UNIVERSITY
OF TRENTO

DEPARTMENT OF INFORMATION AND COMMUNICATION TECHNOLOGY

38050 Povo – Trento (Italy), Via Sommarive 14
http://www.dit.unitn.it

SATISFIABILITY FOR PROPOSITIONAL CONTEXTS

Luciano Serafini and Floris Roelofsen

April 2004

Technical Report # DIT-04-026

Satisfiability for Propositional Contexts

Luciano Serafini[1]
[1]ITC-IRST, Via Sommarive 18
38050 Povo, Trento, Italy
serafini@itc.it

Floris Roelofsen[1,2]
[2]University of Twente, P.O. box 217
7500 AE Enschede, Netherlands
roelofsen@itc.it

Abstract

We propose a sound and complete satisfiability algorithm for propositional multi-context systems. In essence, the algorithm is a distribution policy built on top of local reasoning procedures, one for each context, which can be implemented by (a diversity of) customized state-of-the-art SAT solvers. The foremost intuition that has motivated our algorithm, and the very potential strength of contextual reasoning, is that of keeping reasoning as *local* as possible. In doing so, we improve on earlier established complexity results by Massacci. Moreover, our approach could be applied to enhance recent proposals by Amir and Mcilraith towards a new partition-based reasoning paradigm; particularly, our formalism allows for a more expressive description of interpartition relations, and we provide an algorithm that is explicitly designed to deal with this expressiveness.

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 (1987) pleaded for a formalization of context as a possible solution to the problem of *generality*, whereas Giunchiglia (1993a) emphasized the principle of *locality* – reasoning based on large (common sense) knowledge bases can only be effectively pursued if confined to a manageable subset (context) of that knowledge base.

Contextual knowledge representation has been formalized in several ways. Most notable are the propositional logic of context developed by McCarthy, Buvač and Mason (1993; 1998), and the multi-context systems devised by Giunchiglia and Serafini (1994), which later became associated with the local model semantics (Ghidini & Giunchiglia 2001).

Contexts were first implemented as microtheories in the famed CYC common sense knowledge base (Guha 1991). However, while in CYC local microtheories were a choice, in contemporary settings like the semantic web the notion of local, distributed knowledge is a must. Modern architectures impose highly scattered, heterogeneous knowledge fragments, which a central reasoner is not able to deal with. This engenders a high demand for distributed, contextual reasoning procedures.

However, apart from few exceptions (Weyhrauch 1980; Massacci 1996), a general approach towards the automation of contextual reasoning has so far rarely been pursued. The pioneering work by Weyhrauch (1980) eventuated in an interactive multi-contextual theorem prover called FOL, which was later developed by Giunchiglia (1993b) into a more mature system called GETFOL. Both systems however, support automatic reasoning within a single context only. Cross contextual reasoning is left to their users.

Massacci (1996) was the first to propose a completely automatic tableaux-based decision procedure for contextual reasoning. This procedure however, leaves open a substantial number of efficiency issues and moreover, only applies to propositional logic of context (PLC).

We propose an automatic decision procedure called CSAT that computes satisfiability in multi-context systems (MCS). Furthermore, as MCS has recently been proven strictly more general than PLC (Bouquet & Serafini 2004), we show that CSAT can be applied to settle satisfiability in PLC as well.

The contribution of this paper, then, is threefold:

- CSAT is the first sound and complete decision procedure for propositional multi-context systems.
- CSAT is the first SAT-based decision procedure for contextual reasoning in general, and as such improves (in terms of complexity) both on Massacci's tableaux-based procedure for PLC, and on implicit results (based on equivalence results with modal logics) for MCS obtained from (Serafini & Giunchiglia 2002).
- Our approach could be applied to enhance recent proposals towards a new partition-based reasoning paradigm (Amir & McIlraith 2000; 2004); compared to alternative formalisms, MCS allows for more expressive descriptions of interpartition (intercontextual) relations, and CSAT is deliberately designed to deal with this expressiveness.

We proceed as follows. After defining propositional multi-context systems and their local model semantics, we explicate the contextual satisfiability problem and describe CSAT. Subsequently, we consider CSAT's computational complexity, and conclude with a discussion of the pros and cons of our approach in comparison with similar ones.

Multi-Context Systems

A simple illustration of the intuitions underlying MCS/LMS is provided by the so-called "magic box" example (Ghidini & Giunchiglia 2001), depicted below.

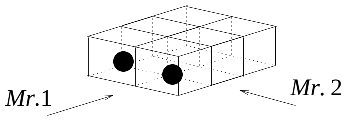

Figure 1: The magic box

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.1 thinks there's a ball in the box, Mr.2 should not believe it's empty.*

We show 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:

$$i_1 : \phi_1, \ldots, i_n : \phi_n \rightarrow i : \varphi \quad (1)$$

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 does not include expressions like $\neg(i : \varphi)$ and $(i : \varphi \wedge j : \psi)$. $i : \varphi$ is called the *consequence* and $i_1 : \phi_1, i_n : \phi_n$ are called *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, respectively.

