



# Research Repository UCD

Provided by the author(s) and University College Dublin Library in accordance with publisher policies. Please cite the published version when available.

<b>Title</b>	The role of unsatisfiable Boolean constraints in lightweight description logics
<b>Authors(s)</b>	Arif, Muhammad Fareed
<b>Publication date</b>	2016
<b>Publisher</b>	University College Dublin. School of Computer Science
<b>Link to online version</b>	<a href="http://dissertations.umi.com/ucd:10119">http://dissertations.umi.com/ucd:10119</a>
<b>Item record/more information</b>	<a href="http://hdl.handle.net/10197/8595">http://hdl.handle.net/10197/8595</a>

Downloaded 2019-07-26T22:28:56Z

The UCD community has made this article openly available. Please share how this access benefits you. Your story matters! (@ucd\_oa)



Some rights reserved. For more information, please see the item record link above.





# The Role of Unsatisfiable Boolean Constraints in Lightweight Description Logics

Doctoral Thesis

M. Fareed Arif

12250832

The thesis is submitted to University College Dublin in fulfillment of the  
requirements for the degree of Doctor of Philosophy in  
the School of Computer Science and Informatics  
College of Science  
University College Dublin (UCD)

Head of School: Professor Dr. Pádraig Cunningham  
Principal Supervisor: Professor Dr. Joao Marques-Silva  
Co-Supervisor: Professor Dr. Alexey Lastovetsky

August 2016

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Preliminaries</b>	<b>8</b>
2.1	Satisfiability Theory (SAT) . . . . .	8
2.2	Lightweight Description Logics . . . . .	15
2.3	Hypergraph Theory . . . . .	27
<b>3</b>	<b>Axiom pinpointing via Enumeration of MCSes/MUSes</b>	<b>31</b>
3.1	MUSes and MCSes Enumeration . . . . .	32
3.1.1	Minimal Hitting Set Enumeration . . . . .	32
3.1.2	Explicit Enumeration . . . . .	34
3.1.3	Implicit Enumeration . . . . .	38
3.2	Axiom pinpointing in Lightweight DL $\mathcal{EL}^+$ . . . . .	41
3.2.1	(EL2MCS) MCS Enumeration using HST . . . . .	42
3.2.2	(HgMUS) group MUS Enumeration of Horn Formula .	44
3.2.3	(BEACON) Lightweight DL Debugger . . . . .	45
3.2.4	Complexity Results . . . . .	46

## CONTENTS

3.2.5	Experimental Assessment . . . . .	48
<b>4</b>	<b>KI '15 Publication</b>	<b>53</b>
<b>5</b>	<b>SAT '15 Publication</b>	<b>63</b>
<b>6</b>	<b>SAT '16 Publication</b>	<b>84</b>
<b>7</b>	<b>JAIR '16 Paper</b>	<b>96</b>
<b>8</b>	<b>Conclusions and Future Work</b>	<b>131</b>
	<b>Bibliography</b>	<b>133</b>

# Abstract

Lightweight Description Logics (e.g.  $\mathcal{EL}$ ,  $\mathcal{EL}^+$  etc.) are commonly used languages to represent life science ontologies. In such languages ontology classification – the problem of computing all the subsumption relations – is tractable and used to characterize all the classes and properties in any given ontology. Despite the fact that classification is tractable in  $\mathcal{EL}^+$ , axiom pinpointing – the problem of computing the reasons of an (unintended) subsumption relation – is still worst-case exponential. This thesis proposes state-of-the art SAT-based axiom pinpointing methods for the Lightweight Description Logic  $\mathcal{EL}^+$ . These axiom pinpointing methods emanate from the analysis of minimal unsatisfiable boolean constraints using hitting set dualization that is also related to Reiter’s model-based diagnosis. Its consequences are significant both in terms of algorithms (i.e., MUS extraction and enumeration methods) as well as understanding axiom pinpointing in Lightweight Description Logics through the prism of minimal unsatisfiable boolean constraints and its related problems (i.e., hypergraph transversal).

# Chapter 1

## Introduction

Description Logics (DLs) are a family of knowledge representation formalisms that model the relationships between well formed concepts and roles in a domain of interest [BCM<sup>+</sup>10]. In such a domain, concepts are set of individuals or individual names in a class and roles define properties or relationship between these individuals. In Web Ontology Language (OWL) [GHM<sup>+</sup>08] classes represent concepts and roles are known as properties. OWL is a standard theoretically founded on DLs which provide empirical reasoning services for ontology languages (i.e., OWL Lite, OWL DL and OWL FULL [PSHH04]) of variable expressivity that have a range of applications in areas of medicine and bioinformatics [GZB06b, SAR<sup>+</sup>07, SCS11], agriculture [SLL<sup>+</sup>04], astronomy [DRPM06], geography [Goo05], military and defense [Sma07, LAF<sup>+</sup>05], among others.

Lightweight Description Logics (i.e.,  $\mathcal{EL}$ ,  $\mathcal{ELH}$  and  $\mathcal{EL}^+$ ) (see [Section 2.2](#))

## CHAPTER 1. INTRODUCTION

are tractable fragments of a well-known description logic  $\mathcal{ALC}$  (Attributive Concept Language with Complements) [SSS91]. The  $\mathcal{EL}$  family of DLs<sup>1</sup> is distinguished because it has polynomial-time reasoning algorithms [Bra04, BBL05] and it is commonly used to represent life science ontologies including NCI (the National Cancer Institute Thesaurus) [SdCH<sup>+</sup>07], GENE [ABB<sup>+</sup>00], FMA (Foundational Model of Anatomy) [RM03, GZB06a], SNOMED CT (Systematized Nomenclature of Medicine – Clinical Terms) [SCC97] and GALEN [RH97]. These medical ontologies gather, access, share, and classify biomedical knowledge and data.

Ontology classification – the problem of computing all subsumption relations – is an important reasoning service that creates a hierarchy of classes (taxonomic hierarchy) that are either subsuming or being subsumed by other classes (or concepts) [BCM<sup>+</sup>10]. Thus, any property that is necessarily true for a subsuming class (or concept) is true for all subsumed classes (or concepts). Rather than merely deciding a subsumption relation among concepts in an ontology; sometimes, it is imperative to find the concise explanations for a consequence [BS08] (i.e., why a subsumption relation holds) and compute repairs (i.e., remove consequence), if necessary. This problem is circumscribed under the topic of axiom pinpointing that is an extensively studied problem [BH95, SC03, MLB06, BLS06, SHCvH07, BP10b, BPS07, KPHS07, BS08, HPS09, MMV11, SV09, HPS09, PS09, PS10b, PS10a, Sun13, ZQS13, SV15, AMMS15a, AMMS15b, MPR16, AMI<sup>+</sup>16] and has wide range of applications

---

<sup>1</sup>Lightweight DLs and  $\mathcal{EL}$  family of DLs refer to the same term and are used interchangeably.

## CHAPTER 1. INTRODUCTION

including ontology debugging [PSK05, PSG06, SHCvH07], explaining logical differences [KLW12], error-tolerant reasoning [LP14b, LP14a] and context based reasoning [BKP12, Peñ15].

For the  $\mathcal{EL}$  family of Description Logics axiom pinpointing defines an important step in debugging (error-prone)  $\mathcal{EL}^+$  ontologies [BPS07, BS08]. It computes one, many or all set-wise minimal set of axioms (MinAs) which explain some (unintended) inference e.g. a subsumption relation. For  $\mathcal{EL}^+$  ontologies, EL<sup>+</sup>SAT [SV09, SV15] is a SAT-based axiom pinpointing approach that encodes the complete inference or classification (which can be viewed as a directed acyclic graph) into a polynomial size Horn formula and enumerates over MinAs using AllSAT-based methods [LNO06a]. Instead, we can make use of more effective MUS extraction and enumeration algorithms [LS08, LS05, LS08, LML<sup>+</sup>09, BMS11, MSL11, BMS12, BLMS12, PM13, LPMMS15]) (see [Section 3.1](#)) for axiom pinpointing in the  $\mathcal{EL}$  family of Description Logics and its related problems [BKP12, LP14b]. Based on these techniques, we have implemented the following state-of-the art SAT-based axiom pinpointing and debugging tools for the Lightweight DL  $\mathcal{EL}^+$ :

- EL2MCS [AMMS15a] is an MCS-enumerator complemented with hitting set duality algorithms for axiom pinpointing in Lightweight DLs (see details [Subsection 3.2.1](#)).
- EL2MUS [AMMS15b], front-end to HGMUS [AMMS15b], is an efficient tool to enumerate MUSes of Horn formulae for axiom pinpointing in Lightweight DLs. HGMUS is a state-of-the-art group-MUS enumerator

## CHAPTER 1. INTRODUCTION

(see details [Subsection 3.2.2](#)).

- BEACON [[AMI<sup>+</sup>16](#)] is an efficient Debugging tool for  $\mathcal{EL}^+$  ontologies that assimilates HGMUS and FORQES [[IPLM15](#)]. FORQES is a state-of-the-art solver for computing the cardinality-minimum MUS (SMUS) (see details [Subsection 3.2.3](#)).

These tools are related and emanate from the analysis of minimal unsatisfiable boolean constraints using hitting set dualization (see [Section 3.2](#)). We have compared these tools with EL<sup>+</sup>SAT [[SV09](#), [SV15](#)], SATPin [[MP15](#)] and other alternatives, such as CEL [[BLS06](#)], JUST [[Lud14](#)] and report categorical performance gains in [Chapter 4](#), [Chapter 5](#), [Chapter 6](#) and [Chapter 7](#). We have also conducted a detailed analysis of existing SAT based axiom pinpointing methods, particularly EL<sup>+</sup>SAT and SATPin, and explain the reasons for their inefficiencies (see [Chapter 5](#)).

From a complexity perspective, the  $\mathcal{EL}$  family of DLs guarantee to yield polynomial complexity of reasoning but axiom pinpointing is an NP-hard problem [[PS09](#), [PS10b](#), [PS10a](#)] and in the worst case, there may exist exponential many minimal subsets (w.r.t. set inclusion) for a consequence [[BPS07](#)]. Even deciding the cardinality-minimum MinA is  $\Delta_2^P$ -complete [[Bie08](#)] and the complexity of enumerating all MinAs is as hard as enumerating the set of minimal transversals<sup>2</sup> of a hypergraph [[PS09](#)], a well studied open problem [[EG95a](#)] (see [Section 2.3](#)). Moreover, hypergraph transversal and its related problems can yield a tight complexity bound for axiom pinpointing in the  $\mathcal{EL}$  family

---

<sup>2</sup>A transversal of a hypergraph is a subset of vertex set which has a non empty intersection with every hyperedge in the graph.

## CHAPTER 1. INTRODUCTION

of DLs because these well-studied problems are closely related with the existence of an output polynomial algorithm<sup>3</sup> for enumeration problems which defines a parameter to measure the time complexity of enumeration algorithms (see [Subsection 3.2.4](#)).

### Thesis Overview

The outline of this report is as follows:

- [Chapter 2](#) defines the basic and related definitions along with relevant decision problems (i.e., minimal unsatisfiability, classification and axiom pinpointing etc.) both in Boolean Satisfiability Theory (SAT) (see [Section 2.1](#)) and Lightweight Description Logic  $\mathcal{EL}^+$  (see [Section 2.2](#)), respectively. This chapter also includes a detailed discussion on the well-known hypergraph transversal problem in relation to decision and functional problems in Lightweight DLs and SAT (see [Section 2.3](#)). We have also provided relevant examples in each section of this chapter.
- [Chapter 3](#) explains the well-known Reiter's model based diagnosis detailed with explicit and implicit minimal hitting set enumeration techniques (see [Section 3.1](#)). Building on this concrete foundation, we propose to solve the problem of axiom pinpointing in the Lightweight Description Logic  $\mathcal{EL}^+$  using unsatisfiable boolean constraints (see [Section 3.2](#)). We conclude this chapter by providing the complexity results

---

<sup>3</sup>An output polynomial algorithm is such that whose running time is bounded by a polynomial depending on the size of inputs as well as outputs.

## CHAPTER 1. INTRODUCTION

of decision and enumeration problems both in SAT and Lightweight DLs and the experimental evaluation of our proposed methods in comparision with existing state of the art axiom pinpointing methods for  $\mathcal{EL}^+$ .

- [Chapter 4](#), [Chapter 5](#), [Chapter 6](#) and [Chapter 7](#) enlist the peer reviewed publications and the contributions the author has made towards solving the problem of axiom pinpointing in the Lightweight Description Logic  $\mathcal{EL}^+$  using SAT-based methods and techniques, explained in [Chapter 3](#).
- [Chapter 8](#) concludes the discussion with a brief summary and some future directions of research.

# Chapter 2

## Preliminaries

This chapter provides an overview of the Boolean Satisfiability Problem (SAT) ([Section 2.1](#)) and the Lightweight Description Logic  $\mathcal{EL}^+$  ([Section 2.2](#)).

### 2.1 Satisfiability Theory (SAT)

The Boolean Satisfiability Problem (SAT), well-known to be NP-complete [[Coo71](#), [Lev73](#)], is an important decision problem that decides (yes or no answer) if there exists an assignment to the variables of a Boolean formula that makes it true. Despite its worst-case exponential behavior, well-engineered Conflict-Driven Clause Learning (CDCL) [[MSLM09](#)] SAT solvers efficiently compute the satisfying assignments for many classes of propositional formulae. Modern CDCL solvers, besides using DPLL [[DLL62](#)], speedup overall problem solving time using several key techniques [[BHvMW09](#)] including clause learning [[MSS96](#), [MSS99](#)], branching heuristics [[MMZ<sup>+</sup>01](#)], lazy data structures

## CHAPTER 2. PRELIMINARIES

using watched literals [MMZ<sup>+</sup>01], search restarts [GSK98, BS00b], clause deletion strategies [GN07], among others. Efficient SAT encoding techniques [ES06, Ms08, CES<sup>+</sup>09, Pre09] and off-the-shelf SAT solvers provide a valuable tool for solving a range of problems. Example of applications include software and hardware verification [BCC<sup>+</sup>99, BCCZ99], interpolation and SAT-based model checking [McM03], automatic test pattern generation (ATPG) [LMS06a], automated planning [KS92, NGT04], bioinformatics [LMS06b], software package dependencies [DCDL<sup>+</sup>06], and cryptography [Bar08]. These problems are either formulated as simple decision problems (i.e., Is a given boolean formula  $\varphi$  satisfiable?) or search problems (i.e., Find a satisfying assignment to  $\varphi$  if it exists else output unsatisfiable) or involve in solving a complex function problem (i.e., compute a prime implicant of  $\varphi$ ) defined over Boolean formulae that call a SAT solver as an oracle to determine the satisfiability. Example of function problems include computing prime implicants or implicants, minimal unsatisfiable subsets of clauses, minimal correction subsets of clauses, minimal and maximal models, etc. These function problems can also be reduced to the problem of computing a minimal set for a given monotone predicate [BM07, MSJ13].

### Syntax of Propositional Formulas

We use standard propositional logic definitions [BL99, BHvMW09]. A Boolean formula  $\mathcal{F}$  is defined inductively over set of propositional variables  $X \triangleq \{x_1, \dots, x_n\}$  using standard logical connectives  $\neg, \vee, \wedge$  that can be extended to include  $\rightarrow$ , and  $\leftrightarrow$ . For example,  $\neg\mathcal{F}$  and  $\mathcal{G}_1 \circ \mathcal{G}_2$  where  $\circ \in \{\vee, \wedge, \rightarrow, \leftrightarrow\}$  are well-formed

## CHAPTER 2. PRELIMINARIES

formulas in propositional logic. Propositional formulae also include conjunctive and disjunctive normal forms (CNF and DNF). A CNF formula  $\mathcal{F}$  is a conjunction of clauses ( $c_1 \wedge \dots \wedge c_m$ ). Each clause  $c$  is a non-tautologous disjunction of literals ( $l_1 \vee \dots \vee l_k$ ) where a literal  $l$  is either a boolean variable  $x \in X$  or its negation  $\neg x$ . A clause is called a Horn clause if it contains at most one positive literal. Conversion of any propositional formula  $\varphi$  into a satisfiability preserving CNF  $\mathcal{F}$  can be done using well-known Tseitin encodings [Tse83, PG86], or transformation techniques [Vel05] that also handle unobservable variables. A CNF formula can also be viewed as set of clauses and a clause is a set of literals. In a CNF formula, a Horn formula is special case that is only composed of Horn clauses. There are number of classes of CNF (i.e., Horn CNF, 2CNF etc.) that are decidable in polynomial time [DG84, IM87, Min88].

### Semantics of Propositional Formulas

A truth assignment, or an interpretation, is a mapping  $\mu : X \rightarrow \{0, 1\}$  that assign truth values (0 or 1) to the variables of  $\mathcal{F}$  and is also called an interpretation. A propositional formula  $\mathcal{F}$  is evaluated using an interpretation and standard truth-functional connectives ( $\neg, \vee, \wedge, \rightarrow, \leftrightarrow$ ) that adheres to the

## CHAPTER 2. PRELIMINARIES

following set theoretic semantics [BHvMW09]:

$$\mu(\mathcal{F}) = \begin{cases} 1 - \mu(\mathcal{F}) & \text{if } \mathcal{F} = \mu(\neg\mathcal{F}) \\ \min\{\mu(\mathcal{G}_1), \mu(\mathcal{G}_2)\} & \text{if } \mathcal{F} = \mu(\mathcal{G}_1 \vee \mathcal{G}_2) \\ \max\{\mu(\mathcal{G}_1), \mu(\mathcal{G}_2)\} & \text{if } \mathcal{F} = \mu(\mathcal{G}_1 \wedge \mathcal{G}_2) \\ \max\{1 - \mu(\mathcal{G}_1), \mu(\mathcal{G}_2)\} & \text{if } \mathcal{F} = \mu(\mathcal{G}_1 \rightarrow \mathcal{G}_2) \\ 1 - |\mu(\mathcal{G}_1), \mu(\mathcal{G}_2))| & \text{if } \mathcal{F} = \mu(\mathcal{G}_1 \leftrightarrow \mathcal{G}_2) \end{cases}$$

In set theory, the min and max are the least and greatest elements based on standard ordering  $\leq$  of the natural numbers in the set, respectively. An interpretation  $\mu$  satisfies a clause  $c$  if it contains at least one of its literals. Given a formula  $\mathcal{F}$ ,  $\mu$  satisfies  $\mathcal{F}$  (written  $\mu \models \mathcal{F}$ ) if it satisfies all its clauses (i.e.,  $\mu(\mathcal{F}) = 1$ ) and it is referred as a *model* of  $\mathcal{F}$ . A propositional formula  $\mathcal{F}$  is either *satisfiable* ( $\mathcal{F} \not\models \perp^1$ ) or *unsatisfiable* ( $\mathcal{F} \models \perp$ ).

### Maximum Satisfiability Problem

The maximum satisfiability problem (MaxSAT) is a generalization of the satisfiability problem. In MaxSAT, a CNF formula  $\mathcal{F}$  is a tuple  $\langle \varphi_{\mathcal{H}}, \varphi_{\mathcal{S}} \rangle$ , where  $\varphi_{\mathcal{H}}$  denotes the hard clauses (must be satisfied) and  $\varphi_{\mathcal{S}}$  denotes the soft clauses (can be relaxed or dropped). Moreover, the weight function  $\omega : \varphi_{\mathcal{H}} \cup \varphi_{\mathcal{S}} \rightarrow \mathbb{R} \cup \top$  assigns a value to every clause in  $\mathcal{F}$ . Such an  $\mathcal{F}$  defines a partial MaxSAT instance that is an optimization problem in SAT. It requires hard clauses in  $\mathcal{F}$  to be satisfied, whereas soft clauses can be discarded and the weight function summed up to define the cost for each satisfying model.

---

<sup>1</sup>  $\perp$  denotes False or 0. Similarly,  $\top$  stand for True or 1.

## CHAPTER 2. PRELIMINARIES

A SAT solver accepts a propositional formula  $\mathcal{F}$ , CNF-encoding in the DIMACS file format [JT96], and returns a tuple  $\langle \text{st}, \mu \rangle = \text{SAT}(\mathcal{F})$ . If  $\mathcal{F}$  is satisfiable (i.e.,  $\text{st} = 1$ ) the truth assignment  $\mu$  is a model or witness of satisfiability for  $\mathcal{F}$ . If  $\mathcal{F}$  is unsatisfiable (i.e.,  $\text{st} = 0$ ) then it may contain unsatisfiable cores  $\mathcal{U} \subseteq \mathcal{F}$ . For any satisfiable formula  $\mathcal{F}$  (i.e.,  $\mathcal{F} \not\models \perp$ ) we make use of maximal models. These models are set-wise maximal subset of the variables in  $\mathcal{F}$  that are assigned a true value (i.e.,  $\mu(x) = 1$  where  $x \in \mathcal{F}$ ). If  $\mu$  and  $\mu'$  are models of  $\mathcal{F}$ , we say  $\mu \leq \mu'$  if  $\mu(x) = 1$  implies  $\mu'(x) = 0$ . A model  $\mu$  is maximal if no other model  $\mu'$  such that  $\mu < \mu'$  and  $\mu$  is minimal if for any other model  $\mu'$ , we have  $\mu' < \mu$ . Moreover, any two models  $\mu$  and  $\mu'$  are incomparable if neither  $\mu \leq \mu'$  nor  $\mu' \leq \mu$ .

Let  $\mathcal{F}$  is satisfiable propositional formula,  $\mu$  is a model of  $\mathcal{F}$  and  $P \subseteq X$  is a set of variables appearing in  $\mu$  with positive polarity. The variables that does not appear in  $P$  are assumed to be falsified.

**Definition 1.** Given  $\mathcal{F} \cup P \not\models \perp$ , the (satisfying) truth assignment  $\mu$  is a maximal model ( $MxM$ ) of  $\mathcal{F}$  iff  $\forall x \in X \setminus P$ ,  $\mathcal{F} \cup P \cup \{x\} \models \perp$ .

We can also obtain a minimal model for an unsatisfiable formula  $\mathcal{F}$  on similar notes.

### Unsatisfiable Boolean Constraints

An over-constrained system [MBB<sup>+</sup>03] is represented by an unsatisfiable formula  $\mathcal{F}$  (i.e.,  $\mathcal{F} \models \perp$ ) that may contain more than one unsatisfiable core  $\mathcal{U} \subseteq \mathcal{F}$  and comprises of clauses that contribute to the *unsatisfiability* of  $\mathcal{F}$ . A

## CHAPTER 2. PRELIMINARIES

minimal subset of clauses  $\mathcal{M}$  from  $\mathcal{U}$  that preserves its unsatisfiability can be seen as a minimal set of reasons of unsatisfiability of  $\mathcal{F}$ .

**Definition 2.** *Given  $\mathcal{F} \models \perp$ ,  $\mathcal{M} \subseteq \mathcal{F}$  is a Minimal Unsatisfiable Subformula (MUS) of  $\mathcal{F}$  iff  $\mathcal{M}$  is unsatisfiable and  $\forall_{\mathcal{M}' \subsetneq \mathcal{M}} \mathcal{M}'$  is satisfiable.*

A minimal correction subset represents a minimal set of clauses that are removed to restore *satisfiability* of  $\mathcal{F}$ .

**Definition 3.** *Let  $\mathcal{F} \models \perp$ ,  $\mathcal{C} \subseteq \mathcal{F}$  is a Minimal Correction Subformula (MCS) of  $\mathcal{F}$  iff  $\mathcal{F} \setminus \mathcal{C}$  is satisfiable and  $\forall_{\mathcal{C}' \subsetneq \mathcal{C}} \mathcal{F} \setminus \mathcal{C}'$  is unsatisfiable.*

An *Maximal Satisfiable Subset* (MSS) is the complement of an MCS [BL03, LS08].

**Definition 4** (MSS). *Let  $\mathcal{F} \models \perp$ , the subset  $\mathcal{S} \subseteq \mathcal{F}$  is a Maximal Satisfiable Subset (MSS) iff  $\forall c \subseteq \mathcal{F} \setminus \mathcal{S} \wedge c \neq \emptyset$  such that  $\mathcal{S} \cup c$  is unsatisfiable.*

The analysis of over-constrained systems [MBB<sup>+</sup>03, OPFP07] finds many relevant applications in type error debugging [dBSW03, BS05], inconsistency measurement [HK10, XM12], debugging of relational specifications [TCJ08, TVD10], over-constrained temporal problems [LMPS05], and software hardware model checking [ALS08]. Motivated by these applications, MUSes and related concepts have been extended to CNF formulae where clauses are partitioned into disjoint sets called groups [LS08].

**Definition 5. (*Group-Oriented MUS*).** *Given an explicitly partitioned unsatisfiable CNF formula  $\mathcal{F} = \mathcal{G}_0 \cup \dots \cup \mathcal{G}_k$ , a group-oriented MUS (or group-*

## CHAPTER 2. PRELIMINARIES

MUSes	$\{\mathcal{G}_1\}$	$\mathcal{B}_1 = (\neg p_1)$	$\mu_1 = \{p_1, p_2, p_3, p_4\}$
	$\{\mathcal{G}_2, \mathcal{G}_3, \mathcal{G}_4\}$	$\mathcal{B}_2 = (\neg p_2 \vee \neg p_3 \vee \neg p_4)$	$\mu_2 = \{p_2, p_3, p_4\}$
MCSes	$\{\mathcal{G}_1, \mathcal{G}_2\}$	$\mathcal{B}_3 = (p_1 \vee p_2)$	$\mu_3 = \{p_3, p_4\}$
	$\{\mathcal{G}_1, \mathcal{G}_4\}$	$\mathcal{B}_4 = (p_1 \vee p_4)$	$\mu_4 = \{p_2, p_3\}$
	$\{\mathcal{G}_1, \mathcal{G}_3\}$	$\mathcal{B}_5 = (p_1 \vee p_3)$	$\mu_5 = \{p_2, p_4\}$

Table 2.1: MUSes and MCSes of  $\mathcal{F}$  using EMUS/MARCO [PM13, LPMMS15]

*MUS*) of  $\mathcal{F}$  is a set of groups  $\mathcal{G} \subseteq \{\mathcal{G}_1, \dots, \mathcal{G}_k\}$ , such that  $\mathcal{G}_0 \cup \mathcal{G}$  is unsatisfiable, and for every  $\mathcal{G}_i \in \mathcal{G}$ ,  $\mathcal{G}_0 \cup (\mathcal{G} \setminus \mathcal{G}_i)$  is satisfiable.

The core group  $\mathcal{G}_0$  (*group-0*) consists of *background* clauses that are included in every group-MUS and supposed to be satisfiable and can be empty. The related concepts of group-MCS and group-MSS can be defined in the same way. For MaxSAT, the use of groups is investigated in [HMM15].

**Example 2.1.1.** Let  $\mathcal{F}$  be an unsatisfiable formula defines  $\mathcal{G}_0$  as core group (clauses  $c_1, c_2, c_3$  must be satisfied) and  $\mathcal{G}_1, \dots, \mathcal{G}_4$  contains unit soft clauses  $(c_4, \dots, c_7)$  assigned to a selection variable from  $p_1, \dots, p_4$ .

$$\begin{aligned} \mathcal{G}_0 &= \overbrace{\{(\neg a \vee \neg b) \wedge (b)\}}^{c_1} \wedge \overbrace{(\neg c \vee \neg d \vee \neg e)}^{c_2} \\ \mathcal{G}_1 &= \overbrace{\{a\}}^{c_4}, \mathcal{G}_2 = \overbrace{\{c\}}^{c_5}, \mathcal{G}_3 = \overbrace{\{d\}}^{c_6}, \mathcal{G}_4 = \overbrace{\{e\}}^{c_7} \\ \mathcal{F} &= \mathcal{G}_0 \cup \{p_1 \rightarrow \mathcal{G}_1\} \cup \{p_2 \rightarrow \mathcal{G}_2\} \cup \{p_3 \rightarrow \mathcal{G}_3\} \cup \{p_4 \rightarrow \mathcal{G}_4\}. \end{aligned}$$

It is shown in Table 2.1 that the propositional formula  $\mathcal{F}$  has two MUSes  $\mathcal{M}_1 = \{c_4\}$ ,  $\mathcal{M}_2 = \{c_5, c_6, c_7\}$  and three MCSes  $\mathcal{C}_1 = \{c_4, c_5\}$ ,  $\mathcal{C}_2 = \{c_4, c_6\}$ ,

## CHAPTER 2. PRELIMINARIES

$\mathcal{C}_3 = \{c_4, c_7\}$  along with maximal models  $\mu_1, \dots, \mu_5$  and respective blocking clauses  $\mathcal{B}_1, \dots, \mathcal{B}_5$ . The maximal models are used to enumerate over the search space whereas the blocking clauses eliminate the redundant search process (also see [Section 3.1](#)).

### Complexity of Decision Problems in SAT

From a query complexity perspective, deciding an MUS is  $\text{DP}^P$ -Complete problem and extracting an MUS, a function problem, is in  $\text{FP}^{NP}$  with lower bound in  $\text{FP}_{\parallel}^{NP}$  [[CT95](#)]. Moreover, deciding the existence of an MUS of size less than or equal to  $k \in \mathbb{N}$  is  $\Sigma_2^P$ -Complete problem [[Lib05](#), [Gup06](#)] and extracting the smallest MUS (SMUS) is a function problem that is in  $\text{FP}^{\Sigma_2^P}$  [[LS04](#), [LML+09](#), [IPLM15](#)].

## 2.2 Lightweight Description Logics

A Description Logic (DL) can be considered as an extension of propositional logic that consists of concepts (unary relations), roles (binary relations), and individuals (constants) [[BCM<sup>+</sup>10](#)]. We have only considered the Lightweight Description Logic  $\mathcal{EL}^+$  [[Bra04](#), [BBL05](#)] and its reasoning services.

Lightweight Description Logics are tractable with limited expressivity which however provide efficient polynomial-time reasoning services. In particular, the Lightweight DL  $\mathcal{EL}^+$  has been extensively used to build large ontologies, most notably from the biomedical domains [[SCC97](#), [ABB<sup>+</sup>00](#), [GZB06b](#), [RM03](#), [GZB06a](#), [SdCH<sup>+</sup>07](#), [SCS11](#)].

## CHAPTER 2. PRELIMINARIES

### Syntax of $\mathcal{EL}^+$

The standard definitions of Description Logic  $\mathcal{EL}^+$  [BBL05] are considered in which the set of concept names in  $N_C$  represents classes and role names<sup>2</sup> in  $N_R$  represents properties or relationship in a problem domain. An  $\mathcal{EL}^+$  concept description  $C$  is an expression that is inductively built from these disjoint sets  $N_C$  and  $N_R$  using the following BNF rule:

$$C ::= A \mid \top \mid C \sqcap C \mid \exists r.C$$

where  $A \in N_C$  is a concept name, as mentioned, represents a class,  $\top$  represents the top concept i.e., the super class of all the classes,  $C \sqcap C$  represents the intersection of two classes and  $\exists r.C$  represent class of things which are related to some member of the class  $C$  via property  $r \in N_R$ .

**Example 2.2.1.** *For example, in a family domain Person can be used as a concept name to represent persons,  $\exists hasChild.Person$  (i.e., class of things which have some person as their child) is a concept description that defines parents. If concept Male represents only males then  $Male \sqcap \exists hasChild.Person$  represent fatherhood.*

A general concept inclusion (GCI<sup>3</sup>) has the form  $C \sqsubseteq D$  where  $C, D$  are  $\mathcal{EL}^+$  concept descriptions and a role inclusion (RI) axiom is of form  $r_1 \circ \dots \circ r_n \sqsubseteq r$  where  $r_1 \circ \dots \circ r_n$  means composition of the properties  $r_1, \dots, r_n$  belong to  $N_R$ . A domain knowledge is described as a finite set of GCIs and RIs, also

---

<sup>2</sup>a role name can be transitive or reflexive.

<sup>3</sup>GCI defines a subset relation.

## CHAPTER 2. PRELIMINARIES

$\mathcal{EL}^+$ Syntax	$\mathcal{EL}^+$ Semantics
$[\top]^{\mathcal{I}}$	$\Delta^{\mathcal{I}}$
$[C \sqcap D]^{\mathcal{I}}$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
$[\exists r.C]^{\mathcal{I}}$	$\{x \in \Delta^{\mathcal{I}}   y \in \Delta^{\mathcal{I}} : (x, y) \in r^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$

Table 2.2: Standard set theoretic Semantics of  $\mathcal{EL}^+$ .

called a TBox  $\mathcal{T}$ . Primitive concepts of  $\mathcal{T}$ , denoted  $\text{PC}_{\mathcal{T}}$ , are the smallest set of concepts that contains the top concept  $\top$ , and all the concept names occurring in  $\mathcal{T}$  and the set of primitive roles, denoted  $\text{PR}_{\mathcal{T}}$ , represent all role names in  $\mathcal{T}$ . We will often use the term axiom to refer to both GCIs and RIs.

**Normalization:** It can be viewed as the process of breaking complex GCIs into simpler ones. A TBox  $\mathcal{T}$  is normalized, denoted  $\mathcal{T}_{\mathcal{N}}$ , if it only contains axioms of form:  $(A_1 \sqcap \dots \sqcap A_k) \sqsubseteq B$  ( $k \geq 1$ ),  $A \sqsubseteq \exists r.B$ ,  $\exists r.A \sqsubseteq B$ , or  $r_1 \circ \dots \circ r_n \sqsubseteq r$  ( $n \geq 1$ ) where  $A, A_1, \dots, A_k, B \in \mathbf{N}_{\mathcal{C}}$  and  $r, r_1, \dots, r_n \in \mathbf{N}_{\mathcal{R}}$ . It is well known that a TBox  $\mathcal{T}$  can be transformed into an equivalent normalized TBox  $\mathcal{T}_{\mathcal{N}}$  of linear size in polynomial time [Bra04].

An Ontology  $\mathcal{O}$  (or sometimes also called a Knowledge Base) is a tuple  $(\mathcal{T}, \mathcal{A})$  where  $\mathcal{T}$  is a TBox and  $\mathcal{A}$  is an ABox which represents assertional knowledge of the problem domain. The entities in ABoxes are individual names specific to a given domain. For a detail definition of ABoxes, we refer to [BCM<sup>+</sup>10]. An ontology  $\mathcal{O}$  is a finite set of all axioms both in TBox  $\mathcal{T}$  and ABox  $\mathcal{A}$ . For simplicity, we only consider ontologies with  $\mathcal{A} = \emptyset$  i.e.,  $\mathcal{T} = \mathcal{O}$  unless stated otherwise. In other words an ontology  $\mathcal{O}$  is a finite set of GCIs and RIs. The cardinality of  $\mathcal{O}$ , denoted  $|\mathcal{O}|$ , is the number of axioms in  $\mathcal{O}$ .

## CHAPTER 2. PRELIMINARIES

### Semantics of $\mathcal{EL}^+$

The semantics of a description logic identifies it as a fragment of first-order predicate logic. We assume standard Tarski-style semantics in which an interpretation is defined as follows [BBL05]:

An  $\mathcal{EL}^+$  interpretation  $\mathcal{I}$  is a pair  $(\Delta^\mathcal{I}, \cdot^\mathcal{I})$  where  $\Delta^\mathcal{I}$  is a non-empty set, called the domain of  $\mathcal{I}$ , and a mapping  $\cdot^\mathcal{I}$  that assigns a set  $C^\mathcal{I} \subseteq \Delta^\mathcal{I}$  to each concept name  $C \in \mathbf{N}_C$  and a binary relation  $r^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I}$  to each role name  $r \in \mathbf{N}_R$ . The notion of interpretation can be extended to concept description in the obvious way as in Table 2.2. An interpretation  $\mathcal{I}$  satisfies a GCI  $C \sqsubseteq D$  if and only if  $C^\mathcal{I} \subseteq D^\mathcal{I}$ . Similarly, an interpretation satisfies an RI  $r_1 \circ \dots \circ r_n \sqsubseteq r$  if and only if the composition of  $r_1^\mathcal{I} \circ \dots \circ r_n^\mathcal{I}$  is a sub-relation of  $r^\mathcal{I}$  i.e.,  $r_1^\mathcal{I} \circ \dots \circ r_n^\mathcal{I} \subseteq r^\mathcal{I}$  where  $\circ$  represents the standard composition of binary relations. In such a case, we write  $\mathcal{I} \models r_1 \circ \dots \circ r_n \sqsubseteq r$ .

**Definition 6** (Model). *Let  $\mathcal{I}$  is a model of  $\mathcal{O}$ , denoted  $\mathcal{I} \models \mathcal{O}$ , if and only if  $\mathcal{I} \models \alpha$  for each  $\alpha \in \mathcal{O}$  i.e.,  $\mathcal{I}$  satisfies all the axioms in  $\mathcal{O}$ .*

An ontology is inconsistent if it has no model, otherwise it is consistent. Moreover, an axiom  $\alpha$  is a consequence of an ontology  $\mathcal{O}$ , denoted  $\mathcal{O} \models \alpha$ , if every model of  $\mathcal{O}$  satisfies  $\alpha$ .

### Reasoning in Lightweight DLs

For any given ontology  $\mathcal{O}$  and an axiom  $\alpha$ , the general reasoning problem is to check the entailment of that axiom (i.e.,  $\mathcal{O} \models \alpha$ ). If  $\alpha$  is a concept

## CHAPTER 2. PRELIMINARIES

inclusion ( $\alpha = C \sqsubseteq D$ ) and  $\mathcal{O}$  consists of axioms in  $\mathcal{T}$ , i.e.,  $\mathcal{O} = \mathcal{T}^4$ , the problem is known as subsumption checking or concept subsumption. The concept subsumption is an elementary reasoning service that is defined as follows [BBL05]:

**Definition 7** (Concept Subsumption). *A concept  $C$  is subsumed by  $D$  w.r.t. TBox  $\mathcal{T}$ , denoted  $\mathcal{T} \models C \sqsubseteq D$  or  $C \sqsubseteq_{\mathcal{T}} D$ , if and only if  $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$  for all models  $\mathcal{I}$  of  $\mathcal{T}$ . Concepts  $C$  and  $D$  are equivalent w.r.t.  $\mathcal{T}$ , denoted  $\mathcal{T} \models C \equiv D$ , if and only if  $\mathcal{T} \models C \sqsubseteq D$  and  $\mathcal{T} \models D \sqsubseteq C$ .*

In the  $\mathcal{EL}^+$  description logic, subsumption checking is an important inference problem that defines a preorder relation<sup>5</sup> on concepts. Concept subsumption is a vital property because any property that is necessarily true for a subsuming concept is true for all the subsumed concepts. In DLs, many reasoning problems (i.e, equivalence, satisfiability and consistency checking) can be reduced to subsumption checking [BCM<sup>+</sup>10].

### Classification and $\mathcal{EL}^+$ SAT Encoding in $\mathcal{EL}^+$

In practice checking entailment of an axiom sometimes requires a reasoning task that consists of checking multiple consequences at once. This problem is resolved under ontology classification, in which the goal is to compute all entailed instances (subsumption relations) of all atomic concepts occurring in TBox  $\mathcal{T}$ . It is an important reasoning service that creates taxonomic hierarchy

---

<sup>4</sup>As stated before, we only consider TBox  $\mathcal{T} = \mathcal{O}$  with  $\mathcal{A} = \emptyset$  unless stated otherwise.

<sup>5</sup>Reflexive and transitive but not antisymmetric.

## CHAPTER 2. PRELIMINARIES

Rules ( $\mathcal{R}$ )	Preconditions		Inferred axiom
$R_1$	$A \sqsubseteq A_i, 1 \leq i \leq n$	$A_1 \sqcap \dots \sqcap A_n \sqsubseteq B$	$A \sqsubseteq B$
$R_2$	$A \sqsubseteq A_1$	$A_1 \sqsubseteq \exists r.B$	$A \sqsubseteq \exists r.B$
$R_3$	$A \sqsubseteq \exists r.B, B \sqsubseteq B_1$	$\exists r.B_1 \sqsubseteq B_2$	$A \sqsubseteq B_2$
$R_4$	$A_{i-1} \sqsubseteq \exists r_i.A_i, 1 \leq i \leq n$	$r_1 \circ \dots \circ r_n \sqsubseteq r$	$A_0 \sqsubseteq \exists r.A_n$

Table 2.3:  $\mathcal{EL}^+$  Completion Rules [BBL05, BPS07].

of concepts – an acyclic graph representing subsumptions between equivalence classes of atomic concepts occurring in an ontology – that are either subsuming or being subsumed by other concepts. Predominantly, the reasoning procedures use tableau-based algorithms [BS00a] to perform ontology classification.

For Lightweight DL  $\mathcal{EL}$ , the completion-based algorithm [Bra04, BBL05, BPS07, BBL08] provides an efficient polynomial time reasoning procedure for ontology classification that constructs a completion graph (canonical model) using inference rules (see Table 2.3) for a candidate subsumption relation. The classification of an  $\mathcal{EL}^+$  ontology (or TBox  $\mathcal{T}$ ) obtained via completion-based algorithm can be encoded as a Horn propositional formula using  $\text{EL}^+\text{SAT}$  encoding scheme [SV09, SV15]. The propositional Horn formula  $\mathcal{H}$ , mimics the classification of a TBox  $\mathcal{T}$ , is constructed as follows:

1. Each axiom  $\alpha_i \in \mathcal{T}$  is initially assigned a unique selector variable  $s_{[\alpha_i]}$ . For trivial GCIs of the form  $C \sqsubseteq C$  or  $C \sqsubseteq \top$ , the variable  $s_{[\alpha_i]}$  is constant **true**.
2. In normalized TBox  $\mathcal{T}_N$  of  $\mathcal{T}$ , an axiom  $\alpha_i$  is substituted by a set of axioms in normal form  $\{\alpha_{i1}, \dots, \alpha_{im_i}\}$  and the following Horn clauses are

## CHAPTER 2. PRELIMINARIES

added to  $\mathcal{H}$ :

$$s_{[\alpha_i]} \rightarrow s_{[\alpha_{ik}]}$$

where  $1 \leq k \leq m_i$ .

3. During the execution of the completion-based algorithm a normalized TBox  $\mathcal{T}_N$  is saturated through an exhaustive application of the so-called *completion rules*  $\mathcal{R} = \{R_1, \dots, R_4\}$ , shown in [Table 2.3](#). Whenever a rule  $R_i \in \mathcal{R}$  is applied, with antecedents  $\text{ant}(R_i)$ , leading to inferring an axiom  $\alpha_i$ , the following Horn clause is added to  $\mathcal{H}$ :

$$\left( \bigwedge_{\alpha_j \in \text{ant}(R_i)} s_{[\alpha_j]} \right) \rightarrow s_{[\alpha_i]}$$

The resulting Horn formula  $\mathcal{H}$  encodes all possible derivations of completion rules inferring any axiom in TBox  $\mathcal{T}$  such that  $C \sqsubseteq_{\mathcal{T}} D$  with  $C, D \in \mathbb{N}_{\mathcal{C}}$ .

### Minimal Axiom Sets (MinAs) and Diagnosis

Sometimes it is imperative to find the concise explanations for a consequence (concept inclusion) from a TBox  $\mathcal{T}$  rather than merely deciding one. Previously, an erroneous subsumption was reported in earlier version of SNOMED CT that was mistakenly classify an amputation of a finger as the amputation of an arm ( $\text{AmputationOfFinger} \sqsubseteq \text{AmputationOfArm}$ ) [[BS08](#)]. Axiom pinpointing is an important step in debugging (error-prone) ontologies that computes concise explanations for a consequence (i.e., concept subsumption) and repairs (remove unintended consequences) through diagnosis, if necessary.

In axiom pinpointing, the goal is to find one or more minimal axiom sets (Mi-

## CHAPTER 2. PRELIMINARIES

nAs) in a TBox  $\mathcal{T}$  from which a consequence can be inferred. The standard definition of MinA and Diagnosis are assumed [BPS07, SV09]:

**Definition 8** (MinA, Diagnosis). *Let  $\mathcal{T}$  is a  $\mathcal{EL}^+$  TBox, and  $C, D \in \text{PC}_{\mathcal{T}}$  are primitive concept names, with  $C \sqsubseteq_{\mathcal{T}} D$ . Let  $\mathcal{S}$  is a subset of  $\mathcal{T}$  such that  $C \sqsubseteq_{\mathcal{S}} D$  and  $C \not\sqsubseteq_{\mathcal{S}'} D$  for  $\mathcal{S}' \subset \mathcal{S}$ , then  $\mathcal{S}$  is a minimal axiom set (MinA) w.r.t.  $C \sqsubseteq_{\mathcal{T}} D$ . A diagnosis for  $C \sqsubseteq D$  w.r.t.  $\mathcal{T}$  is a minimal subset (w.r.t. set inclusion)  $\mathcal{D} \subseteq \mathcal{T}$  such that  $C \not\sqsubseteq_{\mathcal{T} \setminus \mathcal{D}} D$ .*

It is well known that the set of all MinAs can be obtained from the set of all diagnoses, and *vice versa* (also see [Section 3.1](#)).

**Example 2.2.2.** *Let us consider an  $\mathcal{EL}^+$  medical ontology  $\mathcal{T}_{\text{med}}$  adapted from the GALEN medical ontology [[RH97](#)].*

*The ontology expresses a medical condition in which endocarditis is classified as a heart disease (i.e.,  $\text{Endocarditis} \sqsubseteq \text{HeartDisease}$ ). This disease occurs due to a bacteria that damages endocardium, a tissue, that provides a protection*

## CHAPTER 2. PRELIMINARIES

*to the heart valves. The axioms in  $\mathcal{T}_{\text{med}}$  are:*

- 1 :  $\text{Endocarditis} \sqsubseteq \text{Inflammation} \sqcap \exists \text{contIn.Heart}$
- 2 :  $\text{Inflammation} \sqsubseteq \text{Disease}$
- 3 :  $\text{HeartDisease} \equiv \text{Disease} \sqcap \exists \text{contIn.Heart}$
- 4 :  $\text{Endocarditis} \sqsubseteq \exists \text{hasLoc.Endocardium}$
- 5 :  $\text{Endocardium} \sqsubseteq \text{Tissue} \sqcap \exists \text{contIn.HeartValve}$
- 6 :  $\text{Inflammation} \sqcap \exists \text{HeartValve} \sqsubseteq \text{HeartDisease}$
- 7 :  $\text{hasLoc} \circ \text{contIn} \sqsubseteq \text{hasLoc}$

*There are two MinAs for  $\text{Endocarditis} \sqsubseteq \text{HeartDisease}$  w.r.t  $\mathcal{T}_{\text{med}}$ , namely  $\mathcal{S}_1 = \{1, 2, 3\}$  and  $\mathcal{S}_2 = \{1, 4, 5, 6, 7\}$ . Diagnoses for this subsumption relation are  $\mathcal{D}_1 = \{1\}$ ,  $\mathcal{D}_2 = \{7, 2\}$ ,  $\mathcal{D}_3 = \{4, 3\}$ ,  $\mathcal{D}_4 = \{6, 3\}$ ,  $\mathcal{D}_5 = \{5, 3\}$ ,  $\mathcal{D}_6 = \{7, 3\}$ ,  $\mathcal{D}_7 = \{5, 2\}$ ,  $\mathcal{D}_8 = \{6, 2\}$ , and  $\mathcal{D}_9 = \{4, 2\}$ .*

*The medical ontology  $\mathcal{T}_{\text{med}_N}$ , is normalized and classified using the comple-*

## CHAPTER 2. PRELIMINARIES

tion rules in Table 2.3, contains the following axioms:

- 8 : Endocarditis  $\sqsubseteq$  Inflammation

10 : HeartDisease  $\sqsubseteq$  Disease

12 : Endocarditis  $\sqsubseteq$   $\exists \text{contIn.Heart}$

14 : HeartDisease  $\sqsubseteq$   $\exists \text{contIn.Heart}$

16 : Disease  $\sqcap$  #10  $\sqsubseteq$  HeartDisease

18 :  $\exists \text{contIn.Heart}$   $\sqsubseteq$  #10

20 :  $\overbrace{\text{Endocarditis} \sqsubseteq \text{Disease}}^{R_1(8,9), R_1(10,24)}$

21 :  $\overbrace{\text{Endocarditis} \sqsubseteq \#10}^{R_1(12,18)}$

22 :  $\overbrace{\text{HeartDisease} \sqsubseteq \#10}^{R_1(14,18)}$

23 :  $\overbrace{\text{Endocarditis} \sqsubseteq \exists \text{hasLoc.HeartValve}}^{R_4(7,13,15)}$

24 :  $\overbrace{\text{Endocarditis} \sqsubseteq \text{HeartDisease}}^{R_1(16,20,21), R_1(8,17,25)}$

25 :  $\overbrace{\text{Endocarditis} \sqsubseteq \#18}^{R_3(19,23)}$

Note: #10 and #18 are new defined concepts

The  $EL^+SAT$  encoding scheme uses completion rules  $R_1(\mathbb{N}, \mathbb{N}), \dots, R_4(\mathbb{N}, \mathbb{N})$  in Table 2.3 for Horn Encoding. We first assign selector variables to axioms,

## CHAPTER 2. PRELIMINARIES

denoted  $s_{\mathbb{N}}$ , in  $TBox \mathcal{T}_{med}$  and  $TBox \mathcal{T}_{med_N}$ :

$s_1 \rightarrow \text{Endocarditis} \sqsubseteq \text{Inflammation} \sqcap \exists \text{contIn.Heart}$

$s_2 \rightarrow \text{Inflammation} \sqsubseteq \text{Disease}$

$s_3 \rightarrow \text{HeartDisease} \equiv \text{Disease} \sqcap \exists \text{contIn.Heart}$

$s_4 \rightarrow \text{Endocarditis} \sqsubseteq \exists \text{hasLoc.Endocardium}$

$s_5 \rightarrow \text{Endocardium} \sqsubseteq \text{Tissue} \sqcap \exists \text{contIn.HeartValve}$

$s_6 \rightarrow \text{Inflammation} \sqcap \exists \text{HeartValve} \sqsubseteq \text{HeartDisease}$

$s_7 \rightarrow \text{hasLoc} \circ \text{contIn} \sqsubseteq \text{hasLoc}$

$s_8 \rightarrow \text{Endocarditis} \sqsubseteq \text{Inflammation}$

$s_{17} \rightarrow \text{Inflammation} \sqcap \#18 \sqsubseteq \text{HeartDisease}$

$s_9 \rightarrow \text{Inflammation} \sqsubseteq \text{Disease}$

$s_{18} \rightarrow \exists \text{contIn.Heart} \sqsubseteq \#10$

$s_{10} \rightarrow \text{HeartDisease} \sqsubseteq \text{Disease}$

$s_{19} \rightarrow \exists \text{hasLoc.HeartValve} \sqsubseteq \#18$

$s_{11} \rightarrow \text{Endocardium} \sqsubseteq \text{Tissue}$

$s_{20} \rightarrow \text{Endocarditis} \sqsubseteq \text{Disease}$

$s_{12} \rightarrow \text{Endocarditis} \sqsubseteq \exists \text{contIn.Heart}$

$s_{21} \rightarrow \text{Endocarditis} \sqsubseteq \#10$

$s_{13} \rightarrow \text{Endocarditis} \sqsubseteq \exists \text{hasLoc.Endocardium}$

$s_{22} \rightarrow \text{HeartDisease} \sqsubseteq \#10$

$s_{14} \rightarrow \text{HeartDisease} \sqsubseteq \exists \text{contIn.Heart}$

$s_{23} \rightarrow \text{Endocarditis} \sqsubseteq \exists \text{hasLoc.HeartValve}$

$s_{15} \rightarrow \text{Endocardium} \sqsubseteq \exists \text{contIn.HeartValve}$

$s_{24} \rightarrow \text{Endocarditis} \sqsubseteq \text{HeartDisease}$

$s_{16} \rightarrow \text{Disease} \sqcap \#10 \sqsubseteq \exists \text{HeartDisease}$

$s_{25} \rightarrow \text{Endocarditis} \sqsubseteq \#18$

The Horn formula  $\mathcal{H}$ , encodes the classification of  $TBox \mathcal{T}_{med_N}$  using completion rules  $R_1(\mathbb{N}, \mathbb{N})$ , mentioned in superscript of axioms 20 to 25, translated

into following clauses:

## CHAPTER 2. PRELIMINARIES

$$\begin{array}{ll}
s_8 \wedge s_9 \rightarrow s_{20} & s_{19} \wedge s_{23} \rightarrow s_{25} \\
s_{12} \wedge s_{18} \rightarrow s_{21} & s_{10} \wedge s_{24} \rightarrow s_{20} \\
s_{14} \wedge s_{18} \rightarrow s_{22} & s_{16} \wedge s_{20} \wedge s_{21} \rightarrow s_{24} \\
s_7 \wedge s_{13} \wedge s_{15} \rightarrow s_{23} & s_8 \wedge s_{17} \wedge s_{25} \rightarrow s_{24}
\end{array}$$

Since we may require MinAs in terms of original axioms  $\mathcal{T}_{\text{med}}$ , the Horn formula  $\mathcal{H}$  contains the following set of clauses such that each clause maps an original axiom to the normalized version:

$$\begin{array}{lll}
s_1 \rightarrow s_8 & s_2 \rightarrow s_9 & s_3 \rightarrow s_{10} \\
s_1 \rightarrow s_{12} & s_4 \rightarrow s_{13} & s_5 \rightarrow s_{11} \\
s_3 \rightarrow s_{14} & s_5 \rightarrow s_{15} & s_3 \rightarrow s_{16} \\
s_6 \rightarrow s_{17} & s_{16} \rightarrow s_{18} & s_{17} \rightarrow s_{19}
\end{array}$$

For axiom pinpointing in Lightweight DL  $\mathcal{EL}^+$  SAT-based algorithms [SV09, SV15] exploit the ideas from early work on SAT solving [BHvMW09] and AllSMT [LNO06b] and compute one or more MinAs for any given consequence (i.e.,  $C \sqsubseteq D$ ). Instead, we can make use of more effective MUS extraction and enumeration algorithms [LS08, LS05, LS08, LML<sup>+</sup>09, BMS11, MSL11, BMS12, BLMS12, PM13, LPMMS15]) (see Section 3.1) for axiom pinpointing in  $\mathcal{EL}$  family of Description logics and its related problems (see Chapter 4, Chapter 5 and Chapter 6). We relate MinAs with MUSes of a partial MaxSAT

## CHAPTER 2. PRELIMINARIES

$\mathcal{F} = \langle \mathcal{H}, \mathcal{S} \rangle$  or Group Horn formula  $\mathcal{F} = \langle \mathcal{G}_0, \mathcal{G}_1, \dots, \mathcal{G}_n \rangle$ , in which the Horn encoding of the classification of TBox  $\mathcal{T}$  (explained above) and the negated goal (concept inclusion) is encoded as a Hard clause (in  $\mathcal{H}$  or  $\mathcal{G}_0$ ) and labeling axioms of TBox  $\mathcal{T}$  as unit soft clauses (in  $\mathcal{S}$  or  $\mathcal{G}_1, \dots, \mathcal{G}_n$ ), such that the MinAs are enumerated as minimal unsatisfiable subsets of  $\mathcal{F}$  (see details [Section 3.2](#)).

