HAL open science

Semantic Relevance

Fajar Haifani, Christoph Weidenbach

▶ To cite this version:

Fajar Haifani, Christoph Weidenbach. Semantic Relevance. IJCAR, International Joint Conference in Automated Reasoning, Aug 2022, Haifa, Israel. pp.208-227, 10.1007/978-3-031-10769-6_13 . hal-03881904

HAL Id: hal-03881904

https://inria.hal.science/hal-03881904v1

Submitted on 2 Dec 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Semantic Relevance

Fajar Haifani[1,2] and Christoph Weidenbach[1]

[1] Max Planck Institute for Informatics, Saarland Informatics Campus, Saarbrücken Germany
[2] Graduate School of Computer Science, Saarbrücken, Germany

Abstract. A clause C is syntactically relevant in some clause set N, if it occurs in every refutation of N. A clause C is syntactically semi-relevant, if it occurs in some refutation of N. While syntactic relevance coincides with satisfiability (if C is syntactically relevant then $N \setminus \{C\}$ is satisfiable), the semantic counterpart for syntactic semi-relevance was not known so far. Using the new notion of a *conflict literal* we show that for independent clause sets N a clause C is syntactically semi-relevant in the clause set N if and only if it adds to the number of conflict literals in N. A clause set is independent, if no clause out of the clause set is the consequence of different clauses from the clause set.
Furthermore, we relate the notion of relevance to that of a minimally unsatisfiable subset (MUS) of some independent clause set N. In propositional logic, a clause C is relevant if it occurs in all MUSes of some clause set N and semi-relevant if it occurs in some MUS. For first-order logic the characterization needs to be refined with respect to ground instances of N and C.

1 Introduction

In our previous work [11], we introduced a notion of syntactic relevance based on refutations while at the same time generalized the completeness result for resolution by the set-of-support strategy (SOS) [28, 33] as its test. Our notion of syntactic relevance is useful for explaining why a set of clauses is unsatisfiable. In this paper, we introduce a semantic counterpart of syntactic relevance that sheds further light on the relationship between a clause out of a clause set and the potential refutations of this clause set. In the following Section 1.1, we first recall syntactic relevance along with an example and then proceeds to explain it in terms of our new semantic relevance in the later Section 1.2.

1.1 Syntactic Relevance

Given an unsatisfiable set of clauses N, $C \in N$ is *syntactically relevant* if it occurs in all refutations, it is *syntactically semi-relevant* if it occurs in some refutation, otherwise it is called *syntactically irrelevant*. The clause-based notion of relevance is useful in relating the contribution of a clause to refutation (goal conjecture). This has in particular been shown in the context of product scenarios built out of construction kits as they are used in the car industry [8, 32].

For an illustration of our privous notions and results we now consider the following unsatisfiable first-order clause set N where Figure 1 presents a refutation of N.

$$
\begin{aligned}
N = \{\ & (1)\ A(f(a)) \vee D(x_3),\\
& (2)\ \neg D(x_7),\\
& (3)\ \neg B(c,a) \vee B(b,f(x_6)),\\
& (4)\ B(x_1,x_2) \vee C(x_1),\\
& (5)\ \neg C(x_5),\\
& (6)\ \neg A(x_4) \vee \neg B(b,x_4)\}
\end{aligned}
$$

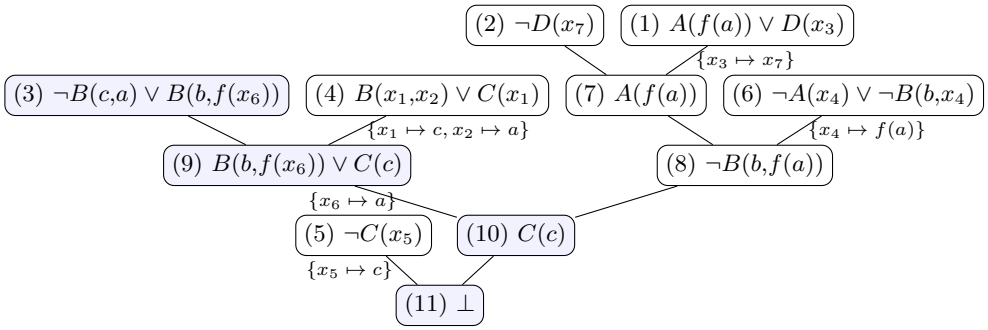

Fig. 1. A refutation of N in tree representation

In essence, inferences in an SOS refutation always involve at least one clause in the SOS and put the resulting clause back in it. So, this refutation is not an SOS refutation from the syntactically semi-relevant clause (3)$\neg B(c,a) \vee B(b,f(x_6))$, because only the shaded part represents an SOS refutation starting with this clause. More specifically, there are two inferences ended in (8)$\neg B(b,f(a))$ which violates the condition for an SOS refutaiton. Nevertheless, it can be transformed into an SOS refutation where the clause (3)$\neg B(c,a) \vee B(b,f(x_6))$ is in the SOS [11], Figure 2. Please note that $N \setminus \{(3)\neg B(c,a) \vee B(b,f(x_6))\}$ is still unsatisfiable and classical SOS completeness [33] is not sufficient to guarantee the existence of a refutation with SOS $\{(3)\neg B(c,a) \vee B(b,f(x_6))\}$ [11].

In addition, $N \setminus \{(3)\neg B(c,a) \vee B(b,f(x_6))\}$ is also a *minimally unsatisfiable subset* (MUS), where Figure 3 presents a respective refutation. A MUS is an unsatisfiable clause set such that removing a clause from this set would render it satisfiable. Consequently, a MUS-based defined notion of semi-relevance on the level of the original first-order clauses is not sufficient here. The clause (3)$\neg B(c,a) \vee B(b,f(x_6))$ should not be disregarded, because it leads to a different grounding of the clauses. For example, in the refutation of Figure 2 clause

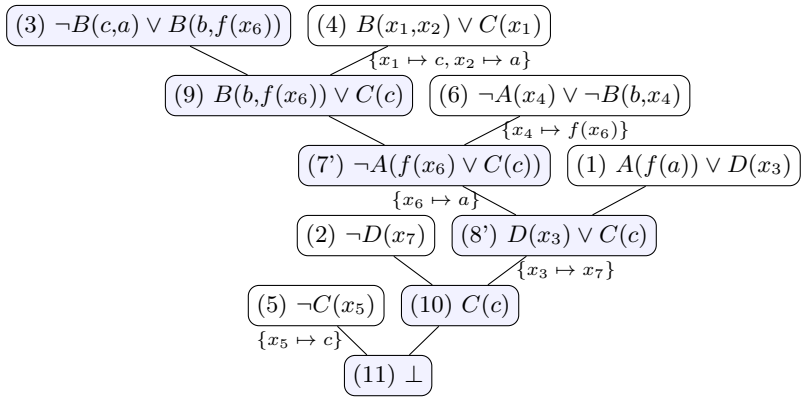

Fig. 2. Semi-relevant clause (3)$\neg B(c,a) \vee B(b,f(x_6))$ in SOS

(5)$\neg C(x_5)$ is necessarily instantiated with $\{x_5 \mapsto c\}$ where in the refutation of Figure 3 it is necessarily instantiated with $\{x_5 \mapsto b\}$. Therefore, the two refutations are different and clause (3)$\neg B(c,a) \vee B(b,f(x_6))$ should be considered semi-relevant. Nevertheless, in propositional logic it is sufficient to consider MUSes to explain unsatisfiability on the original clause level, Lemma 18.

