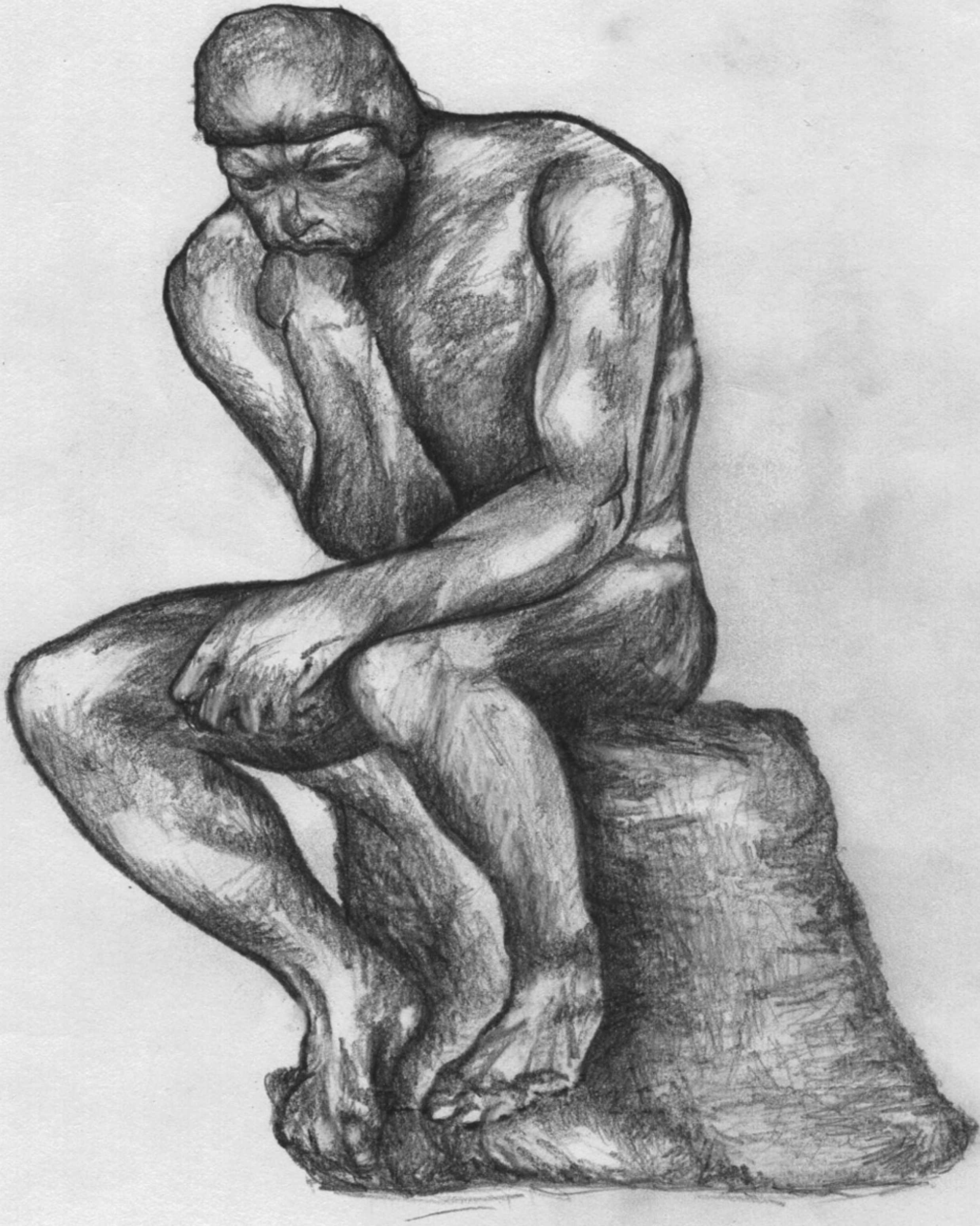
Floris Roelofsen



Contextual Reasoning

Complexity Analysis
Decision Procedures

Front cover by Mirjam Roelofsen-Cohen.

Contextual Reasoning

- Complexity Analysis and Decision Procedures -

Floris Roelofsen

Master Thesis

Human Media Interaction
Department of Computer Science
University of Twente
The Netherlands

* * *

Supervised by:
Luciano Serafini
Anton Nijholt
Jan Kuper

* * *

Enschede, July 2004

Abstract

Formal accounts of contextual reasoning are of great importance for the development of sophisticated Artificial Intelligence theory and applications. This thesis' contribution to the theory of contextual reasoning is twofold.

First, it delineates the computational complexity of contextual reasoning. A first insight is obtained by translating contextual reasoning into a rather simple form of reasoning in bounded modal logic. A more direct and general understanding, as well as more refined complexity results, are established by achieving the so-called bounded model property for contextual satisfiability.

Second, the thesis describes two conceptually orthogonal approaches to automatically deciding satisfiability in a contextual setting. Firstly, the bounded model property is exploited so as to encode contextual satisfiability into propositional satisfiability. This approach provides for the implementation of contextual reasoners based on existing propositional Sat solvers. Subsequently, a distributed decision procedure is proposed, which maximally exploits the potential amenity of localizing reasoning and restricting it to relevant contexts only. The latter approach is shown to be computationally superior to the former translation based procedure, and can be implemented using off-the-shelf efficient reasoning procedures.

Acknowledgments

This thesis is the result of a graduation project carried out at the Institute for Scientific and Technological Research (IRST) in Trento, Italy. Ever since its outset in the late 1980's, fundamental AI research on contextual reasoning has always been pursued most notably by two prominent research groups: one at Stanford, conducted by John McCarthy, and one in Trento, led by Fausto Giunchiglia and Luciano Serafini.

Throughout this project, I have been inspired and advised by Luciano. I think I have been extremely fortunate to have met and worked with him. He has always been readily available, enthusiastically involved, disposed and very sincere. Also, I am indebted to Roberto Sebastiani and Alessandro Cimatti, both at IRST, to Johan van Benthem at the University of Amsterdam and Stanford, and to Jan Kuper at the University of Twente for their comments on earlier drafts and presentations of this work. Moreover, I wish to thank Bernardo Magnini and Oliviero Stock, both at IRST, and Anton Nijholt at the University of Twente for coordinating my visit to Trento.

Parts of this work will be presented at various conferences [58, 59, 60, 62]. Financial support to attend these events has been provided generously by the American Association for Artificial Intelligence, the Canadian Society for the Computational Studies of Intelligence, the Society for the Study of Artificial Intelligence and the Simulation of Behaviour, the European Association for Logic, Language, and Information, IRST, and the University of Twente.

Contents

Chapter 1

Introduction

The goal of this thesis is to advance the understanding of the computational complexity of contextual reasoning, and to contribute to the development of computer programs which materialize contextual reasoning processes. The methodology we adhere to may best be qualified as logic-based Artificial Intelligence. In this chapter we briefly situate and motivate our investigation.

1.1 Logic-Based Artificial Intelligence

The field called Artificial Intelligence (AI) incorporates various aspirations. Some researchers want machines to do things that people call intelligent (making plans, communicating and cooperating with other computers and people, making and understanding jokes, directing movies, playing drums). Others seek to understand what enables people to do such things.

Marvin Minsky, one of the field's very first pioneers, classifies the various attempts that have been pursued to do so into two main categories [52]. The first is called connectionism [28]. Its core idea is to embody knowledge by configurations of the connections in a network of interconnected nodes. Networks of this kind are often called neural networks, as they are intended to simulate (highly idealized) anatomical structures of the human brain.

The alternative paradigm, the one we adopt here, is referred to as logic-based AI. In his 1959 paper *Programs with Common Sense* [47], John McCarthy provided a first, albeit rather vague proposal towards a computer program that represent its knowledge and goals in terms of logical formulas, and that automatically deduces from its knowledge ways to achieve its goals. Roughly speaking, all subsequent work in the field of logic-based AI can be seen as an attempt to refine and implement this proposal.

1.2 Problem of Generality

The logic-based approach has received wide attention in the AI community. Two important virtues of logic that account for its popularity are its clarity, and the pre-existence of many technical mathematical results about logic. But there are some major obstacles to overcome.

The most significant problem, notoriously acknowledged and articulated by McCarthy himself [49], is that of *generality*. Every piece of knowledge seems to hold only within a particular domain. A rule which appears perfectly general in one situation, is very often – if not always – violated in others. Any statement is true only in a certain context. With a little effort, a more general context can usually be depicted in which the precise form of the statement does not hold anymore. Initial efforts in logic-based AI were directed to formalizing well-defined specific domains. Although this resulted in some successful applications, the underlying systems were too inflexible to function well outside the domains for which they were designed.

