On Computing Maximal Subsets of Clauses that Must Be Satisfiable with Possibly Mutually-Contradictory Assumptive Contexts

Philippe Besnard
IRIT
CNRS UMR 5505
118 Route de Narbonne
F-31062 Toulouse, France
besnard@irit.fr

Éric Grégoire and **Jean-Marie Lagniez**
CRIL
Université d'Artois & CNRS
rue Jean Souvraz SP18
F-62307 Lens, France
{gregoire,lagniez}@cril.fr

Abstract

An original method for the extraction of one maximal subset of a set of Boolean clauses that must be satisfiable with possibly mutually contradictory assumptive contexts is motivated and experimented. Noticeably, it performs a direct computation and avoids the enumeration of all subsets that are satisfiable with at least one of the contexts. The method applies for subsets that are maximal with respect to inclusion or cardinality.

Introduction

The extraction of maximal satisfiable subsets (MSSes) of a set of Boolean clauses is a key issue in many knowledge representation and reasoning fields. These subsets can be maximal with respect to set-theoretic inclusion or cardinality. The set-theoretic complement of one MSS (noted Co-MSS) is also of importance in many A.I. areas since it is an (inclusion or cardinality)-minimal subset of clauses that should be corrected in order for the whole set of clauses to be conflict-free. Although there can be an exponential number of MSSes in a contradictory set of clauses, the extraction of one MSS can prove useful when a quick decision needs to be taken and when it is acceptable to base this decision on one MSS, only. Moreover, computing one MSS is also the basic kernel of techniques that either enumerate all MSSes (Marques-Silva et al. 2013) when their number is small, or compute some preferred ones. In addition, many algorithms that enumerate minimal proofs of unsatisfiability, or MUSes for (inclusion)-Minimal Unsatisfiable Sets, in the clausal Boolean framework (Liffiton and Sakallah 2008a; Grégoire, Mazure, and Piette 2007), rely on the enumeration of MSSes. MUSes have also many applications in their own right.

In this respect, techniques for extracting one MSS that prove experimentally viable in many actual situations are thus of a wide scope interest.

In this paper, we investigate the more general situation where the maximal satisfiable subset to be extracted must be conflict-free with a series of potential cases, called assumptive contexts, that are under the form of (possibly mutually contradictory) Boolean formulas. These assumptive contexts are mutually conflicting when they express incompatible possible branches of an alternative. Should any of the assumptive contexts turn out to be true, the (cardinality or inclusion) MSS would not get contradicted. Such a context is widespread in A.I. and everyday life. A few examples:

- *Planning.* A robot needs to plan its next moves. It must take its decisions based on its own knowledge Δ and two newly downloaded maps. Although each map is itself conflict-free, it contains information conflicting with the other map. Since the robot must act quickly and since it assumes that the conflicts inside Δ are of a minor importance, it decides to rely on just one MSS of the set-theoretic union of Δ with the two maps, provided that this MSS is compatible with each map. If the robot is given more time then it computes more MSSes and takes a decision on their global basis.
- *Decision making.* An agent wants to found her decision on the part of her own knowledge and beliefs Δ that does not contradict a given range of possibilities. Since she must decide quickly, she starts by considering one maximal subset of Δ that is satisfiable with each of those possibilities (which again we call assumptive contexts). She is then guaranteed that her decision will not be conflicting with any of the assumptive contexts, should one of them eventually turn out to be true. She feels that conflicts within her own information Δ are of lesser importance, so that she accepts to drop some pieces of information involved in these latter conflicts. Indeed, she regards it more important to take a quick decision that is compatible with each of the assumptive contexts.
- *Model-based diagnosis.* Diagnoses are defined as formulas from a Co-MSS, among other constraints (Hamscher, Console, and de Kleer 1992; DX 2014). Diagnoses can be required to be conflict-free with each member of a set of additional considerations or assumptive contexts that might be mutually conflicting.
- *Reasoning about exceptions and default reasoning.* For example, in default logic (Reiter 1980), when the set of standard logical formulas is consistent, an extension is a

maximal satisfiable set of formulas that, among other additional constraints, are satisfiable together with the so-called *justifications* of the generating default rules for the extension. The *non-commitment to assumptions* property (Delgrande, Schaub, and Jackson 1994) of Reiter's default logic expresses that extensions can be built from generating default rules with mutually conflicting justifications.

- *Argumentation.* In Dung's seminal work on computational argumentation (Dung 1995), one logic-based interpretation of extensions is based on one MSS of a corresponding knowledge base (Vesic 2013). Also, in logic-based argumentation (Besnard and Hunter 2008), it has been shown how arguments can be directly computed from MSSes (Besnard et al. 2010). More generally, arguments can be required to be compatible with some additional pieces of information that might be mutually conflicting, like an argument showing that the defendant is guilty and that is compatible with each of two cases: either the defendant is telling the truth or lying.
- *Belief change.* The extraction of maximal subsets of beliefs that are satisfiable with some additional constraints corresponds to the so-called multiple contraction (Fuhrmann and Hansson 1994) paradigm in belief change. Note also that following Levi's identity (Levi 1977), contraction is a basic building block of belief revision, too (Fermé and Hansson 2011; Alchourrón, Gärdenfors, and Makinson 1985): revising a deductive closed set of beliefs by an incoming belief δ involves the contraction of the set by the negation of δ, which is thus based on the extraction of MSSes that are satisfiable with δ, followed by an expansion by δ, which includes deductive closure.

