Many Hands Make Light Work: Localized Satisfiability For Multi-Context Systems

Floris Roelofsen[1,2] and **Luciano Serafini**[1] and **Alessandro Cimatti**[1]

Abstract. In this paper, we tackle the satisfiability problem for multi-context systems. First, we establish a satisfiability algorithm based on an encoding into propositional logic. Then, we propose a distributed decision procedure that maximally exploits the potential amenity of localizing reasoning and restricting it to relevant contexts. We show that the latter approach is computationally superior to our translation-based procedure, and outline how off-the-shelf efficient reasoning procedures can be used to implement our algorithm.

1 Introduction

The establishment of a solid paradigm for contextual knowledge representation and contextual reasoning is of paramount importance for the development of sophisticated theory and applications in AI.

McCarthy [16] pleaded for a formalization of context as a possible solution to the problem of generality; Giunchiglia [9] emphasized that reasoning based on a large (common sense) knowledge base can only be effectively pursued if suitably confined to a manageable subset (context) of that knowledge base.

Contexts were first implemented as microtheories in the famed CYC common sense knowledge base [12]. However, while in CYC local microtheories were a choice, in modern settings like that of the semantic web the notion of local, distributed knowledge is a must. Contemporary architectures impose highly scattered, heterogeneous knowledge fragments, which a central reasoner cannot deal with. This engenders a high demand for contextual reasoning procedures.

Several formalizations of contextual knowledge representation have been proposed. Most notable are the propositional logic of context developed by McCarthy, Buvač and Mason [17, 18], and the multi-context systems devised by Giunchiglia and Serafini [11], which later became associated with the local model semantics [8]. NP-completeness has been established for PLC [15] and MCS [22]. Recently, MCS has been proven strictly more general than PLC [23].

In this paper, we propose an algorithm that settles satisfiability in MCS in a distributed fashion by a single fixpoint computation. Each iteration of this computation constitutes several local reasoning procedures, which can be implemented by (a diversity of) customized state-of-the-art SAT solvers. We discuss the use of both BDD- and SAT-based techniques for this purpose, and show that our algorithm is computationally superior to translation-based procedures.

We proceed as follows. After defining MCS and explicating the contextual satisfiability problem in section 2, we give its encoding into propositional logic in section 3. A general specification of our distributed approach is given in section 4, while section 5 discusses some implementational issues. Finally, in section 6, we relate our work to other state-of-the-art techniques for distributed reasoning.

[1] ITC-IRST, Italy. [2]University of Twente, Netherlands.

2 Multi-Context Systems

A simple illustration of the intuitions underlying MCS is provided by the so-called "magic box" example [8], depicted below.

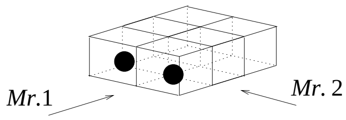

Figure 1. The magic box

Example 1 *Mr.1 and Mr.2 look at a box, which is called "magic" because neither of the observers can make out its depth. Both Mr.1 and Mr.2 maintain a local representation of what they see. These representations must be coherent – for instance, if Mr.2 thinks there's a ball in the box, Mr.1 should believe so too.*

We demonstrate how such interrelated local representations can be captured formally. Our point of departure is a set of indices I. Each index $i \in I$ denotes a *context*, which is described by a corresponding formal (in this case standard propositional) language L_i. To state that a formula φ in L_i holds in context i we utilize so-called *labeled formulas* of the form $i : \varphi$ (when no ambiguity arises we simply refer to *labeled formulas* as *formulas*). Two or more formulas that apply to different contexts may be related by so-called *bridge rules*. These are expressions of the form:

$$i_1 : \phi_1, \ldots, i_n : \phi_n \rightarrow i : \varphi \tag{1}$$

where $i_1, \ldots, i_n, i \in I$ and $\phi_1, \ldots, \phi_n, \varphi$ are formulas. Note that "$\rightarrow$" does not denote implication (we'll use "$\supset$" for this purpose). Also note that our language doesn't include expressions like $\neg(i : \varphi)$ and $(i : \varphi \wedge j : \psi)$. We call $i : \varphi$ the *consequence* and $i_1 : \phi_1, i_n : \phi_n$ the *premises* of bridge rule (1). We write $cons(br)$ and $prem(br)$ for the consequence and the set of all premises of a bridge rule br.

