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

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

[1] University College Dublin, Dublin, Ireland
muhammad.arif.1@ucdconnect.ie
[2] University of Oviedo, Oviedo, Spain
cmencia@gmail.com
[3] University of Lisbon, Lisbon, Portugal
{aignatiev,jpms}@ciencias.ulisboa.pt
[4] TU Dresden, Dresden, Germany
norbert.manthey@tu-dresden.de
[5] Free University of Bozen-Bolzano, Bolzano, Italy
rafael.penaloza@unibz.it
[6] 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.

N. Creignou and D. Le Berre (Eds.): SAT 2016, LNCS 9710, pp. 521–530, 2016.
DOI: 10.1007/978-3-319-40970-2_32

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 $\mathsf{N_C}$ and $\mathsf{N_R}$ of *concept names* and *role names* through the grammar rule $C ::= A \mid \top \mid C \sqcap C \mid \exists r.C$, where $A \in \mathsf{N_C}$ and $r \in \mathsf{N_R}$. 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) $r_1 \circ \cdots \circ r_n \sqsubseteq s$, where $n \geq 1$ and $r_i, s \in \mathsf{N_R}$. We will often use the term *axiom* to refer to both GCIs and RIs. As an example, $\texttt{Appendix} \sqsubseteq \exists\texttt{partOf.Intestine}$ 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 $A \in \mathsf{N_C}$ to a set $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ and every $r \in \mathsf{N_R}$ 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 $r_1 \circ \cdots \circ r_n \sqsubseteq s$ iff $r_1^{\mathcal{I}} \circ \cdots \circ r_n^{\mathcal{I}} \subseteq s^{\mathcal{I}}$, 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].

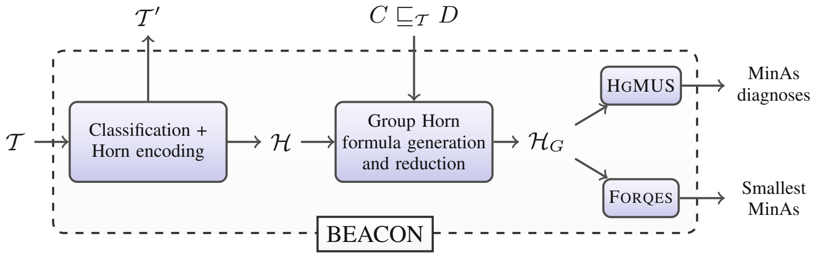
$\mathcal{T}'$
$C \sqsubseteq_{\mathcal{T}} D$
$\mathcal{T}$
Classification + Horn encoding
$\mathcal{H}$
Group Horn formula generation and reduction
$\mathcal{H}_G$
HGMUS
FORQES
MinAs diagnoses
Smallest MinAs
BEACON

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 μ: $X \to \{0, 1\}$. If μ satisfies $\mathcal{F}$, μ 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 $\mathcal{F} = \mathcal{G}_0 \cup \ldots \cup \mathcal{G}_k$, a* group-MUS *of $\mathcal{F}$ is a set of groups $\mathcal{G} \subseteq \{\mathcal{G}_1, \ldots, \mathcal{G}_k\}$, 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$^+$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) $(A_1 \sqcap ... \sqcap A_k) \sqsubseteq B$ $(k \geq 1)$, (ii) $A \sqsubseteq \exists r.B$, (iii) $\exists r.A \sqsubseteq B$, or (iv) $r_1 \circ ... \circ r_n \sqsubseteq s$ $(n \geq 1)$, where $A, A_i, B \in \mathsf{N_C}$ and $r, r_i, s \in \mathsf{N_R}$. This process results in a TBox $\mathcal{T}_\mathcal{N}$ where each axiom $a_i \in \mathcal{T}$ is substituted by a set of axioms in normal form $\{a_{i1}, ..., a_{im_i}\}$. 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, $\mathcal{T}_\mathcal{N}$ 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 $\mathsf{ant}(r)$) leading to inferring an axiom a_i, the Horn clause $(\bigwedge_{\{a_j \in \mathsf{ant}(r)\}} s_{[a_j]}) \rightarrow s_{[a_i]}$ 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 $X, Y \in \mathsf{N_C}$.