In view of these applications, the contribution of this paper is as follows. We introduce and experimentally investigate a method to compute one maximal subset of a set of Boolean clauses Δ such that this subset is satisfiable with each member of a set of possibly mutually conflicting Boolean formulas Γ. Actually, it is not a straightforward issue to transform an experimentally efficient method that computes one MSS into a practical technique that extracts one maximal satisfiable subset that is compatible with each member of Γ. Let us elaborate on this.

Assume first that (1) we have selected an experimentally efficient algorithm that computes one MSS of a set of Boolean clauses Δ, (2) we are given an unsatisfiable set of of Boolean formulas Γ that represents a set of assumptive contexts that are mutually conflicting, (3) formulas from Γ can contradict Δ and (4) Δ can be unsatisfiable. Let us write Clausal(γ) for a set of clauses, equivalent to γ with respect to satisfiability, that is obtained through usual standard rewriting procedures.

One naive method would be as follows: for each $\gamma \in \Gamma$, compute one MSS of $\Delta \cup \{\text{Clausal}(\gamma)\}$ that contains Clausal(γ) and its corresponding Co-MSS, which we denote Ψ_γ. Clearly, $\Delta^* = \Delta \setminus \bigcup_{\gamma\in\Gamma}\{\Psi_\gamma\}$ is satisfiable with every formula γ of Γ. However, Δ^* is not necessarily an intended solution since there might exist a strict superset of Δ^* that is also satisfiable with every formula of Γ. Indeed, we might have dropped two different clauses from Δ when considering two formulas γ_1 and γ_2 of Γ, whereas it is actually enough to drop one clause, only. An illustration of this phenomenon is Example 2 below. Clearly, this drawback holds for both cardinality and inclusion-maximal MSSes. A natural but highly intractable solution to this problem requires the following preliminary step: for each γ, compute every inclusion-maximal MSS of Δ that does not conflict with γ. From this set of MSSes, compute the intended solution.

In this paper, we propose and investigate an approach that avoids the computation of these intermediate MSSes and the corresponding possible computational blow-up. To some extent, the approach pertains to the family of techniques that ensure robust optimal solutions for all possible scenarios (Ben-Tal, Ghaoui, and Nemirovski 2009).

Preliminaries and Technical Background

We consider standard Boolean logic: let $\mathcal{L}$ be a language of formulas over an alphabet $\mathcal{P}$ of Boolean variables, also called *atoms*. Atoms are denoted by $a, b, c, \ldots$ The symbols $\wedge, \vee, \neg, \Rightarrow$ and $\Leftrightarrow$ represent the standard conjunctive, disjunctive, negation, material implication and equivalence connectives, respectively. Formulas are built in the usual way from atoms, connectives and parentheses; they are denoted by $\alpha, \beta, \gamma, \ldots$ Sets of formulas are denoted by $\Phi, \Gamma, \ldots$ The cardinality of a set of formulas Γ is written card(Γ). A *literal* is an atom or a negated atom. A *clause* is a disjunction of literals. A unit clause is formed of one literal.

A set of formulas Γ is *satisfiable* iff there exists at least one model of Δ, namely a truth assignment of all atoms of Γ making all formulas of Γ to be *true* according to usual compositional rules. Any formula γ can be represented as a set of clauses, denoted Clausal(γ), equivalent to γ with respect to satisfiability, that is obtained through usual standard rewriting procedures.

SAT is the NP-complete problem that consists in checking whether a finite set of clauses is satisfiable.

From now on, we assume that Δ is a finite set of clauses, that Φ and Ψ are subsets of Δ and that Γ is a finite set of Boolean formulas, each of the formulas in Γ being (individually) satisfiable. As a shortcut, γ will be identified with Clausal(γ) for $\gamma \in \Gamma$.

Definition 1. Φ is an inclusion-Maximal Satisfiable Subset of Δ, in short, Φ is an $\text{MSS}_\subseteq(\Delta)$, iff Φ is satisfiable and $\forall\alpha \in \Delta \setminus \Phi$, $\Phi \cup \{\alpha\}$ is unsatisfiable.

Definition 2. Φ is a cardinality-Maximal Satisfiable Subset of Δ, in short, Φ is an $\text{MSS}_\#(\Delta)$, iff Φ is an $\text{MSS}_\subseteq(\Delta)$ and $\nexists\Phi'$ s.t. Φ' is an $\text{MSS}_\subseteq(\Delta)$ and card(Φ) < card(Φ').

A Co-MSS of Δ is the set-theoretic complement in Δ of the corresponding MSS. For ease of notation, we write (Co-)MSS instead of (Co-)$\text{MSS}_\subseteq$ and (Co-)$\text{MSS}_\#$, when the context does not make this ambiguous or when no such distinction is necessary.

Definition 3. Ψ is a Minimal Correction Subset (MCS or Co-MSS) of Δ iff $\Psi = \Delta \setminus \Phi$ where Φ is an MSS of Δ.

Accordingly, Δ can always be partitioned into a pair made of one $\text{MSS}_\subseteq$ and one Co-$\text{MSS}_\subseteq$. Unless $P = NP$, extracting one such partition is intractable in the worst case since it belongs to the $FP^{NP}[wit, log]$ class, i.e., the set of function problems that can be computed in polynomial time by executing a logarithmic number of calls to an NP oracle that returns a witness for the positive outcome (Marques-Silva and Janota 2014). Techniques to compute one such partition that prove very often efficient are described in (Grégoire, Lagniez, and Mazure 2014; Marques-Silva et al. 2013). Note that in the worst case the number of MSSes is exponential in the number of clauses in Δ.