Definition 1 (Propositional Multi-Context System MCS) *A propositional multi-context system* $\langle \{L_i\}_{i \in I}, \mathbb{BR} \rangle$ *over 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 (at most) countable and $\mathbb{BR}$ to be finite. Note that the latter assumption does not apply to MCSs with *schematic* bridge rules, such as provability - and multi-agent belief systems (Giunchiglia & Serafini 1994). The question whether our results may be generalized to capture these cases as well is subject to further investigation.

Example 2 *The MCS that formalizes the scenario specified in example 1 consists of two contexts* 1 *and* 2*, 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:*

$$2 : l \vee c \vee r \quad \rightarrow \quad 1 : l \vee r \quad (2)$$

Let M_i denote the class of classical interpretations of L_i. Each interpretation $m \in M_i$ is called a *local model* of L_i. Interpretations of entire MCSs 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*.* 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* $j \in J$*; It is point-wise if* $|c_i| \leq 1$ *for all* $i \in I$*; set-wise otherwise.*

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 unique way. So, set-wise chains correspond to partial knowledge, whereas point-wise chains indicate complete knowledge.

Example 3 *Consider the situation depicted in Figure 1. Both agents have complete knowledge, corresponding to a point-wise chain* $\{\{\{l, r\}\}, \{\{l, \neg c, \neg r\}\}\}$*. We can imagine a scenario however, in which Mr.1 and Mr.2's views are restricted to the right half and the left-most section of the box:*

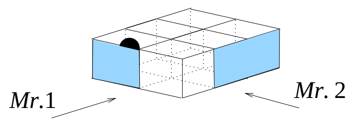

Figure 2: The partially hidden magic box

Now, both Mr.1 and Mr.2 have only partial knowledge; their observations may be interpreted in more than one way. This is reflected by the set-wise chain:

$$\left\{ \begin{array}{c} \{\{l, \neg r\}, \{\neg l, \neg r\}\}, \\ \{\{l, \neg c, \neg r\}, \{l, \neg c, r\}, \{l, c, \neg r\}, \{l, c, r\}\} \end{array} \right\}$$

The epistemic states that a chain consists of concern *one and the same* situation. Therefore, arbitrary sets of local models may not always constitute a "sensible" chain. The somewhat vague conception of "sensibility" is captured by the more formal 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 or not $i : \varphi$ is satisfiable in MS. In this paper, we assume φ to be in conjunctive normal form (CNF). Note however, that our results can easily be extended to the non-CNF case along the lines of (Armando & Giunchiglia 1993).

Example 4 *One way to think of a bridge rule is as a description of the information flow among different contexts. Bridge rule compliance, then, corresponds to effectuating that information flow. Consider example 3, for instance. Mr.2 knows that there is a ball in the box. Then, by bridge rule (2), Mr.1 must also know of the presence of that ball. He already knows that the right sector is empty, so he may conclude that there is a ball in the left sector of the box.*

Multi-context systems cannot be encoded into propositional logic by simply indexing propositions. Such an encoding of the following system, for instance, would be inconsistent.

Example 5 *Consider an MCS with two contexts* 1 *and* 2*, described by* $L(\{p\})$ *and* $L(\{q\})$*, respectively, and subject to 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:*

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

Henceforth 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.

MC Assignments

Instead of directly looking for a chain that satisfies $i:\varphi$ in MS, we will attempt to iteratively construct a so-called *MC assignment*. From this assignment, then, we will generate a suitable chain.

In this subsection, we first define MC assignments and their associated semantics. Then, we restate the satisfiability problem in terms of MC assignments, and specify a way to generate a chain that solves the original problem from an MC assignment that solves its reformulation. We show that this procedure is sound, and, finally, establish that a solution for the reformulated problem exists if and only if a solution for the original problem exists. Let us first introduce the necessary terminology.

Definition 4 *A local truth value assignment* π_i *is a function that assigns truth values (*$true$ *or* $false$*) to propositional atoms of* L_i*. We call* π_i *complete if it assigns a truth value to every propositional atom of* L_i*, and partial otherwise. In the special case that* π_i *does not assign a truth value to any of the propositional atoms of* L_i*, it is called empty.*

A local truth value assignment π_i is represented by a set of literals. This set contains an atom p iff $\pi_i(p) = true$ and its negation $\neg p$ iff $\pi_i(p) = false$.