To overcome this lack of generality, two solutions have been proposed and are currently under diligent development. One is the formalization of *default* or *non-monotonic* reasoning [15]. These are forms of reasoning, which allow for the revision of so-called default assumptions on the base of newly acquired knowledge. The standard example is that of $Tweety$ being a bird ($Bird(Tweety)$), and the rule that birds are able to fly ($\forall x.Bird(x) \supset CanFly(x)$). From this we may conclude that $Tweety$ can fly. Now if we are told that $Tweety$ is a penguin, and thus forms an exception to our rule, we naturally retract our conclusion. This form of reasoning is non-monotonic in the sense that new information *reduces* the knowledge we obtained so far (in the classical logic-based approach new information always extends already established knowledge). Various frameworks for describing non-monotonic forms of reasoning have been investigated [41, 48, 57].

The other proposed solution, the one we are concerned with here, is the formalization of *contextual* reasoning. McCarthy [49] conjectured that the combination of non-monotonic and contextual reasoning mechanisms will constitute an adequate solution to the problem of generality.

1.3 Principle of Locality

Another significant argument for the formalization of contextual reasoning can be found in the work of Fausto Giunchiglia [33]. He emphasized what is called 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. Indeed, to reason about a given goal, people never consider all they know, but rather a very restricted subset of their complete knowledge. It may well be that this mechanism of restricting their attention to a specific context, a particular part of their knowledge, largely accounts for the efficiency with which people are able to reason.

1.4 Characterization of Contexts

What is a context exactly? What are the dimensions along which representations of contextual knowledge may vary?

In a recent publication entitled *Contextual Reasoning Distilled* [6], the many available answers to this question are unified into three fundamental properties.

Partiality A context is partial – it describes only a subset of a more comprehensive state of affairs. This idea is illustrated in figure 1.1. The lower circle represents a state of affairs: "the world", or the whole of expressible knowledge. The circles above depict partial representations of this state of affairs. As figure 1.1 suggests, there may be various kinds of relationships between different partial representations (such as overlap or inclusion).

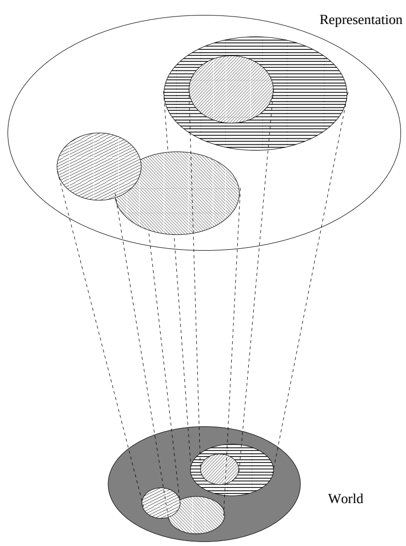


Figure 1.1: Partiality.

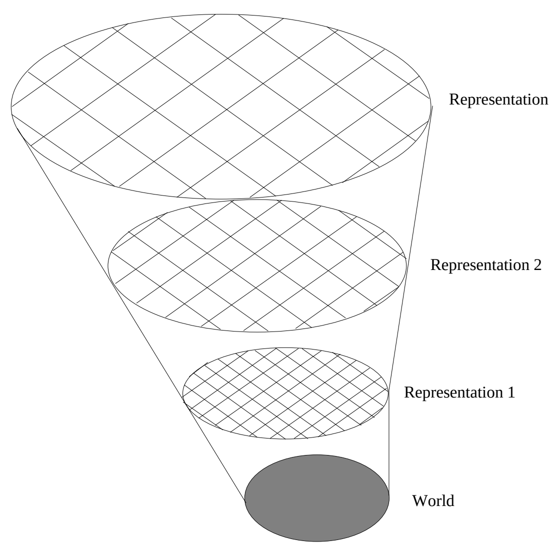


Figure 1.2: Approximation.

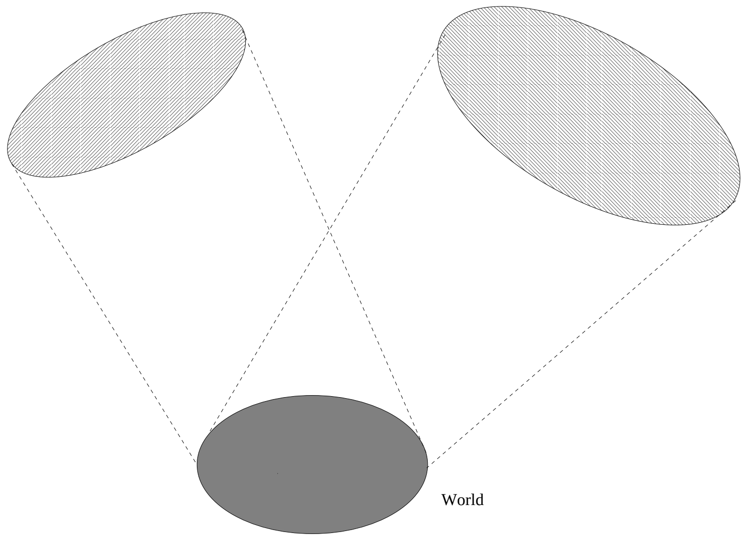


Figure 1.3: Perspective.

Approximation A context is approximate – it is, to a variable extent, an abstraction of the world it describes. This intuition is depicted in figure 1.2. The upper circles correspond to possible representations of the world, at different levels of approximation.

Perspective A context is perspectival – it reflects a mental point of view. A given state of affairs can in general be thought of from several independent perspectives, as illustrated in figure 1.3.

1.5 Formalizations of Contexts

Contextual knowledge representation has been formalized in several ways. Akman and Surav [1] provide a comprehensive overview of the early work on formalizing context, from which two main paradigms have ensued: the propositional logic of context (PLC) developed by McCarthy, Buvač and Mason [17, 50], and the multi-context systems (MCS) devised by Giunchiglia and

Serafini [35], which later became associated with the local model semantics (LMS) introduced by Giunchiglia and Ghidini [30]. MCS has been argued to be most adequate with respect to the properties mentioned above [6], and moreover, has been shown to be technically more general than PLC [61]. Therefore, the lion's share of our analysis will regard MCS-based reasoning, the formal preliminaries of which are extensively revisited in chapter 2. Our main results, however, will be shown to be equally well applicable to PLC.

1.6 Contexts in Practice

Contextual reasoning mechanisms play a significant role in the development of next generation Artificial Intelligence applications. Its importance was first recognized by Lenat and Guha [37, 44], who explicitly designed their notorious CYC common sense knowledge base as a collection of interrelated partial "microtheories". While in CYC, however, the notion of local microtheories was a choice, in contemporary settings the notion of local, distributed knowledge has become a definite must. Maybe the most representative, and surely the currently most widely discussed example of such settings is that of the *semantic web* [3]. Originally envisioned by Tim Berners-Lee (the "father" of the internet), James Hendler, and Ora Lassila [8], the semantic web is an effort to go beyond a purely syntactical annotation of world wide web content (as is presently achieved with HTML). Its objective is to provide for the *semantic* annotation of online documents, in such a way that computers can really understand their contents, and are able to provide much more sophisticated services than they are at present. Such semantic annotation would make reference to online *ontologies*, formal descriptions of particular domains, which would be publishable and editable by anyone willing to do so. As a result, these ontologies will be highly scattered and heterogeneous. Recent work has indeed been focused on the development of languages which allow for the expression of *contextualized* ontologies [11, 13]. A central reasoning system, though, will not be able to deal with such ontologies. This engenders a high demand for distributed, contextual reasoning procedures.

Another research endeavour that has recently received ample attention, and which explicitly seeks to fulfill this demand for distributed reasoning systems is that of *grid computing* [26]. This paradigm has fostered the development of various reasoning systems [18, 27], which demonstrate that implementing logical reasoners as cooperative systems of autonomous local reasoners can indeed improve performance.

Contexts have also been successfully applied to various other fields of AI, including but not limited to meta-reasoning [35], reasoning with viewpoints [4], common sense reasoning [12], reasoning about beliefs [5, 25, 29, 32, 36], multi-agent systems [7, 20], and modeling dialog, argumentation and information integration in electronic commerce [56].

1.7 Outline

The rest of this thesis is organized as follows. First, in chapter 2 we review MCS and explicate the formal notion of contextual reasoning. Then, in chapter 3, we seek to characterize the *inherent difficulty* of contextual reasoning. Our central results are an equivalence theorem with bounded model logic (section 3.1), and the so-called bounded model property for multi-context systems (section 3.2). In section 3.3 these results are applied to obtain new complexity results for McCarthy's propositional logic of context.