The instance of the Max-SAT problem w.r.t. Δ consists in delivering the cardinality of any $\mathrm{MSS}_{\#}(\Delta)$. In the following, we consider the variant of Max-SAT that not only delivers this cardinality but also one such $\mathrm{MSS}_{\#}(\Delta)$. We also make use of the following variant definition of Partial-Max-SAT.

Definition 4. Let Σ_1 and Σ_2 be two sets of clauses. Partial-Max-SAT(Σ_1, Σ_2) computes one cardinality maximal subset of Σ_1 that is satisfiable with Σ_2. Σ_1 and Σ_2 are called the sets of soft and hard constraints, respectively.

These variants of (Partial-)Max-SAT belong to the $Opt - P$ class of problems (Papadimitriou and Yannakakis 1991), namely the class of functions computable by taking the maximum of the output values over all accepting paths of an NP machine.

Problem Statement and Basic Examples

In this paper, we are interested in maximal satisfiable subsets that are computed under a *set of assumptive contexts*.

Definition 5. Φ is an inclusion-Maximal Satisfiable Subset of Δ under a *set of assumptive contexts* Γ, (Φ is an AC-$\mathrm{MSS}_{\subseteq}(\Delta, \Gamma)$ for short), iff
1. Φ is a satisfiable subset of Δ, and
2. $\forall \gamma \in \Gamma, \Phi \cup \{\gamma\}$ is satisfiable, and
3. $\forall \alpha \in \Delta \setminus \Phi, \Phi \cup \{\alpha, \gamma\}$ is unsatisfiable for some $\gamma \in \Gamma$.

Definition 6. Φ is an AC-$\mathrm{MSS}_{\#}(\Delta, \Gamma)$ iff Φ is an AC-$\mathrm{MSS}_{\subseteq}(\Delta, \Gamma)$ and $\nexists \Phi'$ s.t. Φ' is an AC-$\mathrm{MSS}_{\subseteq}(\Delta, \Gamma)$ and $\mathrm{card}(\Phi) < \mathrm{card}(\Phi')$.

The extraction of one AC-$\mathrm{MSS}(\Delta, \Gamma)$ is the addressed problem in this paper. Note that, as is illustrated in the example about the robot that needs to handle two incoming contradictory maps, it is sometimes required to extract one AC-$\mathrm{MSS}(\Delta, \Gamma)$ where $\Gamma \subseteq \Delta$. Also notice that the elements of Γ are used in a pointwise manner: any AC-$\mathrm{MSS}(\Delta, \Gamma)$ must be satisfiable together with any γ, taken individually. This expresses the requirement that each γ represents an assumptive context and that the conjunction of all the assumptive contexts of Γ is not required to be satisfiable.

Let us give very basic examples showing the difficulty in the use of the computation of $\mathrm{MSS}(\Delta)$ to extract one AC-$\mathrm{MSS}(\Delta, \Gamma)$. First, assume that Γ is satisfiable: this occurs when the assumptive contexts are not conflicting. Even in this simple case, we cannot consider all elements of Γ conjunctively and interpret them as forming one unique global constraint of consistency that must be respected, as the following simple counter-example illustrates.

Example 1. $\Delta = \{a \vee b, d\}$. AC-$\mathrm{MSS}(\Delta, \{\neg a, \neg b\}) = \Delta$ *but* AC-$\mathrm{MSS}(\Delta, \{\neg a \wedge \neg b\}) = \{d\}$.

Indeed, even when the assumptions are not logically mutually conflicting by themselves, they interact with Δ and this influences AC-$\mathrm{MSS}(\Delta, \Gamma)$. To circumvent this issue, as is mentioned in the introduction, it would be tempting to extract for each $\gamma \in \Gamma$ one $\mathrm{MSS}(\Delta \cup \{\gamma\})$ that contains γ; let Ψ_γ be the corresponding Co-MSS. Although $\Delta \setminus \bigcup_{\gamma \in \Gamma} \{\Psi_\gamma\}$ is satisfiable with every formula γ of Γ, it is not guaranteed to be an AC-$\mathrm{MSS}(\Delta, \Gamma)$, even when Γ is satisfiable.

Example 2. *Let* $\Delta = \{\neg a \vee b, \neg b, d\}$ *and* $\Gamma = \{a, b\}$. Γ *is satisfiable. There are two* $\mathrm{MSS}(\Delta \cup \{a\})$ *that contain the subset* $\{a\}$*, namely* $\{\neg a \vee b, d, a\}$ *and* $\{\neg b, d, a\}$ *and one* $\mathrm{MSS}(\Delta \cup \{b\})$ *that contains the subset* $\{b\}$*, namely* $\{\neg a \vee b, d, b\}$*. Clearly, when* $\{\neg b, d, a\}$ *is extracted as one* $\mathrm{MSS}(\Delta \cup \{a\})$*, we have that* $\Delta \setminus \bigcup_{\gamma \in \Gamma} \{\Psi_\gamma\}$ *(where* Ψ_γ *is the Co-MSS corresponding to the selected* $\mathrm{MSS}(\Delta \cup \{\gamma\})$*) yields* $\{d\}$*. However,* $\{d\}$ *is not an* AC-$\mathrm{MSS}(\Delta, \Gamma)$*; there is a unique* AC-$\mathrm{MSS}(\Delta, \Gamma)$*, i.e.,* $\{\neg a \vee b, d\}$.

When Γ is unsatisfiable, this needs not prevent AC-$\mathrm{MSS}(\Delta, \Gamma)$ from existing.

Example 3. *Let* $\Delta = \{\neg a \vee b, \neg b, d\}$ *and* $\Gamma = \{a, b, \neg a\}$. $\{\neg a \vee b, d\}$ *is the unique* AC-$\mathrm{MSS}(\Delta, \Gamma)$.

