT Tableau Calculus

We define a Regular MaxSAT tableau calculus and prove its soundness and completeness. The expansion rules we define in Definition 3.2 are graphically represented in Figure 1.

Definition 3.1. *A tableau is a tree with a finite number of branches whose nodes are labelled by either a regular clause or a box ($\Box$). A box in a tableau denotes a contradiction. A branch is a maximal path in a tree, and we assume that branches have a finite number of nodes.*

Definition 3.2. *Let $\Phi = \{C_1, \ldots, C_m\}$ be a multiset of regular clauses. A tableau for Φ is constructed by a sequence of applications of the following rules:*

Initialize *We start by generating a tree with a single branch with m nodes such that each node is labelled with a clause of Φ. Such a tableau is a tableau for Φ, called the initial tableau. All its clauses are declared to be active.*

Given a tableau T for Φ and a branch b of T, the result of applying any of the following rules results in a tableau for Φ with new branches extending b. All the following rules declare the premises to be inactive in the new branches, and declare the new nodes to be active:

- **$\vee$-rule ($\leq$ sign):** *If b contains an active regular clause $\leq i\!:\!l_1 \vee l_2 \vee \cdots \vee l_n$, where, for $1 \leq j \leq n$, l_j is a literal and $n \geq 2$, the resulting tableau contains a new (left) branch with a node below b labelled with $\Box$, and a new (right) branch with one node for every literal l_j below b labelled with $\leq i\!:\!l_j$.*
- **$\vee$-rule ($\geq$ sign):** *If b contains an active regular clause $\geq i\!:\!l_1 \vee l_2 \vee \cdots \vee l_n$, where, for $1 \leq j \leq n$, l_j is a literal and $n \geq 2$, the resulting tableau contains, for every literal l_j, a new branch with a node below b labelled $\geq i\!:\!l_j$.*

$$\begin{array}{c|c} \multicolumn{2}{c}{\leq i : l_1 \lor l_2 \lor \cdots \lor l_n} \\ \hline \Box & \leq i : l_1 \\ & \leq i : l_2 \\ & \dots \\ & \leq i : l_n \end{array} \qquad \begin{array}{c|c|c|c} \multicolumn{4}{c}{\geq i : l_1 \lor l_2 \lor \cdots \lor l_n} \\ \hline \geq i : l_1 & \geq i : l_2 & \dots & \geq i : l_n \end{array}$$

provided that $n \geq 2$

$\lor$-rule

$$\begin{array}{c} \leq i : \neg x \\ \hline \geq (1-i) : x \end{array} \qquad \begin{array}{c} \geq i : \neg x \\ \hline \leq (1-i) : x \end{array} \qquad \begin{array}{c|c} \multicolumn{2}{c}{\leq i : x} \\ \multicolumn{2}{c}{\geq j : x} \\ \hline \Box & \Box \\ \leq i : x & \geq j : x \end{array}$$

provided that $i < j$

$\neg$-rule **$\Box$-rule**

Figure 1. Tableau expansion rules for Regular MaxSAT

$\neg$-rule: *If b contains an active regular clause $\leq i : \neg x$ (resp. $\geq i : \neg x$), where x is a variable, the resulting tableau appends a node below b labelled with $\geq (1-i) : x$ (resp. $\leq (1-i) : x$).*

$\Box$-rule: *If b contains two active regular clauses $\leq i : x$ and $\geq j : x$ such that $i < j$, where x is a variable, the resulting tableau contains a new (left) branch with two nodes below b labelled with $\Box$ and $\leq i : x$, and a new (right) branch with two nodes below b labelled with $\Box$ and $\geq j : x$.*

Definition 3.3. *Let T be a tableau for a multiset of regular propositional clauses Φ. A branch b of T is saturated when no further expansion rules can be applied on b, and T is completed when all its branches are saturated. The cost of a saturated branch is the number of boxes in the branch. The cost of a completed tableau is the minimum cost among all its branches.*

We show below that the minimum number of clauses that can be unsatisfied in a multiset of regular propositional clauses Φ is m iff the cost of a completed tableau for Φ is m. Thus, the systematic construction of a completed tableau for Φ provides an exact method for solving Regular MaxSAT.