In chapter 4, we propose and analyse two conceptually orthogonal ways to *automate* contextual reasoning. First, in section 4.1, we exploit the bounded model property so as to encode contextual reasoning into purely propositional reasoning. This encoding paves the way for the implementation of contextual reasoners based on – and benefiting from the efficiency of – already existing propositional reasoners. Complementarily to this translation-based approach, in section 4.2, we propose a distributed algorithm, which maximally exploits the potential amenity of localizing reasoning and restricting it to relevant contexts only. We show how this algorithm could be implemented by directly reusing off-the-shelf efficient reasoning procedures. Moreover, we argue that the latter approach is, in general, computationally superior to the procedure based on translation into propositional logic.

All this is succeeded by a discussion of some related work in chapter 5. We conclude in chapter 6 with a concise recapitulation of our achievements, and some pointers to future research avenues.

Chapter 2

Multi-Context Systems

In this chapter we shortly revise the multi-context system (MCS) formalism, as first introduced by Giunchiglia [33], and further developed by Giunchiglia and Serafini [35] and Ghidini and Giunchiglia [30].

2.1 Intuition

A simple illustration of the intuitions underlying MCS is provided by the so-called "magic box" example:

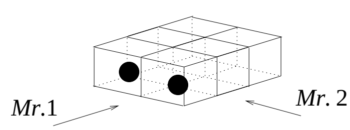


Figure 2.1: A magic box.

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 representation of what they belief to be true about the box. Mr.1's beliefs may regard concepts that are completely meaningless for Mr.2, and* *vice versa. For example, Mr.2 could belief the central section of the box to contain a ball. From Mr.1's viewpoint however, the box doesn't have a central section, so any statement about whether it contains a ball or not is meaningless for him. Mr.1 and Mr.2 may also have concepts in common, but in any case their respective interpretations of those concepts are independent. For example, "the right section of the box contains a ball" is a meaningful statement for both Mr.1 and Mr.2. But it is perfectly conceivably that Mr.1 believes the right section of the box to be empty, while Mr.2 believes it to contain a ball, and vice versa. The bottom line is that both observers have their own* local language *in which they express their beliefs.*

Another important notion is that the observers may have (partial) access to each other's beliefs about the box. For example, Mr.1 may have access to the fact that Mr.2 believes the box to contain a ball. Mr.1 may interpret this fact in terms of his own language, and adapt his beliefs accordingly. We think of this mechanism as an information flow *among different observers.*

In the following we show how collections of local representations and the information flow between them can be captured formally.

2.2 Syntax

Our point of departure is a set of indices I. Each index $i \in I$ denotes a *context*, which is associated with a formal language L_i. Here, for the sake of simplicity, we take each L_i to be a propositional language, but in principle we are not restricted to doing so (the use of first-order languages and description logic has been investigated in [31] and [11], respectively). To state that a propositional formula φ in the language 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*). To model information flow from one or more contexts to another we employ so-called *bridge rules*, which are expressions of the form:

$$\underbrace{i_1 : \varphi_1, \ldots, i_n : \varphi_n}_{premises} \rightarrow \underbrace{i : \varphi}_{consequence} \tag{2.1}$$

Intuitively, bridge rule (2.1) states that the beliefs $\phi_1, \ldots, \phi_n$, holding in contexts $i_1, \ldots, i_n$, respectively, are accessible from - and impose a new belief φ on context i. In this light, the arrow symbol "$\rightarrow$" does not denote material implication here (we will use "$\supset$" for this purpose), but rather a form of "epistemic implication": the propagation of *beliefs* rather than that of facts.

We call $i : \varphi$ the *consequence* and $i_1 : \phi_1, \ldots, i_n : \phi_n$ the *premises* of bridge rule (2.1). 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) *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 thesis, we assume I to be (at most) countable and $\mathbb{BR}$ to be finite. The latter assumption does not apply to MCS with *schematic* bridge rules, such as provability systems and unbounded multi-agent belief systems [35]. The question whether our results may be generalized to capture these cases as well is subject to further investigation.

Also note that the language of a multi-context system does not include expressions like $\neg(i : \varphi)$ and $(i : \varphi \wedge j : \psi)$, which, in some cases, could impose a considerable restriction on the expressiveness of the formalism.

Example 2 *The situation described in example 1 may be formalized by a multi-context system with two contexts* 1 *and* 2*, described by the propositional* *languages* $L_1 = L(\{l, r\})$ *and* $L_2 = L(\{l, c, r\})$*, respectively*[1]*. Intuitively, the atomic propositions* l*,* c*, and* r *correspond to the existence of a ball in the left, center, and right section of the box, respectively, as depicted in figure 2.2. Note that, although* L_1 *and* L_2 *share the atomic propositions* l *and* r*, they are*

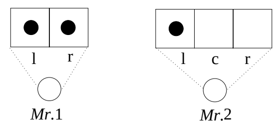


Figure 2.2: Views of Mr.1 and Mr.2.

intended as strictly local and semantically disjunct languages. Interpretations *of* l *and* r *in context* 1 *and* 2 *are in principle entirely independent.*

The information flow described in example 1 (if Mr.2 believes the box to contain a ball, then Mr.1 adopts this belief), is captured by the bridge rule:

$$2 : l \vee c \vee r \rightarrow 1 : l \vee r \tag{2.2}$$

2.3 Semantics

Let M_i be the class of classical interpretations of L_i. Each interpretation $m \in M_i$ is called a *local model* of L_i. In the propositional case, a local model may be denoted by the set of propositional atoms that the model satisfies. Interpretations of entire MCSs are called *chains*. They are constructed from sets of local models.

Definition 2 (Chain) *A chain* $c = \{c_i\}_{i \in I}$ *over a set of indices* I *is a collection of sets of local models (*$c_i \subseteq M_i$ *for all* $i \in I$*). A chain component* c_i *is inconsistent if* $|c_i| = 0$*, point-wise if* $|c_i| = 1$*, and set-wise if* $|c_i| \geq 2$*.*

[1]For a set of atomic propositions P, $L(P)$ is the propositional language defined over P.

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 local model amounts to L_i being interpretable in more than one unique way. Thus, set-wise chain components correspond to partial beliefs (several interpretations are considered possible), whereas point-wise chain components indicate complete beliefs (only one interpretation is conceivable).

Example 3 *Consider the situation depicted in Figure 2.1. Both agents have complete beliefs, corresponding to the following point-wise chain components:*

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

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, respectively:

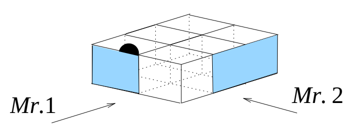


Figure 2.3: The partially hidden magic box

Now, both Mr.1 and Mr.2 have only partial beliefs; their observations may be interpreted in several ways. This is reflected by set-wise chain components:

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

Intuitively, a chain c *satisfies* a formula $i : \varphi$ if φ holds in all interpretations of context i that c considers possible. If c satisfies all the premises of a bridge rule br, then, to *comply with* the information flow specified by br, c should also satisfy br's consequence. More formally we define the following.

Definition 3 (Satisfiability and Compliance) *Let $MS = \langle \{L_i\}_{i \in I}, \mathbb{BR} \rangle$ be a propositional multi-context system over a set of indices I, and let Φ be a set of labeled formulas whose labels constitute a subset J of I.*

1. *A chain c satisfies a labeled formula* $i : \varphi$ *(denoted* $c \models i : \varphi$*) if each local model* $m \in c_i$ *classically satisfies* φ *(denoted* $m \models \varphi$*).*

2. *A chain c satisfies* Φ *(denoted* $c \models \Phi$*) if it satisfies every* $i : \varphi \in \Phi$*.*

3. *A chain c complies with a bridge rule br, if, whenever it satisfies all of br's premises, it also satisfies br's consequence (that is, c either satisfies br's consequence or does not satisfy at least one of br's premises).*

4. *A chain c complies with* $\mathbb{BR}$ *if it complies with every* $br \in \mathbb{BR}$*.*

5. *A chain c is* j*-consistent if* c_j *is nonempty.*

6. *A chain c is* J*-consistent if it is* j*-consistent for all* $j \in J$*.*

7. *We say that* Φ *is consistently satisfiable in* MS *if there is a* J*-consistent chain, which satisfies* Φ *and complies with* $\mathbb{BR}$*.*

The *contextual satisfiability problem* is to determine whether or not a set of labeled formulas Φ is consistently satisfiable in a multi-context system MS. Many contextual reasoning tasks can be expressed in terms of the contextual satisfiability problem. Therefore, as is rather usual in the study of automated reasoning, we will focus our further inquiry on this specific problem.

Intuitively, to solve a contextual satisfiability problem is to construct a chain (by adding and removing local models) that satisfies Φ and complies with $\mathbb{BR}$, but in doing so remains J-consistent. In chapter 4 we will specify a sound and complete algorithm that implements this intuition. For now, we confine ourselves to illustrating it by means of the following example.

