On a Notion of Abduction and Relevance for First-Order Logic Clause Sets

Fajar Haifani

Dissertation

zur Erlangung des Grades
Doktor der Ingenieurwissenschaften (Dr.-Ing.)
der Fakultät für Mathematik und Informatik
der Universität der Saarlandes

Saarbrücken
2022

Tag des Kolloquiums:	9. März 2023
Dekan:	Univ.-Prof. Dr. Jürgen Steimle

Prüfungsausschuss

Vorsitzender:	Prof. Bernd Finkbeiner, Ph.D.
Berichterstatter:	Prof. Dr. Christoph Weidenbach
	Dr. Sophie Tourret
	Prof. Dr. Stephan Schulz
Akademischer Mitarbeiter:	Dr. Emanuele D'Osualdo

Abstract

I propose techniques to help with explaining entailment and non-entailment in first-order logic respectively relying on deductive and abductive reasoning.

First, given an unsatisfiable clause set, one could ask which clauses are necessary for any possible deduction (*syntactically relevant*), usable for some deduction (*syntactically semi-relevant*), or unusable (*syntactically irrelevant*). I propose a first-order formalization of this notion and demonstrate a lifting of this notion to the explanation of an entailment w.r.t some axiom set defined in some description logic fragments. Moreover, it is accompanied by a semantic characterization via *conflict literals* (contradictory simple facts). From an unsatisfiable clause set, a pair of conflict literals are always deducible. A *relevant* clause is necessary to derive any conflict literal, a *semi-relevant* clause is necessary to derive some conflict literal, and an *irrelevant* clause is not useful in deriving any conflict literals. It helps provide a picture of why an explanation holds beyond what one can get from the predominant notion of a minimal unsatisfiable set.

The need to test if a clause is (syntactically) semi-relevant leads to a generalization of a well-known resolution strategy: resolution equipped with the set-of-support strategy is refutationally complete on a clause set N and SOS M if and only if there is a resolution refutation from $N \cup M$ using a clause in M. This result non-trivially improves the original formulation.

Second, abductive reasoning helps find extensions of a knowledge base to obtain an entailment of some missing consequence (called observation). Not only that it is useful to repair incomplete knowledge bases but also to explain a possibly unexpected observation. I particularly focus on TBox abduction in $\mathcal{EL}$ description logic (still first-order logic fragment via some model-preserving translation scheme) which is rather lightweight but prevalent in practice. The solution space can be huge or even infinite. So, different kinds of minimality notions can help sort the chaff from the grain. I argue that existing ones are insufficient, and introduce *connection minimality*. This criterion offers an interpretation of Occam's razor in which hypotheses are accepted only when they help acquire the entailment without arbitrarily using axioms unrelated to the problem at hand. In addition, I provide a first-order technique to compute the connection-minimal hypotheses in a sound and complete way. The key technique relies on prime implicates. While the negation of a single prime implicate can already serve as a first-order hypothesis, a connection-minimal hypothesis which follows $\mathcal{EL}$ syntactic restrictions (a set of simple concept inclusions) would require a combination of them. Termination by bounding the term depth in the prime implicates is provable by only looking into the ones that are also subset-minimal. I also present an evaluation on ontologies from the medical domain by implementing a prototype with SPASS as a prime implicate generation engine.

Zusammenfassung

Ich schlage Techniken vor, die bei der Erklärung von Folgerung und Nichtfolgerung in der Logik erster Ordnung helfen, die sich jeweils auf deduktives und abduktives Denken stützen.

Erstens könnte man bei einer gegebenen unerfüllbaren Klauselmenge fragen, welche Klauseln für eine mögliche Deduktion notwendig (*syntaktisch relevant*), für eine Deduktion verwendbar (*syntaktisch semi-relevant*) oder unbrauchbar (*syntaktisch irrelevant*). Ich schlage eine Formalisierung erster Ordnung dieses Begriffs vor und demonstriere eine Anhebung dieses Begriffs auf die Erklärung einer Folgerung bezüglich einer Reihe von Axiomen, die in einigen Beschreibungslogikfragmenten definiert sind. Außerdem wird sie von einer semantischen Charakterisierung durch *Konfliktliteral* (widersprüchliche einfache Fakten) begleitet. Aus einer unerfüllbaren Klauselmenge ist immer ein Konfliktliteralpaar ableitbar. Eine *relevant*-Klausel ist notwendig, um ein Konfliktliteral abzuleiten, eine *semi-relevant*-Klausel ist notwendig, um ein Konfliktliteral zu generieren, und eine *irrelevant*-Klausel ist nicht nützlich, um Konfliktliterale zu generieren. Es hilft, ein Bild davon zu vermitteln, warum eine Erklärung über das hinausgeht, was man aus der vorherrschenden Vorstellung einer minimalen unerfüllbaren Menge erhalten kann.

Die Notwendigkeit zu testen, ob eine Klausel (syntaktisch) semi-relevant ist, führt zu einer Verallgemeinerung einer bekannten Resolutionsstrategie: Die mit der Set-of-Support-Strategie ausgestattete Resolution ist auf einer Klauselmenge N und SOS M widerlegungsvollständig, genau dann wenn es eine Auflösungswiderlegung von $N \cup M$ unter Verwendung einer Klausel in M gibt. Dieses Ergebnis verbessert die ursprüngliche Formulierung nicht trivial.

Zweitens hilft abduktives Denken dabei, Erweiterungen einer Wissensbasis zu finden, um eine implikantion einer fehlenden Konsequenz (Beobachtung genannt) zu erhalten. Es ist nicht nur nützlich, unvollständige Wissensbasen zu reparieren, sondern auch, um eine möglicherweise unerwartete Beobachtung zu erklären. Ich konzentriere mich besonders auf die TBox-Abduktion in dem leichten, aber praktisch vorherrschenden Fragment der Beschreibungslogik $\mathcal{EL}$, das tatsächlich ein Logikfragment erster Ordnung ist (mittels eines modellerhaltenden Übersetzungsschemas). Der Lösungsraum kann riesig oder sogar unendlich sein. So können verschiedene Arten von Minimalitätsvorstellungen helfen, die Spreu vom Weizen zu trennen. Ich behaupte, dass die bestehenden unzureichend sind, und führe *Verbindungsminimalität* ein. Dieses Kriterium bietet eine Interpretation von Ockhams Rasiermesser, bei der Hypothesen nur dann akzeptiert werden, wenn sie helfen, die Konsequenz zu erlangen, ohne willkürliche Axiome zu verwenden, die nichts mit dem vorliegenden Problem zu tun haben. Außerdem stelle ich eine Technik in Logik erster Ordnung zur Berechnung der verbindungsminimalen

Hypothesen in zur Verfügung korrekte und vollständige Weise. Die Schlüsseltechnik beruht auf Primimplikanten. Während die Negation eines einzelnen Primimplikant bereits als Hypothese in Logik erster Ordnung dienen kann, würde eine Hypothese des Verbindungsminimums, die den syntaktischen Einschränkungen von $\mathcal{EL}$ folgt (einer Menge einfacher Konzeptinklusionen), eine Kombination dieser beiden erfordern. Die Terminierung durch Begrenzung der Termtiefe in den Primimplikanten ist beweisbar, indem nur diejenigen betrachtet werden, die auch teilmengenminimal sind. Außerdem stelle ich eine Auswertung zu Ontologien aus der Medizin vor, Domäne durch die Implementierung eines Prototyps mit SPASS als Primimplikant-Generierungs-Engine.

Contents

List of Figures

List of Tables

Acknowledgments

First, I would like to express my gratitude to my supervisors, Christoph Weidenbach and Sophie Tourret. With deep knowledge and extensive experience in both research and industry (cars in particular), Christoph was able to point my research in interesting directions. He has given an important and unique perspective for both syntactic and semantic relevance. His support with SPASS was also essential in my abduction work. His supervision was continual, and I was always kept within realistic constraints. Therefore, in addition to his expertise, I am overall thankful for his leadership. Most importantly, I am grateful to him for giving me the privilege to be a part of his group in the first place and for bearing with me until the completion of my PhD. I would also like to thank Sophie Tourret for being very supportive with my abduction work. Her introducing me to the prime implicate-based abduction was the key to it. This second part of my dissertation has been uniquely challenging. Searching for an appropriate formalization for the intuitive notion of "connection" was not an easy task. I thank her for being patient with me. Moreover, she examined my works rigorously, and she has high attention to details in particular when it comes to writing. This has shaped my work in a no small way.

Thanks to Patrick Koopmann, Franz Baader, and the others who were involved in the same CPEC Work Package. Patrick has brought his description logic expertise and played an important role in providing suggestions, relevant DL notions, techniques, ontologies, existing tools, etc. His involvement was indispensable, in particular with the termination approach and experimentation. It was also his initiative that we submitted an abstract to the DL Workshop 2022 for which I received the best student contribution award. During our CPEC virtual meetings, Franz, and the others have also posed nice questions and informative insights. All of these, in one way or another, further catalyzed the progress of my PhD.

I am also grateful towards Stephan Schulz for his time and effort reviewing this dissertation.

I am thankful to the other group members (Hendrik Leidinger, Kurt Engesser, Lorenz Leutgeb, Martin Bromberger, Martin Deshraneis, Simon Schwarz, Sybille Möhle, Uwe Waldmann, Anna Rossien, Jennifer Müller, Johanna Meerjanssen, etc.) for their friendliness, help, critics, presentation feedback, administrative assistance, proofreading, etc. I also found a bit of entertainment taking part in their interests every once in a while (e.g., escape room, AStA-Kino, and wall climbing).

I thank my parents for being supportive of my decisions.

My time here certainly has not been a walk in the park. I definitely have not been able to fulfill all of my supervisors' expectations (for which I apologize).

Nonetheless, it has been impactful for me personally, and I am overall happy with how it turned out to be in the end.

This work is a part of Center for Perspicuous Computing (CPEC) project and supported by the International Max Planck Research Schools (IMPRS) and DFG grant 389792660 as part of TRR 248 “Foundations of Perspicuous Software Systems”.

Fajar Haifani

Introduction

Today, with the sheer amount of digital knowledge available, explaining a logical statement in computer (or computerized) systems has become more important than ever. On the one hand, if a sufficient amount of background knowledge is available, there could be many ways to pick the axioms to perform *deductive reasoning*. For example, given that there is some German-speaking minority in Brazil, then we can deduce that there is a German-speaking minority outside of Europe. But Brazil is not the only country useful for an explanation since there are other countries with a sizable minority of German-speaking citizens (e.g., South Africa). On the other hand, with partially available knowledge, an explanation may be also provided via a consistent hypothesis. For instance, if someone walks into a room soaking wet (but we do not know why), we may instinctively ask if it is raining outside. This is known as *abductive reasoning*. If we know that today is sunny from the weather report, then this hypothesis is rejected. In other words, a hypothesis may only be acceptable if *consistent* with the knowledge at hand. However, even with this restriction, there could be other explanations (e.g., problem when fixing the kitchen sink) and the number of hypotheses may be very large, if not infinite. Choosing the ones to use is thus not an easy task. I propose new notions to help picking axioms or hypotheses in both cases.

For the first task, when there is a sufficient amount of knowledge, I propose a notion of relevance according to how the axioms are used in the possible deductions. For the previous German speaker example, the knowledge about what constitutes a minority (e.g., up to 10% of the population) is necessary and thus called *relevant*. The fact that Brazil hosts some German-speaking population can be used, but it is not relevant due to it being replaceable by South Africa. In other words, this is *semi-relevant*. The fact that Antarctica is in the southern hemisphere is simply *irrelevant*. In a more formal setting, existing notions usually come with additional restrictions (e.g., subset minimality, preference for shorter deductions, etc.). However, I argue here that simply using deduction with no further preconditions can more naturally model our intuitive notion of relevance. Moreover, as I will illustrate later via a semantic argument, having additional preconditions may, in some cases, come at the price of losing something interesting.

For the second task, when the knowledge at hand is not sufficient to prove an observation, I propose a notion to restrict the hypotheses to the ones using parts of the background knowledge that are ”connected” to the considered statement. This follows the principle of parsimony (Occam’s razor) because disconnected statements are not used. I call this *connection minimality*. For example, upon coming to a university, if a new student overheard that ”Students pay less in the cafeteria" and he knows that "I will get a student card", "The cafeteria accepts payment by cards", and "The university is subsidized", then he can guess that "The cafeteria charges less with student card payment and its price difference is taken from the subsidy". The student uses his current knowledge only when related to the observation (e.g., "student card" is related to "student" and "subsidy" to "paying less"). Even though he may guess incorrectly (It could be that, the cafeteria takes some money from students' tuition fee instead), it makes much less sense to involve something unrelated such as "There is an art museum near my apartment". I would further argue that this consideration for "connectedness" is very natural with many familiar examples: A detective interrogating people having both motivations and means to perform a crime (not a random janitor in the other building), someone losing a key tracing back only the places he visited (not asking a security guard in a random nearby supermarket), etc. I therefore believe that the potential goes beyond the logical formalism considered in this dissertation.

Both notions are respectively formalized in first-order logic and $\mathcal{EL}$ description logic. In technical terms, the tasks amount to explaining entailment and non-entailment. For the first task, entailment can be shown specifically via a refutation for an unsatisfiable set of clauses. Dealing with unsatisfiable sets of clauses is convenient in first-order logic, since an arbitrary entailment can be easily reduced to an entailment of the empty clause. Existing literature heavily relies on the notion of a subset-minimal set of axioms that is still unsatisfiable (*minimal unsatisfiable set* MUS) in particular if one deals with propositional logic. This is because if the input clauses of a propositional refutation do not correspond to a MUS, then for sure there is a clause that can be inferred from the others (*redundant*) in it. Both refutations and MUSes have shortcomings. On the one hand, a refutation may still use *redundant* clauses (the ones implied by the other clauses) in the input. In other words, it may include clauses that are not that interesting. On the other hand, as we will see later, using MUS may subtly exclude some interesting clauses. To show this, I also provide a semantic characterization for this refutation-based relevance. A set of clauses is unsatisfiable if there are satisfiable instantiations of some of the clauses entailing two conflicting facts (called *conflict literal*). Then, this semantic characterization shows how a clause may contribute to the number of conflict literals: a *relevant* clause must be used to infer any conflict literals, a *semi-relevant* clause must be used to infer some conflict literal otherwise it is *irrelevant*. Via this semantic notion, I will show what kind of interesting clauses are missing if a notion based on MUS is used. On the one hand, MUS is not completely gone. Here, it must be related to the ground instantiation of the original clauses. On the other hand, redundancy must also be refined in terms of its ground instantiation. This, however, may turn the set of MUSes infinite and many MUS-based reasoning tasks that are often available in propositional logic (e.g., listing all MUSes, computing their union) cannot be performed anymore. Thus, our notion of relevance may serve not only as an alternative to MUS-based notions but also offers something new.

In the case of a non-entailment, existing works include using interpretations as counter-examples and using abductive reasoning. The first one is the obvious way because it uses its definition: there is a model of the background knowledge that is not a model of the observation. The disadvantage of this is that a model may contain parts that are unrelated to the entailment. The considered fragments may also problematically allow infinite models. The predominant approach is via abductive reasoning. That is, we want to find some set of clauses to add to the original clause set to obtain the entailment. Our approach is formalized in the description logic $\mathcal{EL}$. Different from other notions, our proposed *connection-minimality* notion follows the principle of parsimony (Occam's razor) by not using the axioms unrelated to the problem at hand. It is based on the fact that, in $\mathcal{EL}$, if we have a set of terminological axioms (TBox) $\mathcal{T}$ and an *is-a* relation $U_1 \sqsubseteq U_2$ between simple concepts s.t. $\mathcal{T} \models U_1 \sqsubseteq U_2$, then there is a possibly complex concept V (called *connecting concept*) s.t. $\mathcal{T} \models U_1 \sqsubseteq V$ and $\mathcal{T} \models V \sqsubseteq U_2$ (and vice versa). Connection-minimality tries to form hypotheses essentially by forming a connecting concept via two concepts V_1 and V_2 s.t. $\mathcal{T} \models U_1 \sqsubseteq V_1$ and $\mathcal{T} \models V_2 \sqsubseteq U_2$. A connecting concept is constructed by considering the syntactical structure of V_1 and V_2 which enables our hypotheses to entail $U_1 \sqsubseteq U_2$. The computation works by translating the problem into first-order logic, generating first-order prime implicates, then reconstructing a solution by combining the prime implicates. I show that it is sound, complete, and terminating for a class of hypotheses that are subset minimal.

To show the potential of our connection-minimality in practice, I also present the result of an experiment on publicly available $\mathcal{EL}$ ontologies from the bio-medical domain. SPASS [WSH+07] serves as a prime implicate generation engine and the pre- and postprocessing tasks take advantage of some DL tools such as ELK and OWL API. At the first-order level, the SOS strategy [WRC65, HTW21] and a restriction on the number of variables in any derived clauses are employed. Note that there exist resolution-based calculi devised natively in DL and have been used immediately for ontology [KDTS20] and ABox [DS19] abduction. In [KES11] first-order SOS resolution (similar to ours) serves as a reasoning tool for ABox abduction in $\mathcal{ALC}$. Apart from being defined on different abduction tasks, the notion of connection-minimality additionally serves as a semantic characterization of the generated hypotheses. This is a novelty that may be viewed independently without any first-order logic terminology. Thus, its adoption by DL community should arguably be comparatively better than [KES11]. Since this minimality notion is then calculus agnostic, either one wants to use first-order prime implicates or native DL calculi, any advances in both can be beneficial for its computation.

The following two sections provide illustrations for these notions on a more technical level.

Explaining Entailment via a New Notion of Relevance

I introduce a syntactic notion of relevance defined on the clauses in terms of how they are used to derive the empty clause. In particular, we rely on the notion of a *refutation*: a sequence of clauses starting from some input clauses s.t. all the later clauses are generated via the resolution and factoring inferences from the previous clauses and every clause in it is used by at least one later clause except for the last (which is the empty clause). Given an unsatisfiable clause set N, $C \in N$ is *syntactically relevant* if any refutation must use this clause, it is *syntactically semi-relevant* if it appears in some refutation otherwise, it is *syntactically irrelevant*. The refutation-based notion of relevance is useful in relating the involvement of a clause to refutation (goal conjecture). This also promises applicability in car industry where product scenarios are built from construction kits [FWW16, WFK17]. In [FWW16], an online *auditor* is attached to a car to determine, for instance, whether some clean air/emission regulation, engine safety, etc. are satisfied. The notion of relevance would then be useful to identify what may cause a violation (e.g., GPS data, gas consumption, car location, etc.). In [WFK17], when car construction violates some constraints (represented in propositional logic), its diagnosis can be delivered via a *maximal satisfiable subset* and *minimal correction subset*. In particular, the minimal correction subset is related to our syntactically relevant clauses. Its use cases have also been shown for real-world automotive configuration data of three German car manufacturers.

As an illustration, we provide an example of how our notion can be useful in explaining an entailment. Consider the following set of formulas

$$\begin{aligned}\Phi = \{&\forall x_1.\neg\mathsf{deal}(x_1, \mathsf{amd}),\\ &(\exists x_2.\mathsf{deal}(x_2, \mathsf{asus})) \rightarrow \mathsf{deal}(\mathsf{asus}, \mathsf{amd}),\\ &\mathsf{deal}(\mathsf{microsoft}, \mathsf{intel})\}\end{aligned}$$

representing the following axioms in natural language:

- no one signs a contract with Amd,
- if a company signs a contract with Asus, then Asus will also sign a contract with Amd, and
- Microsoft signs a contract with Intel.

In Φ, it holds that "Intel has no contract with a company":

$$\Phi \models \exists x_3.\neg\mathsf{deal}(\mathsf{intel}, x_3)$$

One possible way to get this informally is from the first axiom: since no one signs a contract with Amd, it means that Intel does have a contract with Amd. First, we clausify Φ and add the negation of $\exists x_3.\neg\mathsf{deal}(\mathsf{intel}, x_3)$ to get the following clause set N.

$$\begin{aligned}N = \{&\neg\mathsf{deal}(x_1, \mathsf{amd}),\\ &\neg\mathsf{deal}(x_2, \mathsf{asus}) \vee \mathsf{deal}(\mathsf{asus}, \mathsf{amd}),\\ &\mathsf{deal}(\mathsf{microsoft}, \mathsf{intel}),\\ &\mathsf{deal}(\mathsf{intel}, x_3)\}\end{aligned}$$

The translation is rather straightforward. No Skolemization takes place and the original formulas have a one-to-one correspondence with the clauses. It even holds that, $N \setminus \{\mathsf{contract}(\mathsf{intel}, \mathsf{x_3})\}$ is equivalent to Φ. From the clause set $M_1 \subseteq N$

$$\begin{aligned}M_1 = \{&\neg\mathsf{deal}(x_1, \mathsf{amd}),\\ &\mathsf{deal}(\mathsf{intel}, x_3)\}\end{aligned}$$

we know that N is unsatisfiable as can be shown by the refutation in Fig. 1.

As can be seen from the variable grounding, Intel has no contract with Amd, proving the sentence $\exists x_3.\neg\mathsf{deal}(\mathsf{intel}, x_3)$.

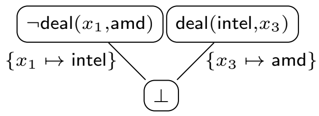

Figure 1: A refutation tree from $M_1 \subseteq N$ with overall substitution $\sigma_1 = \{x_1 \mapsto \mathsf{intel}, x_3 \mapsto \mathsf{amd}\}$

Figure 1 is apparently not the only possible refutation. Figure 2 shows another one using the following clauses from N.

$$M_2 = \{\neg\mathsf{deal}(x_1, \mathsf{amd}),\\ \neg\mathsf{deal}(x_2, \mathsf{asus}) \vee \mathsf{deal}(\mathsf{asus}, \mathsf{amd}),\\ \mathsf{deal}(\mathsf{intel}, x_3)\}$$

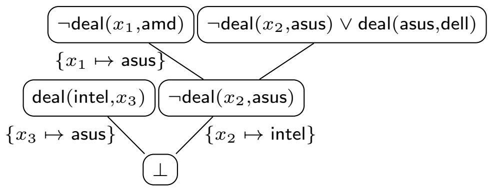

Figure 2: A refutation tree from $M_2 \subseteq N$ with overall substitution $\sigma_2 = \{x_1 \mapsto \mathsf{asus}, x_2 \mapsto \mathsf{intel}, x_3 \mapsto \mathsf{asus}\}$

From these two refutations, we can conclude that in N, we have M_1 consisting of the syntactically relevant clauses, M_2 consisting of the syntactically semi-relevant clauses. The remaining one clause $\mathsf{deal}(\mathsf{microsoft}, \mathsf{intel})$ is an irrelevant clause since it is not even possible to resolve it with anything.

A lot of works have been done in propositional logic and rely on the notion of minimal unsatisfiable set MUS (where removing a clause would render it satisfiable). Propositional clauses outside of any MUSes in some clause set N are either redundant or syntactically irrelevant. For the following clause set,

$$N = \{P, \neg P, \neg P \vee Q, \neg Q \vee P\}$$

the last two clauses are implied by the other. They are not in the only MUS $\{P, \neg P\}$ but can be involved in a refutation as in Fig. 3.

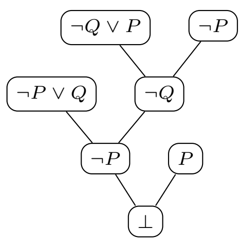

Figure 3: A propositional refutation with dependent clauses

It is obvious that the sub-tree rooted at the non-leaf $\neg P$ is superfluous. One can simply use the last inference between P and $\neg P$. I do not cherry pick this example and I will prove that the slightly general case always holds: if a clause in a ground first-order clause set is outside of any MUSes then it will either be irrelevant or redundant. Nonetheless, first-order logic is different. Some interesting clause that is syntactically semi-relevant may actually be outside of any MUSes. From the example, we know that there is a single MUS $M_1 = \{\mathsf{deal}(\mathsf{intel}, x_3), \neg\mathsf{deal}(x_1, \mathsf{amd})\}$ but the semi-relevant clauses in $M_2 \setminus M_1$ are not empty yet they are not redundant.

To make it more precise what is meant by "interesting", I now describe a semantic relevance accompanying this syntactic notion. First, we need the notion of *conflict literal*. A ground literal L is a conflict literal in a clause set N if $N_1 \models L$ and $N_2 \models \mathsf{comp}(L)$ for some satisfiable sets N_1 and N_2 containing instances of clauses in N. On the one hand, expressing that a clause set is unsatisfiable via the absence of a model (as is defined) is not very helpful since an absence means there is nothing in the first place. On the other hand, using MUS in first-order logic by considering the ground instances (as we will require later on) would mean that the union of the MUSes could be infinite and thus not computable, as in the case in propositional logic. A conflict literal provides some sort of trade-off between the absence of a model and MUSes. Furthermore, I argue that it is more intuitive in the sense that there is a contradiction (in the form of two contradictory simple facts) in the considered clause set.

Similar to propositional logic, first-order clause sets may also contain redundant clauses. However, the usual semantic redundancy is rather too strong and thus we need a refinement done via instantiation called independence: 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. With this, we now have the semantic relevance: given an unsatisfiable independent set of clauses N, a clause is *relevant* in N if there is no conflict literal in $N \setminus C$, it is *semi-relevant* if C is necessary to some conflict literals, and it is *irrelevant* otherwise.

As we have seen earlier, $M_2 \setminus M_1$ consists of the syntactically semi-relevant clauses (but outside of the MUS) and M_1 is the syntactically relevant clauses. Unlike in the propositional case, any clause in $M_2 \setminus M_1$ is not redundant w.r.t. the other clauses. Upon a closer look, even though $M_1 \subsetneq M_2$, this does not hold anymore when the substitutions are respectively applied: $M_1\sigma_1 \not\subseteq M_1\sigma_2$. Moreover, $M_1\sigma_1$ and $M_1\sigma_2$ are MUSes. $M_1\sigma_1$ already consists of the conflict literals $(\neg)\mathsf{deal}(\mathsf{intel}, \mathsf{amd})$ because each of them is already a satisfiable instance on N (thus entailed by themselves). This is also in accordance with our informal explanation that there is a company that Intel does not have a contract with (i.e., Amd). Nevertheless, $(\neg)\mathsf{deal}(\mathsf{intel}, \mathsf{asus})$ are conflict literals because they are also entailed by some satisfiable instances of N:

$$\begin{aligned} \{\neg\mathsf{deal}(x_1, \mathsf{amd}), \neg\mathsf{deal}(x_2, \mathsf{asus}) \vee \mathsf{deal}(\mathsf{asus}, \mathsf{amd})\} &\models \neg\mathsf{deal}(\mathsf{intel}, \mathsf{asus}) \\ \{\mathsf{deal}(\mathsf{intel}, \mathsf{asus})\} &\models \mathsf{deal}(\mathsf{intel}, \mathsf{asus}) \end{aligned}$$

This conflict literal also hints that Intel also does not have a contract with Asus. This can serve as an alternative explanation for the original formula $\exists x. \neg\mathsf{contracts}(\mathsf{intel}, x)$: we can infer that "Intel does not have a contract with Asus" instead of the original one with Amd. Semantically, Both of the following hold:

$$\Phi \models \mathsf{deal}(\mathsf{intel}, \mathsf{amd})$$
$$\Phi \models \mathsf{deal}(\mathsf{intel}, \mathsf{asus})$$

I argue that both should alternatively be useful and it does not make sense to exclude Asus in favor of Amd. However, the entailment $\Phi \models \mathsf{deal}(\mathsf{intel}, \mathsf{asus})$ needs a clause that is not in the original MUS M_1: $\{\neg\mathsf{deal}(x_1, \mathsf{amd}), \mathsf{deal}(\mathsf{intel}, x_3)\}$. In summary, the notion of relevance would then be as follows.

– $\neg\mathsf{deal}(x_1, \mathsf{amd})$ and $\mathsf{deal}(\mathsf{intel}, x_3)$ are relevant

– $\neg\mathsf{deal}(x_2, \mathsf{asus}) \vee \mathsf{deal}(\mathsf{asus}, \mathsf{amd})$ is semi-relevant

– $\mathsf{deal}(\mathsf{microsoft}, \mathsf{intel})$ is irrelevant

I emphasize here that $\neg\mathsf{deal}(x_2, \mathsf{asus}) \vee \mathsf{deal}(\mathsf{asus}, \mathsf{amd})$ is not irrelevant. In the car industry, this example should also sufficiently illustrate that a similar problem may arise when identifying the cause of a constraint violation in a car construction recipe [WFK17], or the car components causing a runtime violation of some, e.g., safety/clean air regulations [FWW16] when defined in first-order logic.

Generalized Completeness for the SOS Strategy

The notions described in the previous section naturally prompt the need for a test. The main calculus used for this is the resolution calculus with the set-of-support (SOS) strategy [WRC65]. This was the first refinement of the refutational completeness proven shortly after the formulation of first-order resolution [Rob65]. The SOS strategy splits the given clause set into two sets, namely N and M, and allows only resolution inferences involving at least one parent from the set-of-support M and puts back the resulting clause to M. Wos et al. [WRC65] proved the SOS strategy complete with the restriction that N is satisfiable. The motivation by Wos et. al. for the SOS strategy was getting rid of "irrelevant" inferences. If N defines a theory and M contains the negation of a conjecture (goal) to be refuted, the strategy emphasizes resolution inferences involving the conjecture. This can be beneficial in terms of efficiency because deductive completeness (modulo subsumption) [Lee67,NdW95] allows the resolution inferences solely performed on clauses from N to enumerate *all* semantic consequences, even if they are not potentially useful in refuting $N \cup M$.

The established SOS refutational completeness is already well suited to test if a clause is (syntactically) relevant. Taking out a (syntactically) relevant clause from an unsatisfiable clause set would turn it satisfiable. This already fits the condition for the existence of an SOS refutation: a clause $C \in N$ is relevant iff $N \setminus \{C\}$ is satisfiable and there is an SOS refutation from $N \setminus \{C\}$ with SOS $\{C\}$.

The key result which guarantees the possibility to test for (syntactic) semi-relevance is the generalization of the original completeness result for the SOS strategy: The resolution calculus with the SOS strategy is complete if and only if there is a clause in M used in a resolution refutation from $N \cup M$, Th. 3.2.7. The key to showing this is via a proof transformation technique. Any (non SOS) refutation from $N \cup M$ can be turned into an SOS refutation with SOS M, if the original refutation uses at least a clause in M.

Syntactic Relevance for Description Logics

Description logic [BHLS17] is used to represent terminological and assertional knowledge in which the key syntactical constructs are concepts (unary predicate), roles (binary predicates), and individuals (constants). It has been used in areas like bio-medicine or the semantic web. The lightweight description logic $\mathcal{EL}$ has been particularly useful to represent knowledge in the biomedical domain. They often consist of a hundred thousand axioms. For instance, SNOMED CT[1] contains over 350,000 axioms, and the Gene Ontology GO[2] defines over 50,000 concepts. Explaining entailments—i.e., explaining why a DL axiom holds—is rather well-known in the literature and also featured in the standard ontology editors [HPS08a, KKS17].

Now, I show how first-order refutations can be used for description logic entailment[3] as described in our workshop paper [HKTW20]. An axiom set $\mathcal{O}$ in description logics is called an ontology and usually further consists of two sets $\mathcal{T}$ and $\mathcal{F}$ which are terminological knowledge (is-a relation between concepts such as $\mathsf{Human} \sqsubseteq \mathsf{LivingBeing}$) and ground facts respectively. Some of its fragments would not allow for unsatisfiable ontologies via its syntactic restriction. The axiom α in an entailment $\mathcal{O} \models \alpha$ is not empty. We consider the clauses translated from $\mathcal{O}$ and the negation of clauses from α. An axiom is then *syntactically relevant* if any refutation always contains some clause out of this axiom, *syntactically semi-relevant* if there is a refutation containing some clause out of this axiom and *syntactically irrelevant* otherwise.

As an example, given an ontology $\mathcal{O} = \mathcal{T} \uplus \mathcal{F}$ with

$$\begin{aligned}\mathcal{T} &= \{ \exists \mathsf{withBase}.\mathsf{PizzaBase} \sqcap \exists \mathsf{withTop}.\mathsf{PizzaTopping} \sqsubseteq \mathsf{Pizza}\} \\ \mathcal{F} &= \{ \mathsf{withTop}(\mathsf{tunaPizza}, \mathsf{tuna}), \mathsf{PizzaTopping}(\mathsf{tuna}), \\ &\quad \mathsf{withTop}(\mathsf{tunaPizza}, \mathsf{cheese}), \mathsf{PizzaTopping}(\mathsf{cheese}), \\ &\quad \mathsf{pizzaBase}(\mathsf{plain}), \mathsf{withBase}(\mathsf{tunaPizza}, \mathsf{plain}), \\ &\quad \mathsf{PizzaTopping}(\mathsf{salami})\}\end{aligned}$$

It holds that $\mathcal{T} \uplus \mathcal{F} \models \mathsf{Pizza}(\mathsf{tunaPizza})$. The axiom

$$\exists \mathsf{withBase}.\mathsf{PizzaBase} \sqcap \exists \mathsf{withTop}.\mathsf{PizzaTopping} \sqsubseteq \mathsf{Pizza}$$

would be translated into

$$\neg\mathsf{withBase}(x, y) \vee \neg\mathsf{PizzaBase}(y) \vee \neg\mathsf{withTop}(x, z) \vee \neg\mathsf{PizzaTopping}(z) \vee \mathsf{Pizza}(x)$$

while all facts in $\mathcal{F}$ can immediately be considered as ground clauses. We would then have syntactically relevant axioms

- $\exists \mathsf{withBase}.\mathsf{PizzaBase} \sqcap \exists \mathsf{withTop}.\mathsf{PizzaTopping} \sqsubseteq \mathsf{Pizza}$
- $\mathsf{PizzaBase}(\mathsf{plain})$, $\mathsf{withBase}(\mathsf{tunaPizza}, \mathsf{plain})$

syntactically semi-relevant axioms

- $\mathsf{withTop}(\mathsf{tunaPizza}, \mathsf{tuna}), \mathsf{PizzaTopping}(\mathsf{tuna})$

[1] https://www.snomed.org/
[2] http://geneontology.org/
[3] A similar notion is mentioned also in [BOPP20] for the DL fragment $\mathcal{ELH}^r$

– withTop(tunaPizza, cheese), PizzaTopping(cheese)

and a syntactically irrelevant axiom PizzaTopping(salami).

The refutation in Fig. 4 shows that the PizzaTopping(tuna) is syntactically semi-relevant. It is moreover not relevant since all occurences of the constant tuna can be replaced with cheese resulting in another refutation without PizzaTopping(tuna) (The refutation starts with the only non-ground clause. The shading helps track the inference results).

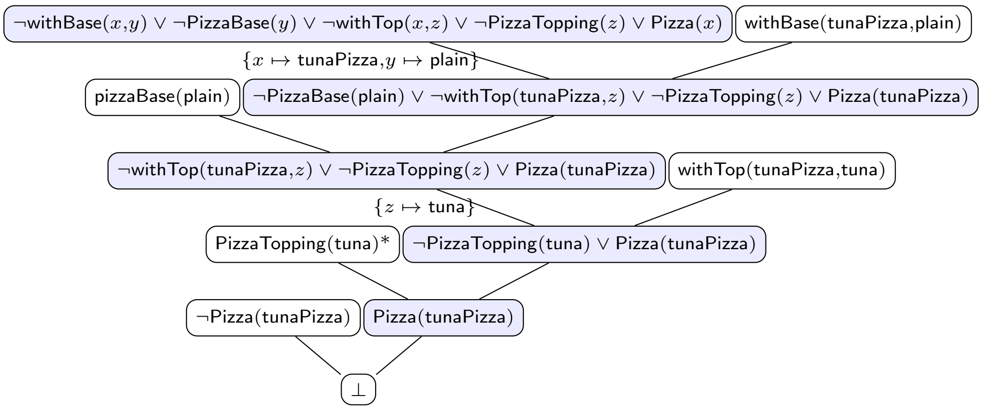

Figure 4: A refutation tree for Pizza(tunaPizza)

Explaining Non-Entailment via Connection-Minimal Abduction

For the task of explaining non-entailment, we focus on a description logic fragment called $\mathcal{EL}$. Abduction is about finding extensions to a knowledge base that are sufficient to imply some given entailment. To avoid undesirable hypotheses, abduction is often equipped with additional restrictions on the solution space and/or minimality criteria that help sort the chaff from the grain.I argue that existing minimality notions suffer from certain limitations, and introduce *connection minimality* as a new notion that overcomes the limitations of earlier notions. Furthermore, we developed and evaluated a method to compute connection-minimal solutions in practice.

Connection-minimality follows Occam's razor in which hypotheses construction do not take into account concept inclusions unrelated the concepts in the observation via $\mathcal{T}$. In other words, for an observation $U_1 \sqsubseteq U_2$, connection minimality only accepts those hypotheses in which every CI in it is "connected" to both U_1 and U_2 in $\mathcal{T}$. The formulation of connection minimality follows the following ideas:

1 Hypotheses for the abduction problem have to create a *connection* between U_1 and U_2, in the form of a concept V that satisfies $\mathcal{T} \cup \mathcal{H} \models U_1 \sqsubseteq V$, $V \sqsubseteq U_2$.

2 To ensure that Occam's razor is followed, we want this connection to be

based on concepts V_1 and V_2 for which we already have $\mathcal{T} \models U_1 \sqsubseteq V_1$ and $\mathcal{T} \models V_2 \sqsubseteq U_2$.

3 We additionally want to make sure that the connecting concepts are not more complex than necessary, and that $\mathcal{H}$ only contains CIs that directly connect parts of V_2 to parts of V_1 by closely following their structure.

