

Unifying Reasoning and Core-Guided Search for Maximum Satisfiability

Jeremias Berg^[0000–0001–7660–8061] and Matti Järvisalo^[0000–0003–2572–063X]

HIIT, Department of Computer Science, University of Helsinki, Finland
{jeremias.berg,matti.jarvisalo}@helsinki.fi

Abstract. A central algorithmic paradigm in maximum satisfiability solving geared towards real-world optimization problems is the core-guided approach. Furthermore, recent progress on preprocessing techniques is bringing in additional reasoning techniques to MaxSAT solving. Towards realizing their combined potential, understanding formal underpinnings of interleavings of preprocessing-style reasoning and core-guided algorithms is important. It turns out that earlier proposed notions for establishing correctness of core-guided algorithms and preprocessing, respectively, are not enough for capturing correctness of interleavings of the techniques. We provide an in-depth analysis of these and related MaxSAT instance transformations, and propose correction set reducibility as a notion that captures inprocessing MaxSAT solving within a state-transition style abstract MaxSAT solving framework. Furthermore, we establish a general theorem of correctness for applications of SAT-based preprocessing techniques in MaxSAT. The results pave way for generic techniques for arguing about the formal correctness of MaxSAT algorithms.

Keywords: maximum satisfiability · core-guided reasoning · preprocessing · instance transformations · inprocessing

1 Introduction

Maximum satisfiability (MaxSAT), the optimization variant of Boolean satisfiability (SAT), provides a competitive approach to various real-world optimization problems arising from AI and industrial applications, see e.g. [18, 34, 25, 14, 33, 7, 6]. Most of the modern MaxSAT solvers are based on iteratively transforming an input problem instance in specific ways towards a representation from which an optimal solution is in some sense “easy” to compute. In particular, a central algorithmic paradigm in modern MaxSAT solving geared towards real-world optimization problems is the so-called core-guided approach [2, 27]. Core-guided MaxSAT solvers reduce the search for an optimal solution to a sequence of SAT instances, forming the next instance in the sequence by transforming the current one based on an unsatisfiable core reported by a SAT solver until a solution is found. In addition to the core-guided approach, MaxSAT preprocessing [4, 5, 8, 19] also iteratively applies instance transformations through simplification (or reasoning) techniques which, however, significantly differ from core-guided transformations.

The formal underpinning of inprocessing SAT solving [17], a popular approach to modern SAT solving based on interleaving preprocessing techniques with conflict-driven clause learning search, is today well-understood. In contrast, preprocessing for MaxSAT, including the realization of liftings of SAT preprocessing to MaxSAT [5], is a more recent line of development. In fact, so far only a few specific liftings of SAT preprocessing techniques have been shown to be correct for MaxSAT [5]. Furthermore, towards lifting the inprocessing SAT solving paradigm to the realm of MaxSAT, understanding how to interleave core-guided MaxSAT inferences and inference steps made by MaxSAT preprocessing techniques is important. While formal notions of instance transformations have been proposed for establishing correctness of core-guided algorithms and preprocessing, respectively, these notions in themselves are not expressive enough for capturing correctness of interleavings of the two types of transformations.

To address these shortcomings, we focus in this paper on providing further understanding of correct instance transformations for generic MaxSAT solving. To this end, we analyze both earlier proposed formal notions of instance transformations [2, 5], and explain why they are fundamentally different and therefore individually not enough to capture interleaving of core-guided and preprocessing-style instance transformations; both types of transformations are required in order to obtain a framework capable of modelling MaxSAT solving without relying on the correctness of SAT solvers. We propose correction set reducibility as a general notion of instance transformations that captures to a far extent transformations applied in both core-guided solvers and MaxSAT preprocessing. We base our analysis on a formal framework as an abstract state transition system based on different sets of sequences of MaxSAT instances. This allows for reasoning about correctness of core-guided solving and MaxSAT preprocessing in a unified manner which can ease the development of new MaxSAT solving methods, including inprocessing. Furthermore, as a further form of instance transformation, we lift the notion of resolution asymmetric tautologies (RAT clauses), a simple yet powerful notion capturing SAT preprocessing techniques at large in a unified way [17], to MaxSAT. By doing so, we establish a general proof of correctness for natural liftings of SAT preprocessing techniques to MaxSAT, thereby significantly generalizing the correctness proofs earlier presented for MaxSAT liftings of specific SAT preprocessing techniques [5]. The results pave way for generic techniques for arguing about the formal correctness of MaxSAT algorithms.

2 Maximum Satisfiability, MaxSAT Solving and Preprocessing

A literal l is a Boolean variable x or its negation $\neg x$. For a set L of literals, the set $\neg L$ contains the negations of the literals in L ; L is consistent if L and $\neg L$ are disjoint. A clause is a disjunction (logical OR) of literals (represented as a set of its literals) and a CNF formula F is a conjunction (logical AND) of clauses (represented as a set of its clauses). A clause C is a tautology if $\{x, \neg x\} \subset C$ for some variable x . The set $\text{VAR}(C)$ contains all variables x for which $x \in C$

or $\neg x \in C$. The set $\text{VAR}(F)$ of the variables of F is $\cup_{C \in F} \text{VAR}(C)$ and the set $\text{LIT}(F)$ is $\cup_{C \in F} C$ (each C seen as a set of literals). For a literal $l \in \text{LIT}(F)$ we use $\text{CL}_F(l)$ to denote the set of clauses in F which contain l , dropping the subscript when clear from context.

A (truth) assignment τ is a consistent set of literals. A literal l is true under τ ($\tau(l) = 1$) if $l \in \tau$ and false ($\tau(l) = 0$) if $\neg l \in \tau$. A literal l true or false under τ is *assigned* in (or assigned by) τ , and unassigned otherwise. An assignment τ satisfies a clause C ($\tau(C) = 1$) if $\tau(l)$ for some literal l in C , i.e., if $\tau \cap C \neq \emptyset$. τ satisfies a formula F ($\tau(F) = 1$) if it satisfies all clauses in it. A formula is satisfiable if there is an assignment that satisfies it, and else unsatisfiable. An assignment τ is a complete assignment to a CNF formula F if every literal $l \in \text{LIT}(F)$ is assigned in τ , else it is partial. The restriction $F|_\tau$ of a formula F under an assignment τ is the CNF formula $F \wedge \bigwedge_{l \in \tau} (l)$.

A (weighted partial) MaxSAT instance is a triplet $\mathcal{F} = (F_h, F_s, w)$ consisting of a set F_h of hard clauses, a set F_s of soft clauses and a weight function $w: F_s \rightarrow \mathbb{N}$. The instance is partial if $F_h \neq \emptyset$ and unweighted if $w(C) = k$ for some constant $k \in \mathbb{N}$ and all $C \in F_s$. The core-guided MaxSAT algorithms we focus on in this work solve the most general case of weighted partial MaxSAT and do not treat any variant of it any differently (cf. Section 2.2). Hence we will refer to weighted partial MaxSAT simply by MaxSAT. The cost $\text{COST}(\mathcal{F}, \tau)$ of a complete assignment to $F_h \wedge F_s$ is ∞ if $\tau(F_h) = 0$ and $\sum_{C \in F_s} (1 - \tau(C)) \cdot w(C)$ otherwise. We say that τ is a solution to \mathcal{F} if $\tau(F_h) = 1$ and optimal if $\text{COST}(\mathcal{F}, \tau) \leq \text{COST}(\mathcal{F}, \tau^*)$ for all complete truth assignments τ^* to $F_h \wedge F_s$. We denote the cost of the optimal solutions to \mathcal{F} by $\text{COST}(\mathcal{F})$. The cost of a partial assignment τ^p to $F_h \wedge F_s$ is the cost of an “optimal extension” of τ^p into a complete assignment, i.e., $\text{COST}(\mathcal{F}, \tau^p) = \text{COST}(\mathcal{F}^{\tau^p})$, where $\mathcal{F}^{\tau^p} = (F_h|_{\tau^p}, F_s, w)$.