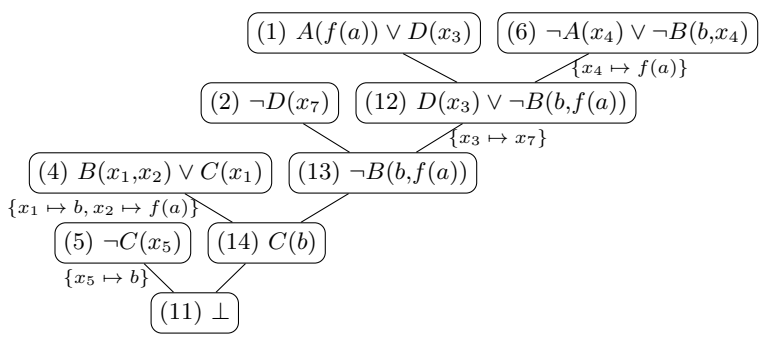

Fig. 3. A refutation of N without (3)$\neg B(c,a) \vee B(b,f(x_6))$

1.2 Semantic Relevance

We now illustrate how our new notion of relevance works on the previous example. First, different from the other works, we propose a way of characterizing semantic relevance by using our novel concept of a *conflict literal*. A ground literal L is a conflict literal in a clause set N if there are some satisfiable sets of instances N_1

and N_2 from N s.t. $N_1 \models L$ and $N_2 \models \text{comp}(L)$. On the one hand, explaining an unsatisfiable clause set as the absence of a model (as it is usually defined) is not that helpful since an absence means there is nothing to discuss in the first place. On the other hand, the contribution of a clause to unsatisfiability of a clause set can only partially be explained using the concept of a MUS which we have discussed before. A conflict literal provides a middle ground to explain the contribution of a clause to unsatisfiability between the absence of a model and MUSes. It also better reflects our intuition that there is a contradiction (in the form of two implied simple facts that cannot be both true at the same time) in an unsatisfiable set of clauses.

From Figure 1, we can already see that $C(c)$ and its complement $\neg C(c)$ are conflict literals because

$$\begin{aligned} N \setminus \{\neg C(x)\} &\models C(c) \\ \neg C(x) &\models \neg C(c) \end{aligned}$$

Also, in addition to that $\{\neg C(x)\}$ is trivially satisfiable, $N \setminus \{\neg C(x)\}$ is also satisfiable. Based on the refutation in Figure 3, $\neg C(x)$ is syntactically relevant due to $N \setminus \{(3)\neg B(c,a) \vee B(b, f(x_6))\}$ being a MUS. We will also show that for a ground MUS any ground literal occurring in it is a conflict literal, Lemma 20. For our ongoing example it is still possible to identify the conflict literals by means of ground MUSes by looking into the refutations from Figure 1 and Figure 3. This leads to the following conflict literals for N, see Definition 10:

$$\begin{aligned} \text{conflict}(N) = \{&(\neg)A(f(a)), \\ &(\neg)B(b, f(a)), (\neg)B(c, a), \\ &(\neg)C(b), (\neg)C(c)\} \quad \cup \\ &\{(\neg)D(t) \mid t \text{ is a ground term}\} \end{aligned}$$

These conflict literals can be identified by pushing the substitutions in the refutations from Figure 1 and Figure 3 towards the input clauses. They correspond to two first-order MUSes M_1 and M_2. All the ground literals are conflict literals and all other ground conflict literals can be obtained by grounding the remaining variables.

$$\begin{aligned} M_1 = \{&(5)\neg C(c), (2)\neg D(x_7), \\ &(1)A(f(a)) \vee D(x_3), \\ &(3)\neg B(c,a) \vee B(b, f(a)), \\ &(4)B(c,a) \vee C(c), \\ &(6)\neg A(f((a))) \vee \neg B(b, f(a))\} \\ M_2 = \{&(5)\neg C(b), \\ &(4)B(b, f(a)), (2)\neg D(x_7), \\ &(1)A(f(a)) \vee D(x_3), \\ &(6)\neg A(f(a)) \vee \neg B(b, f(a))\} \end{aligned}$$

One can see that, despite $(3)\neg B(c,a) \vee B(b, f(x_6))$ is outside of the only MUS on the first-order level, an instance of it does occur in some ground MUS, take M_1 and an arbitrary grounding of x_3 and x_7 to the identical term t, and the conflict literal $(\neg)B(c,a)$ depends on clause (3). Nevertheless, determining conflict literals is not so obvious in the general case since we do not necessarily know beforehand which ground terms should substitute the variables in the clauses. Moreover, there can be an infinite number of such ground MUSes of possibly unbounded size.

Based on conflict literals, here we introduce a notion of relevance that is semantic in nature, Definition 16. This will also serve as an alternative characterization to our previous refutation-based syntactic relevance. As redundant clauses, e.g., tautologies, can also be syntactically semi-relevant, we require independent clause sets for the definition of semantic relevance. A clause set is *independent*, if it does not contain clauses with instances implied by satisfiable sets of instances of different clauses out of the set. Given an unsatisfiable independent set of clauses N, a clause C is *relevant* in N if N without C has no conflict literals, it is *semi-relevant* if C is necessary to some conflict literals, and it is *irrelevant* otherwise.

Similar to our previous work, relevant clauses are the obvious ones because removing them would make our set satisfiable. On the other hand, irrelevant clauses can be freely identified once we know the semi-relevant ones. For our running example, in fact $(3)\neg B(c,a) \vee B(b, f(x_6))$ is semi-relevant because it is necessary for the conflict literals $(\neg)C(c)$ and $(\neg)B(c,a)$. More specifically, the set of conflicts for $N \setminus \{\neg B(c,a) \vee B(b, f(x_6))\}$ does not include $(\neg)C(c)$ and $(\neg)B(c,a)$:

$$\begin{aligned}\text{conflict}(N \setminus \{\neg B(c,a) \vee B(b, f(x_6))\}) = {} & \{(\neg)A(f(a)), (\neg)B(b, f(a)), (\neg)C(b)\} \uplus \\ & \{(\neg)D(t) | t \text{ is a ground term}\}\end{aligned}$$

These are conflict literals identifiable from M_2: Assume that the variables x_3 and x_7 in M_2 are both grounded by an identical term t. Take some ground literal, for example, $A(f(a)) \in \text{conflict}(N \setminus \{\neg B(c,a) \vee B(b, f(x_6))))$, and define

$$\begin{aligned}N_\emptyset &= \{C \in M_2 | A(f(a)) \notin C \text{ and } \neg A(f(a)) \notin C\} \\ &= \{(5)\neg C(b), (4)B(b, f(a)), (2)\neg D(t)\} \\ N_{A(f(a))} &= \{C \in M_2 | A(f(a)) \in C\} \\ &= \{(1)A(f(a)) \vee D(t)\} \\ N_{\neg A(f(a))} &= \{C \in M_2 | \neg A(f(a)) \in C\} \\ &= \{(6)\neg A(f(a)) \vee \neg B(b, f(a))\}\end{aligned}$$

$N_\emptyset \cup N_A(f(a))$ and $N_\emptyset \cup N_{\neg A(f(a))}$ are satisfiable because of the Herbrand model $\{B(b, f(a)), A(f(a))\}$ and $\{B(b, f(a))\}$ respectively. In addition,

$$\begin{aligned}N_\emptyset \cup N_A(f(a)) &\models A(f(a)) \\ N_\emptyset \cup N_{\neg A(f(a))} &\models \neg A(f(a))\end{aligned}$$

because $A(f(a))$ can be acquired using resolution between (1) and (2) for $N_\emptyset \cup N_{A(f(a))}$ and $\neg A(f(a))$ can be acquired using resolution between (4) and (6) for $N_\emptyset \cup N_{\neg A(f(a))}$. In a similar manner, we can show that the other ground literals are also conflict literals.