We call V a *connecting concept*: A concept V *connects* U_1 to U_2 in $\mathcal{T}$ if and only if $\mathcal{T} \models U_1 \sqsubseteq V$ and $\mathcal{T} \models V \sqsubseteq U_2$. Note that if $\mathcal{T} \models U_1 \sqsubseteq U_2$ then both U_1 and U_2 are connecting concepts from U_1 to U_2, and if $\mathcal{T} \not\models U_1 \sqsubseteq U_2$, it means no concept connects them. In Sect. 4.1.1, the connection between V_1 and V_2 is defined formally using a notion of homomorphism between the *description trees* of V_2 and V_1. We show that this notion of minimality is deeply connected with the generation of prime implicates in first-order logic. That is, using a translation scheme from abduction problems to first-order clauses, we are able to reconstruct the connection-minimal hypotheses using the prime implicates of the translation. In addition to soundness and completeness, we show a quadratic bound on the depth of the terms occurring in the prime implicates, which gives us a termination condition for our method and ensures completeness for hypotheses that are both connection-minimal and subset-minimal.

We implemented a prototype consisting of two components: a Java component that takes care of preprocessing, translation into first-order logic, and construction of the hypotheses from prime implicates, and a first-order reasoning component that uses a modified version of the theorem prover SPASS for the prime implicate generation. The prototype was evaluated on a set of ontologies from the medical domain for which we generated abduction problems in different ways, showcasing the practicality of our approach.

We are focusing here on TBox abduction, where the ontology and hypothesis are TBoxes and the observation is a concept inclusion (CI), i.e., a single TBox axiom.

To illustrate this problem, consider the following TBox, about academia,

$$\begin{aligned}\mathcal{T}_{\mathrm{a}} \quad = \{ \quad & \exists \mathsf{employment.ResearchPosition} \sqcap \exists \mathsf{qualification.Diploma} \sqsubseteq \mathsf{Researcher}, \\ & \exists \mathsf{writes.ResearchPaper} \sqsubseteq \mathsf{Researcher},\ \mathsf{Doctor} \sqsubseteq \exists \mathsf{qualification.PhD}, \\ & \mathsf{Professor} \equiv \mathsf{Doctor} \sqcap \exists \mathsf{employment.Chair}, \\ & \mathsf{FundsProvider} \sqsubseteq \exists \mathsf{writes.GrantApplication} \, \}\end{aligned}$$

that states, in natural language:

- “Anyone with a research position and a diploma must be a researcher.”
- “Anyone who writes a research paper is a researcher.”
- “Being a doctor means having completed a PhD level education.”
- “A professor is similar to a doctor having a (university) chair.”
- “A funds provider writes a grant application.”

We intuitively know that “a professor is a researcher” ($\alpha_{\mathrm{a}} = \mathsf{Professor} \sqsubseteq \mathsf{Researcher}$) but it's in fact missing from $\mathcal{T}_{\mathrm{a}}$. TBox abduction then comes into play to recover this entailment.

There are many ways to recover this entailment. An expert may syntactically restrict the hypotheses to only use a predefined set of abducible axioms [CLM+20, PH17], abducible predicates [KDTS20, Koo21a], or syntactic patterns on of the hypotheses [DWM17a]. This external restriction may also be combined with a minimality criterion relying solely on the knowledge base at hand, such as subset minimality, size minimality, or semantic minimality [COSS13]. Even combined, I argue that these minimality criteria are still in some way insufficient. More specifically, they are not in line with the principle of parsimony, (Occam's razor) in a sense that the resulting hypothesis must involve concept inclusions in the TBox that are not in any way related to the given observation. As an illustration, let us return to the previous academia example. Extending $\mathcal{T}_a$ with the following TBoxes

$$\begin{aligned}\mathcal{H}_{a1} &= \{ \quad \mathsf{Chair} \sqsubseteq \mathsf{ResearchPosition}, \mathsf{PhD} \sqsubseteq \mathsf{Diploma}\} \text{ and} \\ \mathcal{H}_{a2} &= \{ \quad \mathsf{Professor} \sqsubseteq \mathsf{FundsProvider}, \mathsf{GrantApplication} \sqsubseteq \mathsf{ResearchPaper}\}\end{aligned}$$

would complete it to also include $\mathsf{Professor} \sqsubseteq \mathsf{Researcher}$.. Following our intuition, we would prefer $\mathcal{H}_{a1}$ to $\mathcal{H}_{a2}$ as the concept inclusions in it make more sense. A professor in general does not provide funds while a grant application is definitely not a research paper. Nevertheless, both of them are subset-minimal, of equal cardinality, and are not semantically comparable. Thus, both of them are indistinguishable w.r.t. these existing notions.

Let us now look more closely at the concepts in $\mathcal{H}_{a1}$. Here, Chair and $\mathsf{ResearchPosition}$ occur in $\mathcal{T}_a$ in concept inclusions where the concepts in α_a also occur. In addition, both PhD and $\mathsf{Diploma}$ are similarly related to α_a but via the role $\mathsf{qualification}$. In contrast, $\mathcal{H}_{a2}$ involves the concepts $\mathsf{FundsProvider}$ and $\mathsf{GrantApplication}$ that are not related to α_a in any way in $\mathcal{T}_a$. Here, one can pick any concept inclusion $P \sqsubseteq \exists \mathsf{writes}.Q$ to construct a hypothesis similar to $\mathcal{H}_{a2}$ where P replaces $\mathsf{FundsProvider}$ and Q replaces $\mathsf{GrantApplication}$. For example, if we use $\mathsf{Student} \sqsubseteq \exists \mathsf{writes}.\mathsf{Homework}$ (assume it exists in $\mathcal{T}_a$), then we can get the hypothesis $\{\mathsf{Professor} \sqsubseteq \mathsf{Student}, \mathsf{Homework} \sqsubseteq \mathsf{ResearchPaper}$. It is exactly the involvement of such unrelated concept inclusions that makes the hypothesis not parsimonious.

Dissertation Structure

This dissertation is generally divided into three parts. The beginning consists of a preliminary (Ch. 1) and a state-of-the-art chapter (Ch. 2). The second part consists of two contribution chapters describing the new notion of relevance (Ch. 3) and the new notion of minimality for abduction (Ch. 4). The dissertation is finalized by a conclusion also describing future works (Ch. 5).

Chapter 1 is the preliminary chapter explaining the considered logical fragments, some relevant transformations/translations within and between them, and the resolution calculus as the main reasoning engine. The first fragment I focus on is first-order logic (Sect. 1.1). I explain the syntax and semantics in Sect. 1.1.1 and the basic clausification technique for it in Sect. 1.1.2. Second, I describe two description logic fragments ($\mathcal{ALC}$ and $\mathcal{EL}$) in Sect. 1.2. The syntax and semantics are in Sect. 1.2.1 and a normalization technique necessary for the later abduction work in Sect. 1.2.2. Lastly, the resolution calculus along with the set-of-support strategy is described in Sect. 1.3 as the main reasoning engine for both first-order logic and description logic.

Chapter 2 is the state-of-the-art chapter giving a literature review related to entailment explanation (Sect. 2.1) and non-entailment explanation (Sect. 2.2). Both sections are further divided w.r.t. the used key notions considering the parallels to the proposed notions in this dissertation. For the entailment explanation, I consider the notion of proof (Sect. 2.1.1), MUS (Sect. 2.1.1), justification (Sect. 2.1.1), irredundant equivalent subset, and conflict variable (Sect. 2.1.1). For the other part regarding non-entailment, I show the notion of counter example (Sect. 2.2.1), an alternative abduction approach (Sect. 2.2.2), and the common abduction approach (Sect. 2.2.3).

Chapter 3 is the first contribution chapter proposing a new notion of relevance and its test via the resolution calculus with set-of-support strategy. The notion itself is described in Sect. 3.1 consisting of the syntactic relevance (Sect. 3.1.1) and its semantic characterization (Sect. 3.1.2). Afterward, the generalized SOS completeness is proved in Sect. 3.2. This completeness result is used for a test as presented in Sect. 3.3. Last but not least, I also demonstrate its use in description logic (Sect. 3.4).

Chapter 4 is the second contribution chapter proposing a new minimality notion for abduction in the $\mathcal{EL}$ description logic. The proposed notion (connection-minimality) is described in Sect. 4.1. Its computation is done via a first-order prime implicate-based abduction in Sect. 4.2. For this technique, I further elaborate on how to make it more efficient in Sect. 4.3 and additionally derive a termination condition in Sect. 4.4. Efficiency is considered in two levels. First, at the first-order level, some inferences can be avoided (Sect. 4.3.1). Second, at the description logic level, one can perform a set of preprocessing steps to remove some of the unrelated concept inclusions (Sect. 4.3.2). Finally, everything discussed thus far is implemented to perform experimentation as described in Sect. 4.5.

Chapter 5 is the last chapter concluding everything that has been done and explaining the possible future works. This is divided into two parts (entailment and non-entailment) also following the division in the contributions.

Contribution Summary

In the most general setting, this dissertation provides new means of explaining entailment and non-entailment. The former is via a pair of syntactic and semantic relevance while the latter is via abduction with a novel minimality notion called *connection-minimality*. More specifically, the primary contributions are as follows:

- A new notion of syntactic relevance accompanied by a test via resolution with the SOS strategy (which completeness is also generalized in this dissertation) [HTW21], its semantic characterization [HW22] , and a demonstration of how it applies in description logic [HKTW20].
- A new notion of connection-minimality for the $\mathcal{EL}$ description logic via a translation to first-order logic [HKT21a, HKTW22a].

They are not stand-alone contributions and as some necessary companions, I also present procedures, (semi-)decidability results, demonstration in a specific fragment, an implementation, and experiments. Nevertheless, due to the need to prove them and to relate them to other notions, there are many other (not necessarily) smaller contributions which may be interesting on their own and in one way or another.

Some of the other contributions are extensions or generalizations of existing notions/results:

- a generalization of the SOS completeness[4] where the completeness guarantee requiring the satisfiability of the non-SOS clause set [WRC65] is no longer needed and replaced by a more general condition where any clause involved in a refutation can be used in the initial SOS set,
- the notion of a $\mathcal{T}$-homomorphism to characterize entailment w.r.t. the TBox $\mathcal{T}$ generalizing the existing notion from [BKM99] which only deals with subsumption between concepts,
- a notion of conflict literal in first-order logic generalizing the one in propositional logic (e.g. in [JMRS17]),
- a semi-relevance notion generalizing the notion of the propositional lean-kernel (e.g. in [KK09, Kul00]), and
- a dependency notion refining the usual redundancy notion (e.g. in [BR00, Lib05]) via ground instantiation.

Another class of contributions is where the existing notions are used in a slightly different manner. These, in one way or another, give a different outlook in relation to how they are used in the existing works:

- using MUS in first-order clauses via instantiation (different from the works e.g. [MM20]) as an alternative characterization of the proposed semantic relevance,
- using the Herbrand domain for the canonical model [BKM99] (instead of strings built from the symbols of individuals, roles, and concepts as in [BO15]) to create some sort of a universal model for the original $\mathcal{EL}$ axiom set via the positive prime implicates, and
- using an additional renaming technique accompanying a FOL translation (the one used here is from [HS02]) to make some relevant concept inclusions recoverable in $\mathcal{EL}$ even after the non-equivalence-preserving Skolemization step.

Another kind of contribution is where a projection of a more general idea is used in a more specific setting. An example of this is:

- using modularization technique [GHKS08] to make the input for the considered abduction problem smaller while preserving subsumee/subsumers in $\mathcal{EL}$.

To summarize, these are collected in Table 1 and 2 linked to their corresponding locations in this dissertation. The former provides the list of contributions in relation to the notion of relevance while the latter is reserved for connection-minimality notion.

[4]The generalized SOS completeness is a particularly interesting contribution as it non-trivially improves a well-known strategy published in 1965 by Wos et. al. [WRC65].

Table 1: Contribution Summary for Entailment Explanation

Contribution	Reference
Syntactic relevance	Definition 3.1.4
(SOS) deduction/refutation	Definition 3.1.1 and 3.1.2
Relation to resolution derivation	Corollary 3.1.3
Procedure	
Relevance test	Lemma 3.3.1
Semi-relevance test	Lemma 3.3.1 and 3.3.4
(Semi-)decidability for Semi-relevance	Corollary 3.3.3
Syntactic relevance (DL)	Definition 3.4.1
Decidability via bounded model property	Lemma 3.4.2
Semantic relevance	Definition 3.1.11
Conflict literal	Definition 3.1.5
Conflict literal vs ground MUS	Lemma 3.1.8
Conflict literal test	Lemma 3.3.6
(In)dependency	Definition 3.1.10
Propositional case w.r.t. the usual MUS	Lemma 3.1.14
First-order case w.r.t. ground instantiated MUS	Lemma 3.1.15
Syntactic vs semantic relevance	Theorem 3.3.5
Generalized SOS completeness	Theorem 3.2.7

Table 2: Contribution Summary for Non-Entailment Explanation

Contribution	Reference
Connection-minimality	Definition 4.1.3
Entailment via $\mathcal{T}$-homomorphism	Lemma 4.1.6
Computability via prime implicates	Corollary 4.2.11
FOL translation preserving subsumee/subsumer	Section 4.2.1
Universality of the positive prime implicates	Lemma 4.2.4
$\mathcal{EL}$ concept reconstructibility from FOL	
Subsumer from positive prime implicates	Lemma 4.2.5 and 4.2.6
Subsumee from negative prime implicates	Lemma 4.2.9
Decidability via bounded term depth	Theorem 4.4.15
Modularization in $\mathcal{EL}$	
Subsumer-preserving	Lemma 4.3.5
Subsumee-preserving	Lemma 4.3.6
CAPI (implementation)	Section 4.5.1
Evaluation showing its usefulness	
Success rate	Table 4.2
Statistical summary	Table 4.1

The content of this dissertation is based on the papers published during my study. There are three primary conference publications:

– Fajar Haifani, Sophie Tourret, and Christoph Weidenbach. Generalized completeness for SOS resolution and its application to a new notion of relevance. In André Platzer and Geoff Sutcliffe, editors, *Automated Deduction - CADE 28 - 28th International Conference on Automated Deduction, Vir-*

tual Event, July 12-15, 2021, Proceedings, volume 12699 of *Lecture Notes in Computer Science*, pages 327–343. Springer, 2021

– Fajar Haifani and Christoph Weidenbach. Semantic relevance. In *Automated Reasoning - 11th International Joint Conference, IJCAR 2022, Held as Part of the Federated Logic Conference, FloC 2022, Haifa, Israel, August 8-10, 2022, Proceedings*, 2022

– Fajar Haifani, Patrick Koopmann, Sophie Tourret, and Christoph Weidenbach. Connection-minimal abduction in EL via translation to FOL. In *Automated Reasoning - 11th International Joint Conference, IJCAR 2022, Held as Part of the Federated Logic Conference, FloC 2022, Haifa, Israel, August 8-10, 2022, Proceedings*, 2022

The first two introduce syntactic [HTW21] and semantic relevance [HW22] while the last one introduces the connection-minimality notion [HKTW22a]. These are accompanied by the following workshop papers and an arxiv preprint:

– Fajar Haifani, Patrick Koopmann, Sophie Tourret, and Christoph Weidenbach. On a notion of relevance. In Stefan Borgwardt and Thomas Meyer, editors, *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*, volume 2663 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2020

– Fajar Haifani, Patrick Koopmann, and Sophie Tourret. Abduction in EL via translation to FOL. In Renate A. Schmidt, Christoph Wernhard, and Yizheng Zhao, editors, *Proceedings of the Second Workshop on Second-Order Quantifier Elimination and Related Topics (SOQE 2021) associated with the 18th International Conference on Principles of Knowledge Representation and Reasoning (KR 2021), Online Event, November 4, 2021*, volume 3009 of *CEUR Workshop Proceedings*, pages 46–58. CEUR-WS.org, 2021

– Fajar Haifani, Patrick Koopmann, and Sophie Tourret. Introducing connection-minimal abduction for el ontologies. In *XLoKR workshop*, 2021

– Fajar Haifani, Patrick Koopmann, Sophie Tourret, and Christoph Weidenbach. Connection-minimal abduction in el via translation to fol–technical report. *arXiv preprint arXiv:2205.08449*, 2022

The first one [HKTW20] demonstrates how the syntactic relevance is lifted to description logic. The other three concerns abduction. In particular, the second one is a workshop paper showing a partial result about the use of first-order prime implicates for $\mathcal{EL}$ abduction [HKT21a]. The third one is an informal workshop paper introducing the connection-minimality notion without computation [HKT21b]. The last one is an extended technical report for the primary abduction paper where all of the technical proofs are presented [HKTW22b].

Chapter 1

Preliminaries

1.1 First Order Logic

First-order logic (FOL) is an important logic dealing with objects, functions and relations. In this section, I introduce the syntax and semantics of first-order logic along with some basic notions we will use in this dissertation.

1.1.1 Syntax and Semantics

I assume a standard unsorted first-order language over a signature $\Sigma_{\mathrm{FOL}} = (\Omega_{\mathrm{FOL}}, \Pi_{\mathrm{FOL}})$ where Ω_{FOL} is a non-empty set of function symbols, Π_{FOL} a non-empty set of predicate symbols both coming with their respective fixed arities denoted by the function arity.

Definition 1.1.1 (Term, Atom, and Literal)**.** Given a first-order signature $\Sigma_{\mathrm{FOL}} = (\Omega_{\mathrm{FOL}}, \Pi_{\mathrm{FOL}})$, the set of terms over an infinite set of variables $\mathcal{X}$ is denoted by $\mathsf{T}(\Sigma_{\mathrm{FOL}}, \mathcal{X})$ and recursively defined as (i) a variable $x \in \mathcal{X}$ or (ii) $f(t_1, \ldots, t_k)$ with $\mathsf{arity}(f) = k$. An atom A is of the form $P(t_1, \ldots, t_k)$ with $\mathsf{arity}(P) = k$. A literal is either an atom A or its negation $\neg A$. The complement of a literal is denoted by the function comp: for any atom A, $\mathsf{comp}(A) = \neg A$ and $\mathsf{comp}(\neg A) = A$. Terms, atoms, and literals are *ground* if they do not contain any variable, and $\mathsf{T}(\Sigma_{\mathrm{FOL}})$ denotes the set of ground terms. A literal is *positive* (*negative*) if it is of the form A ($\neg A$).

Definition 1.1.2 (Formula and Clause)**.** A formula[1] is either an atom or it takes the following form (suppose that φ and ψ are formulas):

Syntax	Description
$\bot$	false
$\top$	true
$\varphi \wedge \psi$	conjunction
$\varphi \vee \psi$	disjunction
$\neg\varphi$	negation

[1] We will only concern ourselves with closed formulas where variables can only occur below quantifiers.

$\varphi \to \psi$	implication
$\varphi \leftrightarrow \psi$	equivalence
$\forall x.\varphi$	universal quantification
$\exists x.\varphi$	existential quantification

A formula is *function-free* if it contains no function symbols. A clause is a formula of the form $L_1 \vee \ldots \vee L_k$ where L_i are literals for all $1 \leq i \leq k$ and all variables are implicitly universally quantified (the symbol $\forall$ is not written). A clause is *Horn* if it contains at most one positive literal; *definite Horn* if it contains exactly one positive literal; *positive* (*negative*) if it contains only positive (negative) literals; *ground* if it contains only ground literals; *unit* if it contains only a single literal. For convenience, we identify a clause with the multiset of its literals. A set of clauses can then be considered as a conjunction of its clauses.

A, B denote atoms; L, K denote literals; P, Q, R, S, T denote predicates (also propositional variables in the case of propositional logic); t, s denote terms; $f, g, h, \mathtt{sk}$ denote functions where $\mathtt{sk}$ is specifically for a function coming out of Skolemization; a, b, c, d denote constants; and x, y, z denote variables; φ, ψ denote formulas, C, D denote clauses; M, N, O denote clause sets; Φ denotes a set of formulas; all possibly annotated. As a clause is a specific kind of formula, Φ, φ, and ψ may also be used for clauses.

By means of *substitution*, variables in clauses can be replaced by any terms in particular when dealing with clauses.

Definition 1.1.3 (Substitution)**.** Substitutions σ, τ are total mappings from variables to terms, where $\mathsf{dom}(\sigma) := \{x \mid x\sigma \neq x\}$ is finite and $\mathsf{codom}(\sigma) := \{t \mid x\sigma = t, x \in \mathsf{dom}(\sigma)\}$. A *renaming* σ is a bijective substitution with codomain in $\mathcal{X}$. It can be written as pairs $\{x_1 \mapsto t_1, \ldots, x_k \mapsto t_k\}$ and used in a postfix notation $x\sigma$ when applied. In this case, $x\sigma$ is called an *instantiation* of x. The application of substitutions is extended to literals, clauses, and sets/sequences of such objects in the usual way.

The semantics of these formulas is defined in terms of interpretations and variable assignments.

Definition 1.1.4 (Interpretation)**.** Given the signature $\Sigma_{\mathrm{FOL}} = (\Omega_{\mathrm{FOL}}, \Pi_{\mathrm{FOL}})$, an *interpretation* is a tuple $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ made of a non-empty *domain* $\Delta^{\mathcal{I}}$ and an *interpretation function* $\cdot^{\mathcal{I}}$ which assigns

- a total function $f^{\mathcal{I}} : \underbrace{\Delta^{\mathcal{I}} \times \ldots \times \Delta^{\mathcal{I}}}_{n} \mapsto \Delta^{\mathcal{I}}$ s.t. $\mathsf{arity}(f) = n$ for all $f \in \Omega_{\mathrm{FOL}}$,
- a relation $P^{\mathcal{I}} \subseteq \underbrace{\Delta^{\mathcal{I}} \times \ldots \times \Delta^{\mathcal{I}}}_{n}$ to every $P \in \Omega_{\mathrm{FOL}}$ s.t. $\mathsf{arity}(P) = n$.

An interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ is a Herbrand Interpretation if

- $\Delta^{\mathcal{I}} = \mathsf{T}(\Sigma_{\mathrm{FOL}})$,
- $f^{\mathcal{I}} : (t_1, \ldots, t_k) \mapsto f(t_1, \ldots, t_k)$ with $\mathsf{arity}(f) = k$ for all $f \in \Omega$ and $t_1, \ldots, t_k$ are ground.

A Herbrand Interpretation $\mathcal{I}$ can also be uniquely represented as a set of ground atoms[2] satisfying

$$P(t_1, \ldots, t_k) \in \mathcal{I} \text{ if and only if } (t_1, \ldots, t_k) \in P^{\mathcal{I}}.$$

Definition 1.1.5 (Variable Assignments)**.** Given an interpretation $\mathcal{I}$, a valuation β is a total function $\mathcal{X} \to \Delta^{\mathcal{I}}$. For a given valuation β, the assignment of some variable x can be modified to $a \in \Delta^{\mathcal{I}}$ by $\beta[x \mapsto a]$. It is extended to terms $\mathcal{I}(\beta) : \mathsf{T}(\Sigma_{\text{FOL}}, \mathcal{X}) \to \Delta^{\mathcal{I}}$ with

(i) $\mathcal{I}(\beta)(x) = \beta(x)$, where $x \in \mathcal{X}$ and

(ii) $\mathcal{I}(\beta)(f(t_1, \ldots, t_n)) = f^{\mathcal{I}}(A(\beta)(t_1), \ldots, A(\beta)(t_n))$, where $f \in \Omega_{\text{FOL}}$, $\mathsf{arity}(f) = n$.

Definition 1.1.6 (Semantics)**.** Interpretations and variable assignments are extended to formulas as follows.

$$
\begin{array}{rcl}
\mathcal{I}(\beta)(\bot) & := & 0 \\
\mathcal{I}(\beta)(\top) & := & 1 \\
\mathcal{I}(\beta)(P(t_1, \ldots, t_n)) & := & 1 \text{ if } (\mathcal{I}(\beta)(t_1), \ldots, \mathcal{I}(\beta)(t_n)) \in P^{\mathcal{I}} \text{ and 0 otherwise} \\
\mathcal{I}(\beta)(\neg\varphi) & := & 1 - \mathcal{I}(\beta)(\varphi) \\
\mathcal{I}(\beta)(\varphi \wedge \psi) & := & \min(\mathcal{I}(\beta)(\varphi), \mathcal{I}(\beta)(\psi)) \\
\mathcal{I}(\beta)(\varphi \vee \psi) & := & \max(\mathcal{I}(\beta)(\varphi), \mathcal{I}(\beta)(\psi)) \\
\mathcal{I}(\beta)(\varphi \to \psi) & := & \max(1 - \mathcal{I}(\beta)(\varphi), \mathcal{I}(\beta)(\psi)) \\
\mathcal{I}(\beta)(\varphi \leftrightarrow \psi) & := & 1 \text{ if } \mathcal{I}(\beta)(\varphi) = \mathcal{I}(\beta)(\psi) \text{ and 0 otherwise} \\
\mathcal{I}(\beta)(\exists x.\varphi) & := & 1 \text{ if } \mathcal{I}(\beta[x \mapsto a])(\varphi) = 1 \text{ for some } a \in \Delta^{\mathcal{I}} \text{ and 0 otherwise} \\
\mathcal{I}(\beta)(\forall x.\varphi) & := & 1 \text{ if } \mathcal{I}(\beta[x \mapsto a])(\varphi) = 1 \text{ for all } a \in \Delta^{\mathcal{I}} \text{ and 0 otherwise}
\end{array}
$$

A formula φ is *satisfiable* if there is an interpretation $\mathcal{I}$ such that $\mathcal{I}(\beta)(\varphi) = 1$ for some variable assignment β. In this case, $\mathcal{I}$ is called a *model* of φ, which is denoted as $\mathcal{I} \models \varphi$. Otherwise, φ is called *unsatisfiable* or *inconsistent*. We can also say that $\mathcal{I}$ *satisfies* φ. In the case of a set of clauses N, then it is satisfiable if there is $\mathcal{I}$ which satisfies all of the clauses in it (also similarly written as $\mathcal{I} \models N$).

Given two formulas (or clauses) φ and ψ, φ *entails* ψ ($\varphi \models \psi$), if for any interpretation $\mathcal{I}$, if $\mathcal{I} \models \varphi$ then $\mathcal{I} \models \psi$. The formulas φ and ψ are *equivalent* ($\varphi \leftrightarrow \psi$), if $\varphi \models \psi$ and $\psi \models \varphi$. The formulas φ and ψ are called *equisatisfiable*, when φ is satisfiable if and only if ψ is satisfiable (not necessarily in the same models). Given two clauses C and D, if there exists a substitution σ such that $C\sigma \subseteq D$ then we say that C *subsumes* D (D is *subsumed* by C). In this case $C \models D$.

The notions of entailment, equivalence and equisatisfiability are naturally extended to sets of formulas by considering them as conjunctions of formulas: Given formula sets M_1 and M_2, $M_1 \models M_2$, if for any interpretation $\mathcal{I}$, if $\mathcal{I} \models \varphi$ for every $\varphi \in M_1$ then $\mathcal{I} \models \psi$ for every $\psi \in M_2$. The sets M_1 and M_2 are equivalent, written $M_1 \leftrightarrow M_2$, if $M_1 \models M_2$ and $M_2 \models M_1$. Given an arbitrary formula φ and a formula set M, $M \models \varphi$ is written to denote $M \models \{\varphi\}$; analogously, $\varphi \models M$ stands for $\{\varphi\} \models M$. It has many names in the literature[3] (MUS) if any strict subset of N is satisfiable [PY84].

[2]We abuse notations by using the letter $\mathcal{I}$ to represent a Herbrand interpretation and its unique associated set. The context will be given when necessary.

[3]It has been called differently in the literature (e.g. minimally unsatisfiable subset, minimal unsatisfiable core, etc.). Here, I use "set" to avoid confusion because in this dissertation, MUSes can be acquired via instantiation.

Given a clause set N, a clause C is an *implicate* of N, if $N \models C$; C is a *prime implicate* of N, if for any other implicate D of N s.t. $D \models C$, it also holds that $C \models D$.

A model is *finite* if it has a finite domain and *bounded* if the size of the domain has an upper bound w.r.t. the input formula (e.g. exponential or doubly exponential). A fragment of first-order logic has a *finite (resp. bounded) model property* if for any formula φ in this language if φ is satisfiable then it has a finite (resp. bounded) model. Boundedness here is stronger than finiteness in the sense that if a model is bounded, then it is also finite.

An unsatisfiable clause set (possibly infinite) enjoys the property of being refutable from a finite subset of it. This is called the compactness theorem.

Theorem 1.1.7 (FOL Compactness [Fit90])**.** *Let N be a set of clauses in first-order logic. Then N is unsatisfiable if and only if there is a finite subset $N' \subseteq N$ such that N' is unsatisfiable.*

An important fragment of FOL which will be considered in this dissertation is *propositional logic*. The difference is that only predicates with arity 0 are allowed. Atoms are written without brackets (i.e. simply P instead of $P()$) and called *propositional variables*. Functions can simply be considered empty. No formulas are quantified and thus substitutions are also not needed. Its semantics also naturally follows FOL but simplified.

1.1.2 Clausification

All notions proposed in this dissertation are applicable to a family of logic fragments called description logics via a suitable translation scheme. A key aspect of the translation is that it still produces non-clausal formulas while the calculus I use operates on sets of clauses (also called *conjunctive normal form*). So, in this section, I show a well-known satisfiability preserving clausification technique [NW01]. Bringing the results back from first-order logic to description logic will be some of the non-trivial parts of our contributions.

The clausification works first by transforming a formula into a prenex normal form and by *Skolemization.* A formula is in a *prenex normal form* if it takes the shape of $\mathcal{Q}_1 x_1 \ldots \mathcal{Q}_k x_k.\varphi$ where φ is quantifier free and $\mathcal{Q}_i \in \{\exists, \forall\}$ for all $1 \leq i \leq k$. In other words, all quantifiers occur in the prefix of the formula . Once a formula is in prenex normal form, a simple Skolemization proceeds by picking the outer-most existential quantifier $\exists x$ and replacing all occurrences of x with a fresh function symbol `sk` taking all variables that x depends on as arguments. One Skolemization step would transform a formula (here, φ may still contain quantifiers) as follows:

$$\forall x_1, \ldots, \forall x_k \exists x.\varphi \longrightarrow_{\mathtt{sk}} \forall x_1, \ldots, \forall x_k.\varphi[\mathtt{sk}(x_1, \ldots, x_k)/x].$$

This must be done until all existential quantifiers are removed.

The number of arguments in the fresh Skolem functions is reducible depending on the syntactical characteristics of the formula. One technique is by a rewriting rule that chooses variables as arguments in a more effective manner with the help of *negation normal form* transformation, *miniscoping*, and *variable renaming* [NW01]. I will not present it here since the need for Skolemization in this dissertation is rather simple and this optimization would have no effects in our context anyway. Moreover, care must be taken when performing any reasoning tasks involving clause generation with some optimized reasoning tools. This is because there is a strong Skolemization technique [NRW98](which is still satisfiability preserving) which non-trivially affects the way clauses are generated. In SPASS, such technique is turned on by default and has to be explicitly disabled to support our reasoning need.

Clausification is described in Table 1.1. At this point, the existential quantifiers are already eliminated and any application of the rules does not reintroduce any existential quantifier. Therefore, we get a set of clauses where all variables are universally quantified once the rules are exhaustively applied.

Table 1.1: Clausification rules

$(\varphi \leftrightarrow \psi)$	$\longrightarrow_{cl}$	$(\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi)$
$(\varphi \rightarrow \psi)$	$\longrightarrow_{cl}$	$(\neg\varphi \vee \psi)$
$\neg(\varphi \vee \psi)$	$\longrightarrow_{cl}$	$(\neg\varphi \wedge \neg\psi)$
$\neg(\varphi \wedge \psi)$	$\longrightarrow_{cl}$	$(\neg\varphi \vee \neg\psi)$
$\neg\neg\varphi$	$\longrightarrow_{cl}$	φ
$(\varphi \wedge \psi) \vee \phi$	$\longrightarrow_{cl}$	$(\varphi \vee \phi) \wedge (\psi \vee \phi)$
$(\varphi \wedge \top)$	$\longrightarrow_{cl}$	φ
$(\varphi \wedge \bot)$	$\longrightarrow_{cl}$	$\bot$
$(\varphi \vee \top)$	$\longrightarrow_{cl}$	$\bot$
$(\varphi \vee \bot)$	$\longrightarrow_{cl}$	φ

The rules in Table 1.1 are the basic ones and there are in fact optimized techniques [NRW98, NW01] to reduce the number of generated clauses (e.g. renaming technique). However, similar to the fact that only simple Skolemization technique is needed, this clausification approach is sufficient for this dissertation.

1.2 Description Logic

Two description logic fragments of interest in this dissertation are $\mathcal{ALC}$ and $\mathcal{EL}$, which are both translatable for FOL. $\mathcal{ALC}$ is the basic description logic which is rather expressive despite being intractable for most common tasks in description logic. $\mathcal{EL}$ is a lightweight fragment of $\mathcal{ALC}$ admitting more efficient procedures for many of these tasks. $\mathcal{EL}$ is also widely used in the industry despite its simplicity. Our notion of relevance, which we later introduce, applies to both $\mathcal{EL}$ and $\mathcal{ALC}$, as well as any DL fragments translatable to FOL.

1.2.1 Syntax and Semantics

I now describe the syntax and semantics of $\mathcal{ALC}$ and $\mathcal{EL}$ description logic [BHLS17]. The signature in DL is a pair $\Sigma_{\text{DL}} = (\Omega_{\mathsf{C}}, \Omega_{\mathsf{R}})$ where Ω_{C} and Ω_{R} are pair-wise disjoint, countably infinite sets of *atomic concepts* and *roles*, respectively. The choice of symbols in the signature is according to how they are translated into first-order language: we use letters P, Q, S, T for atomic concepts; U, V, W for possibly complex concepts; R for roles[4]; a, b, c, d for individuals; $\mathcal{M}$ for modules

[4]In most DL papers, roles are usually written in lowercase but we choose uppercase for compatibility with our FOL notations

and all are possibly annotated. $\mathcal{ALC}$ concepts are built according to the syntax rule

$$U \ ::= \ \top \ | \ \bot \ | \ U \ | \ \neg U \ | \ U \sqcap U \ | \ U \sqcup U \ | \ \exists R.U \ | \ \forall R.U$$

where, in $\mathcal{EL}$, $\bot$, $\neg U$, $U \sqcup U$, and $\forall R.U$ are excluded.

An ontology $\mathcal{O}$ is a pair $(\mathcal{T}, \mathcal{F})$ where the *TBox* $\mathcal{T}$ is a finite set of *concept inclusions* (CIs) of the form $U \sqsubseteq V$. $U \equiv V$ stands for $U \sqsubseteq V$ and $V \sqsubseteq U$. The *Abox* $\mathcal{F}$ consists of ground truth called *instance assertion* of the form $U(a)$ (U is an atomic concept in $\mathcal{EL}$) and $R(a, b)$. The concept inclusion $\sqsubseteq$ here can also be called *subsumption* between concepts similar to the other literatures where, if $U \sqsubseteq V$ holds, we also say that V *subsumes* U, U *is subsumed by* V, U is a *subsumee* of V, and V is a *subsumer* of U. Note that this *subsume* terminology is an overloading as it was already defined for clauses. A concept inclusion or an instance assertion may also be called a *DL axiom*, a and b *individuals* that correspond to constants in first-order logic. While there are other syntactic features that can be added to have more expressivity (e.g. number restrictions, nominals, etc.) concepts, roles and individuals remain the basic ingredients existing in most description logic fragments. Another important aspect of $\mathcal{EL}$ TBox is that it may contain cycles. Here we distinguish two types of them[5] as follows.

Definition 1.2.1 (Cycle)**.** Given an $\mathcal{EL}$ TBox $\mathcal{T}$, an atomic concept P is a concept with

- *forward cycle* if $\mathcal{T} \models P \sqsubseteq W$, and
- *backward cycle* if $\mathcal{T} \models W \sqsubseteq P$

in $\mathcal{T}$, where P occurs under some role restriction in some W. $\mathcal{T}$ is cyclic if there is a concept with either a forward cycle or a backward cycle.

The distinction between forward and backward cycles is useful in the context of our abduction task because, as we will see later, only the existence of both cycles which may cause a termination issue.

Many DL fragments are expressible in first-order logic via a suitable semantic-preserving translation. Furthermore, they have a close relationship with certain modal logic. Table 1.2, following a modal logic-related translation scheme in [HS02], shows the translation of $\mathcal{ALC}$ to first-order logic where the definitional form from [HS02] is left out as it is not needed in this dissertation. A translation of some DL ontology is the conjunction of all axioms in the TBox and ABox after translation. The function `fo` is used to similarly translate concepts, TBox, ABox, and the whole ontology. For simplicity, its clausification is omitted. Note that I will primarily use clausification and Skolemization and use the symbol `sk` specifically for functions resulting from the Skolemization. For this, Π_{S} denotes the set Skolem functions resulting from such translation (with no Skolemization, all formulas are in fact function-free). The signature of the resulting FOL clauses would then be $(\Omega_{\mathsf{C}} \uplus \Omega_{\mathsf{R}}, \Pi_{\mathsf{S}})$ (The symbols for atomic concepts and roles are immediately used as predicate symbols while functions only come from Skolemization).

[5]In the literature (e.g. [BHLS17]), cycle is usually defined for Tbox containing only definitional axioms of the form $P \equiv W$. A Tbox $\mathcal{T}$ is cyclic if there is $P \equiv W \in \mathcal{T}$ where P occurs in W even transitively.