The correct *direct (brute-force) approach* to extract one AC-$\mathrm{MSS}(\Delta, \Gamma)$ is thus as follows. First, for every $\gamma \in \Gamma$ enumerate *all* $\mathrm{MSS}_{\subseteq}(\Delta \cup \{\gamma\})$ that contain γ. Then, select one corresponding Co-$\mathrm{MSS}_{\subseteq}(\Delta \cup \{\gamma\})$ per γ , denoted Ψ_γ, such that $\Delta^* = \Delta \setminus \bigcup_{\gamma \in \Gamma} \{\Psi_\gamma\}$ exhibits the largest cardinality (when AC-$\mathrm{MSS}_{\#}$ is under consideration) or such that Δ^* is inclusion-maximal (for AC-$\mathrm{MSS}_{\subseteq}$). Δ^* is the result.

Since there can be an exponential number of MSSes in a set of clauses, the direct approach is intractable in the worst case, and, as our experimentations will illustrate, in many expectedly-easier situations.

A Transformational Method

On the contrary, we propose an approach that avoids the extraction of these intermediate Co-MSSes and make a direct global extraction of one AC-$\mathrm{MSS}(\Delta, \Gamma)$ in all cases. To this end, both Δ and Γ are transformed into two other sets of clauses and a new (but equivalent) problem is generated.

We shall make use, among other things, of an algorithm that extracts one $\mathrm{MSS}_{\subseteq}(\Delta)$ that is satisfiable with the conjunction of the elements of another set of clauses, when this latter set is satisfiable. Such an algorithm is easily derived from the CMP method from (Grégoire, Lagniez, and Mazure 2014) or any procedure described in (Marques-Silva et al. 2013). For convenience purpose, the notation *Extract-MSS*(Σ_1, Σ_2) where Σ_1 and Σ_2 are two sets of clauses will be used to represent either this algorithm, or, when cardinality-maximal subsets are under consideration, Partial-Max-SAT(Σ_1,Σ_2).

Our method makes one call to *Extract-MSS*, only. It is insensitive to whatever ordering of formulas from Γ or Δ, and builds the set of hard constraints Σ_2 to be a satisfiable set. In addition, the following seemingly contradictory features also need to be accommodated together, among other things. On the one hand, each formula γ of Γ need be treated separately from the other formulas from Γ to ensure that the resulting MSS is satisfiable with γ. Hence, this treatment is local (i.e., performed with respect to γ). On the other hand, this local decision has a global impact on the problem since it impacts the satisfiability of the remaining part of Δ with other γ's. The method satisfies these local and global requirements: it is based on following ideas.

First, note that whenever γ is satisfiable with Δ, γ is satisfiable with any subset of Δ and thus with any MSS of Δ: hence, no clause needs to be dropped from Δ in the local process that ensures that the extracted MSS is satisfiable with γ. In the following, we consider only γ's such that $\Delta \cup \{\gamma\}$ is unsatisfiable.

For each γ, the extraction of one maximal subset of clauses that is satisfiable with γ is rewritten as an independent sub-problem. To this end, for each γ, a corresponding set of clauses $\Delta \cup \{\gamma\}$ is created; all its variables are renamed through the use of fresh atoms. All sub-problems are then linked together as follows, adopting one arbitrary ordering between clauses in Δ. An atom ϵ_i is created and associated to each clause δ_i from Δ. More precisely, each clause δ from Δ in every sub-problem (i.e., $\Delta \cup \{\gamma\}$ that has been renamed using fresh variables), gets the additional disjunct $\neg\epsilon_i$ when δ corresponds to δ_i in Δ. Now, all these $\Delta \cup \{\gamma\}$ endowed with the additional disjuncts will form the hard consistency constraints, i.e., the second parameter in the unique call to *Extract-MSS*. The set consisting of all these clauses is satisfiable (just set every ϵ_i to *false*). Secondly, the set of unit clauses $\{\epsilon_i \text{ s.t. } i \in [1..n]\}$ forms the soft constraints, i.e., the first parameter in the unique call to *Extract-MSS*. Whenever a clause ϵ_i from this set does not belong to the MSS that is extracted, δ_i from Δ does not belong to the (global) computed AC-MSS. Indeed, this means that some clause within the hard constraints that contains the disjunct $\neg\epsilon_i$ actually requires $\neg\epsilon_i$ to be *true*, and the corresponding clause is not to be a member of the AC-MSS under construction.

The *Transformational Approach* algorithm that is depicted below implements this idea. In line 1, the set Σ_1 of soft clauses is created: it is the set of positive unit clauses ϵ_i, called selectors, that are created for every clause $\delta_i \in \Delta$. In line 2, the set of hard clauses Σ_2 is initialized to the empty set. Σ_2 will gather the sub-problems. In line 3, a copy of Δ is created where each clause is weakened by the negation of its corresponding selector ϵ_i. This copy is called Δ'. Whenever γ_j contradicts Δ (l. 4), $\Delta' \cup \{\gamma_j\}$ has all its variables (except the ϵ_i) renamed (l. 6), thereby representing the sub-problem Φ_j devoted to finding an MSS satisfiable with γ_j (l. 4-7). All sub-problems are collected in a single set that forms Σ_2 (l. 7).