### Complexity of Decision Problems in Lightweight DLs

From a query complexity perspective, the concept subsumption in standard description logic  $\mathcal{ALC}$  is PSPACE [[BCM<sup>+</sup>10](#)]. For Lightweight DL  $\mathcal{EL}^+$ , the subsumption check is quite cheap (is in P) but in worst-case there may exist exponential many minimal subsets (w.r.t. set inclusion) for any given consequence (or subsumption) [[BPS07](#)]. Finding a cardinality bound on a MinA is again an NP-hard problem. More concretely, the decision problem that checks the existence of a MinA within a cardinality bound (cardinality  $\leq n$  for a given natural number  $n$ ) is an NP-Complete problem [[BPS07](#)] and computing the cardinality-minimum MinA (smallest MinA) is  $\Delta_2^P[\log n]$ -complete [[Bie08](#)].

## 2.3 Hypergraph Theory

In this section, we discuss hypergraph transversal and its related problems and establishes a link between these well-known problems and axiom pinpointing in  $\mathcal{EL}$  family of Description Logics (see [Section 2.3](#)).

A hypergraph is a generalized graph in which edges can connect any number of vertices and each edge should be non-empty [[Ber85](#)].

## CHAPTER 2. PRELIMINARIES

**Definition 9.** A hypergraph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  is a pair consisting of a vertex set  $\mathcal{V} = \{v_i | 1 \leq i \leq n\}$ , and a set of hyperedges  $\mathcal{E} = \{E_j | 1 \leq j \leq m\}$  where  $E_j \subseteq \mathcal{V}$ .

Each vertex(or node) in  $\mathcal{G}$  should occur in an edge (i.e.,  $\mathcal{V} = \bigcup_{j \in 1}^m E_j \in \mathcal{E}$ ). A transversal of a hypergraph is a subset of vertices in the graph which has a non-empty intersection with every hyperedge.

**Definition 10.** A set of vertices  $W \subseteq \mathcal{V}$  is a transversal of  $\mathcal{G}$  if  $\forall E \in \mathcal{E}, E \cap W \neq \emptyset$  and  $W$  is minimal iff  $\forall W' \subsetneq W, W'$  is not a transversal of  $\mathcal{G}$ .

The set of all minimal transversals of  $\mathcal{G}$ , denoted  $\text{Tr}(\mathcal{G})$ , constitutes another hypergraph on  $\mathcal{V}$ .

**Example 2.3.1.** Let  $\mathcal{G} = \{\{a, b, c\}, \{b, d, e\}, \{c, f\}\}$  be a hypergraph. The set of all minimal transversals of  $\mathcal{G}$  are:

$$\text{Tr}(\mathcal{G}) = \{\{a, d, f\}, \{a, e, f\}, \{c, d\}, \{c, e\}, \{b, f\}, \{b, c\}\}$$

Hypergraph Transversal [Ber85] is a well-known enumeration problem in hypergraph theory that has numerous important applications in boolean satisfiability theory [DK04], combinatorial optimization [Wol09], and model-based diagnosis [EG95b, EG02, EGM03a], among others. In Lightweight Description Logic  $\mathcal{EL}$  a hypergraph, also called proof structure, of the TBox  $\mathcal{T}$  yields all the traces of deduction or completion rules that can be used to drive all subsets of TBoxes  $\mathcal{T}$  for an entailment (concept inclusion) [BP10a, BPS07, CP14].

## CHAPTER 2. PRELIMINARIES

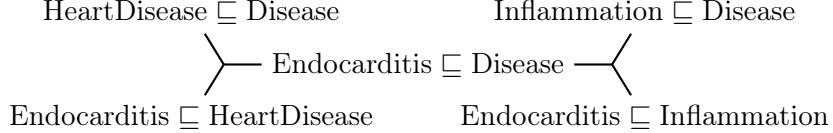


Figure 2.1: Hypergraph explaining concept inclusion  $\text{Endocarditis} \sqsubseteq \text{Disease}$ .

**Example 2.3.2.** Continue with the Example 2.2.2, the Figure 2.1 is a proof structure (or hypergraph) of normalized TBox  $\mathcal{T}_{\text{med}_N}$  of  $\mathcal{T}_{\text{med}}$  that contains explanations using hyperedges for concept inclusion ( $\text{Endocarditis} \sqsubseteq \text{Disease}$ ).

The problem of computing minimal transversals, i.e., generating  $\text{Tr}(\mathcal{G})$ , is defined as follows [Ber85, Ser09]:

**Problem:** Transversal enumeration (TRANS-ENUM)

**Input:** A hypergraph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  on a finite set  $\mathcal{V}$

**Output:** Finding the edges of the transversal hypergraph  $\text{Tr}(\mathcal{G})$

A decision variant of TRANS-ENUM is another well-known problem called Transversal Hypergraph Problem (TRANS-HYP), defined as follow [Ber85, Ser09]:

**Problem:** Transversal Hypergraph (TRANS-HYP)

**Input:** Two hypergraphs  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  and  $\mathcal{H} = (\mathcal{V}, \mathcal{E})$

**Question:** Is  $\mathcal{G}$  the transversal hypergraph of  $\mathcal{H}$ , i.e., does  $\text{Tr}(\mathcal{H}) = \mathcal{G}$  hold?

The computational complexity of TRANS-HYP has also been extensively studied [EG95a, EG02, EGM03b, KS03] and has a range of applications in areas of machine learning [DMP99], data mining [BMR03], description logic [PS09] and probabilistic reasoning [CP14], among others. It is known that TRANS-

## CHAPTER 2. PRELIMINARIES

HYP is solvable in output-polynomial time if and only if TRANS-ENUM is solvable in polynomial time [BI95]. TRANS-HYP is known to be in CoNP ( $\log^2$ ), a subclass of Co-NP and whether its CoNP-hard is an open problem [EG95a]. The best known algorithm for this problem is the quasi-polynomial time algorithm [FK96]. This algorithm tests the duality of a pair of monotone boolean function in disjunctive normal form – Monotone duality problem is same as TRANS-HYP [EG02] – is in time  $n^{0(\log n)}$ . Thus, TRANS-HYP is not an NP-hard problem because a sub-exponential running time algorithm (i.e.,  $n^{0(\log n)}$ ) can not solve an NP-complete problem. Similarly, whether TRANS-ENUM can be solved in output-polynomial time is another open problem. Moreover, it has been investigated that the complexity of axiom pinpointing in  $\mathcal{EL}$  family of description logics is as hard as minimal hypergraph transversal problem [PS09]. Thus the problem of axiom pinpointing cannot be solved in output-polynomial time unless  $P \neq NP$  (see [Subsection 3.2.4](#)). Although the worst case complexity makes it unlikely to find all MinAs but we can compute it efficiently, if possible, using SAT-based methods and techniques explained in next [Chapter 3](#).

## Chapter 3

# Axiom pinpointing via Enumeration of MCSes/MUSes

Reiter’s model-based diagnosis or diagnosis describes minimal subsets of axioms that are discarded to remove conflicts from a system or an ontology and repairs are maximal subsets that do not yield (erroneous) consequences. In the Boolean Satisfiability Problem (SAT) diagnoses are called Minimal Correction Subsets (MCSes), complement to repairs that are Maximal Satisfiable Subsets (MSSes). On the contrary, conflicts are called Minimal Unsatisfiable Subsets (MUSes) that explain inconsistencies in an ontology. In this chapter, we discuss MUSes and MCSes, their extraction and enumeration in relation with Reiter’s model-based diagnosis (see [Section 3.1](#)). Furthermore, we make use of these methods and their extensions to address the problem of axiom pinpointing for the Lightweight Description Logic  $\mathcal{EL}^+$  (see [Section 3.2](#)). We

## CHAPTER 3. AXIOM PINPOINTING VIA ENUMERATION OF MCSES/MUSES

also discuss the complexity results of decision and enumeration problems both in SAT and Lightweight DLs (see [Subsection 3.2.4](#)).

### 3.1 MUSes and MCSes Enumeration

Reiter’s model-based diagnosis [\[Rei87\]](#) explains the encountered inconsistencies between expected and experienced behavior of an over constrained system. The goal is to identify minimal subsets of the components (conflicts) of a system and inconsistent observations, on inputs and outputs, such that discarding some components (diagnosis) restore consistency [\[MJIM15\]](#). Under these settings, minimal correction subsets (MCSes) are called diagnoses and minimal unsatisfiable subsets (MUSes) are referred to as (minimal) conflicts. It is believed that the enumeration of MUSes is far more challenging than the enumeration of MCSes because blocking the computed MUSes is non-trivial and required us to exploit the fundamental minimal hitting set relation between MUSes and MCSes. In the next section, we briefly introduce the minimal hitting set enumeration problem in relation with MUS enumeration and axiom pinpointing in the Lightweight DL  $\mathcal{EL}^+$ .

#### 3.1.1 Minimal Hitting Set Enumeration

The minimal hitting sets of conflict sets provide a solid foundation in model-based diagnostic reasoning. We define the hitting set, its minimality variant and the smallest minimal hitting set w.r.t to the cardinality in the following:

**Definition 11.** *Given a collection of sets  $\mathcal{U} = \{S_1, \dots, S_n\}$ , a set  $h \subseteq \bigcup_{S_i \in \mathcal{U}} S_i$*

### CHAPTER 3. AXIOM PINPOINTING VIA ENUMERATION OF MCSES/MUSES

*is a hitting set for  $\mathcal{U}$ , if and only if any set  $S_i \in \mathcal{U}$  the intersection with  $h$  is non-empty, (i.e.  $h \cap S_i \neq \emptyset$ ). A minimal hitting set (MHS)  $h$  is a subset minimal of hitting sets (i.e.  $\forall h' \subset h$ ,  $h'$  is not a hitting set) of  $\mathcal{U}$  and  $h$  is the smallest minimal hitting set if  $|h| \leq |h'|$  for every minimal hitting set  $h'$ .*

**Example 3.1.1.** For example,  $\mathcal{U} = \overbrace{\{\{a, b, e\}\}}^{S_1}, \overbrace{\{\{b, c, d\}\}}^{S_2}, \overbrace{\{\{a, b\}\}}^{S_3}, \overbrace{\{\{b, d, e\}\}}^{S_4}$  is a collection of sets. Then, the minimal hitting sets (MHSESes) of  $\mathcal{U}$  are  $h_1 = \{b\}$ ,  $h_2 = \{a, c, e\}$ , and  $h_3 = \{a, d\}$ . Moreover, the set  $h_1 = \{b\}$  is the smallest possible minimal hitting set w.r.t to the cardinality.

We refer to the problem of computing all the minimal hitting sets (MHSES) as the Minimal Hitting Set Enumeration (MHS-ENUM) problem that is defined as follows:

**Problem:** Minimal Hitting Set Enumeration (MHS-ENUM)

**Input:**  $\mathcal{U}$  is a collection of sets

**Output:** Finding all the minimal hitting sets of  $\mathcal{U}$

MHS-ENUM problem plays a crucial role in the analysis of unsatisfiable boolean constraints and has been studied extensively [Rei87, BL03, BS05, LS08, LM13, PM13, LPMMS15]. In case the input contains a hypergraph  $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ , the problem of computing all the MHSESes (or minimal transversals) is referred as Transversal Enumeration (TRANS-ENUM) problem that is a well-studied open problem and have applications in both theoretical and applied computer science [EG95a, EGM03b, FHK<sup>+</sup>14] (see details [Section 2.3](#)). The minimal hitting set enumeration (MHS-ENUM) can be made equivalent

## CHAPTER 3. AXIOM PINPOINTING VIA ENUMERATION OF MCSES/MUSES

to transversal enumeration problem (TRANS-ENUM) [EG02] and shares the complexity bounds with this well-known problem.

There is a well known duality relation that exists between MUSes (conflicts) and MCSes (diagnosis) in unsatisfiable boolean constraints (model-based diagnostic reasoning), as shown by the following:

**Theorem 1.** *Let  $\mathcal{F}$  is propositional unsatisfiable formula. Then, each MUS  $\mathcal{M}$  of  $\mathcal{F}$  is a minimal hitting set of the MCSes of  $\mathcal{F}$  and each MCS  $\mathcal{C}$  of  $\mathcal{F}$  is a minimal hitting set of the MUSes of  $\mathcal{F}$ .*

For enumerating MUSes of an unsatisfiable formula  $\mathcal{F}$ , there exist distinct approaches that either exploit explicit hitting set dualization (**S.1**), implicit hitting dualization (**S.2**), or implicit set enumeration [LMS04]. In this report, we only study the following:

**S.1** MUSes obtained by hitting set dualization [LS08, BS05].

**S.2** Implicitly enumerating over MUSes interleaved with MCSes [PM13, LM13, LPMMS15].

We have solved the problem of axiom pinpointing in Description Logic more effectively by exploiting the duality relation between MCSes and MUSes, explicitly (Subsection 3.1.2) as well as implicitly (Subsection 3.1.3).

### 3.1.2 Explicit Enumeration

Computing MUSes and MCSes of an unsatisfiable formula  $\mathcal{F}$  is the same as computing irreducible hitting sets of some collection of sets. An explicit

### CHAPTER 3. AXIOM PINPOINTING VIA ENUMERATION OF MCSES/MUSES

MUS enumeration algorithm [BS05, LS08] organizes the search in two distinct parts. First one computes minimal hitting sets (MCSEs) of  $\mathcal{F}$  using a constraint solver and the second step enumerates over MUSes using an irreducible hitting set computation algorithm [HBC07, LS08, MU11] because the collection of MUSes “encoded” within constraints are already made explicit in the first step. [Algorithm 1](#) presents the pseudo-code of an MCS extraction and enumeration algorithm implemented using MaxSAT. In [Algorithm 1](#) (line 6)  $Enc(\Sigma_{r \in \mathcal{R}}(r \leq \lambda))$  is a boolean cardinality constraint that can be encoded into a CNF using either sequential or parallel counters [Sin05], cardinality networks [ANORC09], pairwise cardinality networks [CZI10] or perfect hashing based encodings [BIMM12]. The first step constrains the upper bound on the number of relaxed clauses in working formula  $\varphi_W$  and extract all MCSEs of this over-constrained boolean formula using [Algorithm 1](#). For extracting and enumerating over MCSEs, [Algorithm 1](#) is quite effective because it only adds one relaxed variable per clause with a single constraint to restrict the number of relaxed clauses. Moreover, finding an MCS is easier than finding an MUS in practice because it is relatively simple to solve problems in NP as compared to the ones in Co-NP [LS08]. The second step of explicit MUS enumeration computes all minimal hitting set of already computed MCSEs of  $\mathcal{F}$  that solves an MHS-ENUM problem because each minimal hitting set (MHS) represents an MUS (see [Theorem 1](#)).

CAMUS [LS08, LS05] and the DAA algorithm [BS05] are the implementation of [S.1](#). CAMUS computes all MCSEs and then exploit the existing duality between MCSEs and MUSes to enumerate over MUSes. While the

CHAPTER 3. AXIOM PINPOINTING VIA ENUMERATION OF  
MCSES/MUSES

---

**Algorithm 1** Explicit Enumeration Algorithm [MLM12]

---

**Input:** CNF formula  $\varphi$   
**Output:** Reports the set of MCSEs of  $\varphi$

```

1 begin
2    $\varphi_W \leftarrow \varphi$                                 //  $\varphi_W$  is the working formula
3    $\mathcal{R} \leftarrow \emptyset$                           //  $\mathcal{R}$  are the set of relaxed variables
4    $\lambda \leftarrow 0$                                  //  $\lambda$  defines a lower bound on relaxed variables
5   while true do
6      $(st, \varphi_C, \mathcal{M}) \leftarrow \text{SAT}(\varphi_W \cup \text{Enc}(\Sigma_{r \in \mathcal{R}}(r \leq \lambda)))$     //  $\varphi_C$  is an unsat core
8     //  $\mathcal{M}$  is a maximal satisfying assignment
9     // "Enc" encodes the cardinality constraint
10    if  $st$  then
11      ReportMCS( $\mathcal{M}$ )                         //  $\mathcal{M}$  is a solution to MaxSAT problem
12    else
13      if ( $|\mathcal{R}| = |\text{soft}(\varphi)|$ ) ||  $\varphi_C \cap \text{soft}(\varphi) = \emptyset$  then
14        return FALSE // if all soft clauses are relaxed but  $\varphi_W$ 
15        is UNSAT then all MCSEs have been found
16
17    if  $\lambda = |\text{soft}(\varphi)|$  then
18      return FALSE // No solution to MaxSAT problem
19
20    if  $|\mathcal{R}| < |\text{soft}(\varphi)|$  then
21      for  $\omega \in \varphi_C \cap \text{soft}(\varphi)$  do
22         $\mathcal{R} \leftarrow \mathcal{R} \cup \{r\}$                 //  $r$  is a new relaxed variable.
23         $\omega_{\mathcal{R}} \leftarrow \omega \cup \{r\}$           // Clause  $\omega$  is relaxed.
24         $\varphi_W \leftarrow \varphi_W \setminus \{\omega\} \cup \{\omega_{\mathcal{R}}\}$ 
25
26     $\lambda \leftarrow \lambda + 1$ 

```

---

DAA algorithm computes one MCS at a time and adds it to the set of computed MCSEs. Every time an MCS is added to the set, an MHS is computed and tested for satisfiability. If the resulting set of clauses becomes unsatisfiable it is an MUS; otherwise, the satisfiable set is used as a starting point to compute the next MCS. CAMUS and DAA algorithms can be decoupled from the choice of MCS extraction and minimal hitting set computation algorithm.

### CHAPTER 3. AXIOM PINPOINTING VIA ENUMERATION OF MCSES/MUSES

Therefore, any improvement in MCS extraction algorithms [MHJ<sup>+</sup>13, BK15] can be harmoniously integrated and an off-the-shelf hypergraph transversal computation tool (e.g., shd [MU11] and MTminer [HBC07]) can be used for the computation of hitting set duals between MCSEs and MUSes because the minimal hitting set enumeration is equivalent to the hypergraph transversal problem. We have used explicit MUS enumeration in axiom pinpointing tool EL2MCS [AMMS15a] for the Lightweight DL  $\mathcal{EL}^+$ . EL2MCS enumerates the MCSEs of a partial MaxSAT problem via internally calling the CAMUS2 tool<sup>1</sup> [MHJ<sup>+</sup>13] and uses the recursive branching algorithm presented in CAMUS [LS08] to compute irreducible hitting sets a.k.a. MUSes (see details [Subsection 3.2.1](#)).

In the worst-case scenario, an unsatisfiable formula might contain exponential many MCSEs [GMP07]. Therefore, it is not always feasible to enumerate MUSes to its completion using CAMUS. Although, the advantage of DAA algorithm (mentioned above) over CAMUS is that some MUSes are returned before the set of MCSEs is exhausted. The drawback is that at each step requires the candidate set to be tested whether it is an MUS that is in contrast with CAMUS. But when the goal is to return MUSes quickly, an implicit MUS extraction and enumeration approach is shown to perform better in practice. The implicit minimal hitting set dualization (see [Subsection 3.1.3](#)) aims to complement, but not replace, the explicit dualization alternative and in settings where enumeration of MCSEs is feasible, explicit minimal hitting set dualization is a preferred option [PM13, IPLM15].

---

<sup>1</sup> Available from <http://logos.ucd.ie/web/doku.php?id=mcsls>.

## CHAPTER 3. AXIOM PINPOINTING VIA ENUMERATION OF MCSES/MUSES

### 3.1.3 Implicit Enumeration

As explained earlier, if the number of MCSes is exponentially large the above mentioned approaches will be unable to compute all MUSes, even if the total number of MUSes is quite small. The second approach [S.2](#) uses the implicit hitting set dualization and guarantees to find one or more MUSes of an over-constrained formula  $\mathcal{F}$ . An implicit enumeration algorithm [[PM13](#), [IPLM15](#)] (see details [Chapter 5](#)) maintains two formulas aimed at enumerating the MUSes and MCSes, as shown in [Figure 3.1](#). Formula  $\mathcal{Q}$  is defined over a set of selector variables  $I = \{p_i \mid c_i \in \mathcal{F}\}$  corresponding to clauses in  $\mathcal{F}$ , and used to enumerate subsets of  $\mathcal{F}$ . Iteratively, a solver computes a maximal model  $P$  of  $\mathcal{Q}$  and test whether the corresponding  $\mathcal{F}' \subseteq \mathcal{F}$  is satisfiable. If  $\mathcal{F}'$  is satisfiable,  $\mathcal{F} \setminus \mathcal{F}'$  is an MCS of  $\mathcal{F}$  because  $\mathcal{F}'$  represents an MSS of  $\mathcal{F}$ . We can block the computed MCS by adding a clause  $I \setminus P$  to  $\mathcal{Q}$ . Otherwise,  $\mathcal{F}'$  is reduced to an MUS, which is also blocked by adding a negative polarity clause to  $\mathcal{Q}$ . This prevents the generation of any superset of the MCS (subset of the MSS). The enumeration process terminates when  $\mathcal{Q}$  becomes unsatisfiable. Thus, it guarantees to find all MUSes and MCSes of  $\mathcal{F}$  in a number of iterations that corresponds to the sum of the number of MUSes and MCSes. Inline with hitting set minimality ([Subsection 3.1.1](#)), each maximal model computed on formula  $\mathcal{Q}$  corresponds to a MHS w.r.t the set of MUSes.

**Example 3.1.2.** *In this example, we compare the enumeration algorithm  $EL^+SAT$  [[SV09](#), [SV15](#)] with [[PM13](#), [LPMMS15](#)]. When [Table 2.1](#) in Example [2.1.1](#) is compared with Example [Table 3.1](#), the later on contains two unnec-*

CHAPTER 3. AXIOM PINPOINTING VIA ENUMERATION OF  
MCSES/MUSES

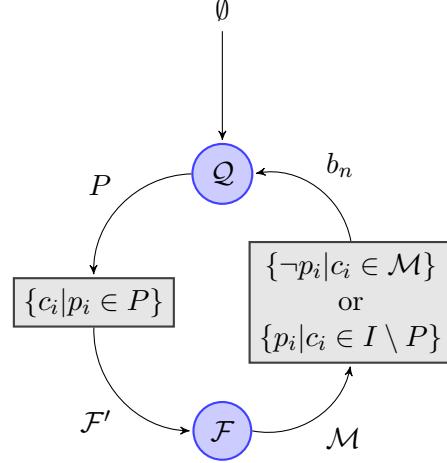


Figure 3.1: Implicit Enumeration Algorithm Model

MUSes	$\{\mathcal{G}_1\}$	$\mathcal{B}_1 = (\neg p_1)$	$\mu_1 = \{p_1, p_2, p_3, p_4\}$
	$\{\mathcal{G}_2, \mathcal{G}_3, \mathcal{G}_4\}$	$\mathcal{B}_2 = (\neg p_2 \vee \neg p_3 \vee \neg p_4)$	$\mu_2 = \{p_2, p_3, p_4\}$
<hr/>			
Non-minimal Sets	$\{\mathcal{G}_1, \mathcal{G}_2\}$	$\mathcal{B}_3 = (\neg p_3 \vee \neg p_4)$	$\mu_3 = \{p_3, p_4\}$
	$\{\mathcal{G}_1, \mathcal{G}_4\}$	$\mathcal{B}_4 = (\neg p_2 \vee \neg p_3)$	$\mu_4 = \{p_2, p_3\}$
	$\{\mathcal{G}_1, \mathcal{G}_3\}$	$\mathcal{B}_5 = (\neg p_2 \vee \neg p_4)$	$\mu_5 = \{p_2, p_4\}$
	$\{\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_4\}$	$\mathcal{B}_6 = (\neg p_3)$	$\mu_6 = \{p_3\}$
	$\{\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_3\}$	$\mathcal{B}_7 = (\neg p_4)$	$\mu_7 = \{p_4\}$
$\left. \right\} \text{EL}^+ \text{SAT}$			

Table 3.1: MUSes of  $\mathcal{F}$  using  $\text{EL}^+ \text{SAT}$  [SV09, SV15] and contains non-minimal MCSes, blocking clauses and models that can be avoided, see Example 2.1.1

essary maximal models  $\mu_6$  and  $\mu_7$  and blocking clauses  $\mathcal{B}_6$  and  $\mathcal{B}_7$ . In [SV09, SV15], the blocking clause  $\mathcal{B}_3 = \neg p_3 \vee \neg p_4$  yields two maximal models: with  $p_3 = 0$ , i.e. pick  $p_4 = 1$  and  $p_4 = 0$ , i.e. pick  $p_3 = 1$ . Instead, we can avoid

### CHAPTER 3. AXIOM PINPOINTING VIA ENUMERATION OF MCSES/MUSES

*this unnecessary computation using a blocking clause  $\mathcal{B}_3 = (p_1 \vee p_2)$  as proposed in Chapter 5.*

The use of maximal models for computing either MCSes of a formula or a set of clauses that contain an MUS was proposed in [PM13], which exploited SAT with preferences for computing maximal models [GM06, RG13]. Computing maximal models of a formula  $\mathcal{Q}$  can be reduced to the problem of extracting an MSS of a formula  $\mathcal{Q}'$  [MHJ<sup>+</sup>13], where the clauses of  $\mathcal{Q}$  are hard and each variable  $x_i \in V(\mathcal{Q})$  constitute a unit soft clause  $c_i \equiv \{x_i\}$ . The state-of-the-art MCS/MSS computation approaches outperform SAT with preferences [MHJ<sup>+</sup>13, GLM14, BDTK14, MPM15].

The tools EMUS [PM13] and MARCO [LM13, LPMMS15] are the implementations of the implicit minimal hitting set dualization approach **S.2**. The critical difference between MARCO and EMUS is the use of maximal model for the computation of the candidate set. In EMUS, if the candidate set is SAT its complement is guaranteed to be an MCS. This is contrast with non-maximal model approach, where more SAT calls are required to obtain an MCS. On the contrary, MARCO relies on a SAT solver for generating new seeds and solves a problem more complex than a pure hypergraph transversal problem.

For axiom pinpointing and debugging in Lightweight DL  $\mathcal{EL}^+$ , we propose state-of-the art SAT-based approach in EL2MCS [AMMS15a], EL2MUS [AMMS15b], and BEACON [AMI<sup>+</sup>16]. EL2MCS is an MCS-enumerator complemented with hitting set duality algorithms for axiom pinpointing in Lightweight DL  $\mathcal{EL}^+$ . EL2MUS, front-end to HGMUS [AMMS15b], is an efficient tool to enumerate MUSes of Horn formulae for axiom pinpointing in Lightweight

## CHAPTER 3. AXIOM PINPOINTING VIA ENUMERATION OF MCSES/MUSES

DLS. HGMUS is a state-of-the-art group-MUSes enumerator based on the linear search MCS extraction algorithm [MHJ<sup>+</sup>13] that is related with the novel Literal-Based extraction algorithm [MPM15]. BEACON is an efficient debugging tool for  $\mathcal{EL}^+$  ontologies that implements a Horn formula generator [SV09, SV15] and assimilates HGMUS with FORQES [IPLM15], a state-of-the-art solver for computing the smallest MUS. The SAT-based axiom pinpointing approach details are explained in the next section [Section 3.2](#).

### 3.2 Axiom pinpointing in Lightweight DL $\mathcal{EL}^+$

For  $\mathcal{EL}$  family of DLs, an application of enumeration of MUSes is realized in axiom pinpointing for description logic  $\mathcal{EL}^+$  [AMMS15a, AMMS15b, AMI<sup>+</sup>16]. As described earlier, axiom pinpointing defines a procedure to single-out axioms that are responsible for a consequence (concept inclusion) and is an important step in debugging (error-prone)  $\mathcal{EL}^+$  ontologies. The possible approach for enumerating MUSes of a Horn formula, encoding  $\mathcal{EL}^+$  ontology classification, is to use an existing solution based on explicit or implicit minimal hitting set dualization (see [Subsection 3.1.2](#) and [Subsection 3.1.3](#)). We have used explicit minimal hitting dualization in EL2MCS ([Subsection 3.2.1](#)) and implicit minimal hitting set dualization approach in HGMUS (see details [Subsection 3.2.2](#)) and BEACON (see details [Subsection 3.2.3](#)).

### CHAPTER 3. AXIOM PINPOINTING VIA ENUMERATION OF MCSES/MUSES

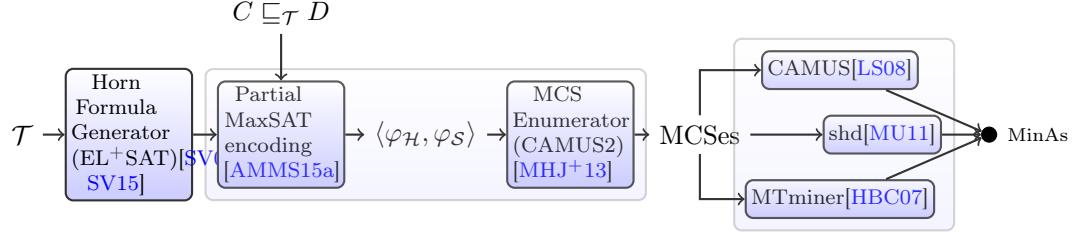


Figure 3.2: EL2MCS organization

#### 3.2.1 (EL2MCS) MCS Enumeration using HST

The organization of the EL2MCS tool is shown in Figure 3.2. EL2MCS encodes the classification of a TBox  $\mathcal{T}$ , into Horn formula  $\mathcal{H}$  using  $\text{EL}^+$ SAT encoding scheme [SV09, SV15] (see details Section 2.2), and generates a partial MaxSAT encoding  $\langle \varphi_{\mathcal{H}}, \varphi_{\mathcal{S}} \rangle$ , where  $\varphi_{\mathcal{H}} = \mathcal{H} \cup \{(\neg s_{[C \sqsubseteq D]})\}$ , added with a negated goal ( $C \sqsubseteq D$ ) and each soft clause  $\mathcal{S} = \{(s_{[\alpha_i]})\}$  is defined using a unit clause labeling every axiom in normalized TBox  $\mathcal{T}_{\mathcal{N}}$ . The purpose is to include as many axioms as possible, leaving out a minimal set which, if included, makes the formula unsatisfiable. Thus, each of these sets represents an MCS (a minimal set of axioms) of the MaxSAT problem formulation which needs to be dropped to refute the entailment (i.e. a *diagnosis* [LP14b]). MCS enumeration can be implemented with a MaxSAT solver [LS08, MLM12] or with a dedicated algorithm [MHJ<sup>+</sup>13]. This approach is in contrast with the  $\text{EL}^+$ SAT axiom pinpointing method [SV09, SV15] that explicitly enumerates the selector variables to axioms in a TBox  $\mathcal{T}$  using AllSMT-inspired approach [LNO06b]. From the encoding of axiom pinpointing in  $\mathcal{EL}^+$  to Horn formula, it is immediate the following result, that relates Theorem 3 in [SV15] with the dedicated algorithm

CHAPTER 3. AXIOM PINPOINTING VIA ENUMERATION OF  
MCSES/MUSES

proposed in earlier work [SV09, SV15]:

**Theorem 2.** *Given an  $\mathcal{EL}^+$  TBox  $\mathcal{T}$ , for every  $\mathcal{S} \subseteq \mathcal{T}$  and for every pair of concept names  $C, D \in \text{PC}_{\mathcal{T}}$ ,  $\mathcal{S}$  is a MinA of  $C \sqsubseteq_{\mathcal{S}} D$  if and only if the Horn propositional formula,*

$$\mathcal{H} \wedge_{\alpha_i \in \mathcal{S}} (s_{[\alpha_i]}) \wedge (\neg s_{[C \sqsubseteq D]}) \quad (3.1)$$

*is minimally unsatisfiable.*

*Proof.* By Theorem 3 in [SV15],  $C \sqsubseteq_{\mathcal{S}} D$  if and only if the associated Horn formula (3.1) is unsatisfiable. For a MinA  $\mathcal{S} \subseteq \mathcal{T}$ , minimal unsatisfiability of (3.1) (with  $\mathcal{T}$  replaced by  $\mathcal{S}$ ) results from the MinA computation algorithm proposed in earlier work [SV09, SV15].  $\square$

In EL2MCS, we extend the CAMUS2 tool [MHJ<sup>+</sup>13] that enumerates over MCSEs of the partial MaxSAT formula  $\langle \varphi_{\mathcal{H}}, \varphi_{\mathcal{S}} \rangle$  and exploits the minimal hitting set duality for computing all the MUSes given the set of MCSEs, as achieved in CAMUS tool<sup>2</sup> [LS08]. The hypergraph transversal computation tools, shd [MU11] and MTminer [HBC07], could be used instead, as shown in Figure 3.2. It should be observed that MCS enumeration in EL2MCS uses CAMUS2 which is an implementation of the MCS enumerator in CAMUS [LS08] and capable of handling partial MaxSAT formulae. We have also considered alternative MCS enumeration approaches [MHJ<sup>+</sup>13] but found them to be not to be as efficient as CAMUS or CAMUS2.

---

<sup>2</sup>Available from <http://sun.iwu.edu/~mliffito/camus/>.

## CHAPTER 3. AXIOM PINPOINTING VIA ENUMERATION OF MCSES/MUSES

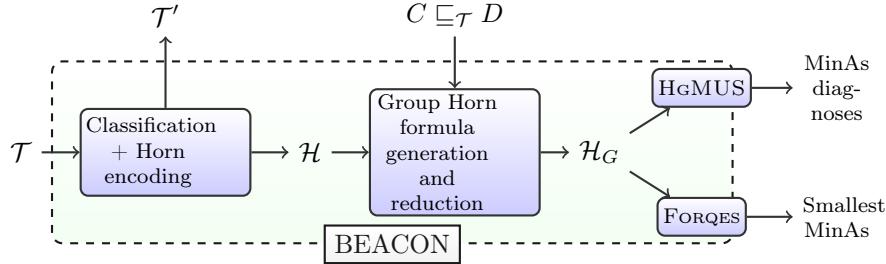
### 3.2.2 (HGMUS) group MUS Enumeration of Horn Formula

HGMUS is an efficient group-MUS enumerator for Horn formulae and used as a back-end to axiom pinpointing tools EL2MUS and BEACON. It is based on the partial MUS enumeration paradigm [PM13, LM13, LPMMS15] and implements implicit hitting set dualization along with efficient blocking of group MUSes and MCSes (see [Subsection 3.1.3](#)).

It shares the main organization of the implicit MUS enumeration algorithm (explained above), with  $\mathcal{F} = \mathcal{G}_0$  and  $\mathcal{Q}$  defined over selector variables for groups  $\mathcal{G}_i$ , with  $i > 0$ . For an optimized enumeration of maximal models, it uses a state-of-the-art MCS extraction algorithm (LBX) [MPM15] and has a built in dedicated insertion-based MUS extraction algorithm. However, it also includes an implementation of a dedicated Horn decision procedure – LTUR [Min88, DG84, IM87, Scu90] for reasoning on Horn formula  $\mathcal{F}$ . LTUR is a simplified implementation of Dowling & Gallier’s algorithm [DG84, Min88]. LTUR is expected to be more efficient than plain unit propagation in MinisAT because only variables assigned value 1 need to be propagated. Thus, performing a one-sided unit-propagation that has no branching factor and result in either no conflict or a single conflict during search. The simplicity of LTUR enables a very efficient implementation, that use adjacency lists for representing clauses instead of using more commonly used watched literal approach. Because, unit propagation in CDCL SAT solvers (i.e. that use watched literals) are not guaranteed to run in linear time [Gen13] as it is the case with Minisat [ES03] and its variants, for which unit propagation

### CHAPTER 3. AXIOM PINPOINTING VIA ENUMERATION OF MCSES/MUSES

Figure 3.3: BEACON organization



runs in worst-case quadratic time. As a result, using an off-the-shelf SAT solver and exploiting only unit propagation, as is done for example in earlier work [SV09, SV15, MP15]), is unlikely to be the most efficient solution (see Subsection 3.2.4). Thus, these features makes HGMUS a very efficient tool for axiom pinpointing in Lightweight DL  $\mathcal{EL}^+$  [AMMS15b, AMI<sup>+</sup>16] and ontology matching [JC11, ES13].

#### 3.2.3 (BEACON) Lightweight DL Debugger

BEACON is a SAT-based debugger for the Lightweight description logic  $\mathcal{EL}^+$ , as shown in Figure 3.3. BEACON encodes the classification of a TBox  $\mathcal{T}$ , into Horn formula  $\mathcal{H}$  using EL<sup>+</sup>SAT encoding scheme [SV09, SV15] (see details Section 2.2), and generates a group Horn formula  $\mathcal{H}_G = \{\mathcal{G}_0, \mathcal{G}_1, \dots, \mathcal{G}_{|\mathcal{T}|}\}$ , where  $\mathcal{G}_0 = \mathcal{H} \cup \{(\neg s_{[C \sqsubseteq D]})\}$ , added with a negated goal ( $C \sqsubseteq D$ ) and each group  $\mathcal{G}_i = \{(s_{[\alpha_i]})\}$  is defined using a unit clause labeling every axiom in normalized TBox  $\mathcal{T}_N$ . The tool also simplifies  $\mathcal{H}_G$  by exploiting simplification techniques like Cone-Of-Influence (C.O.I) modularization technique, as suggested in [SV09, SV15], which filters out irrelevant groups in  $\mathcal{H}_G$ . The group

### CHAPTER 3. AXIOM PINPOINTING VIA ENUMERATION OF MCSES/MUSES

Horn formula  $\mathcal{H}_G$  is unsatisfiable, as shown in [AMMS15a, AMMS15b], and its group-MUSES correspond to the MinAs for  $C \sqsubseteq_{\tau} D$  due to the hitting set duality for MinAs or diagnoses, which also holds for MUSES or MCSes, group-MCSes of  $\mathcal{H}_G$  correspond to diagnoses for  $C \sqsubseteq_{\tau} D$ .

A minimum hitting set corresponds to the smallest minimal unsatisfiable subformula (SMUS) and thus represents the smallest MinA in an inconsistent ontology. For computing smallest MinA, BEACON integrates the state-of-the-art solver for computing the smallest MUS problem (SMUS) called FORQES [IPLM15]. FORQES is based on the hitting set dualization between group MUSES and MCSes of a group CNF formula and, hence shares the ideas explored in MaxHS [DB11], MinUC [IJM13], EMUS/MARCO [PM13, LM13], among others. The algorithm iteratively computes minimum hitting sets of a set of MCSes of a formula and satisfiable minimum hitting sets are grown into an MSS, whose complement is an MCS which is added to the set of MCSes. The process terminates when an unsatisfiable minimum hitting set is identified, representing a smallest MUS of the formula. As described earlier, the decision version of the SMUS problem is known to be  $\Sigma_2^P$ -complete [Lib05, Gup06].

#### 3.2.4 Complexity Results

Enumeration algorithms whose running time is polynomial in the size of the input and the size of the output are called output polynomial time algorithms. The existence of an output polynomial algorithm for the enumeration problems is a natural parameter for measuring the time complexity of an enumeration algorithm. For various enumeration problems it has been shown that no output

### CHAPTER 3. AXIOM PINPOINTING VIA ENUMERATION OF MCSES/MUSES

polynomial time algorithm exists unless  $P = NP$  [KBB<sup>+</sup>08, KBEG08, LLK80]. Moreover, an enumeration algorithm is polynomial delay if the maximum computation time between two outputs is polynomial in the input size. If we have polynomial delay, then we have output polynomial but not the vice-versa.

For the  $\mathcal{EL}$  family of DLs, we examine the complexity of computing all MinAs, in the worst-case it may have exponential many MinAs [BPS07], is not possible to solve MinA-ENUM in polynomial time.

**Problem:** Enumerating over MinAs (MinA-ENUM)

**Input:** An  $\mathcal{EL}$  ontology  $\mathcal{O}$  and a GCI  $C \sqsubseteq D$  such that  $\mathcal{O} \models C \sqsubseteq D$ .

**Output:** Find all MinAs for  $\mathcal{O}$  w.r.t.  $C \sqsubseteq D$ .

**Proposition 1.** *MinA-ENUM cannot be solved in output-polynomial time unless  $P \neq NP$  [PS09].*

It has been shown that recognizing the set of all MinAs is at least as hard as recognizing the set of all minimal transversals of a given hypergraph [PS09] and MinA-ENUM shares the complexity bounds with TRANS-HYP, a well-known open problem (see Section 2.3).

In SAT, for arbitrary CNF formulae, the enumeration of MUSes has some inefficiencies. As explained earlier that the existing approaches used explicitly or implicitly enumerate MCSes of the formula along with minimal hitting set dualization to compute the MUSes. Each computed MCS requires a logarithmic number of calls to a SAT oracle, but there can be exponentially many MCSes. Moreover, computing each MUS from the set of MCSes requires a logarithmic number of calls to a SAT oracle and again there can be exponentially

## CHAPTER 3. AXIOM PINPOINTING VIA ENUMERATION OF MCSES/MUSES

many MUSes. Therefore all existing approaches for enumerating MUSes are not output polynomial.

### 3.2.5 Experimental Assessment

We have evaluated the performance of BEACON<sup>3</sup>, EL2MUS<sup>4</sup>, EL2MCS<sup>5</sup> and HGMUS<sup>6</sup>, a back-end to EL2MUS and BEACON, against all existing state of the art axiom pinpointing tools for the Lightweight DL  $\mathcal{EL}^+$ , namely EL<sup>+</sup>SAT [SV09, SV15], CEL [BLS06], SATPin [MP15] and JUST [Lud14].

For experimental assessment, we have considered four medical medical ontologies. These ontologies are GALEN [RH97], NCI [SdCH<sup>+</sup>07], the Gene Ontology (GO) [ABB<sup>+</sup>00] and SNOMED CT (version 2009) [SCC97]. GALEN ontology is available in two variants, NOT-GALEN and FULL-GALEN. NOT-GALEN is a stripped-down version of GALEN that does not contain role inclusion axioms and FULL-GALEN has excluded the inverse roles. Among these ontologies GENE, GALEN and NCI are freely available at <http://lat.inf.tu-dresden.de/~meng/toyont.html>. The SNOMED CT ontology can be requested from IHTSDO under a nondisclosure license agreement.

BEACON and EL<sup>+</sup>SAT tools normalize, classify and encode above mentioned ontologies into Horn formulae using EL<sup>+</sup>SAT encoding scheme (see details [Section 2.2](#)). [Table 3.2](#) shows the encoded formulae, in terms of variables and clauses, and the CPU runtime (in seconds) from EL<sup>+</sup>SAT and BEACON.

---

<sup>3</sup> Available from <http://logos.ucd.ie/web/doku.php?id=beacon-tool>

<sup>4</sup> Available from <http://logos.ucd.ie/web/doku.php?id=el2mus>

<sup>5</sup> Available from <http://logos.ucd.ie/web/doku.php?id=el2mcs>

<sup>6</sup> Available from <http://logos.ucd.ie/web/doku.php?id=hgmus>

### CHAPTER 3. AXIOM PINPOINTING VIA ENUMERATION OF MCSES/MUSES

Table 3.2:  $\text{EL}^+$ SAT & BEACON Encodings

Ontology	$\text{EL}^+$ SAT			BEACON		
	Runtime(sec.)	#Variable	#Clauses	Runtime(sec.)	#Variables	#Clauses
GENE	0.53	121256	138764	1.29	119480	134006
NCI	71.43	2694204	3760264	46.35	2666383	3624246
NOT-GALEN	6.73	216924	264886	2.11	207852	265245
FULL-GALEN	1.63	291580	296025	3.85	291580	296025
SNOMED CT	286.31	13566946	36530903	196.01	12868265	30617030

BEACON exhibits a clear improvement both in terms of size and performance for encoding NCI, GENE, NOT-GALEN and SNOMED ontologies in to Horn formulae due to better structure and implementation. The exception to this case are GENE and FULL-GALEN ontologies that contain a lot more role inclusion axioms (e.g., FULL has 1014 role inclusion axioms) and where  $\text{EL}^+$ SAT performed better.

For each medical ontology (including both GALEN variants), we considered 100 instances and every instance or query represents the problem of explaining a particular concept subsumption entailed in aforementioned medical ontologies. We have grouped each set of 100 instances into 50 random instances and 50 sorted instances. The random instances were generated by selecting 50 random queries (concept subsumptions) from all the possible existing entailments in a classified ontology. These instances are relatively easier because it yields fewer MinAs. On the contrary, the sorted instances are difficult to solve because they have tend to have a large number of minimal explanations (MinAs). These instances are selected using an order defined on the frequency of variables occurrence in the implied clauses of a Horn formula that encodes the

### CHAPTER 3. AXIOM PINPOINTING VIA ENUMERATION OF MCSES/MUSES

classification of an ontology using SAT-based encoding scheme [SV09, SV15].

Moreover, the set of axioms (variables) which may be responsible for any given subsumption relation were transformed into a group-MUS enumeration problem where the original Horn formula forms group-0 and each axiom constitutes a group containing only a unit clause.

The simplification techniques are vital in axiom pinpointing, such as the so-called reachability-based modularization in the DL domain [BS08], because it allows the extraction of a quite small module or subset of an ontology in which all the MinAs and diagnosis are still preserved for a given subsumption relation. We have also considered the cone-of-influence (COI) to simplify the problem instances (generated Horn formulae) because it performs better than syntax-based reachability-based modularization, see details in [SV15]. The resulting Horn formulae are quite smaller and useful to evaluate the algorithms on instances that have large number of MUSes/MCSes. Similar techniques were also exploited in related work [BLS06, Lud14, MP15].

All the experiments were performed on a Linux cluster (2 GHz) and the algorithms were given a time limit of 3600s and a memory limit of 4 GB. We have committed to 3600s time limit because it is a reasonable amount of time for all the tools to complete their computation for given instances.

Figure 3.4 summarizes the results for BEACON, EL2MUS, EL2MCS, EL<sup>+</sup>SAT, SATPIN and CEL. Although, HGMUS is a common back-end engine in both BEACON and EL2MUS but the former tool has a slight disadvantage over the other in CPU running time especially in the case of a large size ontology (e.g., SNOMED). This is because BEACON is comput-

### CHAPTER 3. AXIOM PINPOINTING VIA ENUMERATION OF MCSES/MUSES

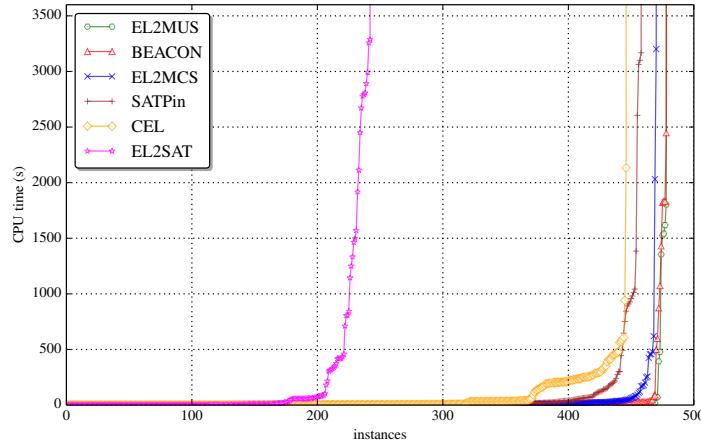


Figure 3.4: Cactus plot comparing  $\text{EL}^+\text{SAT}$ , SATPin, EL2MCS, EL2MUS and BEACON

ing simplification at runtime. The comparison with CEL imposes a number of constraints. Since CEL is limited to compute 10 MinAs; therefore, it has a CPU runtime advantage over all other tools. The tool JUST is excluded from this comparison because it only deals with less expressive Lightweight DL  $\mathcal{ELH}$ . The detailed comparison with CEL and JUST tools is presented in [AMMS15a, AMMS15b, AMI<sup>+</sup>16, AMM<sup>+</sup>16].

Table 3.3 summarizes the results for each family of problem instances and we can see that our proposed methods exhibit a clear improvement against all existing tools and techniques. HGMUS is a shared back-end for EL2MUS and BEACON tools; therefore, these comparison are related w.r.t to it. We have excluded CEL from the table because it is limited to compute only 10 MinAs and many of the considered instances have more than 10 MinAs. Moreover,

### CHAPTER 3. AXIOM PINPOINTING VIA ENUMERATION OF MCSES/MUSES

Table 3.3: SAT-based approaches. COI instances

Ontology	EL <sup>+</sup> SAT		SATPin		EL2MCS			HgMUS		
	#Sol.	#MinA	#Sol.	#MinA	#Sol.	#MinA	#Diag.	#Sol.	#MinA	#Diag.
GENE	87	1126	100	1134	100	1134	20011	100	1134	20011
NCI	77	664	93	661	100	678	78989	100	678	78989
NOT-GALEN	34	159	100	159	100	159	702	100	159	702
FULL-GALEN	17	137	100	139	100	139	776	100	139	776
SNOMED CT	27	1002	65	973	70	439	78839	78	9349	3199135
<b>Total</b>	<b>242</b>	<b>3088</b>	<b>458</b>	<b>3066</b>	<b>470</b>	<b>2549</b>	<b>179317</b>	<b>478</b>	<b>11459</b>	<b>3299613</b>

we have also noticed that some SNOMED CT instances cannot be solved by any tool because it contains a very large number of MinAs/MUSes and diagnoses/MCSes.