Definition 5 *An MC local assignment* Π_i *is a set of local truth value assignments. We say that* Π_i *is consistent if it is nonempty, and inconsistent otherwise. An inconsistent MC local assignment is denoted* $\bot$*. We call* Π_i *complete if it contains exactly one local truth value assignment, and if, moreover, this local truth value assignment is complete. Otherwise, we call* Π_i *partial.*

Definition 6 (MC Assignment) *An MC assignment* Π *over a set of indices* I *is a sequence* $\{\Pi_i\}_{i\in I}$ *of MC local assignments. An MC assignment* Π *is* i*-consistent if* Π_i *is consistent; it is* J*-consistent, for some* $J \subseteq I$*, if it is* j*-consistent for all* $j \in J$*. We call* Π *complete if all its elements are complete; otherwise,* Π_i *is called partial.*

Definition 7 (Bridge Rule Compliance and Satisfiability) *Let* π_i *be a local truth value assignment,* Π_i *an MC local assignment,* Π *an MC assignment, and* φ *a formula in* L_i*.*

1. π_i *satisfies* φ *if* $\bigwedge \pi_i \models \varphi$*, where* $\bigwedge \pi_i$ *is the conjunction of all the elements of* π_i*. We write* $\pi_i \models \varphi$*.*
2. Π_i *satisfies* φ *if* $\pi_i \models \varphi$ *for all* $\pi_i \in \Pi_i$*. We write* $\Pi_i \models \varphi$*.*
3. Π *satisfies* $i:\varphi$ *if* $\Pi_i \models \varphi$*. We write* $\Pi \models i:\varphi$*.*
4. Π *complies with* $\mathbb{BR}$ *if for all* $br \in \mathbb{BR}$ *either* $\Pi \models cons(br)$ *or* $\Pi \not\models i:\xi$ *for some* $i:\xi \in prem(br)$*.*

If there is an i-consistent MC assignment Π that satisfies $i:\varphi$ and complies with $\mathbb{BR}$, we would like to be able to automatically generate from Π a chain that satisfies $i:\varphi$ in MS. This is established as follows:

Definition 8 (Generated Chain) *The set of local models* m^{Π_i} *generated by an MC local assignment* Π_i *consists of exactly those local models of* L_i *that satisfy all the elements of one of the local truth value assignments in* Π_i*. Formally:*

$$m^{\Pi_i} = \{m \in M_i \mid m \models l \text{ for all literals } l \text{ in some } \pi_i \in \Pi_i\}$$

The chain c^{Π} *generated by an MC assignment* Π *is obtained by assigning to every component* c_i^{Π} *of* c^{Π} *the set of local models* m^{Π_i} *generated by* Π_i*.*

Note that an inconsistent MC local assignment generates an empty set of local models. Moreover, a complete MC assignment generates a point-wise chain (corresponding to complete knowledge), whereas a partial MC assignment generates a set-wise chain (indicating partial knowledge).

It is quite straightforward to see that if Π is i-consistent, then c^{Π} is i-consistent. Moreover, if Π complies with a set of bridge rules, then c^{Π} does so too. Finally, if $\Pi \models i:\varphi$, then $c^{\Pi} \models i:\varphi$ holds as well. From these observations we directly obtain the following result.

Proposition 1 (Soundness) *If* Π *is an* i*-consistent MC assignment that satisfies* $i:\varphi$ *and complies with* $\mathbb{BR}$*, then* c^{Π} *satisfies* $i:\varphi$ *in* MS*.*

The opposite holds as well. From a chain c we may obtain an MC assignment Π, whose elements Π_i contain local truth value assignments, each of which directly represents a local model in c_i. If all local models in c_i are represented by a local truth value assignment in Π_i, then it is easy to see that this construction preserves i-consistency, satisfaction, and bridge rule compliance. This leads to the following result.

Proposition 2 (Completeness) *If there is a chain that satisfies* $i:\varphi$ *in* MS*, then there is also an* i*-consistent MC assignment* Π *that satisfies* $i:\varphi$ *and complies with* $\mathbb{BR}$*.*

In summary, in order to solve the contextual satisfiability problem we may first attempt to solve its reformulation in terms of MC assignments. If a suitable MC assignment is determined, we can generate a chain that, by proposition 1, constitutes a solution of the original problem. On the other hand, if no such MC assignment exists, proposition 2 tells us that $i : \varphi$ is unsatisfiable.[1]

CSAT

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

- Π^1 satisfies $i : \varphi$.
- for all $m \in \{2, \ldots, k\}$, Π^m complies with the bridge rules that Π^{m-1} does not comply with.
- for all $m \in \{1, \ldots, k\}$, Π^m extends Π^{m-1} in the following sense:

Definition 9 (Extension) *Let π_i and π'_i be two local truth value assignments, Π_i and Π'_i two MC local assignments, and Π and Π' two MC assignments over a set of indices I.*

1. *π'_i extends π_i if $\pi_i \subseteq \pi'_i$.*
2. *Π'_i extends Π_i if for every $\pi'_i \in \Pi'_i$ there is some $\pi_i \in \Pi_i$ such that π'_i extends π_i.*
3. *Π' extends Π if Π'_i extends Π_i for every $i \in I$.*

It is useful to observe that Π'_i is an extension of Π_i if and only if $m^{\Pi'_i}$ is a subset of m^{Π_i}. So extending an MC local assignment means restricting the set of local models that is generated by that assignment.