Notice that when every γ is satisfiable with Δ, Σ_2 is empty and Δ is delivered as result; this situation can occur independently of Γ being satisfiable or not. Otherwise (l. 9), an MSS of Σ_1 that satisfies all the clauses of the global problem Σ_2 is delivered in Ψ by the call to *Extract-MSS*. The final result is the set of all clauses δ_i of Δ for which the corresponding clause ϵ_i belongs to the set Ψ (which is the output of the single call to *Extract-MSS*). Let m be the number of different formulas in Γ, n' the largest number of clauses encoding one formula of Γ, n the number of clauses in Δ.

```
Transformational Approach
   input : $\Delta = \{\delta_1, \ldots, \delta_n\}$: a set of $n$ Boolean clauses;
           $\Gamma = \{\gamma_1, \ldots, \gamma_m\}$: a set of $m$ satisfiable Boolean formulas;
   output: one $\text{AC-MSS}(\Delta, \Gamma)$
   /* $\Sigma_1$ will be the set of soft clauses */
1  $\Sigma_1 \leftarrow \{\epsilon_i \text{ s.t. } i \in [1..n]\}$ /* every $\epsilon_i$ is a fresh new atom */
   /* $\Sigma_2$ will be the set of hard clauses */
2  $\Sigma_2 \leftarrow \emptyset$;
3  $\Delta' \leftarrow \{\delta_i \vee \neg\epsilon_i \text{ s.t. } \delta_i \in \Delta\}$;
4  foreach $\gamma_j \in \Gamma$ s.t. $\text{UNSAT}(\Delta \cup \{\gamma_j\})$ do
       /* $\Phi_j$ is the sub-problem related to $\gamma_j$ */
5      $\Phi_j \leftarrow \Delta' \cup \{\gamma_j\}$ ;
6      Rename all atoms in $\Phi_j$ (except the $\epsilon_i$) with fresh new atoms;
7      $\Sigma_2 \leftarrow \Sigma_2 \cup \Phi_j$;
8  if $\Sigma_2 \neq \emptyset$ then
9      $\Psi \leftarrow \text{Extract-MSS}(\Sigma_1, \Sigma_2)$;
10     $\Delta \leftarrow \{\delta_i \text{ s.t. } \delta_i \in \Delta \text{ and } \epsilon_i \in \Psi\}$;
11 return $(\Delta)$;
```

Property 1. *The Transformational Approach computes one* $\text{AC-MSS}(\Delta, \Gamma)$.

Property 2. *The Transformational Approach requires m calls to a SAT solver on an instance of size $O(n + n')$, plus one call to Extract-MSS with a set of hard constraints of consistency of size $O(m(n + n'))$ and a set of soft constraints of size $O(n)$.*

From a worst-case complexity analysis, an approach that computes all MSS for each γ as one step of the construction of one $\text{AC-MSS}(\Delta, \Gamma)$ is clearly intractable since the number of MSS of a set of clauses can be exponential in the number of clauses in the set. Property 2 allows the worst-case complexity of the transformational method to be derived. For example, when *Extract-MSS* is related to $\text{MSS}_{\subseteq}$ computation, the method requires $O(m)$ calls to a SAT-solver on an instance of size $O(n + n')$, plus a logarithmic number of calls to a SAT-solver on an instance of initial size $O(m(n + n'))$ that is divided by two at each call.

Formal Specification

Formally, the transformational method can be defined as follows.

Selectors: *Let $(\epsilon_i)_{i=1,\ldots,\text{card}(\Delta)}$ be a family of Boolean variables extracted from $\mathcal{P} \setminus Atom(\Delta \cup \Gamma)$*

Copy substitutions: *A family of substitutions $(\mu_j)_{j=1,\ldots,\text{card}(\Gamma)}$, each with domain $Atom(\Delta \cup \Gamma)$ and codomain $\mathcal{P} \setminus Atom(\Delta \cup \Gamma)$, are defined such that*

- $Image(\mu_j) \cap \{\epsilon_i \mid i = 1, \ldots, \text{card}(\Delta)\} = \emptyset$ *for* $j = 1, \ldots, \text{card}(\Gamma)$
- $Image(\mu_h) \cap Image(\mu_k) = \emptyset$ *whenever* $h \neq k$

Copies: *For* $j = 1, \ldots, \text{card}(\Gamma)$*, let*

$$\Phi_j \overset{\text{def}}{=} \{\mu_j\delta_i \vee \neg\epsilon_i \mid i = 1, \ldots, \text{card}(\Delta)\} \cup \{\mu_j\gamma_j\}$$

Problem: $\text{ExtractMSS}(\Sigma_1, \Sigma_2)$ *where*

- $\Sigma_1 = \{\epsilon_i \mid i = 1, \ldots, \text{card}(\Delta)\}$

- $\Sigma_2 = \bigcup_{j=1,\ldots,\mathrm{card}(\Gamma)} \Phi_j$

For Ω satisfiable, $\mathrm{ExtractMSS}(\Theta, \Omega)$ is supposed to be such that its output is both a maximal satisfiable subset of $\Theta \cup \Omega$ and a superset of Ω.

Experimental Study

The goal of the experimental study was to investigate the actual viability of the transformational approach.

Benchmarks from the planning area as a case study. We have considered 225 instances of usual benchmarks from the planning area as a case study. Let us stress that we do not solve the planning problem in the experimentations. All we do is "filter" the planning instances so that they become consistent with each assumptive context. We handled the instances as mere sets of clauses and we did not assign initial and final states their whole specific roles, since these concepts are not relevant to the other potential application domains. If we want to apply the approach in the planning domain up to computing plans then it would make sense to slightly adapt the framework so that initial and final states, fluents, etc. do match their full epistemological roles. For example, when a goal (final state) contradicts an assumptive context, the user might be asked whether or not she (he) would accept the goal to be transformed so that it becomes consistent with each of these contexts. The benchmarks Δ represent the domain knowledge, as well as the initial and goal states. For example, Δ can represent the environment in which a robot is moving, as well as its initial and goal states, whereas Γ translates some additional assumptive information: for example γ_1 is indicating that the energy level of the robot is too low to reach a target whereas γ_2 asserts the contrary.