From experimental evaluation it can be clearly observed that our tools outperform the existing state of the art axiom pinpointing tools in all of the problem instances and, for many cases, with two or more orders of magnitude improvement.

## Chapter 4

### KI '15 Publication

- Citation ([DOI:10.1007/978-3-319-24489-1\\_17](https://doi.org/10.1007/978-3-319-24489-1_17)): M. Fareed Arif, Carlos Mencía, and Joao Marques-Silva. Efficient Axiom Pinpointing with EL2MCS. In *Proceedings of the 38th Annual German Conference on AI (KI 2015)*, volume 9324, pages 225–233. Springer, 2015
- It is a conference paper that discusses the candidate first effort to solve the problem of axiom pinpointing tool for Lightweight DL  $\mathcal{EL}^+$
- KI 2015 is a CORE C conference, but with attendance by many DL researchers, i.e. it was considered strategic to present the work at KI.
- Contribution: The candidate was responsible for the conception of the idea, design, implementation of EL2MCS tool and its experimental evaluation

# Efficient Axiom Pinpointing with EL2MCS

M. Fareed Arif<sup>1</sup>(✉), Carlos Mencía<sup>1</sup>, and Joao Marques-Silva<sup>1,2</sup>

<sup>1</sup> CASL, University College Dublin, Dublin, Ireland

[farif@ucdconnect.ie](mailto:farif@ucdconnect.ie), [{carlos.mencia,jpms}@ucd.ie](mailto:{carlos.mencia,jpms}@ucd.ie)

<sup>2</sup> INESC-ID, IST, ULisboa, Lisbon, Portugal

**Abstract.** Axiom pinpointing consists in computing a set-wise minimal set of axioms that explains the reason for a subsumption relation in an ontology. Recently, an encoding of the classification of an  $\mathcal{EL}^+$  ontology to a polynomial-size Horn propositional formula has been devised. This enables the development of a method for axiom pinpointing based on the analysis of unsatisfiable propositional formulas. Building on this earlier work, we propose a computation method, termed EL2MCS, that exploits an important relationship between minimal axiom sets and minimal unsatisfiable subformulas in the propositional domain. Experimental evaluation shows that EL2MCS achieves substantial performance gains over existing axiom pinpointing approaches for lightweight description logics.

## 1 Introduction

Axiom pinpointing consists in identifying a minimal set of axioms (MinA) that explains a given subsumption relation in an ontology. This problem is useful for debugging ontologies, and finds several application domains, including medical informatics [15, 20, 32]. Earlier axiom pinpointing algorithms [5, 6] in lightweight Description Logics (i.e.,  $\mathcal{EL}$  and  $\mathcal{EL}^+$ ) generate a (worst-case exponential-size) propositional formula and compute the MinAs by finding its minimal models, which is an NP-hard problem. More recently, a polynomial-size encoding is devised in [33, 34] that encodes the classification of an  $\mathcal{EL}^+$  ontology into a Horn propositional formula (i.e. it can be exponentially more compact than earlier work [5, 6]). This encoding is exploited by the axiom pinpointing algorithm EL<sup>+</sup>SAT [33, 34], based on SAT methods [25] and SMT-like techniques [17]. Although effective at computing MinAs, these dedicated algorithms often fail to enumerate MinAs to completion, or to prove that no additional MinA exists.

Building on this previous work, we present a new approach for axiom pinpointing in  $\mathcal{EL}^+$  DLs, termed EL2MCS. It is based on a relationship between MinAs and minimal unsatisfiable subformulas (MUSes) of the Horn formula encoding [33, 34]. The relationship between MUSes and MinAs makes it possible to benefit from the large recent body of work on extracting MUSes [8, 9, 14, 16, 24, 29], but also minimal correction subsets (MCSes), as well as their minimal hitting set relationship [7, 18, 31], which for the propositional case allows for exploiting the performance of modern SAT solvers. The relationship between

axiom pinpointing and MUS enumeration was also studied elsewhere independently [22], where the proposed approach iteratively computes implicants [12, 24] instead of exploiting hitting set dualization.

Experimental results, considering instances from medical ontologies, show that EL2MCS significantly outperforms existing approaches [4, 19, 22, 34].

The remainder of the paper is structured as follows. Section 2 introduces basic definitions and notation. Section 3 describes the propositional Horn encoding and our proposed axiom pinpointing approach. The experimental results are reported in Section 4 and Section 5 concludes the paper.

## 2 Preliminaries

### 2.1 Lightweight Description Logics

The standard definitions of  $\mathcal{EL}^+$  are assumed [3, 6, 33]. Starting from a set  $N_C$  of *concept names* and a set  $N_R$  of *role names*, every concept name in  $N_C$  is an  $\mathcal{EL}^+$  concept description that also uses  $\top$ ,  $\sqcap$ , and  $\exists r.C$  constructs to define concept descriptions and role chains  $r_1 \circ \dots \circ r_n$  from roles in  $N_R$ . A TBox is a finite set of *general concept inclusion (GCI)* of form  $C \sqsubseteq D$  and *role inclusion (RI)* axioms of form  $r_1 \circ \dots \circ r_n \sqsubseteq s$ . For a TBox  $\mathcal{T}$ ,  $PC_{\mathcal{T}}$  denotes the set of *primitive concepts* of  $\mathcal{T}$ , representing the smallest set of concepts that contains the top concept  $\top$ , and all the concept names in  $\mathcal{T}$ .  $PR_{\mathcal{T}}$  denotes the set of *primitive roles* of  $\mathcal{T}$ , representing all role names in  $\mathcal{T}$ . The main inference problem for  $\mathcal{EL}^+$  is concept subsumption [3, 6]:

**Definition 1 (Concept Subsumption).** Let  $C, D$  represent two  $\mathcal{EL}^+$  concept descriptions and let  $\mathcal{T}$  represent an  $\mathcal{EL}^+$  TBox.  $C$  is subsumed by  $D$  w.r.t.  $\mathcal{T}$  (denoted  $C \sqsubseteq_{\mathcal{T}} D$ ) iff  $C^{\mathcal{T}} \subseteq D^{\mathcal{T}}$  in every model  $\mathcal{I}$  of  $\mathcal{T}$ .

Finding an explanation, termed *axiom pinpointing*, consists of computing a minimal axiom subset (*MinA*) that explains the subsumption relation.

**Definition 2 (MinA).** Let  $\mathcal{T}$  be an  $\mathcal{EL}^+$  TBox, and let  $C, D \in PC_{\mathcal{T}}$  be primitive concept names, with  $C \sqsubseteq_{\mathcal{T}} D$ . Let  $S \subseteq \mathcal{T}$  be such that  $C \sqsubseteq_S D$ . If  $S$  is such that  $C \sqsubseteq_S D$  and  $C \not\sqsubseteq_{S'} D$  for  $S' \subsetneq S$ , then  $S$  is a minimal axiom set (*MinA*) w.r.t.  $C \sqsubseteq_{\mathcal{T}} D$ .

### 2.2 Propositional Satisfiability

Standard propositional satisfiability (SAT) definitions are assumed [10]. We consider propositional CNF formulas and use a clause-set based representation of such formulas. Formulas are represented by  $\mathcal{F}, \mathcal{M}, \mathcal{M}', \mathcal{C}$  and  $\mathcal{C}'$ , but also by  $\varphi$  and  $\phi$ . Horn formulas are such that every clause contains at most one positive literal. In this paper, we explore both MUSes and MCSes of CNF formulas.

**Definition 3 (MUS).**  $\mathcal{M} \subseteq \mathcal{F}$  is a Minimal Unsatisfiable Subformula (MUS) of  $\mathcal{F}$  iff  $\mathcal{M}$  is unsatisfiable and  $\forall_{\mathcal{M}' \subsetneq \mathcal{M}} \mathcal{M}'$  is satisfiable.

**Definition 4 (MCS).**  $\mathcal{C} \subseteq \mathcal{F}$  is a Minimal Correction Subformula (MCS) of  $\mathcal{F}$  iff  $\mathcal{F} \setminus \mathcal{C}$  is satisfiable and  $\forall_{\mathcal{C}' \subsetneq \mathcal{C}} \mathcal{F} \setminus \mathcal{C}'$  is unsatisfiable.

A well-known result, which will be used in the paper is the minimal hitting set relationship between MUSes and MCSes of an unsatisfiable formula  $\mathcal{F}$  [7, 11, 18, 31].

**Theorem 1.** Let  $\mathcal{F}$  be unsatisfiable. Then, each MCS of  $\mathcal{F}$  is a minimal hitting set of the MUSes of  $\mathcal{F}$  and each MUS of  $\mathcal{F}$  is a minimal hitting set of the MCSes of  $\mathcal{F}$ .

A partial MaxSAT formula  $\varphi$  is partitioned into a set of hard ( $\varphi_H$ ) and soft ( $\varphi_S$ ) clauses, i.e.  $\varphi = \{\varphi_H, \varphi_S\}$ . Hard clauses must be satisfied while soft clauses can be relaxed. We have used partial MaxSAT encoding and enumeration of MUSes [7, 18] using minimal hitting set duals [7, 11, 18, 31] in our proposed solution.

### 3 Computation Technique and Tool Overview (EL2MCS)

This section introduces the main organization of our approach. It works over the propositional Horn encoding used in EL<sup>+</sup>SAT [33, 34], and exploits a close relationship between MinAs and MUSes.

#### 3.1 Horn Formula Encoding

In EL<sup>+</sup>SAT, the Horn formula  $\phi_{\mathcal{T}(\text{po})}^{\text{all}}$  mimics the classification of TBox  $\mathcal{T}$  and is constructed as follows [33, 34]:

1. For every axiom (concretely  $ax_i$ ), create an axiom selector propositional variable  $s_{[ax_i]}$ . For trivial GCI of the form  $C \sqsubseteq C$  or  $C \sqsubseteq \top$ ,  $s_{[ax_i]}$  is constant true.
2. During the execution of the classification algorithm [3, 6], for every application of a rule (concretely  $r$ ) generating some assertion (concretely  $a_i$ ), add to  $\phi_{\mathcal{T}(\text{po})}^{\text{all}}$  a clause of the form,

$$\left( \bigwedge_{a_j \in \text{ant}(r)} s_{[a_j]} \right) \rightarrow s_{[a_i]}$$

where  $s_{[a_i]}$  is the selector variable for  $a_i$  and  $\text{ant}(r)$  are the antecedents of  $a_i$  with respect to a completion rule  $r$ .

For axiom pinpointing the SAT-based algorithms [33, 34], exploiting the ideas from early work on SAT solving [25] and AllSMT [17], compute MinAs for any subsumption relation (i.e.,  $C_i \sqsubseteq D_i$ ) using the list of assumption variables  $\{\neg s_{[C_i \sqsubseteq D_i]}\} \cup \{s_{[ax_i]} \mid ax_i \in \mathcal{T}\}$ . The following theorem is fundamental for this work [33, 34], and is extended in the next section to relate MinAs with MUSes of propositional formulas.

**Theorem 2 (Theorem 3 in [34]).** Given an  $\mathcal{EL}^+$  TBox  $\mathcal{T}$ , for every  $\mathcal{S} \subseteq \mathcal{T}$  and for every pair of concept names  $C, D \in \text{PC}_{\mathcal{T}}$ ,  $C \sqsubseteq_{\mathcal{S}} D$  if and only if the Horn propositional formula  $\phi_{\mathcal{T}(\text{po})}^{\text{all}} \wedge (\neg s_{[C \sqsubseteq D]}) \wedge_{ax_i \in \mathcal{S}} (s_{[ax_i]})$  is unsatisfiable.

### 3.2 MinAs as MUSes

Although not explicitly stated, the relation between axiom pinpointing and MUS extraction has been apparent in earlier work [6, 33, 34].

**Theorem 3 ([1]).** *Given an  $\mathcal{EL}^+$  TBox  $\mathcal{T}$ , for every  $\mathcal{S} \subseteq \mathcal{T}$  and for every pair of concept names  $C, D \in \text{PC}_{\mathcal{T}}$ ,  $\mathcal{S}$  is a **MinA** of  $C \sqsubseteq_{\mathcal{S}} D$  if and only if the Horn propositional formula  $\phi_{\mathcal{T}(\text{po})}^{\text{all}} \wedge (\neg s_{[C \sqsubseteq D]} \wedge_{ax_i \in \mathcal{S}} (s_{[ax_i]})$  is **minimally unsatisfiable**.*

Based on Theorem 3 and the MUS enumeration approach in [18], we can now outline our axiom pinpointing approach.

### 3.3 Axiom Pinpointing Using MaxSAT

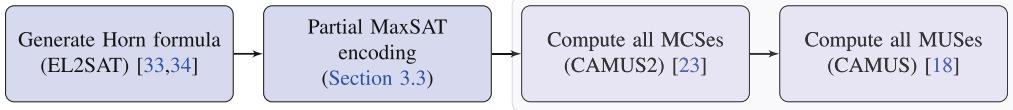
As described earlier, the axiom pinpointing algorithm [33, 34] explicitly enumerates the selection variables (i.e.,  $s_{[ax_i]}$ ) in an AllSMT-inspired approach [17]. In contrast, our approach is to model the problem as partial maximum satisfiability (MaxSAT), and enumerate over the MUSes of the MaxSAT problem formulation. Therefore, all clauses in  $\phi_{\mathcal{T}(\text{po})}^{\text{all}}$  are declared as hard clauses. Observe that, by construction,  $\phi_{\mathcal{T}(\text{po})}^{\text{all}}$  is satisfiable. In addition, the constraint  $C \sqsubseteq_{\mathcal{T}} D$  is encoded with another hard clause, namely  $(\neg s_{[C \sqsubseteq_{\mathcal{T}} D]})$ . Finally, the variable  $s_{[ax_i]}$  associated with each axiom  $ax_i$  denotes a *unit soft clause*. The intuitive justification is that the goal is to include as many axioms as possible, leaving out a minimal set which, if included, would cause the complete formula to be unsatisfiable. Thus, each of these sets represents an MCS of the MaxSAT problem formulation, but also a minimal set of axioms that needs to be dropped for the subsumption relation not to hold (i.e. a *diagnosis* [20]). MCS enumeration can be implemented with a MaxSAT solver [18, 27] or with a dedicated algorithm [23]. It is well-known (e.g. see Theorem 1) that MCSes are minimal hitting sets of MUSes, and MUSes are minimal hitting sets of MCSes [7, 11, 18, 31]. Thus, we use explicit minimal hitting set duality to obtain the MUSes we are looking for, starting from the previously computed MCSes.

### 3.4 EL2MCS Tool

The organization of the EL2MCS tool is shown in Figure 1. The first step is similar to  $\text{EL}^+\text{SAT}$  [33, 34] in that a propositional Horn formula is generated. The next step, however, exploits the ideas in Section 3.3, and generates a partial MaxSAT encoding. We can now enumerate the MCSes of the partial MaxSAT formula using the CAMUS2 tool [23]<sup>1</sup>. The final step is to exploit minimal hitting set duality for computing all the MUSes given the set of MCSes [18]. This is achieved with the CAMUS tool<sup>2</sup>. The hypergraph traversal computation tools, shd [28] and MTminer [13], could be used instead in this phase. It

<sup>1</sup> Available from <http://logos.ucd.ie/web/doku.php?id=mcses>

<sup>2</sup> Available from <http://sun.iwu.edu/~mliffito/camus/>



**Fig. 1.** The EL2MCS tool

should be observed that, although MCS enumeration uses CAMUS2 (a modern implementation of the MCS enumerator in CAMUS [18], capable of handling partial MaxSAT formulae), alternative MCS enumeration approaches were considered [23] but found not to be as efficient.

## 4 Experimental Evaluation

This section presents an empirical evaluation of EL2MCS<sup>3</sup>, which is compared to the state-of-the-art tools EL<sup>+</sup>SAT [34], JUST [19], CEL [4] and SATPin [22]. EL<sup>+</sup>SAT and SATPin are SAT-based approaches, whereas CEL and JUST use dedicated reasoners.

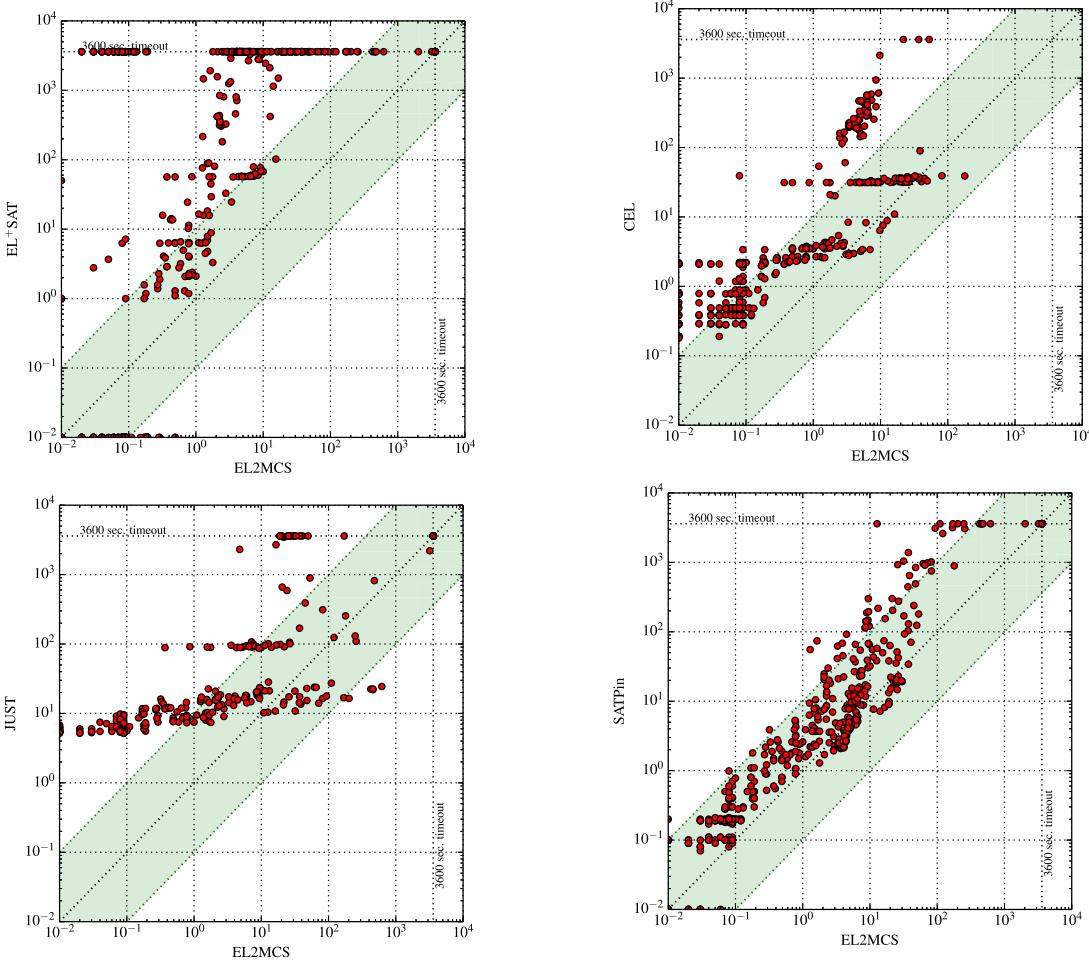
The medical ontologies used in the experiments are GALEN [30] (two variants: FULL-GALEN and NOT-GALEN), Gene [2], NCI [35] and SNOMED-CT [36]. As in earlier work [34], for each ontology 100 subsumption query instances were considered. So, there are 500 instances. In addition, for the SAT-based tools, including EL2MCS, the instances were simplified with the *cone-of-influence* (COI) reduction technique. CEL and JUST use their own similar simplification techniques. The comparison with CEL and JUST imposes additional constraints. CEL reports at most 10 MinAs, so only 397 instances with up to 10 MinAs were considered in the comparison with CEL. JUST is only able to handle a subset of  $\mathcal{EL}^+$ , so the comparison with JUST only considers 292 instances it can return correct results. The experiments were performed on a Linux Cluster (2GHz), with a time limit of 3600s.

By the time limit, out of the 500 instances, EL<sup>+</sup>SAT solves 241, SATPin solves 458 and EL2MCS solves 470. For the few instances EL2MCS does not solve, it computes thousands of MCSes by the time limit without reporting any MinA. In these cases, EL<sup>+</sup>SAT and SATPin are able to return some MinAs, although not achieving complete enumeration. Regarding the comparison with CEL, out of 397 instances, CEL solves 394 and EL2MCS solves all of them. Compared with JUST, out of the 292 instances considered, JUST solves 242 and EL2MCS solves 264. It is worth mentioning that there is no instance some tool is able to solve and EL2MCS is not. Table 1 compares EL2MCS with the other tools in terms of the number of instances it performed better and worse (wins/losses). Unsolved instances where some method computed some MinAs and EL2MCS did not, are counted as losses. As we can observe, in most cases, EL2MCS performs better than any other tool. Figure 2 shows four scatter plots with a pairwise comparison of EL2MCS and each other tool in terms of their

<sup>3</sup> Available from <http://logos.ucd.ie/web/doku.php?id=el2mcs>

**Table 1.** Summary of results comparing EL2MCS with  $\text{EL}^+\text{SAT}$ , SATPin, CEL and JUST.

	vs $\text{EL}^+\text{SAT}$	vs SATPin	vs CEL	vs JUST
#Wins / #Losses	359 / 106	353 / 114	379 / 18	236 / 28
%Wins / %Losses	71.8% / 21.2%	70.6% / 22.8%	96.2% / 4.5%	80.8% / 9.6%



**Fig. 2.** Plots comparing EL2MCS with  $\text{EL}^+\text{SAT}$ , CEL, JUST and SATPin (runtimes in secs).

running times. They reveal very significant differences in favor of EL2MCS in all cases. EL2MCS is remarkably faster than any other tool for most instances, in many cases with performance gaps of more than one order of magnitude. The greatest advantages are over  $\text{EL}^+\text{SAT}$ , CEL and JUST. On a few instances, JUST is faster than EL2MCS. However, note that JUST is only able to handle a subset of  $\mathcal{EL}^+$ , and so it is expected to be very efficient on this kind of instances. SATPin performs better than other alternatives, solving a few instances less than EL2MCS, but still EL2MCS outperforms it consistently in terms of running time.

## 5 Conclusions and Future Work

This paper presents the EL2MCS tool for axiom pinpointing of  $\mathcal{EL}^+$  ontologies. Building on previous work [33, 34], EL2MCS exploits a close relationship between MinAs and MUSes of propositional formulas, and instruments an efficient algorithm that relies on explicit minimal hitting set dualization of MCSes and MUSes of unsatisfiable formulas. Experimental results over well-known benchmarks from medical ontologies reveal that EL2MCS significantly outperforms the state of the art, thus constituting a very effective alternative for this problem. A natural research direction is to attempt to improve EL2MCS by substituting some of its parts by other advanced novel alternatives (e.g. MCS extraction and enumeration [21, 26]).

**Acknowledgment.** We thank R. Sebastiani and M. Vescovi, for authorizing the use of  $\text{EL}^+\text{SAT}$  [34]. We thank N. Manthey and R. Peñaloza, for bringing SAT-Pin [22] to our attention, and for allowing us to use their tool. This work is partially supported by SFI PI grant BEACON (09/IN.1/I2618), by FCT grant POLARIS (PTDC/EIA-CCO/123051/2010), and by national funds through FCT with reference UID/CEC/50021/2013.

## References

1. Arif, M.F., Marques-Silva, J.: Towards efficient axiom pinpointing of  $\mathcal{EL}^+$  ontologies. CoRR, abs/1503.08454 (2015). <http://arxiv.org/abs/1503.08454>
2. Ashburner, M., Ball, C.A., Blake, J.A., Botstein, D., Heather Butler, J., Cherry, M., Davis, A.P., Dolinski, K., Dwight, S.S., Eppig, J.T., et al.: Gene ontology: tool for the unification of biology. *Nature Genetics* **25**(1), 25–29 (2000)
3. Baader, F., Brandt, S., Lutz, C.: Pushing the  $\mathcal{EL}$  envelope. In: IJCAI, pp. 364–369 (2005)
4. Baader, F., Lutz, C., Suntisrivaraporn, B.: CEL — a polynomial-time reasoner for life science ontologies. In: Furbach, U., Shankar, N. (eds.) IJCAR 2006. LNCS (LNAI), vol. 4130, pp. 287–291. Springer, Heidelberg (2006)
5. Baader, F., Peñaloza, R., Suntisrivaraporn, B.: Pinpointing in the description logic  $\mathcal{EL}$ . In: DL (2007)
6. Baader, F., Peñaloza, R., Suntisrivaraporn, B.: Pinpointing in the description logic  $\mathcal{EL}^+$ . In: Hertzberg, J., Beetz, M., Englert, R. (eds.) KI 2007. LNCS (LNAI), vol. 4667, pp. 52–67. Springer, Heidelberg (2007)
7. Bailey, J., Stuckey, P.J.: Discovery of minimal unsatisfiable subsets of constraints using hitting set dualization. In: Hermenegildo, M.V., Cabeza, D. (eds.) PADL 2004. LNCS, vol. 3350, pp. 174–186. Springer, Heidelberg (2005)
8. Belov, A., Heule, M.J.H., Marques-Silva, J.: MUS extraction using clausal proofs. In: Sinz, C., Egly, U. (eds.) SAT 2014. LNCS, vol. 8561, pp. 48–57. Springer, Heidelberg (2014)
9. Belov, A., Lynce, I., Marques-Silva, J.: Towards efficient MUS extraction. *AI Commun.* **25**(2), 97–116 (2012)
10. Biere, A., Heule, M., van Maaren, H., Walsh, T. (eds.): *Handbook of Satisfiability*, vol. 185 (2009)

11. Birnbaum, E., Lozinskii, E.L.: Consistent subsets of inconsistent systems: structure and behaviour. *J. Exp. Theor. Artif. Intell.* **15**(1), 25–46 (2003)
12. Bradley, A.R., Manna, Z.: Checking safety by inductive generalization of counterexamples to induction. In: FMCAD, pp. 173–180 (2007)
13. Hébert, C., Bretto, A., Crémilleux, B.: A data mining formalization to improve hypergraph minimal transversal computation. *Fundamenta Informaticae* **80**(4), 415–433 (2007)
14. Junker, U.: QuickXplain: preferred explanations and relaxations for over-constrained problems. In: AAAI, pp. 167–172 (2004)
15. Kalyanpur, A., Parsia, B., Sirin, E., Cuenca-Grau, B.: Repairing unsatisfiable concepts in OWL ontologies. In: Sure, Y., Domingue, J. (eds.) ESWC 2006. LNCS, vol. 4011, pp. 170–184. Springer, Heidelberg (2006)
16. Lagniez, J.-M., Biere, A.: Factoring out assumptions to speed up MUS extraction. In: Järvisalo, M., Van Gelder, A. (eds.) SAT 2013. LNCS, vol. 7962, pp. 276–292. Springer, Heidelberg (2013)
17. Lahiri, S.K., Nieuwenhuis, R., Oliveras, A.: SMT techniques for fast predicate abstraction. In: Ball, T., Jones, R.B. (eds.) CAV 2006. LNCS, vol. 4144, pp. 424–437. Springer, Heidelberg (2006)
18. Liffiton, M.H., Sakallah, K.A.: Algorithms for computing minimal unsatisfiable subsets of constraints. *J. Autom. Reasoning* **40**(1), 1–33 (2008)
19. Ludwig, M.: Just: a tool for computing justifications w.r.t. EL ontologies. In: ORE 2014, vol. 1207, pp. 1–7 (2014)
20. Ludwig, M., Peñaloza, R.: Error-tolerant reasoning in the description logic  $\mathcal{EL}$ . In: Fermé, E., Leite, J. (eds.) JELIA 2014. LNCS, vol. 8761, pp. 107–121. Springer, Heidelberg (2014)
21. Malitsky, Y., O’Sullivan, B., Previti, A., Marques-Silva, J.: Timeout-sensitive portfolio approach to enumerating minimal correction subsets for satisfiability problems. In: ECAI, pp. 1065–1066 (2014)
22. Manthey, N., Peñaloza, R.: Exploiting SAT technology for axiom pinpointing. Technical Report LTCS 15–05, Chair of Automata Theory, Institute of Theoretical Computer Science, Technische Universität Dresden, April 2015. <https://ddll.inf.tu-dresden.de/web/Techreport3010>
23. Marques-Silva, J., Heras, F., Janota, M., Previti, A., Belov, A.: On computing minimal correction subsets. In: IJCAI (2013)
24. Marques-Silva, J., Janota, M., Belov, A.: Minimal sets over monotone predicates in boolean formulae. In: Sharygina, N., Veith, H. (eds.) CAV 2013. LNCS, vol. 8044, pp. 592–607. Springer, Heidelberg (2013)
25. Marques-Silva, J., Lynce, I., Malik, S.: Conflict-driven clause learning SAT solvers. In: Biere, et al. (eds.) [10], pp. 131–153
26. Mencía, C., Previti, A., Marques-Silva, J.: Literal-based MCS extraction. In: IJCAI (2015)
27. Morgado, A., Liffiton, M., Marques-Silva, J.: MaxSAT-based MCS enumeration. In: Biere, A., Nahir, A., Vos, T. (eds.) HVC. LNCS, vol. 7857, pp. 86–101. Springer, Heidelberg (2013)
28. Murakami, K., Uno, T.: Efficient algorithms for dualizing large-scale hypergraphs. CoRR, abs/1102.3813 (2011)
29. Nadel, A., Ryvchin, V., Strichman, O.: Efficient MUS extraction with resolution. In: FMCAD, pp. 197–200 (2013)
30. Rector, A.L., Horrocks, I.R.: Experience building a large, re-usable medical ontology using a description logic with transitivity and concept inclusions. In: Workshop on Ontological Engineering, pp. 414–418 (1997)

31. Reiter, R.: A theory of diagnosis from first principles. *Artif. Intell.* **32**(1), 57–95 (1987)
32. Schlobach, S., Huang, Z., Cornet, R., van Harmelen, F.: Debugging incoherent terminologies. *J. Autom. Reasoning* **39**(3), 317–349 (2007)
33. Sebastiani, R., Vescovi, M.: Axiom pinpointing in lightweight description logics via horn-SAT encoding and conflict analysis. In: Schmidt, R.A. (ed.) CADE-22. LNCS, vol. 5663, pp. 84–99. Springer, Heidelberg (2009)
34. Sebastiani, R., Vescovi, M.: Axiom pinpointing in large  $\mathcal{EL}^+$  ontologies via SAT and SMT techniques. Technical Report DISI-15-010, DISI, University of Trento, Italy, April 2015. <http://disi.unitn.it/rseba/elsat/elsat.techrep.pdf>
35. Sioutos, N., de Coronado, S., Haber, M.W., Hartel, F.W., Shaiu, W.-L., Wright, L.W.: NCI thesaurus: A semantic model integrating cancer-related clinical and molecular information. *Journal of Biomedical Informatics* **40**(1), 30–43 (2007)
36. Spackman, K.A., Campbell, K.E., Côté, R.A.: SNOMED RT: a reference terminology for health care. In: AMIA (1997)

## Chapter 5

### SAT '15 Publication

- Citation ([DOI:10.1007/978-3-319-24318-4\\_24](https://doi.org/10.1007/978-3-319-24318-4_24)): M. Fareed Arif, Carlos Mencía, and Joao Marques-Silva. Efficient MUS Enumeration of Horn Formulae with Applications to Axiom Pinpointing. In *Theory and Applications of Satisfiability Testing – SAT 2015 – 18th International Conference, Austin, TX, USA, September 24–27, 2015, Proceedings*, volume 9340, pages 324–342. Springer, 2015
- It is a conference paper that develops a novel MUS enumeration algorithm for group Horn formulae and its application. The work is originated from the detailed discussions the candidate have with his supervisor about improving the results presented in [[AMMS15a](#)].
- SAT '15 is a CORE A conference and published paper represents the bulk of the scientific contributions that candidate has made in his research.

## CHAPTER 5. SAT '15 PUBLICATION

- Contribution: The candidate was responsible for extending the idea presented earlier, the design, development and the experimental evaluation of EL2MUS.

# Efficient MUS Enumeration of Horn Formulae with Applications to Axiom Pinpointing

M. Fareed Arif<sup>1</sup>, Carlos Mencía<sup>1(✉)</sup>, and Joao Marques-Silva<sup>1,2</sup>

<sup>1</sup> CASL, University College Dublin, Dublin, Ireland

[farif@ucdconnect.ie](mailto:farif@ucdconnect.ie), [{carlos.mencia,jpms}@ucd.ie](mailto:{carlos.mencia,jpms}@ucd.ie)

<sup>2</sup> INESC-ID, IST, ULisboa, Lisbon, Portugal

**Abstract.** The enumeration of minimal unsatisfiable subsets (MUSes) finds a growing number of practical applications, that includes a wide range of diagnosis problems. As a concrete example, the problem of axiom pinpointing in the  $\mathcal{EL}$  family of description logics (DLs) can be modeled as the enumeration of the group-MUSes of Horn formulae. In turn, axiom pinpointing for the  $\mathcal{EL}$  family of DLs finds important applications, such as debugging medical ontologies, of which SNOMED CT is the best known example. The main contribution of this paper is to develop an efficient group-MUS enumerator for Horn formulae, HGMUS, that finds immediate application in axiom pinpointing for the  $\mathcal{EL}$  family of DLs. In the process of developing HGMUS, the paper also identifies performance bottlenecks of existing solutions. The new algorithm is shown to outperform all alternative approaches when the problem domain targeted by group-MUS enumeration of Horn formulae is axiom pinpointing for the  $\mathcal{EL}$  family of DLs, with a representative suite of examples taken from different medical ontologies.

## 1 Introduction

Description Logics (DLs) are well-known knowledge representation formalisms [4]. DLs find a wide range of applications in computer science, including the semantic web and representation of ontologies, but also in medical bioinformatics.

Given an ontology (that consists of a set of axioms) and a subsumption relation entailed by the ontology, *axiom pinpointing* is the problem of finding minimal axiom sets (MinAs), equivalently minimal sub-ontologies, each one entailing the given subsumption relation [48]. So, each MinA represents a minimal explanation or justification for the subsumption relation. Example applications of axiom pinpointing include context-based reasoning, error-tolerant reasoning [32], and ontology debugging and revision [26, 49]. Axiom pinpointing for different description logics (DLs) has been studied extensively for more than a decade, with related work in the mid 90s [1, 3, 5–8, 25, 31, 33, 37, 39, 40, 42, 48–51, 53].

The  $\mathcal{EL}$  family of DLs is well-known for being tractable (i.e. polynomial-time decidable). Despite being expressive, the  $\mathcal{EL}$  family of DLs, concretely by using the more expressive, and still tractable,  $\mathcal{EL}^+$ , has been used for representing ontologies in the medical sciences, including the well-known SNOMED CT

ontology [55]. Work on axiom pinpointing for the  $\mathcal{EL}$  family of DLs can be traced to 2006, namely the CEL tool [5]. Later, in 2009, the use of SAT was proposed for axiom pinpointing in the  $\mathcal{EL}$  family of DLs [50, 51, 56], concretely for the more expressive DL  $\mathcal{EL}^+$ . This seminal work proposed a propositional Horn encoding that can be exponentially smaller than earlier work [5, 7, 8]. Moreover, the use of SAT for axiom pinpointing for the  $\mathcal{EL}$  family of DLs, named EL<sup>+</sup>SAT [50, 51, 56], was shown to consistently outperform earlier work, concretely CEL [5]. Recent work [1] builds on these propositional encodings, but exploits the relationship between axiom pinpointing and enumeration of minimal unsatisfiable subsets (MUSes) [30], achieving conclusive performance gains over earlier work.

Nevertheless, this recent work has a number of potential drawbacks that will be analyzed later in the paper.

The relationship between axiom pinpointing and MUS enumeration was also studied elsewhere [33]. Instead of exploiting hitting set dualization, this alternative approach exploits the enumeration of implicants [33].

The main contribution of this paper is to develop an efficient group-MUS enumerator for Horn formulae, referred to as HGMUS, that finds immediate application in axiom pinpointing for the  $\mathcal{EL}$  family of DLs. In the process of developing HGMUS, the paper also identifies performance bottlenecks of existing solutions, in particular EL<sup>+</sup>SAT [50, 51]. The new group-MUS enumerator for Horn formulae builds on the large body of recent work on problem solving with SAT oracles. This includes, among others, MUS extraction [12], MCS extraction and enumeration [34], and partial MUS enumeration [28, 29, 44]. HGMUS also exploits earlier work on solving Horn propositional formulae [17, 38], and develops novel algorithms for MUS extraction in propositional Horn formulae. The experimental results, using well-known problem instances, demonstrate conclusive performance improvements over all other existing approaches, in most cases by several orders of magnitude.

The paper is organized as follows. Section 2 introduces the notation and definitions used throughout the paper. Section 3 reviews recent work on MUS enumeration, which serves as the basis for HGMUS. Afterwards, the new group-MUS enumerator HGMUS is described in Section 4. Section 5 compares HGMUS with existing alternatives. Experimental results on well-known problem instances from axiom pinpointing for the  $\mathcal{EL}$  family of DLs are analyzed in Section 6. The paper concludes in Section 7.

## 2 Preliminaries

Standard definitions of propositional logic are assumed [13]. This paper considers Boolean formulae in Conjunctive Normal Form (CNF). A CNF formula  $\mathcal{F}$  is defined over a set of Boolean variables  $V(\mathcal{F}) = \{x_1, \dots, x_n\}$  as a conjunction of clauses ( $c_1 \wedge \dots \wedge c_m$ ). A clause  $c$  is a disjunction of literals ( $l_1 \vee \dots \vee l_k$ ) and a literal  $l$  is either a variable  $x$  or its negation  $\neg x$ . We refer to the set of literals appearing in  $\mathcal{F}$  as  $L(\mathcal{F})$ . Formulae can also be represented as sets of clauses, and clauses as sets of literals.

A truth assignment, or interpretation, is a mapping  $\mu : V(\mathcal{F}) \rightarrow \{0, 1\}$ . If all the variables in  $V(\mathcal{F})$  are assigned a truth value,  $\mu$  is referred to as a *complete* assignment. Interpretations can also be seen as conjunctions or sets of literals. Truth valuations are lifted to clauses and formulae as follows:  $\mu$  satisfies a clause  $c$  if it contains at least one of its literals. Given a formula  $\mathcal{F}$ ,  $\mu$  satisfies  $\mathcal{F}$  (written  $\mu \models \mathcal{F}$ ) if it satisfies all its clauses, being  $\mu$  referred to as a *model* of  $\mathcal{F}$ .

Given two formulae  $\mathcal{F}$  and  $\mathcal{G}$ ,  $\mathcal{F}$  entails  $\mathcal{G}$  (written  $\mathcal{F} \models \mathcal{G}$ ) iff all the models of  $\mathcal{F}$  are also models of  $\mathcal{G}$ .  $\mathcal{F}$  and  $\mathcal{G}$  are equivalent (written  $\mathcal{F} \equiv \mathcal{G}$ ) iff  $\mathcal{F} \models \mathcal{G}$  and  $\mathcal{G} \models \mathcal{F}$ .

A formula  $\mathcal{F}$  is satisfiable ( $\mathcal{F} \not\models \perp$ ) if there exists a model for it. Otherwise it is unsatisfiable ( $\mathcal{F} \models \perp$ ). SAT is the decision problem of determining the satisfiability of a propositional formula. This problem is in general NP-complete [15].

Some applications require computing certain types of models. In this paper, we will make use of maximal models, i.e. models such that a set-wise maximal subset of the variables are assigned value 1:

**Definition 1 (MxM).** Let  $\mathcal{F}$  be a satisfiable propositional formula,  $\mu \models \mathcal{F}$  a model of  $\mathcal{F}$  and  $P \subseteq V(\mathcal{F})$  the set of variables appearing in  $\mu$  with positive polarity.  $\mu$  is a maximal model (MxM) of  $\mathcal{F}$  iff  $\mathcal{F} \cup P \not\models \perp$  and for all  $v \in V(\mathcal{F}) \setminus P$ ,  $\mathcal{F} \cup P \cup \{v\} \models \perp$ .

Herein, we will denote a maximal model by  $P$ , i.e. the set of its positive literals.

Horn formulae constitute an important subclass of propositional logic. These are composed of Horn clauses, which have at most one positive literal. Satisfiability of Horn formulae is decidable in polynomial time [17, 23, 38].

Given an unsatisfiable formula  $\mathcal{F}$ , the following subsets represent different notions regarding (set-wise) minimal unsatisfiability and maximal satisfiability [30, 34]:

**Definition 2 (MUS).**  $\mathcal{M} \subseteq \mathcal{F}$  is a Minimally Unsatisfiable Subset (MUS) of  $\mathcal{F}$  iff  $\mathcal{M}$  is unsatisfiable and  $\forall c \in \mathcal{M}, \mathcal{M} \setminus \{c\}$  is satisfiable.

**Definition 3 (MCS).**  $\mathcal{C} \subseteq \mathcal{F}$  is a Minimal Correction Subset (MCS) iff  $\mathcal{F} \setminus \mathcal{C}$  is satisfiable and  $\forall c \in \mathcal{C}, \mathcal{F} \setminus (\mathcal{C} \setminus \{c\})$  is unsatisfiable.

**Definition 4 (MSS).**  $\mathcal{S} \subseteq \mathcal{F}$  is a Maximal Satisfiable Subset (MSS) iff  $\mathcal{S}$  is satisfiable and  $\forall c \in \mathcal{F} \setminus \mathcal{S}, \mathcal{S} \cup \{c\}$  is unsatisfiable.

An MSS is the complement of an MCS. MUSes and MCSes are closely related by the well-known hitting set duality [10, 14, 46, 54]: Every MCS (MUS) is an irreducible hitting set of all MUSes (MCSes) of  $\mathcal{F}$ . In the worst case, there can be an exponential number of MUSes and MCSes [30, 41]. Besides, MCSes are related to the MaxSAT problem, which consists in finding an assignment satisfying as many clauses as possible. The smallest MCS (largest MSS) represents an optimal solution to MaxSAT.

Motivated by several applications, MUSes and related concepts have been extended to CNF formulae where clauses are partitioned into disjoint sets called groups [30].

**Definition 5 (Group-Oriented MUS).** *Given an explicitly partitioned unsatisfiable CNF formula  $\mathcal{F} = \mathcal{G}_0 \cup \dots \cup \mathcal{G}_k$ , a group-oriented MUS (or group-MUS) of  $\mathcal{F}$  is a set of groups  $\mathcal{G} \subseteq \{\mathcal{G}_1, \dots, \mathcal{G}_k\}$ , such that  $\mathcal{G}_0 \cup \mathcal{G}$  is unsatisfiable, and for every  $\mathcal{G}_i \in \mathcal{G}$ ,  $\mathcal{G}_0 \cup (\mathcal{G} \setminus \mathcal{G}_i)$  is satisfiable.*

Note the special role  $\mathcal{G}_0$  (*group-0*); this group consists of *background* clauses that are included in every group-MUS. Because of  $\mathcal{G}_0$  a group-MUS, as opposed to MUS, can be empty. Nevertheless, in this paper we assume that  $\mathcal{G}_0$  is satisfiable.

Equivalently, the related concepts of group-MCS and group-MSS can be defined in the same way. We omit these definitions here due to lack of space. In the case of MaxSAT, the use of groups is investigated in detail in [22].

### 3 MUS Enumeration in Horn Formulae

Enumeration of MUSes has been the subject of research that can be traced to the seminal work of Reiter [46]. A well-known family of algorithms uses (explicit) minimal hitting set dualization [10, 14, 30]. The organization of these algorithms can be summarized as follows. First compute all the MCSes of a CNF formula. Second, MUSes are obtained by computing the minimal hitting sets of the set of MCSes. The main drawback of explicit minimal hitting set dualization is that, if the number of MCSes is exponentially large, these approaches will be unable to compute MUSes, even if the total number of MUSes is small. As a result, recent work considered what can be described as implicit minimal hitting set dualization [28, 29, 44]. In these approaches (namely EMUS [44] and MARCO [29] MUS enumerators), either an MUS or an MCS is computed at each step of the algorithm, with the guarantee that one or more MUSes will be computed at the outset. In some settings, implicit minimal hitting set dualization is the only solution for finding some MUSes of a CNF formula. As pointed out in this recent work, implicit minimal hitting set dualization aims to complement, but not replace, the explicit dualization alternative, and in some settings where enumeration of MCSes is feasible, the latter may be the preferred option [29, 44].

Algorithm 1 shows the EMUS enumeration algorithm [44], also used in the most recent version of MARCO [29]. It relies on a two-solver approach aimed at enumerating the MUSes/MCSes of an unsatisfiable formula  $\mathcal{F}$ . On the one hand, a formula  $\mathcal{Q}$  is used to enumerate subsets of  $\mathcal{F}$ . This formula is defined over a set of variables  $I = \{p_i \mid c_i \in \mathcal{F}\}$ , each one of them associated with one clause  $c_i \in \mathcal{F}$ . Iteratively until  $\mathcal{Q}$  becomes unsatisfiable, EMUS computes a maximal model  $P$  of  $\mathcal{Q}$  and tests the satisfiability of the corresponding subformula  $\mathcal{F}' \subseteq \mathcal{F}$ . If it is satisfiable,  $\mathcal{F}'$  represents an MSS of  $\mathcal{F}$ , and the clause  $I \setminus P$  is added to  $\mathcal{Q}$ , preventing the algorithm from generating any subset of the MSS (superset of the MCS) again. Otherwise, if  $\mathcal{F}'$  is unsatisfiable, it is reduced to an MUS  $\mathcal{M}$ , which is blocked adding to  $\mathcal{Q}$  a clause made of the variables in  $I$  associated with  $\mathcal{M}$  with negative polarity. This way, no superset of  $\mathcal{M}$  will be generated. Algorithm 1 is guaranteed to find all MUSes and MCSes of  $\mathcal{F}$ , in a number of iterations that corresponds to the sum of the number of MUSes and MCSes.

This paper considers the problem of enumerating the group-MUSes of an unsatisfiable Horn formula. As highlighted earlier, and as discussed later in the

---

**Algorithm 1.** EMUS [44] / MARCO [29]

---

**Input:**  $\mathcal{F}$  a CNF formula  
**Output:** Reports the set of MUSes of  $\mathcal{F}$

```

1  $I \leftarrow \{p_i \mid c_i \in \mathcal{F}\}$                                 // Variable  $p_i$  picks clause  $c_i$ 
2  $\mathcal{Q} \leftarrow \emptyset$ 
3 while true do
4    $(st, P) \leftarrow \text{MaximalModel}(\mathcal{Q})$ 
5   if not  $st$  then return
6    $\mathcal{F}' \leftarrow \{c_i \mid p_i \in P\}$                                 // Pick selected clauses
7   if not  $\text{SAT}(\mathcal{F}')$  then
8      $\mathcal{M} \leftarrow \text{ComputeMUS}(\mathcal{F}')$ 
9      $\text{ReportMUS}(\mathcal{M})$ 
10     $b \leftarrow \{\neg p_i \mid c_i \in \mathcal{M}\}$                                 // Negative clause blocking the MUS
11  else
12     $b \leftarrow \{p_i \mid p_i \in I \setminus P\}$                                 // Positive clause blocking the MCS
13   $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{b\}$ 

```

---

paper, enumeration of the group-MUSes of unsatisfiable Horn formulae finds important applications in axiom pinpointing for the  $\mathcal{EL}$  family of DLs, including  $\mathcal{EL}^+$ . It should be observed that the difference between the enumeration of plain MUSes of Horn formulae and the enumeration of group-MUSes is significant. First, enumeration of group-MUSes of Horn formulae cannot be achieved in total polynomial time, unless  $P = NP$ . This is an immediate consequence from the fact that axiom pinpointing for the  $\mathcal{EL}$  family of DLs cannot be achieved in total polynomial time, unless  $P = NP$  [7], and that axiom pinpointing for the  $\mathcal{EL}$  family of DLs can be reduced in polynomial time to group-MUS enumeration of Horn formulae [1]. Second, enumeration of MUSes of Horn formulae can be achieved in total polynomial time (actually with polynomial delay) [43].

Given the above, a possible approach for enumerating group-MUSes of Horn formulae is to use an existing solution, either based on explicit or implicit minimal hitting set dualization. For example, the use of explicit minimal hitting dualization was recently proposed in EL2MCS [1]. Alternatively, either EMUS [44] or the different versions of MARCO [28, 29] could be used, as also pointed out in [33].

This paper opts instead to exploit the implicit minimal hitting set dualization approach [28, 29, 44], but develops a solution that is specific to the problem formulation. This solution is described in the next section.

## 4 Algorithm for Group-MUS Enumeration in Horn Formulae

This section describes HGMUS, a novel and efficient group-MUS enumerator for Horn formulae based on implicit minimal hitting set dualization. In this section,  $\mathcal{H}$  denotes the group of clauses  $\mathcal{G}_0$ , i.e. the background clauses. Moreover,  $\mathcal{I}$  denotes the set of (individual) groups of clauses, with  $\mathcal{I} = \{\mathcal{G}_1, \dots, \mathcal{G}_k\}$ . So, the unsatisfiable group-Horn formula corresponds to  $\mathcal{F} = \mathcal{H} \cup \mathcal{I}$ . Also, in this

**Algorithm 2.** Computation of Maximal Models

---

**Input:**  $\mathcal{Q}$  a CNF formula  
**Output:**  $(st, P)$ : with  $st$  a Boolean and  $P$  an MxM (if it exists)

- 1  $(P, U, B) \leftarrow (\{\{x\} \mid \neg x \notin L(\mathcal{Q})\}, \{\{x\} \mid \neg x \in L(\mathcal{Q})\}, \emptyset)$
- 2  $(st, P, U) \leftarrow \text{InitialAssignment}(\mathcal{Q} \cup P)$
- 3 **if not**  $st$  **then return**  $(\text{false}, \emptyset)$
- 4 **while**  $U \neq \emptyset$  **do**
- 5    $l \leftarrow \text{SelectLiteral}(U)$
- 6    $(st, \mu) = \text{SAT}(\mathcal{Q} \cup P \cup B \cup \{l\})$
- 7   **if**  $st$  **then**  $(P, U) \leftarrow \text{UpdateSatClauses}(\mu, P, U)$
- 8   **else**  $(U, B) \leftarrow (U \setminus \{l\}, B \cup \{\neg l\})$
- 9 **return**  $(\text{true}, P)$  *// P is an MxM of  $\mathcal{Q}$*

---

section, the formula  $\mathcal{Q}$  shown in [Algorithm 1](#) is defined on a set of variables associated to the groups in  $\mathcal{I}$ . For the problem instances considered later in the paper (obtained from axiom pinpointing for the  $\mathcal{EL}$  family of DLs), each group of clauses contains a single unit clause. However, the algorithm would work for arbitrary groups of clauses.

#### 4.1 Organization

The high-level organization of HGMUS mimics that of EMUS/MARCO (see [Algorithm 1](#)), with a few essential differences. First, the satisfiability testing step (because it operates on Horn formulae) uses the dedicated linear time algorithm LTUR [38]. LTUR can be viewed as one-sided unit propagation, since only variables assigned value 1 are propagated. Moreover, the simplicity of LTUR enables very efficient implementations, that use adjacency lists for representing clauses instead of the now more commonly used watched literals. Second, the problem formulation motivates using a dedicated MUS extraction algorithm, which is shown to be more effective in this concrete case than other well-known approaches [12]. Third, we also highlight important aspects of the EMUS/MARCO implicit minimal hitting set dualization approach, which we claim have been overlooked in earlier work [51, 56].

#### 4.2 Computing Maximal Models

The use of maximal models for computing either MCSes of a formula or a set of clauses that contain an MUS was proposed in earlier work [44], which exploited SAT with preferences for computing maximal models [20, 47]. The use of SAT with preferences for computing maximal models is also exploited in related work [50, 51].