Definition 1 (Multi-Context System MCS) *A propositional multi-context system $\langle \{L_i\}_{i \in I}, \mathbb{BR} \rangle$ over a set of indices I consists of a set of propositional languages $\{L_i\}_{i \in I}$ and a set of bridge rules $\mathbb{BR}$. In this paper, we assume I to be countable and $\mathbb{BR}$ to be finite.*

Example 2 *The scenario in example 1 may be formalized by an MCS consisting of two contexts* 1 *and* 2*, which are described by $L_1 = L(\{l, r\})$ and $L_2 = L(\{l, c, r\})$, respectively. The constraint that Mr.1 should believe the box to be nonempty if Mr.2 believes this to be the case, is formalized by the following bridge rule:*

$$2 : l \vee c \vee r \quad \rightarrow \quad 1 : l \vee r \tag{2}$$

Let M_i denote the class of classical valuations of L_i. Each $m \in M_i$ is called a *local model* of L_i. Interpretations of an entire MCS are called *chains*. They are constructed from sets of local models.

Definition 2 (Chain) *A chain c over a set of indices I is a sequence $\{c_i\}_{i\in I}$, where each $c_i \subseteq M_i$ is a set of local models of L_i. A chain c is i-consistent if c_i is nonempty; it is J-consistent, for some $J \subseteq I$, if it is j-consistent for all $j \in J$.*

A chain can be thought of as a set of "epistemic states", each corresponding to a certain context (or agent). The fact that c_i contains more than one model signifies that L_i is interpretable in more than one way – or alternatively, that agent i has only *partial* knowledge. The epistemic states that a chain consists of all concern *one and the same* situation. Therefore, arbitrary sets of local models may not always constitute a "sensible" chain. This conception is captured by the notion of "bridge rule compliance" specified below.

Definition 3 (Bridge Rule Compliance and Satisfiability) *Let c be a chain, φ a formula over L_i, and $\mathbb{BR}$ the set of bridge rules of a multi-context system MS.*

1. *c satisfies $i : \varphi$ if $m \models \varphi$ for all local models $m \in c_i$. We write $c \models i : \varphi$.*
2. *c complies with $\mathbb{BR}$ if for all $br \in \mathbb{BR}$ either $c \models cons(br)$ or $c \not\models i : \xi$ for some $i : \xi \in prem(br)$.*
3. *$i : \varphi$ is satisfiable in MS if there is an i-consistent chain c that satisfies $i : \varphi$ and complies with $\mathbb{BR}$.*

The contextual satisfiability problem, then, is to determine whether a set of formulas Φ is satisfiable in a multi-context system MS.

In the following we refer to the set of bridge rules of MS as $\mathbb{BR}$, and to the set of contexts involved by formulas in Φ as J.

3 Encoding Into Propositional SAT

In this section we provide an encoding of contextual satisfiability into propositional SAT, and discuss the complexity of the resulting problem in terms of the dimension of the underlying MCS.

We first remark that our encoding cannot simply consist in labeling local propositions with the index of the context that they describe. This is illustrated by the following example:

Example 3 *Consider an MCS with two contexts 1 and 2, described by $L_1 = L(\{p\})$, $L_2 = L(\{q\})$, and the following bridge rules:*

$$\begin{array}{rcl} 1 : p & \rightarrow & 2 : q \\ 1 : \neg p & \rightarrow & 2 : q \end{array}$$

The formula $2 : \neg q$ is satisfied in this system by the chain:

$$\left\{ \begin{array}{c} \{\{p\}, \{\neg p\}\}, \\ \{\{\neg q\}\} \end{array} \right\}$$

Notice that a simple "indexing" encoding of this system into propositional logic would be inconsistent.

To overcome this problem, we express our encoding in terms of atomic propositions p_i^k, whose evaluation corresponds to the truth value assigned to a proposition p in L_i *by the k^{th} local model in c_i*. But how many local models can c_i contain? The following theorem implies that this number may in fact be assumed to be bounded by the number of bridge rules of the MCS under consideration.