This observation has two important implications. First, we obviously want our initial assignment Π^0 to be most "general", that is, we don't want it to be an extension of any other assignment. The only assignment exhibiting this property is the one all of whose components consist of an empty truth value assignment: $\Pi^0 = \{\{\emptyset\}, \ldots, \{\emptyset\}\}$. Note that this assignment corresponds to a chain c^{Π^0} all of whose components $c_i^{\Pi^0}$ contain the entire set of local models M_i. Moreover, note that Π^0 doesn't satisfy any formula. This means, in particular, that Π^0 doesn't satisfy any bridge rule premise, and therefore complies with $\mathbb{BR}$.

The second implication of always *extending* an assignment, and thus restricting the corresponding set of local models, is that once a formula is satisfied by some intermediate assignment Π^m, then it is also satisfied by Π^n, for any $n > m$. This means that (1) if $i : \varphi$ is satisfied by Π^1, then it is also satisfied by Π^m, for any $m \in \{1, \ldots, k\}$. Moreover, (2) if some intermediate assignment Π^m does not comply with a bridge rule $br \in \mathbb{BR}$ - that is, Π^m satisfies br's premises, but does not satisfy its consequence - then any extension of Π^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 Π^{m+1} from Π^m consists in extending Π^m so as to satisfy the consequences of the bridge rules that Π^m does not comply with. Finally, (3) once an intermediate assignment 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.

[1]Note that it remains to be shown that a suitable MC assignment exists only if our algorithm finds one.

Algorithm 1 CSAT

```
CSAT(Φ, Π, BR, I, J)
begin
  I* := {i ∈ I | i : φ ∈ Φ};
  Π* := { Π*_i |  Π*_i = LSAT(φ, Π_i) for i ∈ I*
                  Π*_i = Π_i for i ∈ I/I*        };
  for all j ∈ J do
    if Π*_j = ∅ then
      return False;
    end if
  end for
  BR* := {br ∈ BR | Π* ⊨ i : η forall i : η ∈ prem(br)}
  if BR* = ∅ then
    return Π*;
  end if
  Ψ* := {cons(br) | br ∈ BR*};
  Φ* := { i : φ | φ = ⋀_{i:ξ∈Ψ*} ξ , i ∈ I };
  return CSAT(Φ*, Π*, BR/BR*, I, J);
end
```

This approach is implemented by the CSAT procedure, specified in Algorithm 1. For the sake of generality, and to provide for elegant recursion, CSAT is designed to settle satisfiability of a *set* of labeled formulas. Apart from this set of formulas Φ, it takes as its input an MC assignment Π, a set of bridge rules $\mathbb{BR}$, a set of contexts (indices) I, and finally, a subset $J \subseteq I$ of contexts whose consistency is required.

CSAT is called with Π being the MC assignment over I all of whose components consist of an empty truth value assignment, and yields a J-consistent extension Π^* of Π that satisfies every formula in Φ in compliance with $\mathbb{BR}$, or $False$ if it fails to construct such an MC assignment.

Extensions are always constructed *locally*. That is, CSAT first determines the set I^* of contexts concerned by formulas in Φ, and then, for every $i \in I^*$, calls a sub-procedure LSAT that extends Π_i so as to satisfy $i : \varphi$. The local extensions obtained in this way are simply taken together to form a "global" extension Π^* of Π.

If Π^* is J-inconsistent, any further extension of Π^* will be J-inconsistent as well. Therefore, if such is the case CSAT recognizes a failure, and returns $False$. Otherwise, CSAT determines the set $\mathbb{BR}^*$ of bridge rules *all* of whose premises are satisfied by Π^*. If $\mathbb{BR}^*$ is empty, Π^* is a solution. Otherwise, making Π^* comply with $\mathbb{BR}^*$ yields a new satisfiability problem, namely that of extending Π^* 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 CSAT is addressed to extend Π^* so as to satisfy Φ^*. Recursively proceeding like this, an MC assignment is constructed that, at any stage, satisfies the formulas in Φ, and at some point either becomes J-inconsistent, or complies with the entire set of bridge rules $\mathbb{BR}$.

Algorithm 2 LSAT

LSAT(φ, Π_i)
begin
return $\bigcup_{\pi_i \in \Pi_i}$ DPLL(φ, π_i);
end
DPLL(φ, π_i)
begin
if $\pi_i \models \varphi$ **then**
  return $\{\pi_i\}$;
else if $\pi_i \models \neg\varphi$ **then**
  return $\emptyset$;
else if $assign(\varphi, \pi_i)$ contains a unit clause l **then**
  return DPLL($\varphi, \pi_i \cup \{l\}$);
else
  $l := chooseUnassignedLiteral(\varphi, \pi_i)$;
  return DPLL($\varphi, \pi_i \cup \{l\}$) $\cup$ DPLL($\varphi, \pi_i \cup \{\neg l\}$)
end if
end

LSAT takes as its input a formula φ and an MC local assignment Π_i, and yields a consistent extension of Π_i that satisfies φ, or the empty set if it fails to construct such an extension.