Example 4 *Consider example 3, in which both Mr.1 and Mr.2 have only partial views of the magic box, and information flow is modeled by bridge rule 2.2. A contextual satisfiability problem would be to determine whether the set of formulas* $\Phi = \{1 : \neg r, 2 : l\}$ *were consistently satisfiable, that is,* *whether there exists a chain, whose components are both non-empty, which satisfies* Φ *and complies with bridge rule 2.2.*

In order to do so, we may begin with a chain c that contains all possible local models: this indicates that both observers consider all potential interpretations of the situation possible. Then, we gradually restrict the chain so as to meet the given requirements. To satisfy $1 : \neg r$ *(Mr.1 believes the right sector of the box to be empty), we should remove all those local models from* c_1 *that satisfy* r*. This leaves us with* $c_1 = \{\{l, \neg r\}, \{\neg l, \neg r\}\}$*. Similarly, to satisfy* $2 : l$ *(Mr.2 believes that there is a ball in the leftmost sector of the box), we should remove from* c_2 *all those local models that falsify* l*, which yields* $\{\{l, \neg c, \neg r\}, \{l, \neg c, r\}, \{l, c, \neg r\}, \{l, c, r\}\}$*. But now c satisfies the premise of bridge rule 2.2 (*$2 : l \vee c \vee r$*). In order to make it comply with this bridge rule, we should further restrict* c_1*, by removing from it those local models that do not satisfy the consequence of bridge rule 2.2 (*$1 : l \vee r$*). We obtain the chain:*

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

which is $\{1, 2\}$*-consistent, satisfies* Φ*, and complies with bridge rule 2.2. Notice that, as an effect of the information flow modeled by the system, the ultimate chain conveys that Mr.1 believes that the left sector of the box contains a ball, even if this is not directly entailed by his own observations.*

Chapter 3

Complexity Analysis

In this chapter, we seek to characterize the inherent difficulty of settling a contextual satisfiability problem. The difficulty of solving computational problems is usually expressed by the amount of time and/or memory space that is required to do so, as a function of the input problem size. A range of so-called complexity classes has been identified and intensively studied. Each complexity class comprises problems that are essentially just as hard to solve. One class that plays a central role in our analysis, and in theoretical computer science in general, is called NP. It contains those problems that can be settled by *non-deterministic* computations (computations which are not fixed from beginning to end, but which - at various stages - may involve a random choice between a finite number of alternative proceedings) in a number of timesteps that is bounded by some polynomial function of the size of the input problem.

In addition to complexity class membership, a problem can be further characterized as being *hard* and/or *complete* with respect to a complexity class. A problem P is hard with respect to some complexity class C (e.g., NP-hard) if every problem in C can be efficiently reduced to P. That is, an algorithm for solving any problem P^* in C can be easily obtained from an algorithm for settling P. A problem is complete with respect to a complexity class C if it is both contained by C and C-hard.

An important result due to Cook [21], which we will use in our discussion, establishes that propositional satisfiability (hereafter SAT) is NP-complete. This means that any problem in NP is reducible to SAT, and that any problem in NP to which SAT can be effectively reduced is NP-complete. For a more comprehensive review of computational complexity, we refer to [39].

In the following we will consistently 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.

3.1 Encoding Into Modal Satisfiability

A first insight regarding the complexity of contextual satisfiability may be obtained by investigating its encoding into modal satisfiability. After briefly reviewing the basic multi-modal logic K_n, we will show that any contextual satisfiability problem may be reduced to that of satisfying some formula in K_n, whose depth is equal to one. This problem is known to be in NP [43, 38].

K_n typically describes a state of affairs in terms of a nonempty finite set Φ of propositional atoms, as seen through the eyes of n individual agents, named $1, \ldots, n$. Its language is defined to be the least set of formulas containing Φ, closed under negation, conjunction, and the modal operators $\Box_1, \ldots, \Box_n$. A modal operator $\Box_i$ is often intuitively read as: "agent i believes that …", or "for agent i it is necessarily true that …". The *depth* of a modal formula φ is the maximum number of nested modal operators in φ.

Standard semantics for modal logic is called *possible world semantics* [42]. Interpretations in this framework are called *kripke models*. These are tuples $M = \langle W, \pi, \mathcal{R}_1, \ldots, \mathcal{R}_n \rangle$, in which W is a set of *states* or *possible worlds*, π is a truth value assignment to the atomic propositions in Φ for each state

$w \in W$, and every $\mathcal{R}_i$ is a binary *accessibility relation* on W. Intuitively, $(w_1, w_2) \in \mathcal{R}_i$ if in world w_1 of W, agent i considers w_2 a possible world.

Satisfaction of a formula φ in a world w of a kripke model M, denoted as $M, w \models \varphi$, is defined as in standard propositional logic, with the addition of just one extra clause for modal operators:

$$M, w \models \Box_i \varphi \text{ iff } (M, v) \models \varphi \text{ for all } v \text{ such that } (w, v) \in \mathcal{R}_i$$

The intuition behind this clause is that agent i "believes" φ in world w exactly if φ holds in every world v that i considers possible from w. In general, checking satisfiability of a formula φ in K_n is a rather difficult computational task (PSPACE-complete [38]). However, if one imposes the depth of φ to be bounded, the task becomes significantly more straightforward - it has in fact been proven to fall down right into the class of NP-complete problems [43, 38].

This outline of modal syntax and semantics serves our current purposes. However, modal logics have been extensively discussed in the literature, and the insight they provide into the workings of multi-context systems goes beyond our present discussion. Excellent points of departure are [10, 24].

In order to reduce contextual satisfiability to satisfiability in K_n, we first rename all atomic propositions that are shared by two or more local languages (we could, for example, label them with the context that they are associated with), so as to impose syntactic disjunctness on those languages. Then, we define the following translation $(.)^*$ of labeled formulas into modal formulas:

$$(i : \varphi)^* = \Box_i \varphi$$

For bridge rules we have:

$$(i_1 : \varphi_1, \ldots, i_n : \varphi_n \rightarrow i : \varphi)^* \equiv$$
$$(i_1 : \varphi_1)^* \wedge \ldots \wedge (i_n : \varphi_n)^* \supset (i : \varphi)^*$$

And a j-consistency constraint is captured by:

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

We may now state the following equivalence result:

Theorem 1 *There exists a kripke model* $M = \langle W, \pi, \mathcal{R}_1, \ldots, \mathcal{R}_n \rangle$ *such that* $M, w_0 \models \psi$ *for some* $w_0 \in W$ *and:*

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

iff there is a J*-consistent chain* c^M *that satisfies* Φ *and complies with* $\mathbb{BR}$. *Please notice that, in any case, the depth of* ψ *is equal to (bounded by) one.*

Proof. ($\Rightarrow$) We demonstrate how to construct c^M from M. Let m_w be the interpretation of $\bigcup_{i \in I} L_i$ associated to a world $w \in W$; for any $i \in I$, let $m_w|i$ be the restriction of m_w to L_i and let $c_i^M = \{m_w|i \;\mid w_0 R_i w\}$.

As $M, w_0 \models \Box_i \phi$, we have that $w \models \phi$ for any w with $w_0 R_i w$. Moreover, as $\phi \in L_i$, we have that $m_w|i \models \phi$. This implies that $c^M \models i : \phi$. Bridge rule compliance and J-consistency are established likewise.

($\Leftarrow$) ¿From c^M we may obtain a suitable kripke model M. Let W consist of a world w_0 plus one world w_{m_i} for each local model m_i of every component c_i^M of c^M. Let every $w_{m_i} \in W \setminus \{w_0\}$ evaluate L_i according to m_i, and assign $True$ to the rest of $\bigcup_{i \in I} L_i$. Let w_0 evaluate every atomic proposition to $True$. For all $i \in I$, let:

$$R_i = \{\langle w_0, w_{m_i} \rangle \mid w_{m_i} \text{ corresponds to } m_i \in c_i^M\}$$

The resulting model is schematically depicted in Figure 3.1. One can easily verify that $M, w_0 \models \psi$. □

Contextual satisfiability subsumes SAT and is therefore NP-hard [21]. Theorem 1, plus the fact that satisfiability for bounded K_n is in NP [38], imply that contextual satisfiability is also in NP, and therefore NP-complete.

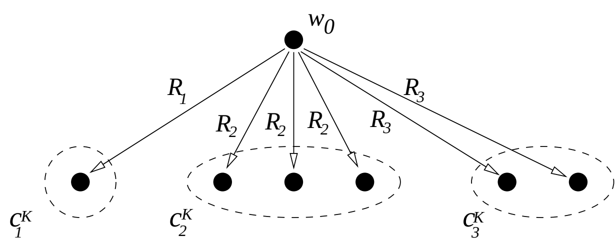


Figure 3.1: A schematic kripke model for ψ.