Theorem 1 (Bounded Model Property) *A set of formulas Φ is satisfiable in a multi-context system MS iff it is satisfied in MS by a chain that contains at most $|\Phi| + |\mathbb{BR}|$ local models.*

Proof. Take any chain c that satisfies Φ in compliance with $\mathbb{BR}$. Let $\mathbb{BR}^* \subseteq \mathbb{BR}$ be the set of bridge rules whose consequences are not satisfied by c. Every $br \in \mathbb{BR}^*$ must have a premise which is not satisfied in some local model $m(br)$ contained by c. On the other hand, every formula $i : \varphi \in \Phi$ must be satisfied in at least one local model $m(i : \varphi)$ in c_i. The chain c^* obtained from c by eliminating all local models except:

$$\bigcup_{br\in\mathbb{BR}^*} m(br) \cup \bigcup_{i:\varphi\in\Phi} m(i : \varphi)$$

is J-consistent, satisfies Φ in compliance with $\mathbb{BR}$ and contains at most $|\Phi| + |\mathbb{BR}^*| \leq |\Phi| + |\mathbb{BR}|$ local models. □

Assuming that $|\mathbb{BR}| \geq 1$, Theorem 1 can be slightly weakened to the statement that Φ is satisfiable in MS iff it is satisfied by a chain c^* all of whose components are either empty or contain exactly $|\mathbb{BR}|$ local models. We now construct a propositional formula ψ that is satisfiable, in a classical sense, exactly if such a chain c^* exists.

Apart from the atomic propositions p_i^k mentioned above, we avail ourselves of propositions e_i, for each index $i \in I$, corresponding to c_i^* being empty. For any formula φ, $i \in I$ and $k \in K$, let φ_i^k denote the formula that results from substituting every atomic proposition p in φ with p_i^k. Let us write $K = \{1, \ldots, |\mathbb{BR}|\}$, and $\varphi_i^K = \bigwedge_{k\in K} \varphi_i^k$. Then, the translation of a labeled formula reads:

$$(i : \varphi)^* = e_i \vee \varphi_i^K$$

For bridge rules we have:

$$\begin{array}{c} (i_1 : \varphi_1, \ldots, i_n : \varphi_n \rightarrow i : \phi)^* = \\ (i_1 : \phi_1)^* \wedge \ldots \wedge (i_n : \phi_n)^* \supset (i : \phi)^* \end{array}$$

And a j-consistency constraint is captured by:

$$(j\text{-cons})^* = \neg e_j$$

Theorem 2 *There is an assignment V to the set of propositions $\{p_i^k | p \in L_i \text{ and } k = 1, \ldots, |\mathbb{BR}|\} \cup \{e_i | i \in I\}$ that satisfies:*

$$\psi = \bigwedge_{i:\phi\in\Phi} (i : \phi)^* \wedge \bigwedge_{j\in J} (j\text{-cons})^* \wedge \bigwedge_{br\in\mathbb{BR}} (br)^*$$

iff there's a J-consistent chain that satisfies Φ and complies with $\mathbb{BR}$.

Proof ($\Rightarrow$) From V construct a chain c^V, such that each component c_i^V is empty if $V(e_i) = True$ and contains exactly $|\mathbb{BR}|$ local models otherwise. In the latter case, let the k^{th} local model of c_i^V evaluate each atomic proposition $p \in L_i$ to $True$ iff $V(p_i^k) = True$. Clearly, c^V is J-consistent and satisfies Φ in compliance with $\mathbb{BR}$.

($\Leftarrow$) If there is a J-consistent chain c that satisfies Φ in compliance with $\mathbb{BR}$, there must also be a J-consistent chain c^* each of whose components is either empty or contains exactly $|\mathbb{BR}|$ local models, and which still satisfies Φ in compliance with $\mathbb{BR}$.

From c^* we obtain V as follows. Let V assign $True$ to an atomic proposition e_i iff c_i^* is empty. Let V assign $True$ to an atomic proposition p_i^k iff the k^{th} local model of c_i^* satisfies p, and any truth value iff c_i^* is empty. It is easy to see that V satisfies ψ. □

The deterministic time complexity of the propositional satisfiability problem resulting from our encoding is $O(2^{|P|})$, where P is the set of propositional variables involved. Let P_i denote the set of propositions that is used to describe context i. Then $|P|$ amounts to:

$$|I| + |\mathbb{BR}| \times \sum_{i\in I} |P_i|$$

4 A Distributed Algorithm

Contextual satisfiability can be settled rather naturally by a single distributed fixpoint computation. We show that this computation is more efficient than the translation-based method presented above.