Example 3.4. *Figure 2 shows how to create a tableau, with the previous calculus, to prove that the minimum number of unsatisfied regular propositional clauses in the multiset $\Phi = \{\leq \frac{1}{3} : \neg x_3, \geq \frac{2}{3} : x_1 \lor x_2, \leq \frac{1}{3} : x_2 \lor x_3\}$ is one, assuming $N = \{0, \frac{1}{3}, \frac{2}{3}, 1\}$. The first tableau (the leftmost tableau on top) is the initial tableau, which contains one node for each regular propositional clause in Φ. The second tableau shows the application of the $\neg$-rule to $\leq \frac{1}{3} : \neg x_3$ and the $\lor$-rule to $\geq \frac{2}{3} : x_1 \lor x_2$. The third tableau shows the application of the $\lor$-rule to $\leq \frac{1}{3} : x_3 \lor x_2$. The fourth tableau shows the application of the $\Box$-rule to $\geq \frac{2}{3} : x_3$ and $\leq \frac{1}{3} : x_3$, and to $\geq \frac{2}{3} : x_2$ and $\leq \frac{1}{3} : x_2$. Finally, the fifth tableau ap-*

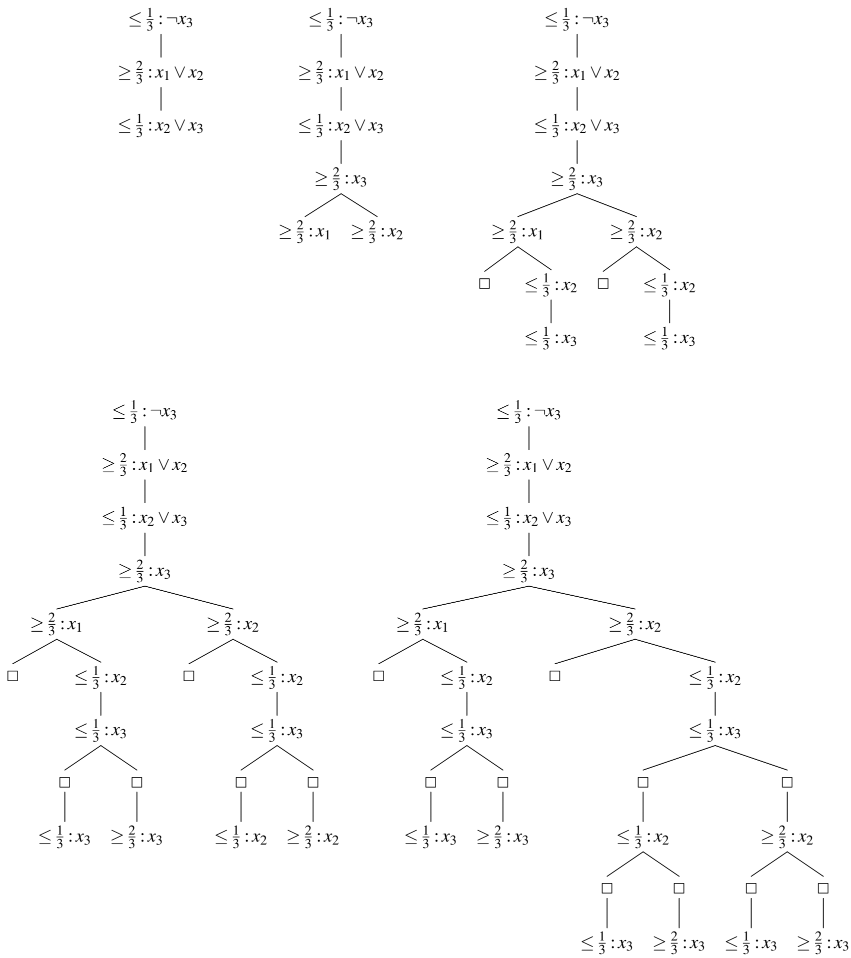

Figure 2. A tableaux for the Regular MaxSAT instance $\leq \frac{1}{3} : \neg x_3, \geq \frac{2}{3} : x_1 \vee x_2, \leq \frac{1}{3} : x_2 \vee x_3$.

plies the $\Box$-rule to $\geq \frac{2}{3} : x_3$ and $\leq \frac{1}{3} : x_3$ on the two rightmost branches. Since the tableau is completed and the minimum number of boxes among its branches is one, the minimum number of regular propositional clauses that can be unsatisfied in Φ is also one.