Related Work: Other works which aim to explain unsatisfiability mostly rely on the notion of MUSes, mainly in propositional logic [14–16, 21, 26]. The complexity of determining whether a clause set is a MUS is D^p-complete for a propositional clause set with at most three literals per clause and at most three occurrences of each propositional variable [25]. In [14], syntactically semi-relevant clauses for propositional logic are called a *plain clause set*. Using the terminology in [16], a clause $C \in N$ is *necessary* if it occurs in all MUSes, it is *potentially necessary* if it occurs in some MUS, otherwise, it is *never necessary*. In addition, a clause is defined to be *usable* if it occurs in some refutation. This is thus similar to our syntactic notion of semi-relevance [11]: Given a clause $C \in N$, C is usable if-and-only-if C is syntactically semi-relevant. It is also argued that a usable clause that is not potentially necessary is semantically superfluous. A different but related notion has also been applied for propositional abduction [7]. The notion of a MUS has also been used for explaining unsatisfiability in first-order logic [20]. There, it has been defined in a more general setting: If a set of clauses N is divided into $N = N' \uplus N''$ with a *non-relaxable* clause set N' and *relaxable* clause set N'' (which must be satisfiable), a MUS is a subset M of N'' s.t. $N' \uplus M$ is unsatisfiable but removing a clause from M would render it satisfiable. There are also some works in satisfiability modulo theory (SMT) [5, 6, 9, 35]. A deletion-based approach well-known in propositional logic has also been used for MUS extraction in SMT [9]. In [5, 6], a MUS is extracted by combining an SMT solver with an arbitrary external propositional core extractor. Another approach is to construct some graph representing the subformulas of the problem instance, recursively remove clauses in a depth-first-search manner and additionally use some heuristics to further improve the runtime [35]. For the function-free and equality-free first-order fragment, there is a "decompose-merge" approach to compute all MUSes [19, 34]. In description logic, a notion that is related to MUS is called *minimal axiom set* (MinA) usually identified by the problem of axiom pinpointing [1, 4, 13, 30]. Its computation is usually divided into two categories: black-box and white-box. A black-box approach picks some inputs, executes it using some sound and complete reasoner, and then interprets the output [13]. On the other hand, white-box approach takes some reasoner and performs an internal modification for it. In this case, Tableau is mostly used [1, 30]. In addition, the concept of a lean kernel has also been used to approximate the union of such MinA's [27]. The way relevance is defined is similar in spirit but usually used for an entailment problem instead of unsatisfiability. The notion of syntactic semi-relevance has also been applied to description logics via a translation scheme to first-order logic [10].

The paper is organized as follows. Section 2 fixes the notations, definitions and existing results in particular from [11]. Section 3 is reserved for our new

notion of semantic relevance. Finally, we conclude our work in Section 4 with a discussion of our results.

2 Preliminaries

We assume a standard first-order language without equality over a signature $\Sigma = (\Omega, \Pi)$ where Ω is a non-empty set of functions symbols, Π a non-empty set of predicate symbols both coming with their respective fixed arities denoted by the function arity. The set of terms over an infinite set of variables $\mathcal{X}$ is denoted by $T(\Sigma, \mathcal{X})$. Atoms, literals, clauses, and clause sets are defined as usual, e.g., see [24]. We identify a clause with its multiset of literals. Variables in clauses are universally quantified. Then N denotes a clause set; C, D denote clauses; L, K denote literals; A, B denote atoms; P, Q, R, T denote predicates; t, s terms; f, g, h functions; a, b, c, d constants; and x, y, z variables, all possibly indexed. The complement of a literal is denoted by the function comp. Atoms, literals, clauses, and clause sets are *ground* if they do not contain any variable.

An interpretation $\mathcal{I}$ with a nonempty *domain* (or *universe*) $\mathcal{U}$ assigns (i) a total function $f^{\mathcal{I}} : \mathcal{U}^n \mapsto \mathcal{U}$ for each $f \in \Omega$ with $\text{arity}(f) = n$ and (ii) a relation $P \subseteq \mathcal{U}^m$ to every predicate symbol $P^{\mathcal{I}} \in \Pi$ with $\text{arity}(P) = m$. A valuation β is a function $\mathcal{X} \mapsto \mathcal{U}$ where the assignment of some variable x can be modified to $e \in \mathcal{U}$ by $\beta[x \mapsto e]$. It is extended to terms as $\mathcal{I}(\beta) : T(\Sigma, \mathcal{X}) \mapsto \mathcal{U}$. Semantic entailment $\models$ considers variables in clauses to be universally quantified. The extension to atoms, literals, disjunctions, clauses and sets of clauses is as follows: $\mathcal{I}(\beta)(P(t_1, \ldots, t_n)) = 1$ if $(\mathcal{I}(\beta)(t_1), \ldots, \mathcal{I}(\beta)(t_n)) \in P^{\mathcal{I}}$ and 0 otherwise; $\mathcal{I}(\beta)(\neg\phi) = 1 - \mathcal{I}(\beta)(\phi)$; for a disjunction $L_1 \vee \ldots \vee L_k$, $\mathcal{I}(\beta)(L_1 \vee \ldots \vee L_k) = \max(\mathcal{I}(\beta)(L_1), \ldots, \mathcal{I}(\beta)(L_k))$; for a clause C, $\mathcal{I}(\beta)(C) = 1$ if for all valuations $\beta = \{x_1 \mapsto e_1, \ldots, x_n \mapsto e_n\}$ where the x_i are the free variables in C there is a literal $L \in C$ such that $\mathcal{I}(\beta)(L) = 1$; for a set of clauses $N = \{C_1, \ldots, C_k\}$, $\mathcal{I}(\beta)(\{C_1, \ldots, C_k\}) = \min(\mathcal{I}(\beta)(C_1), \ldots, \mathcal{I}(\beta)(C_k))$. A set of clauses N is *satisfiable* if there is an $\mathcal{I}$ of N such that $\mathcal{I}(\beta)(N) = 1$, β arbitrary, (in this case $\mathcal{I}$ is called a *model* of N: $\mathcal{I} \models N$) otherwise N is called *unsatisfiable*.

Substitutions σ, τ are total mappings from variables to terms, where $\text{dom}(\sigma) := \{x \mid x\sigma \neq x\}$ is finite and $\text{codom}(\sigma) := \{t \mid x\sigma = t, x \in \text{dom}(\sigma)\}$. A *renaming* σ is a bijective substitution. The application of substitutions is extended to literals, clauses, and sets/sequences of such objects in the usual way. If $C' = C\sigma$ for some substitution σ, then C' is an *instance* of C. A *unifier* σ for a set of terms $t_1, \ldots, t_k$ satisfies $t_i\sigma = t_j\sigma$ for all $1 \leq i, j \leq k$ and it is called a *most general unifier* if for any unifier σ' of $t_1, \ldots, t_k$ there is a substitution τ s.t. $\sigma' = \sigma\tau$. The function mgu denotes the *most general unifier* of two terms, atoms, literals if it exists. We assume that any mgu of two terms or literals does not introduce any fresh variables and is idempotent.

The resolution calculus consists of two inference rules: Resolution and Factoring [28, 29]. The rules operate on a state (N, S) where the initial state for a classical resolution refutation from a clause set N is $(\emptyset, N)$ and for an SOS (Set Of Support) refutation with clause set N and initial SOS clause set S the

initial state is (N, S). We describe the rules in the form of abstract rewrite rules operating on states (N, S). As usual we assume for the resolution rule that the involved clauses are variable disjoint. This can always be achieved by applying renamings into fresh variables.

Resolution $(N, S \uplus \{C \vee K\}) \Rightarrow_{\mathrm{RES}} (N, S \cup \{C \vee K, (D \vee C)\sigma\})$
provided $(D \vee L) \in (N \cup S)$ and $\sigma = \mathrm{mgu}(L, \mathrm{comp}(K))$

Factoring $(N, S \uplus \{C \vee L \vee K\}) \Rightarrow_{\mathrm{RES}} (N, S \cup \{C \vee L \vee K\} \cup \{(C \vee L)\sigma\})$
provided $\sigma = \mathrm{mgu}(L, K)$

The clause $(D \vee C)\sigma$ is the result of a *Resolution inference* between its parents and called a *resolvent*. The clause $(C \vee L)\sigma$ is the result of a *Factoring inference* of its parent and called a *factor*. A sequence of rule applications $(N, S) \Rightarrow^*_{RES} (N, S')$ is called a *resolution derivation*. It is called an *SOS resolution derivation* if $N \neq \emptyset$. In case $\perp \in S'$ it is a called a *(SOS) resolution refutation*. If for two clauses C, D there exists a substitution σ such that $C\sigma \subseteq D$, then we say that C *subsumes* D. In this case $C \models D$.