Our approach is the following. Starting with some initial chain c^0, we attempt to construct a sequence $c^0, c^1, \ldots, c^k$, such that:

- c^1 satisfies Φ.
- for all $m \in \{1, \ldots, k\}$, c^m *extends* c^{m-1}, that is, for every $i \in I$, $c_i^m \subseteq c_i^{m-1}$.
- for all $m \in \{2, \ldots, k\}$, c^m complies with the bridge rules that c^{m-1} doesn't comply with.

Always *extending* a chain, that is, restricting the sets of local models that constitute its components, has two important implications. First, our initial chain c^0 should be most "general", that is, all its components c_i^0 must contain the entire set of local models M_i. Notice that c^0 doesn't satisfy any formula – in particular, c^0 does not satisfy any bridge rule premise, and therefore complies with $\mathbb{BR}$.

The second implication of always extending a chain, is that once a formula is satisfied by some intermediate chain c^m, then it is also satisfied by c^n, for any $n > m$. This means that (1) if Φ is satisfied by c^1, then it is also satisfied by c^m, for any $m \in \{1, \ldots, k\}$. Moreover, (2) if some intermediate chain c^m does not comply with a bridge rule $br \in \mathbb{BR}$ - that is, c^m satisfies br's premises, but does not satisfy its consequence - then any extension of c^m that were to comply with br should satisfy br's *consequence* (it can by no means be made to not-satisfy one of br's premises). So obtaining c^{m+1} from c^m consists in extending c^m so as to satisfy the consequences of the bridge rules that c^m does not comply with. Finally, (3) once an intermediate chain satisfies the consequence of some bridge rule br (and therefore complies with br), any of its extensions will also satisfy br's consequence and thus comply with br.

Algorithm 1 A distributed algorithm for contextual satisfiability.

$\textsc{ContextSat}(\Phi, \mathbb{BR}, I, J, c)$
begin
$I^* := \{i \in I \mid i : \varphi_i \in \Phi\}$;
for all $i \in I^*$ **do**
  $c_i^* := \textsc{Extend}(c_i, \varphi_i)$;
for all $i \in I/I^*$ **do**
  $c_i^* := c_i$;
for all $j \in J$ **do**
  if $c_j^* = \emptyset$ **then**
    return $False$;
$\mathbb{BR}^* := \{br \in \mathbb{BR} \mid c^* \models i : \eta$ for all $i : \eta \in prem(br)\}$
if $\mathbb{BR}^* = \emptyset$ **then**
  return c^*;
$\Psi^* := \{cons(br) \mid br \in \mathbb{BR}^*\}$;
$\Phi^* := \left\{ i : \varphi \mid \varphi = \bigwedge_{i:\xi \in \Psi^*} \xi \,,\; i \in I \right\}$;
return $\textsc{ContextSat}(\Phi^*, \mathbb{BR}/\mathbb{BR}^*, I, J, c^*)$;
end

This approach is implemented by the $\textsc{ContextSat}$ procedure specified in Algorithm 1. It takes as its input a set of formulas Φ, a set of bridge rules $\mathbb{BR}$, a set of contexts (indices) I, a subset $J \subseteq I$ of contexts whose consistency is required, and finally, a chain c. At first, $\textsc{ContextSat}$ is called with c being a chain over I all of whose components consist of the complete set of local models M_i. It yields a J-consistent extension of c that satisfies Φ in compliance with $\mathbb{BR}$, or $False$ if it fails to construct such an extension.