Moreover, the syntax of the formula that results from our translation is highly constrained: it is a conjunction of disjunctions of (negated) boxed formulas. Each disjunction comprises at most one boxed formula that is not negated, and furthermore, each boxed formula is purely propositional. This form strongly alludes to the existence of relatively efficient ways to solve the contextual satisfiability problem. Therefore, in pursuit of a more nuanced understanding of its complexity, we proceed with a firsthand analysis of its semantical properties.

3.2 Bounded Model Property

In this section we establish the bounded model property for propositional multi-context systems. This property implies that, to consistently satisfy a set of formulas Φ in a multi-context system MS, it should suffice to construct a chain, which consists of at most $|J| + |\mathbb{BR}|$ local models. This result significantly restrains the amount of time required for non-deterministically settling contextual satisfiability. Namely, in order to do so, it is sufficient to "guess" a chain with only $|J| + |\mathbb{BR}|$ local models, and then to check whether this chain consistently satisfies Φ in MS.

Let us first introduce some notation and terminology. The size of a labeled formula $i : \varphi$ is denoted by $|i : \varphi|$. Let $P(i : \varphi)$ and $P(\Phi)$ be the set of propositional atoms appearing in a formula $i : \varphi$ or a set of formulas Φ. Let

G_i be the number of local models contained by the i^{th} component of a chain, and let G be the total number of local models comprising that chain. Let $\Xi(br)$ and $\Xi(\mathbb{BR})$ consist of the premises and the consequence(s) of a bridge rule br or a set of bridge rules $\mathbb{BR}$. Finally, let N be the total size of the formulas in Φ and $\Xi(\mathbb{BR})$:

$$N = \sum_{i:\varphi\in\Phi} |i : \varphi| + \sum_{i:\xi\in\Xi(\mathbb{BR})} |i : \xi|$$

Theorem 2 (Bounded Model Property) *A set of formulas Φ is consistently satisfiable in a multi-context system MS if and only if there exists a J-consistent chain that contains at most $|J| + |\mathbb{BR}|$ local models and satisfies Φ in compliance with $\mathbb{BR}$.*

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 by some local model m_{br} in c. On the other hand, for every $j \in J$, there must be at least one local model $m_j \in c_j$ that satisfies all those formulas in Φ that apply to context j. The chain c^* obtained from c by eliminating all local models except for:

$$\bigcup_{j\in J} m_j \cup \bigcup_{br\in\mathbb{BR}^*} m_{br}$$

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

We use this result to prove contextual satisfiability to be NP-complete and to establish a refined upper bound for the amount of time it requires. In order to do so we need the following lemma:

Lemma 1 *Model checking, that is, the problem of determining whether a given chain c consistently satisfies a set of formulas Φ in a multi-context* *system MS, can be performed deterministically in time:*

$$O(\sum_{i:\varphi\in\Phi\cup\Xi(\mathbb{BR})} G_i \times |\varphi|)$$

Proof. Model checking can be split into three sub-tasks:

1. Checking whether c satisfies Φ;
2. Checking whether c complies with $\mathbb{BR}$;
3. Checking whether c is J-consistent.

First consider sub-task 1. Checking whether a particular formula $i : \varphi \in \Phi$ is satisfied by c can be done as follows. Let $\varphi_1, \ldots, \varphi_k$ be an ordering of the subformulas of φ, such that $\varphi_k = \varphi$ and if φ_i is a subformula of φ_j, then $i < j$. Since φ has at most $|\varphi|$ subformulas, we have $k < |\varphi|$. By induction on k' we can label each local model m in c_i with either φ_j or $\neg\varphi_j$, for $j = 1, \ldots, k'$, depending on whether or not $m \models \varphi_j$, in time $O(G_i \times k')$. As a result, checking whether c satisfies Φ can be carried out in time $O(\sum_{i:\varphi\in\Phi} G_i \times |\varphi|)$. Sub-task 2 takes time $O(\sum_{i:\xi\in\Xi(\mathbb{BR})} G_i \times |\xi|)$, as in the worst case it involves checking whether all the consequences and premises of every bridge rule in $\mathbb{BR}$ are satisfied or not. Sub-task 3 merely consists in checking whether c_j is nonempty, for $j \in J$. This can be done in $O(|J|)$ timesteps. The result follows directly. □

Theorem 3 *Contextual satisfiability is NP-complete and can be settled in non-deterministic time:*

$$O((|J| + |\mathbb{BR}|) \times N)$$

Proof. We have already observed that contextual satisfiability is NP-hard. To determine satisfiability we first non-deterministically appoint a set *Cons* of bridge rule consequences, and a set *Prem* of bridge rule premises, such that for every $br \in \mathbb{BR}$, either br's consequence is in $Cons$, or one of br's premises is in $Prem$. Let J, I_{Cons}, and I_{Prem} be the set of contexts involved by Φ, $Cons$, and $Prem$, respectively. Furthermore, let Φ_i, $Cons_i$, and $Prem_i$ be the set of i-formulas contained by Φ, $Cons$, and $Prem$, respectively. Now we construct a chain c, such that:

- For all $i \in I_{Prem}$, c_i contains $|Prem_i|$ local models;
- For all $i \in J/I_{Prem}$, c_i contains exactly one local model;
- For all $i \notin J \cup I_{Prem}$, c_i is empty;
- For all $i \in I$, each $m \in c_i$ evaluates the propositional atoms *not* appearing in $\Phi_i \cup Cons_i \cup Prem_i$ to $True$.

The only "guessing" involved in constructing c, apart from the choice of $Cons$ and $Prem$, are the truth values to which each local model in c_i should evaluate the propositional atoms in $P(\Phi_i \cup Cons_i \cup Prem_i)$. Notice that c contains at most $|J|+|Prem| \leq |J|+|\mathbb{BR}|$ local models, which are distributed over those components c_i of c with $i \in J \cup I_{Prem}$; all the other components of c are empty. Consider a local model m contained in c_i for some $i \in J \cup I_{Prem}$. The number of atomic propositions $|P(\Phi_i \cup Cons_i \cup Prem_i)|$ that m should "explicitly" evaluate is clearly bounded by N. We must appoint at most $|J| + |\mathbb{BR}|$ such explicit valuations (one for each local model in c), so c can be constructed in non-deterministic time $O((|J| + |\mathbb{BR}|) \times N)$.

It remains to be checked whether c consistently satisfies Φ. By lemma 1 this requires deterministic time $O((|J|+|\mathbb{BR}|) \times N)$. Theorem 2 assures that, if Φ is consistently satisfiable in MS, then guessing a chain as described above is bound to result in a suitable one. Thus, consistent satisfiability of Φ in MS can be computed in non-deterministic time $O((|J| + |\mathbb{BR}|) \times N)$. □

3.3 Application to PLC

The results presented above may be applied to improve current complexity bounds for the propositional logic of context (PLC) as described in [17, 50]. Before doing so we briefly review the most important notions of this formalism. In PLC, contexts are represented by *sequences* of labels. Intuitively, a label sequence $\kappa_1\kappa_2$ denotes a context κ_2 *as seen from the viewpoint of context* κ_1. If $\mathbb{K}$ is a set of labels and $\mathbb{K}^*$ the set of finite sequences over $\mathbb{K}$, then the language of PLC is defined as a multi-modal language over a set of atomic propositions $\mathbb{P}$, with modal operators $\mathit{ist}(\overline{\kappa}, \varphi)$ for each label sequence $\overline{\kappa} = \kappa_1 \ldots \kappa_n \in \mathbb{K}^*$. The intuitive meaning of a formula $\mathit{ist}(\kappa_2, \varphi)$, when stated in context κ_1, is that φ holds in κ_2, from the standpoint of κ_1.

A model $\mathfrak{M}$ for PLC associates to each context $\overline{\kappa}$ a set of partial truth value assignments $\mathfrak{M}(\overline{\kappa})$. Associating a *set* of assignments to every context is motivated by intuitions similar to those which underlie possible worlds and local model semantics. A formula φ holds ("is believed to be true") in context $\overline{\kappa}$ if it is satisfied by all the assignments associated to $\overline{\kappa}$.

Partial assignments provide for the simulation of local languages – in each context, only a fragment of the global language is actually meaningful. Formally, a formula φ is meaningful in context $\overline{\kappa}$ if every assignment in $\mathfrak{M}(\overline{\kappa})$ fully determines the truth of φ. So $\mathfrak{M}$ defines a function $\mathsf{Vocab}(\mathfrak{M})$, which associates to every context $\overline{\kappa}$ a set $\mathsf{Vocab}(\mathfrak{M})(\overline{\kappa})$ of meaningful formulas.