Theorem 1 (Soundness and Refutational Completeness of (SOS) Resolution [11, 28, 33]). *Resolution is sound and refutationally complete [28]. If for some clause set N and initial SOS S, N is satisfiable and $N \cup S$ is unsatisfiable, then there is a (SOS) resolution derivation of $\perp$ from (N, S) [33]. If for some clause set N and clause $C \in N$ there exists a resolution refutation from N using C, then there is an SOS derivation of $\perp$ from $(N \setminus \{C\}, \{C\})$ [11].*

Please note that the recent SOS completeness result of [11] generalizes the classical SOS completeness result by [33].

Theorem 2 (Deductive Completeness of Resolution [17, 22]). *Given a set of clauses N and a clause D, if $N \models D$, then there is a resolution derivation of some clause C from $(\emptyset, N)$ such that C subsumes D.*

For deductions we require every clause to be used exactly once, so deductions always have a tree form.

Definition 3 (Deduction [11]). *A* deduction $\pi_N = [C_1, \ldots, C_n]$ *of a clause C_n from some clause set N is a finite sequence of clauses such that for each C_i the following holds:*

1.1 C_i is a renamed, variable-fresh version of a clause in N, or
1.2 there is a clause $C_j \in \pi_N$, $j < i$ s.t. C_i is the result of a Factoring inference from C_j, or
1.3 there are clauses $C_j, C_k \in \pi_N$, $j < k < i$ s.t. C_i is the result of a Resolution inference from C_j and C_k,

and for each $C_i \in \pi_N$, $i < n$:

2.1 there exists exactly one factor C_j of C_i with $j > i$, or
2.2 there exists exactly one C_j and C_k such that C_k is a resolvent of C_i and C_j and $i, j < k$.

We omit the subscript N in π_N if the context is clear.

A deduction π' of some clause $C \in \pi$, where π, π' are deductions from N is a subdeduction of π if $\pi' \subseteq \pi$, where the subset relation is overloaded for sequences. A deduction $\pi_N = [C_1, \ldots, C_{n-1}, \bot]$ is called a *refutation*. While the conditions 3.1.1, 3.1.2, and 3.1.3 are sufficient to represent a resolution derivation, the conditions 3.2.1 and 3.2.2 force deductions to be minimal with respect to C_n.

Note that variable renamings are only applied to clauses from N such that all clauses from N that are introduced in the deduction are variable disjoint. Also recall that our notion of a deduction implies a tree structure. Both assumptions together admit the existence of overall grounding substitutions for a deduction.

Definition 4 (Overall Substitution of a Deduction [11]). *Given a deduction π of a clause C_n the* overall substitution *$\tau_{\pi,i}$ of $C_i \in \pi$ is recursively defined by*

1 if C_i is a factor of C_j with $j < i$ and mgu σ, then $\tau_{\pi,i} = \tau_{\pi,j} \circ \sigma$,
2 if C_i is a resolvent of C_j and C_k with $j < k < i$ and mgu σ, then $\tau_{\pi,i} = (\tau_{\pi,j} \circ \tau_{\pi,k}) \circ \sigma$,
3 if C_i is an initial clause, then $\tau_{\pi,i} = \emptyset$,

and the overall substitution of the deduction is $\tau_\pi = \tau_{\pi,n}$. We omit the subscript π if the context is clear.

Overall substitutions are well-defined because clauses introduced from N into the deduction are variable disjoint and each clause is used exactly once in the deduction. A grounding of an overall substitution τ of some deduction π is a substitution $\tau\delta$ such that $\mathrm{codom}(\tau\delta)$ only contains ground terms and $\mathrm{dom}(\delta)$ is exactly the variables from $\mathrm{codom}(\tau)$.

Definition 5 (SOS Deduction [11]). *A deduction $\pi_{N \cup S} = [C_1, \ldots, C_n]$ is called an* SOS deduction *with SOS S, if the derivation $(N, S_0) \Rightarrow^*_{RES} (N, S_m)$ is an SOS derivation where $C'_1, \ldots, C'_m$ is the subsequence from $[C_1, \ldots, C_n]$ with input clauses removed, $S_0 = S$, and $S_{i+1} = S_i \cup C'_{i+1}$.*

Oftentimes, it is of particular interest to identify the set of clauses that is minimally unsatisfiable, i.e., removing a clause would make it satisfiable. The earliest mention of such a notion is in [26] where it is introduced via a decision problem. Minimally unsatisfiable sets (MUS) have also gained a lot of attention in practice.

Definition 6 (Minimal Unsatisfiable Subset (MUS) [20]). *Given an unsatisfiable set of clauses N, the subset $N' \subseteq N$ is a minimally unsatisfiable subset (MUS) of N if any strict subset of N' is satisfiable.*

In our previous work, we defined a notion of relevance based on how clauses may contribute to unsatisfiability by means of refutations.

Definition 7 (Syntactic Relevance [11]). *Given an unsatisfiable set of clauses N, a clause $C \in N$ is* syntactically relevant *if for all refutations π of N it holds that $C \in \pi$. A clause $C \in N$ is* syntactically semi-relevant *if there exists a refutation π of N in which $C \in \pi$. A clause $C \in N$ is* syntactically irrelevant *if there is no refutation π of N in which $C \in \pi$.*

Syntactic relevance can be identified by using the resolution calculus. A clause $C \in N$ is syntactically semi-relevant if and only if there exists an SOS refutation from SOS $\{C\}$ and $N \setminus \{C\}$.

Theorem 8 (Syntactic Relevance [11]). *Given an unsatisfiable set of clauses N, the clause $C \in N$ is*

1. *syntactically relevant if and only if $N \setminus \{C\}$ is satisfiable,*
2. *syntactically semi-relevant if and only if $(N \setminus \{C\}, \{C\}) \Rightarrow^*_{RES} (N \setminus \{C\}, S \cup \{\bot\})$.*

An open problem from [11] is the question of a semantic counterpart to syntactic semi-relevance. Without any further properties of the clause set N, the notion of semi-relevance can lead to unintuitive results. For example, a tautology could be semi-relevant. Given a refutation showing semi-relevance of some clause C, where, in the refutation, some unary predicate P occurs, the refutation can be immediately extended using the tautology $P(x) \vee \neg P(x)$. We may additionally stumble upon a problem in the case where our set of clauses contains a subsumed clause. For example, if both clauses $Q(a)$ and $Q(x)$ exist in a clause set, they may be both semi-relevant, although from an intuition point of view one may only want to consider $Q(x)$ to be semi-relevant, or even relevant. On the other hand, in some cases, redundant clauses are welcome as semi-relevant clauses.

Example 9 (Redundant Clauses). Given a set of clauses

$$N = \{Q(x), \quad Q(a), \quad \neg Q(a) \vee P(b), \quad \neg P(b), \quad P(x) \vee \neg P(x)\},$$

all clauses are syntactically semi-relevant while $\neg Q(a) \vee P(b)$ and $\neg P(b)$ are syntactically relevant. However, if we disregard the redundant clauses $Q(a)$ and $P(x) \vee \neg P(x)$, then the clause $Q(x)$ becomes a relevant clause. Therefore, for our semantic notion of relevance we will only consider clause sets without clauses implied by other, different clauses from the clause set.

3 Semantic Relevance

Except for the trivially false clause $\bot$, the simplest form of a contradiction is two unit clauses K and L such that K and $\mathrm{comp}(L)$ are unifiable. They will be called *conflict literals*, below. Then the idea for our semantic definition of

semi-relevance is to consider clauses that contribute to the number of conflict literals of a clause set. Furthermore, we will show that in any MUS every literal is a conflict literal.

While conflict literals could straightforwardly be defined in propositional logic having the above idea in mind, in first-order logic we have always to relate properties of literals, clauses to their respective ground instances. This is simply due to the fact that unsatisfiability of a first-order clause set is given by unsatisfiability of a finite set of ground instances from this set. Eventually, we will show that for independent clause sets a clause is semi-relevant, if it contributes to the number of conflict literals.