Computing maximal models of a formula  $\mathcal{Q}$  can be reduced to the problem of extracting an MSS of a formula  $\mathcal{Q}'$  [34], where the clauses of  $\mathcal{Q}$  are hard and, for each variable  $x_i \in V(\mathcal{Q})$ , it includes a unit soft clause  $c_i \equiv \{x_i\}$ . Also, recent work [9, 21, 34, 36] has shown that state-of-the-art MCS/MSS computation approaches outperform SAT with preferences. HGMUS uses a dedicated algorithm based on the LinearSearch MCS extraction algorithm [34], due to its good performance in MCS enumeration. Since all soft clauses are unit, it can also be related with

the novel Literal-Based eXtractor algorithm [36]. Shown in [Algorithm 2](#), it relies on making successive calls to a SAT solver. It maintains three sets of literals:  $P$ , an under-approximation of an MxM (i.e. positive literals s.t.  $\mathcal{Q} \cup P \not\models \perp$ ),  $B$ , with negative literals  $\neg l$  such that  $\mathcal{Q} \cup P \cup \{l\} \models \perp$  (i.e. *backbone literals*), and  $U$ , with the remaining set of positive literals to be tested. Initially,  $P$  and  $U$  are initialized from a model  $\mu \models \mathcal{Q}$ ,  $P$  ( $U$ ) including the literals appearing with positive (negative) polarity in  $\mu$ . Then, iteratively, it tries to extend  $P$  with a new literal  $l \in U$ , by testing the satisfiability of  $\mathcal{Q} \cup P \cup B \cup \{l\}$ . If it is satisfiable, all the literals in  $U$  satisfied by the model (including  $l$ ) are moved to  $P$ . Otherwise,  $l$  is removed from  $U$  and  $\neg l$  is added to  $B$ . This algorithm has a query complexity of  $\mathcal{O}(|V(\mathcal{Q})|)$ .

[Algorithm 2](#) integrates a new technique, which consists in pre-initializing  $P$  with the pure positive literals appearing in  $\mathcal{Q}$  and  $U$  with the remaining ones (line 1), and then requiring the literals of  $P$  to be satisfied by the initial assignment (line 2). It can be easily proved that these pure literals are included in all MxMs of  $\mathcal{Q}$ , so a number of calls to the SAT solver could be avoided. Moreover, the SAT solver will never branch on these variables, easing the decision problems. This technique is expected to be effective in HGMUS. Note that, in this context,  $\mathcal{Q}$  is made of two types of clauses: positive clauses blocking MCSes of the Horn formula, and negative clauses blocking MUSes. So, with this technique, the computation of MxMs is restricted to the variables representing groups appearing in some MUS of the Horn formula.<sup>1</sup>

### 4.3 Adding Blocking Clauses

One important aspect of HGMUS are the blocking clauses created and added to the formula  $\mathcal{Q}$  (see [Algorithm 1](#)). These follow what was first proposed in EMUS [44] and MARCO [28, 29]. For each MUS, the blocking clause consists of a set of negative literals, requiring at least one of the clauses in the MUS *not* to be included in future selected sets of clauses. For each MCS, the blocking clause consists of a set of positive literals, requiring at least one of the clauses in the MCS to be included in future selected sets of clauses. The way MCSes are handled is essential to prevent that MCS and sets containing the same MCS to be selected again. Although conceptually simple, it can be shown that existing approaches may not guarantee that supersets of MCSes (or subsets of the MSSes) are not selected. As argued later, this is the case with EL<sup>+</sup>SAT [51, 56].

### 4.4 Deciding Satisfiability of Horn Formulae

It is well-known that Horn formulae can be decided in linear time [17, 23, 38]. HGMUS implements the LTUR algorithm [38]. There are important reasons for this choice. First, LTUR is expected to be more efficient than plain unit propagation, since only variables assigned value 1 need to be propagated. Second, most implementations of unit propagation in CDCL SAT solvers (i.e. that use watched literals) are not guaranteed to run in linear time [19]; this is for example the case with *all* implementations of Minisat [18] and its variants, for which unit

---

<sup>1</sup> SATPin [33] also exploits this insight of *relevant* variables, but not in the contexts of MxMs.

**Algorithm 3.** Insertion-based [16] MUS extraction using LTUR [38]

---

**Input:**  $\mathcal{H}$ , denotes the  $\mathcal{G}_0$  clauses;  $\mathcal{I}$ , denotes the set of (individual) group clauses

**Output:**  $\mathcal{M}$ , denotes the computed MUS

```

1  $(\mathcal{M}, c_r) \leftarrow (\mathcal{H}, 0)$ 
2  $\text{LTUR\_prop}(\mathcal{M}, \mathcal{M})$  // Start by propagating  $\mathcal{G}_0$  clauses
3 while true do
4   if  $c_r > 0$  then
5      $\mathcal{M} \leftarrow \mathcal{M} \cup \{c_r\}$  // Add transition clause  $c_r$  to  $\mathcal{M}$ 
6     if not  $\text{LTUR\_prop}(\mathcal{M}, \{c_r\})$  then
7        $\text{LTUR\_undo}(\mathcal{M}, \mathcal{M})$ 
8       return  $\mathcal{M} \setminus \mathcal{H}$  // Remove  $\mathcal{G}_0$  clauses from computed MUS
9    $\mathcal{S} \leftarrow \emptyset$ 
10  while true do
11     $c_r \leftarrow \text{SelectRemoveClause}(\mathcal{I})$  // Target transition clause
12     $\mathcal{S} \leftarrow \mathcal{S} \cup \{c_r\}$ 
13    if not  $\text{LTUR\_prop}(\mathcal{M} \cup \mathcal{S}, \{c_r\})$  then
14       $\mathcal{I} \leftarrow \mathcal{S} \setminus \{c_r\}$  // Update working set of groups
15       $\text{LTUR\_undo}(\mathcal{M}, \mathcal{S})$ 
16      break //  $c_r$  represents a transition clause

```

---

propagation runs in worst-case quadratic time. As a result, using an off-the-shelf SAT solver and exploiting only unit propagation (as is done for example in earlier work [33, 50, 51]) is unlikely to be the most efficient solution. Besides the advantages listed above, the use of a linear time algorithm for deciding the satisfiability of Horn formulae turns out to be instrumental for MUS extraction, as shown in the next section. In order to use LTUR for MUS extraction, an incremental version has been implemented, which allows for the incremental addition of clauses to the formula and incremental identification of variables assigned value 1. Clearly, the amortized run time of LTUR, after adding  $m = |\mathcal{F}|$  clauses, is  $\mathcal{O}(|\mathcal{F}|)$ , with  $|\mathcal{F}|$  the number of literals appearing in  $\mathcal{F}$ .

#### 4.5 MUS Extraction in Horn Formulae

For arbitrary CNF formulae, a number of approaches exist for MUS extraction, with the most commonly used one being the deletion-based approach [11, 12], but other alternatives include the QuickXplain algorithm [24] and the more recent Progression algorithm [35]. It is also well-known and generally accepted that, due to its query complexity, the insertion-based algorithm [16] for MUS extraction is in practice not competitive with existing alternatives [12].

Somewhat surprisingly, this is not the case with Horn formulae when (an incremental implementation of) the LTUR algorithm is used. A modified insertion-based MUS extraction algorithm that exploits LTUR is shown in [Algorithm 3](#). LTUR\_prop propagates the consequences of adding some new set of clauses, given some existing incremental context. LTUR\_undo unpropagates the consequences of adding some set of clauses (in order), given some existing

incremental context. The organization of the algorithm mimics the standard insertion-based MUS extraction algorithm [16], but the use of the incremental LTUR yields run time complexity that improves over other approaches. Consider the operation of the standard insertion-based algorithm [16], in which clauses are iteratively added to the working formula. When the formula becomes unsatisfiable, a *transition clause* [12] has been identified, which is then added to the MUS being constructed. The well-known query complexity of the insertion-based algorithm is  $\mathcal{O}(m \times k)$  where  $m$  is the number of clauses and  $k$  is the size of a largest MUS. Now consider that the incremental LTUR algorithm is used. To find the first transition clause, the amortized run time is  $\mathcal{O}(|\mathcal{F}|)$ . Clearly, this holds true for *any* transition clause, and so the run time of MUS extraction with the LTUR algorithm becomes  $\mathcal{O}(|\mathcal{M}| \times |\mathcal{F}|)$ , where  $\mathcal{M} \subseteq \mathcal{I}$  is a largest MUS. Algorithm 3 highlights the main differences with respect to a standard insertion-based MUS extraction algorithm. In contrast, observe that for a deletion-based algorithm the run time complexity will be  $\mathcal{O}(|\mathcal{I}| \times |\mathcal{F}|)$ . In situations where the sizes of MUSes are much smaller than the number of groups in  $\mathcal{I}$ , this difference can be significant. As a result, when extracting MUSes from Horn formulae, and when using a polynomial time incremental decision procedure, an insertion-based algorithm should be used instead of other more commonly used alternatives.

## 5 Comparison with Existing Alternatives

This section compares HGMUS with the group MUS enumerators used in EL<sup>+</sup>SAT [50, 51], EL2MCS [1] and SATPin [33]. An experimental comparison with these and other methods for axiom pinpointing for the  $\mathcal{EL}$  family of DLs is presented in Section 6.

### 5.1 EL<sup>+</sup>SAT

The best known SAT-based approach for axiom pinpointing is EL<sup>+</sup>SAT [50, 51, 56]. EL<sup>+</sup>SAT is composed of two main phases. The first phase compiles the axiom pinpointing problem to a Horn formula. The second phase enumerates the so-called MinAs, and corresponds to group-MUS enumeration for this Horn formula [1]. Although existing references emphasize the enumeration of MinAs (MUSes) using an AllSAT approach (itself inspired by an AllSMT approach [27]), the connection with MUS enumeration is immediate [1]. More importantly, EL<sup>+</sup>SAT shares a number of similarities with implicit minimal hitting set dualization, but also crucial differences, which we now analyze.

Similar to EMUS, EL<sup>+</sup>SAT selects subformulae of an unsatisfiable Horn formula. This is achieved with a SAT solver that always assigns variables value 1 when branching [51]. This corresponds to solving SAT with preferences [20, 47], and so it corresponds to computing a maximal model, inasmuch the same way as EMUS operates.

In EL<sup>+</sup>SAT, the approach for deciding the satisfiability of Horn subformulae is based on running the unit propagation engine of a CDCL SAT solver. As explained earlier, this can be inefficient when compared with the dedicated LTUR algorithm for Horn formulae [38]. Moreover, in EL<sup>+</sup>SAT, MUSes

are extracted with what can be viewed as a deletion-based algorithm [11, 12]. Although more efficient alternatives are suggested, none is as asymptotically as efficient as the dedicated algorithm proposed in [Section 4.5](#).

Finally, the most important drawback is the blocking of sets of clauses that do not contain an MUS/MinA. In our setting of implicit minimal hitting set dualization, this represents one MCS. The approach used in EL<sup>+</sup>SAT consists of creating a blocking clause solely based on the decision variables (which are *always* assigned value 1) [51, 56]<sup>2</sup>. This means that MUSes (or MinAs) and MCSes/MSSes are blocked the same way. Thus, the learned clauses, although blocking one MCS (and corresponding MSS), do *not* block supersets of MCSes (and the corresponding subsets of the MSSes). This can result in exponentially more iterations than necessary, and explains in part the poor performance of EL<sup>+</sup>SAT in practice. It should be further observed that this drawback becomes easier to spot once the problem is described as MUS enumeration by implicit minimal hitting set dualization.

## 5.2 EL2MCS

EL2MCS [1] implements explicit minimal hitting set dualization. In a first phase the MCS enumerator CAMUS2 [34] is used (the original CAMUS cannot be used because the formula has groups). This is achieved by iterated MaxSAT enumeration. In a second phase the MUS enumerator CAMUS [30] is used. The differences to HGMUS are clear, in that EL2MCS uses explicit minimal hitting set dualization and HGMUS uses implicit minimal hitting set dualization. Thus, there are (possibly many) instances for which EL2MCS will be unable to compute MUSes, because it will be unable to enumerate all MCSes, and this will not be the case with HGMUS. Another potential drawback of EL2MCS is that it uses a MaxSAT solver for MCS enumeration, although there are better alternatives [34]. Nevertheless, EL2MCS outperforms other existing approaches [5, 31, 33, 50, 51]. As shown later, the HGMUS approach proposed in this paper is the only one that consistently outperforms EL2MCS.

## 5.3 SATPin

SATPin [33] represents a recent SAT-based alternative for axiom pinpointing for the  $\mathcal{EL}$  family of DLs, that focuses on optimizing the low-level implementation details of the CDCL SAT solver, including the use of incremental SAT solving. As indicated above, HGMUS opts to revisit instead the LTUR [38] algorithm from the late 80s, since it is guaranteed to run in linear time for Horn formulae, and can be implemented with small overhead. The SATPin approach is presented in terms of iteratively computing implicants. Some aspects of the organization of SATPin can be related with those of EL<sup>+</sup>SAT, namely the procedure for extracting MUSes/MinAs. Although the actual enumeration of candidate sets is not detailed in [33], the description of SATPin suggests the use of model enumeration with some essential pruning techniques.

---

<sup>2</sup> The clause learning mechanism used in EL<sup>+</sup>SAT is detailed in [51], page 17, first paragraph.

## 6 Experimental Results

This section evaluates group-MUS enumerators for Horn formulae obtained from axiom pinpointing problems for the  $\mathcal{EL}$  family of DLs, particularly applied to medical ontologies. A set of standard benchmarks is considered. These have been used in earlier work, e.g. [1, 5, 31, 33, 50].

Since all experiments consist of converting axiom pinpointing problems into group-MUS enumeration problems, the tool that uses HGMUS<sup>3</sup> as its back-end is named EL2MUS. Thus, in this section, the results for EL2MUS illustrate the performance of the group-MUS enumerator described in this paper.

### 6.1 Experimental Setup

Each considered instance represents the problem of explaining a particular subsumption relation (query) entailed in a medical ontology. Four medical ontologies<sup>4</sup> are considered: GALEN [45], GENE [2], NCI [52] and SNOMED CT [55]. For GALEN, we consider two variants: FULL-GALEN and NOT-GALEN. The most important ontology is SNOMED CT and, due to its huge size, it also produces the hardest axiom pinpointing instances. For each ontology (including the GALEN variants) 100 queries are considered; 50 random (expected to be easier) and 50 sorted (expected to have a large number of minimal explanations) queries. So, there are 500 queries in total.

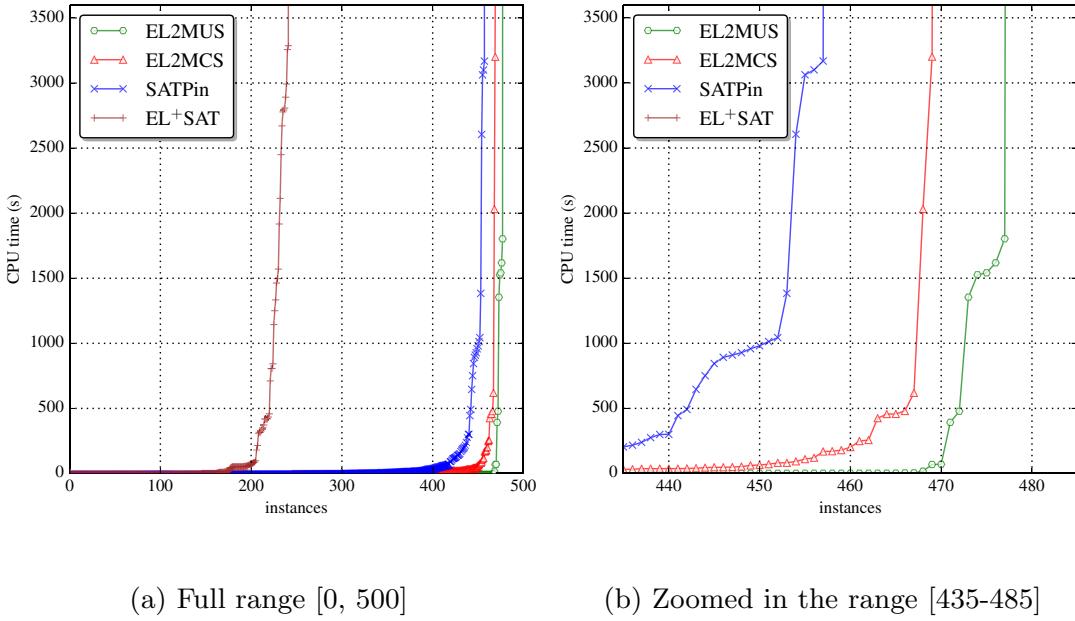
Given an ontology, the encoding proposed in [50, 51] produces a Horn formula that represents the reasoning steps taken in the deduction of all the subsumption relations entailed by the ontology. In this formula, every variable represents a subsumption relation between two concepts. As a result, the encoding also produces a set of variables corresponding to the original axioms of the ontology, which may be responsible for any subsumption relation. Explaining a given subsumption relation (query) can be then transformed into a group-MUS enumeration problem where the original Horn formula and a unit clause with the negated query forms group-0 and each original axiom constitutes a group containing only a unit clause. Noticeably, any general Horn group-MUS problem can be converted to this particular format.

Two different experiments were considered by applying two different simplification techniques to the problem instances, both of which were proposed in [51]. The first one uses the Cone-Of-Influence (COI) reduction. These are reduced instances in both the size of the Horn formula and the number of axioms, but are still quite large. Similar techniques are exploited in related work [5, 31, 33]. The second one considers the more effective reduction technique (which we refer to as x2), consisting in applying the COI technique, re-encoding the Horn formula into a reduced ontology, and encoding this ontology again into a Horn formula. This results in small Horn formulae, which will be useful to evaluate the algorithms when there are a large number of MUSes/MCSes.

---

<sup>3</sup> HGMUS is available at <http://logos.ucd.ie/web/doku.php?id=hgmus>.

<sup>4</sup> GENE, GALEN and NCI ontologies are freely available at <http://lat.inf.tu-dresden.de/~meng/toyont.html>. The SNOMED CT ontology was requested from IHTSDO under a nondisclosure license agreement.



**Fig. 1.** Cactus plots comparing EL<sup>+</sup>SAT, SATPin, EL2MCS and EL2MUS on the COI instances

The experiments compare EL2MUS to different algorithms, namely EL<sup>+</sup>SAT [50, 51], CEL [5], JUST [31], EL2MCS [1] and SATPin [33]. EL<sup>+</sup>SAT [51] has been shown to outperform CEL [5], whereas SATPin [33] has been shown to outperform the MUS enumerator MARCO [29].

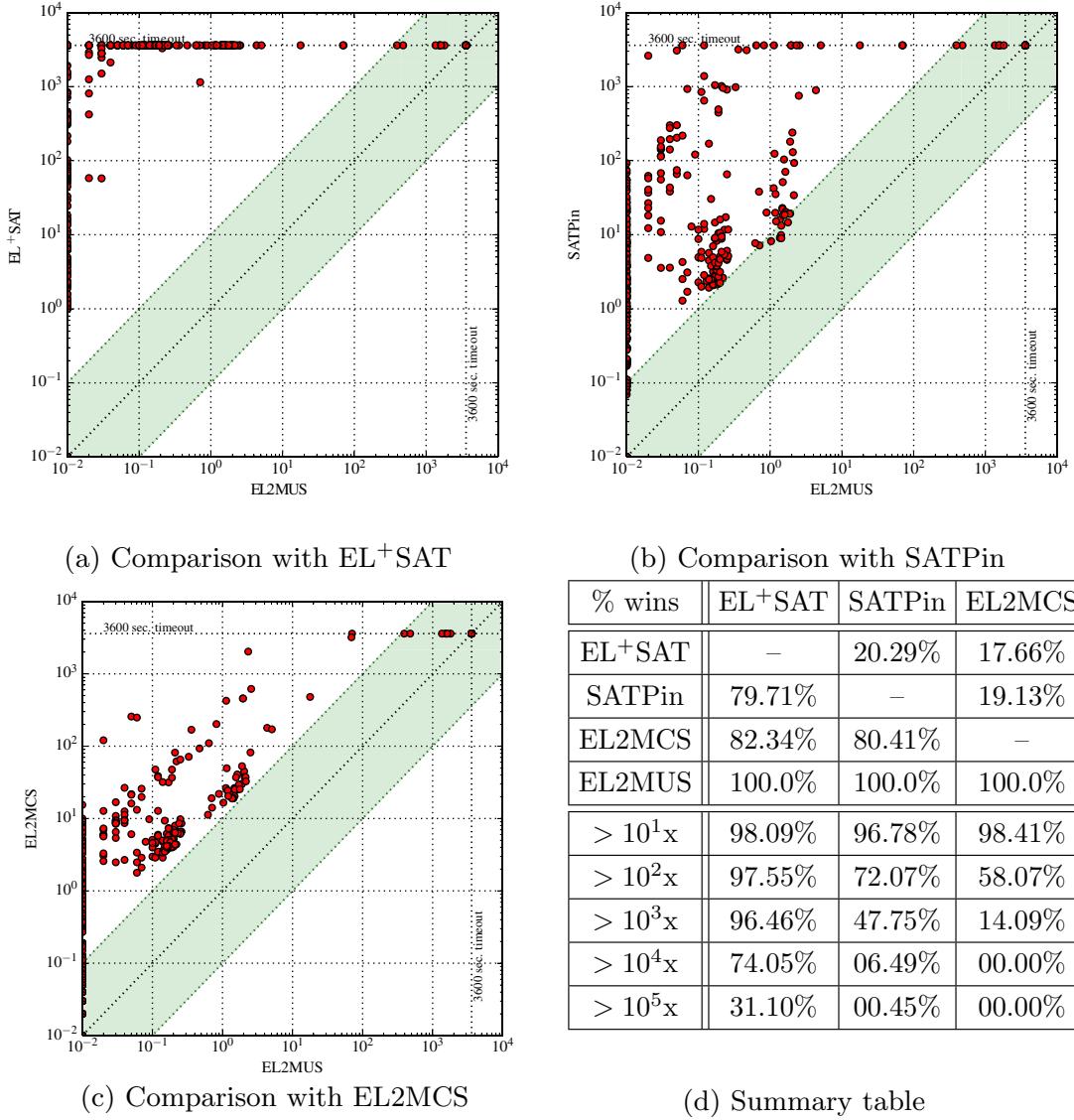
The comparison with CEL and JUST imposes a number of constraints. First, CEL only computes 10 MinAs, so all comparisons with CEL only consider reporting the first 10 MinAs/MUSES. Also, CEL uses a simplification technique similar to COI, so CEL is considered in the first experiments. Second, JUST operates on selected subsets of  $\mathcal{EL}^+$ , i.e. the description logic used in most medical ontologies. As a result, all comparisons with JUST consider solely the problem instances for which JUST can compute correct results. JUST accepts the simplified x2 ontologies, so it is considered in the second experiments. The comparison with these tools is presented at the end of the section.

EL2MUS interfaces the SAT solver Minisat 2.2 [18] for computing maximal models. All the experiments were performed on a Linux cluster (2 GHz) and the algorithms were given a time limit of 3600s and a memory limit of 4 GB<sup>5</sup>.

## 6.2 COI Instances

Figure 1 summarizes the results for EL<sup>+</sup>SAT, EL2MCS, SATPin and EL2MUS. As can be observed, EL2MCS has a slight performance advantage over SATPin, and EL2MUS terminates for more instances than any of the other tools. Figure 2 shows scatter plots comparing the different tools. As can be concluded, and with a few outliers, the performance of EL2MUS exceeds the performance of any of the other tools by at least one order of magnitude (and often by more). Figure 2d

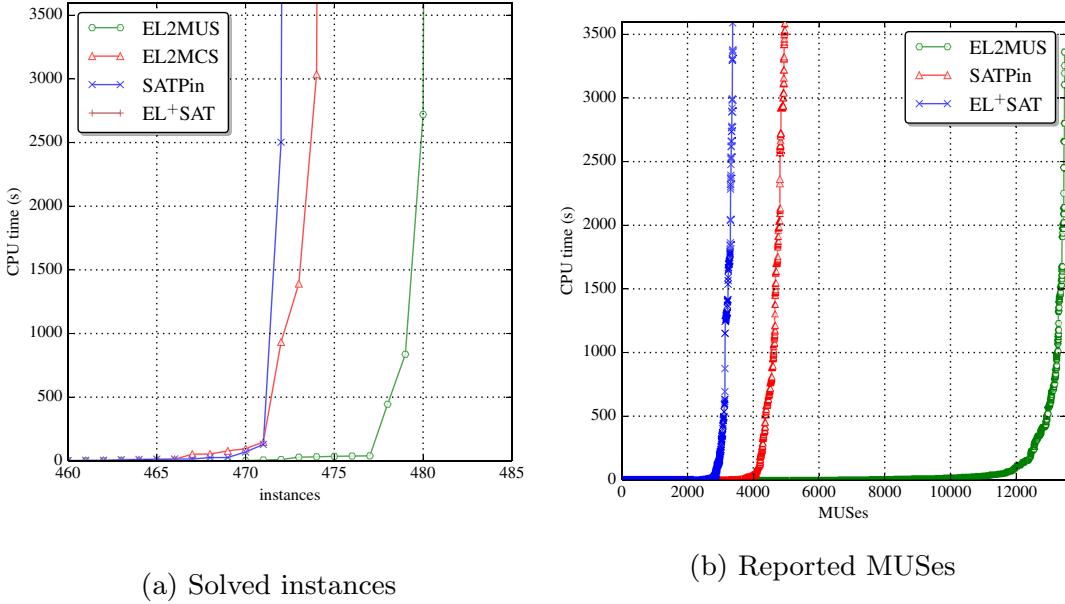
<sup>5</sup> Only a sample of the results can be presented in this section due to space restrictions. Additional results are available at <http://logos.ucd.ie/web/doku.php?id=hgmus>.

**Fig. 2.** Scatter plots for COI instances

summarizes the results in the scatter plots, where the percentages shown are computed for problem instances for which at least one of the tools takes more than 0.001s. As can be observed, EL2MUS outperforms any of the other tools in all of the problem instances and, for many cases, with two or more orders of magnitude improvement.

### 6.3 x2 Instances

The x2 instances are significantly simpler than the COI instances. Thus, whereas the COI instances can serve to assess the scalability of each approach, the x2 instances highlight the expected performance in representative settings. Figure 3a summarizes the performance of the tools EL<sup>+</sup>SAT, SATPin, EL2MCS and EL2MUS. Due to its poor performance, EL<sup>+</sup>SAT does not show in the plot (it terminates on 317 instances). Moreover, and as before in terms of terminated instances, EL2MUS exhibits an observable performance edge.

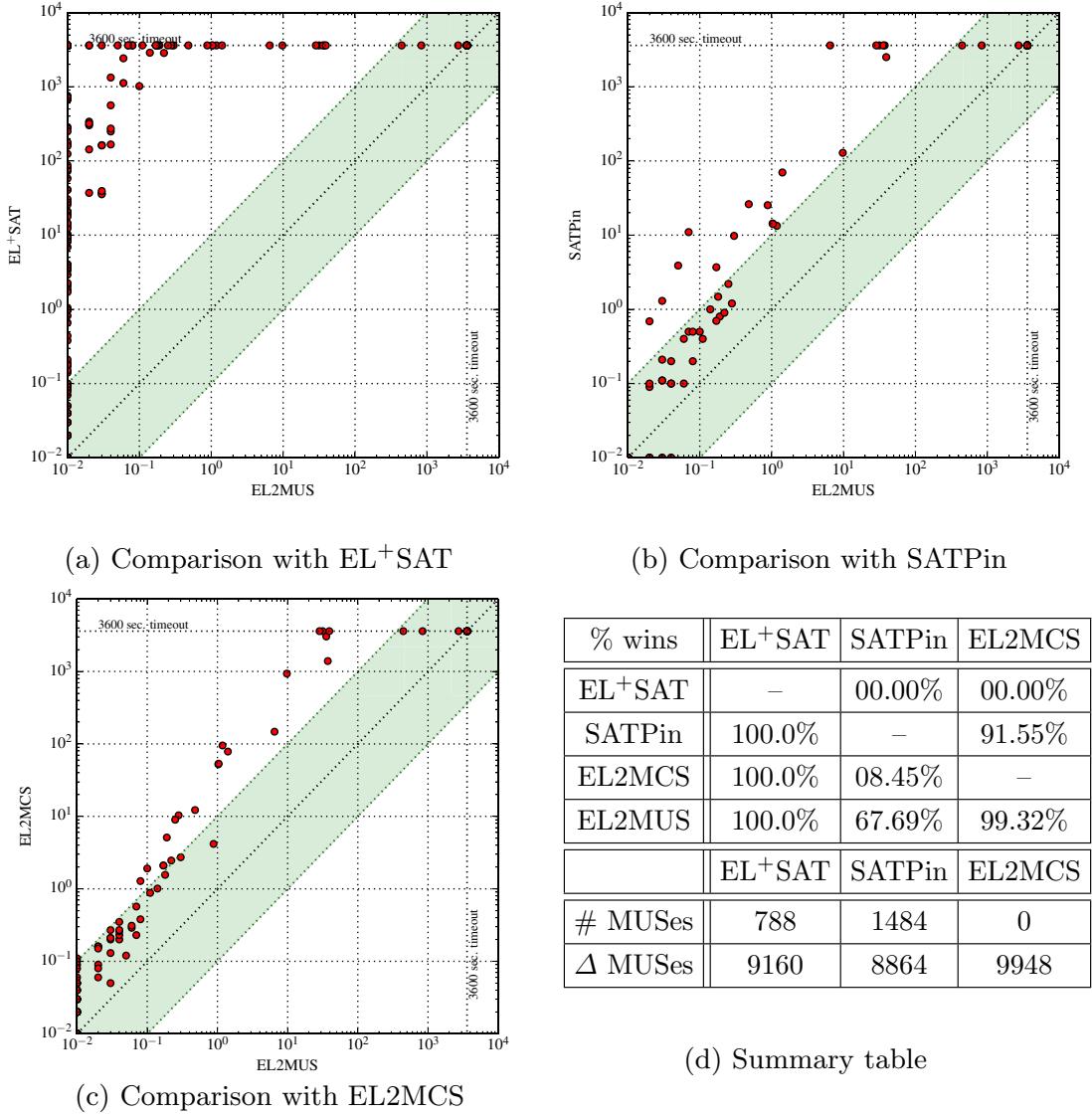


**Fig. 3.** Cactus plots comparing  $\text{EL}^+\text{SAT}$ , SATPin, EL2MCS and EL2MUS on the x2 instances

A pairwise comparison between the different tools is summarized in Figure 4. Although not as impressive as for the COI instances, EL2MUS still consistently outperforms all other tools. Figure 4d summarizes the results, where as before the percentages shown are computed for problem instances for which at least one of the tools takes more than 0.001s. Observe that, for these easier instances, SATPin becomes competitive with EL2MUS. Nevertheless, for instances taking more than 0.1s, EL2MUS outperforms SATPin on 100% of the instances. Thus, the 67.69% shown in the table result from instances for which both SATPin and EL2MUS take at most 0.04s. The summary table also lists the number of computed MUSes for the 19 instances for which EL2MUS does *not* terminate (all of the other tools also do not terminate for these 19 instances). EL2MUS computes 9948 MUSes in total. As can be observed from the table, the other tools lag behind, and compute significantly fewer MUSes. Also, as noted earlier in the paper, the main issue with EL2MCS is demonstrated with these results; for these 19 instances, EL2MCS is unable to compute any MUSes. The comparison with the other tools,  $\text{EL}^+\text{SAT}$  and SATPin, reveals that EL2MUS computes respectively in excess of a factor of 10 and of 5 more MUSes.

EL2MUS not only terminates on more instances than any other approach and computes more MUSes for the unsolved instances; it also reports the sequences of MUSes much faster. Figure 3b shows, for each computed MUS over the whole set of instances, the time each MUS was reported. This figure compares  $\text{EL}^+\text{SAT}$ , SATPin and EL2MUS, as these are the only methods able to report MUSes from the beginning. The results confirm that EL2MUS is able to find many more MUSes in less time than the alternatives.

These experimental results suggest that, not only is EL2MUS the best performing axiom pinpointing tool, on both the COI and x2 problem instances, but

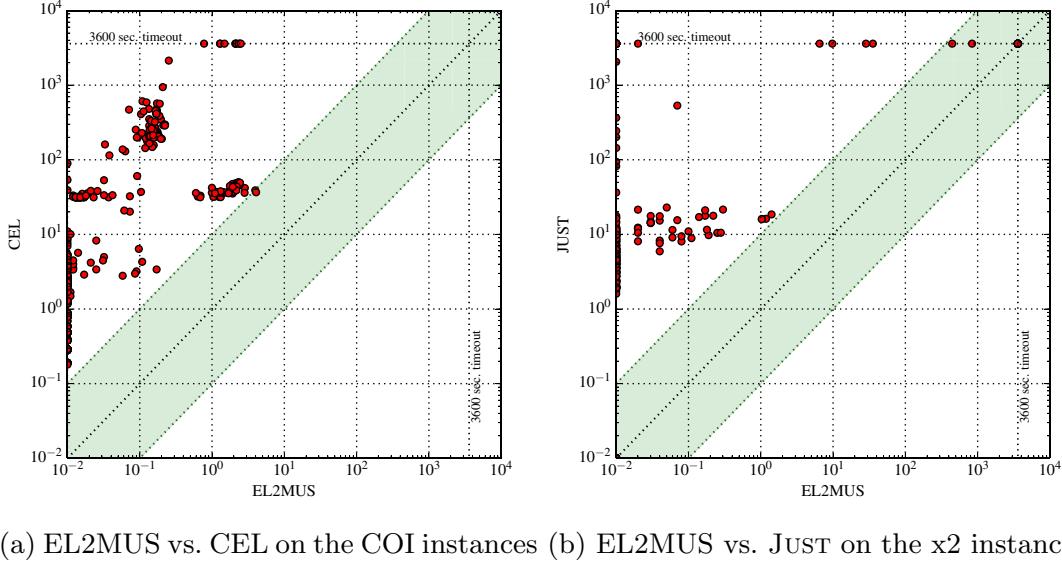
**Fig. 4.** Scatter plots for x2 instances

it is also the one that is expected to scale better for more challenging problem instances, given the results on the COI instances.

#### 6.4 Assessment of Non SAT-Based Axiom Pinpointing Tools

Figure 5 shows scatter plots comparing EL2MUS with CEL [5] and JUST [31], respectively for the COI and x2 instances<sup>6</sup>. As indicated earlier, CEL only computes 10 MinAs, and so the run times shown are for computing the first 10 MinA/MUSes. As can be observed, the performance edge of EL2MUS is clear, with the performance gap exceeding 1 order of magnitude almost without exception. Moreover, JUST [31] is a recent state of the art axiom pinpointing tool for the less expressive  $\mathcal{ELH}$  DL. Thus, not all subsumption relations can be represented and analyzed. The results shown are for the subsumption relations for

<sup>6</sup> Due to lack of space the other scatter plots are not shown, but the conclusions are the same.



(a) EL2MUS vs. CEL on the COI instances (b) EL2MUS vs. JUST on the x2 instances

**Fig. 5.** Comparison of EL2MUS with CEL and with JUST

which JUST gives the correct results. In total, 382 instances could be considered and are shown in the plot. As before, the performance edge of EL2MUS is clear, with the performance gap exceeding 1 order of magnitude without exception. In this case, since the x2 instances are in general much simpler, the performance gap is even more significant.

## 7 Conclusions and Future Work

Enumeration of group MUS for Horn formulae finds important applications, including axiom pinpointing for the  $\mathcal{EL}$  family of DLs. Since the  $\mathcal{EL}$  family of DLs is widely used for representing medical ontologies, namely with  $\mathcal{EL}^+$ , enumeration of group MUSes for Horn formulae represents a promising and strategic application of SAT technology. This includes, among others, SAT solvers, MCS extractors and enumerators, and MUS extractors and enumerators. This paper develops a highly optimized group MUS enumerator for Horn formulae, which is shown to extensively outperform any other existing approach. Performance gains are almost without exception at least one order of magnitude, and most often significantly more than that. More importantly, the experimental results demonstrate that SAT-based approaches are by far the most effective approaches for axiom pinpointing for the  $\mathcal{EL}$  family of DLs. When compared with other non SAT-based approaches, the performance gains are also conclusive.

Future work will exploit integration of additional recent work on SAT-based problem solving, e.g. in MCS enumeration and MUS enumeration, to further improve performance of axiom pinpointing.

**Acknowledgments.** We thank the authors of EL<sup>+</sup>SAT, R. Sebastiani and M. Vescovi, for authorizing the use of the most recent, yet unpublished, version of their work [51]. We thank the authors of SATPin [33], N. Manthey and R. Peñaloza, for bringing SATPin to our attention, and for allowing us to use their tool. We thank A. Biere for

pointing out reference [19], on the complexity of implementing unit propagation when using watched literals. This work is partially supported by SFI PI grant BEACON (09/IN.1/I2618), by FCT grant POLARIS (PTDC/EIA-CCO/123051/2010), and by national funds through FCT with reference UID/CEC/50021/2013.

## References

1. Arif, M.F., Mencía, C., Marques-Silva, J.: Efficient axiom pinpointing with EL2MCS. In: KI (2015)
2. Ashburner, M., Ball, C.A., Blake, J.A., Botstein, D., Butler, H., Cherry, J.M., Davis, A.P., Dolinski, K., Dwight, S.S., Eppig, J.T., et al.: Gene ontology: tool for the unification of biology. *Nature genetics* **25**(1), 25–29 (2000)
3. Baader, F., Hollunder, B.: Embedding defaults into terminological knowledge representation formalisms. *J. Autom. Reasoning* **14**(1), 149–180 (1995)
4. Baader, F., Horrocks, I., Sattler, U.; Description logics. In: van Harmelen, V.L.F., Porter, B. (eds.), *Handbook of Knowledge Representation, Foundations of Artificial Intelligence*, chapter 3, pp. 135–179. Elsevier (2008)
5. Baader, F., Lutz, C., Suntisrivaraporn, B.: CEL — a polynomial-time reasoner for life science ontologies. In: Furbach, U., Shankar, N. (eds.) IJCAR 2006. LNCS (LNAI), vol. 4130, pp. 287–291. Springer, Heidelberg (2006)
6. Baader, F., Peñaloza, R.: Axiom pinpointing in general tableaux. *J. Log. Comput.* **20**(1), 5–34 (2010)
7. Baader, F., Peñaloza, R., Suntisrivaraporn, B.: Pinpointing in the description logic  $\mathcal{EL}^+$ . In: KI, pp. 52–67 (2007)
8. Baader, F., Suntisrivaraporn, B.: Debugging SNOMED CT using axiom pinpointing in the description logic  $\mathcal{EL}^+$ . In: KR-MED (2008)
9. Bacchus, F., Davies, J., Tsimpoukelli, M., Katsirelos, G.: Relaxation search: a simple way of managing optional clauses. In: AAAI, pp. 835–841 (2014)
10. Bailey, J., Stuckey, P.J.: Discovery of minimal unsatisfiable subsets of constraints using hitting set dualization. In: Hermenegildo, M.V., Cabeza, D. (eds.) PADL 2004. LNCS, vol. 3350, pp. 174–186. Springer, Heidelberg (2005)
11. Bakker, R.R., Dikker, F., Tempelman, F., Wognum, P.M.: Diagnosing and solving over-determined constraint satisfaction problems. In: IJCAI, pp. 276–281 (1993)
12. Belov, A., Lynce, I., Marques-Silva, J.: Towards efficient MUS extraction. *AI Commun.* **25**(2), 97–116 (2012)
13. Biere, A., Heule, M., van Maaren, H., Walsh, T. (eds.): *Handbook of Satisfiability, Frontiers in Artificial Intelligence and Applications*, vol. 185. IOS Press (2009)
14. Birnbaum, E., Lozinskii, E.L.: Consistent subsets of inconsistent systems: structure and behaviour. *J. Exp. Theor. Artif. Intell.* **15**(1), 25–46 (2003)
15. Cook, S.A.: The complexity of theorem-proving procedures. In: STOC, pp. 151–158 (1971)
16. de Siqueira, N.J.L., Puget, J.-F.: Explanation-based generalisation of failures. In: ECAI, pp. 339–344 (1988)
17. Dowling, W.F., Gallier, J.H.: Linear-time algorithms for testing the satisfiability of propositional Horn formulae. *J. Log. Program.* **1**(3), 267–284 (1984)
18. Eén, N., Sörensson, N.: An extensible SAT-solver. In: Giunchiglia, E., Tacchella, A. (eds.) SAT 2003. LNCS, vol. 2919, pp. 502–518. Springer, Heidelberg (2004)
19. Gent, I.: Optimal implementation of watched literals and more general techniques. *Journal of Artificial Intelligence Research* **48**, 231–252 (2013)
20. Giunchiglia, E., Maratea, M.: Solving optimization problems with DLL. In: ECAI, pp. 377–381 (2006)

21. Grégoire, É., Lagniez, J., Mazure, B.: An experimentally efficient method for (MSS, CoMSS) partitioning. In: AAAI, pp. 2666–2673 (2014)
22. Heras, F., Morgado, A., Marques-Silva, J.: MaxSAT-based encodings for group MaxSAT. *AI Commun.* **28**(2), 195–214 (2015)
23. Itai, A., Makowsky, J.A.: Unification as a complexity measure for logic programming. *J. Log. Program.* **4**(2), 105–117 (1987)
24. Junker, U.: QuickXplain: preferred explanations and relaxations for over-constrained problems. In: AAAI, pp. 167–172 (2004)
25. Kalyanpur, A., Parsia, B., Horridge, M., Sirin, E.: Finding all justifications of OWL DL entailments. In: Aberer, K., Choi, K.-S., Noy, N., Allemang, D., Lee, K.-I., Nixon, L.J.B., Golbeck, J., Mika, P., Maynard, D., Mizoguchi, R., Schreiber, G., Cudré-Mauroux, P. (eds.) ASWC 2007 and ISWC 2007. LNCS, vol. 4825, pp. 267–280. Springer, Heidelberg (2007)
26. Kalyanpur, A., Parsia, B., Sirin, E., Cuenca-Grau, B.: Repairing unsatisfiable concepts in OWL ontologies. In: Sure, Y., Domingue, J. (eds.) ESWC 2006. LNCS, vol. 4011, pp. 170–184. Springer, Heidelberg (2006)
27. Lahiri, S.K., Nieuwenhuis, R., Oliveras, A.: SMT techniques for fast predicate abstraction. In: Ball, T., Jones, R.B. (eds.) CAV 2006. LNCS, vol. 4144, pp. 424–437. Springer, Heidelberg (2006)
28. Liffiton, M.H., Malik, A.: Enumerating infeasibility: finding multiple MUSes quickly. In: Gomes, C., Sellmann, M. (eds.) CPAIOR 2013. LNCS, vol. 7874, pp. 160–175. Springer, Heidelberg (2013)
29. Liffiton, M.H., Previti, A., Malik, A., Marques-Silva, J.: Fast, flexible MUS enumeration. *Constraints* (2015). <http://link.springer.com/article/10.1007/s10601-015-9183-0>
30. Liffiton, M.H., Sakallah, K.A.: Algorithms for computing minimal unsatisfiable subsets of constraints. *J. Autom. Reasoning* **40**(1), 1–33 (2008)
31. Ludwig, M.: Just: a tool for computing justifications w.r.t. ELH ontologies. In: ORE (2014)
32. Ludwig, M., Peñaloza, R.: Error-tolerant reasoning in the description logic  $\mathcal{EL}$ . In: Fermé, E., Leite, J. (eds.) JELIA 2014. LNCS, vol. 8761, pp. 107–121. Springer, Heidelberg (2014)
33. Manthey, N., Peñaloza, R.: Exploiting SAT technology for axiom pinpointing. Technical Report LTCS 15–05, Chair of Automata Theory, Institute of Theoretical Computer Science, Technische Universität Dresden, April 2015. <https://ddll.inf.tu-dresden.de/web/Techreport3010>
34. Marques-Silva, J., Heras, F., Janota, M., Previti, A., Belov, A.: On computing minimal correction subsets. In: IJCAI, pp. 615–622 (2013)
35. Marques-Silva, J., Janota, M., Belov, A.: Minimal sets over monotone predicates in boolean formulae. In: Sharygina, N., Veith, H. (eds.) CAV 2013. LNCS, vol. 8044, pp. 592–607. Springer, Heidelberg (2013)
36. Mencía, C., Previti, A., Marques-Silva, J.: Literal-based MCS extraction. In: IJCAI, pp. 1973–1979 (2015)
37. Meyer, T.A., Lee, K., Booth, R., Pan, J.Z.: Finding maximally satisfiable terminologies for the description logic  $\mathcal{EL}^+$ . In: AAAI, pp. 269–274 (2006)
38. Minoux, M.: LTUR: A simplified linear-time unit resolution algorithm for Horn formulae and computer implementation. *Inf. Process. Lett.* **29**(1), 1–12 (1988)
39. Moodley, K., Meyer, T., Varzinczak, I.J.: Root justifications for ontology repair. In: Rudolph, S., Gutierrez, C. (eds.) RR 2011. LNCS, vol. 6902, pp. 275–280. Springer, Heidelberg (2011)

40. Nguyen, H.H., Alechina, N., Logan, B.: Axiom pinpointing using an assumption-based truth maintenance system. In: DL (2012)
41. O'Sullivan, B., Papadopoulos, A., Faltings, B., Pu, P.: Representative explanations for over-constrained problems. In: AAAI, pp. 323–328 (2007)
42. Parsia, B., Sirin, E., Kalyanpur, A.; Debugging OWL ontologies. In: WWW, pp. 633–640 (2005)
43. Peñaloza, R., Sertkaya, B.: On the complexity of axiom pinpointing in the EL family of description logics. In: KR (2010)
44. Previti, A., Marques-Silva, J.: Partial MUS enumeration. In: AAAI, pp. 818–825 (2013)
45. Rector, A.L., Horrocks, I.R.: Experience building a large, re-usable medical ontology using a description logic with transitivity and concept inclusions. In: Workshop on Ontological Engineering, pp. 414–418 (1997)
46. Reiter, R.: A theory of diagnosis from first principles. Artif. Intell. **32**(1), 57–95 (1987)
47. Rosa, E.D., Giunchiglia, E.: Combining approaches for solving satisfiability problems with qualitative preferences. AI Commun. **26**(4), 395–408 (2013)
48. Schlobach, S., Cornet, R.: Non-standard reasoning services for the debugging of description logic terminologies. In: IJCAI, pp. 355–362 (2003)
49. Schlobach, S., Huang, Z., Cornet, R., van Harmelen, F.: Debugging incoherent terminologies. J. Autom. Reasoning **39**(3), 317–349 (2007)
50. Sebastiani, R., Vescovi, M.: Axiom pinpointing in lightweight description logics via horn-SAT encoding and conflict analysis. In: Schmidt, R.A. (ed.) CADE-22. LNCS, vol. 5663, pp. 84–99. Springer, Heidelberg (2009)
51. Sebastiani, R., Vescovi, M.: Axiom pinpointing in large  $\mathcal{EL}^+$  ontologies via SAT and SMT techniques. Technical Report DISI-15-010, DISI, University of Trento, Italy, April 2015. Under Journal Submission. <http://disi.unitn.it/rseba/elsat/elsat-techrep.pdf>
52. Sioutos, N., de Coronado, S., Haber, M.W., Hartel, F.W., Shaiu, W., Wright, L.W.: NCI thesaurus: A semantic model integrating cancer-related clinical and molecular information. Journal of Biomedical Informatics **40**(1), 30–43 (2007)
53. Sirin, E., Parsia, B., Grau, B.C., Kalyanpur, A., Katz, Y.: Pellet: A practical OWL-DL reasoner. J. Web Sem. **5**(2), 51–53 (2007)
54. Slaney, J.: Set-theoretic duality: a fundamental feature of combinatorial optimisation. In: ECAI, pp. 843–848 (2014)
55. Spackman, K.A., Campbell, K.E., Côté, R.A.: SNOMED RT: a reference terminology for health care. In: AMIA (1997)
56. Vescovi, M.: Exploiting SAT and SMT Techniques for Automated Reasoning and Ontology Manipulation in Description Logics. Ph.D. thesis, University of Trento (2011)

# Chapter 6

## SAT '16 Publication

- Citation ([DOI:10.1007/978-3-319-40970-2\\_32](https://doi.org/10.1007/978-3-319-40970-2_32)): M. Fareed Arif, Carlos Mencía, Alexey Ignatiev, Norbert Manthey, Rafael Peñaloza, and Joao Marques-Silva. BEACON: An Efficient SAT-Based Tool for Debugging  $\mathcal{EL}^+$  Ontologies. In *Theory and Applications of Satisfiability Testing – SAT 2016 – 19th International Conference, Bordeaux, France, July 5–8, 2016, Proceedings*, volume 9710, pages 521–530, 2016
- It is a tool paper detailing the final results of the work conducted in [[AMMS15a](#), [AMMS15b](#)] and develops the state-of-the art debugging and axiom pin-pointing tool that also supports additional functionalities (e.g., compute the smallest MinA).
- SAT '16 is a CORE A conference and we have published a tool paper in it.

## CHAPTER 6. SAT '16 PUBLICATION

- Contribution: The candidate contributed actively in the discussion process, providing ideas that affected deeply the design, development and the experimental evaluation of the BEACON tool.

# BEACON: An Efficient SAT-Based Tool for Debugging $\mathcal{EL}^+$ Ontologies

M. Fareed Arif<sup>1</sup>, Carlos Mencía<sup>2(✉)</sup>, Alexey Ignatiev<sup>3,6</sup>, Norbert Manthey<sup>4</sup>,  
Rafael Peñaloza<sup>5</sup>, and Joao Marques-Silva<sup>3</sup>

<sup>1</sup> University College Dublin, Dublin, Ireland

[muhammad.arif.1@ucdconnect.ie](mailto:muhammad.arif.1@ucdconnect.ie)

<sup>2</sup> University of Oviedo, Oviedo, Spain

[cmencia@gmail.com](mailto:cmencia@gmail.com)

<sup>3</sup> University of Lisbon, Lisbon, Portugal

[{aignatiev,jpms}@ciencias.ulisboa.pt](mailto:{aignatiev,jpms}@ciencias.ulisboa.pt)

<sup>4</sup> TU Dresden, Dresden, Germany

[norbert.manthey@tu-dresden.de](mailto:norbert.manthey@tu-dresden.de)

<sup>5</sup> Free University of Bozen-Bolzano, Bolzano, Italy

[rafael.penaloza@unibz.it](mailto:rafael.penaloza@unibz.it)

<sup>6</sup> ISDCT SB RAS, Irkutsk, Russia

**Abstract.** Description Logics (DLs) are knowledge representation and reasoning formalisms used in many settings. Among them, the  $\mathcal{EL}$  family of DLs stands out due to the availability of polynomial-time inference algorithms and its ability to represent knowledge from domains such as medical informatics. However, the construction of an ontology is an error-prone process which often leads to unintended inferences. This paper presents the BEACON tool for debugging  $\mathcal{EL}^+$  ontologies. BEACON builds on earlier work relating minimal justifications (MinAs) of  $\mathcal{EL}^+$  ontologies and MUSes of a Horn formula, and integrates state-of-the-art algorithms for solving different function problems in the SAT domain.

## 1 Introduction

The importance of Description Logics (DLs) cannot be overstated, and impact a growing number of fields. The  $\mathcal{EL}$ -family of tractable DLs in particular has been used to build large ontologies from the life sciences [34,35]. Ontology development is an error-prone task, with potentially critical consequences in the life sciences; thus it is important to develop automated tools to help debugging large ontologies. Axiom pinpointing refers to the task of finding the precise axioms in an ontology that cause a (potentially unwanted) consequence to follow [25]. Recent years have witnessed remarkable improvements in axiom pinpointing technologies, including for the case of the  $\mathcal{EL}$  family of DLs [1,2,5–7,20,21,30,31]. Among these, the use of SAT-based methods [1,2,30] was shown to outperform other alternative approaches very significantly. This is achieved by reducing the problem to a propositional Horn formula, which is then analyzed with a dedicated decision engine for Horn formulae.