Extensions are always constructed *locally*. That is, $\textsc{ContextSat}$ first determines the set I^* of contexts involved by formulas in Φ, and then, for every $i \in I^*$, calls a sub-procedure $\textsc{Extend}$ that extends c_i so as to satisfy $i : \varphi$. If the resulting chain is J-inconsistent, any of its further extensions will be J-inconsistent as well. Thus, if such is the case $\textsc{ContextSat}$ recognizes a failure, and returns $False$. If not, it determines the set $\mathbb{BR}^*$ of bridge rules *all* of whose premises are satisfied by c. If $\mathbb{BR}^*$ is empty, c is a solution. Otherwise, making c comply with $\mathbb{BR}^*$ yields a new satisfiability problem, namely that of extending c so as to satisfy the consequence of every $br \in \mathbb{BR}^*$. Bridge rule consequences that concern the same context are taken together in order to obtain a set Φ^* consisting of at most one formula $i : \varphi$ for every context $i \in I$. A new instance of $\textsc{ContextSat}$ is addressed to extend c so as to satisfy Φ^*. Recursively proceeding like this, a chain is constructed that, at any stage, satisfies Φ, and at some point either becomes J-inconsistent, or complies with $\mathbb{BR}$.

4.1 Completeness and Complexity

The set of bridge rule consequences that is satisfied by c is strictly expanded by every recursive call to $\textsc{ContextSat}$. Since the number of bridge rules is finite, $\textsc{ContextSat}$ is bound to terminate.

Soundness is evident; completeness, however, is not yet assured. In order to do so we should enforce $\textsc{Extend}(c_i, \varphi)$ to remove from c_i *exactly* those local models that do not satisfy φ. We say that $\textsc{Extend}(c_i, \varphi)$ should yield a *complete extension of* c_i w.r.t. φ. Notice that this additional constraint implies that extensions of c never *unnecessarily* satisfy a bridge rule premise. In this way the chance of having to re-establish bridge rule compliance is minimized, and therefore further reasoning in other contexts is required only if strictly necessary. This principle of *locality* constitutes an important advantage of our contextual approach w.r.t. centralized procedures. Several techniques can be used to implement $\textsc{Extend}$, such that it indeed yields complete extensions. We'll get back to this in section 5.

In a worst-case scenario $\textsc{Extend}(c_i, \varphi)$ requires time $O(2^{|P_i|})$, where P_i is the set of atomic propositions used to describe context i. Notice that $\textsc{Extend}$ will be called at most $|\mathbb{BR}|$ times.

In general the greater part of computation time will be involved with checking which bridge rule premises are entailed by the current chain. The worst-case scenario consists of two contexts and an even number of bridge rules going back and forth between them. If during each iteration of $\textsc{ContextSat}$ only *one* bridge rule premise is found to be satisfied by the chain constructed so far, the total number of premise-checks is:

$$2 \times (|\mathbb{BR}| + \ldots + 1) = \frac{(|\mathbb{BR}| + 2) \times |\mathbb{BR}|}{4}$$

Each check requires up to time $O(2^{|P_i|})$. Assuming that $Q \equiv |P_1| = |P_2|$, we obtain the following overall upper complexity bound for $\textsc{ContextSat}$:

$$\left(\frac{(|\mathbb{BR}| + 2) \times |\mathbb{BR}|}{4} + |\mathbb{BR}| \right) \times O(2^Q) = O(|\mathbb{BR}|^2 \times 2^Q)$$

In this case the translation-based method, which we outlined in section 3 requires time $O(2^{2 \times |\mathbb{BR}| \times Q})$. In general, this upper bound is (to a great extend) inferior to the upper bound for $\textsc{ContextSat}$. If we take $|\mathbb{BR}| = 10$ and $Q = 5$, for instance, $\textsc{ContextSat}$ takes time in the order of 3200, while the translation-based approach may require a number of timesteps in the order of 10^{30}.

5 Towards an Efficient Implementation

Our algorithm can be implemented using several efficient off-the-shelf propositional reasoners. We sketch two particular ways to go.

5.1 BDD-based implementation

Reduced Ordered Binary Decision Diagrams [4] (or simply BDDs) constitute a canonical representation of propositional formulas. Equivalence of two BDDs can be computed in constant time; boolean transformation (e.g. conjunction, disjunction, negation) and quantification take at most quadratic time in the size of the BDDs involved. Efficient software libraries for the manipulation of BDDs, called BDD packages, are available. BDDs are being used in several application domains, ranging from formal verification [19] to planning [5], safety analysis [3], and diagnosis [24].