Now for a model $\mathfrak{M}$, a context $\overline{\kappa}$, an assignment $\nu \in \mathfrak{M}(\overline{\kappa})$, and a formula $\varphi \in \mathsf{Vocab}(\mathfrak{M})(\overline{\kappa})$, satisfaction is defined as follows:

1. $\mathfrak{M}, \nu \models_{\overline{\kappa}} p$ iff $\nu(p) = \text{true}$
2. $\mathfrak{M}, \nu \models_{\overline{\kappa}} \neg\varphi$ iff not $\mathfrak{M}, \nu \models_{\overline{\kappa}} \varphi$
3. $\mathfrak{M}, \nu \models_{\overline{\kappa}} \varphi \supset \psi$ iff not $\mathfrak{M}, \nu \models_{\overline{\kappa}} \varphi$ or $\mathfrak{M}, \nu \models_{\overline{\kappa}} \psi$
4. $\mathfrak{M}, \nu \models_{\overline{\kappa}} \mathit{ist}(\kappa, \varphi)$ iff for all $\nu' \in \mathfrak{M}(\overline{\kappa}\kappa)$, $\mathfrak{M}, \nu' \models_{\overline{\kappa}\kappa} \varphi$

5. $\mathfrak{M} \models_{\overline{\kappa}} \varphi$ iff for all $\nu \in \mathfrak{M}(\overline{\kappa})$, $\mathfrak{M}, \nu \models_{\overline{\kappa}} \varphi$

If the precondition $\varphi \in \mathsf{Vocab}(\mathfrak{M})(\overline{\kappa})$ does not hold, then neither $\mathfrak{M}, \nu \models_{\overline{\kappa}} \phi$ nor $\mathfrak{M}, \nu \models_{\overline{\kappa}} \neg\phi$. A formula φ is satisfiable in a context $\overline{\kappa}$ if there is a model $\mathfrak{M}$ such that $\mathfrak{M} \models_{\overline{\kappa}} \varphi$.

The best complexity result so far for determining satisfiability in PLC is due to by Massacci [46]. He proposed a tableaux-based decision procedure, which settles satisfiability of a formula φ in non-deterministic time $O(|\varphi|^4)$.

Our approach is to translate a PLC formula φ into a labeled formula $\epsilon : \varphi$ and a multi-context system MCS(φ), so that φ is satisfiable in PLC if and only if $\epsilon : \varphi$ is consistently satisfiable in MCS(φ). Subsequently, we will demonstrate that determining whether or not $\epsilon : \varphi$ is consistently satisfiable in MCS(φ) takes non-deterministic time $O(|\varphi|^2)$.

The translation works as follows.Let MCS(φ) contain a context labeled with the sequence $k_1 \ldots k_n$, for each nesting pattern $ist(k_1, \ldots ist(k_n, \psi) \ldots)$ in φ. Let the language of context $k_1 \ldots k_n$ contain all the atomic propositions in ψ, plus 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[1]:

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

Example 5 *Consider the following PLC formula:*

$$\varphi = p \vee ist(k, q \supset (ist(h, r \wedge s) \supset ist(j, q)))$$

[1]The first two bridge rules correspond to McCarthy's notions of *entering* and *exiting* contexts [50], while the last two bridge rules correspond to the Δ axiom introduced by Buvač and Mason [17]. The specified bridge rules are schematic: the first one, for example, is instantiated for every subformula ψ of any formula θ, which occurs in a nesting pattern $ist(k_1, \ldots ist(k_n, \ldots ist(k, \theta) \ldots) \ldots)$ in φ.

MCS(φ) consists of four contexts which are labeled ϵ (the empty sequence), k, kh, and kj, respectively. The language L_ϵ of context ϵ contains two atomic propositions, p and $ist(k, q \supset (ist(h, r \vee s) \supset ist(j, q)))$. L_k contains q and $ist(h, r \wedge s)$; L_{kh} comprises r and s, and finally, L_{kj} is constructed over q. The bridge rules of MCS(φ) are as stated above.

Theorem 4 ([61]) *A formula φ is satisfiable in PLC if and only if $\epsilon : \varphi$ is consistently satisfiable in MCS(φ).*

Theorem 5 *Satisfiability of a formula φ in PLC can be computed in non-deterministic polynomial time $O(|\varphi|^2)$.*

Proof. By theorem 4 any satisfiability problem in PLC can be transformed into an equivalent satisfiability problem in MCS. 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 $|\mathbb{BR}|$ in MCS(φ) is bounded by $4 \times |\varphi|$. Moreover, by construction, the sum of the lengths of the formulas involved in any bridge rule of MCS(φ) is at most four.

By theorem 3, contextual satisfiability of $\epsilon : \varphi$ in MCS(φ) can be settled in non-deterministic time:

$$O((|\Phi| + |\mathbb{BR}|) \times (\sum_{i:\varphi \in \Phi} |i : \varphi| + \sum_{br \in \mathbb{BR}} \sum_{i:\xi \in \Xi(br)} |i : \xi|))$$

In the light of the above observations, and keeping in mind that Φ merely consists of $\epsilon : \varphi$, we may rewrite this in terms of φ as $O(|\varphi|^2)$. □

Chapter 4

Decision Procedures

We now turn to decision procedures. We describe and compare two fundamentally different approaches to automatically deciding contextual satisfiability. The first, described in section 4.1, is based on the observation that contextual satisfiability must be tractably reducible to purely propositional SAT (by NP-membership of the former and NP-completeness of the latter). In providing such a reduction, we may loose the particular structure of a given problem. However, in doing so we do lay the groundwork for the implementation of purely SAT-based contextual reasoners, which could benefit from the well-advanced techniques developed by the SAT community.

The second approach, discussed in section 4.2, seeks not only to maintain, but moreover to *exploit* the structure of contextual satisfiability problems. This approach leads to the specification of the CONTEXTSAT algorithm, in which reasoning is strictly local and restricted to relevant contexts only. Section 4.2.2 proposes a sufficient condition for the algorithm to be complete, and compares its efficiency to that of the previously mentioned translation based procedure. At last, in section 4.2.3, the use of several efficient propositional reasoning platforms for the implementation of CONTEXTSAT is discussed.

4.1 Encoding Into Propositional SAT

In this section we encode contextual satisfiability problems into propositional logic, and discuss the complexity of the resulting problem in terms of the dimension of the underlying multi-context system.

4.1.1 Encoding

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

Example 6 *Consider an MCS with two contexts* 1 *and* 2*, which are described by* $L(\{p\})$ *and* $L(\{q\})$*, respectively, and subject to the following bridge rules:*

$$1 : p \quad \rightarrow \quad 2 : q \tag{4.1}$$

$$1 : \neg p \quad \rightarrow \quad 2 : q \tag{4.2}$$

The formula $2 : \neg q$ *is consistently 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 exploit the understanding obtained in section 3.2 while establishing the bounded model property. More specifically, if we assume $|\mathbb{BR}| \geq 1$, theorem 2 can be slightly weakened to the statement that Φ is consistently satisfiable iff it is consistently satisfied by a chain c^* all of whose components are either empty or contain exactly $|\mathbb{BR}|$ local models. Our approach is to construct a propositional formula ψ, which is satisfiable if and only if such a chain c^* exists.

We express this formula in a language which contains a propositional atom p_i^k for every $p \in L_i$, and for every $k = 1, \ldots, |\mathbb{BR}|$. Intuitively, the truth value assigned to p_i^k by a propositional valuation of ψ corresponds to the truth value assigned to p by the k^{th} local model in c_i^*. The language also contains an atomic proposition e_i for each index $i \in I$. Intuitively, e_i being assigned $True$ corresponds to c_i^* being empty.

We write $K = \{1, \ldots, |\mathbb{BR}|\}$. For any formula φ, every $i \in I$ and $k \in K$, let φ_i^k denote the formula that results from substituting every propositional atom p in φ with p_i^k. Furthermore, let $\varphi_i^K = \bigwedge_{k \in K} \varphi_i^k$. Then, the translation of a labeled formula reads:

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

For bridge rules we have:

$$(i_1 : \varphi_1, \ldots, i_n : \varphi_n \rightarrow i : \phi)^* \equiv$$
$$(i_1 : \phi_1)^* \wedge \ldots \wedge (i_n : \phi_n)^* \supset (i : \phi)^*$$

And a j-consistency constraint is captured by:

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

Theorem 6 *There exists a truth value assignment V to the propositional atoms $\{p_i^k \mid i \in I \text{ and } k = 1, \ldots, |\mathbb{BR}|\} \cup \{e_i \mid i \in I\}$, which satisfies:*

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

iff there is a J-consistent chain c^V that satisfies Φ in compliance with $\mathbb{BR}$.

Proof. ($\Rightarrow$) From V we construct a chain c^V, such that each component c_i^V is empty if $V(e_i) = True$ and consists of exactly $|\mathbb{BR}|$ local models otherwise. In the latter case, the k^{th} local model of c_i^V is made to evaluate each propositional atom $p \in L_i$ to $True$ iff $V(p_i^k) = True$. It's easy to see that c^V is J-consistent and satisfies Φ in compliance with $\mathbb{BR}$.