Its foremost principle, and the very strength of contextual reasoning, is that of *locality*. The efficiency of a contextual reasoning process ensues from restricting its resources to a small number of contexts. LSAT is designed accordingly. Concretely, this amounts to constructing an assignment Π_i^* that does not *unnecessarily* satisfy any bridge rule premises. In this way the chance of having to re-establish bridge rule compliance is minimized, and therefore reasoning in other contexts is required only if strictly necessary.

It is again useful to reformulate this idea in terms of sets of local models: the desired extension Π_i^* of Π_i should correspond to the set of local models $m^{\Pi_i^*}$ that is obtained by removing from m^{Π_i} *exactly* those local models that do not satisfy φ. This constraint is settled by requiring Π_i^* to be a *complete extension* of Π_i *with respect to* φ.

Definition 10 (Complete Extension) *An extension Π_i^* of an MC local assignment Π_i is called complete with respect to a formula φ if for every complete local truth value assignment π_i' that extends some $\pi_i \in \Pi_i$ and satisfies φ, there is an element π_i^* of Π_i^* so that π_i' is an extension of π_i^*.*

LSAT indeed constructs complete extensions. In fact, LSAT returns the union of the extensions of all the truth value assignments contained by Π_i. These extensions are determined by yet another sub-procedure, called DPLL, which is a variant of the Davis-Putnam-Longemann-Loveland SAT procedure (Davis, Longemann, & Loveland 1962). The so called "pure literal rule" is left out to avoid incomplete extensions, and instead of returning one suitable local truth value assignment (or $False$ if such an assignment does not exist), DPLL yields a compact representation of the set of *all* suitable local truth value assignments (which may be the empty set). The "compactness" of this representation is formally characterized by the notion of *strong non-redundancy*.

Definition 11 (Strong Non-Redundancy) *An MC local assignment Π_i is called strongly non-redundant if for every $\pi_i, \pi_i' \in \Pi_i$, there is a literal l so that $l \in \pi_i$ and $\neg l \in \pi_i'$. In this case, π_i and π_i' are mutually inconsistent.*

DPLL is based on what is called *semantic branching*, that is, branching on truth values of propositional variables. As a result, each branching step generates two mutually inconsistent local truth value assignments, and because of this, DPLL yields a strongly non-redundant set of local truth value assignments. By induction the same holds for LSAT.

As argued and empirically supported in (D'Agostino & Mondadori 1994; Giunchiglia & Sebastiani 1996; 2000) this characteristic implies a significant and fundamental gain of efficiency with respect to reasoning procedures based on what is called *syntactic branching*, that is, branching on the syntactic structure of the to-be-satisfied formula. This has motivated us to use DPLL as a foundation of our algorithm, rather than, for instance, a (syntactically branching) tableau.

Example 6 *Let us describe a simple simulation of the algorithm. Consider example 3. Mr.1 knows that there is no ball in the right section of the box ($1 : \neg r$); Mr.2 knows that there is a ball in the leftmost section of the box ($2 : l$). We let* CSAT *determine whether $\Psi = \{1 : \neg r, 2 : l\}$ is satisfiable or not. It proceeds as follows.*
First, an assignment Π^1 is determined so as to satisfy Ψ:

$$\Pi^1 = \{\{\{l, \neg r\}, \{\neg l, \neg r\}\}, \{\{l\}\}\}$$

Then, to make it comply with bridge rule (2):

$$2 : l \vee c \vee r \rightarrow 1 : l \vee r$$

Π^1 is extended so as to satisfy $1 : l \vee r$. We obtain:

$$\Pi^2 = \{\{\{l, \neg r\}\}, \{\{l\}\}\}$$

which complies with bridge rule (2), and indeed conveys that Mr.1 knows of the ball's presence in the left section of the box (as established earlier in example 4).

Termination, Soundness and Completeness

The set of bridge rule consequences that is satisfied by the MC assignment which is being constructed is strictly expanded by every recursive call to CSAT. Since there is only a finite number of bridge rule consequences that are to be satisfied, CSAT is therefore bound to terminate.

Soundness and completeness of CSAT are easily derived. To see that if a contextual satisfiability problem can be solved, a suitable MC assignment will be constructed by CSAT, it is crucial to realize that every iteration yields a *complete* extension of the assignment constructed so far. On the other hand, if CSAT produces an MC assignment, we already know by proposition 1 that satisfiability is veritable.

Computational Complexity

We have presented CSAT in deterministic form. The lower bound for the complexity of deterministic solutions to the classical (non-contextual) satisfiability problem is $O(2^{P(\varphi)})$, where φ is the to-be-satisfied formula and $P(\varphi)$ is the number of propositional variables occurring in φ.