The instances cover a wide range of planning problems with varying horizon lengths. For example, "*Blocks_right_x*" refers to the usual blocks-world problem involving x blocks. "*Bomb_bx_by*" is the similar information for the problem of neutralizing bx bombs in by locations. "*Coins_px*" is about px coins that need to be tossed for heads and tails so that they all reach the same state. "*Comm_px*" corresponds to an IPC5 problem about communication signals with several stages, packets, and actions. "*Empty-room_dx_dy*" is about a robot navigating in a room of size dx and containing dy objects. "*Safe_n*" is about opening a safe with n possible combinations. "*sort_num_s_x*" is about building circuits of compare-and-swap gates to sort x Boolean variables. "*uts_kx*" is about a network routing problem for mobile ad-hoc networks where a broadcast from an unknown node must reach all nodes; the topology of the network is partially known and each node in the graph has a fixed number kx of connected neighbors.

Each Δ was translated into clausal normal form from its initial PDDL 1.2 (Planing Domain Definition Language) and STRIPS format, using H. Palacios' translator, available from http://www.plg.inf.uc3m.es/~hpalacio/.

Generation of the assumptive contexts. We have experimented the method with a reasonable maximal value m of assumptive contexts, namely 10. We considered the worst-case situation with respect to the grow of the instance size that is encountered through the transformation: this occurs when every formula γ conflicts with Δ. We have also selected each γ as a set of non-unit clauses that can be proved satisfiable in a very short time, following the hypothesis that assumptive contexts are easily shown satisfiable but are not necessarily mere unit clauses. More precisely, each γ has been generated as follows. An unsatisfiable 3-SAT instance, i.e., an unsatisfiable set of ternary clauses Φ was randomly generated after the satisfiability/unsatisfiability threshold, using the variables occurring in clauses in Δ representing the initial state. To this end, we used the standard clauses generator from WALKSAT Version 51 (http://www.cs.rochester.edu/~kautz/walksat/). Accordingly, the γ's are intended to represent a series of possibly conflicting information about the initial state. Φ was then partitioned into m subsets of clauses, each of them forming the seed to build one assumptive context γ. Each seed was then augmented with additional random binary clauses until it became unsatisfiable with Δ and thus formed the intended γ. This was done in such a way that γ remains satisfiable. Accordingly, each γ is one satisfiable set of clauses that is unsatisfiable with Δ and the set $\Gamma = \{\gamma_1, \ldots, \gamma_m\}$ is itself unsatisfiable.

We believe that this generation model provides γ's that can capture reasonably complex items in the target domain.

Hardware, software and time-out. All experimentations have been conducted on Intel Xeon E5-2643 (3.30GHz) processors with 8Gb RAM on Linux CentOS. Time limit was set to 60 minutes per test. All experimentation data, as well as our developed methods, are available from http://www.cril.fr/AAAI15-BGL. For the inclusion-maximal case, we have adapted the CMP method from (Grégoire, Lagniez, and Mazure 2014) in such a way that it implements *Extract-MSS*. We have used CAMUS (Liffiton and Sakallah 2008b) http://sun.iwu.edu/~mliffito/camus/ to enumerate all $\mathrm{MSS}_{\subseteq}$ and Co-$\mathrm{MSS}_{\subseteq}$. MSUNCORE (Morgado, Heras, and Marques-Silva 2012) http://logos.ucd.ie/wiki/doku.php?id=msuncore and MINISAT (Eén and Sörensson 2004) http://minisat.se/ were selected as the Partial-Max-SAT and SAT solvers, respectively.

Table 1 provides a sample of results (the results for the 225 instances tested for each value of $|\Gamma| \in \{2, 3, 5, 10\}$ are available from http://www.cril.fr/AAAI15-BGL). All times are in seconds, rounded down to the immediate strictly lower integer. In the first column, the instance name is given, followed by the size of Δ in terms of the number of its variables ($\#Vars$) and clauses ($\#Clauses$), successively. The names of the instance are postfixed with the planning horizon, written p_tx, as "planning problem with tx steps as horizon". The following parameters about Γ are then listed successively: first, the number of variables in Γ ($\#Var$), which are all the variables from Δ used to encode the initial state of the planning problem. Then, the number of formulas ($|\Gamma|$) and the average size of the γ's ($avg\ |\gamma|$) in terms of their number of clauses are given. Then the table provides the