We use BDDs to represent sets of local models. The chain c we are constructing is implemented as an array, whose i^{th} element points to a BDD B_i, representing the set of local models comprised by c_i.

Initially, each B_i is equal to the BDD $True$. Extending c_i with a formula φ corresponds to replacing B_i by the conjunction of B_i and the BDD representation of φ. Checking for i-consistency requires an equality check between B_i and the BDD $False$. Determining whether B_i entails a bridge rule premise ψ can be done by comparing the BDD $B_i \supset \psi$ to the BDD $False$. A dedicated, more efficient routine to establish entailment is provided by most BDD packages.

Reasoning is always performed locally. Therefore, each context can be represented by a dedicated, completely independent BDD, each local proposition can be associated with a univocal "local" BDD variable, and each context can impose its own variable ordering.

The potential bottleneck of using BDDs is an explosion in space. In general practice, however, suitable variable orderings assure very compact representations of high-dimensional boolean functions.

5.2 SAT-based implementation

Propositional SAT solvers make up another very effective way to manipulate propositional formulas. The typical approach consists in a depth-first search for a satisfying truth value assignment, "splitting" on individual boolean variables [6]. During the last decade enormous progress has been achieved in this field: state-of-the-art SAT solvers are able to process problems with tenths of thousands variables and a million clauses [20], and are applied in several industrial settings ranging from formal verification [2] to planning [13] and ATPG [14].

In a SAT-based implementation the i^{th} component of the chain we are constructing is simply represented by a conjunction ψ_i of formulas that are forced to hold in context i. Initially, each ψ_i is empty. Extending c_i with a formula φ consists in conjuncting ψ_i with φ. Checking i-consistency now becomes a full-fledged call to the SAT solver, with ψ_i as input. Determining whether a bridge rule premise ϕ is entailed by c_i amounts to checking whether ϕ holds in all the models of ψ_i. This can be considered as a SAT problem, reasoning by refutation: ϕ is entailed by c_i iff $\psi_i \wedge \neg\phi$ is unsatisfiable.

Notice that the sequences of problems (consistency checks, premise entailment) presented to the SAT solver is incremental. A consistency / entailment check performed during the j^{th} iteration of CONTEXTSAT is often an extension of a consistency / entailment check carried out during some previous iteration of the algorithm. In this light, it is recommendable to exploit recent developments in *incremental* SAT technology [7]. Significant computational advances can be achieved by retaining learned conflict clauses when adding new clauses to an already processed formula.

6 Related Work

Work by Giunchiglia and Sebastiani [10] can be seen as a first step towards general decision procedures for contextual satisfiability. The objective of this work is to define SAT-based decision procedures for modal logics. Its motivation is highly associated with the possibility of defining a particular class of multi-context systems called hierarchical meta contexts, whose instances are equivalent to various modal logics [11]. Resulting procedures have been proven orders of magnitude faster than previous tableau-based decision procedures. In this paper, we've applied a similar approach to the class of multi-context systems, whose structure is not necessarily hierarchical.

Contextual reasoning (with finite sets of bridge rules) can be translated into a rather simple form of reasoning in multi-modal logic. Rephrasing every bridge rule $i_1 : \phi_1, \ldots, i_n : \phi_n \rightarrow i : \phi$ as a multi-modal implication $\Box_{i_1}\phi_1 \wedge \ldots \wedge \Box_{i_n}\phi_n \supset \Box_i\phi$, contextually satisfying a labeled formula $i : \varphi$ corresponds to modally satisfying a conjunction of the formula $\Box_i\varphi \wedge \neg\Box_i\bot$ and the translation of $\mathbb{BR}$. Notice that the modal *depth* of this conjunction is equal to one.

Fixpoint decision procedures for modal logic have been proposed by Pan, Sattler and Vardi [21], and could in principle be applied to contextual satisfiability as well. In this approach satisfiability of a modal formula φ is computed by constructing a Kripke structure, whose set of possible worlds is constituted by propositionally consistent sets of (possibly negated) sub-formulas of φ. Such sets are called *types*. A type/world a is accessible form a type/world b, if $\Box\phi \in b$ implies $\phi \in a$.