A subset $\kappa \subset F_s$ is an unsatisfiable subset (or core) of \mathcal{F} if $F_h \wedge \kappa$ is unsatisfiable and a minimal unsatisfiable subset (MUS) if $F_h \wedge \kappa_s$ is satisfiable for all $\kappa_s \subsetneq \kappa$. A set H is a correction set (CS) if $F_h \wedge (F_s \setminus H)$ is satisfiable and a minimal correction set (an MCS) if $F_h \wedge (F_s \setminus H_s)$ is unsatisfiable for all $H_s \subsetneq H$. The sets of MUSes and MCSes of \mathcal{F} are denoted by $\text{MUS}(\mathcal{F})$ and $\text{MCS}(\mathcal{F})$, respectively. For a solution τ to \mathcal{F} the set $U(\mathcal{F}, \tau) = \{C \mid \tau(C) = 0\}$ contains the soft clauses falsified by τ . We say that the solution τ corresponds to the correction set $U(\mathcal{F}, \tau)$. Similarly, a correction set H corresponds to a solution τ if $H = U(\mathcal{F}, \tau)$. A correction set H is optimal if it corresponds to an optimal solution τ to \mathcal{F} . It is easy to show that every solution corresponds to a correction set and every correction set corresponds to some solution.

2.1 Core-Guided MaxSAT Solving and MaxSAT-Reducibility

When solving a MaxSAT instance \mathcal{F} , a core-guided MaxSAT algorithm maintains a working instance $\mathcal{F}^i = (F_h^i, F_s^i, w^i)$, initialized to \mathcal{F} . During each iteration of the search, a SAT solver is queried for the satisfiability of $F_h^i \wedge F_s^i$. If the formula is satisfiable, i.e., if $\text{COST}(\mathcal{F}^i) = 0$, any assignment τ^o satisfying $F_h^i \wedge F_s^i$ is an optimal solution to \mathcal{F} . Otherwise, the SAT solver returns a core κ^i of \mathcal{F}^i .

The next working instance \mathcal{F}^{i+1} is then formed by transforming \mathcal{F}^i in a way that rules out κ^i as a source of unsatisfiability and lowers the optimal cost of \mathcal{F}^i by $\min\{w^i(C) \mid C \in \kappa^i\}$. Many existing core-guided algorithms fit this high-level description and differ mainly in the the specific transformation used to form the next working instance [1, 15, 26, 3, 29, 26]. The correctness of such solvers, i.e., that the final assignment returned will indeed be an optimal solution to the input instance, is often established by proving that \mathcal{F}^i is MaxSAT-reducible [2] to \mathcal{F}^{i+1} .

Definition 1. *An instance \mathcal{F} is MaxSAT-reducible (or k -reducible) to the instance \mathcal{F}^R if $\text{COST}(\mathcal{F}, \tau) = \text{COST}(\mathcal{F}^R, \tau) + k$ for some integer k and for all complete assignments τ to \mathcal{F} . The constant k can depend on \mathcal{F} but not on the assignment τ .*

An important motivation for the abstract model of MaxSAT solving we detail in Section 3 relates to the specifics of how cores are extracted with SAT solvers. In practice, a core κ of an instance $\mathcal{F} = (F_h, F_s, w)$ is extracted by extending each soft clause $C_i \in F_s$ with an unique *assumption variable* a_i to form the clause $C_i \vee a_i$. Afterwards the so-called assumption interface of the SAT solver [28] is used to query the satisfiability of $(F_h \wedge F_s^A)|_{\neg \mathcal{A}}$, where F_s^A is the set of extended soft clauses and \mathcal{A} the set of all assumption variables. If the result is satisfiable, the obtained assignment satisfies $(F_h \wedge F_s^A) \wedge \bigwedge_{a \in \mathcal{A}} (\neg a)$ and hence also $(F_h \wedge F_s)$. If the formula is unsatisfiable, the SAT solver instead returns a subset $A_s \subset \mathcal{A}$ for which $F_h \wedge F_s|_{\neg A_s}$ is unsatisfiable as well. Indeed, as we illustrate in Example 1, modern core-guided solvers represent cores in terms of the variables in A_s [1, 26, 29].

2.2 MaxSAT Preprocessing and MCS-Equivalence

MaxSAT preprocessing refers to the application of different simplification and deduction rules to an instance $\mathcal{F} = (F_h, F_s, w)$, resulting in another instance \mathcal{F}^p . A simple example is the so-called subsumption rule which allows removing a clause $D \in (F_h \wedge F_s)$ if there is a clause $C \in F_h$ for which $C \subset D$. The goal of correct and effective preprocessing for MaxSAT is to make the time required to transform \mathcal{F} , solve \mathcal{F}^p and reconstruct an optimal solution to \mathcal{F} lower than the time required to solve \mathcal{F} directly. The previously proposed notion for proving correctness of MaxSAT preprocessing requires the use of the following literal-based definition of MaxSAT [5]. In particular, for the remainder of this paper, we will apply the following literal-based definitions of MaxSAT.

A MaxSAT instance \mathcal{F} consists of a CNF formula $\text{CLAUSES}(\mathcal{F})$ and a weight function $w^{\mathcal{F}}: \text{VAR}(\text{CLAUSES}(\mathcal{F})) \rightarrow \mathbb{N}$, assigning a weight to each variable of \mathcal{F} . Whenever clear from context, we use \mathcal{F} and $\text{CLAUSES}(\mathcal{F})$ interchangeably. A variable $x \in \text{VAR}(\mathcal{F})$ is soft in \mathcal{F} if $w^{\mathcal{F}}(x) > 0$. The set $\mathcal{S}(\mathcal{F})$ contains all soft variables of \mathcal{F} . The complement S^c of a $S \subset \mathcal{S}(\mathcal{F})$ is $\mathcal{S}(\mathcal{F}) \setminus S$.

The other concepts related to MaxSAT discussed earlier are lifted from soft clauses to soft variables in a natural way. An assignment τ is a solution to \mathcal{F} if

$\tau(\mathcal{F}) = 1$ and has cost $\text{COST}(\mathcal{F}, \tau) = \sum_{x \in \text{VAR}(\mathcal{F})} \tau(x) \cdot w(x)$. A set $\kappa \subset \mathcal{S}(\mathcal{F})$ is a core of \mathcal{F} if $\mathcal{F}|_{\neg\kappa}$ is unsatisfiable. Similarly $H \subset \mathcal{S}(\mathcal{F})$ is a CS of \mathcal{F} if $\mathcal{F}|_{\neg(H^c)}$ is satisfiable and an MCS if no $H_s \subsetneq H$ is a CS of \mathcal{F} . Notice that under the literal-based definitions, the set $U(\mathcal{F}, \tau)$ is simply $\tau \cap \mathcal{S}(\mathcal{F})$. Using these definitions, the notion of *MCS-equivalence* has been used as a basis for showing correctness of the liftings of four specific preprocessing rules proposed for SAT solving to MaxSAT [5].