3.1. Soundness and completeness

We prove the soundness and completeness of the proposed calculus. Before presenting the completeness theorem, we prove termination and the soundness of the expansion rules.

Proposition 3.5. *A tableau for a multiset of regular propositional clauses Φ is completed in a finite number of steps.*

Proof. We first create an initial tableau and then apply expansion rules in the newly created branches until they become saturated. The $\vee$- and $\neg$-rule reduce the number of connectives. Since we began with a finite number of connectives, these rules can only be applied a finite number of times. The $\Box$-rule has two regular propositional clauses as premises and each conclusion contains exactly one clause in the conclusion that could be a premise of another rule and, therefore, it can only be applied a finite number of times. Hence, the construction of any completed tableau terminates in a finite number of steps. $\Box$

Proposition 3.6. *The $\vee$-, $\neg$- and $\Box$-rule preserve the number of premises unsatisfied by an assignment v in at least one branch and do not decrease that number in the other branches (if any).*

Proof. We show that the proposition holds for each rule separately:

- **$\vee$-rule ($\leq$ sign):** If v satisfies $\leq i : l_1 \vee l_2 \vee \cdots \vee l_n$, it holds that $v(l_j) \leq i$ for all $1 \leq j \leq n$, because of the maximum function. Hence, v satisfies all the clauses in at least one of the branches of the rule, specifically in the right branch. If v unsatisfies $\leq i : l_1 \vee l_2 \vee \cdots \vee l_n$, it holds that there is at least one l_j such that $v(l_j) > i$. Hence, v unsatisfies one clause of the left branch, and at least one clause of the right branch.
- **$\vee$-rule ($\geq$ sign):** If v satisfies $\geq i : l_1 \vee l_2 \vee \cdots \vee l_n$, no branch of the rule can decrease the number of unsatsified clauses in the premise, which is 0. Moreover, there is at least one l_j such that $v(l_j) \geq i$, and therefore there is at least one branch without unsatisfied clauses. If v unsatisfies $\geq i : l_1 \vee l_2 \vee \cdots \vee l_n$, it holds that $v(l_j) < i$ for all $1 \leq j \leq n$, and therefore v unsatisfies one clause of every branch.
- **$\neg$-rule:** When the premise is $\leq i : \neg x$, v satisfies $\leq i : \neg x$ iff $v(\neg x) = 1 - v(x) \leq i$ and, therefore, v satisfies $\leq i : \neg x$ iff it satisfies $\geq (1-i) : x$. The case where the premise is $\geq i : \neg x$ holds analogously.
- **$\Box$-rule:** Since $i < j$, at least one premise is unsatisfied. If $v(x) \leq i$, exactly one premise and one clause in the left conclusion are unsatisfied, and two in the right conclusion. If $i < v(x) < j$, two premises and two clauses in each conclusion are unsatisfied. If $v(x) \geq j$, exactly one premise and one clause in the right conclusion are unsatisfied, and two in the left conclusion.