Instance	Γ			Brute force			Transformed Inst.			AC-MSS$_{\subseteq}$ Tr. Meth.			AC-MSS$_{\#}$ Tr. Meth.		
Name (#Vars #Clauses)	#Var	$\|\Gamma\|$	$avg\|\gamma\|$	status	T	$avg\|\#coMss\|$	#var	#hard	#soft	status	T	#rm	status	T	#rm
blocks_right_2_p_t5 (406 1903)	67	2	236	MO	?	?	2715	3806	1903	OK	0	78	OK	453	43
bomb_b5_t1_p_t2 (240 443)	66	10	78	OK	61	9433	2843	4430	443	OK	0	6	OK	0	5
bomb_b10_t10_p_t1 (1000 1870)	500	2	1757	MO	?	?	3870	3740	1870	OK	2	6	OK	2007	4
coins_p01_p_t3 (536 1419)	112	10	83	MO	?	?	6779	14190	1419	OK	0	8	OK	0	4
coins_p03_p_t2 (368 951)	112	5	157	MO	?	?	2791	4755	951	OK	0	28	OK	840	11
coins_p03_p_t5 (872 2355)	112	5	157	MO	?	?	6715	11775	2355	OK	0	16	OK	63	10
	112	10	85	MO	?	?	11075	23550	2355	OK	0	10	OK	2	6
coins_p05_p_t2 (368 951)	112	5	157	MO	?	?	2791	4755	951	OK	0	12	OK	196	9
comm_p02_p_t2 (555 1623)	189	10	140	MO	?	?	7173	16230	1623	OK	0	9	OK	5	6
comm_p05_p_t5 (3384 12267)	510	2	1786	MO	?	?	19035	24534	12267	OK	1	361	MO	?	?
	510	10	366	MO	?	?	46107	122670	12267	OK	0	15	OK	72	6
emptyroom_d4_g2_p_t1 (44 130)	32	10	22	OK	922	37875	570	1300	130	OK	0	9	OK	0	5
emptyroom_d8_g4_p_t3 (244 778)	72	10	51	MO	?	?	3218	7780	778	OK	0	21	OK	97	7
ring2_r6_p_t1 (76 215)	54	10	38	MO	?	?	975	2150	215	OK	0	17	OK	11	8
ring2_r6_p_t2 (134 402)	54	2	190	MO	?	?	670	804	402	OK	0	54	OK	0	26
ring_5_p_t1 (114 242)	70	3	164	MO	?	?	584	726	242	OK	0	32	OK	148	12
safe_safe_10_p_t5 (166 357)	21	2	75	OK	635	69924	689	714	357	OK	0	24	OK	0	5
safe_safe_30_p_t5 (486 1347)	61	10	43	MO	?	?	6207	13470	1347	OK	0	82	OK	44	17
sort_num_s_3_p_t1 (39 106)	27	2	96	MO	?	?	184	212	106	OK	0	19	OK	0	10
sort_num_s_3_p_t4 (129 400)	27	10	30	MO	?	?	1690	4000	400	OK	0	12	OK	0	8
sort_num_s_4_p_t5 (486 1810)	88	10	62	MO	?	?	6670	18100	1810	OK	0	21	OK	3353	10
sort_num_s_6_p_t2 (858 3509)	396	3	925	MO	?	?	6083	10527	3509	OK	0	350	MO	?	?
uts_k1_p_t2 (71 204)	25	5	40	OK	337	29328	559	1020	204	OK	0	8	OK	0	7
uts_k2_p_t5 (530 1903)	81	10	57	MO	?	?	7203	19030	1903	OK	0	20	OK	1114	13
uts_k3_p_t3 (682 2695)	169	10	118	MO	?	?	9515	26950	2695	OK	0	46	MO	?	?

Table 1: Sample of Results. The full table can be found at http://www.cril.fr/AAAI15-BGL

experimental results for the first step of the "brute force" method, which extracts for every γ all Co-MSS$_{\subseteq}(\Delta \cup \{\gamma\})$ that do not contain γ. The columns list the status of this step ("OK", or "MO" for memory-out) and the time ("T") to run the method and the average number of computed Co-MSSes (*avg* $|\#CoMSS|$) for each γ. When memory out happened due to combinatorial blow-up, no information about these two parameters could be delivered (hence, the "?" in the columns). Results about the transformational methods are provided next. Firstly, the main parameters of the transformed problem are given, namely the number of variables in the transformed problem, the number of clauses of this latter one split between the clauses that are intended to play the role of hard clauses (#*hard*) and soft clauses (#*soft*), respectively. Secondly, results about computing one $\text{AC-MSS}_{\subseteq}$ are given: for every successfully solved instance (status = "OK"), we list the global time to complete the transformational method (including the call to *Extract-MSS*) and the number of ($\#rm$) clauses that are dropped from Δ to yield the extracted $\text{AC-MSS}_{\subseteq}$. Thirdly, the last columns give the same main parameters when one $\text{AC-MSS}_{\#}(\Delta, \Gamma)$ was to be extracted.

Discussion. As expected, the brute force method to extract one $\text{AC-MSS}_{\subseteq}$ (or one $\text{AC-MSS}_{\#}$) faced computational blow-up most of the times: the full list of intermediate Co-MSSes was delivered for only $81/900$ problem instances within the 1 hour computation allocated per instance. Not surprisingly, the transformational method that extracts one $\text{AC-MSS}_{\subseteq}$ outperformed the brute force one: *all* the instances (i.e., $900/900$) were solved successfully, and the average time to compute one AC-MSS was 0.05 second, reaching real-time performance (only $2/900$ instances needed more than 2 seconds, namely 4.54 and 2.41 seconds). Not surprisingly, applying an additional minimal-cardinality constraint in order to yield one $\text{AC-MSS}_{\#}$ proved less often feasible. However, $415/900$ instances were successfully solved in this way. For example, the *bomb*_$b10_t1_p_t2$ instance involves 1870 clauses and 1000 Boolean variables: it yielded the second slowest run-time (2.41 seconds) for the computation of one $\text{AC-MSS}_{\subseteq}$. Γ was built by using the 500 variables occurring in the description of the initial state of the planning problem, giving rise to two formulas, each of them encoded through 1757 clauses on average. Note that in this respect each γ is intended to represent a very elaborate assumptive context for the concerned planning problem. The ratio #*hard*/#*Clauses*$=$ 2 since the two γ's were built so that they were both contradictory with Δ. The tentative enumeration of all Co-MSSes led to memory overflow. The transformed instance is made of 3740 hard and 1870 soft clauses, making use of 3870 variables. The transformational method that extracted one $\text{AC-MSS}_{\#}$ required 2007 seconds to be completed (including the transformation step), ending with a guaranteed minimal number of 4 clauses to be rejected from the initial instance. The method that extracted one $\text{AC-MSS}_{\subseteq}$ rejected two additional clauses but was very fast (2.41 seconds rounded down to 2 in Table 1).