Definition 10 (Conflict Literal). *Given a set of clauses N over some signature Σ, a ground literal L is a* conflict literal *in a clause set N if there are two satisfiable clause sets N_1, N_2 such that*

1. *the clauses in N_1, N_2 are instances of clauses from N and*
2. *$N_1 \models L$ and $N_2 \models \mathrm{comp}(L)$.*

$\mathrm{conflict}(N)$ *denotes the set of conflict literals in N.*

Our notion of a conflict literal generalizes the respective notion in [12] defined for propositional logic.

Example 11 (Conflict Literal). Given an unsatisfiable set of clauses over the signature $\Sigma = (\{a, b, c, d, f\}, \{P\})$:

$$N = \{\neg P(f(a, x)) \vee \neg P(f(c, y)), P(f(x, d)) \vee P(f(y, b))\}$$

Consider the following satisfiable sets of instances from N

$$N_1 = \{\neg P(f(a, d)) \vee \neg P(f(c, y)), P(f(x, d)) \vee P(f(a, b))\}$$
$$N_2 = \{\neg P(f(a, b)) \vee \neg P(f(c, y)), P(f(x, d)) \vee P(f(c, b))\}$$

$P(f(a, b))$ is a conflict literal because $N_1 \models P(f(a, b))$ and $N_2 \models \neg P(f(a, b))$.

We can show that $N_1 \models P(f(a, b))$ because the resolution calculus is sound. Resolving both literals of $\neg P(f(a, d)) \vee \neg P(f(c, y))$ with the first literal of the clause $P(f(x, d)) \vee P(f(a, b))$ results in the clause $P(f(a, b)) \vee P(f(a, b))$ which can be factorized to $P(f(a, b))$. Moreover, N_1 is satisfiable: An interpretation $\mathcal{I}$ with $\mathcal{I}(P(f(a, b))) = 1$ and $\mathcal{I}(P(t)) = 0$ for all terms $t \neq f(a, b)$ satisfies N_1 and $P(f(a, b))$. $N_2 \models \neg P(f(a, b))$ can also be shown in the same manner.

Example 12 (Conflict Literal). Given

$$\begin{aligned} N = \{ & \neg R(z), R(c) \vee P(a, y), \\ & Q(a), \neg Q(x) \vee P(x, b), \\ & \neg P(a, b)\} \end{aligned}$$

its conflict literals are

$$\begin{aligned}\text{conflict}(N) = \{\ & P(a,b), \neg P(a,b),\\ & R(c), \neg R(c),\\ & Q(a), \neg Q(a)\}\end{aligned}$$

In addition to a refutation, the existence of a conflict literal is another way to characterize unsatisfiability of a clause set. Obviously, conflict literals always come in pairs.

Lemma 13 (Minimal Unsatisfiable Ground Clause Sets and Conflict Literals). *If N is a minimally unsatisfiable set of ground clauses (MUS) then any literal occurring in N is a conflict literal.*

Proof. Take any ground atom A such that A occurs in N. N can be split into three disjoint clause sets:

$$\begin{aligned}N_\emptyset &= \{C \in N | A \notin C \text{ and } \neg A \notin C\}\\ N_A &= \{C \in N | A \in C\}\\ N_{\neg A} &= \{C \in N | \neg A \in C\}\end{aligned}$$

Since N is minimal, N_A and $N_{\neg A}$ are nonempty, because otherwise A is a pure literal and its corresponding clauses can be removed from N preserving unsatisfiability. Obviously $N_\emptyset \cup N_A$ must be satisfiable, for otherwise the initial choice of N was not minimal. However, $N_\emptyset \cup N'_A$, where N'_A results from all N_A by deleting all A literals from the clauses of N_A, must be unsatisfiable, for otherwise we can construct a satisfying interpretation for N. Thus, every model of $N_\emptyset \cup N_A$ must also be a model of A: $N_\emptyset \cup N_A \models A$. Using the same argument, $N_\emptyset \cup N_{\neg A}$ is satisfiable and $N_\emptyset \cup N_{\neg A} \models \neg A$. Therefore, A is a conflict literal. □

Lemma 14 (Conflict Literals and Unsatisfiability). *Given a set of clauses N,* $\text{conflict}(N) \neq \emptyset$ *if and only if N is unsatisfiable.*

Proof. "$\Rightarrow$" Let $L \in \text{conflict}(N)$. By definition, there are two satisfiable subsets of instances N_1, N_2 from N such that $N_1 \models L$ and $N_2 \models \text{comp}(L)$. Towards contradiction, suppose N is satisfiable. Then, there exists an interpretation $\mathcal{I}$ with $\mathcal{I} \models N$ and therefore it holds that $\mathcal{I} \models N_1$ and $\mathcal{I} \models N_2$. Furthermore, by definition of a conflict literal, $\mathcal{I} \models L$ and $\mathcal{I} \models \text{comp}(L)$, a contradiction.
"$\Leftarrow$" Given an unsatisfiable clause set N, we show that there is a conflict literal in N. Since N is unsatisfiable, by compactness of first-order logic there is a minimal set of ground instances N' from N that is also unsatisfiable. The rest follows from Lemma 13. □

Intuitively, a clause that is implied by other clauses is redundant and can be removed from the set of clauses. However, then applying a calculus generating new clauses, this intuitive notion of redundancy may destroy completeness [2, 23]. Still, the detection and elimination of redundant clauses, compatible or

incompatible with completeness, is an important concept to the efficiency of automatic reasoning, e.g., in propositional logic [3, 18]. It is also apparently important when we try to define a semantic notion of relevance. For example, a syntactically relevant clause would step down to be syntactically semi-relevant if it is duplicated. So, in order to have a semantically robust notion of relevance in first-order logic, we need to use a strong notion of (in)dependency.

Definition 15 (Dependency). *A clause C is* dependent *in N if there exists a satisfiable set of instances N' from $N \setminus \{C\}$ such that $N' \models C\sigma$ for some σ. If C is not dependent in N it is* independent *in N. A clause set N is* independent *if it does not contain any dependent clauses.*

A subsumed clause is obviously a dependent clause. However, there could also be non-subsumed clauses that are dependent. For example, in the set of clauses

$$N = \{P(a, y), P(x, b), \neg P(a, b)\}$$

$P(x, b)$ is dependent because $P(a, b)$ is an instance of $P(x, b)$ and it is entailed by $P(a, y)$. Now, we are ready to define the semantic notion of relevance based on conflict literals and dependency.

In some way, our notion of independence of clause sets is a strong assumption because there might be non-redundant clauses that are considered dependent. While this holds by design in some scenarios (e.g. the mentioned car scenario) in others it is violated by design. In addition, one question that may arise is how to acquire an independent clause set out of a dependent one. For example, in a scenario where some theory is developed out of some independent axioms. Then of course proven lemmas, theorems are dependent with respect to the axioms. In this case one could trace out of the proofs the dependency relations between the intermediate lemmas, theorems and the axioms and this way calculate independent clause sets with respect to some proven conjecture. This would then lead again to independent (sub) clause sets with respect to the proven conjecture where our results are applicable.

Definition 16 (Semantic Relevance). *Given an unsatisfiable set of independent clauses N, a clause $C \in N$ is*

1. relevant*, if* $\text{conflict}(N \setminus \{C\}) = \emptyset$
2. semi-relevant*, if* $\text{conflict}(N \setminus \{C\}) \subsetneq \text{conflict}(N)$
3. irrelevant*, if* $\text{conflict}(N \setminus \{C\}) = \text{conflict}(N)$

Example 17 (Dependent Clauses in Propositional Logic).

$$\begin{aligned} N = \{\ & P, \neg P, \\ & \neg P \vee Q, \neg R \vee P, \\ & \neg Q \vee R\} \end{aligned}$$

The existence of dependent clauses $\neg P \vee Q$ and $\neg R \vee P$ causes an independent clause $\neg Q \vee R$ to be a semi-relevant clause. However, $\neg Q \vee R$ is not inside the only MUS $\{P, \neg P\}$.