Table 1.2: $\mathcal{ALC}$ to First-Order Translation

Concepts	
$\texttt{fo}(\top, x) = \top$	$\texttt{fo}(\bot, x) = \bot$
$\texttt{fo}(P, x) = P(x)$	$\texttt{fo}(\neg U, x) = \neg\, \texttt{fo}(U, x)$
$\texttt{fo}(U \sqcap V, x) = \texttt{fo}(U, x) \wedge \texttt{fo}(V, x)$	$\texttt{fo}(U \sqcup V, x) = \texttt{fo}(U, x) \vee \texttt{fo}(V, x)$
$\texttt{fo}(\forall R.U, x) = \forall y.(R(x, y) \to \texttt{fo}(U, x))$	$\texttt{fo}(\exists R.U, x) = \exists y.(R(x, y) \wedge \texttt{fo}(U, x))$
TBox Axioms	
$\texttt{fo}(U \sqsubseteq V) = \forall x.(\texttt{fo}(U, x) \to \texttt{fo}(V, x))$	
$\texttt{fo}(U \equiv V) = \forall x.(\texttt{fo}(U, x) \leftrightarrow \texttt{fo}(V, x))$	
ABox Axioms	
$\texttt{fo}(U(a)) = \texttt{fo}(U, a)$	
$\texttt{fo}(R(a, b)) = R(a, b)$	

Since the semantics of this description logic and its translation coincide [Bor94, BCM$^+$03], it is possible to consult Table 1.2 for the DL semantics by looking at the FOL translation. Nevertheless, as we will work more heavily with $\mathcal{EL}$, I will show its semantics for the axioms as mostly described in the literature. An interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ maps atomic concepts $P \in \Omega_{\mathsf{C}}$ to sets $P^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ and roles $R \in \Omega_{\mathsf{R}}$ to relations $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. The interpretation function $\cdot^{\mathcal{I}}$ is extended to complex concepts as follows:

$$\top^{\mathcal{I}} = \Delta^{\mathcal{I}} \qquad (U \sqcap V)^{\mathcal{I}} = U^{\mathcal{I}} \cap V^{\mathcal{I}}$$
$$(\exists R.U)^{\mathcal{I}} = \{d \in \Delta^{\mathcal{I}} \mid \exists (d, e) \in R^{\mathcal{I}} \text{ s.t. } e \in U^{\mathcal{I}}\}.$$

It holds that $\mathcal{I} \models U \sqsubseteq V$, if $U^{\mathcal{I}} \subseteq V^{\mathcal{I}}$.

All semantical notions except for (prime) implicates and MUS are lifted from FOL as well. These include entailment, equivalence, equisatisfiability, and bounded/finite model property [6]. Nevertheless, there exist analogous notions of prime implicate and MUS in description logics. For prime implicates in DL, one can simply refer to [Bie07] since only the notion of prime implicate in FOL is used in this dissertation. A notion analogous to that of a MUS in description logic is called *justification*: Given a TBox $\mathcal{T}$ and a DL axiom α s.t. $\mathcal{T} \models \alpha$, $\mathcal{T}' \subseteq \mathcal{T}$ is a justification for α if $\mathcal{T}' \models \alpha$ but no strict subset of $\mathcal{T}'$ entails α.

1.2.2 Normalization

A TBox can be normalized to simplify the shape of the concept inclusions while preserving entailment of concept inclusions in the former signature [BN03, BHLS17]. This can often help us prove certain properties more conveniently. In $\mathcal{EL}$, a *normalized* TBox would have only *normalized concept inclusions* of the following shape.

$$P \sqsubseteq Q \qquad P_1 \sqcap P_2 \sqsubseteq Q \qquad \exists R.P \sqsubseteq Q \qquad P \sqsubseteq \exists R.Q$$

Table 1.3 shows a normalization technique for $\mathcal{EL}$ where U' and V' are strictly complex concepts, Q is an existing atomic concept, P is a fresh atomic concept,

[6] $\mathcal{ALC}$ is an example of a DL fragment possessing the bounded model property

and U and V are any concepts. It only takes a linear number of these applications until a TBox is normalized.

Table 1.3: Normalization rules for $\mathcal{EL}$

$U' \sqsubseteq V'$	$\longrightarrow_{nr}$	$U' \sqsubseteq P, P \sqsubseteq V'$
$U \sqcap V' \sqsubseteq Q$	$\longrightarrow_{nr}$	$V' \sqsubseteq P, U \sqcap P \sqsubseteq Q$
$V' \sqcap U \sqsubseteq Q$	$\longrightarrow_{nr}$	$V' \sqsubseteq P, P \sqcap U \sqsubseteq Q$
$\exists R.V' \sqsubseteq Q$	$\longrightarrow_{nr}$	$V' \sqsubseteq P, \exists R.P \sqsubseteq Q$
$Q \sqsubseteq \exists R.V'$	$\longrightarrow_{nr}$	$P \sqsubseteq V', Q \sqsubseteq \exists R.P$
$Q \sqsubseteq U \sqcap V$	$\longrightarrow_{nr}$	$Q \sqsubseteq U, Q \sqsubseteq V$

This results in a *conservative extension* of the original TBox. That is, an entailed concept inclusion in the original signature will remain entailed after normalization.

Proposition 1.2.2 (Normalization [BN03, BHLS17])**.** If an $\mathcal{EL}$ TBox $\mathcal{T}'$ is obtained from $\mathcal{T}$ by applying one of the rules of Table 1.3, then $\mathcal{T}' \models \mathcal{T}$ and $\mathcal{T}'$ is a conservative extension of $\mathcal{T}$.

This normalization technique helps in many aspects of our abduction work. In the following lemma, we show another property of conservative extensions useful to turn abductive solutions of a normalized TBox to solutions for the original TBox without the fresh concepts.

Lemma 1.2.3. For every $\mathcal{EL}$ TBox $\mathcal{T}$, we can compute in polynomial time an $\mathcal{EL}$ TBox $\mathcal{T}'$ in normal form such that for every other TBox $\mathcal{H}$ and every CI $U \sqsubseteq V$ that use only names occurring in $\mathcal{T}$, we have $\mathcal{T} \cup \mathcal{H} \models U \sqsubseteq V$ iff $\mathcal{T}' \cup \mathcal{H} \models U \sqsubseteq V$.

Proof. Based on Prop. 1.2.2, we have

1. $\mathcal{T}' \models \mathcal{T}$, and
2. for every model $\mathcal{I}$ of $\mathcal{T}$, there exists a model $\mathcal{I}'$ of $\mathcal{T}'$ s.t. for every concept name P occurring in $\mathcal{T}$, $P^{\mathcal{I}} = P^{\mathcal{I}'}$, and for every role name $R \in \Omega_{\mathsf{R}}$ occurring in $\mathcal{T}$, $R^{\mathcal{I}} = R^{\mathcal{I}'}$.

Now let $\mathcal{H}$ be a TBox and $U \sqsubseteq V$ a CI such that both only use names occurring in $\mathcal{T}$. If $\mathcal{T} \cup \mathcal{H} \models U \sqsubseteq V$, we observe that by Item 1, we have $\mathcal{T}' \cup \mathcal{H} \models \mathcal{T} \cup \mathcal{H}$, and thus by transitivity of entailment, $\mathcal{T}' \cup \mathcal{H} \models U \sqsubseteq V$. Assume $\mathcal{T} \cup \mathcal{H} \not\models U \sqsubseteq V$. Then there exists a model $\mathcal{I}$ of $\mathcal{T} \cup \mathcal{H}$ s.t. $\mathcal{I} \not\models U \sqsubseteq V$. Since $\mathcal{H}$, U and V only use names occurring in $\mathcal{T}$, by Item 2, we can find a model $\mathcal{I}'$ of $\mathcal{T}'$ s.t. $\mathcal{I}' \models \mathcal{H}$ and $\mathcal{I}' \not\models U \sqsubseteq V$, and consequently, $\mathcal{T}' \cup \mathcal{H} \not\models U \sqsubseteq V$. We obtain that $\mathcal{T} \cup \mathcal{H} \models U \sqsubseteq V$ iff $\mathcal{T}' \cup \mathcal{H} \models U \sqsubseteq V$. □

.

1.3 Deduction using Resolution and SOS Strategy

I introduce the usual resolution calculus for clause sets and the set-of-support strategy. This will serve as the primary reasoning engine for both first-order logic and description logic (by means of translation). The calculus consists of two inference rules: Resolution and Factoring [Rob65, RV01]. The rules operate on a state (N, M) 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 M the initial state is (N, M). I describe the rules in the form of abstract rewrite rules operating on states (N, M). As usual, we assume for the resolution rule that the involved clauses are variable-disjoint. This can always be achieved by substituting similar variables in different clauses.

Two terms t_1 and t_2 are unifiable if there exists a substitution σ so that $t_1\sigma = t_2\sigma$. The substitution σ is then called a *unifier* of t_1 and t_2. The unifier σ is called *most general unifier*, written $\sigma = \mathsf{mgu}(t_1, t_2)$, if any other unifier σ' of t_1 and t_2 can be represented as $\sigma' = \sigma\sigma''$, for some substitution σ''.

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

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

The clause $(D \vee C)\sigma$ is called the result of a *Resolution inference* between its parents. The clause $(C \vee L)\sigma$ is called the result of a *Factoring inference* of its parent. The parents can also be called *premises*. The result of an inference is also called *resolvent*. A sequence of rule applications $(N, M) \Rightarrow^*_{RES} (N, M')$ is called a *resolution derivation*[7]. It is called an *(SOS) resolution derivation* if $N \neq \emptyset$. In case $\bot \in M'$ it is called a *(SOS) resolution refutation*. A derivation is *linear* when the resolvent of one inference is always a premise of the next inference. A clause $D \in N$ is *redundant* in N if there is a resolution derivation from $N \setminus \{D\}$ to some C s.t. C subsumes D (and irredundant otherwise).

First-order logic is not decidable. Nevertheless, resolution calculus has been shown to be sound and *refutationally complete*. That is, given an unsatisfiable clause set M, there is a finite sequence of the rule applications $(\emptyset, M) \Rightarrow^*_{RES} (\emptyset, M')$ s.t. $\bot \in M'$. Moreover, by adding the restriction that N is satisfiable in the initial pair (N, M), the same also holds true for SOS resolution.

Theorem 1.3.1 (Soundness and Refutational Completeness of (SOS) Resolution [Rob65, WRC65])**.** Resolution is sound and refutationally complete [Rob65]. If for some clause set N and initial SOS M, N is satisfiable and $N \cup M$ is unsatisfiable, then there is a derivation of $\bot$ from (N, M) [WRC65].

In more restrictive settings, the set-of-support strategy can still be complete for the clause set N with SOS M. For example, if N is saturated by superposition and does not contain the empty clause (which cannot happen for satisfiable N), then it is also complete under the strong superposition inference restrictions [BG94].

An important property of the resolution calculus for our deductive reasoning task is that for any entailed clause, a subsuming clause can always be derived.

Theorem 1.3.2 (Deductive Completeness of Resolution [NdW95, Lee67])**.** 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.

[7]When the context is clear, one can use *derivation* as well

Chapter 2

State-of-the-Art

There has been a lot of work in regard to explaining entailment and non-entailment. In this section, I will give a brief summary in particular for two fragments: first-order logic and description logic (and in related fragments when relevant). Bear in mind that there are other works in answer set programming, logic programming, non-monotonic reasoning, modal logic, etc. and different areas such as machine learning and NLP have been useful to help with clause set explanation such as finding a shorter proof or relevant axioms (interested readers may refer to, for example, [RS21]) but I consider these to be beyond the scope of this dissertation. In contrast, the works I included here have not necessarily stated that they are for clause set explanations. This is to be expected since a notion may often be useful for a lot of other things such as debugging and repair. The aim here is to collect works that can be put into the perspective of clause set explanation having parallels with the notions proposed in this dissertation.

2.1 Entailment Explanation

In general, I categorize works related to entailment explanation into four domains: relevancy, quality measure, abstraction, and presentation. Relevancy deals with choosing clauses to use for a deduction, the quality measure provides notions of preference on deductions, abstraction deals with intermediate lemmas, and presentation deals with different ways a proof can be delivered to users.

I will primarily focus on relevancy because it is the one related to the first contribution of this dissertation. In addition, one could also argue that inconsistency degree deserves a separate section but I put it here due to it being related to the proposed relevancy notion in this dissertation. So, it must be kept in mind that, despite being brief in this section, the other ones do not necessarily lack interesting references.

2.1.1 Relevancy

A relevancy notion deals with what clauses/axioms to use to show an entailment. Existing works related to relevancy rely on proofs or a notion of a minimal set (e.g. MUS, justification, and irredundant equivalent clause set). Clauses are often defined in terms of whether they are in all of the considered sets/proofs, in at least one of the set/proof, or not at all.

Proof

A clause used in all refutations, in our terminology, is a syntactically relevant clause. In first-order logic, it can be checked via the classical set-of-support strategy [WRC65]. This is more well-known than our notion of syntactic semi-relevance and it can alternatively be defined via MUS, justification (in DL), or irredundant equivalent subset.

A clause used in some refutation, in our terminology, is syntactically semi-relevant clause. In propositional logic, it is called a *usable clause* in [KLM06] and defined to be in the *plain clause set* in [KK09]. Usable clauses outside of any MUSes are semantically superfluous (which is not the case in our first-order setting). In [BOPP20], a result that resembles our notion of semi-relevance is presented (but also not in full first-order logic). DL axioms are classified w.r.t. how they are used in the possible first-order refutations via a translation scheme.

MUS

The following notion is comparable to our notion (they even coincide for clause sets with no redundant clauses). This is because characterizing clause relevance via MUS in propositional logic makes sense since a clause outside of any MUSes would either be irrelevant or redundant.

Definition 2.1.1 (MUS-based Relevance [KLM06])**.** Given an unsatisfiable clause set N in propositional logic, a clause $C \in N$ is

- *necessary* if it occurs in all MUSes,
- *potentially necessary* if it occurs in some MUS,
- *never necessary* if it is not in any MUS.

A different but related notion has also been applied for propositional abduction [EG95].

The use of MUS has also been explored in first-order logic [MM20]. There, the formulation is more general: The given set of clauses N is divided into $N = N' \uplus N''$ where N' is a *non-relaxable* clause set while N'' is a *relaxable* clause set which must be satisfiable. a MUS is defined as a subset M of N'' s.t. $N' \uplus M$ is unsatisfiable but removing a clause from M would render it satisfiable. In this setting, this definition does not strictly conform to ours.

Computation MUS is apparently predominant in propositional logic [PY84, Kul00, KK09, MKIM19, KLM06]. An example of the earliest works regarding complexity is the problem of determining whether a clause set is a MUS. It was proven in [PW88] to be D^p-complete[1] for a propositional clause set with at most three literals per clause and at most three occurrences of each propositional variable .

[1] A language is in D^p if it is in the intersection of a language in NP and a language in coNP [PY84]

There are also some works in satisfiability modulo theory (SMT) [ZXZ+11, GST16, CGS07, CGS11]. A deletion-based approach well-known in propositional logic has also been lifted for MUS extraction in SMT [GST16]. In [CGS07, CGS11], the approach is to combine an SMT solver with some external propositional core extractor. Recursive clause removals using depth-first-search on some graph representation of the subformulas may also serve as an alternative approach [ZXZ+11].

For the function-free and equality-free first-order fragment FEF, there is a "decompose-merge" approach to compute all MUSes [XL16, LL18].

Other works include counting [BM21] and computing union [MKIM19], and enumeration [BK16, AMM15].

Justification

In description logic, a notion that is related to MUS is called justification usually identified via axiom pinpointing [KPHS07, SC03, BP10, BOPP20]. The definition is similar in spirit but normally used for an entailment problem instead of unsatisfiability: justification is basically a minimal axiom set entailing some given DL axiom α. It is in fact loosely related to MUS. For example, if we have $\{P_1 \sqsubseteq P_2 \sqcap P_3, P_2 \sqsubseteq P_4\} \models P_1 \sqsubseteq P_4$ the set $\{P_1 \sqsubseteq P_2 \sqcap P_3, P_2 \sqsubseteq P_4\}$ is a justification but when reformulated in FOL as an unsatisfiable set, we get $\{\neg P_1(x) \vee P_2(x), \neg P_1(x) \vee P_3(x), \neg P_2(x) \vee P_4(x), P_1(\mathtt{sk}_0), \neg P_4(\mathtt{sk}_0)\}$ which is actually not a MUS due to $P_1(x) \vee P_3(x)$.

In [CMPY22], the set of axioms occurring in all justifications and the ones in a justification are distinguished. Both are in some way related to our notion of relevant and semi-relevant clauses respectively. The test of whether an axiom exists in some justification is also discussed in [PS17].

Justifications are in some way related to the so-called pinpointing formula (which is a useful alternative to describe the relation between inferences the considered entailment). Given an entailment α and an ontology $\mathcal{O}$, a *pinpointing formula* is a monotone propositional formula φ over the axioms in $\mathcal{O}$ (as propositional symbols) s.t. every model of the formula entails α, and the set of justifications corresponds exactly to the set of minimal models of φ. Thus, we can use pinpointing formulas to generate justifications. There are methods to compute pinpointing formulas by intercepting reasoning procedures, as, for instance, done for $\mathcal{EL}$ in [BPS07]. Pinpointing formulas are in some way related to provenance semirings in database access [GKT07], recently investigated also in the context of DLs [BO19]. Here, a notion of *provenance polynomial* formula is useful to link a query result to the database entries involved in its computation.

Computation In general, there are two approaches to compute justifications: black box and white box. On the one hand, a black-box approach interacts with an external reasoner but only looks into the input and output [KPHS07]. On the other hand, white box approach immediately modifies the internal workings of a reasoner (e.g. Tableau [BP10, SC03]). For the sake of efficiency, the computation of the *lean kernel* can approximate the union of them [PMIM17].

Irredundant Equivalent Subset

In propositional logic, there exists a notion based on the semantic equivalence [Lib05].

Definition 2.1.2 (Equivalence-based relevance)**.** Given a clause set N in propositional logic, a clause C is

- *necessary*, if it is in all irredundant $N' \subseteq N$ s.t. $N' \leftrightarrow N$;
- *useful*, if it is in some irredundant $N' \subseteq N$ s.t. $N' \leftrightarrow N$; and
- *useless*, if it is not in any irredundant $N' \subseteq N$ s.t. $N' \leftrightarrow N$.

where a clause C is redundant in N if $N \setminus \{C\} \models C$ (and irredundant otherwise)

This notion does not impose the clause set to be unsatisfiable. In the case of unsatisfiable clause set, it coincides with the previous MUS-based relevance (if N is unsatisfiable, then the given N' would be unsatisfiable and irredundancy restriction excludes any clause outside of any MUSes from N'). An approach to compute a minimal irredundant equivalent subset is discussed in [BJLM12].

Inconsistency Degree

Inconsistency degree measures how inconsistent a propositional clause set is. In [HK10], a more general set of axioms that must be satisfied for such degrees is defined. It also provides examples such as the number of MUSes or simply the drastic measure where a clause set gets the measure of 1 when unsatisfiable and 0 otherwise. Despite that works in this direction often do not define some relevancy notion explicitly, this can in fact be useful. For example, also in [HK10], a clause that does not contribute to inconsistency is called free formula (related to the proposed irrelevant clause in this dissertation). A clause set with an inconsistency degree of zero is satisfiable (related in some way to the proposed relevance clauses in this dissertation).

A lot of works proposed different ways to define such measure, but due to its resemblance to the proposed notion of conflict literal in Def. 3.1.5, I specifically focus on a notion of inconsistency degree via the so-called *conflicting variables* from [JMRS17].

Definition 2.1.3 (Conflicting Variable)**.** Given an unsatisfiable propositional clause set N, an atom P is a *conflicting variable* if there exist two satisfiable sets $N', N'' \subseteq N$ s.t.

- $N' \models P$,
- $N'' \models \neg P$
- $N' \cup N''$ is a MUS

The existence of such variables implies unsatisfiability and vice versa.

Lemma 2.1.4 (Conflicting Variables for Unsatisfiability [JMRS17])**.** Given a propositional clause set N, N is inconsistent if and only if N has at least one conflicting variable.

The *inconsistency degree* of an unsatisfiable clause set in propositional logic is the ratio between the number of conflicting variables and the number of all variables. analogous notions have also been defined in description logic, for example in [MQHL07].

2.1.2 Quality Measure

In propositional logic, there has been some interest in finding small refutations. It has been shown that the size of a refutation can be exponential [Hak85] and finding a short refutation is intractable [Iwa97, ABMP98].

In [ABB$^+$21], the notion of proof quality measure is generalized via a class of measures called the *recursive quality measure*. This applies to a wide variety of measures (defined on some calculi) such as size [ABB$^+$20a, ABB$^+$20b], depth, and tree width (if proofs are assumed to be in tree shape).

2.1.3 Abstraction

Works in this domain provide intermediate abstraction or lemmatization to cover users from tedious details in proofs [Hua94, KUV15]. Protégé [KKS17], a DL tool, can also perform some lemmatization when visualizing a proof from a justification.

2.1.4 Presentation

This line of work aims towards transforming or delivering proof that is more easily understandable to users. One way to present proofs is via a linear format. A proof here has also been translated into a natural language (e.g. english). The work in [Cos96] uses functional proof representation.

Another approach is via a non-linear graphical representation. For this, a number of DL tools are available: Evonne [ABB$^+$22], Protégé [KKS17], and ELK [KKS14].

In [Pfe84], it is argued that natural deduction proofs are more comprehensible than refutation proofs and proposed a transformation technique. A notion of proof redirection [Bla13] has also been proposed to output a natural deduction proof with additional features such as case analysis and nested subproof.

Cognitive science has been used also to take into account user cognition [Fie05]. In [DHS06], resolution is used as a basis for an explanation of a DL proof. It works in three steps: translation, finding a refutation, and explanation by traversal of its graph representation.

2.2 Non-Entailment Explanation

I classify three ways of explaining a non-entailment. This is primarily due to [Koo21b] for description logic but I also consider the model-based approach as separate (because it is apparently not uncommon in the FOL literature and that this immediately exploits the definition of non-entailment). The first one tries to find a *counter-example* that holds for N but not for φ. The second one tries to find an extension of the ontology s.t. any model of this extension is not the model of the observation. Last, abduction is in essence finding some other axiom set N' s.t. $N \uplus N' \models \varphi$. The first approach using counter-example has an apparent weakness because models may contain too many parts not useful for the explanation while the second one is reducible to abduction. The one that is rather predominant (and comparatively relevant to our contribution) in the literature is actually abduction. So, in the following, I only present the first two briefly and focus more on abduction.

2.2.1 Counter Example

Suppose we have a non-entailment $N \not\models \neg\mathsf{Cat}(x) \vee \mathsf{Dog}(x)$ where the domain consists of some finite set of animal breeds, habitats, and preys. We can explain this by finding a cat that is not a dog for example $\mathsf{Cat}(\mathsf{bobTail})$ but $\neg\mathsf{Dog}(\mathsf{bobTail})$. One that can be used is the readily available notion of interpretation. We try to find a model $\mathcal{I}$ of N that is not a model of α. This is the most straightforward approach as it directly exploits the definition of an entailment. However, this could be problematic if only models with an infinite domain are available. Even with fragments satisfying the bounded model property such as $\mathcal{EL}$ and $\mathcal{ALC}$, a model can be very large and may contain information that is not necessarily useful. For example, information regarding the habitat of $\mathsf{bobTail}$ may be forced to exist by N for a model of N satisfying $\mathsf{Cat}(\mathsf{bobTail})$ even though, e.g., cats and dogs in N share the same habitat, and thus habitat is not a distinguishing factor.

Finding a finite model (if it exists at all) for a counter-example can be done using a model finder tool. Because $N \not\models \phi$ iff $N \uplus \{\neg\phi\}$ is satisfiable, a model of $N \uplus \{\neg\phi\}$ certifies the non-entailment. There are many that one can use in first-order logic. A few of them are Alloy Analyzer [Jac06], KodKod [TJ07], Paradox [CS03], Mace4 [McC03], and SEM [ZZ95], etc. Interested reader can refer to [ZZ13] for more. In description logic, we can use Tweezer [WP07], SuperModel [BSP09], and Protégé[2] [KKS17, BB07].

2.2.2 Extension with Models not Satisfying the Observation

The idea of this approach is to extend the given background knowledge to prevent the addition of any axioms which could make the observation entailed. This approach is only recently introduced in the workshop paper [Koo21b]. This can in some way be reduced to an abduction problem. There, it is argued that, the other approaches may be too restrictive due to the open-world assumption (unknown axioms are not necessarily false if they do not contradict the existing knowledge). Due to the lack of first-order literature, I explain this specifically for description logic: given an ontology $\mathcal{O}$ and a DL axiom α, we try to find another ontology $\mathcal{H}$ s.t.

(i) $\mathcal{O} \uplus \mathcal{H}$ is consistent, and

(ii) for any interpretation $\mathcal{I}$, if $\mathcal{I} \models \mathcal{O} \uplus \mathcal{H}$ then $\mathcal{I} \not\models \alpha$.

Interested readers may also consult this work for a bit of an outlook for how it may be done via ABox abduction along with its challenges.

[2]with an additional plugin invoking the first-order tool Mace4

2.2.3 Abduction

Abduction is an approach to explaining a non-entailment that is rather prevalent in the literature. Its formulation may come in different flavors [EG95] but I only focus on the logic-based abduction where we try to find an extension of a clause set to make it entail some observation. The number of hypotheses for an abduction problem can be huge and some minimality notions have also been defined. In [EG95] the following notions are discussed: subset minimality, size minimality, priority functions, and penalization function. One way to compute the hypotheses is via prime implicates [Pop73]. This is in essence a resolution-based approach. As alternatives, tableau and sequent calculi have also been considered [MP93] where, different from the resolution calculus, they do not require the initial clausification.

In description logic, abduction problems may also be formalized differently for various purposes. Concept abduction tries to find the subsumees of an atomic concept w.r.t. a TBox [Bie08, CNS$^+$04]. Relaxed abduction problems define multiple observations in which not all of them need to be entailed [Hub16]. Here, the preference regarding which one to entail is restricted via some notion of partial ordering. Abduction where the observation is a single axiom (which I will focus more on) has also attracted a lot of research (e.g., TBox abduction [DWM17b, WDL14]).

There are also other abduction works in first-order related fragments worth mentioning: first-order modulo theories [EPS18], ground equational logic [Tou16], modal logic [MP95]. In fact, the works in [EPS18, Tou16] were initially the ones considered for the prime implicate-based abduction technique to be exploited for the connection-minimality notion. Even though this dissertation eventually generated the prime implicates using the resolution calculus as a white box, these works may still be reconsidered for a computation technique. In contrast, the work in [MP95] may also be interesting due to the close relationship between $\mathcal{ALC}$ and modal logic.

In the following two sections, I will only elaborate more on prime-implicate based abduction in FOL while TBox abduction along with its relevant minimality notions will be the focus of the description logic paragraphs. This is simply due to their importance in our proposed notion of abduction.

First-Order Abduction

While this is not the only problem formalism, the kind of first-order abduction considered here is as follows: given a set of clauses N and another clause ψ s.t. $N \not\models \psi$, abduction tries to find another set of clauses N' s.t. $N \uplus N' \models \psi$. As pointed out in [Pop73], we can rely on deduction. This is the case with the resolution calculus previously introduced. If it holds that $N \not\models \psi$ and we have $\varphi = L_1 \vee \cdots \vee L_k$ as a prime implicate of $N \uplus \{\neg\psi\}$, then $N' = \{\mathsf{comp}(L_1) \wedge \cdots \wedge \mathsf{comp}(L_k)\}$ would be our abductive solution because of the following equivalence.

$$N \uplus \{\neg\psi\} \models \varphi \text{ iff } N \uplus \{\neg\varphi\} \models \psi$$

In other words, there is some sort of duality between abduction and deduction. Some abduction mechanisms using the resolution calculus have been based on this dualism [Mar91, Pop73, CP86, Pop73]. Here, φ is a result of deductive reasoning from $N \uplus \{\mathsf{comp}(\psi)\}$. The key property that any abductive solution must satisfy is that its extension $N \uplus N'$ must be consistent.

DL Abduction

The kind of abduction under consideration in this dissertation is where we try to find an extension of an axiom set to make it entail some observation. I distinguish four kinds of basic abduction problems that are more closely related our setting: Concept abduction [Bie08, CNS+04], ABox abduction [Koo21a, DS19, CLM+20, PH20, PH17, COSS13, DWS14, DQSP12, HB12, KES11], TBox abduction [DWM17b, WDL14] and knowledge base abduction [KDTS20, EKS06].

Definition 2.2.1 (Basic DL abduction problem)**.** The following are the basic abduction problems in DL.

- (*Concept Abduction*) Given an ontology $\mathcal{O}$, a set of atomic concepts $\Sigma \subseteq \Omega_{\mathsf{C}}$, and an atomic concept Q, find $\{P_1, \ldots, P_n\} \subseteq \Sigma$ s.t. $\mathcal{O} \models P_1 \sqcap \cdots \sqcap P_n \sqsubseteq Q$.
- (*ABox Abduction*) Given an ontology $\mathcal{O}$, and an ABox axiom α s.t. $\mathcal{O} \not\models \alpha$, find a set of ABox assertion $\mathcal{H}$ s.t. $\mathcal{O} \uplus \mathcal{H} \models \alpha$.
- (*TBox Abduction*) Given a TBox $\mathcal{T}$, and a concept inclusion $U_1 \sqsubseteq U_2$ s.t. $\mathcal{T} \not\models U_1 \sqsubseteq U_2$, find a set of concept inclusions $\mathcal{H}$ s.t. $\mathcal{O} \uplus \mathcal{H} \models U_1 \sqsubseteq U_2$.
- (*Ontology Abduction*) Given an ontology $\mathcal{O}$ and either a TBox or an ABox axiom α s.t. $\mathcal{O} \not\models \alpha$, find another ontology $\mathcal{H}$ s.t. $\mathcal{O} \uplus \mathcal{H} \models \alpha$.

Here, $\mathcal{H}$ is called a *hypothesis*.

The concept abduction does not explicitly state some non-entailment condition, but this can still be formulated as one is first-order: $\mathcal{O} \models P_1 \sqcap \cdots \sqcap P_n \sqsubseteq Q$ iff $\mathtt{fo}(\mathcal{O}) \wedge \neg Q(\mathtt{sk}_0) \models \neg P_1(\mathtt{sk}_0) \vee \ldots \vee \neg P_n(\mathtt{sk}_0)$ and thus $\mathtt{fo}(\mathcal{O}) \not\models Q(\mathtt{sk}_0)$ with $\mathtt{fo}$ translates $\mathcal{O}$ to equivalent first-order formulas and $\mathtt{sk}_0$ is a constant. From this, we get the following abduction problem: given a non-entailment $\mathtt{fo}(\mathcal{O}) \not\models Q(\mathtt{sk}_0)$ find subset-minimal set of atoms $\{P_1(\mathtt{sk}_0), \ldots, P_n(\mathtt{sk}_0)\}$ s.t. when added to $\mathtt{fo}(\mathcal{O})$, we get the wanted entailment. Another variation of concept abduction is defined in [CNS+04]: Given a non-entailment $\mathcal{O} \not\models U_1 \sqsubseteq U_2$, find another concept V s.t. $\mathcal{O} \models U_1 \sqcap V \sqsubseteq U_2$. One can easily look into the definitions that TBox and ABox abductions are special cases of ontology abduction. In the following, I will focus more on TBox abduction.

The number of hypotheses can potentially be very large or even infinite. Many ways to select them have been devised. I classify these approaches into two sets. The first one takes into account some expert knowledge while the second provides a means of saying "this hypothesis is preferred to that hypothesis" only via properties intrinsic to the abduction problem itself. It is not a strict classification since one can (and usually does) mix and match them together.

Expert knowledge may come in various forms. Signature restriction for expressive DL is used in [KDTS20] (in conjunction with semantic minimality introduced later). In fact, TBox abduction is usually accompanied by a restriction that one can only use simple concept inclusions of the form $P \sqsubseteq Q$ with atomic concepts P and Q. This immediately turns the solution space into a finite set. In [DWM17b, HLP10], a pattern-based restriction from which the hypotheses should be instantiable is defined. An oracle function [WDL14] can also be used to simply limit the possible axioms in the hypotheses.

Definition 2.2.2 (Hypotheses selection with expert knowledge)**.** Given a TBox $\mathcal{T}$ and a concept inclusion $U_1 \sqsubseteq U_2$ s.t. $\mathcal{T} \not\models U_1 \sqsubseteq U_2$, a hypothesis $\mathcal{H}$ can be accepted via the following restriction:

(i) *Signature* [KDTS20]: if $\mathcal{H}$ is built over some predefined signature $\Sigma \subseteq \Omega_{\mathsf{C}} \uplus \Omega_{\mathsf{R}}$

(ii) *Justification pattern* [DWM17b]: if $\mathcal{H}$ is instantiated from some *justification pattern* (i.e. a set of TBox axioms where all atomic concepts and roles in it are considered variables, disjoint from $\Omega_{\mathsf{C}} \uplus \Omega_{\mathsf{R}}$, and replaceable injectively only by atomic concepts and roles in Ω_{C} and Ω_{R}).

(iii) *Oracle function* [WDL14]: if axioms in $\mathcal{H}$ may only be taken from some predefined set.

The second approach does not rely on any external information. In essence, all of them provide some sort of means to compare hypotheses with each other.

Definition 2.2.3 (Hypotheses selection via internal properties [WDL14])**.** Given a TBox $\mathcal{T}$, and a concept inclusion $U_1 \sqsubseteq U_2$ s.t. $\mathcal{T} \not\models U_1 \sqsubseteq U_2$, the hypothesis satisfies

(i) *subset minimality* if there is no other hypothesis $\mathcal{H}'$ s.t. $\mathcal{H}' \subsetneq \mathcal{H}$

(ii) *size minimality* if there is no other hypothesis $\mathcal{H}'$ s.t. $|\mathcal{H}'| < |\mathcal{H}|$

(iii) *semantic maximality* if there is no other hypothesis $\mathcal{H}'$ s.t. $\mathcal{T} \uplus \mathcal{H}' \models \mathcal{H}$

(iv) *semantic minimality* if there is no other hypothesis $\mathcal{H}'$ s.t. $\mathcal{T} \uplus \mathcal{H} \models \mathcal{H}'$

(v) *weak semantic minimality* if there is no other hypothesis $\mathcal{H}'$ s.t. $\mathcal{H} \models \mathcal{H}'$

Subset minimality removes axioms from a hypothesis if it preserves the resulting entailment. Size minimality prefers hypotheses with a smaller size. Semantic minimality prefers something that is semantically closest to the observation and in particular, must generally be accompanied by some other ways to restrict the solutions (e.g. with signature restriction [DS17]) to avoid the trivial hypothesis $\{U_1 \sqsubseteq U_2\}$). Semantic maximality (the opposite of semantic minimality) prefers something more "informative": in a more general sense of abduction, if $\mathcal{T} \uplus \mathcal{H}' \models \mathcal{H}$, then $\mathcal{H}'$ is also an abductive explanation for $\mathcal{H}$. $\mathcal{H}'$ is therefore more informative than $\mathcal{H}$. In [WDL14], there are three different ways of combining semantic maximality and subset minimality (*minmax*, *maxmin*, and *skyline optimal*) which will not be explained further here.

In general, there is really no consensus regarding what constitutes "good" hypotheses. The easiest one to notice is semantic minimality versus semantic maximality because they are directly in opposition to each other. (Weak) semantic minimality is also in some way in opposition with the subset minimality because if $\mathcal{H} \subsetneq \mathcal{H}'$ then $\mathcal{H} \models \mathcal{H}$ and also $\mathcal{T}, \mathcal{H} \models \mathcal{H}$ for any $\mathcal{T}$. The following Ex. 2.2.4 shows how the previous minimality notions may disagree with each other. Here, $\mathcal{H}_1$ is preferred to $\mathcal{H}_2$ w.r.t. (i), (ii), and (iii) while $\mathcal{H}_2$ is preferred to $\mathcal{H}_1$ w.r.t. (iv) and (v). Without signature restriction, semantic minimality here even accepts only $\mathcal{H}_3$. One might wonder why $\mathcal{H}_2$ may even be accepted in the first place. This is due to the way $\mathcal{T}$ is used. $\mathcal{H}_1$ may be generated by axioms in $\mathcal{T}$ involving R_2 while $\mathcal{H}_2$ involves R_1.

Example 2.2.4 (Existing minimality notions)**.** Given

$$\mathcal{T} = \{U_1 \sqsubseteq P,\ U_1 \sqsubseteq \exists R_1.Q, U_1 \sqsubseteq \exists R_2.P,$$
$$S \sqcap \exists R_1.T \sqsubseteq U_2, \exists R_2.S \sqsubseteq U_2\}$$

s.t. $\mathcal{T} \not\models U_1 \sqsubseteq U_2$. The following are three possible hypotheses.

$$\mathcal{H}_1 = \{P \sqsubseteq S\} \qquad \mathcal{H}_2 = \{P \sqsubseteq S, Q \sqsubseteq T\} \qquad \mathcal{H}_3 = \{U_1 \sqsubseteq U_2\}$$

Chapter 3

Explaining Entailment via a new Notion of Relevance

I introduce a notion of syntactic relevance relying on refutations along with semantic relevance which can serve as its semantic characterization. The semantic relevance makes use of an alternative notion called conflict literals to describe unsatisfiability. In addition, the need for a test of these relevance notions leads to a generalization for the completeness of the set-of-support strategy.

Syntactic Relevance

I distinguish clauses into the ones that are necessary for any refutations called *syntactically relevant*, from clauses that are useful called *syntactically semi-relevant*, and from clauses that are not needed at all called *syntactically irrelevant*. For an illustration, consider the following unsatisfiable clause set N with Fig. 3.1 showing a refutation of N in a tree form.

$$
\begin{aligned}
N = \{&(1) : P(f(a)) \vee S(x_3),\\
&(2) : \neg S(x_7),\\
&(3) : \neg Q(c{,}a) \vee Q(b{,}f(x_6)),\\
&(4) : Q(x_1{,}x_2) \vee R(x_1),\\
&(5) : \neg R(x_5),\\
&(6) : \neg P(x_4) \vee \neg Q(b{,}x_4)\}
\end{aligned}
$$

This means all of the clauses in N are syntactically semi-relevant. Unlike the other syntactically semi-relevant clauses, clause (3) is not relevant because there exists another refutation without (3) depicted in Fig. 3.2.