($\Leftarrow$) The above observations imply that, if Φ is consistently satisfiable, there exists a J-consistent chain c^* each of whose components is either empty or contains exactly $|\mathbb{BR}|$ local models, and which satisfies Φ in compliance with $\mathbb{BR}$. ¿From c^* we may obtain a truth value assignment V by proceeding as follows. To a propositional atom e_i, let V assign $True$ iff $c_i^* = \emptyset$. To a propositional atom p_i^k, let V assign $True$ if the k^{th} local model of c_i^* satisfies p, $False$ if the k^{th} local model of c_i^* satisfies $\neg p$, and any truth value if c_i^* is empty. It should be straightforward to see that V satisfies ψ. □

4.1.2 Complexity

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. If P_i denotes the set of propositions that is used to describe context i, then $|P|$ amounts to:

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

Using this translation, contextual satisfiability problems may be solved by means of existing specialized Sat solvers. As an alternative to this approach, the following section describes an algorithm which tackles contextual satisfiability in an entirely distributed fashion.

4.2 A Distributed Algorithm

In this section we describe an algorithm for contextual satisfiability, in which reasoning is always executed locally, and restricted to contexts which are actually relevant for the problem under consideration. This algorithm is shown to be more efficient than the translation based method presented above.

4.2.1 Algorithm

Our approach is the following. Starting with some initial chain c^0, we attempt to construct a sequence $c^0, c^1, \ldots$ such that for all $m \in \{1, 2, \ldots\}$:

- c^m satisfies Φ.
- c^m *extends* c^{m-1}, that is, for every $i \in I$, $c_i^m \subseteq c_i^{m-1}$.
- c^{m+1} complies with the bridge rules that c^m does not 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": its components c_i^0 must contain the entire set of local models M_i. As such c^0 doesn't satisfy any formula. Particularly, c^0 does not satisfy any bridge rule premise, and thus 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^n, for any $n \in \{2, 3, \ldots\}$. 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 hence complies with br), any of its extensions also satisfies br's consequence and thus comply with br.

This approach is implemented by the ContextSat procedure specified in algorithm 1. It takes as its input a set of labeled formulas Φ (for simplicity of the algorithm, we assume that Φ contains at most one formula per context), 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, ContextSat is called with c being a chain over I, whose components all comprise 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.

Algorithm 1 A distributed algorithm for contextual satisfiability.

```
ContextSat(Φ, 𝔹ℝ, I, J, c)
begin
I_Φ := {i ∈ I | i : φ_i ∈ Φ};
for all i ∈ I_Φ do
   c*_i := Extend(c_i, φ_i);
end for
for all i ∈ I/I* do
   c*_i := c_i;
end for
for all j ∈ J do
   if c*_j = ∅ then
      return False;
   end if
end for
𝔹ℝ* := {br ∈ 𝔹ℝ | c* ⊨ i : η for all i : η ∈ prem(br)}
if 𝔹ℝ* = ∅ then
   return c*;
end if
Ψ* := {cons(br) | br ∈ 𝔹ℝ*};
Φ* := { i : φ | φ = ⋀_{i:ξ∈Ψ*} ξ , i ∈ I };
return ContextSat(Φ*, 𝔹ℝ/𝔹ℝ*, I, J, c*);
end
```

Extensions are always constructed *locally*. That is, ContextSat first determines the set I_Φ of contexts involved by formulas in Φ, and then, for every $i \in I^*$, calls a sub-procedure 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 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 ContextSat is addressed to extend c so as to satisfy Φ^*. Recursively proceeding like this, a chain is constructed that satisfies Φ at any stage, and at some point either becomes J-inconsistent, or compliant with $\mathbb{BR}$.

4.2.2 Completeness and Locality

The set of bridge rule consequences that is satisfied by c is strictly expanded by every recursive call to ContextSat. Since the number of bridge rules is finite, ContextSat is bound to terminate. Soundness is evident; in order to assure completeness we should enforce Extend(c_i, φ) to remove from c_i *exactly* those local models that do not satisfy φ. We say that Extend(c_i, φ) should yield a *complete extension of* c_i w.r.t. φ. Notice that this 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 pursuit of *locality* constitutes an important amenity of our contextual approach w.r.t. centralized procedures.

4.2.3 Towards Efficient Implementations

Several efficient off-the-shelf reasoning platforms can be used to implement ContextSat, such that Extend indeed yields complete extensions. We sketch two particular ways to go.

BDD-based implementation

Reduced Ordered Binary Decision Diagrams [16] (or simply BDDs) constitute a canonical representation of propositional formulas. 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 [51] to planning [19], safety analysis [14], and diagnosis [63].

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 $False$. Dedicated routines to establish entailment are provided by most BDD packages.

As reasoning is always performed locally, each context can be represented by a 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.

SAT-based implementation

Propositional Sat solvers make up another very effective way to manipulate propositional formulas. The typical approach is a depth-first search for satisfying truth value assignments, "splitting" on boolean variables [22]. During the last decade enormous progress has been achieved in this field: state-of-the-art Sat solvers are able to process problems with tens of thousands variables and a million clauses [53], and are applied in several industrial settings ranging from formal verification [9] to planning [40] and automatic test pattern generation [45].

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.

The sequence of problems presented to the Sat solver is incremental. A consistency / entailment check carried out during the j^{th} iteration of ContextSat is often an extension of a similar problem solved during some previous iteration of the algorithm. In this light, it is recommendable to exploit recent developments in *incremental* Sat technology [23]. Significant computational advances can be achieved by retaining learned conflict clauses when adding new clauses to an already processed formula.

Simulation

Let us describe a simple simulation of a Sat-based implementation of the ContextSat algorithm.

Example 7 *Reconsider example 3. Suppose that the information flow of the system is modeled by bridge rule 2.2:*

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

We let ContextSat *determine whether* $\Phi = \{1 : \neg r, 2 : l\}$ *is consistently satisfiable or not. It proceeds as follows. During the first iteration* φ_1 *and* φ_2 *are assigned* $1 : \neg r$ *and* $2 : l$*, respectively. The consistency check succeeds and* $\mathbb{BR}^*$ *is determined to consist of bridge rule 2.2. During the second iteration* φ_1 *is updated to* $1 : \neg r \wedge (l \vee r)$*. Consistency is again established and* $\mathbb{BR}^*$ *is now empty, so that the algorithm terminates. The chain corresponding to the final state of* φ_1 *and* φ_2 *is:*

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

It indeed conveys that Mr.1 believes the left section of the box to contain a ball (as established earlier in example 4).

4.2.4 Complexity

Of course, it is important to have an idea of how efficient ContextSat is. We therefore analyse its worst-case complexity and compare it to that of the translation-based procedure described in section 4.1. The results of our analysis hold for both Sat-based and bdd-based implementations.

In general, the greater part of ContextSat's 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 ContextSat only *one* "new" bridge rule premise is found to be satisfied by the last modification of the chain under construction, 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 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 outlined in section 4.1 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 ContextSat. Take $|\mathbb{BR}| = 10$ and $Q = 5$, for instance. ContextSat then takes time in the order of 3200, while the translation based approach may require a number of timesteps in the order of 10^{30}.

Chapter 5

Related Work

Work by Giunchiglia and Sebastiani [34] 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 [35]. Resulting procedures have been proven orders of magnitude faster than previous tableau-based decision procedures. In this thesis, we applied a similar approach to the class of multi-context systems, whose structure is not necessarily hierarchical.

In section 3.1 contextual reasoning with finite sets of bridge rules was shown to be reducible to a simple form of reasoning in bounded modal logic. In particular, it was observed that expressing contextual satisfiability problems in K_n yields formulas whose modal depth is, in any case, equal to one.

Fixpoint decision procedures for modal logic have recently been proposed by Pan, Sattler and Vardi [55], and could, in the light of the above observation, 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 from a type/world b, if $\Box\phi \in b$ implies $\phi \in a$.

The top-down algorithm proposed in [55] 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 [54]. 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.

Amir and McIlraith [2] define a propagation algorithm called MP, which computes satisfiability of a theory T that is partitioned into sub-theories (or *partitions*) $T_1, \ldots, T_n$. Two 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 $T_{i\leq n}$, 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. Partitions could be seen as contexts, and overlap between two partitions could be simulated using bridge rules of the form $i : p \rightarrow j : p$ and $i : \neg p \rightarrow j : \neg p$, for each atomic proposition p 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 improvements with respect to MP. First, bridge rules express more complex relationships 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 describe *symmetric* relations between partitions. At last, while MP requires a partial order between contexts, ContextSat naturally deals with any kind of relational structure between them.