$\Box$

Theorem 3.7. Soundness & completeness. *The cost of a completed tableau T for a multiset of clauses Φ is m iff the minimum number of unsatisfied clauses in Φ is m.*

Proof. (*Soundness:*) T was derived by creating a sequence of tableaux $T_0, \ldots, T_n$ ($n \geq 0$) such that T_0 is an initial tableau for Φ, $T_n = T$, and T_i was obtained by a single application of one of the $\vee$-, $\neg$- and $\Box$-rules on an branch of T_{i-1}, for $i = 1, \ldots, n$. By Proposition 3.5, we know that such a sequence is finite. Since T has cost m, T_n contains one branch b with exactly m boxes and the rest of the branches contain at least m boxes. Moreover, the active clauses in every branch of T_n are regular literals of the form $\leq i : x$ or $\geq i : x$ such that, for each propositional variable x, the intersection of all the signs is non-empty; otherwise, we could yet apply expansion rules and T_n would not be completed. Given a saturated branch with m boxes, the assignment that sets each variable x to one value of the intersection of all the signs of active regular unit clauses containing x only unsatisfies the m boxes. Moreover, there cannot be any assignment satisfying less than m clauses in a branch of T_n, because each branch contains at least m boxes. Therefore, the minimum number of active regular propositional clauses that can be unsatisfied among the branches of T_n is m.

Proposition 3.6 guarantees that the minimum number of unsatisfied active regular clauses is preserved in the sequence of tableaux $T_0, \ldots, T_n$. Thus, the minimum number of unsatisfied regular clauses in T_0 is also m. Since T_0 is formed by a single branch that only contains the clauses in Φ and all these clauses are active, the minimum number of clauses that can be unsatisfied in Φ is m.

(*Completeness:*) Assume that there is a completed tableau T for Φ that does not have cost m. We distinguish two cases:

(i) T has a branch b of cost k, where $k < m$. Then, T has a branch with k boxes and a satisfiable multiset of active clauses because T is completed. This implies that the minimum number of unsatisfied active clauses among the branches of T is at most k. By Proposition 3.6, this also holds for T_0, but this is in contradiction with m being the minimum number of unsatisfied clauses in Φ because $k < m$. Thus, any branch of T has at least cost m.

(ii) T has no branch of cost m. This is in contradiction with m being the minimum number of unsatisfied clauses in Φ. Since the tableau expansion rules preserve the minimum number of unsatisfied clauses and the branches of any completed tableau only contain active clauses that are boxes or regular literals from which we cannot derive contradictions, T must have a saturated branch with m boxes. Thus, T has a branch of cost m.

Hence, each completed tableau T for a multiset of clauses Φ has cost m if the minimum number of clauses that can be unsatisfied in Φ is m. □

4. A Tableau Calculus for Weighted Regular MaxSAT

Figure 3 displays the expansion rules of a complete tableau calculus for Weighted Regular MaxSAT. If there is one premise (C, w), the weighted expansion rules are identical except for the fact that we propagate the weight from the premise to the conclusions. In fact, dealing with weighted regular propositional clauses can be understood as collapsing several unweighted Regular MaxSAT inferences into a single inference, because a weighted clause (C, w) can be replaced by w copies of the unweighted clause C. Therefore, the soundness and completeness proofs for unweighted Regular MaxSAT are trivially extensible to Weighted Regular MaxSAT. If there are two premises (C_1, w_1) and (C_2, w_2) with different weights ($w_1 \neq w_2$), (C_1, w_1) and (C_2, w_2) become inactive but $(C_1, w_1 - w)$ and $(C_2, w_2 - w)$, where $w = \min(w_1, w_2)$, are added as active clauses (clauses with weight 0 are not added). In this case, the conclusions of the inference have weight w. Note that, in the left branch of the $\Box$-rule, the conclusion $(\leq i : x, w_1)$ is the result of merging $(\leq i : x, w)$ and $(\leq i : x, w_1 - w)$, by accumulating the weight $w_1 = w + w_1 - w$ (and the analogous case occurs in the right branch).