In propositional logic, it is sufficient to consider MUSes to explain unsatisfiability on the original clause level, Lemma 3.1.14. However, it is not the case with first-order logic. $N \setminus \{(3) : \neg Q(c, a) \vee Q(b, f(x_6))\}$ is a MUS, for which Fig. 3.2 shows a refutation from these clauses. Clause $(3) : \neg Q(c, a) \vee Q(b, f(x_6))$ is not in any MUS but I argue that this should not be disregarded. This is because the refutation involving (3) uses a different instantiation for the other clauses than the one without (3). In the refutation of Fig. 3.1 clause $(5) : \neg R(x_5)$ is

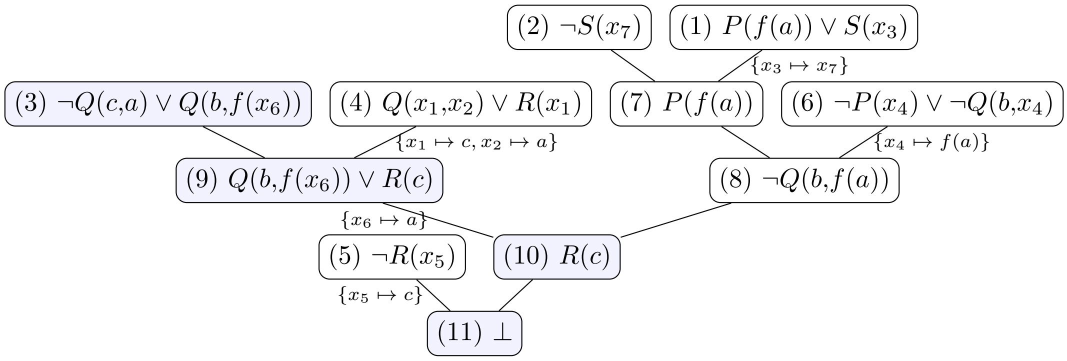

Figure 3.1: A refutation of N depicted as a tree

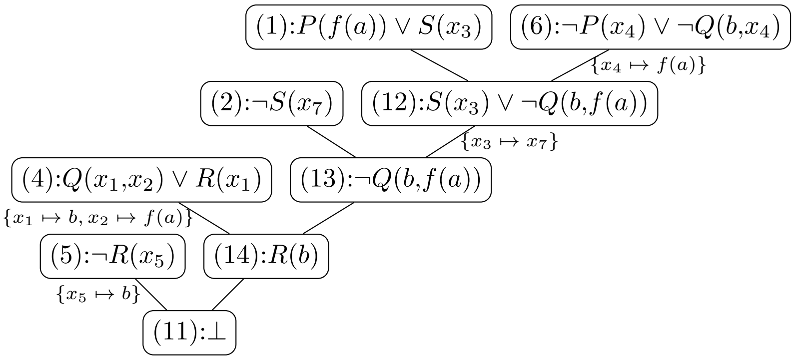

Figure 3.2: A refutation of N without $(3) : \neg Q(c, a) \vee Q(b, f(x_6))$

instantiated with $\{x_5 \mapsto c\}$ where in the refutation of Fig. 3.1 it is instantiated with $\{x_5 \mapsto b\}$. This means that the two refutations are different and clause $(3) : \neg Q(c, a) \vee Q(b, f(x_6))$ should also be taken into account.

Semantic Relevance

As a companion to the syntactic relevance, I also propose a semantic relevance based on the notion 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 $N_2 \models \mathsf{comp}(L)$. I argue that relying on conflict literals is useful for two reasons. First, explaining that a clause set is unsatisfiable via the absence of a model is not very helpful since there is nothing to deal with in the first place. Second, the notion of MUS (as we have seen before) can only partially explain an entailment. In some sense, a conflict literal provides a middle ground to explain how a clause relates to unsatisfiability between the lack of models and MUSes. It also better reflects our intuition that there is a contradiction (in the form of two simple ground consequences that cannot be both true at the same time) in an unsatisfiable set of clauses.

From Fig. 3.1, it can be deduced that the literals $R(c)$ and $\neg R(c)$ are conflict literals because

$$\begin{aligned} N \setminus \{(5) : \neg R(x)\} &\models R(c) \\ \{(5) : \neg R(x)\} &\models \neg R(c) \end{aligned}$$

where both $\{(5) : \neg R(x)\}$ and $N \setminus \{(5) : \neg R(x)\}$ are satisfiable. From the refutation in Fig. 3.2, one can see that $(5) : \neg R(x)$ is syntactically relevant due to $N \setminus \{(3) : \neg Q(c, a) \vee Q(b, f(x_6))\}$ being a MUS. We will also show that in a ground MUS, any ground literal in it is a conflict literal, Lemma 3.1.15. In the example, it is still possible to identify the conflict literals by looking into the ground MUS constructed from the instantiations of the refutations in Fig. 3.1 and Fig. 3.2. This leads to the following conflict literals for N, see Def. 3.1.5:

$$\begin{aligned} \mathsf{conflict}(N) = \{&(\neg)P(f(a)), \\ &(\neg)Q(b, f(a)), (\neg)Q(c, a), \\ &(\neg)R(b), (\neg)R(c)\} \quad \cup \\ \{&(\neg)S(t) \mid t \text{ is a ground term}\} \end{aligned}$$

We can acquire these conflict literals by applying the substitutions in the refutations from Fig. 3.1 and Fig. 3.2 towards the input clauses. They correspond to two first-order MUSes M_1 and M_2. Here, if a literal is ground then it is a conflict literal. The other conflict literals can be obtained by grounding the variables.

$$\begin{aligned} M_1 = \{&(5) : \neg R(c), (2) : \neg S(x_7), \\ &(1) : P(f(a)) \vee S(x_3), \\ &(3) : \neg Q(c, a) \vee Q(b, f(a)), \\ &(4) : Q(c, a) \vee R(c), \\ &(6) : \neg P(f((a))) \vee \neg Q(b, f(a))\} \\ M_2 = \{&(5) : \neg R(b), \\ &(4) : Q(b, f(a)), (2) : \neg S(x_7), \\ &(1) : P(f(a)) \vee S(x_3), \\ &(6) : \neg P(f(a)) \vee \neg Q(b, f(a))\} \end{aligned}$$

Here, even though $(3) : \neg Q(c, a) \vee Q(b, f(x_6))$ is not in 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 $(\neg)Q(c, a)$ depends on clause (3). Note that in general, determining conflict literals is not obvious since we do not necessarily know beforehand which ground substitution to apply. In addition, it is possible that the number of such ground MUSes is not finite and its size is unbounded.

Based on the notion of conflict literals, I introduce a notion of semantic relevance, Def. 3.1.11 which may also characterize the syntactic relevance. Because redundant clauses may induce unexpected behaviors, we additionally impose a refinement of redundancy upon the clause set at the first-order level. This is the notion of independency: 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.

Relevant clauses are the ones whose removal does not preserve the unsatisfiability. Irrelevant clauses can be freely identified once we know the semi-relevant ones. The more interesting case is the semi-relevant clauses. For the running example, $(3) : \neg Q(c, a) \vee Q(b, f(x_6))$ is semi-relevant because it is necessary for the conflict literals $(\neg)R(c)$ and $(\neg)Q(c, a)$. More specifically, the set of conflicts for $N \setminus \{(3) : \neg Q(c, a) \vee Q(b, f(x_6))\}$ does not include $(\neg)R(c)$ and $(\neg)Q(c, a)$:

$$\begin{aligned}\mathsf{conflict}(N \setminus \{(3) : \neg Q(c, a) \vee Q(b, f(x_6))\}) = \{&(\neg)P(f(a)), (\neg)Q(b, f(a)), (\neg)R(b)\} \\ &\uplus\{(\neg)S(t) | t \text{ is a ground term}\}\end{aligned}$$

These conflict literals can be seen from M_2: Suppose 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, $P(f(a)) \in \mathsf{conflict}(N \setminus \{\neg Q(c, a) \vee Q(b, f(x_6))\})$, and define

$$\begin{aligned} N_\emptyset &= \{R \in M_2 | P(f(a)) \notin R \text{ and } \neg P(f(a)) \notin R\} \\ &= \{(5) : \neg R(b), (4) : Q(b, f(a)), (2) : \neg S(t)\} \\ N_{P(f(a))} &= \{R \in M_2 | P(f(a)) \in R\} \\ &= \{(1) : P(f(a)) \vee S(t)\} \\ N_{\neg P(f(a))} &= \{R \in M_2 | \neg P(f(a)) \in R\} \\ &= \{(6) : \neg P(f(a)) \vee \neg Q(b, f(a))\} \end{aligned}$$

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

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

because $P(f(a))$ can be acquired using resolution between (1) and (2) for $N_\emptyset \cup N_{P(f(a))}$ and $\neg P(f(a))$ can be acquired using resolution between (4) and (6) for $N_\emptyset \cup N_{\neg P(f(a))}$. We can similarly show that the other ground literals are also conflict literals.

Generalized SOS Strategy

An SOS refutation is a refutation where there is initially a dedicated non-empty clause set called SOS in which the inferences always involve at least one clause in the SOS and put the resulting clause back in it. So, the refutation in Fig. 3.1 is not an SOS refutation from the syntactically semi-relevant clause $(3) : \neg Q(c,a) \vee Q(b,f(x_6))$, because only the shaded part represents an SOS refutation starting with this clause. More specifically, there are two inferences ending in $(8) : \neg Q(b,f(a))$ which violates the condition for an SOS refutation. Nevertheless, it can be transformed into an SOS refutation where clause $(3) : \neg Q(c,a) \vee Q(b,f(x_6))$ is in the SOS, Fig. 3.3. Please note that $N \setminus \{(3) : \neg Q(c, a) \vee Q(b, f(x_6))\}$ is still unsatisfiable and classical SOS completeness [WRC65] is not sufficient to guarantee the existence of a refutation with SOS $\{(3) : \neg Q(c,a) \vee Q(b,f(x_6))\}$. In fact, I describe how the transformation can be done in Sect. 3.2 to prove Th. 3.2.7, which is a generalization of the SOS strategy that is useful to test for (syntactic) semi-relevancy.

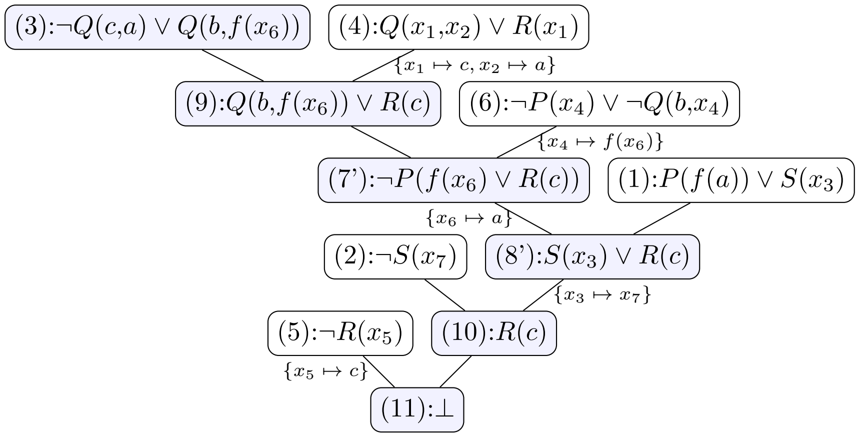

Figure 3.3: Semi-relevant clause $(3) : \neg Q(c, a) \vee Q(b, f(x_6))$ in the SOS

3.1 A New Notion of Relevance

I now introduce the syntactic notion of relevance and its characterization formally. This will be done first by introducing the necessary notions. The notion of refutation is used for syntactic relevance while its semantic characterization relies on conflict literals and dependency.

3.1.1 Syntactic Relevance

Before we jump into the formal definition, one thing that must be clear is that calculus is basically a procedure to test for unsatisfiability. As such, in a derivation, a clause not related to the final derived clause may be generated. For our notion of deduction and refutation, we do not need such clauses. I call a compact form of derivation a deduction where any unrelated clauses are omitted.

Definition 3.1.1 (Deduction)**.** 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 $i, j < k$.

The subscript N in π_N can be omitted 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 for the latter subset relation we identify sequences with multisets. A deduction $\pi_N = [C_1, \ldots, C_{n-1}, \bot]$ is called a *refutation*.

We assume that deductions are variable disjoint and take the form of trees. Variable disjointness can be achieved via variable renamings while one can recall that this notion of a deduction implies a tree structure. Both assumptions allow for the existence of overall grounding subsitutions. A grounding of an overall substitution τ of some deduction π is a substitution $\tau\delta$ such that $\mathsf{codom}(\tau\delta)$ only contains ground terms and $\mathsf{dom}(\delta)$ is exactly the variables from $\mathsf{codom}(\tau)$.

Definition 3.1.2 (SOS Deduction)**.** A deduction $\pi_{N \cup M} = [C_1, \ldots, C_n]$ is called an *SOS deduction* with SOS M, if the derivation $(N, M_0) \Rightarrow^*_{\text{RES}} (N, M_m)$ is an SOS derivation where $C'_1, \ldots, C'_m$ is the subsequence from $[C_1, \ldots, C_n]$ with input clauses removed, $M_0 = M$, and $M_{i+1} = M_i \cup M'_{i+1}$.

Corollary 3.1.3 (Deduction Refutations versus Resolution Refutations)**.** There exists a resolution refutation $(N, M) \Rightarrow^*_{\text{RES}} (N, M' \cup \{\bot\})$ if and only if there exists a deduction refutation $\pi_{(N \cup M)} = [C_1, \ldots, C_{n-1}, \bot]$ where $C_i \in (N \cup M')$ for all i, modulo variable renaming.

On the one hand, a resolution derivation $(N, M) \Rightarrow^*_{\text{RES}} (N, M')$ shows a sequence of inferences deriving new clauses from (N, M) not necessarily useful to derive $\bot$. On the other hand, a deduction minimally focuses on the derivation of a single clause, e.g., the empty clause $\bot$ in case of a refutation. In deductions, every clause is assumed to be used exactly once (thus the tree shape). This is a purely technical restriction, see Corollary 3.1.3, that enables the deduction transformation technique to not worry about of variable renamings beyond the input clauses. More specifically, if the refutation is not in tree form, then there must be a clause that is used only used more than once. We can then simply duplicate such clause and use a different variable set for each duplicate. Corresponding variables may be substituted differently depending on the later clauses using it. For example, if $P(x_1) \lor Q(x_1)$ is resolved with $\neg P(a) \lor R(x_2)$ and $\neg P(b)$, then we can add a new clause $P(x_1) \lor Q(x_1)$. s.t. x_1 is substituted with a while x_2 is substituted by b. In the tree form, clauses are then used only once (even though some clauses were originally from a single clause).

Definition 3.1.4 (Syntactic Relevance)**.** Given an unsatisfiable set of clauses N, a clause $C \in N$ is *syntactically relevant* if for all deduction refutations π of N it holds that $C \in \pi$. A clause $C \in N$ is *syntactically semi-relevant* if there exists a deduction refutation π of N in which $C \in \pi$. A clause $C \in N$ is *syntactically irrelevant* if there is no deduction refutation π of N in which $C \in \pi$.

3.1.2 Semantic Characterization

In this section, I present a semantic relevance that also serves as a semantic characterization for the previous notion of syntactic relevance. As we will see later, this also illustrates that the previous syntactic notion of relevance may have a weakness that it may use redundant clauses while the notion of MUSes are also in some way insufficient. I will first provide an alternative characterization of unsatisfiability via conflict literal. Then, I present a refinement of the notion of redundancy for first-order logic via instantiation. Using these two notions, I describe a semantic relevance where it coincides with the MUS-based relevance in propositional logic while showing that there are first-order cases in which this notion captures more than what can be offered by the usual MUSes.

Conflict Literal

I provide an alternative for what it means for a clause set to be unsatisfiable. As a first observation, except for the trivially false clause $\bot$, the simplest contradiction occurs when we have a pair of ground literals K and L such that $K = \mathsf{comp}(L)$. They will be termed *conflict literals*. In propositional logic, conflict literals can be be defined like in Def. 2.1.3 from [JMRS17]. However, its first-order logic version should reflect the relation of literals and clauses to their respective ground instantiations (recall the compactness theorem for unsatisfiability).

Definition 3.1.5 (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 $N_2 \models \mathsf{comp}(L)$.

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

Obviously, conflict literals always come in pairs which we may call *a pair of conflict literals*.

The following is a rather simple example where, for every pair of conflict literals A and $\mathsf{comp}(A)$, the satisfiable clause sets N_1 and N_2 entailing them come from two different subsets of N.

Example 3.1.6 (Conflict literal 1)**.** Given

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

its conflicts are

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

For example, if we take the pair $P(a, b)$ and $\neg P(a, b)$, then they will respectively be entailed by the following clause sets.

$$\begin{aligned} N_1 &= \{\neg R(z), R(c) \vee P(a, y)\} \\ N_2 &= \{\neg P(a, b)\} \end{aligned}$$

However, this may not happen all the time as the following example shows.

Example 3.1.7 (Conflict literal 2)**.** 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

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

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

Here, all clauses in N_1 and N_2 come from the instantiation of both clauses in N. The first one uses $\{x \mapsto d, y \mapsto a\}$ while the second one uses $\{x \mapsto b, y \mapsto c\}$.

The existence of a conflict literal alternatively characterizes the unsatisfiability of a clause set. This can be viewed as a generalization of Lemma 2.1.4 from [JMRS17]. First, I present this for ground clauses where all literals in a ground MUS are conflict literals.

Lemma 3.1.8 (Minimal Unsatisfiable Ground Clause Sets and Conflict Literals)**.** If N is a minimal unsatisfiable set of ground clauses (ground 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. □

In addition, it is possible to lift Lemma 3.1.8 to clause set with variables with the help of the Compactness Theorem.

Lemma 3.1.9 (Conflict Literals and Unsatisfiability)**.** Given a set of clauses N, $\mathsf{conflict}(N) \neq \emptyset$ if and only if N is unsatisfiable.

Proof. "$\Rightarrow$" Let $L \in \mathsf{conflict}(N)$. By definition, there are two satisfiable subsets of instances N_1, N_2 from N such that $N_1 \models L$ and $N_2 \models \mathsf{comp}(L)$. 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 $\mathcal{I} \models \mathsf{comp}(L)$, 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 Th. 1.1.7, there is a minimal set of ground instances N' from N that is also unsatisfiable. By Lemma 3.1.8, any literal occurring in N' is a conflict literal. This means $\mathsf{conflict}(N')$ is not empty and thus $\mathsf{conflict}(N)$ is also not empty due to N' being instances from N. □

Dependency

Intuitively, removing a clause implied by the others (redundant clause) is semantic preserving. However, this may in some be harmful to completeness when performed during the execution of a calculus [NR01, BG01]. Yet, regardless of its compatibility with completeness, how we deal with redundancy is very important in terms of efficiency, e.g., in propositional logic [BR00, Lib05]. It is also apparently important when we try to define a semantic notion of relevance. I will show this via some non-trivial examples later but as an easy example, a syntactically relevant clause would step down to be syntactically semi-relevant when duplicated. So, in order to have a semantically robust notion of relevance in first-order logic, we need to use the following notion of (in)dependency. One can see this as a refinement of the redundancy notion via instantiation.

Definition 3.1.10 (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.

In propositional logic (and ground first-order clauses), a clause is dependent if and only if it is redundant. A subsumed clause is obviously a dependent clause. However, for the first-order case, 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 there is an instance $P(a, b)$ of $P(x, b)$ entailed by $P(a, y)$. With the same argument, $P(a, y)$ is then also dependent.

Semantic Relevance

Now, we are ready to define the semantic notion of relevance based on conflict literals and dependency.

Definition 3.1.11 (Semantic Relevance)**.** Given an unsatisfiable set of independent clauses N, a clause $C \in N$ is

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

One might think that the set of clauses being independent is rather too restrictive. It is correct in the sense that, it is simply not defined on full first-order clauses that are not independent. While this is by no means closed for future work, I argue that such restriction is still reasonable. This is not only because of (as already mentioned before) incompatibility with completeness [NR01, BG01] but also that, w.r.t. the proposed notions, their existence may non-trivially influence the other clauses. This affects both the syntactic relevance (Ex. 3.1.12) and its semantic characterization (Ex. 3.1.13). Nevertheless, even with this restriction, I argue that our semantic notion here is already a better refinement even considering full first-order logic (remember the contract example in the introduction section). To further strengthen this argument, I will also show in Lemma 3.1.14 that, in the propositional case, MUSes can be used as an alternative characterization similar to what existed in the literature. Second, for first-order logic, I will show how MUSes should be related to the original clause set in Lemma 3.1.15 and additionally show the weakness in Ex. 3.1.16 when it is not adhered to.

Example 3.1.12 (Syntactic semi-relevance vs dependent clause set)**.** Given a clause set

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

the existence of the dependent clauses $\neg P \vee Q$ and $\neg R \vee P$ causes an independent clause $\neg Q \vee R$ to be a syntactically semi-relevant clause.

A refutation for N in Ex. 3.1.12 is shown in Fig. 3.4. When both of the dependent clauses are removed resulting in the clause set $N' = \{P, \neg P, \neg Q \vee R\}$, the clause $\neg Q \vee R$ is actually irrelevant. Semantically, it is thus not clear how $\neg Q \vee R$ should even be treated in the face of the dependent clauses.

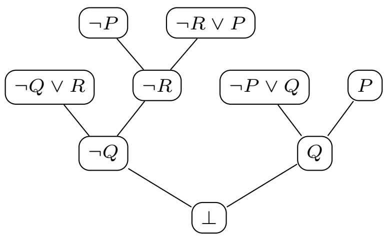

Figure 3.4: A refutation in which an independent clause $\neg Q \vee R$ is only usable due to dependent clauses

The following example additionally shows that extending our notion to include dependent clauses in propositional logic is not as easy as performing the removal of dependent clauses. This is because, if there are more than one dependent clause, on the one hand, it is not always possible to simply remove all of them (without preserving unsatisfiability) while on the other hand, it is not obvious which subset of the dependent clauses should be removed. This justifies the need for independence restriction already for propositional logic not to mention first-order logic. This is illustrated by the following Ex. 3.1.13.

Example 3.1.13 (Semantic semi-relevance vs dependent clause set)**.** Given a clause set

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

$\neg P \vee Q$ and $\neg R \vee Q$ are dependent because

$$\{R, \neg R \vee Q\} \models \neg P \vee Q$$
$$\{P, \neg P \vee Q\} \models \neg R \vee Q$$

From Ex. 3.1.13, only one of the dependent clauses can be removed without making N satisfiable. More specifically, its respective removals would give us the following unsatisfiable clause sets.

$$N_1 = \{P, R, \neg R \vee Q, \neg Q\}$$
$$N_2 = \{P, R, \neg P \vee Q, \neg Q\}$$

Here, we have a problem: R and P are respectively semi-relevant in N_1 and N_2 but irrelevant in N_2 and N_1. The removal of both is even worse because it turns N into a satisfiable clause set $N_3 = \{P, R, \neg Q\}$.

Propositional Logic Case Our notion of (semi-)relevance can also be characterized by MUSes in propositional logic. As a comparison, we consider Def. 2.1.1 from [KLM06]: 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.* This is related to the proposed notion via the following lemma where we consider only independent propositional clause sets.

Lemma 3.1.14 (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 $\mathsf{conflict}(N \setminus \{C\}) = \emptyset$ if and only if $N \setminus \{C\}$ is satisfiable by Lemma 3.1.9 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 Th. 1.3.2 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 3.1.8 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 3.1.8 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. $\square$

First-Order Logic Case I will show that even though notions from propositional logic can often be immediately used as a generalization to first-order logic, in the context of relevance, this is not the case. In our case, MUSes must be considered at its ground level via instantiation as a characterization of the proposed notion of semantic relevance.

Lemma 3.1.15 (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. Thus, by Lemma 3.1.9, it 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 3.1.8, 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, $P \in \mathsf{conflict}(N) \setminus \mathsf{conflict}(N \setminus \{C\})$ and thus C is semi-relevant. □

I illustrate the necessity of this lemma via an example demonstrating that the MUSes where the clauses are simply taken from the original clause set (without instantiation) cannot capture the proposed semi-relevancy notion.

Example 3.1.16 (First-Order (Semi-)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

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

First, $P(a,y)$ is a relevant clause. To show this via the definition, we consider the literals entailed by some satisfiable instance set N' from N where $P(a,y) \notin N'$ are $\{\neg Q(b,d)\} \uplus \{\neg P(a,t), \neg Q(t,e) | t \in \{a,b,c,d,e\}\}$. It is not difficult to see that no two of them contradict each other and thus $\mathsf{conflict}(N \setminus \{C\}) = \emptyset$. Second, the clause $\neg P(a,d) \vee Q(b,d)$ is semi-relevant because we have a conflict literal $Q(b,d) \notin \mathsf{conflict}(N \setminus \{\neg P(a,y) \vee Q(b,d)\})$. Last, clause $Q(z,e)$ is irrelevant.

The only MUS from N is $\{P(a,y), \neg P(x,c)\}$ with grounding substitution $\{x \mapsto a, y \mapsto c\}$. This means the notion of MUS cannot capture the clause $\neg P(a,d) \vee Q(b,d)$ from Ex. 3.1.16 which is not in any MUSes. 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. In the example, we could identify two ground MUSes:

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

and

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

This means that the notion of MUS is not necessarily gone. The semantic relevance can still be viewed in terms of ground MUSes with the help of Lemma 3.1.15. First, $P(a, y)$ is a relevant clause because every MUS contains an instance of it ($P(a, c)$ and $P(a, d)$). Second, The clause $\neg P(a, d) \vee Q(b, d)$ is semi-relevant since it is contained in the second MUS. Third, the clause $Q(z, e)$ is irrelevant since we can see that no MUS contains any instance of $Q(z, e)$.

Note that, in first-order logic,the set of MUSes becomes possibly infinite. Thus, some popular MUS-related tasks such as counting [BM21] and computing union [MKIM19], and enumeration [BK16] are not possible anymore.

3.2 Generalization of Set-of-Support Strategy

Unlike testing for relevant clauses, the semi-relevancy test cannot be supported by the existing SOS refutational completeness. The need for an effective semi-relevancy test led us to a sharper generalization of the existing refutational completeness for SOS in Th. 1.3.1 by [WRC65].

3.2.1 Refutation Transformation

The generalized completeness result of SOS is proven by transforming non-SOS refutations into SOS refutations. As an illustration of the proof transformation technique, we consider the following unsatisfiable set of clauses N.

$$\begin{aligned} N = \{&(1){:}\{1\}\neg Q(x_3,f(a)) \vee \{2\}P(f(a)), \quad (2){:}\{3\}\neg P(x_4) \vee \{4\}\neg Q(b,x_4),\\ &(5){:}\{5\}\neg Q(b,a) \vee \{6\}Q(x_1,f(x_6)),\\ &(6){:}\{7\}Q(b,x_2) \vee \{8\}R(x_2) \vee \{9\}T(c,x_1),\\ &(9){:}\{10\}\neg R(x_5), \qquad (11){:}\{11\}\neg T(c,b)\}\end{aligned}$$

Literals are labeled in N by a singleton set of a unique natural number [LAWRS07]. We will refer to the literal labels during proof transformation in order to keep track of the relevant resolution and factorization steps. The labels are inherited in a resolution inference and united for the factorized literal in a factoring inference. See the factoring inference on clause (3), Fig. 3.5.

Figure 3.5 shows a resolution refutation

$$\pi = [(5), (6), (7), (1), (2), (3), (4), (8), (9), (10), (11), (12)]$$

from N. This resolution refutation is also an SOS refutation with SOS $M = \{(2), (5)\}$ and the remaining clause set $N \setminus M$. It is not an SOS refutation with SOS $M = \{(5)\}$ and the remaining clause set $N \setminus M$ because the resolution step

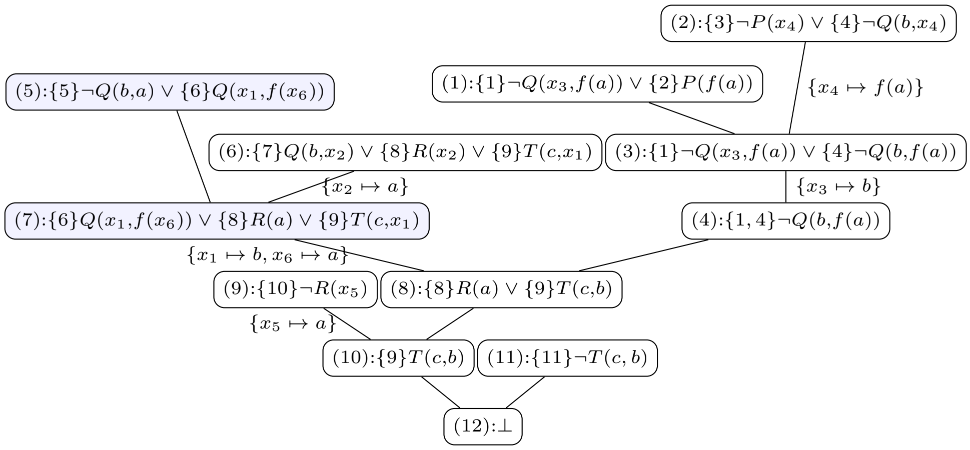

Figure 3.5: Refutation of π of N

between clauses (1) and (2) is not an SOS step. The shaded part of the tree belongs to an SOS deduction with $M = \{(5)\}$.

With respect to the clause set N and its refutation in Fig. 3.5, clause (5) is semi-relevant but not relevant, because the clauses (1), (2), (6), (9), (11) are already unsatisfiable. In contrast, clauses (1), (2), (6), (9), (11) are all relevant.

The transformation works by picking a clause closest to the leaves of the tree, obtained by resolution, that has one parent with SOS subdeduction, but the other parent is not in the SOS nor an input clause. For our example with starting SOS $M = \{(5)\}$ this is clause (8). The parent (7) is already a result of a subdeduction with SOS M but the other parent (4) is not. The overall grounding substitution of π is $\tau = \{x_1 \mapsto b, x_2 \mapsto a, x_3 \mapsto b, x_4 \mapsto f(a), x_5 \mapsto a, x_6 \mapsto a\}$. Now, in a single step of the transformation, we perform the resolution step on the labeled literal $\{1,4\}\neg Q(b,f(a))$ and the respective literal $\{6\}Q(x_1,f(x_6))$ of the SOS derivable clause (7) already on the respective literals from the input clauses yielding (8), here clauses (1) and (2). To this end the derivation $[(5),(6),(7)]$ is copied(where we also introduce fresh variables to make it variable-disjoint), see Fig. 3.6, yielding the clauses (7) and (7′) used in the refutation π' below, see also Fig. 3.7.

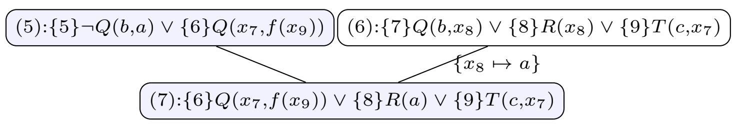

Figure 3.6: The copied subdeductions deriving (7)

The two freshly renamed copies (7) and (7′) are resolved with the respective input clauses (1) and (2). Finally, the rest of the deduction yielding clause (8) is simulated with the resolved input clauses, see Fig. 3.7. Now (8′′′) is exactly clause (8) from the original deduction π, but (8′′′) is derived by an SOS deduction. The deduction can then be continued the same way it was done in π and in this case

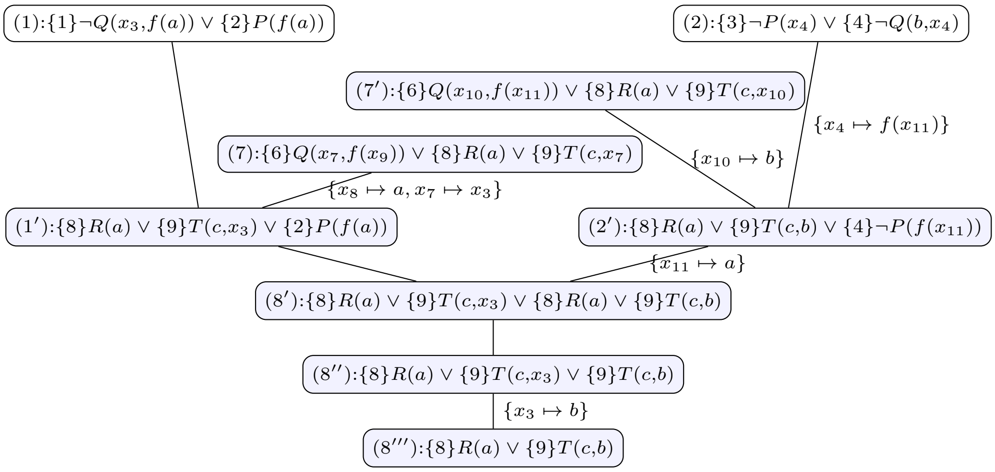

Figure 3.7: The new SOS deduction yielding a copy of clause (8)

will already yield an SOS refutation.

$$\pi' = [(5), (6), (7), (5'), (6'), (7'), (1), (1'), (2), (2'), (8'), (8''), (8'''), (9), (10), (11), (12)].$$

The example motivates the use of literal labeling. Firstly, they help keep track which literals in input clauses to resolve: here by looking into the label of $\{1, 4\}\neg Q(b,f(a))$, we can see that it results from a factoring of the literals $\{1\}\neg Q(x_3,f(a))$ and $\{4\}\neg Q(b,x_4)$. Secondly, they guide additional factoring steps in π' during the simulation of the non-SOS part from π: here the factoring between the two literals labeled $\{8\}$ in clause $(8')$ and the two literals with label $\{9\}$ in clause $(8'')$. The transformation always works because the overall grounding substitution of the initial refutation π is preserved. We only need to extend it to cover the extra variables appearing in the freshly renamed copies of clauses.

The above example shows the importance of keeping track the literals in a deduction in such a way that is easy to identify where they come from in the input clauses. A *labeled literal* is a pair ML where M is a finite non-empty set of natural numbers called the *label* and L is a literal. We identify literals with labeled literals and use the function lb to refer to the label. The function lb is extended to clauses via the union of its respective literal labels. We extend the notion of a clause to that of a labeled clause built on labeled literals in a straightforward way. We call a deduction π_N *label-disjoint* if the clauses from N in the deduction have unique singleton labels. Labels are propagated in a deduction as follows: in the case of a Resolution inference, the labels of the parent clauses are inherited and in the case of the Factoring inference, the label of the remaining literal is the union of labels of the factorized literals.

In general, we need to identify the parts of a deduction that are already contained in an SOS deduction, this is called the *partial* SOS of a deduction, Def. 3.2.1. Then this information can be used to perform the above transformation on any deduction π.

Definition 3.2.1 (PSOS of a Deduction)**.** Let π be a deduction from $N \uplus M$, then the *partial SOS* (PSOS) O^* of $\langle \pi, N, M \rangle$ is defined as $O^* = \bigcup_{i=0}^{m} O^i$, where $O^0 = M$, $O^{i+1} = O^i \cup \{C_j\}$ provided $C_j \in \pi$, $C_j \notin O^i$ and C_j is either the factor of some clause in O^i or the resolvent of two clauses in π where at least one parent is from O^i, and where O^m is such that there is no longer such a C_j in π.

The partial SOS is well-defined because the resulting O^* is independent of the sequence O^i used. For example, for the deduction π from N presented in Fig. 3.5 the set $O^* = \{(5), (6), (7)\}$ is the PSOS of $\langle \pi, N, \{5\} \rangle$. Next we present a criterion when the PSOS of a deduction actually signals an SOS deduction.

Lemma 3.2.2 (SOS Deduction)**.** Let O^* be the PSOS of $\langle \pi, N, M \rangle$. Then π is an *SOS deduction* if $O^* \setminus M = \pi \setminus (N \cup M)$[1], i.e., all inferred clauses in π are contained in O^*.

Proof. Let $\pi_{N \cup M} = [C_1, \dots, C_n]$ and $[C'_1, \dots, C'_m]$ be the subsequence of $\pi_{N \cup M}$ with input clauses removed. Let O^* be the PSOS of $\langle \pi, N, M \rangle$. Then $[C'_1, \dots, C'_m] = O^* \setminus M = \pi \setminus (N \cup M)$ by assumption. We show that $(N, M^0) \Rightarrow^*_{RES} (N, M^m)$ is an SOS derivation, following Def. 3.1.2 by induction on m. If $m = 0$ then π only consists of input clauses and there is nothing to show. For the case $m = 1$, the clause C'_1 is the result of a factoring inference from M or the result of a resolution inference from $N \cup M$ such that at least one parent is in M as for otherwise $C'_1 \notin (O^* \setminus M)$. So $(N, M^0) \Rightarrow^*_{RES} (N, M^0 \cup \{C'_1\})$ is an SOS derivation. For the induction case, assume the property holds for i. If C'_{i+1} is the result of a factoring inference, then its parent C'' is contained in M^i because otherwise $C'' \in N$ (because π is a deduction) and, therefore $C'_{i+1} \notin (O^* \setminus M)$, a contradiction. If C'_{i+1} is the result of a resolution inference, then again all its parents are contained in $N \cup M^i$ because π is a deduction. If both parents are from N, then $C'_{i+1} \notin (O^* \setminus M)$, a contradiction. So, by the induction hypothesis, $(N, M^0) \Rightarrow^*_{RES} (N, M^i) \Rightarrow_{RES} (N, M^{i+1})$ is an SOS derivation. ☐

The rest of this section describes the transformation in detail. Then, we prove the new generalized completeness result for the set-of-support strategy.