This paper describes a tool, BEACON, that builds on recent work on efficient enumeration of Minimal Unsatisfiable Subsets (MUSes) of group Horn formulae, which finds immediate application in axiom pinpointing of  $\mathcal{EL}$  ontologies [2]. In contrast to earlier work [2], which used EL+SAT [30, 31] as front-end, this paper proposes an integrated tool to perform analysis on ontologies, offering a number of new features.

The rest of the paper is organized as follows: Sect. 2 introduces some preliminaries. Section 3 describes the organization of BEACON. Experimental results are given in Sect. 4. Finally, Sect. 5 concludes the paper.

## 2 Preliminaries

### 2.1 The Lightweight Description Logic $\mathcal{EL}^+$

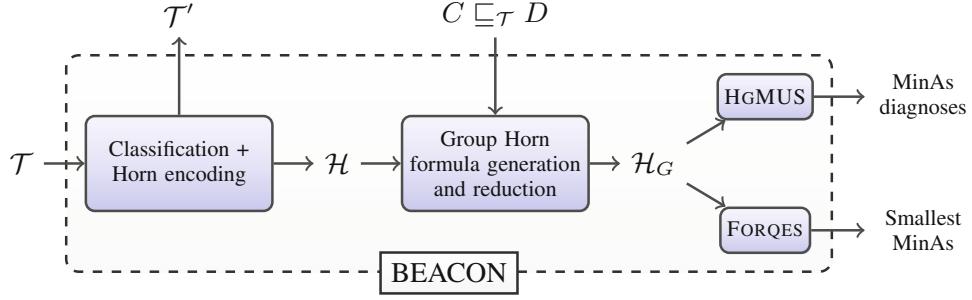
$\mathcal{EL}^+$  [4] is a light-weight DL that has been successfully used to build large ontologies, most notably from the bio-medical domains. As with all DLs, the main elements in  $\mathcal{EL}^+$  are concepts.  $\mathcal{EL}^+$  concepts are built from two disjoint sets  $N_C$  and  $N_R$  of *concept names* and *role names* through the grammar rule  $C ::= A \mid \top \mid C \sqcap C \mid \exists r.C$ , where  $A \in N_C$  and  $r \in N_R$ . The knowledge of the domain is stored in a *TBox* (ontology), which is a finite set of *general concept inclusions* (GCIs)  $C \sqsubseteq D$ , where  $C$  and  $D$  are  $\mathcal{EL}^+$  concepts, and *role inclusions* (RIs)  $r_1 \circ \dots \circ r_n \sqsubseteq s$ , where  $n \geq 1$  and  $r_i, s \in N_R$ . We will often use the term *axiom* to refer to both GCIs and RIs. As an example, Appendix  $\sqsubseteq \exists \text{partOf}.\text{Intestine}$  represents a GCI.

The semantics of this logic is based on *interpretations*, which are pairs of the form  $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$  where  $\Delta^\mathcal{I}$  is a non-empty set called the *domain* and  $\cdot^\mathcal{I}$  is the *interpretation function* that maps every  $A \in N_C$  to a set  $A^\mathcal{I} \subseteq \Delta^\mathcal{I}$  and every  $r \in N_R$  to a binary relation  $r^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I}$ . The interpretation  $\mathcal{I}$  *satisfies* the GCI  $C \sqsubseteq D$  iff  $C^\mathcal{I} \subseteq D^\mathcal{I}$ ; it *satisfies* the RI  $r_1 \circ \dots \circ r_n \sqsubseteq s$  iff  $r_1^\mathcal{I} \circ \dots \circ r_n^\mathcal{I} \subseteq s^\mathcal{I}$ , with  $\circ$  denoting composition of binary relations.  $\mathcal{I}$  is a *model* of  $\mathcal{T}$  iff  $\mathcal{I}$  satisfies all its GCIs and RIs.

The main reasoning problem in  $\mathcal{EL}^+$  is to decide subsumption between concepts. A concept  $C$  is *subsumed* by  $D$  w.r.t.  $\mathcal{T}$  (denoted  $C \sqsubseteq_{\mathcal{T}} D$ ) if for every model  $\mathcal{I}$  of  $\mathcal{T}$  it holds that  $C^\mathcal{I} \subseteq D^\mathcal{I}$ . *Classification* refers to the task of deciding all the subsumption relations between concept names appearing in  $\mathcal{T}$ . Rather than merely deciding whether a subsumption relation follows from a TBox, we are interested in understanding the causes of this consequence, and repairing it if necessary.

**Definition 1 (MinA, diagnosis).** A MinA for  $C \sqsubseteq D$  w.r.t. the TBox  $\mathcal{T}$  is a minimal subset (w.r.t. set inclusion)  $\mathcal{M} \subseteq \mathcal{T}$  such that  $C \sqsubseteq_{\mathcal{M}} D$ . A diagnosis for  $C \sqsubseteq D$  w.r.t.  $\mathcal{T}$  is a minimal subset (w.r.t. set inclusion)  $\mathcal{D} \subseteq \mathcal{T}$  such that  $C \not\sqsubseteq_{\mathcal{T} \setminus \mathcal{D}} D$ .

MinAs and diagnoses are closely related by minimal hitting set duality [19, 29].

**Fig. 1.** BEACON organization

*Example 2.* Consider the TBox  $\mathcal{T}_{\text{exa}} = \{A \sqsubseteq \exists r.A, A \sqsubseteq Y, \exists r.Y \sqsubseteq B, Y \sqsubseteq B\}$ . There are two MinAs for  $A \sqsubseteq B$  w.r.t.  $\mathcal{T}_{\text{exa}}$ , namely  $\mathcal{M}_1 = \{A \sqsubseteq Y, Y \sqsubseteq B\}$ , and  $\mathcal{M}_2 = \{A \sqsubseteq \exists r.A, A \sqsubseteq Y, \exists r.Y \sqsubseteq B\}$ . The diagnoses for this subsumption relation are  $\{A \sqsubseteq Y\}$ ,  $\{A \sqsubseteq \exists r.A, Y \sqsubseteq B\}$ , and  $\{\exists r.Y \sqsubseteq B, Y \sqsubseteq B\}$ .

## 2.2 Propositional Satisfiability

We assume familiarity with propositional logic [9]. A CNF formula  $\mathcal{F}$  is defined over a set of Boolean variables  $X$  as a finite conjunction of clauses, where a clause is a finite disjunction of literals and a literal is a variable or its negation. A truth assignment is a mapping  $\mu: X \rightarrow \{0, 1\}$ . If  $\mu$  satisfies  $\mathcal{F}$ ,  $\mu$  is referred to as a *model* of  $\mathcal{F}$ . Horn formulae are those composed of clauses with at most one positive literal. Satisfiability of Horn formulae is decidable in polynomial time [12, 15, 24]. Given an unsatisfiable formula  $\mathcal{F}$ , the following subsets are of interest [19, 22]:

**Definition 3 (MUS, MCS).**  $\mathcal{M} \subseteq \mathcal{F}$  is a Minimally Unsatisfiable Subset (MUS) of  $\mathcal{F}$  iff  $\mathcal{M}$  is unsatisfiable and  $\forall c \in \mathcal{M}, \mathcal{M} \setminus \{c\}$  is satisfiable.  $\mathcal{C} \subseteq \mathcal{F}$  is a Minimal Correction Subset (MCS) iff  $\mathcal{F} \setminus \mathcal{C}$  is satisfiable and  $\forall c \in \mathcal{C}, \mathcal{F} \setminus (\mathcal{C} \setminus \{c\})$  is unsatisfiable.

MUSes and MCSes are related by hitting set duality [8, 10, 28, 33]. Besides, these concepts have been extended to formulae where clauses are partitioned into *groups* [19].

**Definition 4 (Group-MUS).** Given an explicitly partitioned unsatisfiable CNF formula  $\mathcal{F} = \mathcal{G}_0 \cup \dots \cup \mathcal{G}_k$ , a group-MUS of  $\mathcal{F}$  is a set of groups  $\mathcal{G} \subseteq \{\mathcal{G}_1, \dots, \mathcal{G}_k\}$ , such that  $\mathcal{G}_0 \cup \mathcal{G}$  is unsatisfiable, and for every  $\mathcal{G}_i \in \mathcal{G}$ ,  $\mathcal{G}_0 \cup (\mathcal{G} \setminus \mathcal{G}_i)$  is satisfiable.

## 3 The BEACON Tool

The main problem BEACON is aimed at is the enumeration of the MinAs and diagnoses for a given subsumption relation w.r.t. an  $\mathcal{EL}^+$  TBox  $\mathcal{T}$ . BEACON

consists of three main components: The first one *classifies*  $\mathcal{T}$  and encodes this process into a set of Horn clauses. Given a subsumption to be analyzed, the second component creates and simplifies an unsatisfiable group Horn formula. Finally, the third one computes group-MUSes and group-MCSes, corresponding to MinAs and diagnoses resp. Figure 1 depicts the main organization of BEACON. Each of its components is explained below.

### 3.1 Classification and Horn Encoding

During the classification of  $\mathcal{T}$ , a Horn formula  $\mathcal{H}$  is created according to the method introduced in EL<sup>+</sup>SAT [30, 31]. To this end, each axiom  $a_i \in \mathcal{T}$  is initially assigned a unique selector variable  $s_{[a_i]}$ . The classification of  $\mathcal{T}$  is done in two phases [4, 6].

First,  $\mathcal{T}$  is normalized so that each of its axioms are of the form (i)  $(A_1 \sqcap \dots \sqcap A_k) \sqsubseteq B$  ( $k \geq 1$ ), (ii)  $A \sqsubseteq \exists r.B$ , (iii)  $\exists r.A \sqsubseteq B$ , or (iv)  $r_1 \circ \dots \circ r_n \sqsubseteq s$  ( $n \geq 1$ ), where  $A, A_i, B \in \mathbf{N}_C$  and  $r, r_i, s \in \mathbf{N}_R$ . This process results in a TBox  $\mathcal{T}_N$  where each axiom  $a_i \in \mathcal{T}$  is substituted by a set of axioms in normal form  $\{a_{i1}, \dots, a_{im_i}\}$ . At this point, the clauses  $s_{[a_i]} \rightarrow s_{[a_{ik}]}$ , with  $1 \leq k \leq m_i$ , are added to  $\mathcal{H}$ .

Second,  $\mathcal{T}_N$  is saturated through the exhaustive application of the *completion rules* shown in Table 1, resulting in the extended TBox  $\mathcal{T}'$ . Each of the rows in Table 1 constitute a completion rule. Their application is sound and complete for inferring subsumptions [4]. Whenever a rule  $r$  can be applied (with antecedents  $\text{ant}(r)$ ) leading to inferring an axiom  $a_i$ , the Horn clause  $(\bigwedge_{\{a_j \in \text{ant}(r)\}} s_{[a_j]}) \rightarrow s_{[a_i]}$  is added to  $\mathcal{H}$ .

As a result,  $\mathcal{H}$  will eventually encode all possible derivations of completion rules inferring any axiom such that  $X \sqsubseteq_{\mathcal{T}} Y$ , with  $X, Y \in \mathbf{N}_C$ .

### 3.2 Generation of Group Horn Formulae

After classifying  $\mathcal{T}$ , some axioms  $C \sqsubseteq D$  may be included in  $\mathcal{T}'$  for which a justification or diagnosis may be required. Each of these *queries* will result in a group Horn formula defined as:  $\mathcal{H}_G = \{\mathcal{G}_0, \mathcal{G}_1, \dots, \mathcal{G}_{|\mathcal{T}|}\}$ , where  $\mathcal{G}_0 = \mathcal{H} \cup \{(\neg s_{[C \sqsubseteq D]})\}$  and for each axiom  $a_i$  ( $i > 0$ ) in the original TBox  $\mathcal{T}$ , group  $\mathcal{G}_i = \{(s_{[a_i]})\}$  is defined with a single unit clause.  $\mathcal{H}_G$  is unsatisfiable and, as shown in [1, 2], its

**Table 1.**  $\mathcal{EL}^+$  completion rules

Preconditions		Inferred axiom
$A \sqsubseteq A_i, 1 \leq i \leq n$	$A_1 \sqcap \dots \sqcap A_n \sqsubseteq B$	$A \sqsubseteq B$
$A \sqsubseteq A_1$	$A_1 \sqsubseteq \exists r.B$	$A \sqsubseteq \exists r.B$
$A \sqsubseteq \exists r.B, B \sqsubseteq B_1$	$\exists r.B_1 \sqsubseteq B_2$	$A \sqsubseteq B_2$
$A_{i-1} \sqsubseteq \exists r_i.A_i, 1 \leq i \leq n$	$r_1 \circ \dots \circ r_n \sqsubseteq r$	$A_0 \sqsubseteq \exists r.A_n$

**Algorithm 1.** EMUS [26] / MARCO [18]

---

**Input:**  $\mathcal{F}$  a CNF formula  
**Output:** Reports the set of MUSes (and MCSes) of  $\mathcal{F}$

```

1   $\langle I, \mathcal{Q} \rangle \leftarrow \langle \{p_i \mid c_i \in \mathcal{F}\}, \emptyset \rangle$            // Variable  $p_i$  picks clause  $c_i$ 
2  while true do
3     $(st, P) \leftarrow \text{MaximalModel}(\mathcal{Q})$ 
4    if not  $st$  then return
5     $\mathcal{F}' \leftarrow \{c_i \mid p_i \in P\}$                                 // Pick selected clauses
6    if not  $\text{SAT}(\mathcal{F}')$  then
7       $\mathcal{M} \leftarrow \text{ComputeMUS}(\mathcal{F}')$ 
8       $\text{ReportMUS}(\mathcal{M})$ 
9       $b \leftarrow \{\neg p_i \mid c_i \in \mathcal{M}\}$           // Negative clause blocking the MUS
10   else
11      $\text{ReportMCS}(\mathcal{F} \setminus \mathcal{F}')$ 
12      $b \leftarrow \{p_i \mid p_i \in I \setminus P\}$           // Positive clause blocking the MCS
13    $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{b\}$ 

```

---

group-MUSes correspond to the MinAs for  $C \sqsubseteq_{\mathcal{T}} D$ . Equivalently, due to the hitting set duality for MinAs/diagnoses, which also holds for MUSes/MCSes, group-MCSes of  $\mathcal{H}_G$  correspond to diagnoses for  $C \sqsubseteq_{\mathcal{T}} D$ .

BEACON simplifies  $\mathcal{H}_G$  with the techniques introduced in [30, 31], which often reduce the formulas to a great extent.

### 3.3 Computation of Group-MUSes/Group-MCSes

For enumerating group-MUSes and group-MCSes of the formula  $\mathcal{H}_G$  defined above, BEACON integrates the state-of-the-art HGMUS enumerator [2]. HGMUS exploits hitting set dualization between (group) MCSes and (group) MUSes and, hence, it shares ideas also explored in MaxHS [11], EMUS/MARCO [17, 26], among others. As shown in Algorithm 1, these methods rely on a two (SAT) solvers approach. Formula  $\mathcal{Q}$  is defined over a set of selector variables corresponding to clauses in  $\mathcal{F}$ , and it is used to enumerate subsets of  $\mathcal{F}$ . Iteratively, the algorithm computes a maximal model  $P$  of  $\mathcal{Q}$  and tests whether the subformula  $\mathcal{F}' \subseteq \mathcal{F}$  containing the clauses associated to  $P$  is satisfiable. If it is,  $\mathcal{F} \setminus \mathcal{F}'$  is an MCS of  $\mathcal{F}$ . Otherwise,  $\mathcal{F}'$  is reduced to an MUS. MCSes and MUSes are blocked adding clauses to  $\mathcal{Q}$ .

HGMUS shares the main organization of Algorithm 1, with  $\mathcal{F} = \mathcal{G}_0$  and  $\mathcal{Q}$  defined over selector variables for groups  $\mathcal{G}_i$  of  $\mathcal{H}_G$ , with  $i > 0$ . It also includes some specific features. First, it uses the Horn satisfiability algorithhm LTUR [24]. Besides, it integrates a dedicated insertion-based MUS extractor as well as an efficient algorithm for computing maximal models based on a reduction to computing MCSes [23].

### 3.4 BEACON’s Additional Specific Features

Besides computing MinAs/diagnoses, BEACON offers additional functionalities.

**Diagnosing Multiple Subsumption Relations at a Time.** After classifying  $\mathcal{T}$ , there could be several *unintended* subsumption relations  $C_i \sqsubseteq_{\mathcal{T}} D_i$  that need to be removed. BEACON allows for diagnosing this multiple unintended inferences at the same time. By adding the unit clauses ( $\neg s_{[C_i \sqsubseteq D_i]}$ ) to  $\mathcal{G}_0$  in  $\mathcal{H}_G$ , each computed group-MCS corresponds to a diagnosis that would eliminate *all* the indicated subsumption relations.

**Computing Smallest MinAs.** Alternatively to enumerating all the possible MinAs, one may want to compute only those of the minimum possible size. To enable this functionality, BEACON integrates a state-of-the-art solver for the smallest MUS problem (SMUS) called FORQES [14]. The decision version of the SMUS problem is known to be  $\Sigma_2^P$ -complete (e.g. see [13, 16]). As HGMUS, FORQES is based on the hitting set dualization between (group) MUSes and (group) MCSes. The tool iteratively computes *minimum* hitting sets of a set of MCSes of a formula detected so far. While these minimum hitting sets are satisfiable, they are grown into an MSS, whose complement is an MCS which is added to the set of MCSes. The process terminates when an unsatisfiable minimum hitting set is identified, representing a smallest MUS of the formula.

## 4 Experimental Results

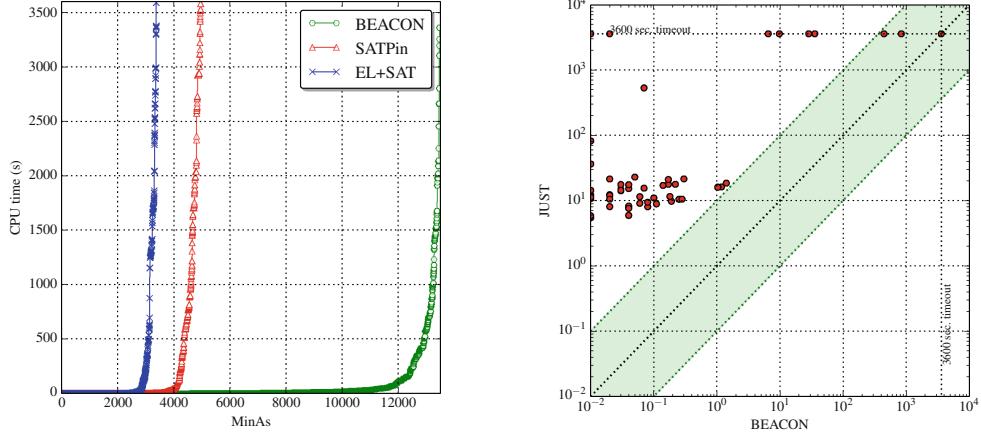
This section reports a summary of results that illustrates the performance of BEACON<sup>1</sup> w.r.t. other  $\mathcal{EL}^+$  axiom pinpointing tools in the literature. It also provides information on its capability of computing diagnoses and enumerating smallest MinAs.

The experiments were run on a Linux cluster (2Ghz) with a limit of 3600s and 4Gbyte, considering 500 subsumption relations from five well-known  $\mathcal{EL}^+$  bio-medical ontologies: GALEN [27] (FULL-GALEN and NOT-GALEN), Gene [3], NCI [32] and SNOMED-CT [34]. The experiments use Horn formulae encoded by EL<sup>+</sup>SAT [30, 31] applying the reduction techniques that BEACON incorporates by default. These formulae are fed to BEACON’s engines, namely HGMUS and FORQES.

The results reported focus on HGMUS and FORQES. Due to lack of space, running times for classifying the ontologies and formula reduction are not reported. Classification is done in polynomial time once for each ontology, so it is amortized among all queries for the ontology. Formula reduction usually takes very short time. Detailed results are available with the distribution of BEACON, including an analysis on the size of the Horn formulae and the reductions achieved.

---

<sup>1</sup> Available at <http://logos.ucd.ie/web/doku.php?id=beacon-tool>.



**Fig. 2.** Plots comparing BEACON to  $\text{EL}^+\text{SAT}$ , SATPin and JUST

**Axiom Pinpointing.** BEACON shows significant improvements over the existing tools  $\text{EL}^+\text{SAT}$  [30, 31], SATPin [21], EL2MCS [1], CEL [5] and JUST [20]. BEACON often achieves remarkable reductions in the running times, and exhibits a clear superiority in enumerating MinAs for 19 very hard instances that cannot be solved by a time limit of 3600 s. This is illustrated in Fig. 2. The cactus plot shows the number of MinAs reported over time. BEACON computes much more MinAs faster than other tools. The scatter plot compares BEACON with JUST regarding the running times on a subset of the instances JUST can cope with. BEACON shows a significant performance gap. Similar results have been observed for EL2MCS and CEL [2].

**Computing Diagnoses.** For all solved instances (481 out of 500), BEACON enumerates all diagnoses, where its number ranges from 2 to 565409. Interestingly, for the 19 aborted instances, the number of reported diagnoses ranges from 1011164 to 1972324. These numbers illustrate the efficiency of BEACON at computing diagnoses, and explain the difficulty of these aborted instances. Of the other tools, only EL2MCS reports diagnoses, which, for hard instances, computes around 33 % fewer diagnoses.

**Computing Smallest MinAs.** The last experiments consider the 19 instances for which BEACON is unable to enumerate all MinAs. Notably, BEACON is very efficient at computing the smallest MinAs using FORQES. In all cases, each set of smallest MinAs is computed in negligible time (less than 0.1 s). The sizes of the smallest MinAs range from 5 to 13 axioms, and their number ranges from 1 to 7.

## 5 Conclusions

This paper describes BEACON, an axiom pinpointing tool for the  $\mathcal{EL}$ -family of DLs. BEACON integrates HGMUS [2], a group MUS enumerator for

propositional Horn formulae, with a dedicated front-end, interfacing a target ontology, and generating group Horn formulae for HGMUS. Besides enumerating MinAs (and associated diagnoses), BEACON enables the simultaneous diagnosis of multiple inferences, and the computation of the smallest MinA (or smallest MUS [14]). The experimental results indicate that the computation of the smallest MinA is very efficient in practice, in addition to the already known top performance of HGMUS.

**Acknowledgement.** This work was funded in part by SFI grant BEACON (09/IN.1/-I2618), by DFG grant DFG HO 1294/11-1, and by Spanish grant TIN2013-46511-C2-2-P. The contribution of the researchers associated with the SFI grant BEACON is also acknowledged.

## References

1. Arif, M.F., Mencía, C., Marques-Silva, J.: Efficient axiom pinpointing with EL2MCS. In: Hölldobler, S., Krötzsch, M., Peñaloza, R., Rudolph, S., Edelkamp, S., Edelkamp, S. (eds.) KI 2015. LNCS, vol. 9324, pp. 225–233. Springer, Heidelberg (2015). doi:[10.1007/978-3-319-24489-1\\_17](https://doi.org/10.1007/978-3-319-24489-1_17)
2. Arif, M.F., Mencía, C., Marques-Silva, J.: Efficient MUS enumeration of Horn formulae with applications to axiom pinpointing. In: Heule, M., Weaver, S. (eds.) SAT 2015. LNCS, vol. 9340, pp. 324–342. Springer, Heidelberg (2015). doi:[10.1007/978-3-319-24318-4\\_24](https://doi.org/10.1007/978-3-319-24318-4_24)
3. Ashburner, M., Ball, C.A., Blake, J.A., Botstein, D., Butler, H., Cherry, J.M., Davis, A.P., Dolinski, K., Dwight, S.S., Eppig, J.T., et al.: Gene ontology: tool for the unification of biology. *Nat. Genet.* **25**(1), 25–29 (2000)
4. Baader, F., Brandt, S., Lutz, C.: Pushing the  $\mathcal{EL}$  envelope. In: IJCAI, pp. 364–369 (2005)
5. Baader, F., Lutz, C., Suntisrivaraporn, B.: CEL — a polynomial-time reasoner for life science ontologies. In: Furbach, U., Shankar, N. (eds.) IJCAR 2006. LNCS (LNAI), vol. 4130, pp. 287–291. Springer, Heidelberg (2006)
6. Baader, F., Peñaloza, R., Suntisrivaraporn, B.: Pinpointing in the description logic  $\mathcal{EL}^+$ . In: KI, pp. 52–67 (2007)
7. Baader, F., Suntisrivaraporn, B.: Debugging SNOMED CT using axiom pinpointing in the description logic  $\mathcal{EL}^+$ . In: KR-MED (2008)
8. Bailey, J., Stuckey, P.J.: Discovery of minimal unsatisfiable subsets of constraints using hitting set dualization. In: Hermenegildo, M.V., Cabeza, D. (eds.) PADL 2004. LNCS, vol. 3350, pp. 174–186. Springer, Heidelberg (2005)
9. Biere, A., Heule, M., van Maaren, H., Walsh, T. (eds.): Handbook of Satisfiability. Frontiers in Artificial Intelligence and Applications, vol. 185. IOS Press, Amsterdam (2009)
10. Birnbaum, E., Lozinskii, E.L.: Consistent subsets of inconsistent systems: structure and behaviour. *J. Exp. Theor. Artif. Intell.* **15**(1), 25–46 (2003)
11. Davies, J., Bacchus, F.: Solving MAXSAT by solving a sequence of simpler SAT instances. In: Lee, J. (ed.) CP 2011. LNCS, vol. 6876, pp. 225–239. Springer, Heidelberg (2011)
12. Dowling, W.F., Gallier, J.H.: Linear-time algorithms for testing the satisfiability of propositional Horn formulae. *J. Log. Program.* **1**(3), 267–284 (1984)

13. Gupta, A.: Learning Abstractions for Model Checking. Ph.D. thesis, Carnegie Mellon University, June 2006
14. Ignatiev, A., Previti, A., Liffiton, M., Marques-Silva, J.: Smallest MUS extraction with minimal hitting set dualization. In: Pesant, G. (ed.) CP 2015. LNCS, vol. 9255, pp. 173–182. Springer, Heidelberg (2015)
15. Itai, A., Makowsky, J.A.: Unification as a complexity measure for logic programming. *J. Log. Program.* **4**(2), 105–117 (1987)
16. Liberatore, P.: Redundancy in logic I: CNF propositional formulae. *Artif. Intell.* **163**(2), 203–232 (2005)
17. Liffiton, M.H., Malik, A.: Enumerating infeasibility: finding multiple MUSes quickly. In: Gomes, C., Sellmann, M. (eds.) CPAIOR 2013. LNCS, vol. 7874, pp. 160–175. Springer, Heidelberg (2013)
18. Liffiton, M.H., Previti, A., Malik, A., Marques-Silva, J.: Fast, flexible MUs enumeration. *Constraints* (2015). Online version: <http://link.springer.com/article/10.1007/s10601-015-9183-0>
19. Liffiton, M.H., Sakallah, K.A.: Algorithms for computing minimal unsatisfiable subsets of constraints. *J. Autom. Reasoning* **40**(1), 1–33 (2008)
20. Ludwig, M.: Just: a tool for computing justifications w.r.t. ELH ontologies. In: ORE (2014)
21. Manthey, N., Peñaloza, R.: Exploiting SAT technology for axiom pinpointing. Technical report LTCS 15–05, Chair of Automata Theory, Institute of Theoretical Computer Science, Technische Universität Dresden, April 2015. <https://dpll.inf.tu-dresden.de/web/Techreport3010>
22. Marques-Silva, J., Heras, F., Janota, M., Previti, A., Belov, A.: On computing minimal correction subsets. In: IJCAI, pp. 615–622 (2013)
23. Mencía, C., Previti, A., Marques-Silva, J.: Literal-based MCS extraction. In: IJCAI, pp. 1973–1979 (2015)
24. Minoux, M.: LTUR: A simplified linear-time unit resolution algorithm for Horn formulae and computer implementation. *Inf. Process. Lett.* **29**(1), 1–12 (1988)
25. Peñaloza, R.: Axiom pinpointing in description logics and beyond. Ph.D. thesis, Dresden University of Technology (2009)
26. Previti, A., Marques-Silva, J.: Partial MUS enumeration. In: AAAI, pp. 818–825 (2013)
27. Rector, A.L., Horrocks, I.R.: Experience building a large, re-usable medical ontology using a description logic with transitivity and concept inclusions. In: Workshop on Ontological Engineering, pp. 414–418 (1997)
28. Reiter, R.: A theory of diagnosis from first principles. *Artif. Intell.* **32**(1), 57–95 (1987)
29. Schlobach, S., Cornet, R.: Non-standard reasoning services for the debugging of description logic terminologies. In: IJCAI, pp. 355–362 (2003)
30. Sebastiani, R., Vescovi, M.: Axiom pinpointing in lightweight description logics via Horn-SAT encoding and conflict analysis. In: Schmidt, R.A. (ed.) CADE-22. LNCS, vol. 5663, pp. 84–99. Springer, Heidelberg (2009)
31. Sebastiani, R., Vescovi, M.: Axiom pinpointing in large  $\mathcal{EL}^+$  ontologies via SAT and SMT techniques. Technical report DISI-15-010, DISI, University of Trento, Italy, Under Journal Submission, April 2015. [http://disi.unitn.it/~rseba/elsat/elsat\\_techrep.pdf](http://disi.unitn.it/~rseba/elsat/elsat_techrep.pdf)
32. Sioutos, N., de Coronado, S., Haber, M.W., Hartel, F.W., Shaiu, W., Wright, L.W.: NCI thesaurus: A semantic model integrating cancer-related clinical and molecular information. *J. Biomed. Inform.* **40**(1), 30–43 (2007)

33. Slaney, J.: Set-theoretic duality: A fundamental feature of combinatorial optimisation. In: ECAI, pp. 843–848 (2014)
34. Spackman, K.A., Campbell, K.E., Côté, R.A.: SNOMED RT: a reference terminology for health care. In: AMIA (1997)
35. Stefan, S., Ronald, C., Spackman, K.A.: Consolidating SNOMED CT's ontological commitment. *Appl. Ontol.* **6**, 111 (2011)

## Chapter 7

### JAIR '16 Paper

- Citation: M. Fareed Arif, Carlos Mencía, Norbert Manthey, Alexey Ignatiev, Rafael Pernaloza, and Joao Marques-Silva. Towards Efficient SAT-Based Axiom Pinpointing in Lightweight Description Logics. *Journal of Artificial Intelligence Research (JAIR)*, 2016. Submitted - Under Review
- Type: Journal Paper (Submitted)
- It is submitted for an A\* journal paper, currently under review.
- Contribution: This paper consolidate the previously published work [[AMMS15a](#), [AMMS15b](#), [AMI<sup>+</sup>16](#)]. The candidate contributed in conception of the main idea and provided insightful suggestions.

# Towards Efficient SAT-Based Axiom Pinpointing in Lightweight Description Logics

**M. Fareed Arif**

*University College Dublin, Ireland*

MUHAMMAD.ARIF.1@UCDCONNECT.IE

**Carlos Mencía**

*University of Oviedo, Spain*

CMENCIA@GMAIL.COM

**Norbert Manthey**

*TU Dresden, Germany*

NORBERT.MANTHEY@TU-DRESDEN.DE

**Alexey Ignatiev**

*University of Lisbon, Portugal*

*ISDCT SB RAS, Irkutsk, Russia*

AIGNATIEV@CIENCIAS.ULISBOA.PT

**Rafael Peñaloza**

*Free University of Bozen-Bolzano, Italy*

RAFAEL.PELANOZA@UNIBZ.IT

**Joao Marques-Silva**

*University of Lisbon, Portugal*

JPMS@CIENCIAS.ULISBOA.PT

## Abstract

Description Logics (DLs) are knowledge representation formalisms that have been developed to represent the terminological knowledge of an application domain through the notion of concepts. Among these, the lightweight description logics, such as the  $\mathcal{EL}$  family of DLs (including  $\mathcal{EL}^+$ ) stand out due to the availability of polynomial-time reasoning algorithms and the fact that, although expressive, they suffice to represent knowledge in several important application domains, such as medical informatics. Building ontologies is an error-prone task, which often results in inconsistencies and unintended inferences. Understanding the causes of (unintended) inferences, and repairing them if necessary is a central task in ontology debugging, which in turn is necessary in the development of reliable ontology-based systems. Axiom pinpointing refers to the task of explaining the causes of an (unintended) subsumption relation to follow from an ontology. More concretely, it consists in computing irreducible subsets of the ontology, called minimal axiom sets (MinAs), which entail the given subsumption relation to be explained. Over the last years, there has been a large body of research in axiom pinpointing, most of the approaches built on specific DL techniques. Notably, recent work has shown that axiom pinpointing in the  $\mathcal{EL}$  family of DLs can be addressed in the propositional domain. This earlier approach relies on encoding axiom pinpointing problems into a Horn formula, and applying propositional satisfiability (SAT) techniques for computing the MinAs. This paper builds on this earlier work and proposes very efficient SAT-based approaches for axiom pinpointing in the  $\mathcal{EL}$  family of DLs. It first reduces the problem of enumerating MinAs to the problem of enumerating the group-MUSes (minimal unsatisfiable subsets) of a Horn formula. Then, it builds on recent work on MUS enumeration and proposes two algorithms based on explicit and implicit hitting set dualization between the group-MUSes and group-MCSes (minimal correction subsets) of the Horn formula. Experimental results on well-known benchmarks derived from medical ontologies show categorical improvements over existing approaches to axiom pinpointing in the  $\mathcal{EL}$  family of DLs.

## Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Preliminaries</b>	<b>5</b>
2.1	The Lightweight Description Logic $\mathcal{EL}^+$	5
2.2	Propositional Satisfiability	6
<b>3</b>	<b>SAT-Based Axiom Pinpointing</b>	<b>9</b>
3.1	Generation of Horn Formulae	9
3.1.1	Classification and Horn Encoding	9
3.1.2	Axiom Pinpointing Instances	10
3.2	$\text{EL}^+$ SAT Computation of MinAs	11
<b>4</b>	<b>Towards Efficient SAT-Based Axiom Pinpointing</b>	<b>12</b>
4.1	Axiom Pinpointing as Group-MUS Enumeration	12
4.2	Group-MUS Enumeration by Explicit Hitting Set Dualization	14
4.2.1	Axiom Pinpointing Using MaxSAT	15
4.2.2	EL2MCS Tool	15
4.3	Group-MUS Enumeration by Implicit Hitting Set Dualization	16
4.3.1	Organization of HGMUS	17
4.3.2	Computing Maximal Models	17
4.3.3	Adding Blocking Clauses	18
4.3.4	Deciding Satisfiability of Horn Formulae	19
4.3.5	MUS Extraction in Horn Formulae	21
<b>5</b>	<b>Experimental Results</b>	<b>22</b>
5.1	Experimental Setup	22
5.2	Assessment of SAT-Based Approaches	23
5.2.1	COI Instances	24
5.2.2	x2 Instances	26
5.3	Assessment of Other Axiom Pinpointing Tools	28
<b>6</b>	<b>Conclusions</b>	<b>28</b>
<b>References</b>		<b>30</b>

## 1. Introduction

Description Logics (DLs) are a well-known family of logic-based knowledge representation formalisms (Baader, Calvanese, McGuinness, Nardi, & Patel-Schneider, 2003). DLs find a wide range of applications in computer science, including the semantic web and representation of ontologies, in particular for the life sciences. A number of reasoning tasks are associated with DLs. Among these, axiom pinpointing corresponds to the problem of computing one (or all) minimal set of axioms (called *MinA*), which explains a subsumption relation in an ontology (Schlobach & Cornet, 2003). Example applications of axiom pinpointing include context-based reasoning, error-tolerant reasoning (Ludwig & Peñaloza, 2014), and ontology debugging and revision (Schlobach, Huang, Cornet, & van Harmelen, 2007; Kalyanpur, Parsia, Sirin, & Grau, 2006), among many others. Axiom pinpointing for different description logics (DLs) has been studied extensively for more than a decade, with related work and techniques arising from the mid 90s (Baader & Hollunder, 1995; Schlobach & Cornet, 2003; Parsia, Sirin, & Kalyanpur, 2005; Meyer, Lee, Booth, & Pan, 2006; Baader, Lutz, & Suntisrivaraporn, 2006; Baader, Peñaloza, & Suntisrivaraporn, 2007; Kalyanpur, Parsia, Horridge, & Sirin, 2007; Sirin, Parsia, Grau, Kalyanpur, & Katz, 2007; Schlobach et al., 2007; Baader & Suntisrivaraporn, 2008; Sebastiani & Vescovi, 2009; Baader & Peñaloza, 2010; Moodley, Meyer, & Varzinczak, 2011; Nguyen, Alechina, & Logan, 2012; Ludwig, 2014; Sebastiani & Vescovi, 2015; Manthey & Peñaloza, 2015; Manthey, Peñaloza, & Rudolph, 2016).

The  $\mathcal{EL}$  family of DLs contains lightweight languages that are well-known for having tractable reasoning problems. Despite being expressive, the DL  $\mathcal{EL}^+$ , has been successfully used for representing ontologies in the medical sciences, including the very large and extensively used SNOMED CT ontology (Spackman, Campbell, & Côté, 1997). Work on axiom pinpointing for the  $\mathcal{EL}$  family of DLs can be traced back to 2007, when several methods were developed and implemented in the CEL tool (Baader et al., 2006). An alternative approach based on the application of SAT solvers was later proposed (Sebastiani & Vescovi, 2009; Vescovi, 2011; Sebastiani & Vescovi, 2015). This seminal work proposed encoding the standard reasoning methods for  $\mathcal{EL}^+$  through a propositional Horn formula and apply methods from SAT to trace the causes of a consequence. This approach was implemented in the EL<sup>+</sup>SAT system (Sebastiani & Vescovi, 2009; Vescovi, 2011; Sebastiani & Vescovi, 2015), which was shown to consistently outperform CEL in axiom-pinpointing tasks. In (Manthey & Peñaloza, 2015; Manthey et al., 2016), the same ideas were further improved through the inclusion of some optimization methods developed in the DL community, and others created *ad-hoc* to exploit the shape of the constructed Horn formula.

In this paper, we build on this earlier work and make several contributions towards efficient SAT-based axiom pinpointing in the  $\mathcal{EL}$  family of Description Logics. First, an important relationship between axiom pinpointing and the enumeration of minimal unsatisfiable subsets (MUSes), or more precisely group-MUSes, of a propositional Horn formula is identified, which serves to capitalize on a large body of recent work on MUS enumeration in the SAT domain. Then, the paper proposes two algorithms for this task that exploit the well-known minimal hitting set duality relationship between MCSes (minimal correction subsets) and MUSes of a propositional formula.

The first method, called EL2MCS (Arif, Mencía, & Marques-Silva, 2015a), exploits hitting set dualization explicitly. More precisely, it first computes the group-MCSes of the Horn formula and then obtains the group-MUSes by computing minimal hitting sets of the set of group-MCSes. The main contribution of this paper is the development of the second method: an efficient group-MUS enumerator for Horn formulae, referred to as HGMUS (Arif, Mencía, & Marques-Silva, 2015b), that does not need to pre-compute the MCSes before-hand. Exploiting the translation from (Sebastiani & Vescovi, 2009), this method finds immediate application in axiom pinpointing in  $\mathcal{EL}^+$ . In the process of developing HGMUS, the paper also identifies important performance bottlenecks of existing solutions, and in particular of the previous tool EL<sup>+</sup>SAT (Sebastiani & Vescovi, 2009, 2015). The new group-MUS enumerator for Horn formulae builds on the large body of recent work on problem solving with SAT oracles. This includes, among others, MUS extraction (Belov, Lynce, & Marques-Silva, 2012), MCS extraction and enumeration (Marques-Silva, Heras, Janota, Previti, & Belov, 2013a), and partial MUS enumeration (Previti & Marques-Silva, 2013; Liffiton & Malik, 2013; Liffiton, Previti, Malik, & Marques-Silva, 2016). HGMUS also exploits earlier work on solving Horn propositional formulae (Dowling & Gallier, 1984; Minoux, 1988), and develops novel algorithms for MUS extraction in propositional Horn formulae.

HGMUS was extensively tested over a large set of well-known problem instances that have been used to evaluate axiom-pinpointing methods for  $\mathcal{EL}^+$  ontologies. The experimental results obtained demonstrate a conclusive improvement in the performance of the new tool over all other previously existing approaches. In most cases, this improvement could be measured by several orders of magnitude.

An outcome of this work is the new axiom pinpointing tool BEACON (Arif, Mencía, Ignatiev, Manthey, Peñaloza, & Marques-Silva, 2016), that integrates a translator from  $\mathcal{EL}^+$  to Horn formulae, a group-MUS enumerator for Horn formulae HGMUS, but also FORQES (Ignatiev, Previti, Liffiton, & Marques-Silva, 2015) for computing smallest MUSes and by extension also smallest MinAs.

The rest of this paper is organized as follows. [Section 2](#) introduces the notation and definitions used throughout the paper. [Section 3](#) overviews past work on applying SAT for axiom pinpointing. Afterwards, [Section 4](#) proposes a number of algorithmic improvements for SAT-based axiom pinpointing. These improvements are motivated by the analysis of EL<sup>+</sup>SAT. Experimental results on well-known problem instances from axiom pinpointing for the  $\mathcal{EL}$  family of DLs are analyzed in [Section 5](#), before providing conclusions and an outlook on future work in [Section 6](#).

## 2. Preliminaries

This section introduces some basic concepts and notation regarding the lightweight description logic  $\mathcal{EL}^+$  and propositional satisfiability (SAT), with special emphasis on the notions used throughout the paper.

### 2.1 The Lightweight Description Logic $\mathcal{EL}^+$

$\mathcal{EL}^+$  (Baader, Brandt, & Lutz, 2005) is a lightweight DL that has been successfully used to build large ontologies, most notably from the bio-medical domains. As with all DLs, the main elements in  $\mathcal{EL}^+$  are concepts that describe classes of individuals from the knowledge domain. Formally,  $\mathcal{EL}^+$  *concepts* are built from two disjoint sets  $N_C$  and  $N_R$  of *concept names* and *role names* through the grammar rule

$$C ::= A \mid \top \mid C \sqcap C \mid \exists r.C,$$

where  $A \in N_C$  and  $r \in N_R$ . The concept  $\top$  is referred to as the *top* concept.

The knowledge of the domain is stored in a *TBox* (or *ontology*), which is a finite set of *general concept inclusions* (GCIs)  $C \sqsubseteq D$ , where  $C$  and  $D$  are  $\mathcal{EL}^+$  concepts, and *role inclusions* (RIs)  $r_1 \circ \dots \circ r_n \sqsubseteq s$ , where  $n \geq 1$  and  $r_i, s \in N_R$ . We will often use the term *axiom* to refer to both GCIs and RIs.<sup>1</sup>

The semantics of this logic is based on *interpretations*, which are pairs of the form  $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$  where  $\Delta^\mathcal{I}$  is a non-empty set called the *domain* and  $\cdot^\mathcal{I}$  is the *interpretation function* that maps every concept name  $A \in N_C$  to a set  $A^\mathcal{I} \subseteq \Delta^\mathcal{I}$  and every role name  $r \in N_R$  to a binary relation  $r^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I}$ . The interpretation function is extended to arbitrary  $\mathcal{EL}^+$  concepts inductively, defining  $\top^\mathcal{I} := \Delta^\mathcal{I}$ ,  $(C \sqcap D)^\mathcal{I} := C^\mathcal{I} \cap D^\mathcal{I}$ , and  $(\exists r.C)^\mathcal{I} := \{\delta \in \Delta^\mathcal{I} \mid \exists \gamma \in \Delta^\mathcal{I}. (\delta, \gamma) \in r^\mathcal{I}, \gamma \in C^\mathcal{I}\}$ . The interpretation  $\mathcal{I}$  *satisfies* the GCI  $C \sqsubseteq D$  iff  $C^\mathcal{I} \subseteq D^\mathcal{I}$ ; it *satisfies* the RI  $r_1 \circ \dots \circ r_n \sqsubseteq s$  iff  $r_1^\mathcal{I} \circ \dots \circ r_n^\mathcal{I} \subseteq s^\mathcal{I}$ , where  $\circ$  denotes the usual composition of binary relations. We say that  $\mathcal{I}$  is a *model* of the TBox  $\mathcal{T}$  iff  $\mathcal{I}$  satisfies all the GCIs and RIs in  $\mathcal{T}$ .

Since every  $\mathcal{EL}^+$  TBox is consistent, the main reasoning problem of interest in this logic is to decide subsumption between concepts. We say that a concept  $C$  is *subsumed* by  $D$  w.r.t. the TBox  $\mathcal{T}$  (denoted by  $C \sqsubseteq_{\mathcal{T}} D$ ) if for every model  $\mathcal{I}$  of  $\mathcal{T}$  it holds that  $C^\mathcal{I} \subseteq D^\mathcal{I}$ . It is well known that subsumption can be restricted w.l.o.g. to subsumption between *concept names*. Hence, for the nonce we will focus on this case only. A generalization of this problem is *classification*, which refers to the task of deciding all the subsumption relations between concept names appearing in the TBox  $\mathcal{T}$ . In the case of  $\mathcal{EL}^+$ , both concept subsumption and classification can be solved in polynomial time (Baader et al., 2005).

**Example 1.** Consider the TBox  $\mathcal{T}_{\text{exa}} = \{A \sqsubseteq \exists r.A, A \sqsubseteq Y, \exists r.Y \sqsubseteq B, Y \sqsubseteq B, B \sqsubseteq C\}$ . Through the transitivity of the subset relation  $\subseteq$ , it is easy to see that  $A \sqsubseteq_{\mathcal{T}_{\text{exa}}} B$  holds. Moreover, classification of  $\mathcal{T}_{\text{exa}}$  yields the subsumption relations

$$\{A \sqsubseteq Y, \quad Y \sqsubseteq B, \quad B \sqsubseteq C, \quad A \sqsubseteq B, \quad A \sqsubseteq C, \quad Y \sqsubseteq C\}.$$

---

1. For the scope of this paper, we do not consider individual names or assertional axioms (ABoxes), which are sometimes included in the definition of an ontology (Baader et al., 2003).

Rather than merely deciding whether a subsumption relation follows from a TBox, we are interested in understanding the causes of this consequence, and repairing it if necessary.

**Definition 1** (MinA, diagnosis, repair). *A minimal axiom set (MinA) for  $C \sqsubseteq D$  w.r.t. the TBox  $\mathcal{T}$  is a minimal subset (w.r.t. set inclusion)  $\mathcal{M} \subseteq \mathcal{T}$  such that  $C \sqsubseteq_{\mathcal{M}} D$ . A diagnosis for  $C \sqsubseteq D$  w.r.t.  $\mathcal{T}$  is a minimal subset (w.r.t. set inclusion)  $\mathcal{D} \subseteq \mathcal{T}$  such that  $C \not\sqsubseteq_{\mathcal{T} \setminus \mathcal{D}} D$ . A repair for  $C \sqsubseteq D$  w.r.t.  $\mathcal{T}$  is a maximal subset (w.r.t. set inclusion)  $\mathcal{R} \subseteq \mathcal{T}$  such that  $C \not\sqsubseteq_{\mathcal{R}} D$ .*

Informally, given a subsumption relation that follows from a TBox  $\mathcal{T}$ , a MinA is an irreducible subset of  $\mathcal{T}$  still entailing the subsumption relation, i.e. MinAs constitute *explanations* or *justifications* for the derivation of the subsumption. A diagnosis is an irreducible set of axioms of  $\mathcal{T}$  such that, if removed from  $\mathcal{T}$ , the resulting TBox no longer entails the subsumption relation, and a repair is a maximal subset of  $\mathcal{T}$  from which the subsumption relation does not follow. In other words, diagnoses and repairs represent possible ways of removing a consequence from a TBox  $\mathcal{T}$  such that only the strictly necessary axioms are dropped from  $\mathcal{T}$ .

Note that given a diagnosis  $\mathcal{D} \subseteq \mathcal{T}$  for a given subsumption relation w.r.t.  $\mathcal{T}$ , its complement (i.e.  $\mathcal{T} \setminus \mathcal{D}$ ) constitutes a repair, and *vice versa*. Besides, it is well known that the set of all MinAs can be obtained from the set of all diagnoses, and *vice versa*, through minimal hitting set computation (Reiter, 1987; Schlobach & Cornet, 2003). In the worst case, there can be an exponential number of MinAs and diagnoses for a given subsumption relation w.r.t. a TBox (Baader et al., 2007; Peñaloza, 2009).

**Example 2.** Consider again the TBox  $\mathcal{T}_{\text{exa}}$  from Example 1. There are two MinAs for  $A \sqsubseteq B$ , namely  $\mathcal{M}_1 = \{A \sqsubseteq Y, Y \sqsubseteq B\}$ , and  $\mathcal{M}_2 = \{A \sqsubseteq \exists r.A, A \sqsubseteq Y, \exists r.Y \sqsubseteq B\}$ . The diagnoses for this subsumption are  $\mathcal{D}_1 = \{A \sqsubseteq Y\}$ ,  $\mathcal{D}_2 = \{A \sqsubseteq \exists r.A, Y \sqsubseteq B\}$ , and  $\mathcal{D}_3 = \{\exists r.Y \sqsubseteq B, Y \sqsubseteq B\}$ , which correspond to the hitting sets of  $\{\mathcal{M}_1, \mathcal{M}_2\}$ . Finally, the repairs for  $A \sqsubseteq B$  are the complements of the diagnoses w.r.t.  $\mathcal{T}_{\text{exa}}$ ; more precisely  $\mathcal{R}_1 = \{A \sqsubseteq \exists r.A, \exists r.Y \sqsubseteq B, Y \sqsubseteq B, B \sqsubseteq C\}$ ,  $\mathcal{R}_2 = \{A \sqsubseteq Y, \exists r.Y \sqsubseteq B, B \sqsubseteq C\}$ , and  $\mathcal{R}_3 = \{A \sqsubseteq \exists r.A, A \sqsubseteq Y, B \sqsubseteq C\}$ .

The central problem studied in this paper is referred to as *axiom pinpointing*, which is the task of enumerating MinAs for a given subsumption relation w.r.t. a TBox.

## 2.2 Propositional Satisfiability

We assume that the reader is familiar with the standard definitions and notation of propositional logic (Biere, Heule, van Maaren, & Walsh, 2009). In this paper we focus on Boolean formulae in conjunctive normal form (CNF). A CNF formula  $\mathcal{F}$  is defined over a set of Boolean variables  $V(\mathcal{F}) = \{x_1, \dots, x_n\}$  as a conjunction of clauses  $(c_1 \wedge \dots \wedge c_m)$ . A clause  $c$  is a disjunction of literals  $(\ell_1 \vee \dots \vee \ell_k)$  and a literal  $\ell$  is either a variable  $x$  or its negation  $\neg x$ . We refer to the set of literals appearing in  $\mathcal{F}$  as  $L(\mathcal{F})$ . Formulae can also be represented as sets of clauses, and clauses as sets of literals. With this representation,  $|\mathcal{F}|$  refers to the number of clauses of  $\mathcal{F}$  and  $||\mathcal{F}||$  is the number of literals in  $\mathcal{F}$ .

A truth assignment, or interpretation, is a mapping  $\mu : V(\mathcal{F}) \rightarrow \{0, 1\}$ . If all the variables in  $V(\mathcal{F})$  are assigned a truth value,  $\mu$  is referred to as a *complete* assignment.

Interpretations can also be seen as conjunctions or sets of literals. Truth valuations are lifted to clauses and formulae as follows:  $\mu$  satisfies a clause  $c$  if it contains at least one of its literals. Given a formula  $\mathcal{F}$ ,  $\mu$  satisfies  $\mathcal{F}$  (written  $\mu \models \mathcal{F}$ ) if it satisfies all its clauses, being  $\mu$  referred to as a *model* of  $\mathcal{F}$ .