Example 4.1. *Figure 4 displays a completed tableau for the Weighted Partial Regular MaxSAT instance* $\{(\geq 1 : x_1 \vee x_3, 3), (\geq 1 : \neg x_3, 2), (\leq \frac{1}{3} : x_1 \vee x_2, 5)\}$*, assuming* $N = \{0, \frac{1}{3}, \frac{2}{3}, 1\}$*. The first tableau (top left) is the initial tableau. The second tableau results*

$$\begin{array}{c|c} \multicolumn{2}{c}{(\leq i : l_1 \vee l_2 \vee \cdots \vee l_n, w)} \\ \hline (\Box, w) & (\leq i : l_1, w) \\ & (\leq i : l_2, w) \\ & \cdots \\ & (\leq i : l_n, w) \end{array} \qquad \begin{array}{c|c|c|c} \multicolumn{4}{c}{(\geq i : l_1 \vee l_2 \vee \cdots \vee l_n, w)} \\ \hline (\geq i : l_1, w) & (\geq i : l_2, w) & \ldots & (\geq i : l_n, w) \end{array}$$

provided that $n \geq 2$

$\vee$-rule

$$\frac{(\leq i : \neg x, w)}{(\geq (1-i) : x, w)} \quad \frac{(\geq i : \neg x, w)}{(\leq (1-i) : x, w)}$$

$\neg$-rule

$$\begin{array}{c|c} \multicolumn{2}{c}{(\leq i : x, w_1)} \\ \multicolumn{2}{c}{(\geq j : x, w_2)} \\ \hline (\Box, w) & (\Box, w) \\ (\leq i : x, w_1) & (\geq j : x, w_2) \\ (\geq j : x, w_2 - w) & (\leq i : x, w_1 - w) \end{array}$$

provided that $i < j$

where $w = \min(w_1, w_2)$

$\Box$-rule

Figure 3. Tableau expansion rules for Weighted Regular MaxSAT

of the application of the $\vee$-rule to $(\geq 1 : x_1 \vee x_3, 3)$, the $\neg$-rule to $(\geq 1 : \neg x_3, 2)$, and the $\vee$-rule to $(\geq 1 : x_1 \vee x_3, 3)$. The third tableau results of the application of the $\Box$-rule to $(\leq \frac{1}{3} : x_1, 5)$ and $(\geq 1 : x_1, 3)$. The last tableau results of the application of the $\Box$-rule to $(\geq 1 : x_3, 3)$ and $(\leq 0 : x_3, 2)$ on both branches of the right subtree. Since the minimum cost among all the branches is 10, the minimum sum of weights of the unsatisfied clauses is 10.

5. Conclusions

We presented a complete tableau calculus for Regular MaxSAT, proved its soundness and completeness and defined its extension to Weighted Regular MaxSAT. As future work, we plan to extend the calculus to other formalisms that may be better suited to encode specific optimization problems. For instance, in Weighted Partial Regular MaxSAT, we have hard and weighted soft clauses; in Regular MinSAT, the goal is to maximize the number of unsatisfied clauses; and in first-order logic, we can make use of quantifiers and predicates.

Acknowledgements: This work has been supported by the French Agence Nationale de la Recherche, reference ANR-19-CHIA-0013-01, and grants PID2019-111544GB-C21, PID2022-139835NB-C21 and TED2021-129319B-I00 funded by MCIN/AEI/10.13039/501100011033.