Let π be a label-disjoint deduction from $N \cup M$ and let $C_k \in \pi$ be a clause of a minimal index such that C_k is the result of a resolution inference from clauses $C_j \in O^*$ and $C_i \notin (N \cup O^*)$. Let τ be an overall ground substitution for π. We transform π into π' to make it "closer" to an SOS derivation by changing the deduction of C_i while preserving τ. Let

$$\begin{aligned} C_j &= C'_j \vee L \\ C_i &= C'_i \vee K \\ C_k &= (C'_i \vee C'_j)\sigma \end{aligned} \tag{3.1}$$

where $\sigma = \mathsf{mgu}(K, \mathsf{comp}(L))$. Without loss of generality, we assume that

$$\pi = [C_1, \dots, C_i, C_{i+1}, \dots, C_j, C_k, C_{k+1}, \dots, C_n] \tag{3.2}$$

where $[C_1, \dots, C_i]$ and $[C_{i+1}, \dots, C_j]$ are subdeductions of π, and the prefixes of these sequences are exactly the introduced renamed copies of input clauses from N that are used to derive C_i and C_j, respectively. Moreover, we will keep track of new clauses that are taken from the previous ones or transformed using

[1]Here, we refer to the removal of all input clauses from O^* and π, respectively.

a notion of *associated clause* and prove its properties later. The transformed derivation will be

$$\pi' = [C^1_{i+1}, \dots, C^1_j, \dots, C^m_{i+1}, \dots, C^m_j, D_1, \dots, D_l, C'_{k+1}, \dots, C'_n] \tag{3.3}$$

where

(a) the subsequences $[C^o_{i+1}, \dots, C^o_j]$ are freshly variable-renamed copies of the sequence $[C_{i+1}, \dots, C_j]$ where $m = |\,\mathsf{lb}(K)|$. For the copies $[C^o_{i+1}, \dots, C^o_j]$ we keep the labels of literals of the original sequence $[C_{i+1}, \dots, C_j]$ for reference in the transformation. The clauses C^o_j are decomposed into $C'^o_j \vee L'$, in the same way, that the clause C_j is decomposed into $C'_j \vee L$. Thus, for each clause from N in the sequence $[C_1, \dots, C_i]$ containing a literal K' with $\mathsf{lb}(K') \subseteq \mathsf{lb}(K)$ we add a deduction deriving a renamed copy of C^p_j; let δ^p be the renaming substitution from the old to the freshly renamed sequence, then we extend τ to τ' as follows: $\tau'_0 = \tau$, $\tau'_{p+1} = \tau'_p \circ \{x\delta^{p+1} \mapsto t \mid x \in \mathsf{dom}(\delta^{p+1}), t = x\tau\}$ for $1 \leq p \leq m$ yielding the overall new grounding substitution $\tau' = \tau'_m$ for π';

(b) the clauses $D_1, \dots, D_l$ are generated by simulating the deduction $[C_1, \dots, C_i]$ eventually producing C_k, up to possible variable renamings: Let C_p be the current clause out of this deduction and let $D_1, \dots, D_q$ be the clauses generated so far until C_{p-1};

   (i) if C_p is an input clause not containing a literal K' with $\mathsf{lb}(K') \subseteq \mathsf{lb}(K)$, then $D_{q+1} = C_p$ and we associate D_{q+1} with C_p;

   (ii) if C_p is an input clause containing a literal K' with $\mathsf{lb}(K') \subseteq \mathsf{lb}(K)$, then $D_{q+1} = C_p$ and D_{q+2} is the resolvent between D_{q+1} and a so far unused clause C^o_j on the literals $K' \in D_{q+1}$ and $L' \in C^o_j$ where $\mathsf{lb}(K') \subseteq \mathsf{lb}(K)$ and $\mathsf{lb}(L') = \mathsf{lb}(L)$ and we associate D_{q+2} with C_p;

   (iii) if C_p is the resolvent between two clauses $C_{i'}, C_{j'}$ then we perform the respective resolution step between the associated clauses and respective associated literals from $D_{q'}, D_{q''}$ yielding D_{q+1} and associate D_{q+1} with C_p;

   (iv) if C_p is the factor on some literal K' with $\mathsf{lb}(K') \subseteq \mathsf{lb}(K)$, then we perform the respective factoring steps $D_{q+1}, \dots, D_{q+s}$ for respective literals with labels from C'_j, where $s = |C'_j|$ and we associate D_{q+s} with C_p,

   (v) if C_p is the factor on some literal K' with $\mathsf{lb}(K') \not\subseteq \mathsf{lb}(K)$, then we perform the respective factoring step on the respective literals with identical labels from clause $D_{q'}$ yielding D_{q+1} and we associate D_{q+1} with C_p;

(c) the clauses $C'_{k+1}, \dots, C'_n$ are obtained by simulating the generation of clauses $C_{k+1}, \dots, C_n$ where C_k is substituted with D_l.

Note that by assumption, the generation of clauses $C_{k+1}, \dots, C_n$ does not depend on clauses $C_1, \dots, C_i, C_{i+1}, \dots, C_j$ but only on C_k and the input clauses. We will prove that $C_k\tau = C_k\tau' = D_l\tau'$ which is then sufficient to prove $C_n\tau = C_n\tau' = C'_n\tau'$ and for the above to be well-defined. In general, the clause D_l is not identical to C_k because we introduce fresh variables in π' and do not make any specific assumptions on the unifiers used to derive D_l.

Mapping the transformation to our running example, Fig. 3.5: $C_j = (7)$, $C_i = (4)$, and $C_k = (8)$. We need two copies of (7) because $K = \{1,4\}\neg Q(b, f(a))$ so $m = |\{1,4\}| = 2$ and $L = \{6\}Q(x_1, f(x_6))$.

3.2.2 SOS Generalized Completeness

In this section, I show that when the previous transformation is done repeatedly on any deduction, we will get an SOS deduction, given that at least one clause from the SOS occurs in the original deduction. Firstly, we show that associated clauses of the transformed deduction preserve the main properties of the original deduction. The extended substitution is identical to the original substitution on old clauses and the changed part of the deduction ends in exactly the same clause.

Lemma 3.2.3 (Properties of Associated Clauses)**.** Let C_j, C_i, C_k, L, K, π, π', τ, τ' be as defined in (3.1), (3.2), and (3.3), page 52. For each clause C out of $[C_1, \ldots, C_i]$ and clause D associated with C:

1. $C\tau = C\tau'$,
2. $K'\tau' = L'\tau'$ if $\mathsf{lb}(K') = \mathsf{lb}(L')$ for any K', L' occurring in either π or π',
3. $\mathsf{lb}(C) \setminus \mathsf{lb}(K) = \mathsf{lb}(D) \setminus \mathsf{lb}(C_j'^o)$ and $\mathsf{lb}(C_j'^o) \subseteq \mathsf{lb}(D)$ if there is $K' \in C$ with $\mathsf{lb}(K') \subseteq \mathsf{lb}(K)$,
4. $C\tau \setminus \{K'\tau \in C \mid \mathsf{lb}(K') \subseteq \mathsf{lb}(K)\} = D\tau' \setminus \{L'\tau' \in D\tau' \mid \mathsf{lb}(L') \in \mathsf{lb}(C_j'^o)\}$ and $C_j'^o\tau' \subseteq D\tau'$ if there is $K' \in C$ with $\mathsf{lb}(K') \subseteq \mathsf{lb}(K)$,
5. $C_k\tau = D_l\tau'$.

Proof. 1. By definition of τ' the additional variables in τ' do not occur in C while τ' is identical to τ on the variables of C, hence $C\tau = C\tau'$.

2. By induction on the generation of π'. For the base case, every literal occurring in $N \cup S$ has a unique label and any renamed clause C_m^o for some $C_m \in (N \cup S)$ has the labels kept. So, for any two literals K' and L' in any non-inferred clauses in π and π', $K'\tau' = L'\tau'$ when the labels are equal. For the induction step, for inferred clauses, $\mathsf{lb}(K') = \mathsf{lb}(L')$ happens when the label of K' is inherited from L' through an inference. The inference uses an mgu which is compatible with τ' due to τ' being an overall ground substitution, so $K'\tau' = L'\tau'$.

3. We prove this property by induction on the length of the derivation $[C_1, \ldots, C_i]$. Let $C = C_p$, $1 \leq p \leq i$, and let $D_1, \ldots, D_q$ be the clauses generated until C_{p-1} for which, by the induction hypothesis the property already holds.

(i) If C is an input clause not containing a literal K' with $\mathsf{lb}(K') \subseteq \mathsf{lb}(K)$, we have $C = C_p = D_{q+1} = D$ and $\{K' \in C\tau \mid \mathsf{lb}(K') \subseteq \mathsf{lb}(K)\} = \{L' \in D_q\tau' \mid \mathsf{lb}(L') \subseteq \mathsf{lb}(C_j'^o)\} = \emptyset$.

(ii) If C is an input clause containing a literal K' with $\mathsf{lb}(K') \subseteq \mathsf{lb}(K)$ then $D = D_{q+2}$ results from a resolution inference between $C = C_p$ and an unused C_j^o on the literals K' and $L' \in C_j^o$ with $\mathsf{lb}(L') = \mathsf{lb}(L)$. Let $C = C' \vee K'$. Then, $D\tau' = (C' \vee C_j'^o)\tau'$ and hence $\mathsf{lb}(C) \setminus \mathsf{lb}(K) = \mathsf{lb}(D) \setminus \mathsf{lb}(C_j'^o)$ because

$\mathsf{lb}(C) \cap \mathsf{lb}(C_j^o) = \emptyset$ as π is a label-disjoint deduction and $\mathsf{lb}(C_j) = \mathsf{lb}(C_j^o)$ by construction.

(iii) If C is a resolvent of $C_{i'} = C'_{i'} \vee L'_{i'}$ and $C_{j'} = C'_{j'} \vee L'_{j'}$ on literals $L'_{i'}, L'_{j'}$, then $C\tau = C'_{i'}\tau \vee C'_{j'}\tau$, and D_{q+1} is a resolvent of some $D_{q'} = D'_{q'} \vee L''_{q'}$ and $D_{q''} = D'_{q''} \vee L''_{q''}$ associated with $C_{i'}$ and $C_{j'}$ respectively. We have $\mathsf{lb}(L'_{i'}) = \mathsf{lb}(L''_{q''})$ and $\mathsf{lb}(L'_{j'}) = \mathsf{lb}(L''_{q''})$ and none of these literals has a label from $\mathsf{lb}(K)$ or $\mathsf{lb}(C_j'^o)$. Hence, the conjecture holds by the induction hypothesis.

(iv) If C results from a factoring on K' from C_{p-1}, we get D_{q+s} by a sequence of s factoring inferences from D_{q+1} associated with C_{p-1}. Any factorings on C_{p-1} and D_{q+1} do not change literal labels because we factorize literals of identical label. So, this property holds by the induction hypothesis. This holds regardless of whether $\mathsf{lb}(K') \subseteq \mathsf{lb}(K)$.

4. From Lemma 3.2.3.3 we know that $\mathsf{lb}(C) \setminus \mathsf{lb}(K) = \mathsf{lb}(D) \setminus \mathsf{lb}(C_j'^o)$ and $\mathsf{lb}(C_j'^o) \subseteq \mathsf{lb}(D)$ if there is $K' \in C$ with $\mathsf{lb}(K') \subseteq \mathsf{lb}(K)$. Since the labels coincide, using Lemma 3.2.3.2, we have $C\tau' \setminus \{K' \in C\tau' \mid \mathsf{lb}(K') \subseteq \mathsf{lb}(K)\} = D\tau' \setminus \{L' \in D\tau' \mid \mathsf{lb}(L') \in \mathsf{lb}(C_j'^o)\}$ and $C_j'^o\tau' \subseteq D\tau'$ if there is $K' \in C$ with $\mathsf{lb}(K') \subseteq \mathsf{lb}(K)$. This hypothesis holds by applying Lemma 3.2.3.1 on literals and clauses from π in the equation.

5. The clause C_k is the result of a resolution inference between C_i and C_j upon K and L: $C_k\tau = C'_i\tau \cup C'_j\tau$. By translation and because $\{K' \in C_i \mid \mathsf{lb}(K') \subseteq \mathsf{lb}(K)\} = \{K\}$, the clause C_i is associated with $D_l \in \pi'$ and $C_i\tau \setminus \{K\tau\} = D_l\tau' \setminus \{L' \in D_l\tau' \mid \mathsf{lb}(L') \in \mathsf{lb}(C_j'^o)\}$. Since $C_j'^o\tau' = C'_j\tau = C_j\tau \setminus \{L\tau\}$, we have $\{L'' \in D_l\tau' \mid \mathsf{lb}(L'') \subseteq \mathsf{lb}(L') \text{ for some } L' \in C_j'^o\} = D_l\tau' \cap C_j^o \setminus \{L\tau\} = C_j \setminus \{L\tau\}$. So $C_i \setminus \{K\tau\} = D_l\tau' \setminus (D_l\tau' \cap C_j \setminus \{L\tau\}) = D_l\tau' \setminus (C_j \setminus \{L\tau\})$. We can add $C_j\tau \setminus \{L\tau\}$ to both sides and get $C_k\tau = C_i\tau \cup C_j\tau \setminus \{K\tau, L\tau\} \supseteq D_l\tau'$. In addition, since $\mathsf{lb}(K) \subseteq \mathsf{lb}(K)$, this means $C_j\tau = C_j'^o\tau' \subseteq D_q\tau'$. Therefore $C_k\tau = C_i\tau \cup C_j\tau \setminus \{K\tau, L\tau\} = D_l\tau'$. □

Next, we need a well-founded measure that decreases with every transformation step and in case of reaching its minimum signals an SOS deduction.

Definition 3.2.4 (SOS Measure)**.** Given a clause set N and an initial SOS S, the *SOS measure* of a deduction π is $\mu(\pi)$ where $\mu(\pi) = \sum_{C_i \in \pi} \mu(C_i, \pi)$ and $\mu(C_i, \pi) = 0$ if $C_i \in N \cup O^*$ otherwise $\mu(C_i, \pi) = 1$.

Lemma 3.2.5 (Properties of μ)**.** Given a clause set N, an initial SOS S, and a deduction π that contains at least one resolution step,

1. $\mu(\pi) \geq 0$, and
2. if $\mu(\pi) = 0$ then π is an SOS deduction.

Proof. 1. Obvious because it is a sum non-negative numbers (0 for input clauses and 1 otherwise).

2. Towards contradiction, suppose $\pi = [C_1, \ldots, C_n]$ is not an SOS deduction. This means $O^* \setminus S \subsetneq \pi \setminus (N \cup S)$ by Lemma 3.2.2. Consider a clause $C_i \in (\pi \setminus (N \cup S)) \setminus (O^* \setminus S)$ of minimal index. Then C_i must be the result of an inference on some C_j and C_k such that both are not in O^*. This means $C_i \notin (N \cup O^*)$. For this clause, μ assigns a nonzero value: $\mu(C_i, \pi) > 0$. Therefore, $\mu(\pi) \neq 0$. □

Next, we combine the properties of associated clauses in one transformation step with the properties of the measure resulting in an overall deduction transformation that can be recursively applied and deduces the same clause modulo some grounding.

Lemma 3.2.6 (Properties of the Transformation)**.** Given a deduction π of a clause C_n from $N \cup S$ that contains at least one resolution step such that $\pi \cap S \neq \emptyset$, an overall ground substitution τ of π and the transformed deduction π' of a clause C'_n as defined in (3.1), (3.2), and (3.3) with overall ground substitution τ', we have:

1. π' is a deduction from $N \cup S$,

2. $C_n\tau = C'_n\tau'$, and

3. $\mu(\pi') < \mu(\pi)$.

Proof. 1. We show that π' is a deduction following Def. 3.1.1. These properties will be carried over from π. Observe that, if π_1 is a deduction of C_k from $N \cup S$ and π_2 is a deduction from $N \cup S \cup \{C_k\}$ using C_k only once, their concatenation $\pi_1 \circ \pi_2$ is a deduction from $N \cup S$. Firstly, the subsequences $[C^o_{i+1}, \ldots, C^o_j]$ are deductions of C^o_j from $N \cup S$ since they are only the renamed copies of the subdeduction $[C_{i+1}, \ldots C_j]$ of π. Secondly, the subsequence $[C_k, \ldots, C_n]$ is a deduction of C_n from $N \cup S \cup \{C_k\}$ since the clauses after C_k do not use any clauses before C_k by the way π is represented as a sequence. Now, by showing that $[C^1_j, \ldots, C^m_j, D_1, \ldots, D_l, C_k]$ is a deduction of C_k from $N \cup S \cup \{C^o_j\}_{o \in [1,m]}$, the sequence $[D_1, \ldots, D_l]$ would then connect the initial copied sequences and the tailing subsequence. Each C^o_j is used for exactly one resolution inference producing some D_q, the other required clauses are copied, and the later resolution and factoring steps in $[D_1, \ldots, D_l]$ are sound while the deduction properties of $[C_1, \ldots, C_i]$ are preserved in its associated clauses: for an inference where $C_{p'}$ (and $C_{p''}$) generates C_p, we have a unique inference between their associated clauses $D_{q'}$, $(D_{q''},)$ D_{q+1} where $D_{q'}$ (and $D_{q''}$) generates D_{q+1}, possibly with additional factoring inferences in between. If C_p is an input clause not containing a literal K' with $\mathsf{lb}(K') \subseteq \mathsf{lb}(K)$, then $D_{q+1} = C_p \in N$. The clause D_{q+1} is used in π' as C_p is used in π; if C_p is an input clause containing a literal K' with $\mathsf{lb}(K') \subseteq \mathsf{lb}(K)$, the resolution between D_{q+1} and a so far unused clause C^o_j is sound as K' and $\mathsf{comp}(L')$ are unifiable by τ'. Here, all C^o_j will be eventually used as there are $m = |\,\mathsf{lb}(K)|$ literals in the clauses from N; if C_p is the resolvent between two clauses $C_{i'}, C_{j'}$ then the respective resolution step between the associated clauses $D_{q'}, D_{q''}$ upon the respective associated literals K' and L' is sound because we can get $K'\tau' = \mathsf{comp}(L')\tau'$ using Lemma 3.2.3; if C_p is the factor on some literal K' with $\mathsf{lb}(K') \subseteq \mathsf{lb}(K)$, then the respective factoring steps $D_{q+1}, \ldots, D_{q+s}$ are also sound: each pair of the s associated literals M and M' from C^o_j and $C^{o'}_j$ are unifiable because $M\tau' = M'\tau'$; if C_p is the factor of C_{p-1} upon some literal K' and L' with $\{\mathsf{lb}(K'), \mathsf{lb}(L')\} \not\subseteq \mathsf{lb}(K)$, the respective factoring step on the associated clause $D_{q'}$ is also sound by Lemma 3.2.3. Therefore π is a deduction from $N \cup S$.

2. By Lemma 3.2.3.5, $C_k\tau = D_l\tau'$. The derivation of clauses $C_k, C_{k+1}, \ldots, C_n$ only depends on the input clauses by assumption. By an inductive argument we get $C_{k+1}\tau = C'_{k+1}\tau'$ yielding $C_n\tau = C'_n\tau'$.

3. The clauses in $[C^o_{i+1}, \ldots, C^o_j]$ have the measure 0 as their original ones in $[C_{i+1}, \ldots, C_j]$ because they are in $N \cup O^*$. The clauses in $[C_k, \ldots, C_n]$ also retain their original measures. The clauses in $[D_1, \ldots, D_l]$ are s.t. $\Sigma^l_{k=1}\mu(\pi', D_k) < \Sigma^i_{k=1}\mu(\pi', C_k)$. More specifically, any $C \in [C_1, \ldots, C_i]$ that is not in $N \cup O^*$(with measure $\mu(C, \pi) \geq 1$) and containing K' with $\mathsf{lb}(K') \subseteq \mathsf{lb}(K)$ is associated with $D_q \in O^* \setminus N$ having the measure $\mu(D_q, \pi') = 0$, while all other clauses in $[D_1, \ldots, D_l]$ are either copied from π with the same measure as before or new in π' but have the measure 0.

By induction on the length of the sequence $[C_1, \ldots, C_i]$ we prove the following property: if D is associated with a clause $C \in [C_1, \ldots, C_i]$ and C contains some literal in $\{K' \mid \mathsf{lb}(K') \subseteq \mathsf{lb}(K)\}$, then $D \in N \cup O^*$ and $\mu(D, \pi') = 0$. Let $C = C_p$. Let $D_1, \ldots, D_q$ be the clauses generated until C_{p-1} s.t. the property already holds.

(i) If C_p is an input clause with no literals in $\{K' \mid \mathsf{lb}(K') \subseteq \mathsf{lb}(K)\}$, it is associated with $D_q = C_p$ s.t. $\mu(C_p, \pi) = \mu(D_q, \pi') = 0$;

(ii) If C_p is an input clause containing $\{K' \mid \mathsf{lb}(K') \subseteq \mathsf{lb}(K)\}$, it is resolved with some $C^o_j \in O^*$ resulting in $D_{q+1} \in O^*$. Here we have $\mu(C_p, \pi) = \mu(D_q, \pi') = 0$;

(iii) If C_p is the resolvent between two clauses $C_{i'}, C_{j'}$ then we perform the respective resolution step between the associated clauses $D_{q'}, D_{q''}$ yielding the clause D_q associated with C_p. If either $C_{i'}$ or $C_{j'}$ contains some literal from $\{K' \mid \mathsf{lb}(K') \subseteq \mathsf{lb}(K)\}$ then C_p contains this literal as well and either $D_{q'} \in O^*$ or $D_{q''} \in O^*$ by the induction hypothesis. So, we get $D_q \in O^*$ and $\mu(D_q, \pi') = 0$. Otherwise, $\mu(D_q, \pi') = \mu(C_p, \pi) = 1$;

(iv) If C_p is the factor of C_{p-1} on some literal K' with $\mathsf{lb}(K') \subseteq \mathsf{lb}(K)$, then we have the respective factoring steps $D_{q+1}, \ldots, D_{q+s}$ where D_{q+1} is associated with C_{p-1}. By the induction hypothesis, $D_{q+1} \in O^*$. Therefore $D_{q+1}, \ldots, D_{q+s} \in O^*$ with $\mu(D_{q+t}, \pi') = 0$ for $1 \leq t \leq s$;

(v) If C_p is the factor of C_{p-1} (associated with D_q) on some literal K' and L' with $\{\mathsf{lb}(K'), \mathsf{lb}(L')\} \not\subseteq \mathsf{lb}(K)$, the factoring happens to the associated clauses in π' with similar measure.

Finally, by the choice of C_i, C_j, and C_k, there must exist at least one C_p with some literal from $\{K' \mid \mathsf{lb}(K') \subseteq \mathsf{lb}(K)\}$ but associated with some D such that $D \in O^*$ from case (iii) or (iv) before. This also means $\mu(D, \pi') = 0$. The clause C_i has this property as it contains K. In addition, any C_p has a nonzero measure because $C_i \notin N \cup O^*$ and C_p is used to prove C_i. Therefore, we have $\mu(C_p, \pi) > \mu(D, \pi') = 0$. As these clauses are never copied to π', $\mu(\pi') < \mu(\pi)$. □

One transformation step decreases the SOS measure and eventually, by an inductive argument, the main result can be proven since at some point the measure will become zero and we get the SOS resolution refutation.

Theorem 3.2.7 (Generalized SOS Completeness)**.** There is an SOS resolution refutation from (N, M) if and only if there is resolution refutation from $N \cup M$ that contains at least one clause from M.

Proof. "$\Rightarrow$": Obvious: If there is no refutation from $N \cup M$ using a clause M then there can also not be any SOS resolution refutation from (N, M).

"$\Leftarrow$": If there is a deduction refutation π from $N \cup M$ that contains at least one clause from M, then by an inductive argument on μ it can be transformed into an SOS deduction refutation with SOS M, and the result follows by Corollary 3.1.3. If $\mu(\pi) = 0$ then π is already an SOS deduction, Lemma 3.2.5. For otherwise, we transform the deduction π into a deduction π' according to (3.1), (3.2), and (3.3). A refutation always contains at least one resolution step, so by Lemma 3.2.6, π' is also a refutation from $N \cup M$ and $\mu(\pi') < \mu(\pi)$. Eventually, π' can be transformed into a label-disjoint deduction by assigning fresh labels to all used clauses from $N \cup M$. □

For the "$\Rightarrow$" direction, we can consider the clause set $N = \{P, \neg P\}$ and SOS $M = \{Q\}$. Here, it is easy to see that there is no refutation of $N \cup M$ using Q and there is no SOS refutation. Th. 3.2.7 also guarantees that the consecutive application of the proof transformation steps (3.1), (3.2), and (3.3), page 52, results in an effective recursive procedure that transforms non-SOS refutations into SOS refutations.

3.3 Test for (Syntactic) Semi-Relevance

In this section, I describe a test for the syntactic and semantic notion of relevance. The primary tool for this is basically the resolution calculus with the set-of-support strategy. For the semi-relevancy test, in particular, the generalized completeness proven in the previous section will be necessary.

Syntactic Relevance

First, I show a lemma useful to test whether a clause $C \in N$ is syntactically relevant. Even though the proof is obvious, note that we must also test if $N \setminus \{C\}$ is satisfiable which could be problematic in full first-order logic.

Lemma 3.3.1 (Syntactic Relevance)**.** Given an unsatisfiable set of clauses N, the clause $C \in N$ is syntactically relevant if and only if $N \setminus \{C\}$ is satisfiable.

Proof. Obvious: if $N \setminus \{C\}$ is satisfiable, there is no resolution refutation and since N is unsatisfiable, C must occur in all refutations. If C occurs in all refutations, there is no refutation without C so $N \setminus \{C\}$ is satisfiable. □

For the syntactically semi-relevant clauses, the generalized completeness for the SOS strategy comes into play.

Lemma 3.3.2 (Syntactic Semi-Relevance Test)**.** Given a set of clauses N, and a clause $C \in N$, C is syntactically semi-relevant if and only if $(N \setminus \{C\}, \{C\}) \Rightarrow^*_{\mathrm{RES}}$ $(N \setminus \{C\}, S \cup \{\bot\})$.

Proof. If $(N \setminus \{C\}, \{C\}) \Rightarrow^*_{\mathrm{RES}} (N \setminus \{C\}, S \cup \{\bot\})$ then we have found a refutation containing C. On the other hand, by Th. 3.2.7, Lemma 3.2.2 and Corollary 3.1.3, if there is a refutation containing C, then there is also an SOS refutation with SOS $\{C\}$. □

An immediate consequence of the above lemma is the following corollary.

Corollary 3.3.3 ((Semi-)decidability of the Semi-Relevance Test)**.** Testing syntactic semi-relevance in first-order logic is semi-decidable. It is decidable for all fragments where resolution constitutes a decision procedure.

Fragments where our syntactic semi-relevance test is guaranteed to terminate are for example first-order fragments enjoying the bounded model property, such as the Bernays-Schoenfinkel fragment [BS28] and the description logic $\mathcal{ALC}$.

The procedure for syntactic semi-relevance in any clause sets was proven in a technically heavy manner. For independent clause sets, we can apparently use the MUS characterization to show that a semi-relevant clause can be identified using SOS strategy. This is shorter as Th. 3.2.7, Lemma 3.2.2 and Corollary 3.1.3 are no longer used.

Lemma 3.3.4 (Semi-Relevance Test (Alternative Proof))**.** Given an independent clause set N, and a clause $C \in N$, C is semi-relevant if and only if $(N \setminus \{C\}, \{C\}) \Rightarrow^*_{\text{RES}} (N \setminus \{C\}, S \cup \{\bot\})$.

Proof. If C is semi-relevant in the independent clause set N, then there is a ground MUS M using an instance C' of C then there is an SOS resolution from $(M \setminus \{C'\}, \{C'\})$ because $\{M \setminus C'\}$ is satisfiable and Th. 1.3.1. By using the original clauses in N to be used in π, we get the SOS resolution as unifiability is obviously preserved. if there is a refutation containing C, then there is also an SOS refutation with SOS $\{C\}$. □

Semantic Relevance

For syntactic and semantic relevance, they obviously coincide. However, semi-relevance coincides only for independent clause sets. This means a test for syntactic (semi-)relevance is also useful for semantic (semi-)relevance on independent clause set.

Theorem 3.3.5 (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 3.1.9. Now, we show it for semi-relevant clauses.

"$\Rightarrow$" Let L be a ground literal with $L \in \mathsf{conflict}(N) \setminus \mathsf{conflict}(N \setminus \{C\})$. 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 $N_2 \models \mathsf{comp}(L)$ where $N_1 \cup N_2$ contains at least one instance of C, for otherwise $L \notin \mathsf{conflict}(N) \setminus \mathsf{conflict}(N \setminus \{C\})$. By the deductive completeness, Th. 1.3.2, and the fact that L and $\mathsf{comp}(L)$ 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 $K_2\sigma = \mathsf{comp}(L)$ 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 $\mathsf{comp}(L)$ respectively. By Lemma 3.1.8, L and $\mathsf{comp}(L)$ are also conflict literals in $M\sigma$. So, by Def. 3.1.10, the clause containing $\mathsf{comp}(L)$ 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 3.1.8, 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, $L \in \mathsf{conflict}(N) \setminus \mathsf{conflict}(N \setminus \{C\})$ and thus C is semi-relevant. □

In the case of a ground MUS, all of the literals in it are conflict literals. However, identifying the conflict literals of a full first-order clause set is not trivial. A naive approach is by enumerating all MUSes and checking if L is contained in some. This works for propositional logic even though it is 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 via resolution with the set-of-support strategy.

Lemma 3.3.6. 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 $(N, \{L \lor \mathsf{comp}(L)\})$.

Proof. "$\Rightarrow$" By the deductive completeness, Th. 1.3.2, and the fact that L and $\mathsf{comp}(L)$ 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 $K_2\sigma = \mathsf{comp}(L)$ 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 $\pi_1.\pi_2.(L \lor \neg L).(\mathsf{comp}(L)).\bot$ where K_2 is resolved with $L \lor \mathsf{comp}(L)$ to get $\mathsf{comp}(L)$ which will be resolved with K_1 from π_1 to get $\bot$. By Th. 3.2.7, it means there is an SOS refutation from $(N, \{L \lor \neg L\})$
"$\Leftarrow$" Given an SOS refutation π using $\{L \lor \mathsf{comp}(L)\}$, i.e., an SOS refutation π from $N \setminus \{\{L \lor \mathsf{comp}(L)\}\}$ with SOS $\{\{L \lor \mathsf{comp}(L)\}\}$, 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 3.1.8, K and $\mathsf{comp}(K)$ are also conflict literals in $N'\sigma \setminus M$. So, by Def. 3.1.10, 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 3.1.8, L is a conflict literal. □

As a final remark, the fact that the clause sets are used in three different settings should not cause any issues: (1) In the derivations generated by the resolution calculus, clauses can be used multiple times (via possibly different instantiations) and some generated clauses may even be useless for the final refutation; (2) For the transformation, refutations are assumed to be in tree-form where clauses are ground and used exactly once; (3) In the relation between syntactic and semantic relevance, the coincidence between them must be predicated upon the fact that the clause sets are independent. Derivations and tree-shaped ground refutations from (1) and (2) are related via the overall substitution. The fact that clauses are used only once in the trees is helpful to prove the generalized SOS completeness by enabling μ from Def. 3.2.4 to strictly decrease due to the transformation (Lemma 3.2.6). Besides the copying only happens for clauses that are already in the SOS and therefore does not affect the SOS measure. For (3), defining semantic relevance for independent clause sets means we do not even concern multiple clauses that can interchangeably be used to derive a conflict literal. For example, in the clauses $N = \{P(a,b), \neg P(a,x), \neg P(y,b)\}$, we cannot relate the syntactic semi-relevance (Def. 3.1.4) to its semantic characterization (Def. 3.1.11) because $\neg P(a,x)$ and $\neg P(y,b)$ are dependent w.r.t each other.

3.4 Syntactic Relevance Notion in Description Logic

In this section, I demonstrate how the proposed notion of syntactic relevance can be useful in explaining entailment for fragments with a translation technique to FOL [HKTW20]. As an illustration, I will use the description logic $\mathcal{ALC}$.

Lifting the Syntactic Relevance to DL Axioms

This works via translation via Table 1.2. Whenever we want to prove the entailment of a DL axiom φ from an ontology $\mathcal{O}$, $\mathcal{O} \models \varphi$, we refute the clausification $\mathtt{fo}(\mathcal{O}) \cup \mathtt{fo}(\neg\varphi)$ using the resolution calculus.

Definition 3.4.1 (DL Syntactic Relevance)**.** Given an entailment $\mathcal{O} \models \phi$ and a clausification Φ of $\mathtt{fo}(\mathcal{O}) \cup \mathtt{fo}(\neg\varphi)$, the DL axiom ϕ is

- *syntactically relevant*, if any refutation of Φ uses a clause from $\mathtt{fo}(\phi)$,
- *syntactically semi-relevant*, if a refutation of Φ uses a clause from $\mathtt{fo}(\phi)$, and
- *syntactically irrelevant*, otherwise

a silent labeling of clauses is assumed to keep track of different versions of the same clause.

The silent labeling of clauses is needed since multiple DL axioms may produce some equivalent clauses [LAWRS07]. Note that a relevant DL axiom may translate into several clauses consisting of more than one semi-relevant clauses which are not relevant individually.

Our notion of relevance can be effectively tested for a number of description logics, including $\mathcal{EL}$ and $\mathcal{ALC}$.

Many description logics satisfy the *finite model property*, in which the relevant finite model for some clause set can be explicitly a priori generated [HS02]. In this case, the logics enjoy the *bounded model property.* In particular, if resolution is not a decision procedure for the logic under consideration, an explicit bound on the Herbrand universe is needed.

In [HS02], the bounded model property of $\mathcal{EL}$, $\mathcal{ALC}$, and other logics is used to provide for a translation-based decision procedure for these logics. In general, this approach can however be used for many description logic fragments that have the finite model property, including more expressive description logics such as $\mathcal{SHOI}$.

Lemma 3.4.2 (Syntactic (Semi)-Relevance Decidability in Description Logics)**.** For ontologies in a DL fragment that enjoys the bounded model property, syntactic (semi-)relevance of an axiom for a given property is decidable.

Proof. We translate both the (negated) property and the ontology to first-order logic. For relevance, we first check satisfiability of the resulting clause set without the clauses from the axiom via resolution using the bounded model property. This terminates because we only have to consider terms out of the finite Herbrand base generated by these logics [HS02]. If the clause set is unsatisfiable, the axiom is either irrelevant or semi-relevant but not relevant. If this clause set is satisfiable the axiom is either relevant or irrelevant because then semi-relevance implies relevance. In both cases, testing for semi-relevance provides the final classification of the axiom.

To test for semi-relevance, we perform an SOS resolution proof attempt where the set of support contains the clauses corresponding to the axiom. If this results in a refutation, the axiom is semi-relevant or relevant for the property depending on the previous test, otherwise it is irrelevant.

All resolution proof attempts terminate, because they can be stopped once the generated terms exceed the expected bounded Herbrand universe. □

Example 3.4.3. As an illustration for our notion of relevance in the DL context, we consider an $\mathcal{EL}$ ontology $\mathcal{O} = \mathcal{T} \cup \mathcal{F}$, where

$$\begin{aligned}\mathcal{T} = \{&\mathsf{LuxurySedan} \sqcap \exists \mathsf{hasEngine}.\mathsf{HighPerformanceEngine} \sqsubseteq \mathsf{PerformanceCar},\\ &\mathsf{LuxurySedan} \sqcup \mathsf{PerformanceCar} \sqsubseteq \mathsf{ExecutiveCar}\}\end{aligned}$$

In natural language, the first axiom says, a luxurious sedan with a high performance engine is a performance car. The second one says, luxurious sedans and performance cars are executive cars. Suppose, in addition, we also have an ABox:

$$\begin{aligned}\mathcal{F} = \{&\mathsf{LuxurySedan}(\mathsf{mercedes}),\\ &\mathsf{hasEngine}(\mathsf{mercedes}, \mathsf{v8}),\\ &\mathsf{HighPerformanceEngine}(\mathsf{v8})\\ &\mathsf{PerformanceCar}(\mathsf{lamborghini})\}.\end{aligned}$$

The first-order translation of $\mathcal{T}$ according to Table 1.2 results in

$$\neg\mathsf{LuxurySedan}(x) \vee \neg\mathsf{hasEngine}(x, z) \vee \neg\mathsf{HighPerformanceEngine}(z) \vee \mathsf{PerformanceCar}(x)$$
$$(\neg\mathsf{LuxurySedan}(x) \wedge \neg\mathsf{PerformanceCar}(x)) \vee \mathsf{ExecutiveCar}(x)$$

and we want to prove the entailment $\mathcal{O} \models \mathsf{ExecutiveCar}(\mathsf{mercedes})$. In order to find (semi-)relevant axioms for this entailment, we consider $\neg\mathsf{ExecutiveCar}(\mathsf{mercedes})$ and the translation of $\mathcal{O}$ to first-order logic.

The following refutation π_1 (in linear format) can be used to show that $\mathcal{O} \models \mathsf{ExecutiveCar}(\mathsf{mercedes})$.

$$\begin{aligned} \pi_1 = [(1) &: \neg\mathsf{LuxurySedan}(\mathsf{x}) \vee \mathsf{ExecutiveCar}(\mathsf{x}) \\ (2) &: \mathsf{LuxurySedan}(\mathsf{mercedes}), \\ (3) &: \mathsf{ExecutiveCar}(\mathsf{mercedes}) \\ (4) &: \neg\mathsf{ExecutiveCar}(\mathsf{mercedes}) \\ (5) &: \bot]. \end{aligned}$$

π_1 is not the only possible refutation. Another refutation with different clauses is as follows.

$$\begin{aligned} \pi_2 = [(1) &: \neg\mathsf{LuxurySedan}(x) \vee \\ &\quad \neg\mathsf{hasEngine}(x, z) \vee \neg\mathsf{HighPerformanceEngine}(z) \vee \\ &\quad \mathsf{PerformanceCar}(x), \\ (2) &: \neg\mathsf{PerformanceCar}(\mathsf{x}) \vee \mathsf{ExecutiveCar}(\mathsf{x}) \\ (3) &: \mathsf{LuxurySedan}(\mathsf{mercedes}), \\ (4) &: \mathsf{hasEngine}(\mathsf{mercedes}, \mathsf{v8}), (5) : \mathsf{HighPerformanceEngine}(\mathsf{v8}), \\ (6) &: \neg\mathsf{LuxurySedan}(x) \vee \\ &\quad \neg\mathsf{hasEngine}(x, z) \vee \neg\mathsf{HighPerformanceEngine}(z) \vee \\ &\quad \mathsf{ExecutiveCar}(x), \\ (7) &: \neg\mathsf{hasEngine}(\mathsf{mercedes}, z) \vee \neg\mathsf{HighPerformanceEngine}(z) \vee \\ &\quad \mathsf{ExecutiveCar}(\mathsf{mercedes}), \\ (8) &: \neg\mathsf{HighPerformanceEngine}(\mathsf{v8}) \vee \mathsf{ExecutiveCar}(\mathsf{mercedes}), \\ (9) &: \mathsf{ExecutiveCar}(\mathsf{mercedes}) \\ (10) &: \neg\mathsf{ExecutiveCar}(\mathsf{mercedes}) \\ (11) &: \bot]. \end{aligned}$$

The clauses (1) to (5) are the input clauses. The next four clauses result from consecutive resolution steps between clause (1) and the other clauses (2) to (5) from $\tau(\mathcal{A})$. This will result in the respective clauses (6) to (9). Clause (9) is then resolved with the negated conjecture, (10), resulting in $\bot$.