Given two formulae  $\mathcal{F}$  and  $\mathcal{G}$ ,  $\mathcal{F}$  entails  $\mathcal{G}$  (written  $\mathcal{F} \models \mathcal{G}$ ) iff all the models of  $\mathcal{F}$  are also models of  $\mathcal{G}$ .  $\mathcal{F}$  and  $\mathcal{G}$  are equivalent (written  $\mathcal{F} \equiv \mathcal{G}$ ) iff  $\mathcal{F} \models \mathcal{G}$  and  $\mathcal{G} \models \mathcal{F}$ . A formula  $\mathcal{F}$  is satisfiable ( $\mathcal{F} \not\models \perp$ ) if there exists a model for it. Otherwise it is unsatisfiable ( $\mathcal{F} \models \perp$ ). SAT is the decision problem of determining the satisfiability of a propositional formula. This problem is in general NP-complete (Cook, 1971).

Some applications require computing certain types of models. In this paper, we will make use of maximal models, i.e. models such that a set-wise maximal subset of the variables are assigned value 1.

**Definition 2** (MxM). *Let  $\mathcal{F}$  be a satisfiable propositional formula,  $\mu \models \mathcal{F}$  a model of  $\mathcal{F}$  and  $P \subseteq V(\mathcal{F})$  the set of variables appearing in  $\mu$  with positive polarity.  $\mu$  is a maximal model (MxM) of  $\mathcal{F}$  iff  $\mathcal{F} \cup P \not\models \perp$  and for all  $v \in V(\mathcal{F}) \setminus P$ ,  $\mathcal{F} \cup P \cup \{v\} \models \perp$ .*

In the following we will often express a maximal model simply by providing the set  $P$  of the positive literals it contains.

**Example 3.** Consider the satisfiable formula  $\mathcal{F} = \{(x_1 \vee x_2 \vee x_3), (\neg x_1 \vee \neg x_2), (\neg x_1 \vee \neg x_3)\}$ . There are two maximal models of  $\mathcal{F}$ :  $P_1 = \{x_1\}$  and  $P_2 = \{x_2, x_3\}$ .

Horn formulae constitute an important subclass of propositional logic. These are composed of Horn clauses, which have at most one positive literal. A Horn clause is said to be a *definite* clause if it has one positive, and it is a *goal* clause otherwise. Satisfiability of Horn formulae is known to be decidable in linear time (Dowling & Gallier, 1984; Itai & Makowsky, 1987; Minoux, 1988).

Given an unsatisfiable formula  $\mathcal{F}$ , the following subsets represent different notions regarding (set-wise) minimal unsatisfiability and maximal satisfiability (Liffiton & Sakallah, 2008; Marques-Silva et al., 2013a):

**Definition 3** (MUS, MCS, MSS).  *$\mathcal{M} \subseteq \mathcal{F}$  is a Minimally Unsatisfiable Subset (MUS) of  $\mathcal{F}$  iff  $\mathcal{M}$  is unsatisfiable and for all  $\mathcal{M}' \subsetneq \mathcal{M}$ ,  $\mathcal{M}'$  is satisfiable.  $\mathcal{C} \subseteq \mathcal{F}$  is a Minimal Correction Subset (MCS) iff  $\mathcal{F} \setminus \mathcal{C}$  is satisfiable and for all  $\mathcal{C}' \subsetneq \mathcal{C}$ ,  $\mathcal{F} \setminus \mathcal{C}'$  is unsatisfiable.  $\mathcal{S} \subseteq \mathcal{F}$  is a Maximal Satisfiable Subset (MSS) iff  $\mathcal{S}$  is satisfiable and for all  $\mathcal{S}'$  s.t.  $\mathcal{S} \subsetneq \mathcal{S}' \subseteq \mathcal{F}$ ,  $\mathcal{S}'$  is unsatisfiable.*

In other words, an MUS is an irreducible unsatisfiable subset of  $\mathcal{F}$  and, as such, it provides an *explanation* for the unsatisfiability of  $\mathcal{F}$ . An MCS is an irreducible set of clauses which, if removed from  $\mathcal{F}$ , would render the formula satisfiable. An MSS is satisfiable subset of the clauses of  $\mathcal{F}$  such that if it was added any other clause from  $\mathcal{F}$  it would become unsatisfiable. Thus, MCSes and MSSes represent ways of *restoring* the satisfiability of  $\mathcal{F}$  so that only the strictly necessary clauses are dropped from it.

Given an MCS  $\mathcal{C} \subseteq \mathcal{F}$ , its complement (i.e.  $\mathcal{F} \setminus \mathcal{C}$ ) represents an MSS (and *vice versa*). MUSes and MCSes are closely related by the well-known hitting set duality (Reiter, 1987; Bailey & Stuckey, 2005; Birnbaum & Lozinskii, 2003; Slaney, 2014): Every MCS (MUS)

is an irreducible hitting set of all MUSes (MCSes) of  $\mathcal{F}$ . In the worst case, there can be an exponential number of MUSes and MCSes (Liffiton & Sakallah, 2008; O’Sullivan, Papadopoulos, Faltungs, & Pu, 2007).

**Example 4.** Consider the formula  $\mathcal{F} = \{(x_1), (\neg x_1 \vee \neg x_2), (x_2), (x_3), (\neg x_2 \vee \neg x_3)\}$ , which is unsatisfiable. Then,

- its MUSes are  $\mathcal{M}_1 = \{(x_1), (\neg x_1 \vee \neg x_2), (x_2)\}$ , and  $\mathcal{M}_2 = \{(x_2), (x_3), (\neg x_2 \vee \neg x_3)\}$ ;
- $\mathcal{F}$  has the five MCSes  $\mathcal{C}_1 = \{(x_2)\}$ ,  $\mathcal{C}_2 = \{(x_1), (x_3)\}$ ,  $\mathcal{C}_3 = \{(x_1), (\neg x_2 \vee \neg x_3)\}$ ,  $\mathcal{C}_4 = \{(\neg x_1 \vee \neg x_2), (x_3)\}$ , and  $\mathcal{C}_5 = \{(\neg x_1 \vee \neg x_2), (\neg x_2 \vee \neg x_3)\}$ ; and
- $\mathcal{F}$  has five MSSes  $\mathcal{S}_1 = \{(x_1), (\neg x_1 \vee \neg x_2), (x_3), (\neg x_2 \vee \neg x_3)\}$ ,  $\mathcal{S}_2 = \{(\neg x_1 \vee \neg x_2), (x_2), (\neg x_2 \vee \neg x_3)\}$ ,  $\mathcal{S}_3 = \{(\neg x_1 \vee \neg x_2), (x_2), (x_3)\}$ ,  $\mathcal{S}_4 = \{(x_1), (x_2), (\neg x_2 \vee \neg x_3)\}$ , and  $\mathcal{S}_5 = \{(x_1), (x_2), (x_3)\}$ .

MCSes are closely related to the MaxSAT problem, which consists in finding an assignment satisfying as many clauses as possible. The smallest MCS (largest MSS) represents an optimal solution to MaxSAT. This setting is often generalized by allowing formulae to be partitioned into sets of *hard* and *soft* clauses, i.e.,  $\mathcal{F} = \{\mathcal{F}_H, \mathcal{F}_S\}$ . Hard clauses must be satisfied, while soft clauses can be relaxed if necessary. In this case, MCSes are required to be subsets of  $\mathcal{F}_S$ . Motivated by several applications, MUSes and related concepts have been extended to CNF formulae where clauses are partitioned into several disjoint sets called groups (Liffiton & Sakallah, 2008).

**Definition 4** (Group-Oriented MUS). *Given an explicitly partitioned unsatisfiable CNF formula  $\mathcal{F} = \mathcal{G}_0 \cup \dots \cup \mathcal{G}_k$ , a group-oriented MUS (or group-MUS) of  $\mathcal{F}$  is a set of groups  $\mathcal{G}_M \subseteq \{\mathcal{G}_1, \dots, \mathcal{G}_k\}$ , such that  $\{\mathcal{G}_0\} \cup \mathcal{G}_M$  is unsatisfiable, and for all  $\mathcal{G}'_M \subsetneq \mathcal{G}_M$ ,  $\{\mathcal{G}_0\} \cup \mathcal{G}'_M$  is satisfiable.*

Note the special role  $\mathcal{G}_0$  (*group-0*); this group consists of *background* clauses that are included in every group-MUS. Because of  $\mathcal{G}_0$  a group-MUS, as opposed to a MUS, can be empty. Nevertheless, in this paper we assume that  $\mathcal{G}_0$  is satisfiable. Equivalently, the related concepts of group-MCS and group-MSS can be defined in the same way.

**Definition 5** (Group-Oriented MCS, MSS). *Given an explicitly partitioned unsatisfiable CNF formula  $\mathcal{F} = \mathcal{G}_0 \cup \dots \cup \mathcal{G}_k$ , a group-oriented MCS (or group-MCS) of  $\mathcal{F}$  is a set of groups  $\mathcal{G}_C \subseteq \{\mathcal{G}_1, \dots, \mathcal{G}_k\}$ , such that  $\mathcal{F} \setminus \mathcal{G}_C$  is satisfiable, and for all  $\mathcal{G}'_C \subsetneq \mathcal{G}_C$ ,  $\{\mathcal{F} \setminus \mathcal{G}'_C\}$  is unsatisfiable. A group-oriented MSS (or group-MSS) of  $\mathcal{F}$  is a set of groups  $\mathcal{G}_S \subseteq \{\mathcal{G}_1, \dots, \mathcal{G}_k\}$ , such that  $\{\mathcal{G}_0\} \cup \mathcal{G}_S$  is satisfiable, and for every  $\mathcal{G}'_S$  such that  $\mathcal{G}_S \subsetneq \mathcal{G}'_S \subseteq \{\mathcal{G}_1, \dots, \mathcal{G}_k\}$ ,  $\{\mathcal{G}_0\} \cup \mathcal{G}'_S$  is unsatisfiable.*

**Example 5.** Consider the unsatisfiable group formula  $\mathcal{F} = \mathcal{G}_0 \cup \mathcal{G}_1 \cup \mathcal{G}_2 \cup \mathcal{G}_3$ , with the groups  $\mathcal{G}_0 = \{(\neg x_1 \vee \neg x_2), (\neg x_2 \vee \neg x_3)\}$ ,  $\mathcal{G}_1 = \{(x_1)\}$ ,  $\mathcal{G}_2 = \{(x_2)\}$ , and  $\mathcal{G}_3 = \{(x_3)\}$ .

- The group-MUSes of  $\mathcal{F}$  are  $\mathcal{G}_{M1} = \{\mathcal{G}_1, \mathcal{G}_2\}$ , and  $\mathcal{G}_{M2} = \{\mathcal{G}_2, \mathcal{G}_3\}$ .
- The group-MCSes of  $\mathcal{F}$  are  $\mathcal{G}_{C1} = \{\mathcal{G}_2\}$ , and  $\mathcal{G}_{C2} = \{\mathcal{G}_1, \mathcal{G}_3\}$ .
- The group-MSSes of  $\mathcal{F}$  are  $\mathcal{G}_{S1} = \{\mathcal{G}_1, \mathcal{G}_2\}$ , and  $\mathcal{G}_{S2} = \{\mathcal{G}_2\}$ .

In the case of MaxSAT, the use of groups is investigated in detail in (Heras, Morgado, & Marques-Silva, 2015).

Table 1:  $\mathcal{EL}^+$  Completion Rules

Rules	Preconditions	Inferred axiom
$R_1$	$A \sqsubseteq A_i, 1 \leq i \leq n$	$A_1 \sqcap \dots \sqcap A_n \sqsubseteq B$
$R_2$	$A \sqsubseteq A_1$	$A_1 \sqsubseteq \exists r.B$
$R_3$	$A \sqsubseteq \exists r.B, B \sqsubseteq B_1$	$\exists r.B_1 \sqsubseteq B_2$
$R_4$	$A_{i-1} \sqsubseteq \exists r_i.A_i, 1 \leq i \leq n$	$r_1 \circ \dots \circ r_n \sqsubseteq r$
		$A_0 \sqsubseteq \exists r.A_n$

### 3. SAT-Based Axiom Pinpointing

To our best knowledge,  $\text{EL}^+\text{SAT}$  (Sebastiani & Vescovi, 2009; Vescovi, 2011; Sebastiani & Vescovi, 2015) was the first SAT-based approach to axiom pinpointing in  $\mathcal{EL}^+$  ontologies proposed in the literature. This section overviews the main ideas  $\text{EL}^+\text{SAT}$  is based on. Some of these ideas will be used in the remaining sections of the paper.

$\text{EL}^+\text{SAT}$  consists of two phases. First, a given  $\mathcal{EL}^+$  Tbox  $\mathcal{T}$  is classified, i.e. all the subsumption relations between concept names in  $\mathcal{T}$  are inferred, and this reasoning process is encoded into a (satisfiable) Horn formula. Each of the variables of this Horn formula corresponds to either an original axiom in  $\mathcal{T}$  or a consequence derived during the classification of  $\mathcal{T}$ . Then, in a second step, given a subsumption relation to be explained, the Horn formula is extended and analyzed through SAT and SMT-based techniques in order to compute the MinAs of the subsumption relation w.r.t.  $\mathcal{T}$ .

#### 3.1 Generation of Horn Formulae

Recall *classification* refers to the task of inferring all the subsumption relations that follow from a TBox  $\mathcal{T}$ . In the case of the  $\mathcal{EL}$  family of DLs, classification can be done in polynomial time by first normalizing  $\mathcal{T}$ , and then exhaustively applying the *completion rules* shown in Table 1 (Baader et al., 2005, 2007). The main idea behind  $\text{EL}^+\text{SAT}$  is to encode the classification process of  $\mathcal{T}$  into a Horn formula. Then, given a subsumption relation inferred in the classification of  $\mathcal{T}$ , this Horn formula will serve to represent the corresponding axiom pinpointing problem in the SAT domain.

##### 3.1.1 CLASSIFICATION AND HORN ENCODING

The encoding is obtained in the following way:

1. As an initial step, each axiom  $a_i \in \mathcal{T}$  is assigned a unique selector variable  $s_{[a_i]}$ . For trivial GCIs of the form  $C \sqsubseteq C$  or  $C \sqsubseteq \top$ ,  $s_{[a_i]}$  is the constant `true`. The Horn formula  $\phi_{\mathcal{T}(\text{po})}^{\text{all}}$  is initialized to the empty formula.
2. Then,  $\mathcal{T}$  is normalized so that its axioms are in any of the forms

$$\begin{aligned} & (A_1 \sqcap \dots \sqcap A_k) \sqsubseteq B \quad (k \geq 1), \\ & A \sqsubseteq \exists r.B, \\ & \exists r.A \sqsubseteq B, \text{ or} \\ & r_1 \circ \dots \circ r_n \sqsubseteq s \quad (n \geq 1), \end{aligned}$$

where  $A, A_i, B \in \mathbf{N}_C$  and  $r, r_i, s \in \mathbf{N}_R$ .

This process can be performed in linear time and it results in a TBox  $\mathcal{T}_N$  where each axiom  $a_i \in \mathcal{T}$  is substituted by a set of axioms in normal form  $\{a_{i1}, \dots, a_{im_i}\}$ .

At this point, the Horn clauses  $s_{[a_i]} \rightarrow s_{[a_{ik}]}$ , with  $1 \leq k \leq m_i$ , are added to  $\phi_{\mathcal{T}(\text{po})}^{\text{all}}$ .

3. After normalizing the ontology,  $\mathcal{T}_N$  is saturated through the exhaustive application of the *completion rules* shown in [Table 1](#), resulting in the extended TBox class( $\mathcal{T}$ ). Each of the rows in [Table 1](#) constitutes a completion rule. Their application is sound and complete for inferring subsumption relations (Baader et al., 2005). Whenever a rule  $r$  can be applied (with antecedents  $\text{ant}(r)$ ) leading to inferring an axiom  $a_i$ , the Horn clause

$$\left( \bigwedge_{a_j \in \text{ant}(r)} s_{[a_j]} \right) \rightarrow s_{[a_i]}$$

is added to  $\phi_{\mathcal{T}(\text{po})}^{\text{all}}$ .

As a result,  $\phi_{\mathcal{T}(\text{po})}^{\text{all}}$  will eventually encode all possible derivations of completion rules inferring any axiom such that  $X \sqsubseteq_{\mathcal{T}} Y$ , with  $X, Y \in \mathbf{N}_{\mathcal{C}}$ .

**Example 6.** Consider again the ontology  $\mathcal{T}_{\text{exa}}$  from [Example 1](#), which is already in normal form. On the left, the example shows the consequences derived by the classification of  $\mathcal{T}_{\text{exa}}$ . On the right, the example shows the Horn encoding of the classification of  $\mathcal{T}_{\text{exa}}$ , indicating the completion rule that was applied in each case.

$$\begin{aligned} \text{class}(\mathcal{T}_{\text{exa}}) = \{ & a_1 : A \sqsubseteq \exists r.A \\ & a_2 : A \sqsubseteq Y, \quad \phi_{\mathcal{T}_{\text{exa}}(\text{po})}^{\text{all}} = \{s_{[a_1]} \wedge s_{[a_2]} \wedge s_{[a_3]} \rightarrow s_{[a_6]}, \quad (R_3) \\ & a_3 : \exists r.Y \sqsubseteq B, \quad s_{[a_2]} \wedge s_{[a_4]} \rightarrow s_{[a_6]}, \quad (R_1) \\ & a_4 : Y \sqsubseteq B, \quad s_{[a_4]} \wedge s_{[a_5]} \rightarrow s_{[a_7]}, \quad (R_1) \\ & a_5 : B \sqsubseteq C, \quad s_{[a_5]} \wedge s_{[a_6]} \rightarrow s_{[a_8]}, \quad (R_1) \\ & a_6 : A \sqsubseteq B, \quad s_{[a_2]} \wedge s_{[a_7]} \rightarrow s_{[a_8]}\} \quad (R_1) \\ & a_7 : Y \sqsubseteq C, \\ & a_8 : A \sqsubseteq C \} \end{aligned}$$

The Horn encoding of the classification of  $\mathcal{T}$  only needs to be done once. Using this formula, one can then find all the MinAs for any subsumption relation that can be derived from  $\mathcal{T}$ . Thus, the cost of constructing  $\phi_{\mathcal{T}(\text{po})}^{\text{all}}$  can be amortized if several consequences of the TBox are explained and corrected.

### 3.1.2 AXIOM PINPOINTING INSTANCES

For computing MinAs of a given subsumption relation  $C \sqsubseteq_{\mathcal{T}} D$ , EL<sup>+</sup>SAT builds on the following fundamental result.

**Theorem 1** (Theorem 3 in (Sebastiani & Vescovi, 2015)). *Given an  $\mathcal{EL}^+$  TBox  $\mathcal{T}$ , for every  $\mathcal{S} \subseteq \mathcal{T}$  and for every pair of concept names  $C, D \in \mathbf{N}_{\mathcal{C}}$ ,  $C \sqsubseteq_{\mathcal{S}} D$  if and only if the Horn propositional formula  $\phi_{\mathcal{T}(\text{po})}^{\text{all}} \wedge (\neg s_{[C \sqsubseteq D]}) \wedge \bigwedge_{ax_i \in \mathcal{S}} (s_{[ax_i]})$  is unsatisfiable.*

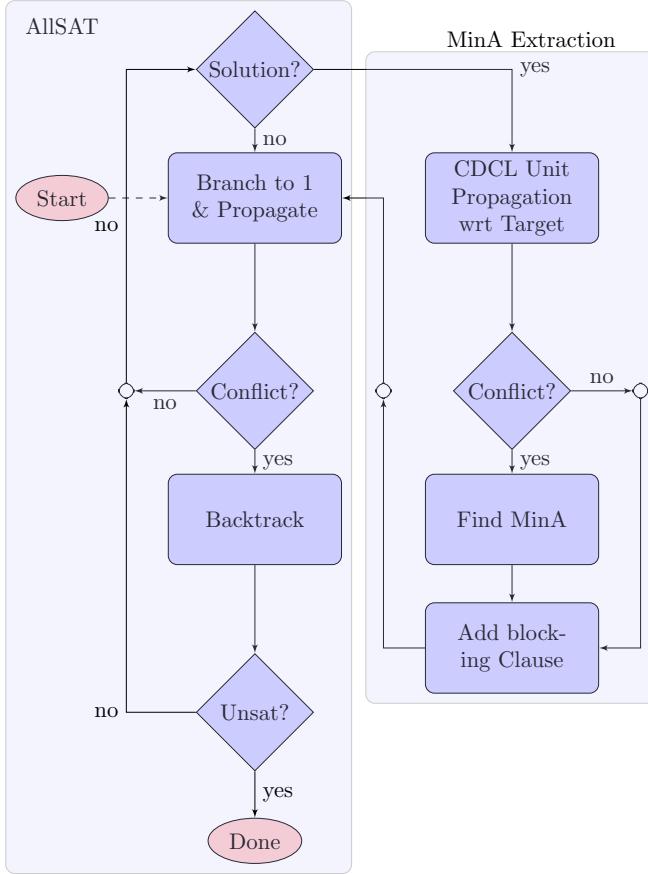
This theorem states that, given the Horn encoding of the classification of  $\mathcal{T}$   $\phi_{\mathcal{T}(\text{po})}^{\text{all}}$ , the set of Boolean variables  $\{s_{[ax_i]} \mid ax_i \in \mathcal{T}\}$  representing the original axioms of  $\mathcal{T}$ , and the Boolean variable  $s_{[C \sqsubseteq D]}$  associated with the subsumption relation to be explained, the MinAs of  $\mathcal{T}$  w.r.t.  $C \sqsubseteq D$  correspond to subsets of  $\{s_{[ax_i]} \mid ax_i \in \mathcal{T}\}$  whose activation (setting these variables to 1) makes  $\phi_{\mathcal{T}(\text{po})}^{\text{all}} \wedge (\neg s_{[C \sqsubseteq D]})$  unsatisfiable. More concretely, from the enumeration mechanism of EL<sup>+</sup>SAT, MinAs will correspond exactly to minimal subsets of  $\{s_{[ax_i]} \mid ax_i \in \mathcal{T}\}$ . Noticeably, (Sebastiani & Vescovi, 2015) introduces effective techniques for simplifying the formulas that encode concrete MinA computation instances, such as the *Cone-Of-Influence* (C.O.I.) modularization technique or the so-called x2 optimization. These techniques allow for reducing to a great extent the size of the Horn formula and the number of assumption variables in the original ontology that take place in the enumeration step, and they play a very important role in the efficiency of EL<sup>+</sup>SAT. These techniques can also be implemented in polynomial time. For further information we refer the reader to (Vescovi, 2011; Sebastiani & Vescovi, 2015).

### 3.2 EL<sup>+</sup>SAT Computation of MinAs

As mentioned earlier, the second component of EL<sup>+</sup>SAT computes the MinAs of the subsumption relation by analyzing the causes of unsatisfiability of an extended formula. The focus on this section is to analyze the MinA enumerator used in EL<sup>+</sup>SAT, highlighting a number of issues in its design. [Figure 1](#) provides a high-level perspective of the MinA enumerator EL<sup>+</sup>SAT, which is based on an AllSAT/AllSMT approach (Lahiri, Nieuwenhuis, & Oliveras, 2006).

In this approach, branching is executed in such a manner that variables are always assigned value 1. This guarantees that maximal models are computed. The downside is that the branching heuristic of the SAT solver is affected. The AllSAT approach searches for an assignment that satisfies the Horn formula without taking into account the unit negative clause associated with the target subsumption relation. Every time a solution to the Horn formula is found, EL<sup>+</sup>SAT enters a dedicated mode, where CDCL-based unit propagation is used to identify a conflict and, if a conflict is found, a MinA is computed and reported. The computation of a MinA uses a standard deletion-based algorithm (Belov et al., 2012). Blocking is based on the branching variables, and so it is solely composed of negative literals, i.e. it is also a Horn formula. This is true independently of having computed a MinA or not. The process is then repeated while solutions to the Horn formula exist.

The description above summarizes the key steps of the MinA enumerator used in EL<sup>+</sup>SAT, but also highlights some of the performance issues that can exist with EL<sup>+</sup>SAT. First, forcing branching to always assign variables to 1 affects the branching heuristic of the SAT solver. This can have an important impact on the performance of the tool. Second, unit propagation in a CDCL-based SAT solver is quadratic in the worst-case (Gent, 2013), while Horn formula satisfiability is decidable in linear time. The theoretical solution is to use different implementation of how the watched pointers are handled, but this is seldom used in practice. Third, as shown in [Section 4](#) for the concrete setting of Horn formulae, there are more efficient solutions than using deletion-based algorithms for computing MinAs. Finally, and most importantly, the fact that non-MinAs are blocked in the same way

Figure 1: High-level perspective of  $\text{EL}^+\text{SAT}$  MinA enumeration

as MinAs, can result in an exponentially larger number of iterations when compared with distinguished blocking for MinAs and non-MinAs (Arif et al., 2015b).

#### 4. Towards Efficient SAT-Based Axiom Pinpointing

This section builds on the ideas introduced by  $\text{EL}^+\text{SAT}$  and develops them further. First, it shows that axiom pinpointing in  $\mathcal{EL}^+$  ontologies can be reduced to the problem of enumerating (group) MUSes of a Horn formula. Then, it describes two approaches for this problem, which are based on explicit and implicit hitting set dualization between the (group) MUSes and (group) MCSes of unsatisfiable formulae, giving rise to the state-of-the-art algorithms EL2MCS (Arif et al., 2015a) and HGMUS (Arif et al., 2015b) respectively.

##### 4.1 Axiom Pinpointing as Group-MUS Enumeration

Although not explicitly stated, the relationship between axiom pinpointing and MUS extraction has been apparent in earlier work (Baader et al., 2007; Sebastiani & Vescovi, 2009, 2015). Building on [Theorem 1](#), the computation of the MinAs for a subsumption relation

$C \sqsubseteq_{\mathcal{T}} D$  can be reduced to the problem of enumerating group-MUSes of the group Horn formula  $\mathcal{H}_{[C \sqsubseteq_{\mathcal{T}} D]} = \{\mathcal{G}_0, \mathcal{G}_1, \dots, \mathcal{G}_{|\mathcal{T}|}\}$ , where

- $\mathcal{G}_0 = \phi_{\mathcal{T}(\text{po})}^{\text{all}} \cup \{(\neg s_{[C \sqsubseteq D]})\}$ ; and
- each axiom  $ax_i \in \mathcal{T}, 1 \leq i \leq |\mathcal{T}|$  defines a group  $\mathcal{G}_i = \{(s_{[ax_i]})\}$  with a single unit clause.

Then, we obtain the following result.

**Theorem 2.** *Given an  $\mathcal{EL}^+$  TBox  $\mathcal{T}$ , a set  $\mathcal{M} \subseteq \mathcal{T}$ , and  $C, D \in \mathbf{N}_{\mathcal{C}}$ ,  $\mathcal{M}$  is a MinA for  $C \sqsubseteq_{\mathcal{T}} D$  if and only if the set  $\mathcal{G}_{\mathcal{M}} = \bigcup_{\{i | ax_i \in \mathcal{M}\}} \mathcal{G}_i$  is a group-MUS of  $\mathcal{H}_{[C \sqsubseteq_{\mathcal{T}} D]}$ .*

*Proof. (If)* By definition, since  $\mathcal{G}_{\mathcal{M}}$  is a group-MUS of  $\mathcal{H}_{[C \sqsubseteq_{\mathcal{T}} D]}$ , it holds that the formula  $\phi_{\mathcal{T}(\text{po})}^{\text{all}} \wedge \bigwedge_{ax_i \in \mathcal{M}} (s_{[ax_i]} \wedge (\neg s_{[C \sqsubseteq D]})$  is unsatisfiable, and for all  $\mathcal{M}' \subsetneq \mathcal{M}$  the formula  $\phi_{\mathcal{T}(\text{po})}^{\text{all}} \wedge \bigwedge_{ax_i \in \mathcal{M}'} (s_{[ax_i]} \wedge (\neg s_{[C \sqsubseteq D]})$  is satisfiable. By [Theorem 1](#), it follows that  $C \sqsubseteq_{\mathcal{M}} D$  and for all  $\mathcal{M}' \subsetneq \mathcal{M}$ ,  $C \not\sqsubseteq_{\mathcal{M}'} D$ . Thus,  $\mathcal{M}$  is a MinA for  $C \sqsubseteq_{\mathcal{T}} D$ .

*(Only if)* Since  $\mathcal{M}$  is a MinA for  $C \sqsubseteq_{\mathcal{T}} D$ , we have that  $C \sqsubseteq_{\mathcal{M}} D$  and for all  $\mathcal{M}' \subsetneq \mathcal{M}$ ,  $C \not\sqsubseteq_{\mathcal{M}'} D$ . By [Theorem 1](#), the formula  $\phi_{\mathcal{T}(\text{po})}^{\text{all}} \wedge \bigwedge_{ax_i \in \mathcal{M}} (s_{[ax_i]} \wedge (\neg s_{[C \sqsubseteq D]})$  is unsatisfiable (and hence also is  $\{\mathcal{G}_0\} \cup \mathcal{G}_{\mathcal{M}}$  in  $\mathcal{H}_{[C \sqsubseteq_{\mathcal{T}} D]}$ ) and for all  $\mathcal{M}' \subsetneq \mathcal{M}$ , the formula  $\phi_{\mathcal{T}(\text{po})}^{\text{all}} \wedge \bigwedge_{ax_i \in \mathcal{M}'} (s_{[ax_i]} \wedge (\neg s_{[C \sqsubseteq D]})$  is satisfiable (and so is  $\mathcal{G}_0 \cup \mathcal{G}_{\mathcal{M}'}$  in  $\mathcal{H}_{[C \sqsubseteq_{\mathcal{T}} D]}$ ). Thus,  $\mathcal{G}_{\mathcal{M}}$  is a group-MUS of  $\mathcal{H}_{[C \sqsubseteq_{\mathcal{T}} D]}$ .  $\square$

The minimal hitting set duality relationship between MinAs and diagnoses for a given subsumption relation w.r.t. a TBox, which as seen holds also between group-MUSes and group-MCSes of unsatisfiable group formulae, allows us to map diagnoses to group-MCSes. At the same time, since each repair is the complement of a diagnosis, and each group-MSS is the complement of a group-MCS, repairs can be related with group-MSSes.

**Corollary 1.** *Let  $\mathcal{T}$  be an  $\mathcal{EL}^+$  TBox, and  $C, D$  be two concept names. Then:*

- $\mathcal{D} \subseteq \mathcal{T}$  is a diagnosis for  $C \sqsubseteq_{\mathcal{T}} D$  if and only if  $\mathcal{G}_{\mathcal{D}} = \bigcup_{\{i | ax_i \in \mathcal{D}\}} \mathcal{G}_i$  is a group-MCS of  $\mathcal{H}_{[C \sqsubseteq_{\mathcal{T}} D]}$ , and
- $\mathcal{R} \subseteq \mathcal{T}$  is a repair for  $C \sqsubseteq_{\mathcal{T}} D$  if and only if  $\mathcal{G}_{\mathcal{R}} = \bigcup_{\{i | ax_i \in \mathcal{R}\}} \mathcal{G}_i$  is a group-MSS of  $\mathcal{H}_{[C \sqsubseteq_{\mathcal{T}} D]}$ .

The following example illustrates these results.

**Example 7.** *Let us consider the  $\mathcal{EL}^+$  ontology  $\mathcal{T}_{\text{exa}}$  from Example 1. The classification of  $\mathcal{T}_{\text{exa}}$  and the definition of  $\phi_{\mathcal{T}_{\text{exa}}(\text{po})}^{\text{all}}$  are shown in Example 6. As we can observe, the axiom  $a_6 : A \sqsubseteq B$  belongs to the classification of  $\mathcal{T}_{\text{exa}}$ . For this subsumption relation, we define  $\mathcal{H}_{[A \sqsubseteq_{\mathcal{T}_{\text{exa}}} B]} = \{\mathcal{G}_0, \mathcal{G}_1, \dots, \mathcal{G}_5\}$ , where  $\mathcal{G}_0 = \phi_{\mathcal{T}_{\text{exa}}(\text{po})}^{\text{all}} \cup \{(\neg s_{[a_6]})\}$ , and for  $i = 1, \dots, 5$ ,  $\mathcal{G}_i = \{(s_{[a_i]})\}$ , i.e.*

$$\begin{aligned}
\mathcal{G}_0 = & \{(\neg s_{[a_1]} \vee \neg s_{[a_2]} \vee \neg s_{[a_3]} \vee s_{[a_6]}), \\
& (\neg s_{[a_2]} \vee \neg s_{[a_4]} \vee s_{[a_6]}), \\
& (\neg s_{[a_4]} \vee \neg s_{[a_5]} \vee s_{[a_7]}), \\
& (\neg s_{[a_5]} \vee \neg s_{[a_6]} \vee s_{[a_8]}), \\
& (\neg s_{[a_2]} \vee \neg s_{[a_7]} \vee s_{[a_8]}), \\
& (\neg s_{[a_6]}))\} \\
\mathcal{G}_1 = & \{(s_{[a_1]})\} \\
\mathcal{G}_2 = & \{(s_{[a_2]})\} \\
\mathcal{G}_3 = & \{(s_{[a_3]})\} \\
\mathcal{G}_4 = & \{(s_{[a_4]})\} \\
\mathcal{G}_5 = & \{(s_{[a_5]})\}
\end{aligned}$$

$\mathcal{H}_{[A \sqsubseteq B]}$  has two group-MUSes:  $\mathcal{G}_{\mathcal{M}_1} = \{\mathcal{G}_2, \mathcal{G}_4\}$  and  $\mathcal{G}_{\mathcal{M}_2} = \{\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_3\}$  corresponding to the MinAs  $\mathcal{M}_1 = \{A \sqsubseteq Y, Y \sqsubseteq B\}$ , and  $\mathcal{M}_2 = \{A \sqsubseteq \exists r.A, A \sqsubseteq Y, \exists r.Y \sqsubseteq B\}$  respectively. The group-MCSes of  $\mathcal{H}_{[A \sqsubseteq B]}$  are  $\mathcal{G}_{\mathcal{C}_1} = \{\mathcal{G}_2\}$ ,  $\mathcal{G}_{\mathcal{C}_2} = \{\mathcal{G}_1, \mathcal{G}_4\}$ , and  $\mathcal{G}_{\mathcal{C}_3} = \{\mathcal{G}_3, \mathcal{G}_4\}$ , corresponding to the diagnoses  $\mathcal{D}_1 = \{A \sqsubseteq Y\}$ ,  $\mathcal{D}_2 = \{A \sqsubseteq \exists r.A, Y \sqsubseteq B\}$ , and  $\mathcal{D}_3 = \{\exists r.Y \sqsubseteq B, Y \sqsubseteq B\}$  respectively.

It should be observed that the difference between the enumeration of plain MUSes of Horn formulae and the enumeration of group-MUSes is significant in terms of complexity. First, enumeration of group-MUSes of Horn formulae cannot be achieved in total polynomial time, unless P = NP. This is an immediate consequence from the fact that axiom pinpointing for the  $\mathcal{EL}$  family of DLs cannot be achieved in total polynomial time, unless P = NP (Baader et al., 2007), and that axiom pinpointing for the  $\mathcal{EL}$  family of DLs can be reduced in polynomial time to group-MUS enumeration of Horn formulae. Second, the MUSes of Horn formulae can be enumerated with only a polynomial delay between answers, which in particular implies a total polynomial time (Peñaloza & Sertkaya, 2010).

The following two subsections develop two approaches for enumerating the group-MUSes of Horn formulae. The first one, EL2MCS (Arif et al., 2015a) is based on explicit hitting set dualization between the MUSes and the MCSes of the Horn formula. The second approach, HGMUS (Arif et al., 2015b) follows an implicit hitting set dualization instead.

## 4.2 Group-MUS Enumeration by Explicit Hitting Set Dualization

Enumeration of MUSes has been the subject of research that can be traced to the seminal work of Reiter (Reiter, 1987). A well-known family of algorithms uses (explicit) minimal hitting set dualization (Birnbaum & Lozinskii, 2003; Bailey & Stuckey, 2005; Liffiton & Sakallah, 2008). The organization of these algorithms can be summarized as follows: First compute all the MCSes of a CNF formula. Second, MUSes are obtained by computing the minimal hitting sets of the set of MCSes.

EL2MCS (Arif et al., 2015a) is an approach based on this explicit hitting set dualization. This method relies on maximum satisfiability (MaxSAT): In the first step, it enumerates the group-MCSes of the Horn formula following methods developed for MaxSAT. Then, in the second step, it computes the minimal hitting sets (i.e. the group-MUSes) using an existing algorithm (Liffiton & Sakallah, 2008).

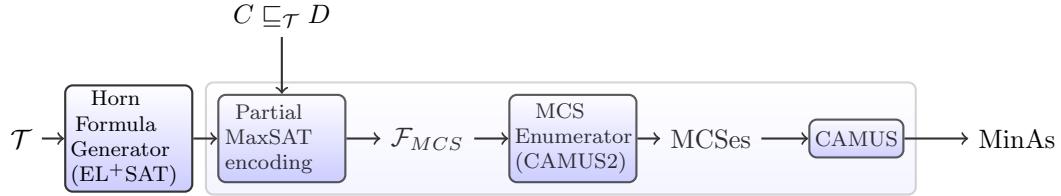


Figure 2: The EL2MCS tool

#### 4.2.1 AXIOM PINPOINTING USING MAXSAT

The approach followed by EL2MCS is to model the problem as partial maximum satisfiability (MaxSAT), and enumerate over the MUSes of the MaxSAT problem formulation. As mentioned before, the first phase of EL2MCS consists in enumerating the MCSes of a MaxSAT problem formulation, as follows. A formula  $\mathcal{F}_{MCS}$  is defined as follows: all the clauses in  $\mathcal{G}_0$  of the Horn formula  $\mathcal{H}_{[C \sqsubseteq_T D]}$  defined above are declared as *hard clauses*, i.e. both  $\phi_{T(po)}^{all}$  and  $(\neg s_{[C \sqsubseteq D]})$  are encoded as hard clauses. In addition, the variable  $s_{[ax_i]}$  associated with each axiom  $ax_i \in \mathcal{T}$  (corresponding to each group  $\mathcal{G}_i$ , with  $i > 0$ , in  $\mathcal{H}_{[C \sqsubseteq_T D]}$ ) denotes a *unit soft clause* ( $s_{[ax_i]}$ ). The intuitive justification for this construction is that the goal is to include as many axioms as possible, leaving out a minimal set which, if included, would cause the complete formula to be unsatisfiable. Thus, each of these sets represents an MCS of the MaxSAT problem formulation (and so a group-MCS of  $\mathcal{H}_{[C \sqsubseteq_T D]}$ ), but also a minimal set of axioms that needs to be dropped for the subsumption relation not to hold (i.e. a *diagnosis*). MCS enumeration can be implemented with a MaxSAT solver (Liffiton & Sakallah, 2008; Morgado, Liffiton, & Marques-Silva, 2012) or with a dedicated algorithm (Marques-Silva et al., 2013a). Extensive experimental results have shown that the first option yields better performance, and hence it is the adopted approach.

In a second phase, in order to get the MUSes of the MaxSAT problem formulation (and so the group-MUSes of  $\mathcal{H}_{[C \sqsubseteq_T D]}$ ), EL2MCS enumerates the minimal hitting sets of the set of all the MCSes computed in the first phase. This is achieved by exploiting the hitting set dualization algorithm used in CAMUS (Liffiton & Sakallah, 2008). (The use of SAT-based approaches turns out not to be efficient when compared to CAMUS and related algorithms (Liffiton & Sakallah, 2008).)

**Example 8.** Let us consider the group Horn formula  $\mathcal{H}_{[A \sqsubseteq_{T_{exa}} B]}$  in Example 7.

In the first phase, the MaxSAT formula  $\mathcal{F}_{MCS}$  is built, with the hard clauses  $\phi_{T_{exa}(po)}^{all} \cup \{(\neg s_{a_6})\}$ , and the soft clauses  $\{(s_{[a_1]}, (s_{[a_2]}, (s_{[a_3]}, (s_{[a_4]}, (s_{[a_5]})\}$ . The MCSes of  $\mathcal{F}_{MCS}$  are  $\mathcal{C}_1 = \{(s_{[a_2]})\}$ ,  $\mathcal{C}_2 = \{(s_{[a_1]}, (s_{[a_4]})\}$ , and  $\mathcal{C}_3 = \{(s_{[a_3]}, (s_{[a_4]})\}$ . As shown in Example 7, these sets correspond to the group-MCSes of  $\mathcal{H}_{[A \sqsubseteq_{T_{exa}} B]}$  (diagnoses for  $A \sqsubseteq_{T_{exa}} B$ ).

In the second phase, CAMUS is used to compute the minimal hitting sets of the MCSes, obtaining  $\mathcal{M}_1 = \{(s_{[a_2]}, (s_{[a_4]})\}$  and  $\mathcal{M}_2 = \{(s_{[a_1]}, (s_{[a_2]}, (s_{[a_3]})\}$ . As shown in Example 7, these sets correspond to the group-MUSes of  $\mathcal{H}_{[A \sqsubseteq_{T_{exa}} B]}$  (MinAs for  $A \sqsubseteq_{T_{exa}} B$ ).

#### 4.2.2 EL2MCS TOOL

The organization of the EL2MCS tool is depicted in Figure 2. The first step is similar to EL<sup>+</sup>SAT (Sebastiani & Vescovi, 2009, 2015) in that the propositional Horn formula  $\phi_{T(po)}^{all}$  is generated. The next step, however, produces a partial MaxSAT encoding. From this

---

**Algorithm 1:** EMUS (Previti & Marques-Silva, 2013) / MARCO (Liffiton et al., 2016)

---

**Input:**  $\mathcal{F}$  a CNF formula

```

1  $I \leftarrow \{p_i \mid c_i \in \mathcal{F}\}$                                 // Variable  $p_i$  picks clause  $c_i$ 
2  $\mathcal{Q} \leftarrow \emptyset$ 
3 while true do
4    $(st, P) \leftarrow \text{MaximalModel}(\mathcal{Q})$ 
5   if not  $st$  then return
6    $\mathcal{F}' \leftarrow \{c_i \mid p_i \in P\}$                                 // Pick selected clauses
7   if not  $\text{SAT}(\mathcal{F}')$  then
8      $\mathcal{M} \leftarrow \text{ComputeMUS}(\mathcal{F}')$ 
9      $\text{ReportMUS}(\mathcal{M})$ 
10     $b \leftarrow \{\neg p_i \mid c_i \in \mathcal{M}\}$                                 // Negative clause blocking the MUS
11  else
12     $b \leftarrow \{p_i \mid p_i \in I \setminus P\}$                                 // Positive clause blocking the MCS
13   $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{b\}$ 

```

---

encoding it becomes possible to enumerate the MCSes of the partial MaxSAT formula using an off-the-shelf tool like CAMUS2 (Marques-Silva et al., 2013a), which is a modern implementation of the of MCS enumerator available in CAMUS (Liffiton & Sakallah, 2008) capable of solving MaxSAT problems. The final step is to exploit minimal hitting set duality for computing all the MUSes given the set of MCSes (Liffiton & Sakallah, 2008). This is achieved with the help of the original CAMUS tool. The hypergraph traversal computation tools, shd (Murakami & Uno, 2011) and MTminer (Hébert, Bretto, & Crémilleux, 2007), could be used instead for this latter phase. It should be observed that, although the MCS enumeration uses CAMUS2, other alternative MCS enumeration approaches were considered during development (Marques-Silva et al., 2013a). The choice of CAMUS2 arose from the better performance that it demonstrated.

#### 4.3 Group-MUS Enumeration by Implicit Hitting Set Dualization

The main drawback of using explicit minimal hitting set dualization for finding group-MUSes is that it must first compute all group-MCSes, before the search for the first group-MUS starts. Thus, if the number of MCSes is large, this approach will take a long time find any MUS, and even longer to find them all, even if the total number of MUSes is small. As a result, recent work considered what can be described as implicit minimal hitting set dualization (Liffiton & Malik, 2013; Previti & Marques-Silva, 2013; Liffiton et al., 2016). In these approaches (implemented in systems like EMUS (Previti & Marques-Silva, 2013) and MARCO (Liffiton et al., 2016)), either an MUS or an MCS is computed at each step of the algorithm, with the guarantee that at least one MUS will be obtained at the outset. In some settings, implicit minimal hitting set dualization is the only solution for finding some MUSes of a CNF formula. As pointed out in this recent work, implicit minimal hitting set dualization aims to complement, but not replace, the explicit dualization alternative, and in some settings where enumeration of MCSes is feasible, the latter may be the preferred option (Previti & Marques-Silva, 2013; Liffiton et al., 2016).

[Algorithm 1](#) shows the EMUS enumeration algorithm (Previti & Marques-Silva, 2013), also used in the most recent version of MARCO (Liffiton et al., 2016). It relies on a two-solver approach aimed at enumerating the MUSes/MCSes of an unsatisfiable formula  $\mathcal{F}$ . On the one hand, a formula  $\mathcal{Q}$  is used to enumerate subsets of  $\mathcal{F}$ . This formula is defined over a set of variables  $I = \{p_i \mid c_i \in \mathcal{F}\}$ , each one of them associated with one clause  $c_i \in \mathcal{F}$ . Iteratively until  $\mathcal{Q}$  becomes unsatisfiable, EMUS computes a maximal model  $P$  of  $\mathcal{Q}$  and tests the satisfiability of the corresponding subformula  $\mathcal{F}' \subseteq \mathcal{F}$ . If it is satisfiable,  $\mathcal{F}'$  represents an MSS of  $\mathcal{F}$ , and the clause  $I \setminus P$  is added to  $\mathcal{Q}$ , preventing the algorithm from generating any subset of the MSS (superset of the MCS) again. Otherwise, if  $\mathcal{F}'$  is unsatisfiable, it is reduced to an MUS  $\mathcal{M}$ , which is blocked adding to  $\mathcal{Q}$  a clause made of the variables in  $I$  associated with  $\mathcal{M}$  with negative polarity. This way, no superset of  $\mathcal{M}$  will be generated. [Algorithm 1](#) is guaranteed to find all MUSes and MCSes of  $\mathcal{F}$ , in a number of iterations that corresponds to the sum of the number of MUSes and MCSes.

HGMUS is a novel and efficient group-MUS enumerator for Horn formulae based on implicit minimal hitting set dualization, which incorporates specific features for the problem formulation. In this context,  $\mathcal{H}$  denotes the group of clauses  $\mathcal{G}_0$ , i.e. the background clauses. Moreover,  $\mathcal{I}$  denotes the set of (individual) groups of clauses, with  $\mathcal{I} = \{\mathcal{G}_1, \dots, \mathcal{G}_k\}$ . So, the unsatisfiable group-Horn formula corresponds to  $\mathcal{F} = \mathcal{H} \cup \mathcal{I}$ . Also, the formula  $\mathcal{Q}$  shown in [Algorithm 1](#) is defined on a set of variables associated to the groups in  $\mathcal{I}$ . For the problem instances obtained from axiom pinpointing for the  $\mathcal{EL}$  family of DLs,  $(\mathcal{H}_{[C \sqsubseteq \tau D]})$  each group of clauses contains a single unit clause. However, the algorithm works for arbitrary groups of clauses.

#### 4.3.1 ORGANIZATION OF HGMUS

The high-level organization of HGMUS mimics that of EMUS/MARCO (see [Algorithm 1](#)), with a few essential differences. First, the satisfiability testing step (because it operates on Horn formulae) uses the dedicated linear time algorithm LTUR (Minoux, 1988). Moreover, the simplicity of LTUR enables very efficient implementations, that use adjacency lists for representing clauses instead of the now more commonly used watched literals. Second, the problem formulation motivates using a dedicated MUS extraction algorithm, which is shown to be more effective in this concrete case than other well-known approaches (Belov et al., 2012). Third, we also highlight important aspects of the EMUS/MARCO implicit minimal hitting set dualization approach, which we claim have been overlooked in earlier work (Vescovi, 2011; Sebastiani & Vescovi, 2015).

The following subsections describe the different components of HGMUS in detail.

#### 4.3.2 COMPUTING MAXIMAL MODELS

The use of maximal models for computing either MCSes of a formula or a set of clauses that contain an MUS was proposed in earlier work (Previti & Marques-Silva, 2013), which exploited SAT with preferences for computing maximal models (Giunchiglia & Maratea, 2006; Rosa & Giunchiglia, 2013). The use of SAT with preferences for computing maximal models is also exploited in related work (Sebastiani & Vescovi, 2009, 2015).

Computing maximal models of a formula  $\mathcal{Q}$  can be reduced to the problem of extracting an MSS of a formula  $\mathcal{Q}'$  (Marques-Silva et al., 2013a), where the clauses of  $\mathcal{Q}$  are hard

**Algorithm 2:** Computation of Maximal Models

---

**Input:**  $\mathcal{Q}$  a CNF formula  
**Output:**  $(st, P)$ : with  $st$  a Boolean and  $P$  an MxM (if it exists)

```

1  $(P, U, B) \leftarrow (\{\{x\} \mid \neg x \notin L(\mathcal{Q})\}, \{\{x\} \mid \neg x \in L(\mathcal{Q})\}, \emptyset)$ 
2  $(st, P, U) \leftarrow \text{InitialAssignment}(\mathcal{Q} \cup P)$ 
3 if not  $st$  then return (false,  $\emptyset$ )
4 while  $U \neq \emptyset$  do
5    $l \leftarrow \text{SelectLiteral}(U)$ 
6    $(st, \mu) = \text{SAT}(\mathcal{Q} \cup P \cup B \cup \{l\})$ 
7   if  $st$  then  $(P, U) \leftarrow \text{UpdateSatClauses}(\mu, P, U)$ 
8   else  $(U, B) \leftarrow (U \setminus \{l\}, B \cup \{\neg l\})$ 
9 return (true,  $P$ )                                     //  $P$  is an MxM of  $\mathcal{Q}$ 
```

---

and, for each variable  $x_i \in V(\mathcal{Q})$ , it includes a unit soft clause  $c_i = (x_i)$ . Also, recent work (Marques-Silva et al., 2013a; Grégoire, Lagniez, & Mazure, 2014; Bacchus, Davies, Tsimpoukelli, & Katsirelos, 2014; Mencía, Previti, & Marques-Silva, 2015) has shown that state-of-the-art MCS/MSS computation approaches outperform SAT with preferences. HGMUS uses a dedicated algorithm based on the LinearSearch MCS extraction algorithm (Marques-Silva et al., 2013a), due to its good performance in MCS enumeration. Since all soft clauses are unit, it can also be related with the novel Literal-Based eXtractor algorithm (Mencía et al., 2015). Shown in [Algorithm 2](#), it relies on making successive calls to a SAT solver. It maintains three sets of literals:  $P$ , an under-approximation of an MxM (i.e. positive literals s.t.  $\mathcal{Q} \cup P \not\models \perp$ ),  $B$ , with negative literals  $\neg l$  such that  $\mathcal{Q} \cup P \cup \{l\} \models \perp$  (i.e. *backbone literals*), and  $U$ , with the remaining set of positive literals to be tested. Initially,  $P$  and  $U$  are initialized from a model  $\mu \models \mathcal{Q}$ ,  $P$  including the literals appearing with positive polarity in  $\mu$  and  $U$  including the literals with negative polarity in  $\mu$ . Then, iteratively, it tries to extend  $P$  with a new literal  $l \in U$ , by testing the satisfiability of  $\mathcal{Q} \cup P \cup B \cup \{l\}$ . If it is satisfiable, all the literals in  $U$  satisfied by the model (including  $l$ ) are moved to  $P$ . Otherwise,  $l$  is removed from  $U$  and  $\neg l$  is added to  $B$ . This algorithm has a query complexity of  $\mathcal{O}(|V(\mathcal{Q})|)$ .