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: $\mathcal{H}_G = \{\mathcal{G}_0, \mathcal{G}_1, ..., \mathcal{G}_{|\mathcal{T}|}\}$, 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
$A \sqsubseteq A_i, 1 \leq i \leq n$	$A_1 \sqcap .... \sqcap A_n \sqsubseteq B$	$A \sqsubseteq B$
$A \sqsubseteq A_1$	$A_1 \sqsubseteq \exists r.B$	$A \sqsubseteq \exists r.B$
$A \sqsubseteq \exists r.B$, $B \sqsubseteq B_1$	$\exists r.B_1 \sqsubseteq B_2$	$A \sqsubseteq B_2$
$A_{i-1} \sqsubseteq \exists r_i.A_i, 1 \leq i \leq n$	$r_1 \circ ... \circ r_n \sqsubseteq r$	$A_0 \sqsubseteq \exists r.A_n$

Algorithm 1. eMUS [26] / MARCO [18]

```
Input: $\mathcal{F}$ a CNF formula
Output: Reports the set of MUSes (and MCSes) of $\mathcal{F}$
$\langle I, \mathcal{Q}\rangle \leftarrow \langle\{p_i \mid c_i \in \mathcal{F}\}, \emptyset\rangle$        // Variable $p_i$ picks clause $c_i$
while true do
    $(st, P) \leftarrow$ MaximalModel($\mathcal{Q}$)
    if not $st$ then return
    $\mathcal{F}' \leftarrow \{c_i \mid p_i \in P\}$        // Pick selected clauses
    if not SAT($\mathcal{F}'$) then
        $\mathcal{M} \leftarrow$ ComputeMUS($\mathcal{F}'$)
        ReportMUS($\mathcal{M}$)
        $b \leftarrow \{\neg p_i \mid c_i \in \mathcal{M}\}$        // Negative clause blocking the MUS
    else
        ReportMCS($\mathcal{F} \setminus \mathcal{F}'$)
        $b \leftarrow \{p_i \mid p_i \in I \setminus P\}$        // Positive clause blocking the MCS
    $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{b\}$
```

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 Σ_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[1] 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 (2 Ghz) with a limit of 3600 s 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$^+$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.

[1] Available at http://logos.ucd.ie/web/doku.php?id=beacon-tool.

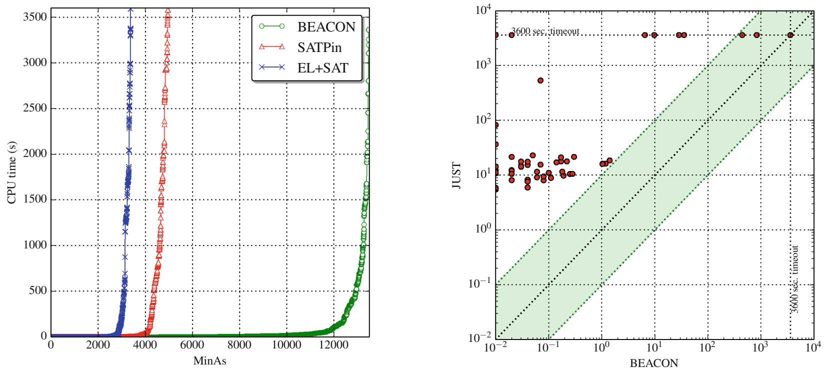

Fig. 2. Plots comparing BEACON to EL$^+$SAT, SATPin and JUST

Axiom Pinpointing. BEACON shows significant improvements over the existing tools EL$^+$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
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
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://ddll.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. Peñaoza, 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
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)