Very often, concepts from propositional logic can be generalized to first-order logic. However, in the context of relevance this is not the case. Our notion of (semi-)relevance can also be characterized by MUSes in propositional logic, but not in first-order logic without considering instances of clauses.

Lemma 18 (Propositional Clause Sets and Relevance). *Given an independent unsatisfiable set of propositional clauses N, the relevant clauses coincide with the intersection of all MUSes and the semi-relevant clauses coincide with the union of all MUSes.*

Proof. For the case of relevance: Given $C \in N$, C is relevant if and only if $\text{conflict}(N \setminus \{C\}) = \emptyset$ if and only if $N \setminus \{C\}$ is satisfiable by Lemma 14 if and only if C is contained in all MUSes N' of N.

For the case of semi-relevance: Given $C \in N$, we show C is semi-relevant if and only if C is in some MUS $N' \subseteq N$.
"$\Rightarrow$": Towards contradiction, suppose there is a semi-relevant clause C that is not in any MUS. By definition of semi-relevant clauses, there are satisfiable sets N_1 and N_2 and a propositional variable P such that $N_1 \models P$, $N_2 \models \neg P$ but the MUS M out of $N_1 \cup N_2$ does not contain C. By Theorem 2 there exist deductions π_1 and π_2 of P and $\neg P$ from N_1 and N_2, respectively. Since a deduction is connected, some clauses in M and $(N_1 \cup N_2) \setminus M$ must have some complementary propositional literals Q and $\neg Q$, respectively to be eventually resolved upon in either π_1 or π_2. At least one of these deductions must contain this resolution step between a clause from M and one from $(N_1 \cup N_2) \setminus M$. Now by Lemma 13 the literals Q and $\neg Q$ are conflict literals in M. Thus, there are satisfiable subsets from M which entail Q and $\neg Q$, respectively. Therefore, the clause containing Q or $\neg Q$ in $(N_1 \cup N_2) \setminus M$ is dependent contradicting the assumption that N does not contain dependent clauses.
"$\Leftarrow$": If C is in some MUS $N' \subseteq N$, then, $N' \setminus \{C\}$ is satisfiable. So invoking Lemma 13 any literal $L \in C$ is a conflict literal in N'. In addition, L is not a conflict literal in $N \setminus \{C\}$ for otherwise C is dependent: Suppose L is a conflict literal in $N \setminus \{C\}$ then, by definition, there is satisfiable subset from $N \setminus \{C\}$ which entails L. However, since $L \models C$, it means C is dependent. □

The next example demonstrates that the notion of a MUS cannot be carried over straightforwardly to the level of clauses with variables to characterize semi-relevant clauses in first-order logic.

Example 19 (First-Order Relevant Clauses). Given a set of clauses

$$\begin{aligned} N = \{\ & P(a,y), \neg P(a,d) \vee Q(b,d), \\ & \neg P(x,c), \neg Q(b,d) \vee P(d,c), Q(z,e)\} \end{aligned}$$

over $\Sigma = (\{a,b,c,d,e\}, \{P,Q\})$. The conflict literals are

$$\{(\neg)P(a,c), (\neg)Q(b,d), (\neg)P(d,c), (\neg)P(a,d)\}.$$

The clause $P(a, y)$ is relevant. The literals entailed by some satisfiable instances N' from N such that $P(a, y) \notin N'$ are $\{\neg Q(b, d)\} \uplus \{\neg P(t, c), \neg Q(t, e) \mid t \in \{a, b, c, d, e\}\}$ and no two of them are complementary. Thus, $\text{conflict}(N \setminus \{P(a, y)\}) = \emptyset$. The clause $\neg P(a, d) \vee Q(b, d)$ is semi-relevant: $Q(b, d) \notin \text{conflict}(N \setminus \{\neg P(a, d) \vee Q(b, d)\})$. The clause $Q(z, e)$ is irrelevant.

With respect to a MUS, the clause $\neg P(a, d) \vee Q(b, d)$ from Example 19 is irrelevant. The only MUS from N is $\{P(a, y), \neg P(x, c)\}$ with grounding substitution $\{x \mapsto a, y \mapsto c\}$. However, in first-order logic we should not ignore the clauses $\neg P(a, d) \vee Q(b, d)$, $\neg Q(b, d) \vee P(d, c)$, because together with the clauses $P(a, y), \neg P(x, c)$ they result in a different grounding $\{x \mapsto d, y \mapsto d\}$. So, we argue that MUS-based (semi-)relevance on the original clause set is not sufficient to characterize the way clauses are used to derive a contradiction for full first-order logic. However, it does so if ground instances are considered.

Lemma 20 (Relevance and MUSes on First-Order Clauses). *Given an unsatisfiable set of independent first-order clauses N. Then a clause C is relevant in N, if all MUSes of unsatisfiable sets of ground instances from N contain a ground instance of C. The clause C is semi-relevant in N, if there exists a MUS of an unsatisfiable set of ground instances from N that contains a ground instance of C.*

Proof. (Relevance) Since all ground instances from N contain a ground instance of C, then, if $N \setminus \{C\}$ contains a ground MUS from N it means that some ground instance of C is entailed by $N \setminus \{C\}$. This violates our assumption that N contains no dependent clauses. Thus, $N \setminus \{C\}$ contains no ground MUSes. This further means that $N \setminus \{C\}$ is satisfiable by the compactness theorem of first-order logic. By Lemma 14 it therefore has no conflict literals and C is relevant.
(Semi-Relevance) Take some ground MUS M containing some ground instance C' of C. Due to Lemma 13, any literal $P \in C'$ is a conflict literal in M and consequently also in N. In addition, P is not a conflict literal in $N \setminus \{C\}$ for otherwise C is dependent: Suppose P is a conflict literal in $N \setminus \{C\}$. Then, by definition, there is some satisfiable instances from $N \setminus \{C\}$ which entails P. However, since $P \models C'$, it means C is dependent. In conclusion, $P \in \text{conflict}(N) \setminus \text{conflict}(N \setminus \{C\})$ and thus C is semi-relevant. □

In Example 19, we could identify two ground MUSes:

$$\{P(a, c), \neg P(a, c)\}$$

and

$$\{P(a, d), \neg P(a, d) \vee Q(b, d), \neg P(d, c), \neg Q(b, d) \vee P(d, c)\}$$

Our notion of relevance is thus alternatively explainable using Lemma 20: $P(a, y)$ is relevant because every MUS contains an instance of it ($P(a, c)$ and $P(a, d)$). The clause $\neg P(a, d) \vee Q(b, d)$ is semi-relevant as it is immediately contained in the second MUS. The clause $Q(z, e)$ is irrelevant since no MUS contains any instance of $Q(z, e)$. On the other hand, we may still encounter the case where a dependent

clause is actually categorized as syntactically semi-relevant. Therefore, by using the dependency notion while at the same time not restricting a refutation to only use MUS as the input set, we can show that (semi-)relevance actually coincides with the syntactic (semi-)relevance. So, the semi-decidability result also follows.