Definition 2. *The instance \mathcal{F} is MCS-equivalent with the instance \mathcal{F}^R (and vice-versa) if $\text{MCS}(\mathcal{F}) = \text{MCS}(\mathcal{F}^R)$ and $w^{\mathcal{F}}(x) = w^{\mathcal{F}^R}(x)$ for all $x \in \text{VAR}(\mathcal{F}) \cap \text{VAR}(\mathcal{F}^R)$.*

As we will demonstrate, defining MaxSAT based on soft variables instead of soft clauses allows reasoning about core-guided solving and MaxSAT preprocessing in a unified manner. We emphasize that the literal-based definition is equivalent to the clause-based one. Furthermore, the literal-based definitions correspond more closely with the representation of MaxSAT instances that core-guided solvers and MaxSAT preprocessors actually operate on. Given any MaxSAT instance (partial or not), a core-guided solver and similarly a MaxSAT preprocessor will add an explicit variable a_i to each soft clause. During solving and preprocessing, the extended soft clauses and hard clauses are treated equally. Instead, special treatment is given to the added variables; for correctness, a preprocessor is restricted from resolving on them, and a core-guided solver extracts cores and applies transformations in terms of the a_i 's, instead of in terms of the soft clauses directly. The concept of a soft variable makes the role of these “special” variables explicit, highlighting the similarities of core-guided solving and preprocessing. Furthermore, applications of specific preprocessing techniques such as bounded variable elimination will result in clauses with several soft variables; the literal-based view also uniformly covers this. The literal-based definitions also allow describing the transformations used by modern core-guided solvers in a succinct manner.

Example 1. Let \mathcal{F} be a MaxSAT instance, $\kappa = \{l_1, \dots, l_n\}$ a core of \mathcal{F} , and $w_\kappa = \min_{l \in \kappa} \{w^{\mathcal{F}}(l)\}$. The instance transformation used by the PMRES core-guided algorithm [29] forms the instance $\mathcal{F}^R = \mathcal{F} \wedge \bigwedge_{i=1}^{n-1} (c_i \leftrightarrow (c_{i+1} \vee l_{i+1})) \wedge \bigwedge_{i=1}^{n-1} ((\neg c_i \vee \neg l_i \vee r_i)) \wedge (\neg c_n)$ with the equivalence expressed as clauses in the standard way. Each c_i and r_i are new variables that do not appear in $\text{VAR}(\mathcal{F})$. The weights of the variables \mathcal{F}^R are modified by (i) decreasing the weight of each $l \in \kappa$ by w_κ , (ii) setting the weight of each c_i to 0, (iii) setting the weight of each r_i to w_κ and (iv) keeping the weights of all other variables the same as in \mathcal{F} . The fact that \mathcal{F} is MaxSAT-reducible to \mathcal{F}^R was first shown in [29].

3 An Abstract MaxSAT Solving Framework

In the rest of this work we study MaxSAT-reducibility, MCS-equivalence and other notions of transformation properties in an abstract framework based on

sequences of MaxSAT instances (or sequences for short). For example, solving an instance \mathcal{F} with a core-guided MaxSAT solver is identified with a sequence $\langle \mathcal{F} = \mathcal{F}_1, \dots, \mathcal{F}_n \rangle$, where each instance \mathcal{F}_i is MaxSAT-reducible to \mathcal{F}_{i+1} and $\text{COST}(\mathcal{F}_n) = 0$. Similarly, preprocessing \mathcal{F} is identified with a sequence $\langle \mathcal{F} = \mathcal{F}_1, \dots, \mathcal{F}_n \rangle$, where each \mathcal{F}_i is MCS-equivalent with \mathcal{F}_{i+1} . The notion of MaxSAT-reducibility (MCS-equivalence) is lifted to sequences of instances by the set MSRED (MCSEQ) containing all sequences $\langle \mathcal{F}_1, \dots, \mathcal{F}_n \rangle$ for which \mathcal{F}_i is MaxSAT-reducible to (MCS-equivalent with) \mathcal{F}_{i+1} for all $i = 1, \dots, n-1$.

More generally, the framework captures MaxSAT solving techniques that iteratively transform an input MaxSAT instance toward a specific final instance, from which an optimal solution to the input instance can then be obtained based on an optimal solution to the final instance. As the final instance we use the (unique) MaxSAT instance $\mathcal{F}_F = \emptyset$ that contains no clauses and to which any assignment τ is an optimal solution to. The following notion of a terminating sequence represents MaxSAT solving within the general framework.

Definition 3. *A sequence $\langle \mathcal{F}, \dots, \mathcal{F}_n \rangle$ is terminating if $\mathcal{F}_n = \mathcal{F}_F$.*

An important observation to make regarding the sets of sequences that we work with is that the membership of a sequence $\langle \mathcal{F}_1, \dots, \mathcal{F}_n \rangle$ in each set can be determined “locally” by checking if some property holds between \mathcal{F}_i and \mathcal{F}_{i+1} for all $i = 1, \dots, n-1$. For example, $\langle \mathcal{F}_1, \dots, \mathcal{F}_n \rangle \in \text{MSRED}$ can be checked by verifying that \mathcal{F}_i is MaxSAT-reducible to \mathcal{F}_{i+1} for all i . More formally, we say that a set S of sequences is decomposable if $\langle \mathcal{F}_1, \dots, \mathcal{F}_n \rangle \in S$ if and only if $\langle \mathcal{F}_i, \mathcal{F}_{i+1} \rangle \in S$ for all $i = 1, \dots, n-1$. All sets of sequences that we consider in this work with are decomposable, including the already defined MSRED and MCSEQ.

The following notion allows for combining sets of sequences for modelling interleavings of different types of instance transformations.

Definition 4. *The combination $S_1 \circ S_2$ of two sets S_1 and S_2 of sequences contains all sequences $\langle \mathcal{F}_1, \dots, \mathcal{F}_n \rangle$ for which $\langle \mathcal{F}_i, \mathcal{F}_{i+1} \rangle \in S_1 \cup S_2$.*

For example, the set $\text{MCSEQ} \circ \text{MSRED}$ contains all sequences $\langle \mathcal{F}_1, \dots, \mathcal{F}_n \rangle$ where \mathcal{F}_i is either MaxSAT-reducible to, or MCS equivalent with, \mathcal{F}_{i+1} for all i . Informally speaking, $\text{MCSEQ} \circ \text{MSRED}$ models inprocessing MaxSAT solving, interleaving preprocessing and core-guided search.

When analyzing a set S of sequences, we focus on three central properties that are interesting in the context of MaxSAT solving. The first property is *sufficiency*: that for any instance \mathcal{F} there is a terminating sequence in S that starts from \mathcal{F} .