The concept inclusion $\mathsf{LuxurySedan} \sqcup \mathsf{PerformanceCar} \sqsubseteq \mathsf{ExecutiveCar}$ is therefore syntactically relevant since both proofs must contain at least one clause out of its first-order translation. The ABox axiom $\mathsf{LuxurySedan}(\mathsf{mercedes})$ is also syntactically relevant. Moreover, the TBox axioms

$$\mathsf{LuxurySedan} \sqcap \exists \mathsf{hasEngine}.(\mathsf{HighPerformanceEngine}) \sqsubseteq \mathsf{PerformanceCar}$$

and the ABox axioms

$$\begin{aligned} &\mathsf{LuxurySedan}(\mathsf{mercedes}), \\ &\mathsf{hasEngine}(\mathsf{mercedes}, \mathsf{v8}), \\ \text{and } &\mathsf{HighPerformanceEngine}(\mathsf{v8}) \end{aligned}$$

are syntactically semi-relevant but not syntactically relevant. Finally, the ABox axiom

$$\mathsf{PerformanceCar}(\mathsf{lamborghini})$$

is irrelevant.

Comparison between Axioms Involved in Refutations and Justifications

Analogous to the comparison between MUS-based and refutation based relevance in FOL, we now compare our notions with the one using justification. Consider the ontology $\mathcal{O} = \mathcal{T} \cup \mathcal{F}$ consisting of the following TBox and ABox.

$$\mathcal{T} = \{P \sqsubseteq Q, Q \sqcap R \sqsubseteq S, Q \sqcup S \sqsubseteq T\}$$
$$\mathcal{F} = \{P(a), R(a)\}$$

and consider the entailment $\mathcal{O} \models T(a)$. There is only one justification for this entailment, namely $\{P \sqsubseteq Q, Q \sqcup S \sqsubseteq T, P(a)\}$ so that only those axioms are MinA-relevant for $T(a)$. In contrast, all axioms in the ontology are syntactically semi-relevant for $T(a)$. A first proof in first-order logic uses, in addition to the negation of $T(a)$ the axioms $P(a)$, $P \sqsubseteq Q$, and $Q \sqcup S \sqsubseteq T$ as is shown in Fig. 3.8.

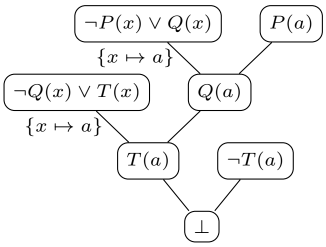

Figure 3.8: A refutation in FOL for the $\mathcal{ALC}$ ontology $\mathcal{T} \uplus \mathcal{F}$ using $P(a)$

The second proof also uses $P(a)$, $P \sqsubseteq Q$, and $Q \sqcup S \sqsubseteq T$, but additionally includes $R(a)$ and $Q \sqcap R \sqsubseteq S$ as shown in Fig. 3.9.

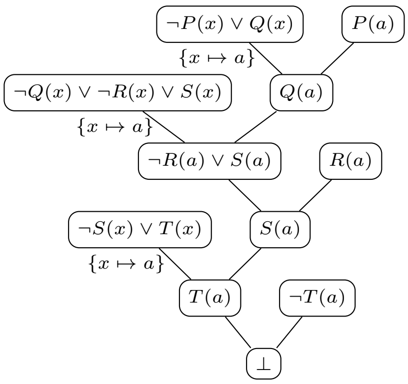

Figure 3.9: A refutation in FOL for the $\mathcal{ALC}$ ontology $\mathcal{T} \uplus \mathcal{F}$ using $R(a)$

Due to the fact that syntactic semi-relevance is based on the first-order translation of its axioms, this notion then possesses a connection to *laconic justifications* [HPS08b]. On the one hand, a syntactically semi-relevant DL axiom may be translated into several clauses where some of them are irrelevant at first-order level. On the other hand, a laconic justification can be seen as resulting from some sort of axiom simplifications (such as conjunct elimination) in a justification. In some way, a simplification of DL axiom would amount to irrelevant clauses elimination.

Relation between Syntactically Relevant Axioms and DL Repairs

Relevant axioms are related to the (classical) *repairs* where we want to remove some unwanted consequence. A *repair* for an entailment $\mathcal{O} \models \alpha$ is a subset-minimal set $\mathfrak{R} \subseteq \mathcal{O}$ s.t. $\mathcal{O} \setminus \mathfrak{R} \not\models \alpha$ [PSK05]. In this case, a repair for $\mathcal{O} \models \alpha$ then always contains one axiom from every justification for $\mathcal{O} \models \alpha$. In the context of syntactic relevance, a relevant axiom corresponds to the special case of a repair of size one. Thus relevancy test also provides a test for whether a set of size one is a repair. This can be an alternative approach to the existing justification-based method (via minimal hitting set algorithm) [Rei87].

Relations between Syntactically Semi-Relevant Axioms and Lean Kernel/ MinA-preserving module

Syntactically semi-relevant axioms are related to the following: lean kernel [KK09, Kul00] and MinA preserving module [PMIM17]. Given an unsatisfiable set F of propositional clauses, the lean kernel $N_a(F) \subseteq N$ consists exactly of the clauses involved in some refutation of N and thus, the syntactically semi-relevant clauses. This notion has been extended to description logics in [PMIM17] where arbitrary consequence-based reasoning procedures are considered.

The notion of lean kernal is generalized to *MinA-preserving module* [PMIM17]: given an ontology $\mathcal{O}$ and an axiom α, a MinA-preserving module for α in $\mathcal{O}$ is a subset $\mathcal{M} \subseteq \mathcal{O}$ s.t. every justification for $\mathcal{O} \models \alpha$ is a subset of $\mathcal{M}$. Subset minimality is not imposed and thus a MinA-preserving module may contain axioms outside of any justification. Since every axiom in a justification is also semi-relevant, the set of all syntactically semi-relevant axioms is also a MinA-preserving module.

Chapter 4

Explaining Non-Entailment via Connection-Minimal Abduction

As we have seen in Sect. 2.2, explaining a non-entailment can be done in many ways. One that I will focus on is abduction in $\mathcal{EL}$. For this, I introduce a new minimality criterion called *connection minimality* (Sect. 4.1). This criterion characterizes hypotheses for $\mathcal{T}$ and α that connect the left- and right-hand sides of the observation α directly without creating spurious connections involving unrelated CIs.

To generate the connection-minimal hypotheses, I present a sound and complete approach using first-order prime implicates. This consists of three steps (Sect. 4.2). First, the abduction problem is translated to first-order clause set Φ. Second, the prime implicates of Φ are generated, that is, a set of minimal logical consequences of Φ that subsume all other consequences of Φ. Last, I construct the wanted hypotheses from these generated prime implicates.

For the implementation, the SPASS theorem prover [WSH$^+$07] is used as a restricted SOS resolution [WRC65, HTW21] engine for the computation of prime implicates (Sect. 4.3.1). At the DL level, efficiency is further improved by performing some preprocessing steps in Java also taking advantage of the OWL API [HB11] and the DL tool ELK [KKS14] (Sect. 4.3.2). Termination is guaranteed for the class of hypotheses that are subset minimal (Sect. 4.4). The resulting implementation is called CAPI (Sect. 4.5.1) and I used it to perform experiments on some publicly available bio-medical ontologies in $\mathcal{EL}$ (Sect. 4.5.2). The results show that, despite the possibly high theoretical cost, it is not prohibitive in practice.

4.1 A New Minimality Notion for $\mathcal{EL}$ Abduction

I will now introduce the *connection-minimality* notion. This section is divided into two parts. The first one offers an alternative way of characterizing the entailment of a concept inclusion via $\mathcal{T}$-homomorphism and connecting concept. Afterwards, I introduce the connection-minimality notion taking advantage of this entailment characterization. Now, I recall the example in page 10 where $\mathcal{H}_{a1}$ is the one that is connection-minimal.

Example 4.1.1 (Academia abduction)**.**

$$\begin{aligned}\mathcal{T}_a \quad = \{ \quad & \exists \mathsf{employment.ResearchPosition} \sqcap \exists \mathsf{qualification.Diploma} \sqsubseteq \mathsf{Researcher}, \\ & \exists \mathsf{writes.ResearchPaper} \sqsubseteq \mathsf{Researcher},\ \mathsf{Doctor} \sqsubseteq \exists \mathsf{qualification.PhD}, \\ & \mathsf{Professor} \equiv \mathsf{Doctor} \sqcap \exists \mathsf{employment.Chair}, \\ & \mathsf{FundsProvider} \sqsubseteq \exists \mathsf{writes.GrantApplication}\ \}\end{aligned}$$

The observation $\alpha_a = \mathsf{Professor} \sqsubseteq \mathsf{Researcher}$ is not entailed by $\mathcal{T}_a$ with considered hypotheses

$$\begin{aligned}\mathcal{H}_{a1} \quad &= \{ \quad \mathsf{Chair} \sqsubseteq \mathsf{ResearchPosition},\ \mathsf{PhD} \sqsubseteq \mathsf{Diploma}\} \text{ and} \\ \mathcal{H}_{a2} \quad &= \{ \quad \mathsf{Professor} \sqsubseteq \mathsf{FundsProvider},\ \mathsf{GrantApplication} \sqsubseteq \mathsf{ResearchPaper}\}\end{aligned}$$

This example will be referred to a couple of times later.

4.1.1 Acquiring $\mathcal{EL}$ Entailment via Connecting Concept

Ultimately, the goal of an abductive reasoning is to obtain an entailment. In relation to the proposed connection minimality, I introduce in this section the characterization of subsumption via connecting concepts and how it can be constructed using $\mathcal{T}$-homomorphism.

Connecting concept

I provide a simple yet useful notion of *connecting concept*. The idea is that if we have an entailment $\mathcal{T} \models U_1 \sqsubseteq U_2$, then there is something in between which "connects" them. Otherwise, they are disconnected (which is the case in the abduction problem). Connection-minimality then constructs such a connection. The notion of connecting concepts is formalized in the following.

Definition 4.1.2. Let U_1 and U_2 be concepts. A concept V *connects* U_1 to U_2 in $\mathcal{T}$ if and only if $\mathcal{T} \models U_1 \sqsubseteq V$ and $\mathcal{T} \models V \sqsubseteq U_2$. D is thus called a *connecting concept* between U_1 and U_2.

Note that if $\mathcal{T} \models U_1 \sqsubseteq U_2$ then both U_1 and U_2 are connecting concepts from U_1 to U_2, and if $\mathcal{T} \not\models U_1 \sqsubseteq U_2$ neither of them are. Note that a connecting concept may have removable conjuncts. In this case, it is preferable to find one that is minimal. So, we define a partial order $\preceq_\sqcap$ on concepts, s.t. $U \preceq_\sqcap V$ if we can turn V into U by removing conjuncts in subexpressions, e.g., $\exists R'.Q \preceq_\sqcap \exists R.P \sqcap \exists R'.(Q \sqcap Q')$. The following definition shows its formalization.

Definition 4.1.3. Let U and V be arbitrary concepts. Then $U \preceq_\sqcap V$ if either:

- $U = V$,
- $V = V' \sqcap V''$, and $U \preceq_\sqcap V'$, or
- $U = \exists R.U'$, $V = \exists R.V'$ and $U' \preceq_\sqcap V'$.

A concept V is $\preceq_\sqcap$-minimal w.r.t. some property if there is no other I s.t. $U \preceq_\sqcap V$ satisfying the same property.

For subsumers, we have the following property: if $\mathcal{T} \models U \sqsubseteq V$, then for any V' s.t. $V' \preceq_\sqcap V$, it holds that $\mathcal{T} \models U \sqsubseteq V'$. Intuitively any possible conjunct removal from V preserves subsumption. In contrast, for subsumees, we have the following property: if $\mathcal{T} \models U \sqsubseteq V$, then for any U' s.t. $U \preceq_\sqcap U'$, it holds that $\mathcal{T} \models U' \sqsubseteq V$. Any addition of conjuncts in U preserves the subsumption relation. So, in our case, it is preferable to remove conjuncts from the subsumee to get the gist of the connecting concept.

Constructing a connecting concept via $\mathcal{T}$-homomorphism

Now, by looking into the syntactic characterization of non-atomic concepts, we can create the connection by means of a new $\mathcal{T}$-homomorphism. In some sense, a connection-minimal hypothesis would then provide the "missing link" in the $\mathcal{T}$-homomorphism. Semantically, subsumption $U \sqsubseteq V$ between two concepts means there is subset inclusion between $U^\mathcal{I} \subseteq V^\mathcal{I}$ for any $\mathcal{I}$. In [BKM99], this can be syntactically characterized via the existence of some notion of homomorphism between the $\mathcal{EL}$ *description trees* of U and V. For this, I generalize a subsumption characterization from [BKM99] by means of homomorphism between $\mathcal{EL}$ description trees.

Our new notion of homomorphism also relies on the notion of $\mathcal{EL}$ description tree. Given a concept U, we can construct a tree where the nodes hold the information regarding the atomic concepts in U while the edges represent roles (originally from Baader et al. [BKM99]). $\mathcal{EL}$ description trees capture the syntax of concepts graphically.

Definition 4.1.4. An $\mathcal{EL}$ *description tree* is a labeled tree $\mathfrak{T} = (\mathcal{V}, \mathcal{E}, v_0, l)$ where $\mathcal{V}$ is a set of nodes with root $v_0 \in \mathcal{V}$, the nodes $v \in \mathcal{V}$ are labeled with $l(v) \subseteq \Omega_\mathsf{C}$, and the (directed) edges $vRw \in \mathcal{E}$ are such that $v, w \in \mathcal{V}$ and are labeled with $R \in \Omega_\mathsf{R}$.

Relation between $\mathcal{EL}$ description trees and concepts Given a tree $\mathfrak{T} = (\mathcal{V}, \mathcal{E}, v_0, l)$ and $v \in \mathcal{V}$, we denote by $\mathfrak{T}(v)$ the subtree of $\mathfrak{T}$ that is rooted at v. If $l(v_0) = \{A_1, \ldots, A_k\}$ and $v_1, \ldots, v_n$ are all the children of v_0, we can define the concept represented by $\mathfrak{T}$ recursively using $U_\mathfrak{T} = P_1 \sqcap \ldots \sqcap P_k \sqcap \exists R_1.U_{\mathfrak{T}(v_1)} \sqcap \ldots \sqcap \exists R_l.U_{\mathfrak{T}(v_l)}$ where for $j \in \{1, \ldots, n\}$, $v_0 R_j v_j \in \mathcal{E}$. Conversely, we can define $\mathfrak{T}_U$ for a concept $U = P_1 \sqcap \ldots \sqcap P_k \sqcap \exists R_1.U_1 \sqcap \ldots \sqcap \exists R_n.U_n$ inductively based on the pairwise disjoint description trees $\mathfrak{T}_{U_i} = \{\mathcal{V}_i, \mathcal{E}_i, v_i, l_i\}$, $i \in \{1, \ldots, n\}$. Specifically, $\mathfrak{T}_U = (\mathcal{V}_U, \mathcal{E}_U, v_U, l_U)$, where

$$\mathcal{V}_U = \{v_0\} \cup \textstyle\bigcup_{i=1}^n \mathcal{V}_i, \qquad l_U(v) = l_i(v) \text{ for } v \in \mathcal{V}_i,$$
$$\mathcal{E}_U = \{v_0 R_i v_i \mid 1 \leq i \leq n\} \cup \textstyle\bigcup_{i=1}^n \mathcal{E}_i, \qquad l_U(v_0) = \{P_1, \ldots, P_k\}.$$

We are now ready to introduce the $\mathcal{T}$-homomorphism generalizing the notion from [BKM99]. If $\mathcal{T} = \emptyset$, this corresponds exactly to the one introduced in [BKM99].

Definition 4.1.5. Let $\mathfrak{T}_1 = (\mathcal{V}_1, \mathcal{E}_1, v_0, l_1)$ and $\mathfrak{T}_2 = (\mathcal{V}_2, \mathcal{E}_2, w_0, l_2)$ be two description trees and $\mathcal{T}$ a TBox. A mapping $\phi : \mathcal{V}_2 \to \mathcal{V}_1$ is a $\mathcal{T}$*-homomorphism* from $\mathfrak{T}_2$ to $\mathfrak{T}_1$ if and only if the following conditions are satisfied:

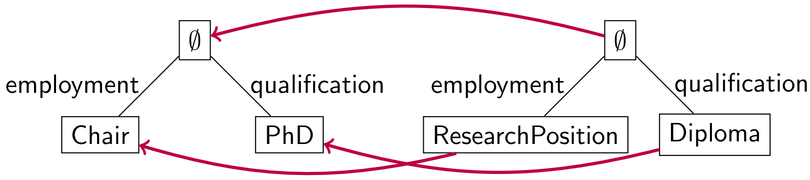

Figure 4.1: Description trees of V_1 (left) and V_2 (right).

1. $\phi(w_0) = v_0$
2. $\phi(v)R\phi(w) \in \mathcal{E}_1$ for all $vRw \in \mathcal{E}_2$
3. for every $v \in \mathcal{V}_1$ and $w \in \mathcal{V}_2$ with $v = \phi(w)$, $\mathcal{T} \models \bigsqcap l_1(v) \sqsubseteq \bigsqcap l_2(w)$

If only 1 and 2 are satisfied, then ϕ is called a *weak* homomorphism.

$\mathcal{T}$-homomorphisms for a given TBox $\mathcal{T}$ capture subsumption w.r.t. $\mathcal{T}$. If there exists a $\mathcal{T}$-homomorphism ϕ from $\mathfrak{T}_2$ to $\mathfrak{T}_1$, then $\mathcal{T} \models U_{\mathfrak{T}_1} \sqsubseteq U_{\mathfrak{T}_2}$. This can be shown easily by structural induction using the definitions. Weak homomorphisms are used to reveal missing links between a subsumee V_2 of U_2 and a subsumer V_1 of U_1, that can be added using a hypothesis.

Lemma 4.1.6. Let $\mathfrak{T}_1 = (\mathcal{V}_1, \mathcal{E}_1, v_0, l_1)$ and $\mathfrak{T}_2 = (\mathcal{V}_2, \mathcal{E}_2, w_0, l_2)$ be $\mathcal{EL}$ description trees, with a $\mathcal{T}$-homomorphism Φ from $\mathfrak{T}_2$ to $\mathfrak{T}_1$. Then $\mathcal{T} \models U_{\mathfrak{T}_1} \sqsubseteq U_{\mathfrak{T}_2}$.

Proof. We prove this result by induction on the structure of $\mathfrak{T}_2$.

If $\mathfrak{T}_2 = (\{w_0\}, \emptyset,\ w_0, l_2)$, then $U_{\mathfrak{T}_2} = l_2(w_0)$. Moreover $U_{\mathfrak{T}_1} \sqsubseteq l_1(v_0)$ by definition of $U_{\mathfrak{T}_1}$. Finally $\mathcal{T} \models U_{\mathfrak{T}_1} \sqsubseteq U_{\mathfrak{T}_2}$ since $\mathcal{T} \models \bigsqcap l_1(\phi(w_0)) \sqsubseteq \bigsqcap l_2(w_0)$ and $\phi(w_0) = v_0$.

In the general case, let us consider any child w_i of w_0 in $\mathfrak{T}_2$ since there must be at least one. Then there is a corresponding child v_i of v_0 in $\mathfrak{T}_1$ s.t. $v = \phi(w)$. The $\mathcal{T}$-homomorphism ϕ from $\mathfrak{T}_2$ to $\mathfrak{T}_1$ is also a $\mathcal{T}$-homomorphism from $\mathfrak{T}_2(w)$ to $\mathfrak{T}_1(v)$, thus by induction $\mathcal{T} \models U_{\mathfrak{T}_1(v)} \sqsubseteq U_{\mathfrak{T}_2(w)}$, and in particular, for the R_i such that $w_0 R_i w \in \mathcal{E}_2$, we have $\mathcal{T} \models \exists R_i.U_{\mathfrak{T}_1(v)} \sqsubseteq \exists R_i.U_{\mathfrak{T}_2(w)}$. This applies to all the children $w_1, \ldots, w_n$ of w_0, and since $\mathcal{T} \models \bigsqcap l_1(v_0) \sqsubseteq \bigsqcap l_2(w_0)$, it follows that $\mathcal{T} \models \bigsqcap U_{\mathfrak{T}_1} \sqsubseteq U_{\mathfrak{T}_2}$. □

In the end a connection-minimal hypothesis for an abduction problem creates the connection between the concepts U_1 and U_2. As argued above, this is established via concepts V_1 and V_2 that satisfy $\mathcal{T} \models U_1 \sqsubseteq V_1$, $V_2 \sqsubseteq U_2$.

Example 4.1.7. Consider the concepts

$$V_1 = \exists \mathsf{employment}.\mathsf{Chair} \sqcap \exists \mathsf{qualification}.\mathsf{PhD}$$
$$V_2 = \exists \mathsf{employment}.\mathsf{ResearchPosition} \sqcap \exists \mathsf{qualification}.\mathsf{Diploma}$$

from Ex. 4.1.1. Fig. 4.1 illustrates description trees for V_1 (left) and V_2 (right). The curved arrows show a weak homomorphism from $\mathfrak{T}_{V_2}$ to $\mathfrak{T}_{V_1}$ that can be strengthened into a $\mathcal{T}$-homomorphism for some TBox $\mathcal{T}$ that corresponds to the set of CIs in $\mathcal{H}_{a1} \cup \{\top \sqsubseteq \top\}$.

4.1.2 Connection-Minimality Notion

Before getting into the formalization, I present a TBox abduction problem that is slightly different from the one usually formulated in the literature. In the existing literature, TBox abduction normally restricts the hypotheses to consists of only simple concept inclusion of the form $P \sqsubseteq Q$. For our notion, it is somewhat too restrictive and we need to relax it by allowing conjunctions. The problem would then be formulated as follows.

Definition 4.1.8 (TBox Abduction)**.** An *$\mathcal{EL}$ TBox abduction problem* (shortened to *abduction problem*) is a tuple $\langle \mathcal{T}, \Sigma, U_1 \sqsubseteq U_2 \rangle$, where $\mathcal{T}$ is a TBox called the *background knowledge*, Σ is a set of atomic concepts called the *abducible signature*, and $U_1 \sqsubseteq U_2$ is a CI called the *observation*, s.t. $\mathcal{T} \not\models U_1 \sqsubseteq U_2$. A solution to this problem is a TBox

$$\mathcal{H} \subseteq \{P_1 \sqcap \cdots \sqcap P_n \sqsubseteq Q_1 \sqcap \cdots \sqcap Q_m \mid \{P_1, \ldots, P_n, Q_1, \ldots, Q_m\} \subseteq \Sigma\}$$

where $m > 0$, $n \geq 0$ and such that $\mathcal{T} \cup \mathcal{H} \models C_1 \sqsubseteq C_2$ and, for all CIs $\alpha \in \mathcal{H}$, $\mathcal{T} \not\models \alpha$. A solution to an abduction problem is called a *hypothesis*.

Existing minimality notions suffer from a certain limitation in that they may unnecessarily use concepts that are completely unrelated to the observation. This is not a good idea so far as Occam's razor is concerned. In other words, the common minimality notions are not parsimonious as already illustrated in Ex. 4.1.1.

To address the lack of parsimony in the common minimality notions, I now introduce the *connection minimality.* Intuitively, the connection minimality only accepts those hypotheses ensuring that every concept inclusion in the hypothesis is connected to both U_1 and U_2 in $\mathcal{T}$, as is the case for $\mathcal{H}_{\text{a1}} = \{\mathsf{Chair} \sqsubseteq \mathsf{ResearchPosition}, \mathsf{PhD} \sqsubseteq \mathsf{Diploma}\}$ in Ex. 4.1.1. The definition of connection minimality is based on the following ideas:

1 Hypotheses for the abduction problem have to create a *connection* between U_1 and U_2, in the form of a concept V that satisfies $\mathcal{T} \cup \mathcal{H} \models U_1 \sqsubseteq V$, $V \sqsubseteq U_2$.

2 To ensure that Occam's razor is followed, we want this connection to be based on concepts V_1 and V_2 for which we already have $\mathcal{T} \models U_1 \sqsubseteq V_1$ and $\mathcal{T} \models V_2 \sqsubseteq U_2$.

3 We additionally want to make sure that the connecting concepts are not more complex than necessary, and that $\mathcal{H}$ only contains CIs that directly connect parts of V_2 to parts of V_1 by closely following their structure.

Definition 4.1.9 (Connection-Minimal Abduction)**.** Given an abduction problem $\langle \mathcal{T}, \Sigma, U_1 \sqsubseteq U_2 \rangle$, a hypothesis $\mathcal{H}$ is *connection-minimal* if there exist concepts V_1 and V_2 built over $\Sigma \cup \Omega_{\mathsf{R}}$ and a mapping ϕ satisfying each of the following conditions:

1. $\mathcal{T} \models U_1 \sqsubseteq V_1$,

2. V_2 is a $\preceq_{\sqcap}$-minimal concept s.t. $\mathcal{T} \models V_2 \sqsubseteq U_2$,

3. ϕ is a weak homomorphism from the tree $\mathfrak{T}_{V_2} = (V_2, E_2, w_0, l_2)$ to the tree $\mathfrak{T}_{V_1} = (V_1, E_1, v_0, l_1)$, and

4. $\mathcal{H} = \{\bigsqcap l_1(\phi(w)) \sqsubseteq \bigsqcap l_2(w) \mid w \in V_2 \wedge \mathcal{T} \not\models \bigsqcap l_1(\phi(w)) \sqsubseteq \bigsqcap l_2(w)\}$.

$\mathcal{H}$ is additionally called *packed* if the left-hand sides of the CIs in $\mathcal{H}$ cannot hold more conjuncts than they do, which is formally stated as: for $\mathcal{H}$, there is no $\mathcal{H}'$ defined from the same V_2 and a V_1' and ϕ' s.t. there is a node $w \in V_2$ for which $l_1(\phi(w)) \subsetneq l_1'(\phi'(w))$ and $l_1(\phi(w')) = l_1'(\phi'(w'))$ for $w' \neq w$.

The straightforward consequences of Def. 4.1.9 include that ϕ is a $(\mathcal{T} \cup \mathcal{H})$-homomorphism from $\mathfrak{T}_{V_2}$ to $\mathfrak{T}_{V_1}$ and that V_1 and V_2 are connecting concepts from U_1 to U_2 in $\mathcal{T} \cup \mathcal{H}$ so that $\mathcal{T} \cup \mathcal{H} \models U_1 \sqsubseteq U_2$ as wanted.

While minimality notions from Def. 2.2.2 with external information can be very important in practice (e.g., for debugging), it is in some way difficult to compare our notion with them due to the differing assumptions. On the one hand, the idea of "connectedness" relies only on the TBox without the necessity of external help while on the other hand, external information such as an oracle is often the key in controlling the preferred hypotheses but we have no means to assess their quality (because they are simply given). For example, the signature restriction (equipped with semantic minimality) must exclude some signature in order to be meaningful (otherwise only $\{U_1 \sqsubseteq U_2\}$ is acceptable). On the other hand, $\{U_1 \sqsubseteq U_2\}$ is perfectly acceptable w.r.t our notion. In the trivial abduction problem with empty Tbox $\emptyset \not\models U_1 \sqsubseteq U_2$, even our notion would accept $\{U_1 \sqsubseteq U_2\}$. Moreover, I would further argue that relying on external information can be like passing the bucket. Having an external guide may amount to having another abduction problem at a different level (e.g., "why is this justification pattern meaningful?"). This can be fine, for example, when a domain expert is trying to fix an ontology. Therefore, I think it is not possible to have an apple-to-apple comparison.

So, it makes more sense to compare it with the notions relying on the problem itself such as the ones defined in Def. 2.2.3. On the one hand, our notion excludes a subset of the hypotheses accepted by the other minimality notions. On the other hand, our minimality also offers hypotheses that are not accepted by others. I show this in particular for the subset minimality notion.

Example 4.1.10 (Non-subset-minimal 1)**.**

$$\begin{aligned}\mathcal{T} = \{ \quad & \mathsf{Tycoon} \sqsubseteq \mathsf{CEO} \sqcap \mathsf{BusinessOwner} \\ & \mathsf{CEO} \sqsubseteq \exists \mathsf{manages.Company} \sqcap \mathsf{ShareHolder} \\ & \mathsf{BusinessOwner} \sqsubseteq \exists \mathsf{owns.Company} \\ & \exists \mathsf{owns.Enterprise} \sqsubseteq \mathsf{SuperRich} \\ & \exists \mathsf{manages.Enterprise} \sqcap \mathsf{DividendReceiver} \sqsubseteq \mathsf{SuperRich}\}\end{aligned}$$

if the observation is $\mathsf{CEO} \sqsubseteq \mathsf{SuperRich}$, then one possible hypothesis is

$$\mathcal{H}_1 = \{\mathsf{Company} \sqsubseteq \mathsf{Enterprise}, \mathsf{ShareHolder} \sqsubseteq \mathsf{DividendReceiver}\}$$

constructed from

$$V_1 = \exists \mathsf{manages.Company} \sqcap \mathsf{ShareHolder}$$
$$V_2 = \exists \mathsf{manages.Enterprise} \sqcap \mathsf{DividendReceiver}$$

If the observation is $\mathsf{BusinessOwner} \sqsubseteq \mathsf{SuperRich}$, then one possible hypothesis is

$$\mathcal{H}_2 = \{\mathsf{Company} \sqsubseteq \mathsf{Enterprise}\}$$

constructed from

$$V_1' = \exists \mathsf{owns}.\mathsf{Company}$$
$$V_2' = \exists \mathsf{owns}.\mathsf{Enterprise}$$

If the observation is $\mathsf{Tycoon} \sqsubseteq \mathsf{SuperRich}$, then both of the previous hypotheses are accepted but $\mathcal{H}_1$ is not subset minimal. This is because V_1 and V_2 are also usable for $\mathsf{Tycoon} \sqsubseteq \mathsf{SuperRich}$. In other words, our connection minimality criterion considers that both $\mathcal{H}_1$ and $\mathcal{H}_2$ are important for $\mathsf{Tycoon} \sqsubseteq \mathsf{SuperRich}$. In conjunction with the ontology, $\mathcal{H}_1$ explains why the observation holds in different manner than $\mathcal{H}_2$ and not just simply $\mathcal{H}_2$ with an additional superfluous concept inclusion. That is, hypothesis $\mathcal{H}_1$ is accepted when we explain the observation because tycoons are CEO's and hypothesis $\mathcal{H}_2$ is accepted if we consider tycoons to be business owners.

The second difference to consider is whether or not cycles (from Def. 1.2.1) are used for constructing the connecting concepts. In the following, a forward cycle and a backward cycle might turn out to be matching in a way that introduces an extra mapping in the homomorphism.

Example 4.1.11 (Non-subset-minimal 2 + Termination Issue)**.**

$$\begin{aligned}\mathcal{T} = \{&\mathsf{Bat} \sqsubseteq \exists \mathsf{canBite}.\mathsf{Bat},\\ &\mathsf{Bat} \sqsubseteq \exists \mathsf{canBite}.\mathsf{Human},\\ &\mathsf{Bat} \sqsubseteq \exists \mathsf{canHost}.\mathsf{Rabies},\\ &\mathsf{DiseaseVector} \sqsubseteq \mathsf{Organism},\\ &\mathsf{Human} \sqsubseteq \mathsf{Organism}\}\\ &\exists \mathsf{canBite}.\mathsf{Organism} \sqcap \exists \mathsf{canHost}.\mathsf{Virus} \sqsubseteq \mathsf{DiseaseVector}\}\end{aligned}$$

where we have a non entailment

$$\mathcal{T} \not\models \mathsf{Bat} \sqsubseteq \mathsf{DiseaseVector}$$

For $\mathsf{Bat} \sqsubseteq \mathsf{DiseaseVector}$, we could have the following connection-minimal hypotheses

$$\begin{aligned}\mathcal{H}_3 &= \{\mathsf{Bat} \sqsubseteq \mathsf{DiseaseVector}\}\\ \mathcal{H}_4 &= \{\mathsf{Rabies} \sqsubseteq \mathsf{Virus}\}\\ \mathcal{H}_5 &= \{\mathsf{Human} \sqsubseteq \mathsf{DiseaseVector}, \mathsf{Rabies} \sqsubseteq \mathsf{Virus}\}\end{aligned}$$

$\mathcal{H}_3$ is the obvious one because it is basically the observation. $\mathcal{H}_4$ represents the fact that bats transfer viruses to humans. The concepts used to create a connecting concept are

$$V_1 = \exists \mathsf{canBite}.\mathsf{Human} \sqcap \exists \mathsf{canHost}.\mathsf{Rabies}$$
$$V_2 = \exists \mathsf{canBite}.\mathsf{Human} \sqcap \exists \mathsf{canHost}.\mathsf{Virus}$$

$\mathcal{H}_5$ is a connection-minimal hypothesis with the following concepts as the ingredients for a connecting concept.

$$V_1' = \exists \mathsf{canBite.Human} \sqcap \exists \mathsf{canHost.Rabies}$$
$$V_2' = \exists \mathsf{canBite.DiseaseVector} \sqcap \exists \mathsf{canHost.Virus}$$

$\mathcal{H}_5$ also illustrates a termination problem. A rabies-carrying bat can transfer the rabies to a different bat indefinitely before finally reaching a human. In $\mathcal{EL}$ axiom, one can see that from $\mathsf{Bat} \sqsubseteq \exists \mathsf{canBite.Bat} \sqcap \exists \mathsf{canHost.Rabies}$, the atomic concept Bat can always be replaced indefinitely by $\mathsf{Bat} \sqcap \exists \mathsf{canHost.Rabies}$ producing successively larger concepts. The same goes with $\exists \mathsf{canBite.Organism} \sqcap \exists \mathsf{canHost.Virus}$. Every repetition always produces a different pair of V_1 and V_2 with which we can produce a connection-minimal solution. However, these infinite number of pairs will only be usable to generate $\mathcal{H}_5$.

4.2 Computation via First-Order Prime Implicates

In order to compute connection-minimal hypotheses in practice, I propose a method based on first-order prime implicates. This is illustrated in Fig. 4.2.

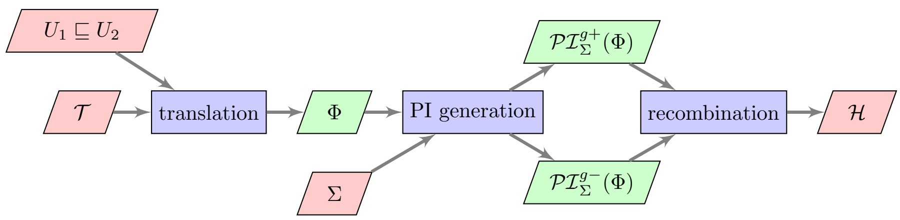

Figure 4.2: $\mathcal{EL}$ abduction using prime implicate generation in FOL.

The problem is first translated into a set Φ of Horn clauses. Prime implicates can be computed using an off-the-shelf tool [NIIR10, EPS18] or, in our case, a slight extension of the resolution-based version of the SPASS theorem prover [WSH+07] with some additional inference restrictions. We will only collect prime implicates containing either only positive literals or only negative literals. Let $\Sigma \subseteq \Omega_{\mathsf{C}}$ be a set of unary predicates. Then $\mathcal{PI}^{g+}_{\Sigma}(\Phi)$ denotes the set of all positive ground prime implicates of Φ that only use predicate symbols from $\Sigma \cup \Omega_{\mathsf{R}}$, while $\mathcal{PI}^{g-}_{\Sigma}(\Phi)$ denotes the set of all negative ground prime implicates of Φ that only use predicates symbols from $\Sigma \cup \Omega_{\mathsf{R}}$. Since Φ is Horn, $\mathcal{PI}^{g+}_{\Sigma}(\Phi)$ contains only unit clauses. A final recombination step looks at the clauses in $\mathcal{PI}^{g-}_{\Sigma}(\Phi)$ one after the other. These correspond to candidates for the connecting concepts V_2 of Def. 4.1.9. Recombination attempts to match each literal in one such clause with unit clauses from $\mathcal{PI}^{g+}_{\Sigma}(\Phi)$ by looking into their ground terms. If such a match is possible, it produces a suitable V_1 (also from Def. 4.1.9) to match V_2, and allows the creation of a hypothesis to the abduction problem.

4.2.1 Translation to First-Order Logic

Our abduction technique reconstructs $\mathcal{EL}$ hypotheses from first-order prime implicates where we use a translation involving clausification with Skolemization. Unsurprisingly, this breaks equivalence. However, using a concept renaming technique and the fact that (Herbrand) models of the Skolemized clauses can be used as models of the original TBox, one can still recover some useful entailed concept inclusions, in particular the subsumers of U_1 and the subsumee of U_2. I now present this in detail.

I assume the $\mathcal{EL}$ TBox in the input is in normal form, otherwise, a normalization technique as in Table 1.3 can be used because it is supported by Prop. 1.2.2. Thus, every CI is of one of the following forms:

$$P \sqsubseteq Q \qquad P_1 \sqcap P_2 \sqsubseteq Q \qquad \exists R.P \sqsubseteq Q \qquad P \sqsubseteq \exists R.Q$$

where $P, P_1, P_2, Q \in \Omega_\mathsf{C} \cup \{\top\}$. Every $\mathcal{EL}$ TBox can be transformed in polynomial time into this normal form through the introduction of fresh atomic concepts. Moreover, if we normalize an ontology $\mathcal{T}$ into $\mathcal{T}'$, then for any other TBox (and possible hypothesis) $\mathcal{H}$ that is not using names introduced by the normalization, $\mathcal{T} \cup \mathcal{H}$ and $\mathcal{T}' \cup \mathcal{H}$ entail the same CIs in the signature of $\mathcal{T} \cup \mathcal{H}$.