Chapter 6

Conclusion

The achievement of a solid paradigm for contextual knowledge representation and contextual reasoning is of paramount importance for the development of sophisticated theory and applications in Artificial Intelligence. Substantial theoretical arguments to support this claim can be found in the work of McCarthy, who pleaded for a formalization of context as a possible solution to the problem of *generality*, and in the work of Giunchiglia, who emphasized the principle of *locality*. Moreover, recent practical research endeavours, most notably those related to the Semantic Web and the Grid, render the need for contextual reasoners even more urgent. Our contribution to fulfilling this need has been twofold.

First, we have investigated the computational complexity of contextual reasoning based on propositional multi-context systems with finite sets of bridge rules. Our main results in this regard are an equivalence theorem with bounded multi-modal K_n, which is well-known to be NP-complete, and the so-called bounded model property for multi-context systems, which provides considerable insight into the complexity of contextual reasoning in general, and has important implications for the complexity of multi-context systems in particular. Our results have also been applied to improve complexity results for the satisfiability problem in McCarthy's propositional logic of context, by showing that the latter problem can be considered a special case of the contextual satisfiability problem in multi-context systems.

Our second contribution is the proposal and analysis of two orthogonal approaches to automatically deciding satisfiability in multi-context systems. By providing a tractable encoding of contextual satisfiability problems into purely propositional ones, a solid groundwork has been laid for SAT-based implementations of contextual reasoning systems. On the other hand, we have also proposed a distributed algorithm, called CONTEXTSAT, which seeks to exploit the potential benefit of localizing reasoning and restricting it to relevant contexts only. CONTEXTSAT has been shown to be more efficient, in general, than our translation based procedure, and to be implementable using off-the-shelf efficient reasoning platforms, such as BDDs and propositional SAT solvers. Moreover, CONTEXTSAT has been designed to suit a possible distributed peer-to-peer implementation. It is *modular*, i.e. global reasoning is made up of local reasoning procedures, and it 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.

Future work will encompass experimentation with both native peer-to-peer and translation based contextual reasoning platforms. Also, we are interested in the extent to which our results may be generalized so as to apply to multi-context systems with schematic bridge rules as well.

Bibliography

[1] V. Akman and M. Surav. Steps toward formalizing context. *AI Magazine*, pages 55–72, 1996.

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

[3] G. Antoniou and F. van Harmelen. *A Semantic Web Primer*. MIT Press, 2004.

[4] G. Attardi and M. Simi. A formalisation of viewpoints. *Fundamenta Informaticae*, 23(2–4):149–174, 1995.

[5] M. Benerecetti, P. Bouquet, and C. Ghidini. Formalizing belief reports – the approach and a case study. In *Artificial Intelligence, Methodology, Systems, and Applications (AIMSA 98)*, pages 62–75, 1998.

[6] M. Benerecetti, P. Bouquet, and C. Ghidini. Contextual reasoning distilled. *Journal of Experimental and Theoretical Artificial Intelligence*, 12(3):279–305, 2000.

[7] M. Benerecetti, F. Giunchiglia, and L. Serafini. Model checking multiagent systems. *Journal of Logic and Computation*, 8(3):401–423, 1998.

[8] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, 284(5):34–43, 2001.

[9] 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)*, pages 317–320, 1999.

[10] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*. Cambridge University Press, 2001.

[11] A. Borgida and L. Serafini. Distributed description logics: Assimilating information from peer sources. *Journal of Data Semantics*, 1:153–184, 2003.

[12] P. Bouquet and F. Giunchiglia. Reasoning about theory adequacy. a new solution to the qualification problem. *Fundamenta Informaticae*, 23(2–4):247–262, 1995.

[13] P. Bouquet, F. Giunchiglia, F. van Harmelen, L. Serafini, and H. Stuckenschmidt. C-OWL: Contextualizing ontologies. In *International Semantic Web Conference*, pages 164–179, 2003.

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

[15] G. Brewka, J. Dix, and K. Konolige. *Nonmonotonic Reasoning - An Overview*. CSLI publications, Stanford, 1997.

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

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

[18] W. Chrabakh and R. Wolski. Gridsat: a chaff-based distributed SAT solver for the grid. In *Supercomputing Conference (SC03)*, 2003.

[19] 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.

[20] A. Cimatti and L. Serafini. Multi-agent reasoning with belief contexts: the approach and a case study. In *Intelligent Agents Conference – Workshop on Agent Theories, Architectures, and Languages*, pages 71–85, 1995.

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

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

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

[24] R. Fagin, J.Y. Halpern, Y. Moses, and M.Y. Vardi. *Reasoning about Knowledge*. MIT Press, 1995.

[25] M. Fisher and C. Ghidini. Programming resource-bounded deliberative agents. In *International Joint Conference on Artificial Intelligence (IJCAI 99)*, pages 200–206, 1999.

[26] I. Foster. *The Grid : Blueprint for a New Computing Infrastructure.* Morgan-Kaufmann, 2^{nd} edition, 2004.

[27] T. Hune G. Behrmann and F.W. Vaandrager. Distributed timed model checking - how the search order matters. In *Computer Aided Design (CAV 00)*, pages 216–231, 2000.

[28] J. Garson. Connectionism. In *The Stanford Encyclopedia of Philosophy*. 2002.

[29] C. Ghidini. Modeling (un)bounded beliefs. In *Modeling and Using Context (CONTEXT 99)*, pages 145–158, 1999.

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

[31] C. Ghidini and L. Serafini. Distributed first order logics. In D. Gabbay and M. de Rijke, editors, *Frontiers Of Combining Systems 2*, 1998.

[32] E. Giunchiglia and F. Giunchiglia. Ideal and real belief about belief. In *International Conference on Formal and Applied Practical Reasoning (FAPR 96)*, pages 261–275, 1996.

[33] F. Giunchiglia. Contextual reasoning. *Epistemologia*, XVI:345–364, 1993.

[34] 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.

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

[36] F. Giunchiglia, L. Serafini, E. Giunchiglia, and M. Frixione. Non-omniscient belief as context-based reasoning. In *International Joint Conference on Artificial Intelligence (IJCAI 93)*, pages 548–554, 1993.

[37] R.V. Guha. *Contexts: A Formalization and some Applications*. PhD thesis, Stanford University, 1991.

[38] J. Halpern and Y. Moses. A guide to completeness and complexity for modal logics of knowledge and belief. *Artificial Intelligence*, 54(3):311–379, 1992.

[39] J. E. Hopcroft, R. Motwani, and J. D. Ullman. *Introduction to Automata Theory, Language, and Computation*. Addison–Wesley, 2nd edition edition, 2001.

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

[41] S. Kraus, D. Lehmann, and M. Magidor. Nonmonotonic reasoning, preferential models and cumulative logics. *Artificial Intelligence*, 44:167–207, 1990.

[42] S. Kripke. A semantical analysis of modal logic. *Zeitschrift fur Mathematische Logik und Grundlagen der Mathematik*, 9:67–96, 1963.

[43] R. Ladner. The computational complexity of provability in systems of modal propositional logic. *SIAM Journal of Computing*, 6:467–480, 1977.

[44] D. B. Lenat and R. V. Guha. *Building Large Knowledge Based Systems*. Addison Wesley, 1990.

[45] J.P. Marques-Silva and K.A. Sakallah. Robust search algorithms for test pattern generation. In *Symposium on Fault-Tolerant Computing*, pages 152–161, 1997.

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

[47] J. McCarthy. Programs with common sense. In *Teddington Conference on the Mechanization of Thought Processes*, pages 75–91, 1959.

[48] J. McCarthy. Circumscription: A form of non-monotonic reasoning. *Artificial Intelligence*, 13:27–39, 1980.

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

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

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

[52] M. Minsky. Logical versus analogical or symbolic versus connectionist or neat versus scruffy. *AI Magazine*, 12(2):34–51, 1991.

[53] 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)*, pages 530–535, 2001.

[54] G. Pan. *BDD-based decision procedures for modal logic K*. Master thesis, Rice University, 2003.

[55] G. Pan, U. Sattler, and M. Vardi. Bdd-based decision procedures for K. In *Conference on Automated Deduction (CADE 00)*, 2002.

[56] S. Parsons, C. Sierra, and N.R. Jennings. Agents that reason and negotiate by arguing. *Journal of Logic and Computation*, 8(3):261–292, 1998.

[57] R. Reiter. A logic for default reasoning. *Artificial Intelligence*, 13:81–132, 1980.

[58] F. Roelofsen. Bounded model property for multi-context systems. In *European Summer School in Logic, Language, and Information - Student Session (ESSLLISS 04)*, 2004. Accepted for publication.

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

[60] F. Roelofsen, L. Serafini, and A. Cimatti. Many hands make light work: Localized satisfiability for multi-context systems. In *European Conference on Artificial Intelligence (ECAI 04)*, 2004. Accepted for pulication.

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

[62] L. Serafini and F. Roelofsen. Satisfiability for propositional contexts. In *Principles of Knowledge Representation and Reasoning (KR 04)*, pages 369–376, 2004.

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

Index