Definition 5. *Let \mathcal{F} be a MaxSAT instance. A set S of sequences is sufficient (for reaching the final state) if there is a terminating sequence $\langle \mathcal{F}, \dots, \mathcal{F}_F \rangle \in S$.*

The second property, *effectiveness*, captures the idea that for practical applicability, an optimal solution τ_n to the last instance \mathcal{F}_n in a sequence $\langle \mathcal{F}, \dots, \mathcal{F}_n \rangle$ should be applicable or “useful” for obtaining an optimal solution τ to \mathcal{F} . In the following, we say that a function that on input $\langle \mathcal{F}, \dots, \mathcal{F}_n \rangle$ and τ_n computes τ is a reconstruction function for $\langle \mathcal{F}, \dots, \mathcal{F}_n \rangle$.

Definition 6. Let S be a set of sequences and $\langle \mathcal{F}, \mathcal{F}^T \rangle \in S$ any sequence of length two in S . The set S is *effective* (for MaxSAT solving) if there is a reconstruction function for $\langle \mathcal{F}, \mathcal{F}^T \rangle$ computable in polynomial time with respect to $|\mathcal{F}|$.

If the set S is decomposable, as all of the sets we work with are, the ability to reconstruct optimal solutions is extended to sequences of arbitrary lengths.

Observation 1 Let S be an effective decomposable set of sequences and consider a sequence $\langle \mathcal{F}_1, \dots, \mathcal{F}_n \rangle \in S$. Assume that $|\mathcal{F}_i|$ is polynomial in $|\mathcal{F}_1|$ for all i . Then there is a reconstruction function for $\langle \mathcal{F}_1, \dots, \mathcal{F}_n \rangle$ computable in time $\mathcal{O}(n \times g(|\mathcal{F}_1|))$ for some polynomial g .

An alternative view of effectiveness is hence that a sequence $\langle \mathcal{F}_1, \dots, \mathcal{F}_n \rangle \in S$ of an effective decomposable set S is one where each \mathcal{F}_{i+1} is formed from \mathcal{F}_i by a transformation that preserves enough information to allow effective reconstruction of an optimal solution. For example, a set containing $\langle \mathcal{F}, \mathcal{F}_F \rangle$ for all instances \mathcal{F} is clearly sufficient. However, it is not effective as no “useful” information is preserved when transforming \mathcal{F} to \mathcal{F}_F (in most cases).

Finally, *generality* allows for comparing sets of sequences in a natural way.

Definition 7. Let S_1 and S_2 be two sets of sequences of instances. We say that

- (i) S_1 is at least as general as S_2 if $S_2 \subset S_1$;
- (ii) S_2 is not as general as S_1 if $S_1 \setminus S_2 \neq \emptyset$; and
- (iii) S_1 is more general than S_2 if S_1 is at least as general and S_2 is not as general.

4 Overview of Results

Figure 1 gives an overview of our main results. Considering previously proposed types of instance transformations, we establish that the sets MSRED and MCSEQ are individually not expressive enough to be sufficient within our generic framework, while their combination MSRED \circ MCSEQ is. Indeed, MSRED and MCSEQ are orthogonal in the sense that neither one is as general as the other; we will give concrete examples of sequences $\langle \mathcal{F}, \mathcal{F}^R \rangle \in \text{MSRED} \setminus \text{MCSEQ}$ and $\langle \mathcal{F}, \mathcal{F}^R \rangle \in \text{MCSEQ} \setminus \text{MSRED}$. Thus, neither one of these previously proposed formalisms for MaxSAT solving techniques is expressive enough to capture hybrid forms of MaxSAT solving that combine the core-guided approach with preprocessing-style reasoning.

In addition to MSRED and MCSEQ, we will also consider several other sets of sequences. The set MSEQUIV in Figure 1 contains sequences where subsequent instances are MaxSAT-equivalent with each other. MaxSAT-equivalence has previously been shown to be a special case of MaxSAT resolution [10, 2] and included in this work mainly for completeness. We also propose two new sets of sequences, MAX-RAT and CSRED, with different motivations. As detailed in Section 6, we propose MAX-RAT as a natural lifting of the notion of resolution asymmetric tautology, which has been shown to give a basis for formalizing

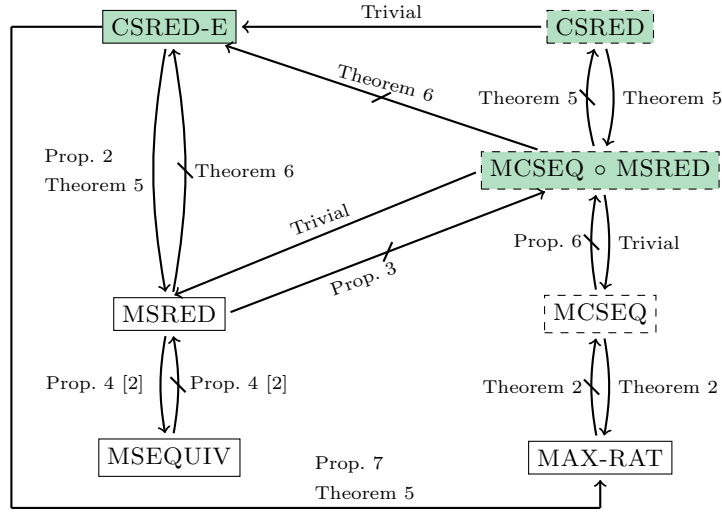


Fig. 1. Relationships between sets of sequences. Here $S_1 \rightarrow S_2$ denotes that S_1 is at least as general as S_2 , $S_1 \not\rightarrow S_2$ that S_1 is not as general as S_2 . Transitive edges are not shown. The types of transformations that are sufficient for reaching the final state are coloured green, and effective transformations are drawn with continuous lines.

inprocessing SAT solving [17], to the context of MaxSAT. As a main result, we show that MAX-RAT yields a general proof of correctness of liftings of SAT preprocessing techniques to MaxSAT, noticeably generalizing earlier correctness proofs of liftings of specific SAT preprocessing techniques [5]. Towards more general instance transformations, we also propose the notion of correction-set reducibility, CSRED containing sequences in which each instance is correction-set reducible to the next one. We show that CSRED surpasses the combination of MSRED and MCSEQ; even its effective subset CSRED-E captures essentially all current core-guided and preprocessing-style transformations we are aware of.

5 Analysis of Known Transformations

We begin the detailed analysis with MaxSAT-reducibility and MCS-equivalence, their combination, and MaxSAT-equivalence.

5.1 MaxSAT-Reducibility

First, we show that MSRED is not sufficient for reaching the final state. Informally, the result follows from the fact that MaxSAT-reducibility requires preserving all of the solutions to an instance while not being expressive enough to affect the cost of different solutions in different ways. Hence any instance \mathcal{F} which has two solutions τ^1 and τ^2 such that $\text{COST}(\mathcal{F}, \tau^1) \neq \text{COST}(\mathcal{F}, \tau^2)$ is

not MaxSAT-reducible to the final instance \mathcal{F}_F to which all solutions have the same cost. We generalize this argument to sequences to establish the following.