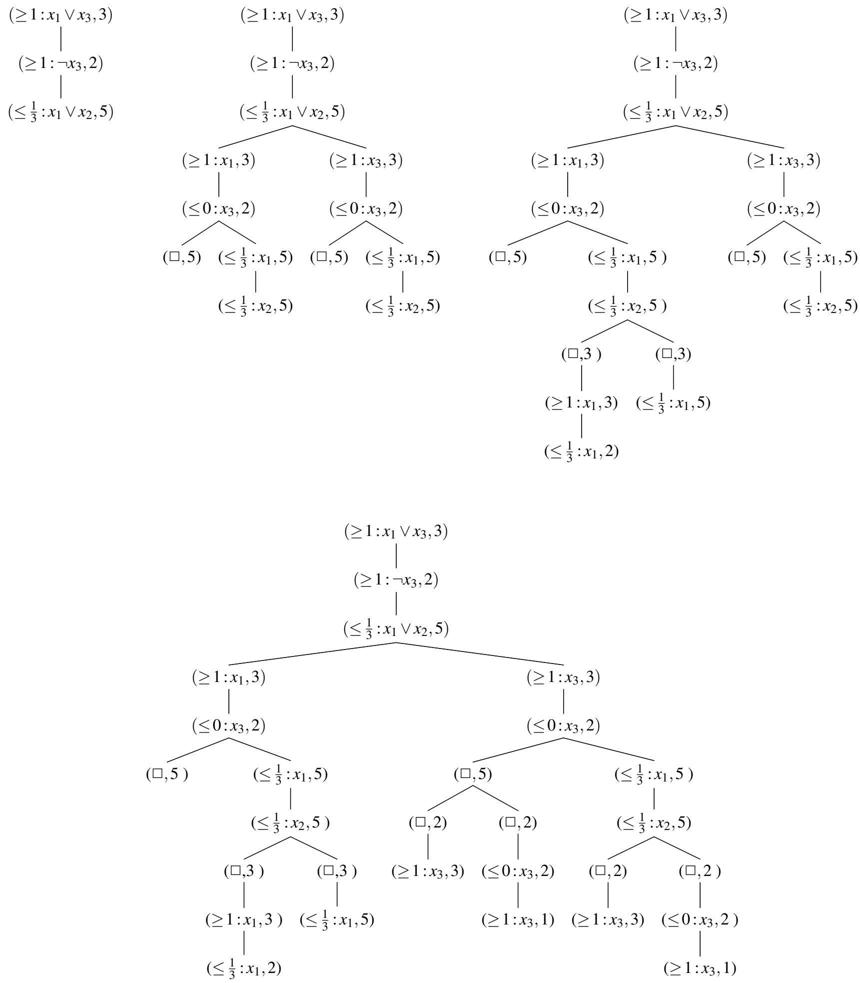

Figure 4. A tableau for the Regular MaxSAT instance $\{(\geq 1 : x_1 \vee x_3, 3), (\leq \frac{1}{3} : x_1 \vee x_2, 5), (\geq 1 : \neg x_3, 2)\}$.

References

[1] C. Ansótegui, M. L. Bonet, J. Levy, and F. Manyà. The logic behind weighted CSP. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence, IJCAI-2007, Hyderabad, India*, pages 32–37, 2007.

[2] C. Ansótegui, M. L. Bonet, J. Levy, and F. Manyà. Resolution procedures for multiple-valued optimization. *Information Sciences*, 227:43–59, 2013.

[3] C. Ansótegui, F. Manyà, J. Ojeda, J. M. Salvia, and E. Torres. Incomplete MaxSAT approaches for combinatorial testing. *Journal of Heuristics*, 28(4):377–431, 2022.

[4] J. Argelich, C. M. Li, F. Manyà, and J. R. Soler. Clause tableaux for maximum and minimum satisfiability. *Logic Journal of the IGPL*, 29(1):7–27, 2021.

[5] F. Bacchus, M. Järvisalo, and M. Ruben. Maximum satisfiability. In A. Biere, M. Heule, H. van Maaren, and T. Walsh, editors, *Handbook of Satisfiability*, pages 929–991. IOS Press, 2021.

[6] B. Beckert, R. Hähnle, and F. Manyà. The SAT problem of signed CNF formulas. In D. Basin, M. D'Agostino, D. Gabbay, S. Matthews, and L. Viganò, editors, *Labelled Deduction*, volume 17 of

Applied Logic Series, pages 61–82. Kluwer, Dordrecht, 2000.

[7] M. Bofill, J. Coll, M. Garcia, J. Giráldez-Cru, G. Pesant, J. Suy, and M. Villaret. Constraint solving approaches to the business-to-business meeting scheduling problem. *Journal of Artificial Intelligence Research*, 74:263–301, 2022.

[8] M. L. Bonet, J. Levy, and F. Manyà. A complete calculus for Max-SAT. In *Proceedings of the 9th International Conference on Theory and Applications of Satisfiability Testing, SAT-2006, Seattle, USA*, pages 240–251. Springer LNCS 4121, 2006.