After the normalization, we eliminate occurrences of $\top$, replacing this concept everywhere by the fresh atomic concept $P_\top$. We furthermore add $\exists R.P_\top \sqsubseteq P_\top$ and $Q \sqsubseteq P_\top$ in $\mathcal{T}$ for every role R and atomic concept Q occurring in $\mathcal{T}$. This simulates the semantics of $\top$ for $P_\top$, namely the implicit property that $U \sqsubseteq \top$ holds for any U no matter what the TBox is. In particular, this ensures that whenever there is a positive prime implicate $Q(t)$ or $R(t,t')$, $P_\top(t)$ also becomes a prime implicate. Note that normalization and $\top$ elimination extend the signature, and thus potentially the solution space of the abduction problem. This is remedied by intersecting the set of abducible predicates Σ with the signature of the original input ontology. $\mathcal{T}$ is thus assumed to be in normal form and without $\top$ in the rest of the paper. In the end, $\mathtt{fo}(\mathcal{T})$ contains only formulas of the following shapes:

$$\begin{aligned}
&\forall x.\neg P(x) \vee Q(x),\\
&\forall x.\neg P_1(x) \vee \neg P_2(x) \vee Q(x),\\
&\forall x.\neg(\exists y.R(x,y) \wedge P(y)) \vee Q(x),\\
&\forall x.\neg P(x) \vee \exists y.(R(x,y) \wedge Q(y)).
\end{aligned}$$

$\mathcal{T}^-$ denotes the result of renaming all atomic concepts P in $\mathcal{T}$ using fresh *duplicate* symbols P^-. This renaming is done only on concepts but not on roles, and on U_2 but not on U_1 in the observation. This ensures that the literals in a clause of $\mathcal{PI}^{g-}_\Sigma(\Phi)$ all relate to the conjuncts of a $\preceq_\sqcap$-minimal subsumee of U_2. Without it, some of these conjuncts would not appear in the negative implicates due to the presence of their positive counterparts as atoms in $\mathcal{PI}^{g+}_\Sigma(\Phi)$. The translation of the abduction problem $\langle \mathcal{T}, \Sigma, U_1 \sqsubseteq U_2 \rangle$ is defined as the Skolemization and clausification of

$$\mathtt{fo}(\mathcal{T} \uplus \mathcal{T}^-) \wedge \neg\, \mathtt{fo}(U_1 \sqsubseteq U_2^-)$$

where $\mathtt{sk}_0$ is used as the unique fresh Skolem constant such that the Skolemization of $\neg\, \mathtt{fo}(U_1 \sqsubseteq U_2^-)$ results in $\{U_1(\mathtt{sk}_0), \neg U_2^-(\mathtt{sk}_0)\}$. This translation is denoted as $\Phi = \mathtt{FO}(\mathcal{T}, U_1 \sqsubseteq U_2)$.

Example 4.2.1. Consider the abduction problem $\langle \mathcal{T}_a, \Sigma, \alpha_a \rangle$ with Σ from Ex. 4.1.1. For the translation Φ of this problem, we have

$$\mathcal{PI}^{g+}_{\Sigma}(\Phi) = \{\mathsf{Professor}(\mathtt{sk}_0),\ \mathsf{Doctor}(\mathtt{sk}_0),\ \mathsf{Chair}(\mathtt{sk}_1(\mathtt{sk}_0)),\ \mathsf{PhD}(\mathtt{sk}_2(\mathtt{sk}_0))\}$$

$$\mathcal{PI}^{g-}_{\Sigma}(\Phi) = \{\neg\mathsf{Researcher}^{-}(\mathtt{sk}_0),\ \neg\mathsf{ResearchPosition}^{-}(\mathtt{sk}_1(\mathtt{sk}_0)) \vee \neg\mathsf{Diploma}^{-}(\mathtt{sk}_2(\mathtt{sk}_0))\}$$

where $\mathtt{sk}_1$ is the Skolem function introduced for $\mathsf{Professor} \sqsubseteq \exists\mathsf{employment}.\mathsf{Chair}$ and $\mathtt{sk}_2$ is introduced for $\mathsf{Doctor} \sqsubseteq \exists\mathsf{qualification}.\mathsf{PhD}$.

Detailed Example

Now, I illustrate the translation via a more detailed example. Consider the abduction problem $\langle \mathcal{T}, \Sigma, U_1 \sqsubseteq U_2 \rangle$ where

$$\begin{aligned}\mathcal{T} = \{ & U_1 \sqsubseteq P_1,\ U_1 \sqsubseteq P_2,\ U_1 \sqsubseteq \exists R_1.Q_1, \\ & Q_1 \sqsubseteq \exists R_2.S_1,\ Q_1 \sqsubseteq \exists R_2.T_1, \\ & \exists R_1.W_1 \sqcap P_3 \sqsubseteq U_2,\ Q_2 \sqcap W_2 \sqsubseteq W_1, \\ & \exists R_2.S_1 \sqcap \exists R_2.W_3 \sqsubseteq W_2,\ T_2 \sqcap P_2 \sqsubseteq W_3 \}\end{aligned}$$

and $\Sigma = \{P_1, P_2, P_3, Q_1, Q_2, S_1, T_1, T_2, T_3\}$. Consider the concepts

$$\begin{aligned}V_1 &= P_1 \sqcap P_2 \sqcap \exists R_1.(Q_1 \sqcap \exists R_2.S_1 \sqcap \exists R_2.T_1), \\ V_2 &= P_3 \sqcap \exists R_1.(Q_2 \sqcap \exists R_2.S_1 \sqcap \exists R_2.(T_2 \sqcap T_3)),\end{aligned}$$

Indeed, the concepts V_1 and V_2 are such that $\mathcal{T} \models U_1 \sqsubseteq V_1$ and $\mathcal{T} \models V_2 \sqsubseteq U_2$. Moreover, any concept V' s.t. $V' \preceq_\sqcap V_2$ and $V' \not\equiv V_2$ is not a subsumee of U_2. So, V_2 is a $\preceq_\sqcap$-minimal concept such that $\mathcal{T} \models V_2 \sqsubseteq U_2$. There is also a weak homomorphism from $\mathfrak{T}_{V_2}$ to $\mathfrak{T}_{V_1}$, as illustrated in Fig. 4.3. Thus,

$$\mathcal{H} = \{P_1 \sqcap P_2 \sqsubseteq P_3, Q_1 \sqsubseteq Q_2, T_1 \sqsubseteq T_2 \sqcap T_3\}$$

is a connection-minimal hypothesis. Note that the tautology $S_1 \sqsubseteq S_1$, that is one of the entailments that must hold in $\mathcal{T} \cup \mathcal{H}$, as is visible in Fig. 4.3, is not included in $\mathcal{H}$ since it is a tautology and thus $\mathcal{T} \models S_1 \sqsubseteq S_1$. The hypothesis $\mathcal{H}$ is even packed. In contrast,

$$\begin{aligned}\mathcal{H}_1 &= \{P_1 \sqsubseteq P_3,\ Q_1 \sqsubseteq Q_2,\ T_1 \sqsubseteq T_2 \sqcap T_3\} \text{ and} \\ \mathcal{H}_2 &= \{P_2 \sqsubseteq P_3,\ Q_1 \sqsubseteq Q_2,\ T_1 \sqsubseteq T_2 \sqcap T_3\}\end{aligned}$$

that are both connection-minimal but lack either P_1 or P_2 on the left-hand side of their first CI when compared with $\mathcal{H}$, are not packed.

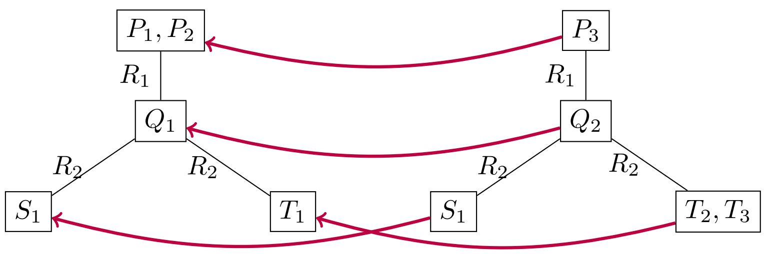

Figure 4.3: Two description trees with a weak homomorphism between them.

Before the translation, the TBox under consideration must be normalized. So, We additionally perform a normalization to $\mathcal{T}$. The CIs to normalize are $\exists R_1.W_1 \sqcap P_3 \sqsubseteq U_2$ and $\exists R_2.S_1 \sqcap \exists R_2.W_3 \sqsubseteq W_2$ for which we introduce the fresh concepts V_1', V_2' and V_3' and corresponding CIs $\exists R_2.S_1 \sqsubseteq V_1'$, $\exists R_2.W_3 \sqsubseteq V_2'$ and $\exists R_1.W_1 \sqsubseteq V_3'$, along with the normalized form of the two initial CIs, i.e., $V_3' \sqcap P_3 \sqsubseteq U_2$ and $V_1' \sqcap V_2' \sqsubseteq W_2$.

I do not show the axioms using $P_\top$ to simulate $\top$ since it will not be needed for the illustration of how hypotheses can be acquired here. The following shows the result of the normalization for $\mathcal{T}$ and their first-order translation after Skolemization (shown side-by-side).

$$
\begin{array}{rl}
U_1 \sqsubseteq P_1 & \neg U_1(x) \vee P_1(x) \\
U_1 \sqsubseteq P_2 & \neg U_1(x) \vee P_2(x) \\
U_1 \sqsubseteq \exists R_1.Q_1 & \begin{cases} \neg U_1(x) \vee R_1(x, \mathtt{sk}_1(x)) \\ \neg U_1(x) \vee Q_1(\mathtt{sk}_1(x)) \end{cases} \\
Q_1 \sqsubseteq \exists R_2.S_1 & \begin{cases} \neg Q_1(x) \vee R_2(x, \mathtt{sk}_2(x)) \\ \neg Q_1(x) \vee S_1(\mathtt{sk}_2(x)) \end{cases} \\
Q_1 \sqsubseteq \exists R_2.T_1 & \begin{cases} \neg Q_1(x) \vee R_2(x, \mathtt{sk}_2'(x)) \\ \neg Q_1(x) \vee T_1(\mathtt{sk}_2'(x)) \end{cases} \\
\exists R_1.W_1 \sqsubseteq V_3' & \neg R_1(x,y) \vee \neg W_1(y) \vee V_3'(x) \\
V_3' \sqcap P_3 \sqsubseteq U_2 & \neg V_3'(x) \vee \neg P_3(x) \vee U_2(x) \\
Q_2 \sqcap W_2 \sqsubseteq W_1 & \neg Q_2(x) \vee \neg W_2(x) \vee W_1(x) \\
\exists R_2.S_1 \sqsubseteq V_1' & \neg R_2(x,y) \vee \neg S_1(y) \vee V_1'(x) \\
\exists R_2.W_3 \sqsubseteq V_2' & \neg R_2(x,y) \vee \neg W_3(y) \vee V_2'(x) \\
V_1' \sqcap V_2' \sqsubseteq W_2 & \neg V_1'(x) \vee \neg V_2'(x) \vee W_2(x) \\
T_2 \sqcap P_2 \sqsubseteq W_3 & \neg T_2(x) \vee \neg P_2(x) \vee W_3(x)
\end{array}
$$

The translation of $\mathcal{T}^-$ is identical to that of $\mathcal{T}$ up to the replacement of every unary predicate with its duplicate and the introduction of fresh Skolem functions distinct from the ones used for $\mathcal{T}$. Let Φ denote the full translation of the problem. The ground prime implicates for $\Sigma = \{P_1, P_2, P_3, Q_1, Q_2, S_1, T_1, T_2, T_3\}$ are as follows:

$$
\begin{array}{lll}
\mathcal{PI}^{g+}_\Sigma(\Phi) & = \{ & P_1(\mathtt{sk}_0),\ P_2(\mathtt{sk}_0),\ Q_1(\mathtt{sk}_1(\mathtt{sk}_0)), \\
& & S_1(\mathtt{sk}_2(\mathtt{sk}_1(\mathtt{sk}_0))),\ T_1(\mathtt{sk}_2'(\mathtt{sk}_1(\mathtt{sk}_0)))\} \\
\mathcal{PI}^{g-}_\Sigma(\Phi) & = \{ & \neg P_3^-(\mathtt{sk}_0) \vee \neg Q_2^-(\mathtt{sk}_1(\mathtt{sk}_0)) \vee \neg S_1^-(\mathtt{sk}_2(\mathtt{sk}_1(\mathtt{sk}_0))) \vee \\
& & \quad \neg T_2^-(\mathtt{sk}_2'(\mathtt{sk}_1(\mathtt{sk}_0))) \vee \neg P_2^-(\mathtt{sk}_2'(\mathtt{sk}_1(\mathtt{sk}_0)))\}
\end{array}
$$

where $\mathtt{sk}_1$ is the Skolem function corresponding to the existential quantifier introduced by the translation of $\exists R_1.W_1$ to first-order logic, and where $\mathtt{sk}_2$ and $\mathtt{sk}_2'$ correspond respectively to $\exists R_2.S_1$ and $\exists R_2.W_3$. The only constructible hypothesis out of this configuration is $\mathcal{H}$, the packed connection-minimal hypothesis already introduced. Finally, $\mathcal{H}_3$ is the smallest TBox that fixes all entailments missing between V_1 and V_2, ensuring the connection minimality of $\mathcal{H}_3$ and it is packed, contrarily to $\mathcal{H}_1$ and $\mathcal{H}_2$ that lack either P_1 or P_2 on the left-hand side of their first concept inclusion. Note that the signature restriction has been made to capture only these hypotheses, but there would be many more if we considered the whole signature after normalization for Σ. In particular, including $P_\top$ to Σ would produce all solutions where $\top$ replaces the left-hand side of some concept inclusions in another hypothesis, so it should generally be avoided.

4.2.2 Canonical Model for $\mathcal{EL}$ Description Tree

In this section, I relate a set of atoms taken from the prime implicates, Herbrand models, and concepts altogether. The domain for the models is the set of possible ground terms (Herbrand universe) over $\Phi = \mathtt{FO}(\mathcal{T}, U_1 \sqsubseteq U_2)$ denoted as $\mathsf{T}_{\mathtt{sk}_0}(\Pi_{\mathsf{S}})$ where the set of Skolem functions $\Pi_{\mathsf{S}} \uplus \{sk_0\}$ is over Φ. First, I adapt the definition of a canonical model from [BKM99] by using $\mathsf{T}_{\mathtt{sk}_0}(\Pi_{\mathsf{S}})$ for the domain[1] of the $\mathcal{EL}$-description tree $\mathfrak{T}$ corresponding to a subsumer of C_1.

Definition 4.2.2 (Canonical Model)**.** Given a description tree $\mathfrak{T} = (\mathcal{V}, \mathcal{E}, v_0, l)$, a *Skolem labeling* $sl_{\mathfrak{T}} : \mathcal{V} \to \mathsf{T}_{\mathtt{sk}_0}(\Pi_{\mathsf{S}})$ of $\mathfrak{T}$ maps the vertices of $\mathfrak{T}$ to ground Skolem terms. A *canonical model* $\mathcal{M}(sl_{\mathfrak{T}})$ of $\mathfrak{T}$ is a Herbrand interpretation consisting of the following atoms:

- $R(sl_{\mathfrak{T}}(v), sl_{\mathfrak{T}}(w))$ for all $vRw \in E$
- $P(sl_{\mathfrak{T}}(v))$ for all $P \in l(v)$ and $v \in \mathcal{V}$

We denote by $\mathcal{M}_{\Omega_{\mathsf{C}}}(sl_{\mathfrak{T}})$ the subset of $\mathcal{M}(sl_{\mathfrak{T}})$ made of all atoms built over unary predicates, and by $\mathcal{M}_{\Omega_{\mathsf{R}}}(sl_{\mathfrak{T}})$, the rest of $\mathcal{M}(sl_{\mathfrak{T}})$, that contains all atoms built over binary predicates.

It is always possible to find a canonical model of an $\mathcal{EL}$-description tree $\mathfrak{T}$ as a subset in any Herbrand interpretation $\mathcal{I}$ for which $(U_{\mathfrak{T}})^{\mathcal{I}}$ is not empty. This is formally stated, and proven, in the following lemma.

Lemma 4.2.3. Given an $\mathcal{EL}$-description tree $\mathfrak{T} = (\mathcal{V}, \mathcal{E}, v_0, l)$ and a Herbrand interpretation $\mathcal{I}$, if $t \in (U_{\mathfrak{T}})^{\mathcal{I}}$ then there exists a Skolem labeling $sl_{\mathfrak{T}}$ s.t. $sl_{\mathfrak{T}}(v_0) = t$ and $\mathcal{M}(sl_{\mathfrak{T}}) \subseteq \mathcal{I}$

Proof. Given an $\mathcal{EL}$-description tree $\mathfrak{T} = (\mathcal{V}, \mathcal{E}, v_0, l)$, a Herbrand interpretation $\mathcal{I}$ and a Skolem term t, such that $t \in (U_{\mathfrak{T}})^{\mathcal{I}}$, let us construct the suitable Skolem labeling $sl_{\mathfrak{T}}$. We proceed inductively on the depth of $\mathfrak{T}$.

$$sl_{\mathfrak{T}}(v) = \begin{cases} t & \text{if } v = v_0, \\ sl_{\mathfrak{T}(w)}(v) & \text{if } v_0Rw \in \mathcal{E} \text{ for some } R, \text{ and } v \in \mathcal{V}_{\mathfrak{T}(w)}, \end{cases}$$

where $\mathfrak{T}(w)$ is the subtree of $\mathfrak{T}$ rooted in w, $\mathcal{V}_{\mathfrak{T}(w)}$ is the subset of $\mathcal{V}$ that occurs in $\mathfrak{T}(w)$ and $sl_{\mathfrak{T}(w)}$ is defined as $sl_{\mathfrak{T}}$ but on t' instead of t, for a t' such that $r(t, t') \in \mathcal{I}$ and $t' \in (U_{\mathfrak{T}(w)})^{\mathcal{I}}$. Such a t' must exist because $\exists R.U_{\mathfrak{T}(w)}$ is a conjunct in $U_{\mathfrak{T}}$ and $t \in (U_{\mathfrak{T}})^{\mathcal{I}}$. Hence $sl_{\mathfrak{T}(w)}$ is well-defined. This construction terminates because the depth of all $\mathfrak{T}(w)$ is strictly smaller than that of $\mathfrak{T}$.

If $\mathfrak{T}$ is of depth 0, then $sl_{\mathfrak{T}}$ is simply defined on v_0 such that $sl_{\mathfrak{T}}(v_0) = t$, and $U_{\mathfrak{T}}$ is a conjunction of atomic concepts $P \in l(v_0)$. Thus, $t \in (U_{\mathfrak{T}})^{\mathcal{I}}$ is equivalent to $P(t) \in \mathcal{I}$ for all $P \in l(v_0)$. Hence, any atom $P(sl_{\mathfrak{T}}(v_0)) = P(t) \in \mathcal{M}(sl_{\mathfrak{T}})$ is also in $\mathcal{I}$ for all $P \in l(v_0)$.

[1] In [BKM99], the canonical interpretation uses the set of vertices as its domain.

If $\mathfrak{T}$ is of depth $i > 0$, for any $v \in \mathcal{V} \setminus \{v_0\}$, there exists a $w \in \mathcal{V}$ such that $v_0 R w \in E$ and $v \in V_{\mathfrak{T}(w)}$, i.e., v must belong to a subtree rooted in one of the children w of the root of $\mathfrak{T}$. Then $sl_{\mathfrak{T}}(v) = sl_{\mathfrak{T}(w)}(v)$. By induction, $\mathcal{M}(sl_{\mathfrak{T}(w)}) \subseteq \mathcal{I}$. Moreover $P(t) \in \mathcal{I}$ for all $P \in l(v_0)$ as in the base case; and $R(t, sl_{\mathfrak{T}(w)}(w)) \in \mathcal{I}$ and $sl_{\mathfrak{T}(w)}(w) \in (U_{\mathfrak{T}(w)})^{\mathcal{I}}$ for all w children of v_0 by construction of $sl_{\mathfrak{T}(w)}$. Thus $\mathcal{I}(sl_{\mathfrak{T}}) \subseteq \mathcal{I}$. □

4.2.3 Subsumers of U_1 and Positive Prime Implicates

I first establish the link between the positive prime implicates in $\mathcal{PI}^{g+}_{\Sigma}(\Phi)$ and the subsumers of U_1, then I do the same for the negative side. Furthermore, I show how to extend a canonical model so that it also satisfies $\mathcal{T}$ and link the existence of such a model built for some $U_{\mathfrak{T}}$ and $\mathtt{sk}_0$ to the existence of the entailment $\mathcal{T} \models U_1 \sqsubseteq U_{\mathfrak{T}}$, while showing that this model is in fact a subset of $\mathcal{PI}^{g+}_{\Sigma}(\Phi)$. This relation is established at the semantics level, by constructing a Herbrand model of $\mathcal{T}$ and showing it necessarily contains the prime implicates of Φ.

We assume $\Sigma = \Omega_{\mathtt{C}}$. For Th. 4.2.11, the case where $\Sigma \subsetneq \Omega_{\mathtt{C}}$ trivially follows, but that is not the case for the intermediate results.

First, I show that $\mathcal{PI}^{g+}_{\Sigma}(\Phi)$ holds the role of universal Herbrand model for $\Phi = \mathtt{FO}(\mathcal{T}, U_1 \sqsubseteq U_2)$. The proof adapts a result by Bienvenu et al. [BO15] to the case with only one constant but a possibly infinite domain.

Lemma 4.2.4 ($\mathcal{PI}^{g+}_{\Sigma}(\Phi)$ as a universal model)**.** Given the translation Φ of an abduction problem, the set $\mathcal{PI}^{g+}_{\Sigma}(\Phi)$ considered as a Herbrand interpretation is a model of Φ and for any other Herbrand model $\mathcal{I}$ of Φ, $\mathcal{PI}^{g+}_{\Sigma}(\Phi) \subseteq \mathcal{I}$.

Proof. By the definition of a prime implicate, any model of Φ must be a model of any $\varphi \in \mathcal{PI}^{g+}_{\Sigma}(\Phi)$. Moreover, a positive prime implicate can only be an atom since Φ contains only Horn clauses. Thus all Herbrand models must contain $\mathcal{PI}^{g+}_{\Sigma}(\Phi)$.

To show that $\mathcal{PI}^{g+}_{\Sigma}(\Phi)$ is itself a Herbrand model, we construct the Herbrand Interpretation $\mathcal{I} = \bigcup_i \mathcal{I}_i$ for $i \in \mathbb{N}$ where:

- $\mathcal{I}_0 = \{U_1(\mathtt{sk}_0)\}$ and,
- given $\mathcal{I}_j$,

$$\begin{aligned}\mathcal{I}_{j+1} \quad = \quad & \mathcal{I}_j \cup \{Q(t) \mid t \in (V)^{\mathcal{I}_j},\ \neg\,\mathtt{fo}(V,x) \vee Q(x) \in \Phi\} \\ & \cup \{Q(\mathtt{sk}(t)), R(t,\mathtt{sk}(t)) \mid t \in (P)^{\mathcal{I}_j}, \\ & \qquad \neg\,\mathtt{fo}(P,x) \vee Q(\mathtt{sk}(x)) \in \Phi,\ \neg\,\mathtt{fo}(P,x) \vee R(x,\mathtt{sk}(x)) \in \Phi\} \\ & \cup \{Q(t) \mid \mathtt{sk}(t) \in (P)^{\mathcal{I}_j},\ (t,\mathtt{sk}(t)) \in R^{\mathcal{I}_j}, \\ & \qquad \neg R(x,y) \vee \neg P(y) \vee Q(x) \in \Phi\}.\end{aligned}$$

I show that $\mathcal{I}$ is a model of Φ. We know that $\mathcal{I} \models U_1(\mathtt{sk}_0)$ by construction of $\mathcal{I}_0$, and that all other clauses containing only non-duplicated literals are also satisfied by $\mathcal{I}$, again by construction. Note that there are cases where no $\mathcal{I}_j$ alone is enough to satisfies a clause, but they are all satisfied at the limit by $\mathcal{I}$ (e.g., if $\mathcal{T}$ includes a concept inclusion $P \sqsubseteq \exists R.P$, possibly leading to the presence of infinitely many atoms of the form $P(sk^n(t)) \in \mathcal{I}$). Regarding the remaining clauses in Φ, they all contain at least one literal of the form $\neg P^-(x)$ and since $\mathcal{I}$ includes no atom $P^-(t)$ at all, $\mathcal{I}$ also satisfies that part, thus $\mathcal{I}$ is a model of Φ.

It remains only to show that $\mathcal{I} \subseteq \mathcal{PI}^{g+}_{\Sigma}(\Phi)$ to have $\mathcal{PI}^{g+}_{\Sigma}(\Phi) = \mathcal{I}$, thus showing that $\mathcal{PI}^{g+}_{\Sigma}(\Phi)$ is a model of Φ. This is done by induction. Clearly $\mathcal{I}_0 \subseteq \mathcal{PI}^{g+}_{\Sigma}(\Phi)$, and, assuming $\mathcal{I}_j \subseteq \mathcal{PI}^{g+}_{\Sigma}(\Phi)$ for some $j \geq 0$, then any atom in $\mathcal{I}_j$ can be derived by resolution from Φ, thus, by construction, any atom in $\mathcal{I}_{j+1} \setminus \mathcal{I}_j$ can be derived from Φ by one additional resolution step, making them implicates of Φ. Because they are atoms, they must be prime implicates, thus $\mathcal{I}_{j+1} \subseteq \mathcal{PI}^{g+}_{\Sigma}(\Phi)$, completing the induction. □

Note that if Φ was a set of *definite* Horn clauses, the above result would be immediate because it is well-known in logic programming [Llo87]. The presence of the negative clause $\neg U_2^-(\mathtt{sk}_0) \in \Phi$ is what justifies the existence of the current proof.

Now that $\mathcal{PI}^{g+}_{\Sigma}(\Phi)$ has been established as the universal Herbrand model of Φ, the atoms it contains can be used to reconstruct concepts subsuming U_1 by means of a canonical model.

Lemma 4.2.5 (Canonical Model and $\mathcal{PI}^{g+}_{\Sigma}(\Phi)$)**.** Given an abduction problem $\langle \mathcal{T}, \mathcal{H}, U_1 \sqsubseteq U_2 \rangle$, its first-order translation Φ and an $\mathcal{EL}$-description tree $\mathfrak{T} = (\mathcal{V}, \mathcal{E}, v_0, l)$, the entailment $\mathcal{T} \models U_1 \sqsubseteq U_{\mathfrak{T}}$ holds if and only if there exists a Skolem labeling $sl_{\mathfrak{T}}$ such that $sl_{\mathfrak{T}}(v_0) = \mathtt{sk}_0$ and $\mathcal{M}(sl_{\mathfrak{T}}) \subseteq \mathcal{PI}^{g+}_{\Sigma}(\Phi)$.

Proof. Given the preconditions of the lemma, let us first assume $\mathcal{T} \models U_1 \sqsubseteq U_2$ to show the existence of a Skolem labeling $sl_{\mathfrak{T}}$ such that $sl_{\mathfrak{T}}(v_0) = \mathtt{sk}_0$ and $\mathcal{M}(sl_{\mathfrak{T}}) \subseteq \mathcal{PI}^{g+}_{\Sigma}(\Phi)$. Since $U_1(\mathtt{sk}_0) \in \Phi$ by definition, we have $\mathtt{sk}_0 \in (U_1)^{\mathcal{I}}$ for any Herbrand model $\mathcal{J}$ of Φ. Moreover, because $\mathcal{T} \models U_1 \sqsubseteq U_{\mathfrak{T}}$, it follows that $\mathtt{sk}_0 \in (U_{\mathfrak{T}})^{\mathcal{I}}$ for any Herbrand model $\mathcal{J}$ of Φ. By Lemma 4.2.4, we know that $\mathcal{PI}^{g+}_{\Sigma}(\Phi)$ can be seen as a Herbrand model of Φ, thus $\mathtt{sk}_0 \in (U_{\mathfrak{T}})^{\mathcal{PI}^{g+}_{\Sigma}(\Phi)}$. The existence of a Skolem labeling with the desired properties follows by Lemma 4.2.3.

To prove the opposite implication, we assume a Skolem labeling verifying $sl_{\mathfrak{T}}(v_0) = \mathtt{sk}_0$ and $\mathcal{M}(sl_{\mathfrak{T}}) \subseteq \mathcal{PI}^{g+}_{\Sigma}(\Phi)$ and show that $\mathcal{T} \models U_1 \sqsubseteq U_{\mathfrak{T}}$ by contradiction. Then $\mathtt{sk}_0 \in (U_{\mathfrak{T}})^{\mathcal{M}(sl_{\mathfrak{T}})}$ because $sl_{\mathfrak{T}}(v_0) = \mathtt{sk}_0$. Towards contradiction, we assume $\mathcal{T} \not\models U_1 \sqsubseteq U_{\mathfrak{T}}$. Then $\mathtt{fo}(\mathcal{T}) \not\models \mathtt{fo}(U_1 \sqsubseteq U_{\mathfrak{T}})$, since the standard translation from $\mathcal{EL}$ to first-order logic preserves entailment [BHLS17]. Thus, $\mathtt{fo}(\mathcal{T}) \land \neg\, \mathtt{fo}(U_1 \sqsubseteq U_{\mathfrak{T}})$ is satisfiable and hence, the Skolemizations of

$$\mathtt{fo}(\mathcal{T}) \land \neg\, \mathtt{fo}(U_1 \sqsubseteq U_{\mathfrak{T}}) = \mathtt{fo}(\mathcal{T}) \land \exists x.(U_1(x) \land \neg\, \mathtt{fo}(U_{\mathfrak{T}}, x))$$

are also satisfiable. Let us consider the particular Skolemization φ of $\mathtt{fo}(\mathcal{T}) \land \neg\, \mathtt{fo}(U_1 \sqsubseteq U_{\mathfrak{T}})$ that coincides with Φ on the Skolemization of $\mathcal{T}$ and uses $\mathtt{sk}_0$ to Skolemize the existential variable in $\neg\, \mathtt{fo}(U_1 \sqsubseteq U_{\mathfrak{T}})$. Let $\mathcal{I}'$ be a minimal Herbrand model of φ. It verifies $\mathtt{sk}_0 \in (U_1)^{\mathcal{I}'}$ and $\mathtt{sk}_0 \notin (U_{\mathfrak{T}})^{\mathcal{I}'}$. We show that $\mathcal{I}'$ is a model of Φ, which will allow us to raise a contradiction on that last statement. Since, by design, φ contains all the non-renamed clauses in Φ, it follows that $\mathcal{I}'$ satisfies these non-renamed clauses also for Φ. Since φ does not include renamed atoms, the minimality of $\mathcal{I}'$ ensures that it does not include any renamed atoms. This ensures that $\mathcal{I}'$ also models the renamed part of Φ: for any renamed $U^- \sqsubseteq V^-$, it holds that $(U^-)^{\mathcal{I}'} = (V^-)^{\mathcal{I}'} = \emptyset$, and $\neg U_2^-(\mathtt{sk}_0)$ is also true in $\mathcal{I}$'. Thus, $\mathcal{I}'$ is a model of Φ. However, since $\mathcal{M}(sl_{\mathfrak{T}}) \subseteq \mathcal{PI}^{g+}_{\Sigma}(\Phi)$, and $\mathcal{PI}^{g+}_{\Sigma}(\Phi) \subseteq \mathcal{I}'$ by Lemma 4.2.4, it follows that $\mathcal{M}(sl_{\mathfrak{T}}) \subseteq \mathcal{I}'$ must hold.

In addition, since $U_{\mathfrak{T}}(\mathtt{sk}_0) \in \mathcal{M}(sl_{\mathfrak{T}})$ because $sl_{\mathfrak{T}}(v_0) = \mathtt{sk}_0$, it follows that $\mathtt{sk}_0 \in (U_{\mathfrak{T}})^{\mathcal{I}'}$, a contradiction. □

Lemma 4.2.5 establishes a relation between the FOL encoding Φ and the original $\mathcal{EL}$ problem. Note that the tree and Skolem labeling is not necessarily unique. For $\mathcal{A} = \{R(\mathtt{sk}_0, sk_1(\mathtt{sk}_0)), P(\mathtt{sk}_1(\mathtt{sk}_0)), Q(\mathtt{sk}_1(\mathtt{sk}_0))\}$, two possible $U_{\mathfrak{T}}$'s are $\exists R.(P \sqcap Q)$ and $\exists R.P \sqcap \exists R.Q$.

Apparently, the prime implicates using role predicates are not even necessary to get the subsumer of U_1. We can prove a stronger result to know how to construct the $U_{\mathfrak{T}}$ such that $\mathcal{T} \models U_1 \sqsubseteq U_{\mathfrak{T}}$ from $\mathcal{PI}^{g+}_{\Sigma}(\Phi)$. Lemma 4.2.6 does the job, by showing that it is only necessary to collect the atomic prime implicates about unary predicates (the ones from Ω_{C}) to construct all relevant $U_{\mathfrak{T}}$.

Lemma 4.2.6 (Construction of Subsumers of U_1)**.** Given an abduction problem $\langle \mathcal{T}, \mathcal{H}, U_1 \sqsubseteq U_2 \rangle$, its first-order translation Φ and a set $\mathcal{A} = \{P_1(t_1), \dots, P_n(t_n)\} \subseteq \mathcal{PI}^{g+}_{\Sigma}(\Phi)$ where $n > 0$, there exists $\mathfrak{T} = (\mathcal{V}, \mathcal{E}, v_0, l)$ and $sl_{\mathfrak{T}}$ s.t. $\mathcal{A} = \mathcal{M}_{\Omega_{\mathsf{C}}}(sl_{\mathfrak{T}})$ and $\mathcal{T} \models U_1 \sqsubseteq U_{\mathfrak{T}}$.

Proof. Given an abduction problem $\langle \mathcal{T}, \mathcal{H}, U_1 \sqsubseteq U_2 \rangle$, its first-order translation Φ and a set $\mathcal{A} = \{P_1(t_1), \dots, P_n(t_n)\} \subseteq \mathcal{PI}^{g+}_{\Sigma}(\Phi)$ where $n > 0$, we notice that every singleton set $\{P_i(t_i)\} \subseteq \mathcal{A}$ also verifies that $\{P_i(t_i)\} \subseteq \mathcal{PI}^{g+}_{\Sigma}(\Phi)$. Thus to prove the property for any $\mathcal{A}$, we first show it for singletons and then we show how to construct a description tree for any $\mathcal{A}$ given the description trees for each singleton containing an element of $\mathcal{A}$.

Let $\mathcal{A} = \{P(t)\}$ be a singleton. In practice, we need a slightly stronger property: we show the existence of a $\mathfrak{T} = (\mathcal{V}, \mathcal{E}, v_0, l)$ and $sl_{\mathfrak{T}}$ such that $\{P(t)\} = \mathcal{M}_{\Omega_{\mathsf{C}}}(sl_{\mathfrak{T}})$, $sl_{\mathfrak{T}}(v_0) = \mathtt{sk}_0$ and $\mathcal{M}(sl_{\mathfrak{T}}) \subseteq \mathcal{PI}^{g+}_{\Sigma}(\Phi)$. Then $\mathcal{T} \models U_1 \sqsubseteq U_{\mathfrak{T}}$ follows by Lemma 4.2.5. By Lemma 4.2.4, $\mathcal{PI}^{g+}_{\Sigma}(\Phi)$ is a Herbrand model of Φ and thus $U_1(\mathtt{sk}_0) \in \mathcal{PI}^{g+}_{\Sigma}(\Phi)$. As shown in the proof of Lemma 4.2.4, we can write $\mathcal{PI}^{g+}_{\Sigma}(\Phi)$ as $\bigcup_{i \in \mathbb{N}} \mathcal{I}_i$, where:

- $\mathcal{I}_0 = \{U_1(\mathtt{sk}_0)\}$ and,
- given $\mathcal{I}_j$,

$$\begin{aligned}\mathcal{I}_{j+1} \quad = \quad & \mathcal{I}_j \cup \{Q(t) \mid t \in (V)^{\mathcal{I}_j},\ \neg\,\mathtt{fo}(V,x) \vee Q(x) \in \Phi\} \\ & \cup \{Q(\mathtt{sk}(t)), R(t, \mathtt{sk}(t)) \mid t \in (P)^{\mathcal{I}_j}, \\ & \qquad \neg\,\mathtt{fo}(P,x) \vee Q(\mathtt{sk}(x)) \in \Phi,\ \neg\,\mathtt{fo}(P,x) \vee R(x, \mathtt{sk}(x)) \in \Phi\} \\ & \cup \{Q(t) \mid \mathtt{sk}(t) \in (P)^{\mathcal{I}_j},\ (t, \mathtt{sk}(t)) \in R^{\mathcal{I}_j}, \\ & \qquad \neg R(x,y) \vee \neg P(y) \vee Q(x) \in \Phi\}.\end{aligned}$$

Since $P(t) \in \mathcal{PI}^{g+}_{\Sigma}(\Phi)$, there exists an $i \in \mathbb{N}$ that is the smallest such that $P(t) \in \mathcal{I}_i$. We construct $\mathfrak{T}$ inductively, depending on the value of i. If $i = 0$, then $P(t) = U_1(\mathtt{sk}_0)$ and thus defining $\mathfrak{T}$ and $sl_{\mathfrak{T}}$ as $\mathfrak{T} = (\{v_0\}, \emptyset, v_0, \{v_0 \mapsto \{U_1\}\})$ and $sl_{\mathfrak{T}} = \{v_0 \mapsto \mathtt{sk}_0\}$ ensures additionally that $\mathcal{A} = \{U_1(\mathtt{sk}_0)\} = \mathcal{M}_{\Omega_{\mathsf{C}}}(sl_{\mathfrak{T}}) = \mathcal{M}(sl_{\mathfrak{T}}) \subseteq \mathcal{PI}^{g+}_{\Sigma}(\Phi)$. Assuming we know how to construct suitable description trees and Skolem labellings up to a given $j \in \mathbb{N}$, when $i = j+1$, the construction of $\mathfrak{T}$ depends on the reason for which $P(t) \in \mathcal{I}_{j+1} \setminus \mathcal{I}_j$.