Theorem 21 (Semantic versus Syntactic Relevance). *Given an independent, unsatisfiable set of clauses N in first-order logic, then (semi)-relevant clauses coincide with syntactically (semi)-relevant clauses.*

Proof. We show the following: if N contains no dependent clause, C is (semi-)relevant if and only if C is syntactically (semi-)relevant. The case for relevant clauses is a consequence of Lemma 14. Now, we show it for semi-relevant clauses.
"$\Rightarrow$" Let L be a ground literal with $L \in \text{conflict}(N) \setminus \text{conflict}(N \setminus \{C\})$. We can construct a refutation using C. There are two satisfiable subsets of instances N_1, N_2 from N such that $N_1 \models L$ and $N_2 \models \text{comp}(L)$ where $N_1 \cup N_2$ contains at least one instance of C, for otherwise $L \notin \text{conflict}(N) \setminus \text{conflict}(N \setminus \{C\})$. By the deductive completeness, Theorem 2, and the fact that L and $\text{comp}(L)$ are ground literals, there are two variable disjoint deductions π_1 and π_2 of some literals K_1 and K_2 such that $K_1\sigma = L$ and $K_2\sigma = \text{comp}(L)$ for some grounding σ. Obviously, the two variable disjoint deductions can be combined to a refutation $\pi_1.\pi_2.\bot$ containing C. Thus, C is syntactically semi-relevant in N.
"$\Leftarrow$" Given an SOS refutation π using C, i.e., an SOS refutation π from $N \setminus \{C\}$ with SOS $\{C\}$ and overall grounding substitution σ, we show that C is semantically semi-relevant. Let N' be the variable renamed versions of clauses from $N \setminus \{C\}$ used in the refutation and S' be the renamed copies of C used in the refutation. First, we show that $N'\sigma$ is satisfiable. Towards contradiction, suppose $N'\sigma$ is unsatisfiable and let $M\sigma \subseteq N'\sigma$ be its MUS. Since π is connected, some clauses in $M\sigma$ and $S'\sigma \cup (N'\sigma \setminus M\sigma)$ contains literals L and $\text{comp}(L)$ respectively. By Lemma 13, L and $\text{comp}(L)$ are also conflict literals in $M\sigma$. So, by Definition 15, the clause containing $\text{comp}(L)$ in $S'\sigma \cup (N'\sigma \setminus M\sigma)$ is dependent violating our initial assumption.
Now, since $N'\sigma$ is satisfiable, there is a ground MUS from $(N' \cup S')\sigma$ containing some $C'\sigma \in S\sigma$. Due to Lemma 13, any $L \in C'\sigma$ is a conflict literal in N' (and consequently also in N). In addition, L is not a conflict literal in $N \setminus \{C\}$ for otherwise C is dependent: Suppose L is a conflict literal in $N \setminus \{C\}$. Then, by definition, there is some satisfiable instances from $N \setminus \{C\}$ which entails L. However, since $L \models C'\sigma$, it means C is dependent. In conclusion, $L \in \text{conflict}(N) \setminus \text{conflict}(N \setminus \{C\})$ and thus C is semi-relevant. □

When we have a ground MUS, identifification of conflict literals is obvious because all of the literals in it are. However, testing if a literal L is a conflict literal is not trivial, in general. One can try enumerating all MUSes and check if L is contained in some. This definitely works for propositional logic despite being computationally expensive. In first-order logic, this is problematic because there could potentially be an infinite number of MUSes and determining a MUS is not even semi-decidable, in general. The following lemma provides a semi-decidable test using the SOS strategy.

Lemma 22. *Given a ground literal L and an unsatisfiable set of clauses N with no dependent clauses, L is a conflict literal if and only if there is an SOS refutation from* $(N, \{L \vee \mathrm{comp}(L)\})$.

Proof. "$\Rightarrow$" By the deductive completeness, Theorem 2, and the fact that L and $\mathrm{comp}(L)$ are ground literals, there are two variable disjoint deductions π_1 and π_2 of some literals K_1 and K_2 such that $K_1\sigma = L$ and $K_2\sigma = \mathrm{comp}(L)$ for some grounding σ. Obviously, the two variable disjoint deductions can be combined to a refutation $\pi_1.\pi_2.\bot$. We can then construct a refutation $\pi_1.\pi_2.(L \vee \neg L).(\mathrm{comp}(L)).\bot$ where K_2 is resolved with $L \vee \mathrm{comp}(L)$ to get $\mathrm{comp}(L)$ which will be resolved with K_1 from π_1 to get $\bot$. By Theorem 7, it means there is an SOS refutation from $(N, \{L \vee \neg L\})$
"$\Leftarrow$" Given an SOS refutation π using $\{L \vee \mathrm{comp}(L)\}$, i.e., an SOS refutation π from $N \setminus \{\{L \vee \mathrm{comp}(L)\}\}$ with SOS $\{\{L \vee \mathrm{comp}(L)\}\}$, Let N' be the variable renamed versions of clauses from N and overall grounding substitution σ. $N'\sigma$ is a MUS for otherwise there is a dependent clause: Suppose $N'\sigma \setminus M$ is an MUS where M is non-empty. Since π is connected, some clause D' in M must be resolved with some $D \in N'\sigma$ upon some literal K. Thus, by Lemma 13, K and $\mathrm{comp}(K)$ are also conflict literals in $N'\sigma \setminus M$. So, by Definition 15, the clause subsuming D' in N is dependent violating our initial assumption. Finally, because L occurs in $N'\sigma$ and $N'\sigma$ is an MUS, by Lemma 13, L is a conflict literal. $\square$

4 Conclusion

The main results of this paper are: (i) a semantic notion of relevance based on the existence of conflict literals, Definition 10, and Definition 16, (ii) its relationship to syntactic relevance, namely, both notions coincide for independent clause sets, Theorem 21, and (iii) the relationship of semantic relevance to minimal unsatisfiable sets, MUSes, both for propositional logic, Lemma 18, and first-order logic, Lemma 20.

The semantic relevance notion sheds some further light on the way clauses may contribute to a refutation beyond what can be offered by the notion of MUSes. While the syntactic notion of semi-relevance also considers redundant clauses such as tautologies to be semi-relevant, the semantic notion rules out redundant clauses. Here, the notions only coincide for independent clause sets. Still, the syntactic notion is "easier" to test and there are applications where clause sets do not contain implied clauses by construction. Hence, the syntactic-relevance coincides with semantic relevance. For example, first-order toolbox formalizations have this property because every tool is formalized by its own distinct predicate. Still a goal, refutation, can be reached by the use of different tools. The classic example is the toolbox for car/truck/tractor building [8, 31].

Acknowledgments: This work was partly funded by DFG grant 389792660 as part of TRR 248. We thank Christopher Lynch and David Plaisted for a number of discussions on semantic relevance. We thank the anonymous reviewers for their constructive and detailed comments.

References

1. Baader, F., Peñaloza, R.: Axiom pinpointing in general tableaux. J. Log. Comput. **20**(1), 5–34 (2010)
2. Bachmair, L., Ganzinger, H.: Resolution theorem proving. In: Robinson, A., Voronkov, A. (eds.) Handbook of Automated Reasoning, vol. I, chap. 2, pp. 19–99. Elsevier (2001)
3. Boufkhad, Y., Roussel, O.: Redundancy in random SAT formulas. In: Kautz, H.A., Porter, B.W. (eds.) Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on on Innovative Applications of Artificial Intelligence, July 30 - August 3, 2000, Austin, Texas, USA. pp. 273–278. AAAI Press / The MIT Press (2000)
4. Bourgaux, C., Ozaki, A., Peñaloza, R., Predoiu, L.: Provenance for the description logic elhr. In: Bessiere, C. (ed.) Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020. pp. 1862–1869. ijcai.org (2020)
5. Cimatti, A., Griggio, A., Sebastiani, R.: A simple and flexible way of computing small unsatisfiable cores in SAT modulo theories. In: Marques-Silva, J., Sakallah, K.A. (eds.) Theory and Applications of Satisfiability Testing - SAT 2007, 10th International Conference, Lisbon, Portugal, May 28-31, 2007, Proceedings. Lecture Notes in Computer Science, vol. 4501, pp. 334–339. Springer (2007)
6. Cimatti, A., Griggio, A., Sebastiani, R.: Computing small unsatisfiable cores in satisfiability modulo theories. J. Artif. Intell. Res. **40**, 701–728 (2011)
7. Eiter, T., Gottlob, G.: The complexity of logic-based abduction. Journal of the ACM **42**(1), 3–42 (1995)
8. Fetzer, C., Weidenbach, C., Wischnewski, P.: Compliance, functional safety and fault detection by formal methods. In: Margaria, T., Steffen, B. (eds.) Leveraging Applications of Formal Methods, Verification and Validation: Discussion, Dissemination, Applications - 7th International Symposium, ISoLA 2016, Imperial, Corfu, Greece, October 10-14, 2016, Proceedings, Part II. Lecture Notes in Computer Science, vol. 9953, pp. 626–632 (2016)
9. Guthmann, O., Strichman, O., Trostanetski, A.: Minimal unsatisfiable core extraction for SMT. In: Piskac, R., Talupur, M. (eds.) 2016 Formal Methods in Computer-Aided Design, FMCAD 2016, Mountain View, CA, USA, October 3-6, 2016. pp. 57–64. IEEE (2016)
10. Haifani, F., Koopmann, P., Tourret, S., Weidenbach, C.: On a notion of relevance. In: Borgwardt, S., Meyer, T. (eds.) Proceedings of the 33rd International Workshop on Description Logics (DL 2020) co-located with the 17th International Conference on Principles of Knowledge Representation and Reasoning (KR 2020), Online Event [Rhodes, Greece], September 12th to 14th, 2020. CEUR Workshop Proceedings, vol. 2663. CEUR-WS.org (2020)
11. Haifani, F., Tourret, S., Weidenbach, C.: Generalized completeness for SOS resolution and its application to a new notion of relevance. In: Platzer, A., Sutcliffe, G. (eds.) Automated Deduction - CADE 28 - 28th International Conference on Automated Deduction, Virtual Event, July 12-15, 2021, Proceedings. Lecture Notes in Computer Science, vol. 12699, pp. 327–343. Springer (2021)
12. Jabbour, S., Ma, Y., Raddaoui, B., Sais, L.: Quantifying conflicts in propositional logic through prime implicates. Int. J. Approx. Reason. **89**, 27–40 (2017)
13. Kalyanpur, A., Parsia, B., Horridge, M., Sirin, E.: Finding all justifications of OWL DL entailments. In: Aberer, K., Choi, K., Noy, N.F., Allemang, D., Lee, K., Nixon,