The complexity of CSAT is given not only in terms of the formula whose satisfiability is to be determined, but also in terms of the bridge rules of the MCS that is considered. Let $\Psi(\mathbb{BR})$ denote the set of consequences of bridge rules in $\mathbb{BR}$. We immediately obtain the following complexity results:

$$\begin{aligned} tc(\text{DPLL}(\varphi, \pi_i)) &= O(2^{P(\varphi)}) \\ tc(\text{LSAT}(\varphi, \Pi_i)) &= O(2^{P(\varphi)}) \\ tc(\text{CSAT}(\Phi, \Pi, \mathbb{BR}, I, J)) &= O(\max_{i:\varphi \in \Phi \cup \Psi(\mathbb{BR})} 2^{P(\varphi)}) \end{aligned}$$

Notice that CSAT is *optimal* in that it does not exceed the lower bound inflicted by the complexity of classical SAT. Moreover, the only parameters of CSAT that exert an influence on its complexity, apart from the input formula itself, are the *consequences* of the bridge rules in $\mathbb{BR}$. That is, bridge rule premises can be unlimitedly intricate, without having any impact on CSAT's overall complexity.

We now construct a non-deterministic variant of CSAT that operates in polynomial time. First, we provide a non-deterministic version of LSAT. Observe that LSAT's essential task is to produce an extension Π_i^* of a given MC local assignment Π_i that satisfies, and is complete with respect to a given formula φ. This is established non-deterministically by proceeding as follows:

1. Generate an extension Π_i^* of Π_i
2. Check satisfaction: for every local truth value assignment $\pi_i^* \in \Pi_i^*$, verify that $\pi_i^* \models \varphi$
3. Check completeness of Π_i^* with respect to φ: for every complete local truth value assignment π_i' which extends some $\pi_i \in \Pi_i$, but which does *not* extend any $\pi_i^* \in \Pi_i^*$, check if $\pi_i' \not\models \varphi$, or equivalently, as π_i' is complete, if $\pi_i' \models \neg\varphi$

Unfortunately, all these processes may require exponential time, as the number of possible ways to appoint an extension Π_i^* of Π_i (process 1), as well as the number of local truth value assignments π_i^* and π_i' for which $\pi_i^* \models \varphi$ and $\pi_i' \models \neg\varphi$ must be verified (process 2 and process 3, respectively) may be exponential.

We will show, however, that CSAT can be adapted such that this is never the case. To this end, we return to viewing the course of our algorithm in terms of local models. As we remarked earlier, extending an MC local assignment Π_i corresponds to removing certain local models from m^{Π_i}. Then, to verify that Π_i^* satisfies φ is to make sure that every local model contained in $m^{\Pi_i^*}$ does so. Similarly, checking whether an extension Π_i^* of Π_i is complete with respect to φ consists in verifying whether the models that were removed from m^{Π_i} did indeed not satisfy φ. Now if only $|m^{\Pi_i}|$ was non-exponential to start with, then extending Π_i (process 1), and checking for satisfaction (process 2) and completeness (process 3) could surely be executed in polynomial time. By the following proposition, this may in fact be assured.

Proposition 3 (Bounded Model Property) *A set of formulas Φ is satisfiable in a multi-context system MS iff it is satisfied by a chain that contains at most $|\Phi|+|\mathbb{BR}|$ local models.*

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

So the non-deterministic variant of CSAT may take as its initial MC assignment, instead of $\Pi^0 = \{\{\emptyset\}, \ldots, \{\emptyset\}\}$ (corresponding to the chain each of whose components is the entire set of local models M_i of L_i), an assignment $\Pi^{\mathbb{BR}}$ that corresponds to a chain $c^{\mathbb{BR}}$, which contains at most $|\Phi|+|\mathbb{BR}|$ local models. Clearly, any component of $c^{\mathbb{BR}}$ contains at most $|\Phi| + |\mathbb{BR}|$ local models as well. In this case, process 1 requires time $O(|\Phi| + |\mathbb{BR}|)$, whereas process 2 and 3 are guaranteed to terminate in $O((|\Phi| + |\mathbb{BR}|) \times |\varphi|)$. As a result, the non-deterministic variant of LSAT described above terminates in time $O((|\Phi| + |\mathbb{BR}|) \times |\varphi|)$.

The corresponding version of CSAT consists in first constructing an initial assignment $\Pi^{\mathbb{BR}}$ and then non-deterministically executing LSAT at most $|\mathbb{BR}|$ times.

Constructing $\Pi^{\mathbb{BR}}$, or its associated chain $c^{\mathbb{BR}}$, merely involves a non-deterministic assignment of truth values to the atomic propositions in $(P(\Phi) \cup P(\mathbb{BR}))$ for each of its (at most $|\Phi| + |\mathbb{BR}|$) local models. This can be done in time $O((|\Phi| + |\mathbb{BR}|) \times (|P(\Phi)| + |P(\mathbb{BR})|))$.

Contextual satisfiability clearly subsumes classical satisfiability and is therefore NP-hard (Cook 1971). Putting things together we obtain:

Theorem 1 *Contextual satisfiability is NP-complete. Satisfiability of a set of formulas Φ in compliance with a set of bridge rules $\mathbb{BR}$ can be determined in non-deterministic polynomial time:*

$$O((|\Phi| + |\mathbb{BR}|) \times \left(\begin{array}{c} (|P(\Phi)| \quad + \quad |P(\mathbb{BR})|) \\ + \\ (\max_{i:\varphi \in \Phi \cup \Psi(\mathbb{BR})} |\varphi| \times |\mathbb{BR}|) \end{array} \right)$$

Related Work

Recent work by Giunchiglia and Sebastiani (2000) can be considered as a first step towards general decision procedures for contextual satisfiability. The aim 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 (Giunchiglia & Serafini 1994). Resulting procedures have been proven orders of magnitude faster than previous tableau-based decision procedures.