Proposition 1. *MSRED is not sufficient for reaching the final state.*

Note that in practice core-guided solvers that use MaxSAT-reducible transformations terminate once the cost of the working instance becomes 0, which is not the same as the working instance being \mathcal{F}_F . This “contrast” arises from the fact that core-guided solvers rely on a SAT solver for termination.

The effectiveness of MSRED follows by showing that τ_2 , an optimal solution to \mathcal{F}_2 in $\langle \mathcal{F}_1, \mathcal{F}_2 \rangle \in \text{MSRED}$ satisfies \mathcal{F}_1 and assigns all variables in $\mathcal{S}(\mathcal{F}_1)$. Thus an optimal solution to \mathcal{F}_1 can be obtained by i) restricting τ_2 to $\text{LIT}(\mathcal{F}_1)$ and ii) assigning any unassigned variables of $\text{VAR}(\mathcal{F}_1)$ arbitrarily.

Proposition 2. *MSRED is effective.*

The next proposition implies that MaxSAT-reducibility can not be used as basis for reasoning about the correctness of MaxSAT preprocessing.

Proposition 3. *MSRED is not as general as MCSEQ.*

Proof. Consider the sequence $\langle \mathcal{F}, \mathcal{F}_f \rangle$, where $\mathcal{F} = \{(l \vee x)\}$ with $w^{\mathcal{F}}(l) = 1$ and $w^{\mathcal{F}}(x) = 0$. Since $\text{COST}(\mathcal{F}) = 0$ it follows that $\text{MCS}(\mathcal{F}) = \{\emptyset\} = \text{MCS}(\mathcal{F}_F)$. This implies that $\langle \mathcal{F}, \mathcal{F}_f \rangle \in \text{MCSEQ}$. To see that $\langle \mathcal{F}, \mathcal{F}_f \rangle \notin \text{MSRED}$, consider the solutions $\tau^1 = \{l, \neg x\}$ and $\tau^2 = \{\neg l, x\}$ to \mathcal{F} . Since $\text{COST}(\mathcal{F}, \tau^1) = 1 = 0+1 = \text{COST}(\mathcal{F}_F, \tau^1)+1$ while $\text{COST}(\mathcal{F}, \tau^2) = 0 = 0+0 = \text{COST}(\mathcal{F}_F, \tau^2)+0$, it follows that \mathcal{F} is not MaxSAT-reducible (for any k) to \mathcal{F}_F . \square

Finally, *MaxSAT-equivalence* is a special case of MaxSAT-reducibility [10, 2].

Definition 8. *The instance \mathcal{F} is MaxSAT equivalent to the instance \mathcal{F}^R if $\text{COST}(\mathcal{F}, \tau) = \text{COST}(\mathcal{F}^R, \tau) + k$ for some positive integer k and all complete truth assignments τ for both \mathcal{F} and \mathcal{F}^R .*

Again, the constant k may depend on \mathcal{F} but not on the particular assignment τ . The set MSEQUIV contains all sequences of MaxSAT instances where subsequent instances are MaxSAT equivalent. MaxSAT-reducibility subsumes MaxSAT-equivalence in terms of the generality of MSRED and MSEQUIV, which follows from comparing the definitions. The following result was first shown in [2] and included here for completeness.

Proposition 4 (Restated from [2]). *The set MSRED is more general than the set MSEQUIV.*

5.2 MCS-Equivalence

Similarly to MSRED, MCSEQ is not sufficient for reaching the final state. The result follows by noting that any terminating sequence $\langle \mathcal{F}, \dots, \mathcal{F}_F \rangle$ containing an instance \mathcal{F}_i for which $\text{MCS}(\mathcal{F}_i) \neq \{\emptyset\} = \text{MCS}(\mathcal{F}_F)$ can not be in MCSEQ.

Proposition 5. *MCSEQ is not sufficient for reaching the final state.*

We expect MCSEQ not to be effective. For some intuition, consider a sequence $\langle \mathcal{F}, \mathcal{F}_2 \rangle \in \text{MCSEQ}$. In the general case, the only information obtainable from an optimal solution τ_2 to \mathcal{F}_2 is an optimal MCS $H \in \text{MCS}(\mathcal{F})$. In this case, reconstructing an optimal solution to \mathcal{F} requires computing a satisfying assignment to $\mathcal{F}|_{\neg(H^c)}$.

We show that MCS-equivalence can not be used in order to reason about the correctness of core-guided solving, i.e., that MCSEQ is not as general as MSRED. Informally, the result follows by noting that $\text{COST}(\mathcal{F}) = \text{COST}(\mathcal{F}_2)$ for any MCS-equivalent instances \mathcal{F}_1 and \mathcal{F}_2 . In contrast, there are sequences $\langle \mathcal{F}_1, \dots, \mathcal{F}_n \rangle \in \text{MSRED}$ for which $\text{COST}(\mathcal{F}_1) > \text{COST}(\mathcal{F}_n)$.

Proposition 6. *MCSEQ is not as general as MSRED.*

5.3 Combining MSRED and MCSEQ

So far, we have established that neither MSRED nor MCSEQ is individually sufficient for reaching the final state, and that neither one is as general as the other. The reasons for insufficiency, however, are in a sense orthogonal. While there are sequences $\langle \mathcal{F}_1, \dots, \mathcal{F}_n \rangle \in \text{MSRED}$ for which $\text{COST}(\mathcal{F}_1) > \text{COST}(\mathcal{F}_n)$, any solution to \mathcal{F}_1 is also a solution to \mathcal{F}_n . In other words, MaxSAT-reducibility can lower the optimal cost of instances but not rule out non-optimal solutions. In contrast, while $\text{COST}(\mathcal{F}_i) = \text{COST}(\mathcal{F}_j)$ for any two instances in a sequence $\langle \mathcal{F}_1, \dots, \mathcal{F}_n \rangle \in \text{MCSEQ}$, there can be solutions to \mathcal{F}_1 that are not solutions to \mathcal{F}_n . More informally, MCS-equivalence can be used to rule out non-optimal solutions, but not to lower the optimal cost of instances. Since a terminating sequence starting from an arbitrary instance \mathcal{F} requires both lowering the optimal cost of \mathcal{F} to 0 and ruling out non-optimal solutions, using both MSRED and MCSEQ obtains a sufficient set of sequences.

Theorem 1. *MSRED \circ MCSEQ is sufficient for reaching the final state.*

Proof. (Sketch) Let \mathcal{F} be a MaxSAT instance. By correctness of the PMRES algorithm discussed in Example 1 [29], there is a sequence $\langle \mathcal{F}, \dots, \mathcal{F}_S \rangle \in \text{MSRED}$ for which $\text{COST}(\mathcal{F}_S) = 0$ and hence $\text{MCS}(\mathcal{F}_S) = \{\emptyset\}$. The terminating sequence $\langle \mathcal{F}, \dots, \mathcal{F}_S, \mathcal{F}_F \rangle \in \text{MSRED} \circ \text{MCSEQ}$ witnesses the claim. \square

6 RAT Clauses in MaxSAT

We propose and analyze two novel notions for transforming MaxSAT instances and their corresponding sets of sequences, MAX-RAT and CSRED. First we adapt the idea of resolution asymmetric tautologies (RAT) to MaxSAT to obtain MAX-RAT, an effective subset of MCSEQ. RAT is a simple yet powerful notion for characterizing preprocessing and inprocessing in SAT solving [17] which

provides a basis for a general proof of correctness for natural liftings of SAT preprocessing techniques that can be expressed as sequences adding and removing RAT clauses.