[9] M. L. Bonet, J. Levy, and F. Manyà. Resolution for Max-SAT. *Artificial Intelligence*, 171(8–9):240–251, 2007.

[10] D. D'Almeida and É. Grégoire. Model-based diagnosis with default information implemented through MAX-SAT technology. In *Proceedings of the IEEE 13th International Conference on Information Reuse & Integration, IRI, Las Vegas, NV, USA*, pages 33–36, 2012.

[11] G. Fiorino. New tableau characterizations for non-clausal MaxSAT problem. *Logic Journal of the IGPL*, 30(3):422–436, 2022.

[12] J. Larrosa and F. Heras. Resolution in Max-SAT and its relation to local consistency in weighted CSPs. In *Proceedings of the International Joint Conference on Artificial Intelligence, IJCAI-2005, Edinburgh, Scotland*, pages 193–198. Morgan Kaufmann, 2005.

[13] J. Larrosa, F. Heras, and S. de Givry. A logical approach to efficient Max-SAT solving. *Artificial Intelligence*, 172(2–3):204–233, 2008.

[14] C. Li, Z. Xu, J. Coll, F. Manyà, D. Habet, and K. He. Combining clause learning and branch and bound for MaxSAT. In *Proceedings of the 27th International Conference on Principles and Practice of Constraint Programming, CP, Montpellier, France*, volume 210 of *LIPIcs*, pages 38:1–38:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.

[15] C. M. Li and F. Manyà. MaxSAT, hard and soft constraints. In A. Biere, M. Heule, H. van Maaren, and T. Walsh, editors, *Handbook of Satisfiability*, pages 903–927. IOS Press, 2021.

[16] C. M. Li and F. Manyà. Inference in MaxSAT and MinSAT. In W. Ahrendt, B. Beckert, R. Bubel, and E. B. Johnsen, editors, *The Logic of Software. A Tasting Menu of Formal Methods - Essays Dedicated to Reiner Hähnle on the Occasion of His 60th Birthday*, volume 13360 of *Lecture Notes in Computer Science*, pages 350–369. Springer, 2022.

[17] C. M. Li, F. Manyà, and J. R. Soler. A clause tableaux calculus for MaxSAT. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence, IJCAI-2016, New York, USA*, pages 766–772, 2016.

[18] C. M. Li, F. Manyà, and J. R. Soler. A tableau calculus for non-clausal maximum satisfiability. In *Proceedings of the 28th International Conference on Automated Reasoning with Analytic Tableaux and Related Methods, TABLEAUX, London, UK*, pages 58–73. Springer LNCS 11714, 2019.

[19] C. M. Li, F. Manyà, and A. Vidal. Tableaux for maximum satisfiability in Łukasiewicz logic. In *Proceedings of the 50th IEEE International Symposium on Multiple-Valued Logic, ISMVL, Miyazaki, Japan*, pages 243–248, 2020.

[20] S. Li, J. Coll, D. Habet, C. M. Li, and F. Manyà. A tableau calculus for signed maximum satisfiability. In *Proceedings of the 53rd IEEE International Symposium on Multiple-Valued Logics (ISMVL), Matsue, Japan*, pages 170–175, 2023.

[21] F. Manyà, S. Negrete, C. Roig, and J. R. Soler. Solving the team composition problem in a classroom. *Fundamamenta Informaticae*, 174(1):83–101, 2020.

[22] J. Marques-Silva, J. Argelich, A. Graça, and I. Lynce. Boolean lexicographic optimization: algorithms & applications. *Annals of Matematics and Artificial Intelligence*, 62(3-4):317–343, 2011.

[23] S. Safarpour, H. Mangassarian, A. G. Veneris, M. H. Liffiton, and K. A. Sakallah. Improved design debugging using maximum satisfiability. In *Proceedings of 7th International Conference on Formal Methods in Computer-Aided Design, FMCAD, Austin, Texas, USA*, pages 13–19, 2007.

[24] L. Zhang and F. Bacchus. MAXSAT heuristics for cost optimal planning. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence, Toronto, Ontario, Canada*, pages 1846–1852, 2012.