The top-down algorithm proposed in [21] takes as its initial set of worlds *all* possible types. Then, it iteratively discards those worlds/types which contain a formula $\neg\Box\psi$ but do not have access to any world/type containing $\neg\psi$. A type corresponding to a formula φ is represented by an array of binary variables each of which conveys whether the type contains either a certain sub-formula of φ, or its negation. This representation seems redundant as far as capturing the *propositional* structure of formulas is concerned. It turns out to be very effective, however, in treating the *modal* aspects of a problem. It is especially useful when processing formulas which exhibit deep nestings of modal operators. The encoding of contextual satisfiability problems into modal logic generates formulas, which do not exhibit any nesting at all. Therefore, directly applying this approach to our contextual setting does not seem to be a fruitful endeavor.

We'd like to remark that, if applied to a contextual satisfiability problem, the algorithm proposed in [21] takes exactly one iteration. More generally, in order to decide satisfiability of a formula with n nested modal operators, the algorithm performs at most n iterations.

Amir and McIlraith [1] define a propagation algorithm called MP, which computes satisfiability of a theory T that is partitioned into sub-theories (or *partitions*) $T_1, \ldots, T_n$. Partitions are related by the overlap between the signatures of their respective languages, which are called *communication languages* between these partitions. Roughly speaking, to check satisfiability of a partitioned theory $T_{i\leq n}$, MP determines a partial order $\prec$ over $T_{i\leq n}$, and subsequently - iterating over $T_{i\leq n}$ according to $\prec$, and propagating logical consequences of one partition to the next through the communication language between those two partitions - identifies models of T.

At a first glance, there is a strict analogy between multi-context systems and partitioned theories. Each partition could be seen as a context, and overlap between two partitions can be simulated via bridge rules of the form $i : p \rightarrow j : p$ and $i : \neg p \rightarrow j : \neg p$, where p is in the communication language between T_i and T_j. However, the analogy breaks at the semantical level. The semantics of a partitioned theory can be seen as the projection of a global semantics for T onto each local language T_i. Or, the other way around, a model for T is the combination of one model for each T_i. Conversely, a chain associates to every context a *set of local models*. Therefore, it cannot be considered as a set of chunks of a global model. In other words, in Amir and McIlraith's approach each T_i represents a partial theory of the world, while in ours each context represents an epistemic/belief state about the world. However, the analogy can be *made* to work, by only considering chains all of whose components contain exactly one local model. The two approaches should be compared subject to this hypothesis.

CONTEXTSAT, then, exhibits two main improvements w.r.t. MP. First, bridge rules express more complex relations between contexts (partitions) than communication languages do. For instance, we can relate three (or more) contexts via a bridge rule $i : \varphi, j : \psi \rightarrow k : \chi$, whereas MP is limited to considering the overlap between pairs of partitions. Furthermore, bridge rules are *directional*, i.e. $i : p \rightarrow j : p$ does not imply $j : p \rightarrow i : p$. Communication languages always describe *symmetric* relations between partitions. At last, whereas MP requires a partial order between contexts, CONTEXTSAT naturally deals with any kind of relational structure between them.

7 Conclusion

In this paper, we have investigated the satisfiability problem for propositional multi-context systems with finite sets of bridge rules. First, we provided a decision procedure based on the translation of contextual satisfiability into propositional logic. Next, we proposed an algorithm, which settles contextual satisfiability in a distributed fashion, exploiting the potential benefit of localizing reasoning and restricting it to relevant contexts only. We showed that this approach is more efficient than our translation-based procedure, and outlined how off-the-shelf reasoning platforms, like BDDs and propositional SAT solvers, can be used to implement our algorithm.

While designing our algorithm we have kept in mind a distributed peer-to-peer implementation. As a result, CONTEXTSAT is *modular*, i.e. global reasoning is made up of local reasoning procedures, and CONTEXTSAT is *backtrack-free*, i.e. solutions are build - or rather confined - incrementally, imposing a minimal restriction at each step. These features support a natural implementation in a peer-to-peer architecture, in which peers perform local reasoning and propagate their conclusions to neighbor peers via bridge rules. Modularity supports local reasoning, while backtrack-freeness avoids infinite loops.