Given two clauses $C = l \vee C'$ and $D = \neg l \vee D'$, the resolvent $C \bowtie_l D = C' \vee D'$ of C and D is obtained by *resolving* them on l . Resolution is extended to sets of clauses by $\text{CL}(l) \bowtie_l \text{CL}(\neg l) = \{C \bowtie_l D \mid C \in \text{CL}(l), D \in \text{CL}(\neg l)\}$. Let \mathcal{F} be a MaxSAT instance and C a clause. Denote by $\text{ALA}(\mathcal{F}, C)$ (asymmetric literal addition) the unique clause resulting from repeating the following until fixpoint: if $l_1, \dots, l_k \in C$ and there is a clause $(l_1 \vee \dots \vee l_k \vee l) \in \mathcal{F} \setminus \{C\}$, set $C := C \cup \{\neg l\}$. We say that C has *solution resolution asymmetric tautology* (SRAT) with respect to \mathcal{F} if either $C = (x \vee \neg x)$ for some variable x or there is a literal $l \in C \setminus \mathcal{S}(\mathcal{F})$ such that $\text{ALA}(\mathcal{F}, C \bowtie_l D)$ is a tautology for all clauses $D \in \text{CL}(\neg l)$. In the second case, we say that C has SRAT on l . When \mathcal{F} is clear from context, we say that C is an SRAT clause. We emphasize that the only restriction put on l is that it is not a soft variable. Specifically, l can still be the negation of a soft variable. Notice also that any clause C that has SRAT with respect to an instance F also has RAT as defined in [17] with respect to the CNF formula $\text{CLAUSES}(\mathcal{F})$ but the converse is not true. We use the terms SRAT clause to refer to the concept for MaxSAT as defined here, and RAT clause to refer to the SAT-specific concept from [17].

Given a MaxSAT instance \mathcal{F} and an SRAT clause C , the instance $\text{ADD}(\mathcal{F}, C)$ is obtained by (i) adding C to \mathcal{F} and (ii) extending $w^{\mathcal{F}}$ by setting the weight of each variable (if any) in $\text{VAR}(C) \setminus \text{VAR}(\mathcal{F})$ arbitrarily. Similarly, the instance $\text{REMOVE}(\mathcal{F}, C)$ is obtained by (i) removing C from \mathcal{F} and (ii) restricting $w^{\mathcal{F}}$ onto $\text{VAR}(\mathcal{F} \setminus \{C\})$. These transformations are lifted to sequences of instances by the set MAX-RAT.

Definition 9. *The set MAX-RAT contains all sequences $\langle \mathcal{F}_1, \dots, \mathcal{F}_n \rangle$, where $\mathcal{F}_{i+1} = \text{ADD}(\mathcal{F}_i, C)$ or $\mathcal{F}_{i+1} = \text{REMOVE}(\mathcal{F}_i, C)$ for all $i = 1, \dots, n - 1$.*

While simple, the sequences in MAX-RAT are fairly expressive. As discussed in [17], SAT solving techniques, including forms of preprocessing, can very generally be viewed as sequences of adding and removing RAT clauses and can thus be easily lifted to MaxSAT with SRAT.

Example 2. Let \mathcal{F} be an instance, $x \in \text{VAR}(\mathcal{F}) \setminus \mathcal{S}(\mathcal{F})$, $\text{CL}(x) \cup \text{CL}(\neg x) = \{C_1, \dots, C_n\}$ and $\text{CL}(x) \bowtie_x \text{CL}(\neg x) = \{D_1, \dots, D_t\}$. The well-known variable elimination rule allows eliminating x from \mathcal{F} by transforming it to $\mathcal{F}^R = \mathcal{F} \setminus (\text{CL}(x) \cup \text{CL}(\neg x)) \cup (\text{CL}(x) \bowtie_x \text{CL}(\neg x))$. This corresponds to the sequence $\langle \mathcal{F} = \mathcal{F}_0, \mathcal{F}_1, \dots, \mathcal{F}_t, \dots, \mathcal{F}_{t+n} \rangle$ with $\mathcal{F}_i = \text{ADD}(\mathcal{F}_{i-1}, D_i)$ for $i = 1, \dots, t$ and $\mathcal{F}_i = \text{REMOVE}(\mathcal{F}_{i-1}, C_{i-t})$ for $i = t + 1, \dots, t + n$. This sequence is in MAX-RAT.

Both $\text{ADD}(\mathcal{F}, C)$ and $\text{REMOVE}(\mathcal{F}, C)$ are MCS-equivalent with \mathcal{F} ; this is because soft variables of \mathcal{F} contained only in C are not members of any MCSes of \mathcal{F} .

Theorem 2. *MCSEQ is more general than MAX-RAT.*

In addition, generalizing results from [17], we show that MAX-RAT is effective. Given an instance \mathcal{F} and a clause C that has SRAT with respect to a literal l , we show that if τ is a solution to \mathcal{F} but not $\text{ADD}(\mathcal{F}, C)$, then $\tau^R = \tau \setminus \{\neg l\} \cup \{l\}$ is a solution to both \mathcal{F} and $\text{ADD}(\mathcal{F}, C)$. While this also holds for RAT clauses [17], the added assumption $l \notin \mathcal{S}(\mathcal{F})$ of SRAT is needed to show $\text{COST}(\mathcal{F}, \tau) \geq \text{COST}(\mathcal{F}, \tau^R) = \text{COST}(\text{ADD}(\mathcal{F}, C), \tau^R)$, which in turn implies the existence of effective reconstruction functions for sequences in MAX-RAT.

Proposition 7. *MAX-RAT is effective.*

More generally, MAX-RAT provides a natural way of correctly lifting all of the preprocessing rules proposed for SAT solving that can be modelled with RAT based transformations to MaxSAT; this noticeably generalizes correctness proofs of the four particular liftings considered in [5].

Theorem 3. *Let \mathcal{F} be a MaxSAT instance and \mathcal{F}^R the instance obtained by applying a SAT preprocessing technique that can be viewed as a sequence $\langle \mathcal{F}, \dots, \mathcal{F}^R \rangle$ of RAT clause additions and deletions. Assume that all of the added and removed clauses also have SRAT. Then an optimal solution to \mathcal{F} can be effectively computed based on an optimal solution to \mathcal{F}^R .*

RAT and, by extension, SRAT are very powerful concepts, allowing for simulating essentially all SAT preprocessing techniques, including both resolution-based and clause elimination techniques [17]. Hence Theorem 3 gives a very general proof of correctness for natural liftings of SAT-based preprocessing techniques to MaxSAT.

The MAX-RAT sequences also detail the relationship between core-guided MaxSAT solvers and the abstract model of MaxSAT solving that we work with. Since the transformations in MAX-RAT can model SAT solving [17], the abstract state transition system models both the transformations employed by the outer core-guided MaxSAT algorithm, and the internal SAT solver used by it to extract cores and compute satisfying assignments. For example, the soundness of keeping learned clauses of the internal SAT solver between iterations follows easily from the fact that a SAT solver only learns clauses that have SRAT with respect to the current instance. Therefore the combination of MAX-RAT and MSRED captures the correctness of core-guided algorithms and their interleavings with liftings of SAT-based preprocessing techniques.