CSAT can be seen as a generalization of this approach to the entire class of multi-context systems, as opposed to the particular class of hierarchical meta contexts.

Massacci (1996) introduced a tableaux-based procedure that determines satisfiability in PLC. He established a non-deterministic time complexity $O(|\varphi|^4)$, where $|\varphi|$ is the length of the to-be-satisfied formula. CSAT is designed to determine satisfiability in MCS/LMS. However, as PLC can be embedded in MCS/LMS (Bouquet & Serafini 2004), CSAT may be applied to settle satisfiability in PLC as well. In doing so we attain the following complexity result:

Theorem 2 *Satisfiability of a formula φ in PLC can be computed in non-deterministic polynomial time $O(|\varphi|^3)$.*

Let φ be a formula in PLC. We construct a multi-context system MCS(φ) and a labeled formula $\epsilon : \varphi$, so that $\epsilon : \varphi$ is satisfiable in MCS(φ) iff φ is satisfiable in PLC. For every nesting pattern $ist(k_1, \ldots, ist(k_n, \psi) \ldots)$ in φ, let MCS(φ) contain a context labeled with the sequence $k_1 \ldots k_n$. Let the language of context $k_1 \ldots k_n$ consist of all the atomic propositions in ψ, in addition to a new atomic proposition for each formula of the form $ist(k, \chi)$ occurring in ψ. Finally, equip MCS(φ) with the following bridge rules[2]:

$$\begin{array}{l} \bar{k}k : \psi \rightarrow \bar{k} : ist(k,\psi) \\ \bar{k} : ist(k,\psi) \rightarrow \bar{k}k : \psi \\ \bar{k} : \neg ist(k, ist(h,\psi)) \rightarrow \bar{k}k : \neg ist(h,\chi) \\ \bar{k} : \neg ist(k, \neg ist(h,\psi)) \rightarrow \bar{k}k : ist(h,\chi) \end{array}$$

where $\bar{k} = k_1 \ldots k_n$ refers to any context of MCS(φ), whose language contains $ist(k,\psi)$ or $ist(k, ist(h,\chi))$, respectively.

Example 7 *Consider $\varphi = p \vee ist(k, q \supset (ist(h, r \wedge s) \supset ist(j,q)))$. MCS($\varphi$) consists of four contexts which are labeled ϵ (the empty sequence), k, kh, and kj, respectively. The language of ϵ, L_ϵ, contains two propositions, p and $ist(k, q \supset (ist(h, r \vee s) \supset ist(j,q)))$; L_k contains two propositions, q and $ist(h, r \wedge s)$; $L_{kh} = L(\{r,s\})$; $L_{kj} = L(\{q\})$. The bridge rules of MCS(φ) are as stated above.*

Proposition 4 (Bouquet & Serafini, 2004) *φ is satisfiable in PLC if and only if $\epsilon : \varphi$ is satisfiable in MCS(φ).*

Proof for theorem 2. By proposition 4 satisfiability problems in PLC can be transformed into equivalent satisfiability problems in MCS/LMS. This transformation can be established in linear time.

Every bridge rule in MCS(φ) involves at least one proposition of the form $ist(k,\psi)$. Every such proposition occurs in at most four bridge rules. Every subformula of φ of the form $ist(k,\psi)$ (and nothing else) results in a proposition of the form $ist(k,\psi)$ in the language of exactly one context in MCS(φ). The number of subformulas of φ of the form $ist(k,\psi)$ is bounded by $|\varphi|$. From these observations, we may conclude that the number of bridge rules $|\mathbb{BR}|$ of MCS(φ) is bounded by $4 \times |\varphi|$. By construction, the number of propositional atoms involved in any bridge rule of MCS(φ) is at most two. Furthermore, the length $|cons(br)|$ of the consequence of any bridge rule $br \in \mathbb{BR}$ is bounded by $|\varphi|$. This implies that $\max_{i:\psi \in \{\epsilon:\varphi\} \cup \Psi(\mathbb{BR})} |\psi| = |\varphi|$.

By theorem 1, satisfiability of $\Phi = \{\epsilon : \varphi\}$ in MCS(φ) can be determined in time:

$$O((|\Phi| + |\mathbb{BR}|) \times \left(\begin{array}{c} (|P(\Phi)| \quad + \quad |P(\mathbb{BR})|) \\ + \\ (\max_{i:\psi \in \Phi \cup \Psi(\mathbb{BR})} |\psi| \times |\mathbb{BR}|) \end{array} \right)$$

In the light of the above observations this is reducible to:

$$O(|\varphi|^3)$$

□

Recently Amir and McIlraith (2000; 2004) have defined forward/backward propagation algorithms (called MP and BMP, respectively) that compute satisfiability of a theory T, which 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 is called the *communication language* between these partitions. Roughly speaking, to check satisfiability of a partitioned theory $T_{i \le n}$ (B)MP determines a partial order $\prec$ over $T_{i \le n}$, and then - iterating over $T_{i \le n}$ according to $\prec$, and propagating logical consequences of one partition to the next through the communication language between two consecutive partitions - determines all the models of $T_{i \le n}$.