[Algorithm 2](#) integrates a new technique, which consists in pre-initializing  $P$  with the pure positive literals appearing in  $\mathcal{Q}$  and  $U$  with the remaining ones (line 1), and then requiring the literals of  $P$  to be satisfied by the initial assignment (line 2). It can be easily proved that these pure literals are included in all MxMs of  $\mathcal{Q}$ , so a number of calls to the SAT solver could be avoided. Moreover, the SAT solver will never branch on these variables, easing the decision problems. Note that, in this context,  $\mathcal{Q}$  is made of two types of clauses: positive clauses blocking MCSes of the Horn formula, and negative clauses blocking MUSes. So, with this technique, the computation of MxMs is restricted to the variables representing groups appearing in some MUS of the Horn formula.<sup>2</sup>

#### 4.3.3 ADDING BLOCKING CLAUSES

One important aspect of HGMUS are the blocking clauses created and added to the formula  $\mathcal{Q}$  (see [Algorithm 1](#)). These follow what was first proposed in eMUS (Previti & Marques-

---

2. SATPin (Manthey & Peñaloza, 2015; Manthey et al., 2016) also exploits this insight of *relevant* variables, but not in the context of MxMs.

Silva, 2013) and MARCO (Liffiton & Malik, 2013; Liffiton et al., 2016). For each MUS, the blocking clause consists of a set of negative literals, requiring at least one of the clauses in the MUS *not* to be included in future selected sets of clauses. For each MCS, the blocking clause consists of a set of positive literals, requiring at least one of the clauses in the MCS to be included in future selected sets of clauses. The way MCSes are handled is essential to prevent that MCS and sets containing the same MCS to be selected again. Although conceptually simple, it can be shown that existing approaches may not guarantee that supersets of MCSes (or subsets of the MSSes) are not selected. As argued before, this is the case with EL<sup>+</sup>SAT (Vescovi, 2011; Sebastiani & Vescovi, 2015).

The following example illustrates the importance of blocking clauses.

**Example 9.** Let us consider the group Horn formula  $\mathcal{H}_{[A \sqsubseteq_{\mathcal{T}_{\text{exa}}} B]}$  in Example 7.  $\mathcal{Q}$  is defined over variables  $\{p_1, \dots, p_5\}$ , each associated with a group in  $\{\mathcal{G}_1, \dots, \mathcal{G}_5\}$ . The following table illustrates the operation of HGMUS, showing, for each iteration, the maximal model, the group-MUS or group-MCS extracted and the blocking clause added to  $\mathcal{Q}$ .

$\mathcal{Q}$	MxM model	MUS	MCS	Blocking clause
$\emptyset$	$\{p_1, p_2, p_3, p_4, p_5\}$	$\{\mathcal{G}_2, \mathcal{G}_4\}$	-	$b_1 = (\neg p_2 \vee \neg p_4)$
$\{b_1\}$	$\{p_1, p_3, p_4, p_5\}$	-	$\{\mathcal{G}_2\}$	$b_2 = (p_2)$
$\{b_1, b_2\}$	$\{p_1, p_2, p_3, p_5\}$	$\{\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_3\}$	-	$b_3 = (\neg p_1 \vee \neg p_2 \vee \neg p_3)$
$\{b_1, b_2, b_3\}$	$\{p_2, p_3, p_5\}$	-	$\{\mathcal{G}_1, \mathcal{G}_4\}$	$b_4 = (p_1 \vee p_4)$
$\{b_1, b_2, b_3, b_4\}$	$\{p_1, p_2, p_5\}$	-	$\{\mathcal{G}_3, \mathcal{G}_4\}$	$b_5 = (p_3 \vee p_4)$
$\{b_1, b_2, b_3, b_4, b_5\}$	UNSAT	-	-	-

If maximal models were blocked (instead of MCSes), the algorithm would require much more iterations. As illustrated in the following table, it would compute a large number of (non minimal) correction subsets (CS) before terminating.

$\mathcal{Q}$	MxM model	MUS	CS	Blocking clause
$\emptyset$	$\{p_1, p_2, p_3, p_4, p_5\}$	$\{\mathcal{G}_2, \mathcal{G}_4\}$	-	$b_1 : (\neg p_2 \vee \neg p_4)$
$\{b_1\}$	$\{p_1, p_3, p_4, p_5\}$	-	$\{\mathcal{G}_2\}$	$b_2 : (\neg p_1 \vee \neg p_3 \vee \neg p_4 \vee \neg p_5)$
$\{b_1, b_2\}$	$\{p_3, p_4, p_5\}$	-	$\{\mathcal{G}_1, \mathcal{G}_2\}$	$b_3 : (\neg p_3 \vee \neg p_4 \vee \neg p_5)$
$\{b_1, \dots, b_3\}$	$\{p_1, p_4, p_5\}$	-	$\{\mathcal{G}_2, \mathcal{G}_3\}$	$b_4 : (\neg p_1 \vee \neg p_4 \vee \neg p_5)$
$\{b_1, \dots, b_4\}$	$\{p_1, p_3, p_4\}$	-	$\{\mathcal{G}_2, \mathcal{G}_5\}$	$b_5 : (\neg p_1 \vee \neg p_3 \vee \neg p_4)$
$\{b_1, \dots, b_5\}$	$\{p_1, p_4\}$	-	$\{\mathcal{G}_2, \mathcal{G}_3, \mathcal{G}_5\}$	$b_6 : (\neg p_1 \vee \neg p_4)$
$\{b_1, \dots, b_6\}$	$\{p_1, p_2, p_3, p_5\}$	$\{\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_3\}$	-	$b_7 : (\neg p_1 \vee \neg p_2 \vee \neg p_3)$
$\{b_1, \dots, b_7\}$	$\{p_1, p_2, p_5\}$	-	$\{\mathcal{G}_3, \mathcal{G}_4\}$	$b_8 : (\neg p_1 \vee \neg p_2 \vee \neg p_5)$
$\{b_1, \dots, b_8\}$	$\{p_1, p_2\}$	-	$\{\mathcal{G}_3, \mathcal{G}_4, \mathcal{G}_5\}$	$b_9 : (\neg p_1 \vee \neg p_2)$
...	...	...	...	...

#### 4.3.4 DECIDING SATISFIABILITY OF HORN FORMULAE

It is well-known that Horn formulae can be decided in linear time (Dowling & Gallier, 1984; Itai & Makowsky, 1987; Minoux, 1988). HGMUS implements the LTUR algorithm (Minoux, 1988). There are important reasons for this choice. First, LTUR is expected to be more efficient than plain unit propagation, since only variables assigned value 1 need to be

**Algorithm 3:** LTUR

---

**Input:**  $\mathcal{H}$ : input Horn formula  
**Output:** falsified clause, if any;  $\alpha$ : antecedents

```

1    $(Q, \eta, \gamma, \alpha) \leftarrow \text{Initialize}(\mathcal{H})$ 
2   while  $Q \neq \emptyset$  do                                //  $Q$ : queue of variables assigned 1
3      $x_j \leftarrow \text{ExtractFirstVariable}(Q)$ 
4     foreach  $c_i \in N(x_j)$  do
5        $\eta(c_i) \leftarrow \eta(c_i) - 1$ 
6       if  $\eta(c_i) = 0$  then
7         if  $\gamma(c_i)$  then return  $(\{c_i\}, \alpha)$ 
8          $x_r \leftarrow \text{PickVariable}(P(c_i))$ 
9         if  $\alpha(x_r) = \emptyset$  then
10        AppendToQueue( $Q, x_r$ )
11         $\alpha(x_r) \leftarrow \{c_i\}$ 
12
13   return  $(\emptyset, \alpha)$ 
```

---

propagated. Second, most implementations of unit propagation in CDCL SAT solvers (i.e. that use watched literals) are not guaranteed to run in linear time (Gent, 2013); this is for example the case with *all* implementations of Minisat (Eén & Sörensson, 2003) and its variants, for which unit propagation runs in worst-case quadratic time. As a result, using an off-the-shelf SAT solver and exploiting only unit propagation (as is done for example in earlier work (Sebastiani & Vescovi, 2009, 2015; Manthey & Peñaloza, 2015)) is unlikely to be the most efficient solution.

The organization of LTUR is summarized in [Algorithm 3](#). LTUR can be viewed as one-sided unit propagation, in the sense that only variables assigned value 1 are propagated; all other variables are assigned value 0 by default. In this context,  $\eta : \mathcal{H} \rightarrow \mathbb{N}_0$  associates a counter with each clause, representing the number of negative literals *not* assigned value 0. The function  $\gamma : \mathcal{H} \rightarrow \{0, 1\}$  denotes whether a clause  $c \in \mathcal{F}$  is a goal clause, in which case  $\gamma(c) = 1$ . Finally,  $\alpha : V(\mathcal{H}) \rightarrow 2^{\mathcal{H}}$  is a function that assigns to each variable  $x$  a set of clauses  $\alpha(x)$  that are deemed responsible for assigning value 1 to  $x$ . If the value of  $x$  is not determined to be 1, then  $\alpha(x) = \emptyset$ . Besides, we denote by  $N(x)$  the set of clauses where the variable  $x$  appears in negative polarity, and  $P(c)$  denotes the set of variables appearing in positive polarity in clause  $c$ . The initialization step sets the initial values of the  $\eta$  counters, the  $\alpha$  values, and the value of the  $\gamma$  flag. The queue  $Q$  is initialized with the variables in the unit positive clauses. The main loop analyzes the variables assigned value 1 in order. Notice that a variable  $x$  is assigned value 1 iff  $\alpha(x) \neq \emptyset$ . For each variable  $x \in Q$ , the counter  $\eta$  of the clauses where  $x$  occurs as a negative literal is decreased. If  $\eta(c) = 0$  for some clause  $c$ , then either the formula is inconsistent, if  $c$  is a goal clause, or the positive literal of  $c$  is assigned value 1 and added to  $Q$ . The process is repeated while variables are assigned value 1.

Besides the advantages listed above, the use of a linear time algorithm for deciding the satisfiability of Horn formulae turns out to be instrumental for MUS extraction, as shown in the next section. In order to use LTUR for MUS extraction, an incremental version has been implemented, which allows for the incremental addition of clauses to the formula and incremental identification of variables assigned value 1. Clearly, the amortized run time of

---

**Algorithm 4:** Insertion-based (de Siqueira N. & Puget, 1988) MUS extraction using LTUR (Minoux, 1988)

---

**Input:**  $\mathcal{H}$ , denotes the  $\mathcal{G}_0$  clauses;  $\mathcal{I}$ , denotes the set of (individual) group clauses  
**Output:**  $\mathcal{M}$ , denotes the computed MUS

```

1   $(\mathcal{M}, c_r) \leftarrow (\mathcal{H}, 0)$ 
2   $\text{LTUR\_prop}(\mathcal{M}, \mathcal{M})$                                 // Start by propagating  $\mathcal{G}_0$  clauses
3  while true do
4    if  $c_r > 0$  then
5       $\mathcal{M} \leftarrow \mathcal{M} \cup \{c_r\}$                                 // Add transition clause  $c_r$  to  $\mathcal{M}$ 
6      if not  $\text{LTUR\_prop}(\mathcal{M}, \{c_r\})$  then
7         $\text{LTUR\_undo}(\mathcal{M}, \mathcal{M})$ 
8        return  $\mathcal{M} \setminus \mathcal{H}$                                 // Remove  $\mathcal{G}_0$  clauses from computed MUS
9
10    $\mathcal{S} \leftarrow \emptyset$ 
11   while true do
12      $c_r \leftarrow \text{SelectRemoveClause}(\mathcal{I})$                       // Target transition clause
13      $\mathcal{S} \leftarrow \mathcal{S} \cup \{c_r\}$ 
14     if not  $\text{LTUR\_prop}(\mathcal{M} \cup \mathcal{S}, \{c_r\})$  then
15        $\mathcal{I} \leftarrow \mathcal{I} \setminus \{c_r\}$                                 // Update working set of groups
16        $\text{LTUR\_undo}(\mathcal{M}, \mathcal{S})$ 
17       break                                //  $c_r$  represents a transition clause

```

---

LTUR, after adding  $m = |\mathcal{F}|$  clauses, is  $\mathcal{O}(|\mathcal{F}|)$ , with  $|\mathcal{F}|$  the number of literals appearing in  $\mathcal{F}$ .

#### 4.3.5 MUS EXTRACTION IN HORN FORMULAE

For arbitrary CNF formulae, a number of approaches exist for MUS extraction, with the most commonly used one being the deletion-based approach (Bakker, Dikker, Tempelman, & Wognum, 1993; Belov et al., 2012), but other alternatives include the QuickXplain algorithm (Junker, 2004) and the more recent Progression algorithm (Marques-Silva, Janota, & Belov, 2013b). It is also well-known and generally accepted that, due to its query complexity, the insertion-based algorithm (de Siqueira N. & Puget, 1988) for MUS extraction is in practice not competitive with existing alternatives (Belov et al., 2012).

Somewhat surprisingly, this is not the case with Horn formulae when (an incremental implementation of) the LTUR algorithm is used. A modified insertion-based MUS extraction algorithm that exploits LTUR is shown in [Algorithm 4](#).  $\text{LTUR\_prop}$  propagates the consequences of adding some new set of clauses, given some existing incremental context.  $\text{LTUR\_undo}$  unpropagates the consequences of adding some set of clauses (in order), given some existing incremental context. Besides, in this algorithm, clauses are represented as integers (0 means no clause). The organization of the algorithm mimics the standard insertion-based MUS extraction algorithm (de Siqueira N. & Puget, 1988), but the use of the incremental LTUR yields run time complexity that improves over other approaches. Consider the operation of the standard insertion-based algorithm (de Siqueira N. & Puget, 1988), in which clauses are iteratively added to the working formula. When the formula becomes unsatisfiable, a *transition clause* (Belov et al., 2012) has been identified, which is then added to the MUS being constructed. The well-known query complexity of the

insertion-based algorithm is  $\mathcal{O}(m \times k)$  where  $m$  is the number of clauses and  $k$  is the size of a largest MUS. Now consider that the incremental LTUR algorithm is used. To find the first transition clause, the amortized run time is  $\mathcal{O}(|\mathcal{F}|)$ . Clearly, this holds true for *any* transition clause, and so the run time of MUS extraction with the LTUR algorithm becomes  $\mathcal{O}(|\mathcal{M}| \times |\mathcal{F}|)$ , where  $\mathcal{M} \subseteq \mathcal{I}$  is a largest MUS. [Algorithm 4](#) highlights the main differences with respect to a standard insertion-based MUS extraction algorithm. In contrast, observe that for a deletion-based algorithm the run time complexity will be  $\mathcal{O}(|\mathcal{I}| \times |\mathcal{F}|)$ . In situations where the sizes of MUSes are much smaller than the number of groups in  $\mathcal{I}$ , this difference can be significant. As a result, when extracting MUSes from Horn formulae, and when using a polynomial time incremental decision procedure, an insertion-based algorithm should be used instead of other more commonly used alternatives.

## 5. Experimental Results

This section<sup>3</sup> evaluates the methods proposed in the paper, EL2MCS and HGMUS, and compares them with other state-of-the-art tools for axiom pinpointing in  $\mathcal{EL}^+$ . For this purpose, a set of benchmarks from well-known bio-medical ontologies was considered. These have been used in earlier work, e.g. (Sebastiani & Vescovi, 2009, 2015; Arif et al., 2015a, 2015b; Manthey & Peñaloza, 2015; Manthey et al., 2016). Since all experiments consist of converting axiom pinpointing problems into group-MUS enumeration problems, the tool that uses HGMUS<sup>4</sup> as its back-end is named EL2MUS. Thus, in this section, the results for EL2MUS illustrate the performance of the group-MUS enumerator described in [Section 4.3](#). HGMUS was also used as a back-end in the BEACON tool (Arif et al., 2016), which incorporates its own implementation of the Horn encoding used by  $\mathcal{EL}^+$ SAT (Sebastiani & Vescovi, 2009; Vescovi, 2011; Sebastiani & Vescovi, 2015). BEACON also integrates the state-of-the-art tool FORQES (Ignatiev et al., 2015), also based on implicit minimal hitting set dualization, for computing smallest MUSes, which correspond to the smallest MinAs for a given subsumption relation w.r.t. a TBox. In this section, EL2MUS uses  $\mathcal{EL}^+$ SAT as a front-end (for generating the Horn encoding) for comparison purposes, but the BEACON tool's front-end could be used instead.

### 5.1 Experimental Setup

Each considered instance represents the problem of explaining a particular subsumption relation (query) entailed in a medical ontology. Four standard medical ontologies<sup>5</sup> are considered: GALEN (Rector & Horrocks, 1997), the Gene Ontology (GO) (Ashburner, Ball, Blake, Botstein, Butler, Cherry, Davis, Dolinski, Dwight, Eppig, & et al., 2000), NCI (Sioutos, de Coronado, Haber, Hartel, Shaiu, & Wright, 2007) and SNOMED CT (version 2009) (Spackman et al., 1997). For GALEN, we consider two variants: FULL-GALEN and NOT-GALEN. The most important ontology is SNOMED CT and, due to its huge size, it also produces the hardest axiom pinpointing instances. For each ontology

- 
- 3. Additional material and detailed results for the submission are available at <http://logos.ucd.ie/web/doku.php?id=jair16-submission>.
  - 4. HGMUS is available at <http://logos.ucd.ie/web/doku.php?id=hgmus>.
  - 5. GO, GALEN and NCI ontologies are freely available at <http://lat.inf.tu-dresden.de/~meng/toyont.html>. The SNOMED CT ontology was requested from IHTSDO under a nondisclosure license agreement.

(including the GALEN variants) 100 queries are considered; 50 random (expected to be easier) and 50 sorted (expected to have a large number of minimal explanations) queries. These queries were proposed in previous work, e.g. (Vescovi, 2011; Sebastiani & Vescovi, 2015). Hence, the experimental setup consists of a total of 500 axiom pinpointing problem instances.

As mentioned before, all the instances have been obtained from the encoding proposed in (Sebastiani & Vescovi, 2009; Vescovi, 2011; Sebastiani & Vescovi, 2015). In addition, two different experiments were considered by applying two different simplification techniques to the problem instances, both of which were proposed in (Vescovi, 2011; Sebastiani & Vescovi, 2015). Simplification techniques are of common use in axiom pinpointing, such as the so-called *reachability-based modularization* in the DL domain (Baader & Suntisrivaraporn, 2008). These techniques allow for extracting an (often small) subset of the ontology such that it contains all the MinAs and diagnoses for a given subsumption relation, and their application often plays an important role in the efficiency of axiom pinpointing tools. The first technique considered in these experiments uses the *Cone-Of-Influence* (COI) reduction. These are reduced instances in both the size of the Horn formula and the number of original axioms in the ontology, but are still quite large. Similar techniques are exploited in related work (Baader et al., 2006; Ludwig, 2014; Manthey & Peñaloza, 2015; Manthey et al., 2016). The second one considers the more effective reduction technique (referred to as x2), consisting in applying the COI technique, re-encoding the Horn formula into a reduced ontology, and encoding this ontology again into a Horn formula. This results in small Horn formulae, which will be useful to evaluate the algorithms when there are a large number of MUSes or MCSes.

The experiments compare EL2MCS and EL2MUS to different algorithms; specifically, EL<sup>+</sup>SAT (Sebastiani & Vescovi, 2009, 2015), CEL (Baader et al., 2006), JUST (Ludwig, 2014) and SATPin (Manthey & Peñaloza, 2015; Manthey et al., 2016). EL<sup>+</sup>SAT has been previously shown to outperform CEL, whereas SATPin has been shown to outperform the MUS enumerator MARCO (Liffiton et al., 2016).

The comparison with CEL and JUST imposes a number of constraints. First, CEL only computes 10 MinAs, so all comparisons with CEL only consider reporting the first 10 MinAs/MUSes. Also, CEL uses a simplification technique similar to COI, so CEL is considered in the first experiments. Second, JUST operates on selected subsets of  $\mathcal{EL}^+$ , since it is unable to handle general role inclusions. As a result, all comparisons with JUST consider solely the problem instances for which JUST can compute correct results. JUST accepts the simplified x2 ontologies, so it is considered in the second series of experiments. The comparison with these tools is presented at the end of the section.

EL2MUS interfaces the SAT solver Minisat 2.2 (Eén & Sörensson, 2003) for computing maximal models. All the experiments were performed on a Linux cluster (2 GHz) and the algorithms were given a time limit of 3600s and a memory limit of 4 GB.

## 5.2 Assessment of SAT-Based Approaches

The first series of experiments aims at evaluating the performance of EL2MCS and EL2MUS compared to EL<sup>+</sup>SAT and SATPin. All these methods work on the Horn encoding provided

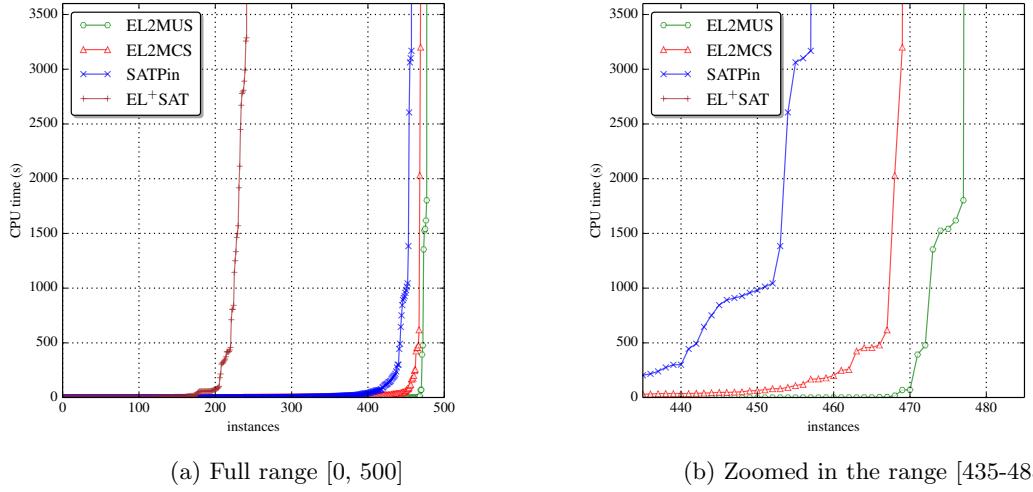


Figure 3: Cactus plots comparing  $\text{EL}^+\text{SAT}$ , SATPin, EL2MCS and EL2MUS on the COI instances

by  $\text{EL}^+\text{SAT}$ . Hence, the systems were run on the same Horn formulae, simplified with either the COI or x2 reduction techniques. The results are presented in separate subsections.

### 5.2.1 COI INSTANCES

For the COI instances, Figure 3 summarizes the results for  $\text{EL}^+\text{SAT}$ , EL2MCS, SATPin and EL2MUS. It shows the running times for solving the different axiom pinpointing instances, i.e. enumerating all the MinAs and proving that there is no MinA left. As it can be easily observed, EL2MCS has a slight performance advantage over SATPin which, in turn solves much more instances than  $\text{EL}^+\text{SAT}$ . On the other hand, EL2MUS terminates for more instances than any of the other tools. Noticeably, by the time limit EL2MCS is able to solve all the instances  $\text{EL}^+\text{SAT}$  or SATPin can solve, and EL2MUS solves all the instances solved by EL2MCS. The differences among SATPin, EL2MCS and ELMUS can be better observed in Figure 3b, which shows a clear performance gap in favor of EL2MCS and EL2MUS.

Figure 4 shows scatter plots comparing EL2MUS with the different tools. From this figure it can be concluded that the performance of EL2MUS exceeds the performance of any of the other tools by at least one order of magnitude (and often by more), except for a few outlying instances. Figure 4d summarizes the results in the scatter plots, where the percentages shown are computed for problem instances for which at least one of the tools takes more than 0.001s. The first four rows of the table show a pairwise comparison among the methods, reporting the percentage of the instances each method takes less time than other ones in solving the instances. The last five rows of the table show the percentage of the instances for which the performance gap in favor of EL2MUS is greater than one to five orders of magnitude. Observe that EL2MUS outperforms all other tools in all of the problem instances and, for many cases, with two or more orders of magnitude improvement.

The results presented so far focus on the ability of the tools for completing the enumeration of the MinAs, and do not provide information about their performance when a complete enumeration is not possible. Table 2 shows, for each family of instances (each with 100), the number of instances solved (#Sol.) and the total number of MinAs computed (#Mi-

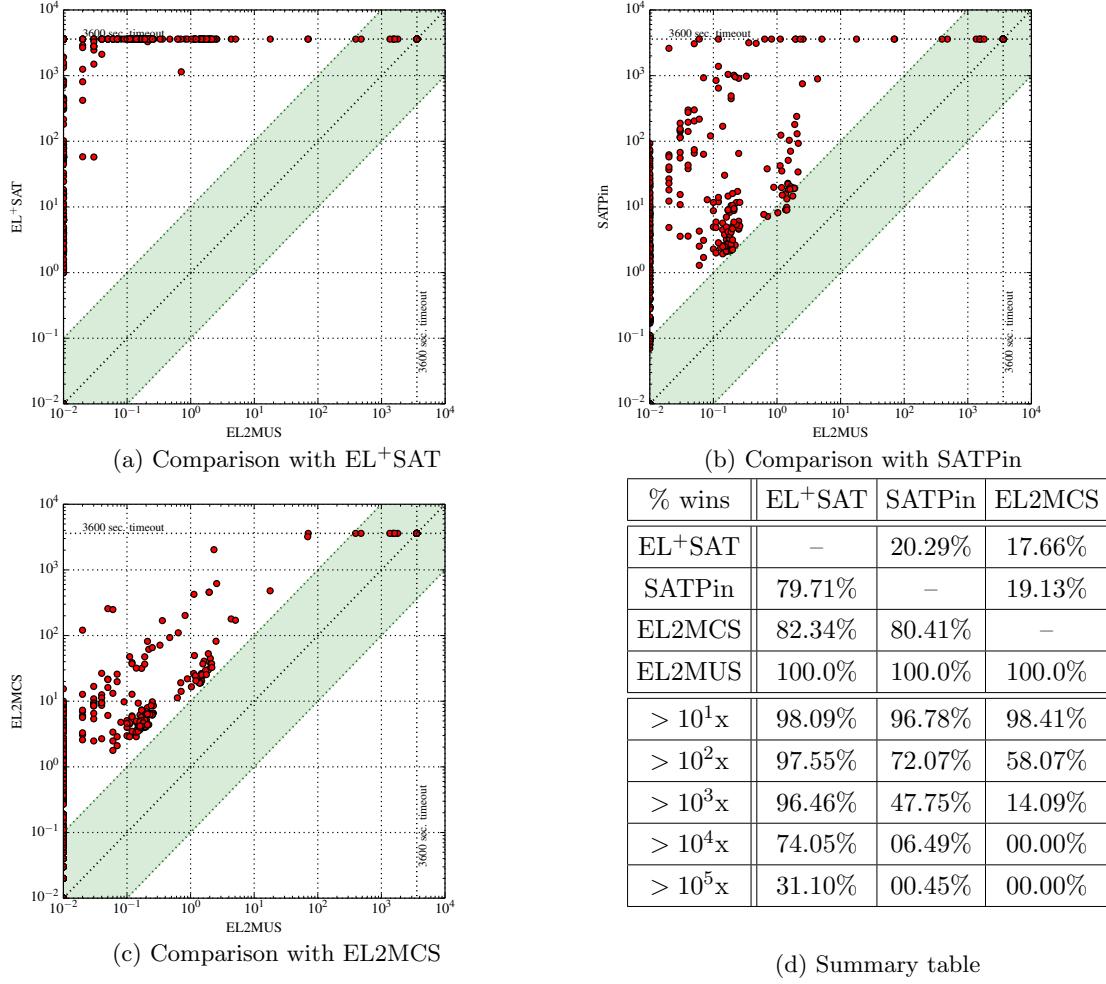
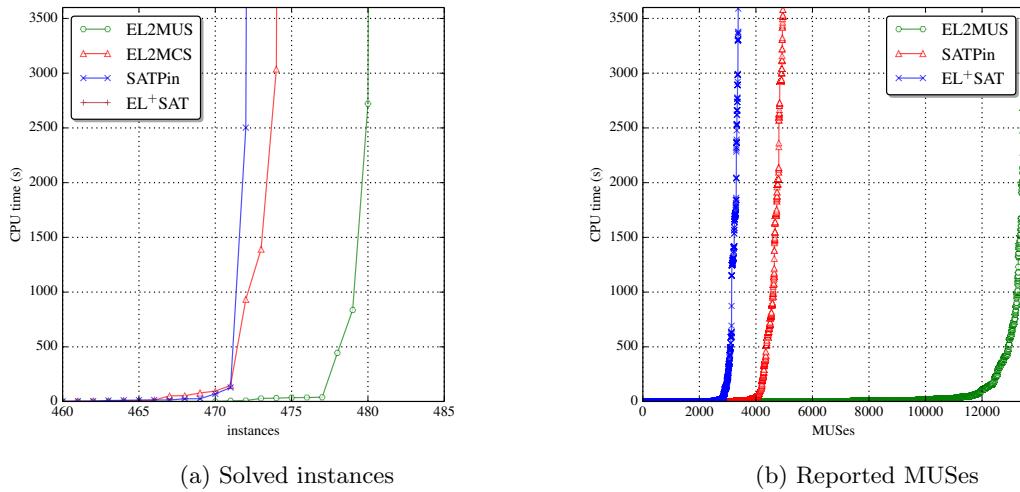


Figure 4: Scatter plots for COI instances

nAs) by the time limit by each of the methods (including this information for the instances that were not solved). Since EL2MCS and EL2MUS compute diagnoses as well, the total number of them is also reported (#Diag.) for these tools. The hardest instances come from the SNOMED CT ontology, where EL2MUS is able to report more MinAs in about a factor of 10 than any other tool by the time limit. For the other ontologies, both EL2MCS and EL2MUS are able to solve all the problem instances, SATPin is able to solve most of them and EL<sup>+</sup>SAT computes most of the MinAs by the time limit, although it is often unable to terminate. Noticeably, the few SNOMED CT instances that cannot be solved by EL2MCS and EL2MUS contain a very large number of diagnoses (that is, MCSes). This illustrates the main drawback of EL2MCS which, in contrast to the other tools is unable to report a single MinA by the time limit in these cases as a result of being unable to complete the enumeration of the diagnoses. At the same time, the results suggest that the number of diagnoses for a given subsumption relation plays an important role in the difficulty of axiom pinpointing. EL2MUS stands out in both its capability for computing diagnoses and MinAs for these hardest instances compared to EL2MCS and any other tool respectively.

Table 2: SAT-based approaches. COI instances

Ontology	EL <sup>+</sup> SAT		SATPin		EL2MCS			EL2MUS		
	#Sol.	#MinA	#Sol.	#MinA	#Sol.	#MinA	#Diag.	#Sol.	#MinA	#Diag.
GENE	87	1126	100	1134	100	1134	20011	100	1134	20011
NCI	77	664	93	661	100	678	78989	100	678	78989
NOT-GALEN	34	159	100	159	100	159	702	100	159	702
FULL-GALEN	17	137	100	139	100	139	776	100	139	776
SNOMED CT	27	1002	65	973	70	439	78839	78	9349	3199135
<b>Total</b>	<b>242</b>	<b>3088</b>	<b>458</b>	<b>3066</b>	<b>470</b>	<b>2549</b>	<b>179317</b>	<b>478</b>	<b>11459</b>	<b>3299613</b>

Figure 5: Cactus plots comparing EL<sup>+</sup>SAT, SATPin, EL2MCS and EL2MUS on the x2 instances

### 5.2.2 x2 INSTANCES

The x2 instances contain exactly the same MinAs and diagnoses than the COI instances, but these are significantly simpler in terms of size. Thus, whereas the COI instances can serve to assess the scalability of each approach, the x2 instances highlight the expected performance in representative settings for axiom pinpointing.

Figure 5a summarizes the performance of the tools EL<sup>+</sup>SAT, SATPin, EL2MCS and EL2MUS. Due to its poor performance, EL<sup>+</sup>SAT is not included in this plot. This tool terminated on only 317 instances. As in the previous experiment, EL2MUS exhibits an observable performance edge in terms of solved instances.

A pairwise comparison between the different tools is summarized in Figure 6. Although not as impressive as for the COI instances, EL2MUS still consistently outperforms all other tools. Figure 6d summarizes the results, where as before the percentages shown are computed for problem instances for which at least one of the tools takes more than 0.001s. Observe that, for these easier instances, SATPin becomes competitive with EL2MUS. Nevertheless, for instances taking more than 0.1s, EL2MUS outperforms SATPin on 100% of the instances. Thus, the 67.69% shown in the table result from instances for which both SATPin and EL2MUS take at most 0.04s. The summary table also lists the number of computed MUSes/MinAs for the 19 instances for which EL2MUS does *not* terminate (all of the other tools also do not terminate for these 19 instances). EL2MUS computes 9948 MUSes in total. As can be observed from the table, the other tools lag behind, and com-

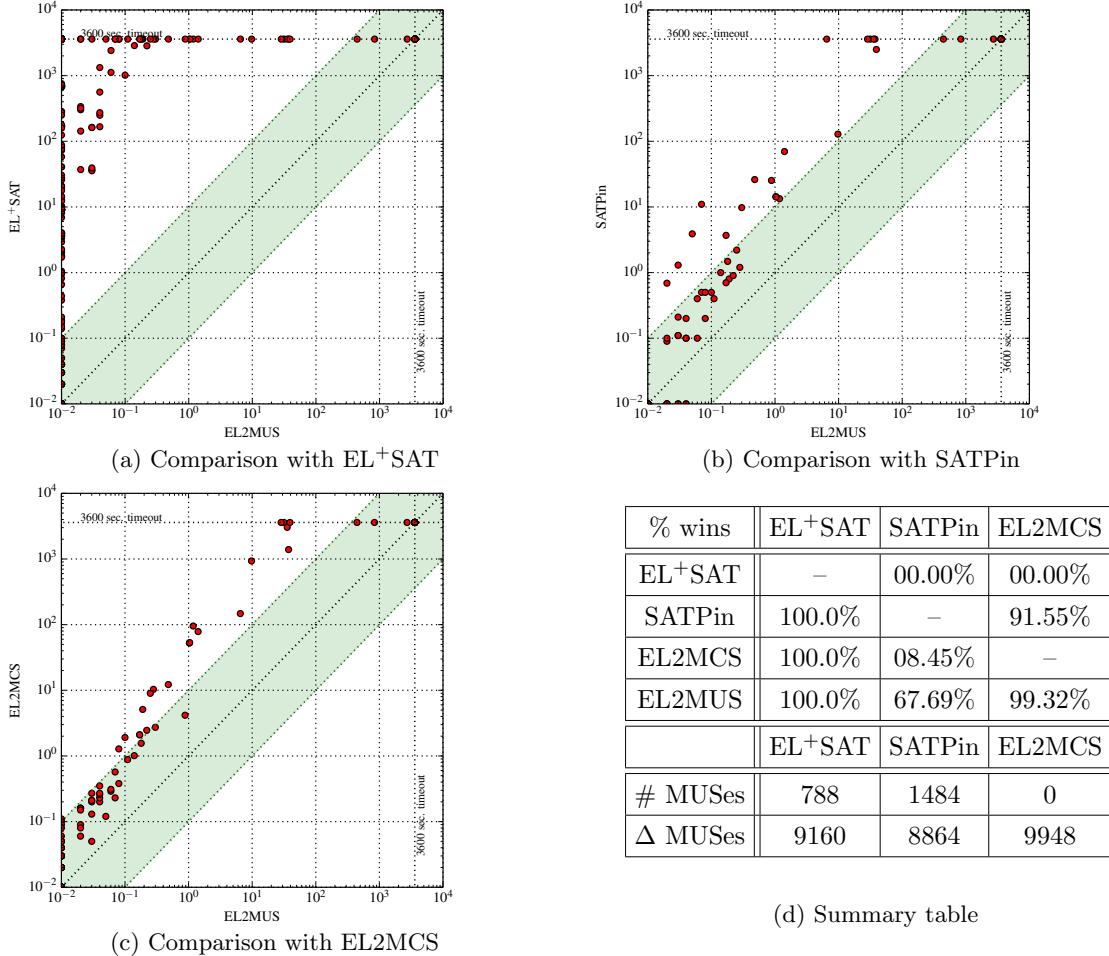


Figure 6: Scatter plots for x2 instances

pute significantly fewer MUSes. Also, as noted earlier in the paper, the main issue with EL2MCS is demonstrated with these results; for these 19 instances, EL2MCS is unable to compute any MUSes. The comparison with the other tools,  $\text{EL}^+\text{SAT}$  and SATPin, reveals that EL2MUS computes respectively in excess of a factor of 10 and of 5 more MUSes.

EL2MUS not only terminates on more instances than any other approach and computes more MUSes for the unsolved instances; it also reports the sequences of MUSes much faster. Figure 5b shows, for each computed MUS over the whole set of instances, the time each MUS was reported. This figure compares  $\text{EL}^+\text{SAT}$ , SATPin and EL2MUS, as these are the only methods able to report MUSes from the beginning. The results confirm that EL2MUS is able to find many more MUSes in less time than the alternatives.

Finally, Table 3 summarizes the results for each family of problem instances. As can be observed, all the tools exhibit a clear improvement w.r.t. the COI instances (see Table 2). The simplified instances allow  $\text{EL}^+\text{SAT}$ , SATPin, EL2MCS and EL2MUS for solving 75, 15, 5 and 3 more instances respectively. The improvement of EL2MCS and EL2MUS in terms of solved instances is not as dramatic as for the other tools, since the open instances are really challenging, as suggested by the huge number of diagnoses reported by EL2MUS for the

Table 3: SAT-based approaches. x2 instances

<b>Ontology</b>	<b>EL<sup>+</sup>SAT</b>		<b>SATPin</b>		<b>EL2MCS</b>			<b>EL2MUS</b>		
	#Sol.	#MinA	#Sol.	#MinA	#Sol.	#MinA	#Diag.	#Sol.	#MinA	#Diag.
GENE	99	1134	100	1134	100	1134	20011	100	1134	20011
NCI	86	672	100	678	100	678	78989	100	678	78989
NOT-GALEN	77	159	100	159	100	159	702	100	159	702
FULL-GALEN	24	137	100	139	100	139	776	100	139	776
SNOMED CT	31	1570	73	2847	75	919	22330370	81	11364	32430103
<b>Total</b>	<b>317</b>	<b>3672</b>	<b>473</b>	<b>4957</b>	<b>475</b>	<b>3029</b>	<b>22430848</b>	<b>481</b>	<b>13474</b>	<b>32530581</b>

SNOMED CT instances. These very large numbers confirm that the number of diagnoses serves as a very accurate indicator of the difficulty of the axiom pinpointing problems. It should be noted that EL2MUS is able to solve more COI instances (478) than any other tool x2 instances, and the same applies for the number of reported MinAs. Noticeably, EL2MCS improves to a great extent regarding its capability for computing diagnosis and EL2MUS is able to compute a factor of 10 more diagnoses for the unsolved instances w.r.t the COI instances. In the x2 instances EL2MUS is able to compute around 33% more diagnoses than EL2MCS.

These experimental results confirm the superiority of EL2MUS over any other SAT-based axiom pinpointing tool. Not only EL2MUS is able to complete the enumeration of the MinAs for more instances than the other tools, but it also shows a clear advantage in enumerating MinAs for the problem instances that cannot be solved by the time limit, as well as in computing diagnoses.

### 5.3 Assessment of Other Axiom Pinpointing Tools

This section compares EL2MUS to DL-based axiom pinpointing tools, namely CEL (Baader et al., 2006) and JUST (Ludwig, 2014). Figure 7 shows scatter plots comparing EL2MUS with CEL and JUST, respectively for the COI and x2 instances.

As indicated earlier, CEL only computes 10 MinAs, and so the run times shown are for computing the first 10 MinA/MUSes. For problem instances with less than 10 MinAs, the reported run time corresponds to the time taken by the algorithms in terminating after a complete enumeration of the MinAs. As can be observed, the performance edge of EL2MUS is clear, with the performance gap exceeding 1 order of magnitude almost without exception.

JUST (Ludwig, 2014) is a recent state-of-the-art axiom pinpointing tool for the less expressive  $\mathcal{ELH}$  DL. Thus, not all subsumption relations can be represented and analyzed. The results shown are for the subsumption relations for which JUST gives the correct results. In total, 382 instances could be considered and are shown in the plot. As before, the performance edge of EL2MUS is clear, with the performance gap exceeding 1 order of magnitude without exception. In this case, since the x2 instances are in general much simpler, the performance gap is even more significant.

## 6. Conclusions

Axiom pinpointing is a fundamental reasoning task for the development and maintenance of description logic ontologies, as it constitutes the basis for debugging and repair of the

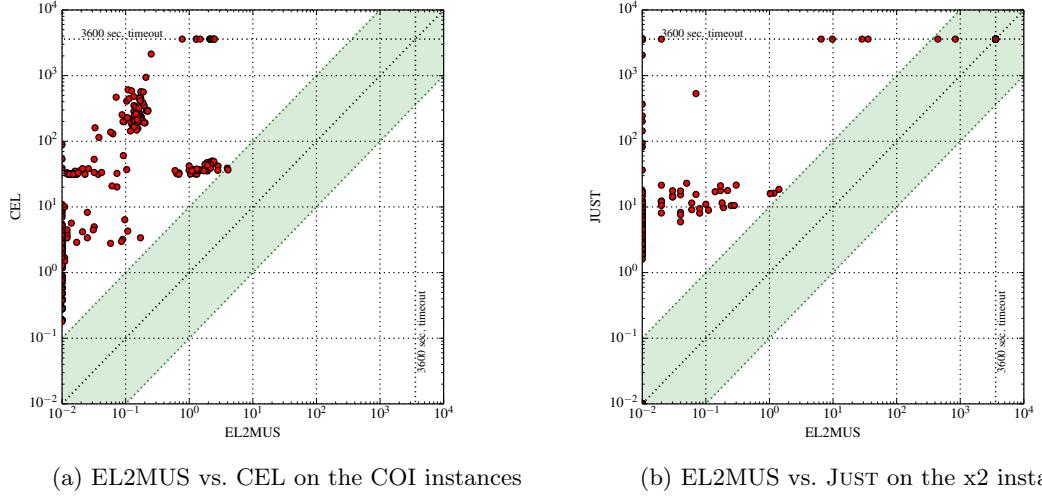


Figure 7: Comparison of EL2MUS with CEL and with JUST

specified knowledge. This paper builds on earlier work relating axiom pinpointing in the  $\mathcal{EL}$  family of lightweight DLs and propositional satisfiability. The present work makes several contributions towards efficient SAT-based axiom pinpointing in these logics, which can be applied in existing large-scale ontologies.

First, the paper shows that given the propositional Horn encoding of the classification of an ontology proposed in earlier work (Sebastiani & Vescovi, 2009, 2015), axiom pinpointing in the  $\mathcal{EL}^+$  corresponds to enumerating all the group-MUSes of a Horn formula. This reduction serves to capitalize on state-of-the-art SAT-based approaches for addressing group-MUS enumeration as a back-end for an efficient enumeration of all MinAs of a given subsumption relation.

Building on recent work on MUS enumeration, the paper proposes two approaches for enumerating the group-MUSes of Horn formulae. In particular, these approaches yield new methods for enumerating all MinAs and solving other problems related to axiom pinpointing. Both methods rely on the well-known hitting set duality relationship between MUSes and MCSes of propositional formulae. The first method, known as EL2MCS, follows an explicit hitting set dualization scheme, in which all the group-MCSes are computed first and then the group-MUSes are obtained by computing the minimal hitting sets of this set of group-MCSes. The second method, called HGMUS, develops an implicit hitting set dualization approach, interleaving the computation of group-MCSes and group-MUSes of the Horn formula. HGMUS builds on partial MUS enumerators, such as other MUS tools like MARCO and EMUS, but incorporates a number of features specifically designed for dealing with Horn formulae which improve the overall performance of the method.

It is interesting to notice that, although the work on axiom pinpointing in DLs usually focuses on finding or enumerating MinAs, there exist several related tasks where computing repairs is more important. For example, in error-tolerant reasoning (Ludwig & Peñaloza, 2014; Bienvenu & Rosati, 2013; Bienvenu, Bourgaux, & Goasdoué, 2014) and iterative ontology update (Peñaloza & Thuluva, 2015), one is interested in analysing some properties on the set of all repairs for a consequence. Given the close connection between the repairs of a consequence and the MCSes obtained from the Horn encoding, the two methods presented

here are directly capable of enumerating repairs, making them specially suitable in those circumstances.

Both methods were implemented, and their performance compared to other existing techniques for axiom pinpointing in lightweight DLs over a large set of benchmarks derived from existing ontologies developed for the bio-medical domains. The experimental results obtained indicate very remarkable improvements of the proposed methods over the state of the art. Both EL2MCS and HGMUS are able to solve more instances than any other method. The results also show a clear advantage of HGMUS over EL2MCS, specially for instances where there is a very large number of MCSes/diagnoses, where EL2MCS is often unable to deliver a single MUS/MinA, whereas HGMUS is able to compute many solutions. In a few of these hard instances, HGMUS was able to generate all MinAs; a feat that had not been accomplished by any tool before.

The results obtained with these tools suggest that propositional satisfiability can contribute very significantly to the area of description logics in general, and in particular for axiom pinpointing and other non-standard reasoning tasks. This positive outcome encourages further research on establishing deeper relationships between DL reasoning problems and SAT. This deeper interleaving of both areas could result in the development of more efficient algorithms for solving a wide range of problems in Description Logics. In the short term, one of the important questions to be answered is whether some of the techniques developed in this paper can be extended to cover more expressive DLs. It would be also interesting to understand how to adapt the methods to solve other related reasoning tasks efficiently.

## Acknowledgments

This work was funded in part by SFI grant BEACON (09/IN.1/I2618), by DFG grant DFG HO 1294/11-1, and by Spanish grant TIN2013-46511-C2-2-P. The contribution of the researchers associated with the SFI grant BEACON is also acknowledged.

## References

- Arif, M. F., Mencía, C., Ignatiev, A., Manthey, N., Peñaloza, R., & Marques-Silva, J. (2016). BEACON: an efficient SAT-based tool for debugging *EL+* ontologies. In *SAT*, pp. 521–530.
- Arif, M. F., Mencía, C., & Marques-Silva, J. (2015a). Efficient axiom pinpointing with EL2MCS. In *KI*, pp. 225–233.
- Arif, M. F., Mencía, C., & Marques-Silva, J. (2015b). Efficient MUS enumeration of Horn formulae with applications to axiom pinpointing. In *SAT*, pp. 324–342.
- Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., Davis, A. P., Dolinski, K., Dwight, S. S., Eppig, J. T., & et al. (2000). Gene ontology: tool for the unification of biology. *Nat. Genet.*, 25(1), 25–29.
- Baader, F., Brandt, S., & Lutz, C. (2005). Pushing the  $\mathcal{EL}$  envelope. In *IJCAI*, pp. 364–369.

- Baader, F., Calvanese, D., McGuinness, D., Nardi, D., & Patel-Schneider, P. F. (Eds.). (2003). *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press.
- Baader, F., & Hollunder, B. (1995). Embedding defaults into terminological knowledge representation formalisms. *J. Autom. Reasoning*, 14(1), 149–180.
- Baader, F., Lutz, C., & Suntisrivaraporn, B. (2006). CEL - A polynomial-time reasoner for life science ontologies. In *IJCAR*, pp. 287–291.
- Baader, F., & Peñaloza, R. (2010). Axiom pinpointing in general tableaux. *J. Log. Comput.*, 20(1), 5–34.
- Baader, F., Peñaloza, R., & Suntisrivaraporn, B. (2007). Pinpointing in the description logic  $\mathcal{EL}^+$ . In *KI*, pp. 52–67.
- Baader, F., & Suntisrivaraporn, B. (2008). Debugging SNOMED CT using axiom pinpointing in the description logic  $\mathcal{EL}^+$ . In *KR-MED*.
- Bacchus, F., Davies, J., Tsimpoukelli, M., & Katsirelos, G. (2014). Relaxation search: A simple way of managing optional clauses. In *AAAI*, pp. 835–841.
- Bailey, J., & Stuckey, P. J. (2005). Discovery of minimal unsatisfiable subsets of constraints using hitting set dualization. In *PADL*, pp. 174–186.
- Bakker, R. R., Dikker, F., Tempelman, F., & Wognum, P. M. (1993). Diagnosing and solving over-determined constraint satisfaction problems. In *IJCAI*, pp. 276–281.
- Belov, A., Lynce, I., & Marques-Silva, J. (2012). Towards efficient MUS extraction. *AI Commun.*, 25(2), 97–116.
- Bienvenu, M., Bourgaux, C., & Goasdoué, F. (2014). Querying inconsistent description logic knowledge bases under preferred repair semantics. In Brodley, C. E., & Stone, P. (Eds.), *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, pp. 996–1002. AAAI Press.
- Bienvenu, M., & Rosati, R. (2013). Tractable approximations of consistent query answering for robust ontology-based data access. In Rossi, F. (Ed.), *Proceedings of the 23rd International Joint Conference on Artificial Intelligence*. IJCAI/AAAI.
- Biere, A., Heule, M., van Maaren, H., & Walsh, T. (Eds.). (2009). *Handbook of Satisfiability*, Vol. 185 of *Frontiers in Artificial Intelligence and Applications*. IOS Press.
- Birnbaum, E., & Lozinskii, E. L. (2003). Consistent subsets of inconsistent systems: structure and behaviour. *J. Exp. Theor. Artif. Intell.*, 15(1), 25–46.
- Cook, S. A. (1971). The complexity of theorem-proving procedures. In *STOC*, pp. 151–158.
- de Siqueira N., J. L., & Puget, J.-F. (1988). Explanation-based generalisation of failures. In *ECAI*, pp. 339–344.
- Dowling, W. F., & Gallier, J. H. (1984). Linear-time algorithms for testing the satisfiability of propositional Horn formulae. *J. Log. Program.*, 1(3), 267–284.
- Eén, N., & Sörensson, N. (2003). An extensible SAT-solver. In *SAT*, pp. 502–518. MiniSat 2.2 is available from: <https://github.com/niklasso/minisat.git>.

- Gent, I. (2013). Optimal implementation of watched literals and more general techniques. *Journal of Artificial Intelligence Research*, 48, 231–252.
- Giunchiglia, E., & Maratea, M. (2006). Solving optimization problems with DLL. In *ECAI*, pp. 377–381.
- Grégoire, É., Lagniez, J., & Mazure, B. (2014). An experimentally efficient method for (MSS, CoMSS) partitioning. In *AAAI*, pp. 2666–2673.
- Hébert, C., Bretto, A., & Crémilleux, B. (2007). A data mining formalization to improve hypergraph minimal transversal computation.. *Fundamenta Informaticae*, 80(4), 415–433.
- Heras, F., Morgado, A., & Marques-Silva, J. (2015). MaxSAT-based encodings for group MaxSAT. *AI Commun.*, 28(2), 195–214.
- Ignatiev, A., Previti, A., Liffiton, M., & Marques-Silva, J. (2015). Smallest MUS extraction with minimal hitting set dualization. In *CP*.
- Itai, A., & Makowsky, J. A. (1987). Unification as a complexity measure for logic programming. *J. Log. Program.*, 4(2), 105–117.
- Junker, U. (2004). QuickXplain: Preferred explanations and relaxations for over-constrained problems. In *AAAI*, pp. 167–172.
- Kalyanpur, A., Parsia, B., Horridge, M., & Sirin, E. (2007). Finding all justifications of OWL DL entailments. In *ISWC*, pp. 267–280.
- Kalyanpur, A., Parsia, B., Sirin, E., & Grau, B. C. (2006). Repairing unsatisfiable concepts in owl ontologies... pp. 170–184.
- Lahiri, S. K., Nieuwenhuis, R., & Oliveras, A. (2006). SMT techniques for fast predicate abstraction. In *CAV*, pp. 424–437.
- Liffiton, M. H., & Malik, A. (2013). Enumerating infeasibility: Finding multiple MUSes quickly. In *CPAIOR*, pp. 160–175.
- Liffiton, M. H., Previti, A., Malik, A., & Marques-Silva, J. (2016). Fast, flexible MUS enumeration. *Constraints*, 21(2), 223–250.
- Liffiton, M. H., & Sakallah, K. A. (2008). Algorithms for computing minimal unsatisfiable subsets of constraints. *J. Autom. Reasoning*, 40(1), 1–33.
- Ludwig, M. (2014). Just: a tool for computing justifications w.r.t. ELH ontologies. In *ORE*.
- Ludwig, M., & Peñaloza, R. (2014). Error-tolerant reasoning in the description logic  $\mathcal{EL}$ . In *JELIA*, pp. 107–121.
- Manthey, N., & Peñaloza, R. (2015). Exploiting SAT technology for axiom pinpointing. Tech. rep. LTCS 15-05, Chair of Automata Theory, Institute of Theoretical Computer Science, Technische Universität Dresden. Available from <https://ddll.inf.tu-dresden.de/web/Techreport3010>.
- Manthey, N., Peñaloza, R., & Rudolph, S. (2016). Efficient axiom pinpointing in EL using SAT technology. In *Proceedings of the 29th International Workshop on Description Logics*, Vol. 1577 of *CEUR Workshop Proceedings*. CEUR-WS.org.