Theorem 4. *MSRED \circ MAX-RAT is effective and sufficient for reaching the final state.*

However, as we will discuss next, there are instance transformations, some of which arise from MaxSAT-specific preprocessing techniques without counterparts in SAT preprocessing, that are not captured by MSRED \circ MAX-RAT. This motivates the study of more expressive instance transformations.

7 Correction Set Reducible Transformations

The second novel notion for transforming MaxSAT instances that we propose is correction set reducibility (CS-reducibility). CS-reducibility is a more general form of instance transformations, surpassing the combination of MaxSAT-reducibility and MCS-equivalence, and thereby providing a wide basis for reasoning about the correctness of MaxSAT solving.

Definition 10. *The instance \mathcal{F} is correction set reducible (CS-reducible) to the instance \mathcal{F}^R if $U(\mathcal{F}, \tau^R) = \tau^R \cap \mathcal{S}(\mathcal{F})$ is an optimal MCS of \mathcal{F} whenever τ^R is an optimal solution to \mathcal{F}^R*

Example 3. Let $\mathcal{F} = \{(l_1 \vee l_2)\}$ and $\mathcal{F}^R = \{(l_1 \vee l_2), (\neg l_2)\}$ be two instances with $w^{\mathcal{F}}(l_1) = w^{\mathcal{F}^R}(l_1) = 1$ and $w^{\mathcal{F}}(l_2) = w^{\mathcal{F}^R}(l_2) = 2$. Then \mathcal{F} is CS reducible to \mathcal{F}^R which follows from $\tau^R = \{l_1, \neg l_2\}$ being the only optimal solution of \mathcal{F}^R and $\tau^R \cap \mathcal{S}(\mathcal{F}) = \{l_1\}$ being an optimal MCS of \mathcal{F} .

Similarly to other transformations, let CSRED be the set of sequences that contains all sequences $\langle \mathcal{F}_1, \dots, \mathcal{F}_n \rangle$ for which \mathcal{F}_i is CS-reducible to \mathcal{F}_{i+1} for $i = 1, \dots, n-1$. In contrast to MaxSAT-reducibility, CS-reducibility does not require uniformly changing costs of all assignments. This allows transformations that rule out non-optimal solutions. In contrast to MCS-equivalence, CS-reducibility only requires that an optimal solution to the transformed instance corresponds to an optimal MCS of the original instance; an optimal MCS of the original instance does not have to be an MCS of the transformed instance nor do all MCSes of the instance need to be preserved. This allows transformations that lower the optimal cost of instances.

Theorem 5. *CSRED is more general than MSRED \circ MCSEQ.*

Notice that Theorem 5 also implies that CSRED is sufficient for reaching the final state.

As CSRED is at least as general as MCSEQ we do not expect it to be effective. However, CSRED-E, the effective subset of CSRED, is in itself relatively expressive.

Theorem 6. *CSRED-E is sufficient for reaching the final state. Furthermore, MSRED \circ MCSEQ, is not as general as CSRED-E.*

Proof. The first claim follows directly from Theorems 4 and 5. For the second claim, consider the sequence $\langle \mathcal{F}, \mathcal{F}^R \rangle$ formed by the instances defined in Example 3. The claim $\langle \mathcal{F}, \mathcal{F}^R \rangle \in \text{CSRED-E}$ follows from the fact that the sets of optimal solutions of \mathcal{F} and \mathcal{F}^R are equal. To show that $\langle \mathcal{F}, \mathcal{F}^R \rangle \notin \text{MSRED} \circ \text{MCSEQ}$ we demonstrate that \mathcal{F} is neither MaxSAT-reducible to, or MCS-equivalent with, \mathcal{F}^R . The former follows by considering the solution $\tau = \{l_2, \neg l_1\}$ to \mathcal{F} and the latter from $\text{MCS}(\mathcal{F}) = \{\{l_1\}, \{l_2\}\} \neq \{\{l_1\}\} = \text{MCS}(\mathcal{F}^R)$. \square

Note that Theorem 6 implies that $\text{MSRED} \circ \text{MAX-RAT}$ is not as general as CSRED-E and that the sequence $\langle \mathcal{F}, \mathcal{F}^R \rangle$ corresponds to applying the MaxSAT-specific preprocessing of subsumed label elimination [8] on \mathcal{F} . Thus effective CS-reducibility captures existing MaxSAT preprocessing techniques not captured by $\text{MSRED} \circ \text{MAX-RAT}$.

8 Related Work

In terms of previously proposed formal systems for MaxSAT, MaxSAT resolution [10, 20] was proposed as a generalization of the resolution proof system. It is a complete rule for MaxSAT in that iteratively applying MaxSAT resolution to the clauses of a MaxSAT instance \mathcal{F} gives another instance \mathcal{F}^{RES} such that $\text{COST}(\mathcal{F}^{\text{RES}}) = 0$ and any satisfying assignment to $\mathcal{F}^S|_{\neg S(\mathcal{F}^S)}$ is an optimal solution to \mathcal{F} . The correctness of MaxSAT resolution was shown by establishing that \mathcal{F} is MaxSAT-equivalent to \mathcal{F}^{RES} . As implied by our analysis, this means that MaxSAT resolution can be used to determine the optimal cost of an instance, but finding an optimal solution requires computing a satisfying assignment to a satisfiable CNF formula. While MaxSAT resolution and its restrictions give rise to simplification rules used in conjunction with branch-and-bound MaxSAT solvers [4, 16, 21, 22] (and also yields a proof system for SAT [9]), we focus on the current state-of-the-art core-guided approaches and central SAT-based preprocessing techniques. (The recent clause tableaux proof system for MaxSAT [23] does not capture core-guided transformations or MaxSAT preprocessing, either.) The recent formalization of implicit hitting set (IHS) algorithms for optimization modulo theories of [13] captures solely extensions of the IHS approach [12, 32], extending the DPLL(T) [31] which has also earlier been extended to optimization modulo theories [30] and adapted for formalizing answer set solvers [11, 24] (without optimization statements).

9 Conclusions

We studied the formal underpinnings of unifying preprocessing-style reasoning and core-guided transformations for MaxSAT. To this end, we formalized a generic framework for MaxSAT solving based on sequences of MaxSAT instances, and analyzed previously proposed notions of instance transformations underlying core-guided search and SAT-based preprocessing for MaxSAT within the framework. We showed that these notions individually do not capture each other (i.e., inprocessing core-guided MaxSAT solving), and therefore neither can be used for arguing about the correctness of the other. We proposed correction set reducibility as a new type of MaxSAT instance transformation which unifies core-guided MaxSAT solving and MaxSAT preprocessing, including SAT-based preprocessing lifted to MaxSAT. Furthermore, we generalized the concept of resolution asymmetric tautologies from SAT solving to MaxSAT solving, thereby

obtaining a very general proof of correctness for lifting SAT preprocessing techniques to MaxSAT. All in all, the results build ground for generic techniques for arguing about the formal correctness of MaxSAT algorithms.