The real-time performance reached in the experimentations should be interpreted as a positive point: the experimental conditions that allow for these results are reasonable and possibly widespread even outside the planning field. Especially, the number and size of assumptive contexts and the limited size of the computed Co-MSSes are all important criteria: relaxing any of them can lead to possible performance degradation. Finally, we believe that although all the tested instances are related to the same generic domain (i.e., planning), these criteria about the "logical structure" of the instances are also relevant to other potential application domains.

Conclusion and Perspectives

We believe that the results in this paper could open various promising paths for further research. Firstly, as motivated in the introduction, the technique from this paper could be exported to the specific settings of various relevant A.I. subfields and problems, which might however require additional technical developments. Secondly, as the method circumvents the possibly exponential number of MSSes and of CoMSSes, it might be fruitful to explore to which extent this can prove useful in the practical handling of other problems that usually rely on the enumeration of these latter sets, like the enumeration of MUSes (Minimal Unsatisfiable Subsets). Finally, the additional minimization required by the extraction of cardinality-maximal AC-MSS entails some significant computational overhead: in view of the better performance of the computation of one AC-MSS$_{\subseteq}$ in practical situations, it might be worth defining and investigating strategies for the approximation of one AC-MSS$_{\#}$ through the guided computation and handling of several AC-MSS$_{\subseteq}$.

Acknowledgments

The authors are grateful to the reviewers for their useful comments and they are indebted to B. Mazure for discussions on the topics of this paper.

References

Alchourrón, C. E.; Gärdenfors, P.; and Makinson, D. 1985. On the logic of theory change: Partial meet contraction and revision functions. *Journal of Symbolic Logic* 50(2):510–530.

Ben-Tal, A.; Ghaoui, L. E.; and Nemirovski, A. 2009. *Robust Optimization*. Princeton University Press.

Besnard, P., and Hunter, A. 2008. *Elements of Argumentation*. MIT Press.

Besnard, P.; Grégoire, É.; Piette, C.; and Raddaoui, B. 2010. MUS-based generation of arguments and counter-arguments. In *Proceedings of the IEEE International Conference on Information Reuse and Integration (IEEE IRI'10)*, 239–244.

Delgrande, J. P.; Schaub, T.; and Jackson, W. K. 1994. Alternative approaches to default logic. *Artificial Intelligence* 70(1-2):167–237.

Dung, P. M. 1995. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence* 77(2):321–358.

DX. 2014. *25th International Workshop on Principles of Diagnosis (DX'14)*. http://dx-2014.ist.tugraz.at/dx14_program_papers.html (on-line proceedings).

Eén, N., and Sörensson, N. 2004. An extensible SAT-solver. In *Proceedings of the 6th International Conference on Theory and Applications of Satisfiability Testing (SAT'03). Selected Revised Papers*, volume 2919 of *Lecture Notes in Computer Science*, 502–518. Springer.

Fermé, E. L., and Hansson, S. O. 2011. AGM 25 years - twenty-five years of research in belief change. *Journal of Philosophical Logic* 40(2):295–331.

Fuhrmann, A., and Hansson, S. O. 1994. A survey of multiple contractions. *Journal of Logic, Language, and Information* 3(1):39–75.

Grégoire, É.; Lagniez, J.-M.; and Mazure, B. 2014. An experimentally efficient method for (MSS,CoMSS) partitioning. In *Proceedings of the 28th Conference on Artificial Intelligence (AAAI'14)*.

Grégoire, É.; Mazure, B.; and Piette, C. 2007. Boosting a complete technique to find MSS and MUS thanks to a local search oracle. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI'07)*, 2300–2305.

Hamscher, W.; Console, L.; and de Kleer, J., eds. 1992. *Readings in model-based diagnosis*. Morgan Kaufmann.

Levi, I. 1977. Subjunctives, dispositions and chances. *Synthese* 34:423–455.

Liffiton, M. H., and Sakallah, K. A. 2008a. Algorithms for computing minimal unsatisfiable subsets of constraints. *Journal of Automated Reasoning* 40(1):1–33.

Liffiton, M., and Sakallah, K. 2008b. Algorithms for computing minimal unsatisfiable subsets of constraints. *Journal of Automated Reasoning* 40(1):1–33.

Marques-Silva, J., and Janota, M. 2014. On the query complexity of selecting few minimal sets. *Electronic Colloquium on Computational Complexity (ECCC)* 21:31.

Marques-Silva, J.; Heras, F.; Janota, M.; Previti, A.; and Belov, A. 2013. On computing minimal correction subsets. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI'13)*.

Morgado, A.; Heras, F.; and Marques-Silva, J. 2012. Improvements to core-guided binary search for MaxSAT. In *Proceedings of the 15th International Conference on Theory and Applications of Satisfiability Testing (SAT'12)*, volume 7317 of *Lecture Notes in Computer Science*, 284–297. Springer.

Papadimitriou, C. H., and Yannakakis, M. 1991. Optimization, approximation, and complexity classes. *Journal of Computer and System Sciences* 43(3):425 – 440.

Reiter, R. 1980. A logic for default reasoning. *Artificial Intelligence* 13(1-2):81–132.

Vesic, S. 2013. Identifying the class of maxi-consistent operators in argumentation. *Journal of Artificial Intelligence Research (JAIR)* 47:71–93.