- Marques-Silva, J., Heras, F., Janota, M., Previti, A., & Belov, A. (2013a). On computing minimal correction subsets. In *IJCAI*, pp. 615–622.
- Marques-Silva, J., Janota, M., & Belov, A. (2013b). Minimal sets over monotone predicates in Boolean formulae. In *CAV*, pp. 592–607.
- Mencía, C., Previti, A., & Marques-Silva, J. (2015). Literal-based MCS extraction. In *IJCAI*, pp. 1973–1979.
- Meyer, T. A., Lee, K., Booth, R., & Pan, J. Z. (2006). Finding maximally satisfiable terminologies for the description logic  $\mathcal{ALC}$ . In *AAAI*, pp. 269–274.
- Minoux, M. (1988). LTUR: A simplified linear-time unit resolution algorithm for Horn formulae and computer implementation. *Inf. Process. Lett.*, 29(1), 1–12.
- Moodley, K., Meyer, T., & Varzinczak, I. J. (2011). Root justifications for ontology repair. In *RR*, pp. 275–280.
- Morgado, A., Liffiton, M. H., & Marques-Silva, J. (2012). MaxSAT-based MCS enumeration. In *HVC*, pp. 86–101.
- Murakami, K., & Uno, T. (2011). Efficient algorithms for dualizing large-scale hypergraphs. *CoRR*, *abs/1102.3813*.
- Nguyen, H. H., Alechina, N., & Logan, B. (2012). Axiom pinpointing using an assumption-based truth maintenance system. In *DL*.
- O’Sullivan, B., Papadopoulos, A., Faltings, B., & Pu, P. (2007). Representative explanations for over-constrained problems. In *AAAI*, pp. 323–328.
- Parsia, B., Sirin, E., & Kalyanpur, A. (2005). Debugging OWL ontologies. In *WWW*, pp. 633–640.
- Peñaloza, R. (2009). *Axiom pinpointing in description logics and beyond*. Ph.D. thesis, Dresden University of Technology.
- Peñaloza, R., & Sertkaya, B. (2010). On the complexity of axiom pinpointing in the EL family of description logics. In *KR*.
- Peñaloza, R., & Thuluva, A. S. (2015). Iterative ontology updates using context labels. In Papini, O., Benferhat, S., Garcia, L., Mugnier, M., Fermé, E. L., Meyer, T., Wassermann, R., Hahmann, T., Baclawski, K., Krisnadhi, A., Klinov, P., Borgo, S., Kutz, O., & Porello, D. (Eds.), *Proceedings of the Joint Ontology Workshops 2015*, Vol. 1517 of *CEUR Workshop Proceedings*. CEUR-WS.org.
- Previti, A., & Marques-Silva, J. (2013). Partial MUS enumeration. In *AAAI*, pp. 818–825.
- Rector, A. L., & Horrocks, I. R. (1997). Experience building a large, re-usable medical ontology using a description logic with transitivity and concept inclusions. In *Workshop on Ontological Engineering*, pp. 414–418.
- Reiter, R. (1987). A theory of diagnosis from first principles. *Artif. Intell.*, 32(1), 57–95.
- Rosa, E. D., & Giunchiglia, E. (2013). Combining approaches for solving satisfiability problems with qualitative preferences. *AI Commun.*, 26(4), 395–408.
- Schlobach, S., & Cornet, R. (2003). Non-standard reasoning services for the debugging of description logic terminologies. In *IJCAI*, pp. 355–362.

- Schlobach, S., Huang, Z., Cornet, R., & van Harmelen, F. (2007). Debugging incoherent terminologies. *J. Autom. Reasoning*, 39(3), 317–349.
- Sebastiani, R., & Vescovi, M. (2009). Axiom pinpointing in lightweight description logics via Horn-SAT encoding and conflict analysis. In *CADE*, pp. 84–99.
- Sebastiani, R., & Vescovi, M. (2015). Axiom pinpointing in large  $\mathcal{EL}^+$  ontologies via SAT and SMT techniques. Tech. rep. DISI-15-010, DISI, University of Trento, Italy. Under Journal Submission. Available as [http://disi.unitn.it/~rseba/elsat/elsat\\_techrep.pdf](http://disi.unitn.it/~rseba/elsat/elsat_techrep.pdf).
- Sioutos, N., de Coronado, S., Haber, M. W., Hartel, F. W., Shaiu, W., & Wright, L. W. (2007). NCI thesaurus: A semantic model integrating cancer-related clinical and molecular information. *J. Biomed. Inform.*, 40(1), 30–43.
- Sirin, E., Parsia, B., Grau, B. C., Kalyanpur, A., & Katz, Y. (2007). Pellet: A practical OWL-DL reasoner. *J. Web Sem.*, 5(2), 51–53.
- Slaney, J. (2014). Set-theoretic duality: A fundamental feature of combinatorial optimisation. In *ECAI*, pp. 843–848.
- Spackman, K. A., Campbell, K. E., & Côté, R. A. (1997). SNOMED RT: a reference terminology for health care. In *AMIA*.
- Vescovi, M. (2011). *Exploiting SAT and SMT Techniques for Automated Reasoning and Ontology Manipulation in Description Logics*. Ph.D. thesis, University of Trento.

## Chapter 8

# Conclusions and Future Work

The Lightweight Description Logics stands out due to the availability of polynomial-time inference algorithms and its ability to represent knowledge from domains such as medicine, bioinformatics, geography, military and defense, among others. However, the construction of an ontology is an error-prone process which often leads to unintended inferences. The problem of axiom pinpointing explains these (unintended) conclusions and thus have a range of applications including ontology debugging and revision, context-based reasoning, error-tolerant reasoning and ontology matching, among others. In this thesis, we have exploited unsatisfiable boolean constraints to solve the problem of axiom pinpointing for the  $\mathcal{EL}$  family of DLs. we proposed state-of-the-art axiom pinpointing and debugging tools for the Lightweight Description Logic  $\mathcal{EL}^+$  that employ SAT solvers, MCS/MUS extractors and enumerators. Experimental results over well-known benchmarks from medical ontologies show categorical

## CHAPTER 8. CONCLUSIONS AND FUTURE WORK

performance gains of our methods over the existing axiom pinpointing methods and techniques for the  $\mathcal{EL}$  family of DLs. Since the  $\mathcal{EL}$  family of DLs is widely used for a range of applications our work provides a promising SAT-based solution. Future work will pursue a number of lines of research:

1. Besides axiom pinpointing in Lightweight Description Logics, HGMUS can be exploited by different applications (e.g., ontology matching [[JC11](#), [ES13](#)]) that suggested to use a dedicated Horn decision procedure [[DG84](#), [IM87](#), [Min88](#), [Scu90](#)].
2. Recently, a preference-based axiom pinpointing method [[MP14](#)] is proposed that sort MinAs using a partial order with respect to any given conclusion [[JQH09](#)]. We can extend this method either using EL2MCS or EL2MUS.
3. Context dependent granularity computation algorithms [[BKP12](#)] constrain an ontology according to an access control policy. The SAT-based axiom pinpointing methods, EL2MCS and EL2MUS, provides a good fit for solving this problem.
4. A SAT-based method [[WK13](#)] is proposed to validate PLTL (Propositional Linear Temporal Logic) specifications [[FKSV08](#), [RV10](#), [SD11](#)]. The proposed technique used explicit MUS enumeration [[LS08](#)] but we can extend it using an implicit MUS enumeration algorithm, HGMUS and some complex search procedure [[LH10](#)].

# Bibliography

- [ABB<sup>+</sup>00] Michael Ashburner, Catherine A. Ball, Judith A. Blake, David Botstein, Heather Butler, J. Michael Cherry, Allan P. Davis, Kara Dolinski, Selina S. Dwight, Janan T. Eppig, and et al. Gene ontology: tool for the unification of biology. *Nature genetics*, 25(1):25–29, 2000.
- [ALS08] Zaher S. Andraus, Mark H. Liffiton, and Karem A. Sakallah. Reveal: A formal verification tool for verilog designs. In *Logic for Programming, Artificial Intelligence, and Reasoning, 15th International Conference, LPAR 2008, Doha, Qatar, November 22–27, 2008. Proceedings*, volume 5330, pages 343–352, 2008.
- [AMI<sup>+</sup>16] M. Fareed Arif, Carlos Mencía, Alexey Ignatiev, Norbert Manthey, Rafael Peñaloza, and Joao Marques-Silva. BEACON: An Efficient SAT-Based Tool for Debugging  $\mathcal{EL}^+$  Ontologies. In *Theory and Applications of Satisfiability Testing – SAT 2016 – 19th International Conference, Bordeaux, France, July 5–8, 2016, Proceedings*, volume 9710, pages 521–530, 2016.

## BIBLIOGRAPHY

- [AMM<sup>+</sup>16] M. Fareed Arif, Carlos Mencía, Norbert Manthey, Alexey Ignatiev, Rafael Pe naloza, and Joao Marques-Silva. Towards Efficient SAT-Based Axiom Pinpointing in Lightweight Description Logics. *Journal of Artificial Intelligence Research (JAIR)*, 2016.  
Submitted - Under Review.
- [AMMS15a] M. Fareed Arif, Carlos Mencía, and Joao Marques-Silva. Efficient Axiom Pinpointing with EL2MCS. In *Proceedings of the 38th Annual German Conference on AI (KI 2015)*, volume 9324, pages 225–233. Springer, 2015.
- [AMMS15b] M. Fareed Arif, Carlos Mencía, and Joao Marques-Silva. Efficient MUS Enumeration of Horn Formulae with Applications to Axiom Pinpointing. In *Theory and Applications of Satisfiability Testing – SAT 2015 – 18th International Conference, Austin, TX, USA, September 24-27, 2015, Proceedings*, volume 9340, pages 324–342. Springer, 2015.
- [ANORC09] Roberto Asín, Robert Nieuwenhuis, Albert Oliveras, and Enric Rodríguez-Carbonell. Cardinality Networks and Their Applications. In *Theory and Applications of Satisfiability Testing – SAT 2009 – 12th International Conference, Swansea, UK, June 30 - July 3, 2009. Proceedings*, volume 5584, pages 167–180. Springer, 2009.

## BIBLIOGRAPHY

- [Bar08] G.V. Bard. Algorithms for the solution of polynomial and linear systems of equations over finite fields, with an application to the cryptanalysis of keeloq. Technical report, University of Maryland, 2008.
- [BBL05] Franz Baader, Sebastian Brandt, and Carsten Lutz. Pushing the  $\mathcal{EL}$  envelope. In *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence IJCAI-05*, pages 364–369, 2005.
- [BBL08] Franz Baader, Sebastian Brandt, and Carsten Lutz. Pushing the  $\mathcal{EL}$  Envelope Further. In *In Proceedings of the OWLED 2008 DC Workshop on OWL: Experiences and Directions*, volume 496. CEUR-WS.org, 2008.
- [BCC<sup>+</sup>99] Armin Biere, Alessandro Cimatti, Edmund M. Clarke, Masahiro Fujita, and Yunshan Zhu. Symbolic model checking using SAT procedures instead of BDDs. In *Proceedings of the 36th Annual ACM/IEEE Design Automation Conference*, DAC 99, pages 317–320, 1999.
- [BCCZ99] Armin Biere, Alessandro Cimatti, Edmund M. Clarke, and Yunshan Zhu. Symbolic model checking without BDDs. In *International conference on tools and algorithms for the construction and analysis of systems*, volume 1579 of *TACAS 99*, pages 193–207, 1999.

## BIBLIOGRAPHY

- [BCM<sup>+</sup>10] Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider. *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, New York, NY, USA, 2nd edition, 2010.
- [BDTK14] Fahiem Bacchus, Jessica Davies, Maria Tsimpoukelli, and George Katsirelos. Relaxation Search: A Simple Way of Managing Optional Clauses. In *Proceedings of the Twenty-Eighth Conference on Artificial Intelligence*, AAAI '14, pages 835–841, 2014.
- [Ber85] C. Berge. *Graphs and Hypergraphs*. Elsevier Science Ltd., Oxford, UK, UK, 1985.
- [BH95] Franz Baader and Bernhard Hollunder. Embedding Defaults into Terminological Knowledge Representation Formalisms. *Journal of Automated Reasoning*, 14(1):149–180, 1995.
- [BHvMW09] Armin Biere, Marijn Heule, Hans van Maaren, and Toby Walsh, editors. *Handbook of Satisfiability*, volume 185 of *Frontiers in Artificial Intelligence and Applications*. IOS Press, 2009.
- [BI95] Jan C. Bioch and Toshihide Ibaraki. Complexity of identification and dualization of positive boolean functions. *Information and Computation*, 123:50–63, 1995.
- [Bie08] Meghyn Bienvenu. Complexity of Abduction in the EL Family of Lightweight Description Logics. In Gerhard Brewka and Jérôme

## BIBLIOGRAPHY

- Lang, editors, *Proceedings of the Eleventh International Conference on Principles of Knowledge Representation and Reasoning (KR08)*, pages 220–230, 2008.
- [BIMM12] Yael Ben-Haim, Alexander Ivrii, Oded Margalit, and Arie Matsliah. Perfect hashing and CNF encodings of cardinality constraints. In *Theory and Applications of Satisfiability Testing - SAT 2012 - 15th International Conference, Trento, Italy, June 17-20, 2012. Proceedings*, pages 397–409, 2012.
- [BK15] Fahiem Bacchus and George Katsirelos. Using minimal correction sets to more efficiently compute minimal unsatisfiable sets. In *Computer Aided Verification - 27th International Conference, CAV 2015, San Francisco, CA, USA, July 18-24, 2015, Proceedings, Part II*, pages 70–86, 2015.
- [BKP12] Franz Baader, Martin Knechtel, and Rafael Peñaloza. Context-Dependent Views to Axioms and Consequences of Semantic Web Ontologies. *Journal of Web Semantics*, 12–13:22–40, 2012.
- [BL99] Hans K. Buning and T. Letterman. *Propositional Logic: Deduction and Algorithms*. Cambridge University Press, New York, NY, USA, 1999.
- [BL03] Elazar Birnbaum and Eliezer L. Lozinskii. Consistent subsets of inconsistent systems: structure and behaviour. *Journal of*

## BIBLIOGRAPHY

- Experimental and Theoretical Artificial Intelligence*, 15(1):25–46, 2003.
- [BLMS12] Anton Belov, Inês Lynce, and Joao Marques-Silva. Towards efficient MUS extraction. *AI Communications*, 25(2):97–116, 2012.
- [BLS06] Franz Baader, Carsten Lutz, and Boontawee Suntisrivaraporn. CEL - A polynomial-time reasoner for life science ontologies. In *Automated Reasoning, Third International Joint Conference, IJCAR 2006, Seattle, WA, USA, August 17-20, 2006, Proceedings*, pages 287–291, 2006.
- [BM07] Aaron R. Bradley and Zohar Manna. Checking Safety by Inductive Generalization of Counterexamples to Induction. In *Formal Methods in Computer-Aided Design, 7th International Conference, FMCAD 2007, Austin, Texas, USA, November 11-14, 2007, Proceedings*, pages 173–180. IEEE Computer Society, 2007.
- [BMR03] James Bailey, Thomas Manoukian, and Kotagiri Ramamohanarao. A Fast Algorithm for Computing Hypergraph Transversals and its Application in Mining Emerging Patterns. In *Proceedings of the 3rd IEEE International Conference on Data Mining (ICDM 2003), 19-22 December 2003, Melbourne, Florida, USA*, pages 485–488. IEEE Computer Society, 2003.
- [BMS11] Anton Belov and Joao Marques-Silva. Accelerating MUS extraction with recursive model rotation. In *International Conference*

## BIBLIOGRAPHY

- on Formal Methods in Computer-Aided Design, FMCAD '11,  
Austin, TX, USA, October 30 - November 02, 2011, pages 37–  
40, 2011.*
- [BMS12] Anton Belov and Joao Marques-Silva. MUSer2: An Efficient MUS Extractor. *Journal on Satisfiability, Boolean Modeling and Computation*, 8(3/4):123–128, 2012.
- [BP10a] Franz Baader and Rafael Peñaloza. Automata-based axiom pinpointing. *Journal of Automated Reasoning*, 45(2):91–129, August 2010. Special Issue: Selected Papers from IJCAR 2008.
- [BP10b] Franz Baader and Rafael Peñaloza. Axiom pinpointing in general tableaux. *Journal of Logic Computation*, 20(1):5–34, 2010.
- [BPS07] Franz Baader, Rafael Peñaloza, and Boontawee Suntisrivaraporn. Pinpointing in the description logic  $\mathcal{EL}^+$ . In *Advances in Artificial Intelligence (KI 2007), 30th Annual German Conference on AI, Osnabrück, Germany, September 10-13, 2007, Proceedings*, pages 52–67, 2007.
- [Bra04] Sebastian Brandt. Polynomial Time Reasoning in a Description Logic with Existential Restrictions, GCI Axioms, and—What Else? In R. López de Mantáras and L. Saitta, editors, *Proceedings of the 16th European Conference on Artificial Intelligence (ECAI-2004)*, pages 298–302. IOS Press, 2004.

## BIBLIOGRAPHY

- [BS00a] Franz Baader and Ulrike Sattler. Tableau algorithms for description logics. *Studia Logica*, 69:2001, 2000.
- [BS00b] Luís Baptista and João P. Marques Silva. Using randomization and learning to solve hard real-world instances of satisfiability. In *Principles and Practice of Constraint Programming - CP 2000, 6th International Conference, Singapore, September 18-21, 2000, Proceedings*, pages 489–494, 2000.
- [BS05] James Bailey and Peter J. Stuckey. Discovery of minimal unsatisfiable subsets of constraints using hitting set dualization. In *PADL*, pages 174–186, 2005.
- [BS08] Franz Baader and Boontawee Suntisrivaraporn. Debugging SNOMED CT using axiom pinpointing in the description logic  $\mathcal{EL}^+$ . In *Proceedings of the 3rd Knowledge Representation in Medicine (KR-MED'08): Representing and Sharing Knowledge Using SNOMED*, 2008.
- [CES<sup>+</sup>09] Koen Claessen, Niklas Eén, Mary Sheeran, Niklas Sörensson, Alexey Voronov, and Knut Åkesson. Sat-solving in practice, with a tutorial example from supervisory control. *Discrete Event Dynamic Systems*, 19(4):495–524, 2009.
- [Coo71] Stephen A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the 3rd Annual ACM Symposium on*

## BIBLIOGRAPHY

- Theory of Computing, May 3-5, 1971, Shaker Heights, Ohio, USA*, pages 151–158, 1971.
- [CP14] İsmail İlkan Ceylan and Rafael Peñaloza. Tight complexity bounds for reasoning in the description logic BEL. In Eduardo Fermé and João Leite, editors, *Proceedings of the 14th European Conference on Logics in Artificial Intelligence (JELIA 2014)*, volume 8761 of *Lecture Notes in Computer Science*, pages 77–91. Springer, 2014.
- [CT95] Zhi-Zhong Chen and Seinosuke Toda. The complexity of selecting maximal solutions. *Inf. Comput.*, 119:231–239, 1995.
- [CZI10] Michael Codish and Moshe Zazon-Ivry. Pairwise cardinality networks. In Edmund M. Clarke and Andrei Voronkov, editors, *LPAR (Dakar)*, volume 6355 of *Lecture Notes in Computer Science*, pages 154–172. Springer, 2010.
- [DB11] Jessica Davies and Fahiem Bacchus. Solving MAXSAT by solving a sequence of simpler SAT instances. In *Principles and Practice of Constraint Programming CP 2011 - 17th International Conference, CP 2011, Perugia, Italy, September 12-16, 2011. Proceedings*, volume 6876, pages 225–239, 2011.
- [DCDL<sup>+</sup>06] Roberto Di Cosmo, Berke Durak, Xavier Leroy, Fabio Mancinelli, and Jérôme Vouillon. Maintaining large software distributions:

## BIBLIOGRAPHY

- new challenges from the FOSS era. In , EASST Newsletter, volume 12, pages 7–20, Vienna, Austria, April 2006. EASST.
- [DG84] William F. Dowling and Jean H. Gallier. Linear-time algorithms for testing the satisfiability of propositional Horn formulae. *Journal of Logic Programming*, 1(3):267–284, 1984.
- [DK04] Vijay Durairaj and Priyank Kalla. Exploiting hypergraph partitioning for efficient boolean satisfiability. In *Ninth IEEE International High-Level Design Validation and Test Workshop 2004, Sonoma Valley, CA, USA, November 10-12, 2004*, pages 141–146, 2004.
- [dlBSW03] Maria J. García de la Banda, Peter J. Stuckey, and Jeremy Wazny. Finding all minimal unsatisfiable subsets. In *Proceedings of the 5th International ACM SIGPLAN Conference on Principles and Practice of Declarative Programming, 27-29 August 2003, Uppsala, Sweden*, pages 32–43, 2003.
- [DLL62] Martin Davis, George Logemann, and Donald W. Loveland. A machine program for theorem-proving. *Communications of the ACM*, 5(7):394–397, 1962.
- [DMP99] Carlos Domingo, Nina Mishra, and Leonard Pitt. Efficient read-restricted monotone CNF/DNF dualization by learning with membership queries. *Machine Learning*, 37(1):89–110, 1999.

## BIBLIOGRAPHY

- [DRPM06] S. Derriere, A. Richard, and A. Preite-Martinez. An ontology of astronomical object types for the virtual observatory. In *Proc. of the 26th meeting of the IAU: Virtual Observatory in Action: New Science, New Technology, and Next Generation Facilities*, pages 17–18, 2006.
- [EG95a] Thomas Eiter and Georg Gottlob. Identifying the minimal transversals of a hypergraph and related problems. *SIAM Journal on Computing*, 24:1278–1304, 1995.
- [EG95b] Thomas Eiter and Georg Gottlob. Identifying the minimal transversals of a hypergraph and related problems. *SIAM J. Comput.*, 24(6):1278–1304, 1995.
- [EG02] Thomas Eiter and Georg Gottlob. Hypergraph transversal computation and related problems in logic and ai. In *Logics in Artificial Intelligence, European Conference, JELIA 2002, Cosenza, Italy, September, 23-26, Proceedings*, volume 2424, pages 549–564, 2002.
- [EGM03a] Thomas Eiter, Georg Gottlob, and Kazuhisa Makino. New results on monotone dualization and generating hypergraph transversals. *SIAM J. Comput.*, 32(2):514–537, 2003.
- [EGM03b] Thomas Eiter, Georg Gottlob, and Kazuhisa Makino. New results on monotone dualization and generating hypergraph transversals. *SIAM J. Comput.*, pages 514–537, 2003.

## BIBLIOGRAPHY

- [ES03] Niklas Eén and Niklas Sörensson. An extensible SAT-solver. In *Theory and Applications of Satisfiability Testing – SAT 2003 – 6th International Conference, Santa Margherita Ligure, Italy, May 5-8, 2003 Selected Revised Papers*, volume 2919, pages 502–518, 2003.
- [ES06] Niklas Eén and Niklas Sörensson. Translating Pseudo-Boolean Constraints into SAT. *Journal on Satisfiability, Boolean Modeling and Computation*, 2(1-4):1–26, 2006.
- [ES13] Jérôme Euzenat and Pavel Shvaiko. *Ontology Matching, Second Edition*. Springer, 2013.
- [FHK<sup>+</sup>14] Fedor V. Fomin, Pinar Heggernes, Dieter Kratsch, Charis Papadopoulos, and Yngve Villanger. Enumerating minimal subset feedback vertex sets. *Algorithmica*, 69(1):216–231, 2014.
- [FK96] Michael Fredman and Leonid Khachiyan. On the complexity of dualization of monotone disjunctive normal forms. *J. Algorithms*, 21(3):618–628, 1996.
- [FKSV08] D. Fisman, O. Kupferman, S. Seinvald, and M.Y. Vardi. A framework for inherent vacuity. In *hvc08*, volume 5394, pages 7–22, 2008.
- [Gen13] Ian Gent. Optimal implementation of watched literals and more general techniques. *Journal of Artificial Intelligence Research*, 48:231–252, 2013.

## BIBLIOGRAPHY

- [GHM<sup>+</sup>08] Bernardo Cuenca Grau, Ian Horrocks, Boris Motik, Bijan Parsia, Peter Patel-Schneider, and Ulrike Sattler. OWL 2: The Next Step for OWL. *Web Semantics*, 6(4):309–322, 2008.
- [GLM14] Éric Grégoire, Jean-Marie Lagniez, and Bertrand Mazure. An experimentally efficient method for (MSS, CoMSS) partitioning. In *Proceedings of the Twenty-Eighth Conference on Artificial Intelligence*, AAAI ’14, pages 2666–2673, 2014.
- [GM06] Enrico Giunchiglia and Marco Maratea. Solving optimization problems with DLL. In *ECAI*, pages 377–381, 2006.
- [GMP07] Éric Grégoire, Bertrand Mazure, and Cédric Piette. Boosting a complete technique to find MSS and MUS thanks to a local search oracle. In *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6-12, 2007*, pages 2300–2305, 2007.
- [GN07] Eugene Goldberg and Yakov Novikov. Berkmin: A fast and robust sat-solver. *Discrete Applied Mathematics*, 155(12):1549–1561, 2007.
- [Goo05] John Goodwin. Experiences of using OWL at the ordnance survey. In *Proceedings of the OWLED\*05 Workshop on OWL: Experiences and Directions, Galway, Ireland, November 11-12, 2005*, volume 188, 2005.

## BIBLIOGRAPHY

- [GSK98] Carla P. Gomes, Bart Selman, and Henry A. Kautz. Boosting combinatorial search through randomization. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence and Tenth Innovative Applications of Artificial Intelligence Conference, AAAI 98, IAAI 98, July 26-30, 1998, Madison, Wisconsin, USA.*, pages 431–437, 1998.
- [Gup06] Anubhav Gupta. *Learning Abstractions for Model Checking*. PhD thesis, Carnegie Mellon University, June 2006.
- [GZB06a] Christine Golbreich, Songmao Zhang, and Olivier Bodenreider. The foundational model of anatomy in owl: Experience and perspectives. *Web Semantics: Science, Services and Agents on the World Wide Web*, pages 181–195, 2006.
- [GZB06b] Christine Golbreich, Songmao Zhang, and Olivier Bodenreider. The Foundational Model of Anatomy in OWL: Experience and Perspectives. *Web Semantics: Science, Services and Agents on the World Wide Web*, 4(3), 2006.
- [HBC07] Céline Hébert, Alain Bretto, and Bruno Crémilleux. A data mining formalization to improve hypergraph minimal transversal computation. *Fundamenta Informaticae*, 80(4):415–433, 2007.
- [HK10] Anthony Hunter and Sébastien Konieczny. On the measure of conflicts: Shapley inconsistency values. *Artif. Intell.*, 174(14):1007–1026, 2010.

## BIBLIOGRAPHY

- [HMM15] Federico Heras, António Morgado, and Joao Marques-Silva. MaxSAT-based encodings for group MaxSAT. *AI Communication*, 28(2):195–214, 2015.
- [HPS09] Matthew Horridge, Bijan Parsia, and Ulrike Sattler. Explaining inconsistencies in owl ontologies. In Lluis Godo and Andrea Editors Pugliese, editors, *Scalable Uncertainty Management*, volume 5785, pages 124–137. Springer, 2009.
- [IJM13] Alexey Ignatiev, Mikolás Janota, and João Marques-Silva. Quantified Maximum Satisfiability: A Core-Guided Approach. In *16th International Conference on Theory and Applications of Satisfiability Testing – SAT 2013 – July 8-12, 2013, Helsinki, Finland*, pages 250–266, 2013.
- [IM87] Alon Itai and Johann A. Makowsky. Unification as a complexity measure for logic programming. *Journal of Logic Programming*, 4(2):105–117, 1987.
- [IPLM15] Alexey Ignatiev, Alessandro Previti, Mark H. Liffiton, and Joao Marques-Silva. Smallest MUS extraction with minimal hitting set dualization. In *Principles and Practice of Constraint Programming - 21st International Conference, CP 2015, Cork, Ireland, August 31 - September 4, 2015, Proceedings*, pages 173–182, 2015.

## BIBLIOGRAPHY

- [JC11] Ernesto Jiménez-Ruiz and Bernardo Cuenca Grau. LogMap: Logic-based and scalable ontology matching. In *The Semantic Web - ISWC 2011 - 10th International Semantic Web Conference, Bonn, Germany, October 23-27, 2011, Proceedings, Part I*, pages 273–288, 2011.
- [JQH09] Qiu Ji, Guilin Qi, and Peter Haase. A relevance-directed algorithm for finding justifications of DL entailments. In *The Semantic Web, Fourth Asian Conference, ASWC 2009, Shanghai, China, December 6-9, 2009. Proceedings*, pages 306–320, 2009.
- [JT96] David J. Johnson and Michael A. Trick, editors. *Cliques, Coloring, and Satisfiability: Second DIMACS Implementation Challenge, Workshop, October 11-13, 1993*. American Mathematical Society, Boston, MA, USA, 1996.
- [KBB<sup>+</sup>08] Leonid Khachiyan, Endre Boros, Konrad Borys, Khaled M. Elbassioni, and Vladimir Gurvich. Generating all vertices of a polyhedron is hard. *Discrete & Computational Geometry*, 39(1-3):174–190, 2008.
- [KBEG08] Leonid Khachiyan, Endre Boros, Khaled M. Elbassioni, and Vladimir Gurvich. On enumerating minimal dicuts and strongly connected subgraphs. *Algorithmica*, 50(1):159–172, 2008.
- [KLW12] Boris Konev, Michel Ludwig, Dirk Walther 0002, and Frank Wolter. The Logical Difference for the Lightweight Description

## BIBLIOGRAPHY

- Logic EL. *Journal of Artificial Intelligence Research (JAIR)*, 44:633–708, 2012.
- [KPHS07] Aditya Kalyanpur, Bijan Parsia, Matthew Horridge, and Evren Sirin. Finding all justifications of OWL DL entailments. In *The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007, Busan, Korea, November 11-15, 2007*, pages 267–280, 2007.
- [KPSG06] Aditya Kalyanpur, Bijan Parsia, Evren Sirin, and Bernardo Cuenca Grau. Repairing Unsatisfiable Concepts in OWL Ontologies. In *The Semantic Web: Research and Applications, 3rd European Semantic Web Conference, ESWC 2006, Budva, Montenegro, June 11-14, 2006, Proceedings*, pages 170–184, 2006.
- [KS92] Henry A. Kautz and Bart Selman. Planning as satisfiability. In *Foundation of Knowledge Representation and Reasoning, ECAI-92*, pages 359–363, 1992.
- [KS03] Dimitris J. Kavvadias and Elias C. Stavropoulos. Monotone Boolean dualization is in co-NP[ $\log^2 n$ ]. *Information Processing Letters*, pages 1–6, 2003.
- [LAF<sup>+</sup>05] Lee Lacy, Gabriel Aviles, Karen Fraser, William Gerber, Alice M. Mulvehill, and Robert Gaskill. Experiences using OWL in military applications. In *Proceedings of the OWLED 05 Workshop*

## BIBLIOGRAPHY

- on OWL: Experiences and Directions, Galway, Ireland, November 11-12, 2005.*
- [Lev73] Leonid A Levin. Universal sequential search problems. *Problemy Peredachi Informatsii*, 9(3):115–116, 1973.
- [LH10] Michel Ludwig and Ullrich Hustadt. Implementing a fair monodic temporal logic prover. *AI Communications*, 23(2–3):69–96, 2010.
- [Lib05] Paolo Liberatore. Redundancy in logic I: CNF propositional formulae. *Artif. Intell.*, 163(2):203–232, 2005.
- [LLK80] Eugene L. Lawler, Jan Karel Lenstra, and A. H. G. Rinnooy Kan. Generating all maximal independent sets: Np-hardness and polynomial-time algorithms. *SIAM J. Comput.*, 9(3):558–565, 1980.
- [LM13] Mark H. Liffiton and Ammar Malik. Enumerating infeasibility: Finding multiple MUSes quickly. In *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems, 10th International Conference, CPAIOR 2013, Yorktown Heights, NY, USA, May 18-22, 2013. Proceedings*, pages 160–175, 2013.
- [LML<sup>+</sup>09] Mark H. Liffiton, Maher N. Mneimneh, Inês Lynce, Zaher S. Andraus, João Marques-Silva, and Karem A. Sakallah. A branch

## BIBLIOGRAPHY

- and bound algorithm for extracting smallest minimal unsatisfiable subformulas. *Constraints*, 14(4):415–442, 2009.
- [LMPS05] Mark H. Liffiton, Michael D. Moffitt, Martha E. Pollack, and Karem A. Sakallah. Identifying conflicts in overconstrained temporal problems. In *IJCAI-05, Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence, Edinburgh, Scotland, UK, July 30-August 5, 2005*, pages 205–211, 2005.
- [LMS04] Inês Lynce and Joao Marques-Silva. On computing minimum unsatisfiable cores. In *SAT 2004 - The Seventh International Conference on Theory and Applications of Satisfiability Testing, 10-13 May 2004, Vancouver, BC, Canada, Online Proceedings*, 2004.
- [LMS06a] Luciano Lavagno, Grant Martin, and Louis Scheffer. *Electronic Design Automation for Integrated Circuits Handbook - 2 Volume Set*. CRC Press, Inc., Boca Raton, FL, USA, 2006.
- [LMS06b] Inês Lynce and Joao Marques-Silva. Efficient haplotype inference with boolean satisfiability. In *Proceedings, The Twenty-First National Conference on Artificial Intelligence and the Eighteenth Innovative Applications of Artificial Intelligence Conference, July 16-20, 2006, Boston, Massachusetts, USA*, pages 104–109, 2006.

## BIBLIOGRAPHY

- [LNO06a] S. K. Lahiri, R. Nieuwenhuis, and A. Oliveras. SMT Techniques for Fast Predicate Abstraction. In T. Ball and R. B. Jones, editors, *18th International Conference on Computer Aided Verification, CAV'06*, volume 4144 of *Lecture Notes in Computer Science*, pages 413–426. Springer, 2006.
- [LNO06b] Shuvendu K. Lahiri, Robert Nieuwenhuis, and Albert Oliveras. SMT techniques for fast predicate abstraction. In *Computer Aided Verification, 18th International Conference, CAV 2006, Seattle, WA, USA, August 17-20, 2006, Proceedings*, pages 424–437, 2006.
- [LP14a] Michel Ludwig and Rafael Peñaloza. Brave and cautious reasoning in el. In *Proceedings of the 27th International Workshop on Description Logics (DL'14)*, volume 1193, pages 274–286, 2014.
- [LP14b] Michel Ludwig and Rafael Peñaloza. Error-tolerant reasoning in the description logic  $\mathcal{EL}$ . In *Logics in Artificial Intelligence - 14th European Conference, JELIA 2014, Funchal, Madeira, Portugal, September 24-26, 2014. Proceedings*, pages 107–121, 2014.
- [LPMMS15] Mark H Liffiton, Alessandro Previti, Ammar Malik, and Joao Marques-Silva. Fast, flexible MUS enumeration. *Constraints*, pages 1–28, 2015.
- [LS04] Inês Lynce and João P. Marques Silva. On computing minimum unsatisfiable cores. In *SAT 2004 - The Seventh International*

## BIBLIOGRAPHY

- Conference on Theory and Applications of Satisfiability Testing, 10-13 May 2004, Vancouver, BC, Canada, Online Proceedings*, 2004.
- [LS05] Mark H. Liffiton and Karem A. Sakallah. On finding all minimally unsatisfiable subformulas. In *Theory and Applications of Satisfiability Testing, 8th International Conference, SAT 2005, St. Andrews, UK, June 19-23, 2005, Proceedings*, volume 3569, pages 173–186, 2005.
- [LS08] Mark H. Liffiton and Karem A. Sakallah. Algorithms for computing minimal unsatisfiable subsets of constraints. *J. Autom. Reasoning*, 40(1):1–33, 2008.
- [Lud14] Michel Ludwig. Just: a tool for computing justifications w.r.t. ELH ontologies. In *ORE*, 2014.
- [MBB<sup>+</sup>03] Pedro Meseguer, Noureddine Bouhmala, Taoufik Bouzoubaa, Morten Irgens, and Martí Sánchez. Current approaches for solving over-constrained problems. *Constraints*, 8:9–39, 2003.
- [McM03] Kenneth L. McMillan. Interpolation and sat-based model checking. In *Computer Aided Verification, 15th International Conference, CAV 2003, Boulder, CO, USA, July 8-12, 2003, Proceedings*, pages 1–13, 2003.
- [MHJ<sup>+</sup>13] Joao Marques-Silva, Federico Heras, Mikoláš Janota, Alessandro Previti, and Anton Belov. On Computing Minimal Correction

## BIBLIOGRAPHY

- Subsets. In *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*, 2013.
- [Min88] Michel Minoux. LTUR: A simplified linear-time unit resolution algorithm for Horn formulae and computer implementation. *Inf. Process. Lett.*, 29(1):1–12, 1988.
- [MJIM15] João Marques-Silva, Mikolás Janota, Alexey Ignatiev, and António Morgado. Efficient model based diagnosis with maximum satisfiability. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, pages 1966–1972, 2015.
- [MLBP06] Thomas Andreas Meyer, Kevin Lee, Richard Booth, and Jeff Z. Pan. Finding maximally satisfiable terminologies for the description logic  $\mathcal{ALC}$ . In *AAAI*, pages 269–274, 2006.
- [MLM12] António Morgado, Mark H. Liffiton, and Joao Marques-Silva. MaxSAT-based MCS enumeration. In *Hardware and Software: Verification and Testing - 8th International Haifa Verification Conference, HVC 2012, Haifa, Israel, November 6-8, 2012. Revised Selected Papers*, pages 86–101, 2012.
- [MMV11] Kodylan Moodley, Thomas Meyer, and Ivan José Varzinczak. Root justifications for ontology repair. In *Web Reasoning and*

## BIBLIOGRAPHY

*Rule Systems - 5th International Conference, RR 2011, Galway, Ireland, August 29-30, 2011. Proceedings*, pages 275–280, 2011.

- [MMZ<sup>+</sup>01] Matthew W. Moskewicz, Conor F. Madigan, Ying Zhao, Lintao Zhang, and Sharad Malik. Chaff: Engineering an efficient SAT solver. In *Proceedings of the 38th Design Automation Conference, DAC 2001, Las Vegas, NV, USA, June 18-22, 2001*, pages 530–535, 2001.
- [MP14] João Marques-Silva and Alessandro Previti. On computing preferred muses and mcses. In *Theory and Applications of Satisfiability Testing - SAT 2014 - 17th International Conference, Held as Part of the Vienna Summer of Logic, VSL 2014, Vienna, Austria, July 14-17, 2014. Proceedings*, pages 58–74, 2014.
- [MP15] Norbert Manthey and Rafael Peñaloza. Exploiting SAT technology for axiom pinpointing. Technical Report LTCS 15-05, Chair of Automata Theory, Institute of Theoretical Computer Science, Technische Universität Dresden, April 2015. Available from <https://dd11.inf.tu-dresden.de/web/Techreport3010>.
- [MPM15] Carlos Mencía, Alessandro Previti, and João Marques-Silva. Literal-based MCS extraction. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, pages 1973–1979, 2015.

## BIBLIOGRAPHY

- [MPR16] Norbert Manthey, Rafael Peñaloza, and Sebastian Rudolph. Efficient axiom pinpointing in EL using SAT technology. In *Proceedings of the 29th International Workshop on Description Logics, Cape Town, South Africa, April 22-25, 2016.*, 2016.
- [Ms08] Joao Marques-silva. Practical applications of boolean satisfiability. In *In Workshop on Discrete Event Systems (WODES*. IEEE Press, 2008.
- [MSJ13] Joao Marques-Silva and Mikolás Janota. Minimal sets over monotone predicates in Boolean formulae. In *Computer Aided Verification - 25th International Conference, CAV 2013, Saint Petersburg, Russia, July 13-19, 2013. Proceedings*, pages 592–607, 2013.
- [MSL11] Joao P. Marques-Silva and Inês Lynce. On improving MUS extraction algorithms. In *Theory and Applications of Satisfiability Testing - SAT 2011 - 14th International Conference, SAT 2011, Ann Arbor, MI, USA, June 19-22, 2011. Proceedings*, volume 6695, pages 159–173, 2011.
- [MSLM09] Joao Marques-Silva, Ines Lynce, and Sharad Malik. Conflict-driven clause learning sat solvers. In Armin Biere, Marijn Heule, Hans van Maaren, and Toby Walsh, editors, *Handbook of Satisfiability*, chapter 04, pages 127–149. IOS Press, 2009.

## BIBLIOGRAPHY

- [MSS96] Joao P. Marques-Silva and Karem A. Sakallah. GRASP - a new search algorithm for satisfiability. In *Proceedings of the ACM/IEEE International Conference on CAD-96 (ICCAD), San Jose, CA, November 1996*, pages 220–227, 1996.
- [MSS99] Joao P. Marques-Silva and Karem A. Sakallah. GRASP: A search algorithm for propositional satisfiability. *IEEE Trans. Computers*, 48(5):506–521, 1999.
- [MU11] Keisuke Murakami and Takeaki Uno. Efficient algorithms for dualizing large-scale hypergraphs. *CoRR*, abs/1102.3813, 2011.
- [NGT04] Dana Nau, Malik Ghallab, and Paolo Traverso. *Automated Planning: Theory & Practice*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2004.
- [OPFP07] Barry O’Sullivan, Alexandre Papadopoulos, Boi Faltings, and Pearl Pu. Representative explanations for over-constrained problems. In *AAAI*, pages 323–328, 2007.
- [Peñ15] Rafael Peñaloza. Reasoning with annotated description logic ontologies, September 2015. Habilitation Thesis.
- [PG86] David A. Plaisted and Steven Greenbaum. A structure-preserving clause form translation. *J. Symb. Comput.*, 2(3):293–304, September 1986.

## BIBLIOGRAPHY

- [PM13] Alessandro Previti and Joao Marques-Silva. Partial MUS enumeration. In *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence, July 14-18, 2013, Bellevue, Washington, USA.*, 2013.
- [Pre09] Steven Prestwich. *Handbook of Satisfiability*, volume 185 of *Frontiers in Artificial Intelligence and Applications*, chapter CNF Encodings. IOS Press, 2009.
- [PS09] Rafael Peñaloza and Barış Sertkaya. Axiom pinpointing is hard. In Bernardo Cuenca Grau, Ian Horrocks, Boris Motik, and Ulrike Sattler, editors, *Proceedings of the 2009 International Workshop on Description Logics (DL2009)*, volume 477 of *CEUR-WS*, 2009.
- [PS10a] Rafael Peñaloza and Barış Sertkaya. Complexity of axiom pinpointing in the dl-lite family of description logics. In Helder Coelho, Rudi Studer, and Michael Wooldridge, editors, *Proceedings of the 19th European Conference on Artificial Intelligence (ECAI 2010)*, volume 215 of *Frontiers in Artificial Intelligence and Applications*, pages 29–34. IOS Press, 2010.
- [PS10b] Rafael Peñaloza and Barış Sertkaya. On the complexity of axiom pinpointing in the el family of description logics. In Fangzhen Lin, Ulrike Sattler, and Mirosław Truszczyński, editors, *Proceedings of the Twelfth International Conference on Principles*

## BIBLIOGRAPHY

- of Knowledge Representation and Reasoning (KR 2010)*. AAAI Press, 2010.
- [PSHH04] Peter F. Patel-Schneider, Patrick Hayes, and Ian Horrocks. OWL Web Ontology Language - Semantics and Abstract Syntax. W3C Recommendation, W3C, 2004.
- [PSK05] Bijan Parsia, Evren Sirin, and Aditya Kalyanpur. Debugging owl ontologies. In *Proceedings of the 14th international conference on World Wide Web*, pages 633–640. ACM, 2005.
- [Rei87] Raymond Reiter. A theory of diagnosis from first principles. *Artif. Intell.*, 32(1):57–95, 1987.
- [RG13] Emanuele Di Rosa and Enrico Giunchiglia. Combining approaches for solving satisfiability problems with qualitative preferences. *AI Communications*, 26(4):395–408, 2013.
- [RH97] Alan L. Rector and Ian R. Horrocks. Experience building a large, re-usable medical ontology using a description logic with transitivity and concept inclusions. In *Workshop on Ontological Engineering*, pages 414–418, 1997.
- [RM03] Cornelius Rosse and José L. V. Mejino, Jr. A reference ontology for biomedical informatics: The foundational model of anatomy. *J. of Biomedical Informatics*, 36:478–500, 2003.

## BIBLIOGRAPHY

- [RV10] K.Y. Rozier and M.Y. Vardi. LTL satisfiability checking. *International Journal on Software Tools for Technology Transfer (STTT)*, pages 123–137, 2010.
- [SAR<sup>+</sup>07] Barry Smith, Michael Ashburner, Cornelius Rosse, Jonathan Bard, William Bug, Werner Ceusters, Louis J Goldberg, Karen Eilbeck, Amelia Ireland, Christopher J Mungall, Neocles Leontis, Philippe Rocca-Serra, Alan Ruttenberg, Susanna-Assunta Sansone, Richard H Scheuermann, Nigam Shah, Patricia L Whetzel, and Suzanna Lewis. The obo foundry: coordinated evolution of ontologies to support biomedical data integration. *Nature Biotechnology*, 25:1251–1255, 2007.
- [SC03] Stefan Schlobach and Ronald Cornet. Non-standard reasoning services for the debugging of description logic terminologies. In *IJCAI*, pages 355–362, 2003.
- [SCC97] Kent A. Spackman, Keith E. Campbell, and Roger A. Côté. SNOMED RT: a reference terminology for health care. In *AMIA*, 1997.
- [SCS11] Stefan Schulz, Ronald Cornet, and Kent A. Spackman. Consolidating snomed ct’s ontological commitment. *Applied Ontology*, 6(1):1–11, 2011.

## BIBLIOGRAPHY

- [Scu90] Maria Grazia Scutellà. A note on Dowling and Gallier’s top-down algorithm for propositional Horn satisfiability. *J. Log. Program.*, 8(3):265–273, 1990.
- [SD11] Viktor Schuppan and Luthfi Darmawan. Evaluating LTL satisfiability solvers. In *Automated Technology for Verification and Analysis, 9th International Symposium, ATVA 2011, Taipei, Taiwan, October 11-14, 2011. Proceedings*, pages 397–413, 2011.
- [SdCH<sup>+</sup>07] Nicholas Sioutos, Sherri de Coronado, Margaret W. Haber, Frank W. Hartel, Wen-Ling Shaiu, and Lawrence W. Wright. NCI thesaurus: A semantic model integrating cancer-related clinical and molecular information. *Journal of Biomedical Informatics*, 40(1):30–43, 2007.
- [Ser09] Baris Sertkaya. Some computational problems related to pseudo-intents. In Sébastien Ferré and Sebastian Rudolph, editors, *Formal Concept Analysis, 7th International Conference, ICFCA 2009, Darmstadt, Germany, May 21-24, 2009, Proceedings*, volume 5548 of *Lecture Notes in Computer Science*, pages 130–145. Springer, 2009.
- [SHCvH07] Stefan Schlobach, Zhisheng Huang, Ronald Cornet, and Frank van Harmelen. Debugging incoherent terminologies. *Journal of Automated Reasoning*, 39(3):317–349, 2007.

## BIBLIOGRAPHY

- [Sin05] Carsten Sinz. Towards an optimal cnf encoding of boolean cardinality constraints. In Peter van Beek, editor, *CP*, volume 3709, pages 827–831. Springer, 2005.
- [SLL<sup>+</sup>04] D. Soergel, B. Lauser, A. Liang, F. Fisseha, J. Keizer, and S. Katz. Reengineering thesauri for new applications: The agrovoc example. *Journal of Digital Information*, 4, 2004.
- [Sma07] Paul R Smart. Rule-based intelligence on the semantic web: Implications for military capabilities. Technical report, University of Southampton, 2007.
- [SSS91] Manfred Schmidt-Schaub&and Gert Smolka. Attributive concept descriptions with complements. *Artificial Intelligence*, 48(1):1–26, 1991.
- [Sun13] B. Suntisrivaraporn. Finding all justifications in snomed ct. *ScienceAsia*, 39(1):79–90, 2013.
- [SV09] Roberto Sebastiani and Michele Vescovi. Axiom pinpointing in lightweight description logics via Horn-SAT encoding and conflict analysis. In *Automated Deduction - CADE-22, 22nd International Conference on Automated Deduction, Montreal, Canada, August 2-7, 2009. Proceedings*, pages 84–99, 2009.
- [SV15] Roberto Sebastiani and Michele Vescovi. Axiom pinpointing in lightweight description logics via Horn-SAT encoding and

## BIBLIOGRAPHY

- conflict analysis. Technical Report DISI-09-014, University of Trento, 2015.
- [TCJ08] Emina Torlak, Felix Sheng-Ho Chang, and Daniel Jackson. Finding minimal unsatisfiable cores of declarative specifications. In *FM 2008: Formal Methods, 15th International Symposium on Formal Methods, Turku, Finland, May 26-30, 2008, Proceedings*, volume 5014, pages 326–341, 2008.
- [Tse83] G. S. Tseitin. On the complexity of derivation in propositional calculus. In J. Siekmann and G. Wrightson, editors, *Automation of Reasoning 2: Classical Papers on Computational Logic 1967-1970*, pages 466–483, Berlin, Heidelberg, 1983. Springer.
- [TVD10] Emina Torlak, Mandana Vaziri, and Julian Dolby. MemSAT: checking axiomatic specifications of memory models. In *Proceedings of the 2010 ACM SIGPLAN Conference on Programming Language Design and Implementation, PLDI 2010, Toronto, Ontario, Canada, June 5-10, 2010*, pages 341–350, 2010.
- [Vel05] Miroslav N. Velev. Comparison of schemes for encoding unobservability in translation to sat. In *ASP-DAC*, pages 1056–1059. ACM Press, 2005.
- [WK13] Richard Williams and Boris Konev. Propositional Temporal Proving with Reductions to a SAT Problem. In *Automated Deduction - CADE-24 - 24th International Conference on Auto-*

## BIBLIOGRAPHY

- mated Deduction, Lake Placid, NY, USA, June 9-14, 2013. Proceedings*, pages 421–435, 2013.
- [Wol09] Michael M Wolf. *Hypergraph-Based Combinatorial Optimization of Matrix-Vector Multiplication*. PhD thesis, University of Illinois at Urbana-Champaign, 2009.
- [XM12] Guohui Xiao and Yue Ma. Inconsistency measurement based on variables in minimal unsatisfiable subsets. In *ECAI 2012 - 20th European Conference on Artificial Intelligence, System Demonstrations Track, Montpellier, France, August 27-31 , 2012*, volume 242, pages 864–869, 2012.
- [ZQS13] Zhangquan Zhou, Guilin Qi, and Boontawee Suntisrivaraporn. A new method of finding all justifications in OWL 2 EL. In *2013 IEEE/WIC/ACM International Conferences on Web Intelligence, WI 2013, Atlanta, GA, USA, November 17-20, 2013*, pages 213–220, 2013.