Acknowledgments

The work has been financially supported by Academy of Finland (grants 276412 and 312662) and University of Helsinki Doctoral Programme in Computer Science.

References

1. Alviano, M., Dodaro, C., Ricca, F.: A MaxSAT algorithm using cardinality constraints of bounded size. In: Proc. IJCAI. pp. 2677–2683. AAAI Press (2015)
2. Ansótegui, C., Bonet, M., Levy, J.: SAT-based MaxSAT algorithms. *Artificial Intelligence* **196**, 77–105 (2013)
3. Ansótegui, C., Didier, F., Gabàs, J.: Exploiting the Structure of Unsatisfiable Cores in MaxSAT. In: Proc. IJCAI. pp. 283–289. AAAI Press (2015)
4. Argelich, J., Li, C.M., Manyà, F.: A preprocessor for Max-SAT solvers. In: Proc. SAT. Lecture Notes in Computer Science, vol. 4996, pp. 15–20. Springer (2008)
5. Belov, A., Morgado, A., Marques-Silva, J.: SAT-based preprocessing for MaxSAT. In: Proc. LPAR-19. Lecture Notes in Computer Science, vol. 8312, pp. 96–111. Springer (2013)
6. Benedetti, M., Mori, M.: Parametric RBAC maintenance via Max-SAT. In: Proc. SACMAT. pp. 15–25. ACM (2018)
7. Berg, J., Järvisalo, M.: Cost-optimal constrained correlation clustering via weighted partial maximum satisfiability. *Artificial Intelligence* **244**, 110–143 (2017)
8. Berg, J., Saikko, P., Järvisalo, M.: Subsumed label elimination for maximum satisfiability. In: Proc. ECAI. *Frontiers in Artificial Intelligence and Applications*, vol. 285, pp. 630–638. IOS Press (2016)
9. Bonet, M.L., Buss, S., Ignatiev, A., Marques-Silva, J., Morgado, A.: MaxSAT resolution with the dual rail encoding. In: AAAI. AAAI Press (2018)
10. Bonet, M.L., Levy, J., Manyà, F.: Resolution for Max-SAT. *Artificial Intelligence* **171**(8-9), 606–618 (2007)
11. Brochenin, R., Maratea, M., Lierler, Y.: Disjunctive answer set solvers via templates. *Theory and Practice of Logic Programming* **16**(4), 465–497 (2016)
12. Davies, J., Bacchus, F.: Exploiting the power of MIP solvers in MaxSAT. In: Proc. SAT. Lecture Notes in Computer Science, vol. 7962, pp. 166–181. Springer (2013)
13. Fazekas, K., Bacchus, F., Biere, A.: Implicit hitting set algorithms for maximum satisfiability modulo theories. In: Proc. IJCAR. Lecture Notes in Computer Science, vol. 10900, pp. 134–151. Springer (2018)
14. Feng, Y., Bastani, O., Martins, R., Dillig, I., Anand, S.: Automated synthesis of semantic malware signatures using maximum satisfiability. In: Proc. NDSS. The Internet Society (2017)
15. Heras, F., Morgado, A., Marques-Silva, J.: Core-Guided Binary Search Algorithms for Maximum Satisfiability. In: Proc. AAAI. AAAI Press (2011)

16. Heras, F., Larrosa, J., Oliveras, A.: MiniMaxSAT: An efficient weighted Max-SAT solver. *Journal of Artificial Intelligence Research* **31**, 1–32 (2008)
17. Järvisalo, M., Heule, M., Biere, A.: Inprocessing rules. In: Proc. IJCAR. *Lecture Notes in Computer Science*, vol. 7364, pp. 355–370. Springer (2012)
18. Jose, M., Majumdar, R.: Cause clue clauses: error localization using maximum satisfiability. In: Proc. PLDI. pp. 437–446. ACM (2011)
19. Korhonen, T., Berg, J., Saikko, P., Järvisalo, M.: MaxPre: An extended MaxSAT preprocessor. In: Proc. SAT. *Lecture Notes in Computer Science*, vol. 10491, pp. 449–456 (2017)
20. Larrosa, J., Heras, F.: Resolution in Max-SAT and its relation to local consistency in weighted CSPs. In: Proc. IJCAI. pp. 193–198. Professional Book Center (2005)
21. Larrosa, J., Heras, F., de Givry, S.: A logical approach to efficient Max-SAT solving. *Artificial Intelligence* **172**(2-3), 204–233 (2008)
22. Li, C.M., Manyà, F., Mohamedou, N.O., Planes, J.: Resolution-based lower bounds in MaxSAT. *Constraints* **15**(4), 456–484 (2010)
23. Li, C.M., Manyà, F., Soler, J.R.: A clause tableau calculus for MinSAT. In: Proc. CCIA. *Frontiers in Artificial Intelligence and Applications*, vol. 288, pp. 88–97. IOS Press (2016)
24. Lierler, Y., Truszczynski, M.: On abstract modular inference systems and solvers. *Artificial Intelligence* **236**, 65 – 89 (2016)
25. Marques-Silva, J., Janota, M., Ignatiev, A., Morgado, A.: Efficient model based diagnosis with maximum satisfiability. In: Proc. IJCAI. pp. 1966–1972. AAAI Press (2015)
26. Morgado, A., Dodaro, C., Marques-Silva, J.: Core-Guided MaxSAT with Soft Cardinality Constraints. In: Proc. CP. *Lecture Notes in Computer Science*, vol. 8656, pp. 564–573. Springer (2014)
27. Morgado, A., Heras, F., Liffiton, M., Planes, J., Marques-Silva, J.: Iterative and core-guided MaxSAT solving: A survey and assessment. *Constraints* **18**(4), 478–534 (2013)
28. Nadel, A., Rychin, V.: Efficient SAT solving under assumptions. In: Proc. SAT. LNCS, vol. 7317, pp. 242–255. Springer (2012)
29. Narodytska, N., Bacchus, F.: Maximum satisfiability using core-guided MaxSAT resolution. In: Proc. AAAI. pp. 2717–2723. AAAI Press (2014)
30. Nieuwenhuis, R., Oliveras, A.: On SAT modulo theories and optimization problems. In: Biere, A., Gomes, C.P. (eds.) Proc. SAT. LNCS, vol. 4121, pp. 156–169. Springer (2006)
31. Nieuwenhuis, R., Oliveras, A., Tinelli, C.: Solving sat and sat modulo theories: From an abstract davis-putnam-logemann-loveland procedure to `dpll(t)`. *Journal of the ACM* **53**, 2006 (2006)
32. Saikko, P., Berg, J., Järvisalo, M.: LMHS: A SAT-IP hybrid MaxSAT solver. In: Proc. SAT. *Lecture Notes in Computer Science*, vol. 9710, pp. 539–546. Springer (2016)
33. Wallner, J.P., Niskanen, A., Järvisalo, M.: Complexity results and algorithms for extension enforcement in abstract argumentation. *Journal Artificial Intelligence Research* **60**, 1–40 (2017)
34. Zhu, C., Weissenbacher, G., Malik, S.: Post-silicon fault localisation using maximum satisfiability and backbones. In: Proc. FMCAD. pp. 63–66. FMCAD Inc. (2011)