REFERENCES

[1] E. Amir and S. McIlraith, 'Partition-based logical reasoning for first-order and propositional theories', *Artificial Intelligence*, (2004). Accepted for publication.

[2] A. Biere, A. Cimatti, E. M. Clarke, M. Fujita, and Y. Zhu, 'Symbolic model checking using SAT procedures instead of BDDs', in *Design Automation Conference (DAC 99)*, pp. 317–320, (1999).

[3] M. Bozzano and A. Villafiorita, 'Integrating fault tree analysis with event ordering information', in *European Safety and Reliability Conference (ESREL 03)*, pp. 247–254, (2003).

[4] R. E. Bryant, 'Symbolic Boolean manipulation with ordered binary-decision diagrams', *ACM Computing Surveys*, **24**(3), 293–318, (1992).

[5] A. Cimatti, M. Pistore, M. Roveri, and P. Traverso, 'Weak, strong, and strong cyclic planning via symbolic model checking', *Artificial Intelligence*, **147**(1/2), 35–84, (2003).

[6] S. Davis, G. Longemann, and D. Loveland, 'A machine program for theorem proving', *Journal of the ACM*, **5**(7), 394–397, (1962).

[7] N. Eén and N. Sörensson, 'An extensible SAT solver', in *International Conference on Theory and Applications of Satisfiability Testing (SAT 03)*, (2003).

[8] C. Ghidini and F. Giunchiglia, 'Local models semantics, or contextual reasoning = locality + compatibility', *Artificial Intelligence*, **127**(2), 221–259, (2001).

[9] F. Giunchiglia, 'Contextual reasoning', *Epistemologia*, **XVI**, 345–364, (1993).

[10] F. Giunchiglia and R. Sebastiani, 'Building decision procedures for modal logics from propositional decision procedures - the case study of modal k_m', *Journal of Information and Computation*, **162**(1/2), 158–178, (2000).

[11] F. Giunchiglia and L. Serafini, 'Multilanguage hierarchical logics, or: how we can do without modal logics', *Artificial Intelligence*, **65**(1), 29–70, (1994).

[12] R.V. Guha, *Contexts: A Formalization and some Applications*, Ph.D. dissertation, Stanford University, 1991.

[13] H. Kautz and B. Selman, 'Unifying SAT-based and graph-based planning', in *International Joint Conference on Artificial Intelligence (IJCAI 99)*, (1999).

[14] J.P. Marques-Silva and K.A. Sakallah, 'Robust search algorithms for test pattern generation', in *Symposium on Fault-Tolerant Computing*, pp. 152–161, (1997).

[15] F. Massacci, 'Contextual reasoning is NP complete', in *National Conference on Artificial Intelligence (AAAI 96)*, pp. 621–626, (1996).

[16] J. McCarthy, 'Generality in artificial intelligence', *Communications of ACM*, **30**(12), 1030–1035, (1987).

[17] J. McCarthy, 'Notes on formalizing context', in *International Joint Conference on Artificial Intelligence (IJCAI 93)*, pp. 555–560, (1993).

[18] J. McCarthy and S. Buvač, 'Formalizing context (expanded notes)', in *Computing Natural Language*, volume 81 of *CSLI Lecture Notes*, 13–50, (1998).

[19] K.L. McMillan, *Symbolic Model Checking*, Kluwer Academic Publ., 1993.

[20] M.W. Moskewicz, C.F. Madigan, Y. Zhao, L. Zhang, and S. Malik, 'Chaff: Engineering an efficient SAT solver', in *Design Automation Conference (DAC 01)*, pp. 530–535, (2001).

[21] G. Pan, U. Sattler, and M. Vardi, 'Bdd-based decision procedures for k', in *Conference on Automated Deduction (CADE 00)*, (2002).

[22] F. Roelofsen and L. Serafini, 'Complexity of contextual reasoning', in *National Conference on Artificial Intelligence (AAAI 04)*, (2004). Accepted for pulication.

[23] L. Serafini and P. Bouquet, 'Comparing formal theories of context in AI', *Artificial Intelligence*, **155**, 41–67, (2004).

[24] P. Torasso and G. Torta, 'Computing minimum-cardinality diagnoses using obdds', in *German Conference on Artificial Intelligence (KI 03)*, (2003).