L.J.B., Golbeck, J., Mika, P., Maynard, D., Mizoguchi, R., Schreiber, G., Cudré-Mauroux, P. (eds.) The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11-15, 2007. Lecture Notes in Computer Science, vol. 4825, pp. 267–280. Springer (2007)
14. Kleine Büning, H., Kullmann, O.: Minimal unsatisfiability and autarkies. In: Biere, A., Heule, M., van Maaren, H., Walsh, T. (eds.) Handbook of Satisfiability, Frontiers in Artificial Intelligence and Applications, vol. 185, pp. 339–401. IOS Press (2009)
15. Kullmann, O.: Investigations on autark assignments. Discret. Appl. Math. **107**(1-3), 99–137 (2000)
16. Kullmann, O., Lynce, I., Marques-Silva, J.: Categorisation of clauses in conjunctive normal forms: Minimally unsatisfiable sub-clause-sets and the lean kernel. In: Biere, A., Gomes, C.P. (eds.) Theory and Applications of Satisfiability Testing - SAT 2006, 9th International Conference, Seattle, WA, USA, August 12-15, 2006, Proceedings. Lecture Notes in Computer Science, vol. 4121, pp. 22–35. Springer (2006)
17. Lee, C.T.: A Completeness Theorem and a Computer Program for Finding Theorems Derivable from Given Axioms. Phd thesis, University of Berkeley, California, Department of Electrical Engineering (1967)
18. Liberatore, P.: Redundancy in logic I: CNF propositional formulae. Artif. Intell. **163**(2), 203–232 (2005)
19. Liu, S., Luo, J.: FMUS2: an efficient algorithm to compute minimal unsatisfiable subsets. In: Fleuriot, J.D., Wang, D., Calmet, J. (eds.) Artificial Intelligence and Symbolic Computation - 13th International Conference, AISC 2018, Suzhou, China, September 16-19, 2018, Proceedings. Lecture Notes in Computer Science, vol. 11110, pp. 104–118. Springer (2018)
20. Marques-Silva, J., Mencía, C.: Reasoning about inconsistent formulas. In: Bessiere, C. (ed.) Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020. pp. 4899–4906. ijcai.org (2020)
21. Mencía, C., Kullmann, O., Ignatiev, A., Marques-Silva, J.: On computing the union of muses. In: Janota, M., Lynce, I. (eds.) Theory and Applications of Satisfiability Testing - SAT 2019 - 22nd International Conference, SAT 2019, Lisbon, Portugal, July 9-12, 2019, Proceedings. Lecture Notes in Computer Science, vol. 11628, pp. 211–221. Springer (2019)
22. Nienhuys-Cheng, S., de Wolf, R.: The equivalence of the subsumption theorem and the refutation-completeness for unconstrained resolution. In: Kanchanasut, K., Lévy, J. (eds.) Algorithms, Concurrency and Knowledge: 1995 Asian Computing Science Conference, ACSC '95, Pathumthani, Thailand, December 11-13, 1995, Proceedings. Lecture Notes in Computer Science, vol. 1023, pp. 269–285. Springer (1995)
23. Nieuwenhuis, R., Rubio, A.: Paramodulation-based theorem proving. In: Robinson, A., Voronkov, A. (eds.) Handbook of Automated Reasoning, vol. I, chap. 7, pp. 371–443. Elsevier (2001)
24. Nonnengart, A., Weidenbach, C.: Computing small clause normal forms. In: Robinson, A., Voronkov, A. (eds.) Handbook of Automated Reasoning, vol. 1, chap. 6, pp. 335–367. Elsevier (2001)
25. Papadimitriou, C.H., Wolfe, D.: The complexity of facets resolved. J. Comput. Syst. Sci. **37**(1), 2–13 (1988)
26. Papadimitriou, C.H., Yannakakis, M.: The complexity of facets (and some facets of complexity). J. Comput. Syst. Sci. **28**(2), 244–259 (1984)
27. Peñaloza, R., Mencía, C., Ignatiev, A., Marques-Silva, J.: Lean kernels in description logics. In: Blomqvist, E., Maynard, D., Gangemi, A., Hoekstra, R., Hitzler, P., Hartig,

O. (eds.) The Semantic Web - 14th International Conference, ESWC 2017, Portorož, Slovenia, May 28 - June 1, 2017, Proceedings, Part I. Lecture Notes in Computer Science, vol. 10249, pp. 518–533 (2017)
28. Robinson, J.A.: A machine-oriented logic based on the resolution principle. Journal of the ACM **12**(1), 23–41 (January 1965)
29. Robinson, J.A., Voronkov, A. (eds.): Handbook of Automated Reasoning (in 2 volumes). Elsevier and MIT Press (2001)
30. Schlobach, S., Cornet, R.: Non-standard reasoning services for the debugging of description logic terminologies. In: Gottlob, G., Walsh, T. (eds.) IJCAI-03, Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, Acapulco, Mexico, August 9-15, 2003. pp. 355–362. Morgan Kaufmann (2003)
31. Sinz, C., Kaiser, A., Küchlin, W.: Formal methods for the validation of automotive product configuration data. Artif. Intell. Eng. Des. Anal. Manuf. **17**(1), 75–97 (2003)
32. Walter, R., Felfernig, A., Küchlin, W.: Constraint-based and sat-based diagnosis of automotive configuration problems. J. Intell. Inf. Syst. **49**(1), 87–118 (2017)
33. Wos, L., Robinson, G., Carson, D.: Efficiency and completeness of the set of support strategy in theorem proving. Journal of the ACM **12**(4), 536–541 (1965)
34. Xie, H., Luo, J.: An algorithm to compute minimal unsatisfiable subsets for a decidable fragment of first-order formulas. In: 28th IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2016, San Jose, CA, USA, November 6-8, 2016. pp. 444–451. IEEE Computer Society (2016)
35. Zhang, J., Xu, W., Zhang, J., Shen, S., Pang, Z., Li, T., Xia, J., Li, S.: Finding first-order minimal unsatisfiable cores with a heuristic depth-first-search algorithm. In: Yin, H., Wang, W., Rayward-Smith, V.J. (eds.) Intelligent Data Engineering and Automated Learning - IDEAL 2011 - 12th International Conference, Norwich, UK, September 7-9, 2011. Proceedings. Lecture Notes in Computer Science, vol. 6936, pp. 178–185. Springer (2011)