At a first glance, there is a strict analogy between partitioned theories and multi-context systems. Each partition can 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 considering only I-consistent point-wise chains. So the two approaches should be compared subject to this hypothesis.

CSAT, then, exhibits two main improvements w.r.t. (B)MP. First, bridge rules allow us to express more complex relations between contexts (partitions) than communication languages do. For instance, we can relate three (or more) contexts via a bridge rule $i : \varphi, j : \psi \rightarrow k : \chi$, while (B)MP is limited to considering 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 can only describe *symmetric* relations between partitions. At last, CSAT is more general than (B)MP in that it does not require any partial order between contexts. It naturally deals with any kind of relational structure between them.

[2]The first two bridge rules correspond to the notions of *entering* and *exiting* contexts (McCarthy & Buvač 1998), while the others correspond to the Δ axiom introduced in (Buvač & Mason 1993).

Conclusion

This paper describes CSAT, an algorithm which computes satisfiability in propositional multi-context systems. Both from a representational and a computational point of view CSAT is an improvement on state-of-the-art contextual/ modular reasoning systems. It brings down the worst case complexity bound of contextual satisfiability to non-deterministic $O(|\phi|^3)$ (from the earlier established $O(|\phi|^4)$). Moreover, from a representational perspective, it computes satisfiability of a set of theories interacting via bridge rules, as opposed to a set of simply overlapping partial theories. Finally, while designing our algorithm we have kept in mind a distributed peer-to-peer implementation. As a result, CSAT is *modular*, i.e. global reasoning consists in composing local reasoning procedures, and CSAT is *backtrack-free*, i.e. solutions are build - or rather confined - incrementally, imposing a minimal restriction at every step. These features support a natural implementation of CSAT in a peer-to-peer architecture, in which each peer performs local reasoning within a context and propagates its conclusions to neighbor peers via bridge rules. Modularity supports local reasoning; backtrack-freeness avoids infinite loops.

Acknowledgments

We wish to thank Johan van Benthem, Roberto Sebastiani, and the anonymous referees for their very useful comments.

References

Amir, E., and McIlraith, S. 2000. Partition-based logical reasoning. In *Principles of Knowledge Representation and Reasoning (KR 00)*, 389–400.

Amir, E., and McIlraith, S. 2004. Partition-based logical reasoning for first-order and propositional theories. *Artificial Intelligence*. Accepted for publication. Preprint available at www.cs.berkeley.edu / ~ eyal / papers / oo-reason-general-aij2001.pdf.

Armando, A., and Giunchiglia, E. 1993. Embedding complex decision procedures inside an interactive theorem prover. *Annals of Mathematics and Artificial Intelligence* 8(3/4):475–502.

Bouquet, P., and Serafini, L. 2004. Comparing formal theories of context in AI. *Artificial Intelligence*. Accepted for publication.

Buvač, S., and Mason, I. A. 1993. Propositional logic of context. In *National Conference on Artificial Intelligence (AAAI 93)*.

Cook, S. 1971. The complexity of theorem proving procedures. In *ACM Symposium on the Theory of Computing*, 151–158.

D'Agostino, M., and Mondadori, M. 1994. The taming of the cut: Classical refutations with analytic cut. *Journal of Logic and Computation* 4:285–319.

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

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

Giunchiglia, F., and Sebastiani, R. 1996. A sat-based decision procedure for ALC. In *Principles of Knowledge Representation and Reasoning (KR 96)*, 304–314.

Giunchiglia, F., and Sebastiani, R. 2000. 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.

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

Giunchiglia, F. 1993a. Contextual reasoning. *Epistemologia* XVI:345–364.

Giunchiglia, F. 1993b. GETFOL: Interactive multicontext theorem proving (abstract). In *Workshop on Automated Theorem Proving (IJCAI 93)*, 43–43.

Guha, R. 1991. *Contexts: A Formalization and some Applications*. Ph.D. Dissertation, Stanford University.

Massacci, F. 1996. Contextual reasoning is NP complete. In *National Conference on Artificial Intelligence (AAAI 96)*, 621–626.

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

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

McCarthy, J. 1993. Notes on formalizing context. In *International Joint Conference on Artificial Intelligence (IJCAI 93)*, 555–560.

Serafini, L., and Giunchiglia, F. 2002. Ml systems: A proof theory for contexts. *Journal of Logic, Language and Information* 11(4):471–518.

Weyhrauch, R. 1980. Prolegomena to a theory of mechanized formal reasoning. *Artificial Intelligence* 13(1):133–176.