- If $P(t) \in \{Q(t) \mid t \in (V)^{\mathcal{I}_j},\ \neg\,\mathtt{fo}(V,x) \vee Q(x) \in \Phi\}$, where V is in fact an atomic concept Q then $Q(t) \in \mathcal{I}_j$ and thus $Q(t) \in \mathcal{PI}^{g+}_{\Sigma}(\Phi)$. By induction, let $\mathfrak{T}' = (\mathcal{V}', \mathcal{E}', v_0, l')$ and $sl_{\mathfrak{T}'}$ be a description tree and Skolem labeling such that $sl_{\mathfrak{T}'}(v_0) = \mathtt{sk}_0$, $\mathcal{M}(sl_{\mathfrak{T}'}) \subseteq \mathcal{PI}^{g+}_{\Sigma}(\Phi)$ and $\{Q(t)\} = \mathcal{M}_{\Omega_\mathsf{C}}(sl_{\mathfrak{T}'})$. Let $v \in \mathcal{V}'$ be the node such that $l_{\mathfrak{T}'}(v) = \{Q\}$ and $sl_{\mathfrak{T}'}(v) = t$. Then the Skolem labeling $sl_{\mathfrak{T}}$ is defined as identical to $sl_{\mathfrak{T}'}$ and we define $\mathfrak{T}$ as $(\mathcal{V}', \mathcal{E}', v_0, l'[v \mapsto \{P\}])$, where $l'[v \mapsto \{P\}]$ denotes the function l' except on v for which the value returned is $\{P\}$ so that $\{P(t)\} = \mathcal{M}_{\Omega_\mathsf{C}}(sl_{\mathfrak{T}})$ as wanted. Since $\mathcal{M}(sl_{\mathfrak{T}}) \subseteq \{P(t)\} \cup \mathcal{M}(sl_{\mathfrak{T}'})$ and $P(t) \in \mathcal{PI}^{g+}_{\Sigma}(\Phi)$, it follows that $\mathcal{M}(sl_{\mathfrak{T}}) \subseteq \mathcal{PI}^{g+}_{\Sigma}(\Phi)$.

- If $P(t) \in \{Q(t) \mid t \in (V)^{\mathcal{I}_j}, \neg\, \mathtt{fo}(V, x) \vee Q(x) \in \Phi\}$, where V is in fact the conjunction of two atomic concepts Q_1 and Q_2, then both $Q_1(t)$ and $Q_2(t)$ belong to $\mathcal{I}_j$ and thus to $\mathcal{PI}^{g+}_{\Sigma}(\Phi)$. We adapt exactly as in the last case any of the description trees $\mathfrak{T}_1$ or $\mathfrak{T}_2$ and associated Skolem labeling $sl_{\mathfrak{T}_1}$ or $sl_{\mathfrak{T}_2}$, that respectively correspond to Q_1 and Q_2 and verify the properties by induction.

- If $P(t) \in \{Q(\mathtt{sk}(t)) \mid t \in (P')^{\mathcal{I}_j}, R(t, \mathtt{sk}(t)) \in \mathcal{I}_{j+1}, \neg\, \mathtt{fo}(P', x) \vee Q(\mathtt{sk}(x)) \in \Phi, \neg\, \mathtt{fo}(P', x) \vee R(x, \mathtt{sk}(x)) \in \Phi\}$ then $t = \mathtt{sk}(t')$ for some $\mathtt{sk}$ and t' such that $P'(t') \in \mathcal{I}_j$, $\neg P'(x) \vee P(\mathtt{sk}(x)), \neg P'(x) \vee R(x, \mathtt{sk}(x)) \in \Phi$ for some R and P'. Since $P'(t') \in \mathcal{PI}^{g+}_{\Sigma}(\Phi)$, there exists a description tree $\mathfrak{T}' = (\mathcal{V}', \mathcal{E}', v_0, l')$ such that $sl_{\mathfrak{T}'}(v_0) = \mathtt{sk}_0$, $\mathcal{M}_{\Omega_\mathsf{C}}(sl_{\mathfrak{T}'}) = \{P'(t')\}$, and $\mathcal{M}(sl_{\mathfrak{T}'}) \sqsubseteq \mathcal{PI}^{g+}_{\Sigma}(\Phi)$. Let v' be the leaf node such that $l'(v) = \{P'\}$ and $sl_{\mathfrak{T}'}(v) = t'$. We introduce a fresh node v to define $\mathfrak{T}$ as $(\mathcal{V}' \cup \{v\}$, $\mathcal{E}' \cup \{v'Rv\}$, v_0, $l'[v' \mapsto \emptyset] \cup \{v \mapsto \{P\}\})$ and $sl_{\mathfrak{T}} = sl_{\mathfrak{T}'} \cup \{v \mapsto t\}$. Therefore we have $sl_{\mathfrak{T}}(v_0) = sl_{\mathfrak{T}'}(v_0) = \mathtt{sk}_0$, $\mathcal{M}_{\Omega_\mathsf{C}}(sl_{\mathfrak{T}}) = \{P(t)\}$ and $\mathcal{M}(sl_{\mathfrak{T}}) \subseteq \{P(\mathtt{sk}(t)), R(t, \mathtt{sk}(t))\} \cup \mathcal{M}(sl_{\mathfrak{T}'}) \subseteq \mathcal{PI}^{g+}_{\Sigma}(\Phi)$ because it holds that $\{P(\mathtt{sk}(t')), R(t', \mathtt{sk}(t'))\} \subseteq \mathcal{PI}^{g+}_{\Sigma}(\Phi)$.

- If $P(t) \in \{Q(t) \mid \mathtt{sk}(t) \in (P')^{\mathcal{I}_j}, (t, \mathtt{sk}(t)) \in R^{\mathcal{I}_j}, \neg R(x, y) \vee \neg P'(y) \vee Q(x) \in \Phi\}$ then there exist P', R and $\mathtt{sk}$ such that $P'(\mathtt{sk}(t)) \in \mathcal{I}_j$, $R(t, \mathtt{sk}(t)) \in \mathcal{I}_j$, and $\neg R(x, y) \vee \neg P'(y) \vee P(x) \in \Phi$. By induction, we consider a description tree $\mathfrak{T}' = (\mathcal{V}', \mathcal{E}', v_0, l')$ and associated Skolem labeling $sl_{\mathfrak{T}'}$ for which $sl_{\mathfrak{T}'}(v_0) = \mathtt{sk}_0$, $\mathcal{M}_{\Omega_\mathsf{C}}(sl_{\mathfrak{T}'}) = \{P'(t')\}$, and $\mathcal{M}(sl_{\mathfrak{T}'}) \subseteq \mathcal{PI}^{g+}_{\Sigma}(\Phi)$. Let v be the leaf in $\mathcal{V}'$ such that $l'(v) = \{P'\}$ and w be its parent in the tree, such that $wR'v \in \mathcal{E}'$ for some R'. We define $\mathfrak{T}$ as $(\mathcal{V}' \setminus \{v\}, \mathcal{E}' \setminus \{wR'v\}, v_0, l'[w \mapsto \{P\}] \setminus \{v \mapsto \{P'\}\})$ and $sl_{\mathfrak{T}} = sl_{\mathfrak{T}'} \setminus \{v \mapsto \mathtt{sk}(t)\}$. Thus, $sl_{\mathfrak{T}}(v_0) = sl_{\mathfrak{T}'}(v_0) = \mathtt{sk}_0$, $\mathcal{M}_{\Omega_\mathsf{C}}(sl_{\mathfrak{T}}) = \{P(t)\}$ and $\mathcal{M}(sl_{\mathfrak{T}}) \subseteq \{P(t)\} \cup \mathcal{M}(sl_{\mathfrak{T}'}) \subseteq \mathcal{PI}^{g+}_{\Sigma}(\Phi)$.

Let us now consider the case of non-singleton $\mathcal{A} = \{P_1(t_1), \ldots, P_n(t_n)\}$ $(n > 1)$. We have just seen how to obtain description trees $\mathfrak{T}_i = (\mathcal{V}_i, E_i, v_0, l_i)$ and Skolem labellings $sl_{\mathfrak{T}_i}$ for $i \in \{1, \ldots, n\}$ such that $\{P_i(t_i)\} = \mathcal{M}_{\Omega_\mathsf{C}}(sl_{\mathfrak{T}_i})$, $sl_{\mathfrak{T}_i}(v_0) = \mathtt{sk}_0$ and $\mathcal{M}(sl_{\mathfrak{T}_i}) \subseteq \mathcal{PI}^{g+}_{\Sigma}(\Phi)$. We define $\mathfrak{T} = (\mathcal{V}, \mathcal{E}, v_0, l)$ and $sl_{\mathfrak{T}}$ by introducing a node $v \in \mathcal{V}$ for each $t \in \bigcup_{i=1}^{n}\{sl_{\mathfrak{T}_i}(v') \mid v' \in \mathcal{V}_i\}$ and setting $sl_{\mathfrak{T}}(v) = t$ in each case. For $t = \mathtt{sk}_0$, the introduced node $v \in \mathcal{V}$ is named v_0 and declared as the root of $\mathfrak{T}$. It remains to define $\mathcal{E}$ and l. For $\mathcal{E}$ we collect all edges from the description trees $\mathfrak{T}_i$ to obtain

$$\mathcal{E} = \bigcup_{i=1}^{n}\{vRw \mid v'Rw' \in E_i, sl_{\mathfrak{T}_i}(v') = sl_{\mathfrak{T}}(v), sl_{\mathfrak{T}_i}(w') = sl_{\mathfrak{T}}(w)\}.$$

For l, we proceed similarly to collect labels, producing for each $v \in \mathcal{V}$,

$$l(v) = \bigcup_{i=1}^{n} \{l_i(v') \mid v' \in \mathcal{V}_i, sl_{\mathfrak{T}_i}(v') = sl_{\mathfrak{T}}(v)\}.$$

Thus

$$\begin{aligned}\mathcal{M}_{\Omega_{\mathsf{C}}}(sl_{\mathfrak{T}}) &= \{P(sl_{\mathfrak{T}}(v)) \mid P \in l(v), v \in \mathcal{V}\} \\ &= \bigcup_{i=1}^{n} \{P(sl_{\mathfrak{T}_i}(v)) \mid P \in l_i(v'), v' \in \mathcal{V}_i\} \\ &= \bigcup_{i=1}^{n} \mathcal{M}_{\Omega_{\mathsf{C}}}(sl_{\mathfrak{T}_i}) = \mathcal{A}\end{aligned}$$

and since also

$$\begin{aligned}\mathcal{M}_{\Omega_{\mathsf{R}}}(sl_{\mathfrak{T}}) &= \{R(sl_{\mathfrak{T}}(v), sl_{\mathfrak{T}}(w)) \mid vRw \in \mathcal{E}\} \\ &= \bigcup_{i=1}^{n} \{R(sl_{\mathfrak{T}_i}(v'), sl_{\mathfrak{T}_i}(w')) \mid v'Rw' \in E_i, sl_{\mathfrak{T}_i}(v') = sl_{\mathfrak{T}}(v), sl_{\mathfrak{T}_i}(w') = sl_{\mathfrak{T}}(w)\} \\ &= \bigcup_{i=1}^{n} \mathcal{M}_{\Omega_{\mathsf{R}}}(sl_{\mathfrak{T}_i}),\end{aligned}$$

we have $\mathcal{M}(sl_{\mathfrak{T}}) = \bigcup_{i=1}^{n} \mathcal{M}(sl_{\mathfrak{T}_i}) \subseteq \mathcal{PI}^{g+}_{\Sigma}(\Phi)$, and by Lemma 4.2.5, it holds that $\mathcal{T} \models U_1 \sqsubseteq U_{\mathfrak{T}}$. □

4.2.4 Subsumees of U_2 and Negative Prime Implicates

I show how the renamed negative ground prime implicates, relate to the $\preceq_{\sqcap}$-minimal subsumees of U_2. However, I first need to present this for the not necessarily prime implicate case and then strengthen it later.

To that aim, we need the canonical model again, but this time for the renamed version of some $U_{\mathfrak{T}}$ subsumee of U_2, with the restriction that there must exist a weak homomorphism from a subsumer of U_1 to this $U_{\mathfrak{T}}$, the idea being that $\mathcal{H}$ is built to provide the missing CIs that will turn the weak homomorphism into a $(\mathcal{T} \cup \mathcal{H})$-homomorphism.

Lemma 4.2.7 ($\mathcal{T}$-Homomorphism and Negative Implicates)**.** Given an abduction problem $\langle \mathcal{T}, \Sigma, U_1 \sqsubseteq U_2 \rangle$, Φ denotes its translation to first-order logic and $\mathcal{A} = \{P_1(t_1), \ldots, P_k(t_k)\} \subseteq \mathcal{PI}^{g+}_{\Sigma}(\Phi)$. As allowed by Lemma 4.2.6, let $\mathfrak{T}_1 = (\mathcal{V}_1, \mathcal{E}_1, v_0, l_1)$ and $sl_{\mathfrak{T}_1}$ denote an $\mathcal{EL}$-description tree and associated Skolem labeling s.t. $sl_{\mathfrak{T}_1}(v_0) = \mathtt{sk}_0$, $\mathcal{A} = \mathcal{M}_{\Omega_{\mathsf{C}}}(sl_{\mathfrak{T}_1})$ and $\mathcal{T} \models U_1 \sqsubseteq U_{\mathfrak{T}_1}$.

For any $\mathcal{EL}$-description tree $\mathfrak{T}_2 = (\mathcal{V}_2, \mathcal{E}_2, w_0, l_2)$ with a weak homomorphism ϕ from $\mathfrak{T}_2$ to $\mathfrak{T}_1$, the following equivalence holds:

(EL) $\mathcal{T} \models U_{\mathfrak{T}_2} \sqsubseteq U_2$

if and only if

(FO) there is a Skolem labeling $sl_{\mathfrak{T}_2}$ for $\mathfrak{T}_2$ such that

$sl_{\mathfrak{T}_2}(v) = sl_{\mathfrak{T}_1}(\phi(v))$ for all $v \in \mathcal{V}_2$, and

$\Phi \models \bigvee_{v\in\mathcal{V}_2, Q\in l_{\mathfrak{T}_2}(v)} \neg Q^-(sl_{\mathfrak{T}_2}(v))$.

Proof. We first show that **(EL)** implies **(FO)**. We thus assume that $\mathcal{T} \models U_{\mathfrak{T}_2} \sqsubseteq U_2$. We define $sl_{\mathfrak{T}_2}$ as $sl_{\mathfrak{T}_2}(v) = sl_{\mathfrak{T}_1}(\phi(v))$ for all $v \in \mathcal{V}_{\mathfrak{T}_2}$. Since ϕ is a weak homomorphism from $\mathfrak{T}_2$ to $\mathfrak{T}_1$, $sl_{\mathfrak{T}_2}(v)$ is a Skolem labeling for $U_{\mathfrak{T}_2}$. It remains only to show that $\Phi \models \bigvee_{v\in\mathcal{V}_2, Q\in l_{\mathfrak{T}_2}(v)} \neg Q^-(sl_{\mathfrak{T}_2}(v))$.

By assumption $\mathcal{T} \models U_{\mathfrak{T}_2} \sqsubseteq U_2$, thus $\mathtt{fo}(\mathcal{T}) \models \neg\, \mathtt{fo}(U_{\mathfrak{T}_2}, x) \vee \mathtt{fo}(U_2, x)$ by a direct translation, from which we deduce $\mathtt{fo}(\mathcal{T}) \wedge \neg\, \mathtt{fo}(U_2, \mathtt{sk}_0) \models \neg\, \mathtt{fo}(U_{\mathfrak{T}_2}, \mathtt{sk}_0)$. This entailment also holds for the renamed versions of $\mathcal{T}$, $U_{\mathfrak{T}_2}$ and U_2 so $\Phi \models \neg\, \mathtt{fo}(U^-_{\mathfrak{T}_2}, \mathtt{sk}_0)$. The clause $\neg\, \mathtt{fo}(U^-_{\mathfrak{T}_2}, \mathtt{sk}_0)$ has the following form:

$$\neg\, \mathtt{fo}(U^-_{\mathfrak{T}_2}, \mathtt{sk}_0) = \bigvee_{urv\in\mathcal{E}_2} \neg R(t_u, t_v) \vee \bigvee_{v\in\mathcal{V}_2,\, Q\in l_2(v)} \neg Q^-(t_v)$$

where, $t_{w_0} = \mathtt{sk}_0$ and for all $v \in \mathcal{V}_{\mathfrak{T}_2} \setminus \{w_0\}$, t_v is a variable uniquely associated with v.

Since $\mathcal{M}(sl_{\mathfrak{T}_1}) \subseteq \mathcal{PI}^{g+}_{\Sigma}(\Phi)$, in particular $\mathcal{M}_{\Omega_\mathsf{R}}(sl_{\mathfrak{T}_1}) \subseteq \mathcal{PI}^{g+}_{\Sigma}(\Phi)$. Moreover, for each edge $urv \in E_{\mathfrak{T}_2}$, $\Phi \models R(sl_{\mathfrak{T}_2(u)}, sl_{\mathfrak{T}_2(v)})$ and thus $R(sl_{\mathfrak{T}_2(u)}, sl_{\mathfrak{T}_2(v)}) \in \mathcal{M}_{\Omega_\mathsf{R}}(sl_{\mathfrak{T}_1})$ can be derived from Φ using the resolution calculus. Each of these atomic clauses can be resolved away with the corresponding literal $\neg R(t_u, t_v)$ in $\neg\, \mathtt{fo}(U^-_{\mathfrak{T}_2}, \mathtt{sk}_0)$. In this derivation, all t_v variables are replaced with $sl_{\mathfrak{T}_2}(v)$. In addition $\mathtt{sk}_0 = sl_{\mathfrak{T}_2}(w_0)$, thus the clause $\bigvee_{v\in\mathcal{V}_2,\, Q\in l_2(v)} \neg Q^-(sl_{\mathfrak{T}_2}(v))$ is derivable from Φ by resolution, and thus $\Phi \models \bigvee_{v\in\mathcal{V}_2,\, Q\in l_2(v)} \neg Q^-(sl_{\mathfrak{T}_2}(v))$, so **(FO)** holds.

Let us now assume **(FO)** in order to prove **(EL)**. We consider a Herbrand interpretation $\mathcal{I} = \mathcal{PI}^{g+}_{\Sigma}(\Phi) \cup \bigcup_i \mathcal{I}_i$ where $\mathcal{I}_i$ for $i \in \mathbb{N}$ is defined inductively as:

- $\mathcal{I}_0 = \{Q^-(t) \mid Q(t) \in \mathcal{M}_{\Omega_\mathsf{C}}(sl_{\mathfrak{T}_2})\} \cup \mathcal{M}_{\Omega_\mathsf{R}}(sl_{\mathfrak{T}_2})$ and,
- given $\mathcal{I}_j$,

$$\begin{aligned}\mathcal{I}_{j+1} \;=\; & \mathcal{I}_j \cup \{Q^-(t) \mid t \in (V^-)^{\mathcal{I}_j},\, \neg\, \mathtt{fo}(V^-, x) \vee Q^-(x) \in \Phi\} \\ & \cup \{Q^-(\mathtt{sk}(t)), R(t, \mathtt{sk}(t)) \mid t \in (P^-)^{\mathcal{I}_j}, \\ & \quad \neg\, \mathtt{fo}(P^-, x) \vee Q^-(\mathtt{sk}(x)) \in \Phi, \\ & \quad \neg\, \mathtt{fo}(P^-, x) \vee R(x, \mathtt{sk}(x)) \in \Phi\} \\ & \cup \{Q^-(t) \mid \mathtt{sk}(t) \in (P^-)^{\mathcal{I}_j},\, (t, \mathtt{sk}(t)) \in R^{\mathcal{I}_j}, \\ & \quad \neg R(x,y) \vee \neg P^-(y) \vee Q^-(x) \in \Phi\}.\end{aligned}$$

The $\mathcal{I}_i$s are built to collect all the elements necessary to make the renamed part of Φ true, one step at a time, starting from an interpretation that satisfies $\bigwedge_{v\in\mathcal{V}_{\mathfrak{T}_2}, Q\in l_{\mathfrak{T}_2}(v)} Q^-(sl_{\mathfrak{T}_2}(v)) \subseteq \mathcal{M}(sl_{\mathfrak{T}_2})$, and is thus incompatible with Φ under the **(FO)** assumption. Indeed since $\mathcal{T}$, and by extension $\mathcal{T}^-$, is in normal form, it contains only concept inclusions of the form $V \sqsubseteq Q$, $P \sqsubseteq \exists R.Q$ and $\exists R.P \sqsubseteq Q$, where V is either an atomic concept or a conjunction of two atomic concepts. These correspond in Φ respectively to the clauses $\neg\, \mathtt{fo}(V^-, x) \vee Q^-(x)$, to the pair of clauses $\{\neg\, \mathtt{fo}(V^-, x) \vee Q^-(\mathtt{sk}(x)),\, \neg\, \mathtt{fo}(V^-, x) \vee R(x, \mathtt{sk}(x))\}$ and to the clause $\neg R(x, \mathtt{sk}(x)) \vee \neg P^-(\mathtt{sk}(x)) \vee Q^-(x)$. Hence, if $t \in (V^-)^{\mathcal{I}}$ (resp. $\mathtt{sk}(t) \in (P^-)^{\mathcal{I}}$ and $(t, \mathtt{sk}(t)) \in R^{\mathcal{I}}$) then there exists some $i \in \mathbb{N}$ such that $t \in (V^-)^{\mathcal{I}_i}$ (resp. $\mathtt{sk}(t) \in (P^-)^{\mathcal{I}_i}$ and $(t, \mathtt{sk}(t)) \in R^{\mathcal{I}_i}$) and then all concept inclusions where V^- occurs on the right-hand side (resp. where $\exists R.P^-$ occurs on the right-hand side for some R) are satisfied in $\mathcal{I}_{i+1}$.

Since $\mathcal{I}$ includes $\mathcal{PI}_{\Sigma}^{g+}(\Phi)$, the satisfiability of $U_1(\mathtt{sk}_0)$ and the Skolemization of $\mathtt{fo}(\mathcal{T})$ can be shown as in Lemma 4.2.6. Thus, given that $\mathcal{I} \models \Phi \setminus \{\neg U_2^-(\mathtt{sk}_0)\}$ but $\mathcal{I} \not\models \Phi$, necessarily $\mathcal{I} \models U_2^-(\mathtt{sk}_0)$.

To make use of that fact, we must first prove the following statement:

($*$) For any set $\mathcal{B} = \{Q_1^-(t_1), \ldots, Q_k^-(t_k)\} \subseteq \mathcal{I}$, there exists $\mathfrak{T} = (\mathcal{V}, \mathcal{E}, v_0, l)$ and $sl_{\mathfrak{T}}$ s.t. $\mathcal{B} = \{Q^-(sl_{\mathfrak{T}}(v)) \mid Q \in l(v),\, v \in \mathcal{V}\}$ and $\mathcal{T} \models U_{\mathfrak{T}_2} \sqsubseteq U_{\mathfrak{T}}$.

Without loss of generality, we can consider the biggest such $\mathcal{B}$, that is the set of all atoms of the form $P^-(t)$ in $\mathcal{I}$. The smaller $\mathcal{B}$s simply correspond to concepts $U_{\mathfrak{T}}$ with fewer conjuncts.

Since $\mathcal{PI}_{\Sigma}^{g+}(\Phi)$ only contains non-renamed concepts, we prove ($*$) by induction on the $\mathcal{I}_j$ for $j \in \mathbb{N}$. When $j = 0$, $\mathcal{I}_0 = \mathcal{M}(sl_{\mathfrak{T}_2})$, thus $U_T = U_{\mathfrak{T}_2}$ and the result directly follows. Assuming the result holds for a given $\mathcal{I}_j$, let $\mathcal{B} = \{Q_1^-(t_1), \ldots, Q_k^-(t_k) \mid Q_i^-(t_i) \in \mathcal{I}_{j+1}\}$, and let $\mathcal{B}^* = \{Q_1^-(t_1), \ldots, Q_l^-(t_l) \mid Q_i^-(t_i) \in \mathcal{I}_j\}$. The induction hypothesis applies to $\mathcal{B}^*$ and we conclude that there exists an $\mathcal{EL}$-description tree $\mathfrak{T}^* = (\mathcal{V}_*, E_*, v_0, l_*)$ and a Skolem labeling $sl_{\mathfrak{T}^*}$ s.t. $\mathcal{T} \models U_{\mathfrak{T}_2} \sqsubseteq U_{\mathfrak{T}^*}$ and $\mathcal{B}^* = \{Q^-(sl_{\mathfrak{T}}(v)) \mid Q \in l_*(v),\, v \in \mathcal{V}_*\}$. Let us now consider the literals in $\mathcal{B} \setminus \mathcal{B}^*$. They are all of the form $Q^-(t)$ and belong to $\mathcal{I}_{j+1}$. We define $\mathfrak{T}$ and $sl_{\mathfrak{T}}$ by extending $\mathfrak{T}^*$ and $sl_{\mathfrak{T}^*}$. The extension for each $Q^-(t)$ depends of which set it originates from.

- If $Q^-(t) \in \{Q^-(t) \mid t \in (V^-)^{\mathcal{I}_j},\, \neg\, \mathtt{fo}(V^-, x) \vee Q^-(x) \in \Phi\} \setminus \mathcal{I}_j$, let v be the node in $\mathfrak{T}^*$ such that $l_*(v)$ contains all atomic concepts from V and $sl_{\mathfrak{T}^*}(v) = t$. We add Q to $l_*(v)$ and the rest of $\mathfrak{T}^*$ and $sl_{\mathfrak{T}^*}$ is unchanged. Note that, in that case, $V \sqsubseteq Q \in \mathcal{T}$ by construction of Φ.
- If $Q^-(t) \in \{Q^-(t) \mid \mathtt{sk}(t) \in (P^-)^{\mathcal{I}_j},\, (t, \mathtt{sk}(t)) \in R^{\mathcal{I}_j},\, \neg R(x, y) \vee \neg P^-(y) \vee Q^-(x) \in \Phi\} \setminus \mathcal{I}_j$, let v be the node in $\mathfrak{T}^*$ such that $vRw \in E_*$, $sl_{\mathfrak{T}^*}(v) = t$, $sl_{\mathfrak{T}^*}(v) = \mathtt{sk}(t)$, and $P \in l_*(w)$. As in the previous case, we simply add Q to $l_*(v)$. In that case, $\exists R.P \sqsubseteq Q \in \mathcal{T}$ for the corresponding R.
- If $Q^-(\mathtt{sk}(t)) \in \{Q^-(\mathtt{sk}(t)), R(t, \mathtt{sk}(t)) \mid t \in (V^-)^{\mathcal{I}_j},\, \neg\, \mathtt{fo}(V^-, x) \vee Q^-(\mathtt{sk}(x)) \in \Phi,\, \neg\, \mathtt{fo}(V^-, x) \vee R(x, \mathtt{sk}(x)) \in \Phi\} \setminus \mathcal{I}_j$, let v be the node such that $sl_{\mathfrak{T}^*}(v) = t$. Then we add a fresh node w to $\mathcal{V}_*$ as well as an edge vRw to E_*. We also extend $sl_{\mathfrak{T}^*}$ so that w is mapped to t. In that case, $P \sqsubseteq \exists R.Q \in \mathcal{T}$ for the corresponding R.

Note that in all cases, $\mathcal{T} \models U_{\mathfrak{T}^*} \sqsubseteq U_{\mathfrak{T}}$ because the conjunct(s) added from $U_{\mathfrak{T}^*}$ to $U_{\mathfrak{T}}$ is(/are) justified by the concept inclusion from $\mathcal{T}$ that is ultimately to blame for the existence of $Q^-(t)$ in $\mathcal{I}_{j+1} \setminus \mathcal{I}_j$. Since, by the induction hypothesis, $\mathcal{T} \models U_{\mathfrak{T}_2} \sqsubseteq U_{\mathfrak{T}^*}$, $\mathfrak{T}$ and $sl_{\mathfrak{T}}$ are s.t. $\mathcal{B} = \{Q^-(sl_{\mathfrak{T}}(v)) \mid Q \in l(v),\, v \in \mathcal{V}\}$ and $\mathcal{T} \models U_{\mathfrak{T}_2} \sqsubseteq U_{\mathfrak{T}}$, ($*$) holds for that case and thus also for $\mathcal{I}$.

Because $\mathcal{I} \models U_2^-(\mathtt{sk}_0)$, necessarily $U_2^-(\mathtt{sk}_0) \in \mathcal{I}$. Thus, by ($*$), $\mathcal{T} \models U_{\mathfrak{T}_2} \sqsubseteq U_2$ because the $\mathcal{EL}$-description tree $\mathfrak{T}$ from ($*$) in that case is such that $U_{\mathfrak{T}} = U_2$. □

Lemma 4.2.8 ($\mathcal{EL}$-Description Tree and $\preceq_\sqcap$)**.** Given two $\mathcal{EL}$-description trees $\mathfrak{T}_1 = (\mathcal{V}_1, \mathcal{E}_1, v_0, l_1)$ and $\mathfrak{T}_2 = (\mathcal{V}_2, \mathcal{E}_2, w_0, l_2)$, $U_{\mathfrak{T}_1} \preceq_\sqcap U_{\mathfrak{T}_2}$ if and only if there is an (injective) $\emptyset$-homomorphism from $\mathfrak{T}_1$ to $\mathfrak{T}_2$.

Proof. If $U_{\mathfrak{T}_1} \equiv U_{\mathfrak{T}_2}$, we are done because then $\mathfrak{T}_1$ and $\mathfrak{T}_2$ obviously have the same shape. Otherwise, the missing conjuncts in U_1 would correspond to either:

– some missing atomic concepts in some $l_2(w)$ from $\mathfrak{T}_2$ or

– a subtree of $\mathfrak{T}_2$ that is not in the image of the $\emptyset$-homomorphism from $\mathfrak{T}_1$. $\square$

Last, I show how prime implicates are related to the connection-minimal solutions of the abduction problem.

Lemma 4.2.9 ($\mathcal{T}$-Homomorphism and Negative Implicates)**.** Given an abduction problem $\langle \mathcal{T}, \Sigma, U_1 \sqsubseteq U_2 \rangle$, Φ its translation to first-order logic, and a subset of the positive prime implicates $\mathcal{A} = \{P_1(t_1), \ldots, P_k(t_k)\} \subseteq \mathcal{PI}^{g+}_{\Sigma}(\Phi)$. As allowed by Lemma 4.2.6, let $\mathfrak{T}_1 = (\mathcal{V}_1, \mathcal{E}_1, v_0, l_1)$ and $sl_{\mathfrak{T}_1}$ denote an $\mathcal{EL}$-description tree and associated Skolem labeling s.t. $\mathcal{A} = \{P(sl_{\mathfrak{T}_1}(v)) \mid P \in l_1(v), v \in \mathcal{V}_1\}$ and $\mathcal{T} \models U_1 \sqsubseteq U_{\mathfrak{T}_1}$.

For any $\mathcal{EL}$-description tree $\mathfrak{T}_2 = (\mathcal{V}_2, \mathcal{E}_2, w_0, l_2)$ with a weak homomorphism ϕ from $\mathfrak{T}_2$ to $\mathfrak{T}_1$, the following equivalence holds:

$U_{\mathfrak{T}_2}$ is a $\preceq_\sqcap$-minimal concept s.t. $\mathcal{T} \models U_{\mathfrak{T}_2} \sqsubseteq U_2$

if and only if

there is a Skolem labeling $sl_{\mathfrak{T}_2}$ for $\mathfrak{T}_2$ s.t.

$sl_{\mathfrak{T}_2}(v) = sl_{\mathfrak{T}_1}(\phi(v))$ for all $v \in \mathcal{V}_{\mathfrak{T}_2}$, and

$\bigvee_{v \in \mathcal{V}_2, Q \in l_2(v)} \neg Q^-(sl_{\mathfrak{T}_2}(v)) \in \mathcal{PI}^{g-}_{\Sigma}(\Phi)$.

Proof. Thanks to Lemma 4.2.7, we know that, in the conditions of the lemma, the existence of a $U_{\mathfrak{T}_2}$ such that $\mathcal{T} \models U_{\mathfrak{T}_2} \sqsubseteq U_2$ is equivalent to the existence of a Skolem labeling $sl_{\mathfrak{T}_2}$ for $\mathfrak{T}_2$ such that $sl_{\mathfrak{T}_2}(v) = sl_{\mathfrak{T}_1}(\phi(v))$ for all $v \in \mathcal{V}_{\mathfrak{T}_2}$, and $\Phi \models \bigvee_{v \in \mathcal{V}_2, Q \in l_2(v)} \neg Q^-(sl_{\mathfrak{T}_2}(v))$. It remains to show the equivalence between the $\preceq_\sqcap$-minimality of $U_{\mathfrak{T}_2}$ and the fact that $\bigvee_{v \in \mathcal{V}_2, Q \in l_2(v)} \neg Q^-(sl_{\mathfrak{T}_2}(v)) \in \mathcal{PI}^{g-}_{\Sigma}(\Phi)$.

By Lemma 4.2.8, for any two trees $\mathfrak{T}'_2$ and $\mathfrak{T}''_2$ and corresponding Skolem labellings $sl_{\mathfrak{T}'_2}$ and $sl_{\mathfrak{T}''_2}$ for which there are respective weak homomorphisms ϕ_1 and ϕ_2 to $\mathfrak{T}_1$, $U_{\mathfrak{T}'_2} \preceq_\sqcap U_{\mathfrak{T}''_2}$ if and only if $\{Q^-(sl_{\mathfrak{T}'_2}(v)) \mid v \in \mathcal{V}'_2, Q \in l'_2(v)\} \subseteq \{Q^-(sl_{\mathfrak{T}''_2}(v)) \mid v \in \mathcal{V}''_2, Q \in l''_2(v)\}$. Thus it is not possible for $U_{\mathfrak{T}_2}$ to be $\preceq_\sqcap$-minimal if $\bigvee_{v \in \mathcal{V}_2, Q \in l_2(v)} \neg Q^-(sl_{\mathfrak{T}_2}(v))$ is not prime and vice-versa. $\square$

4.2.5 Hypothesis Reconstruction

We are now ready to see how to reconstruct the hypotheses in $\mathcal{EL}$ from the prime implicates in FOL. This is basically a consequence of Lemma 4.2.9. We focus on hypotheses of a certain shape called *constructible hypothesis*. A constructible hypothesis is built from the concepts in *one* negative prime implicate in $\mathcal{PI}^{g-}_{\Sigma}(\Phi)$ and *all* matching concepts from prime implicates in $\mathcal{PI}^{g+}_{\Sigma}(\Phi)$. The matching itself is determined by the Skolem terms that occur in all these clauses. The subterm relation between the terms of the clauses in $\mathcal{PI}^{g+}_{\Sigma}(\Phi)$ and $\mathcal{PI}^{g-}_{\Sigma}(\Phi)$ is the same as the ancestor relation in the description trees of subsumers of U_1 and subsumees of U_2 respectively. The terms matching in positive and negative prime implicates allow us to identify where the missing entailments between a subsumer V_1 of U_1 and a subsumee V_2 of U_2 are. These missing entailments become the constructible $\mathcal{H}$. The condition $U_{\mathcal{B},t} \not\preceq_\sqcap U_{\mathcal{A},t}$ is a way to write that $U_{\mathcal{A},t} \sqsubseteq U_{\mathcal{B},t}$ is not a tautology, which can be tested by subset inclusion.

Example 4.2.10. Consider the abduction problem $\langle \mathcal{T}_a, \Sigma, \alpha_a \rangle$ where $\Sigma = \Omega_\mathsf{C}$ from Ex. 4.1.1. For the translation Φ of this problem, we have

$$\begin{aligned}\mathcal{PI}^{g+}_{\Sigma}(\Phi) &= \{\mathsf{Professor}(\mathtt{sk}_0),\ \mathsf{Doctor}(\mathtt{sk}_0),\ \mathsf{Chair}(\mathtt{sk}_1(\mathtt{sk}_0)),\ \mathsf{PhD}(\mathtt{sk}_2(\mathtt{sk}_0))\}\\ \mathcal{PI}^{g-}_{\Sigma}(\Phi) &= \{\neg\mathsf{Researcher}^-(\mathtt{sk}_0),\\ &\quad\ \neg\mathsf{ResearchPosition}^-(\mathtt{sk}_1(\mathtt{sk}_0)) \vee \neg\mathsf{Diploma}^-(\mathtt{sk}_2(\mathtt{sk}_0))\}\end{aligned}$$

where $\mathtt{sk}_1$ is the Skolem function introduced for $\mathsf{Professor} \sqsubseteq \exists\mathsf{employment.Chair}$ and $\mathtt{sk}_2$ is introduced for $\mathsf{Doctor} \sqsubseteq \exists\mathsf{qualification.PhD}$. This leads to two constructible solutions: $\{\mathsf{Professor} \sqcap \mathsf{Doctor} \sqsubseteq \mathsf{Researcher}\}$ and $\mathcal{H}_{a1}$, that are both packed connection-minimal hypotheses if $\Sigma = \Omega_\mathsf{C}$.

Theorem 4.2.11. Let $\langle \mathcal{T}, \Sigma, U_1 \sqsubseteq U_2 \rangle$ be an abduction problem and Φ be its first-order translation. Then, a TBox $\mathcal{H}'$ is a packed connection-minimal solution to the problem if and only if an equivalent hypothesis $\mathcal{H}$ can be constructed from non-empty sets $\mathcal{A}$ and $\mathcal{B}$ of atoms verifying:

- $\mathcal{B} = \{Q_1(t_1), \ldots, Q_m(t_m)\}$ s.t. $\big(\neg Q_1^-(t_1) \vee \cdots \vee \neg Q_m^-(t_m)\big) \in \mathcal{PI}^{g-}_{\Sigma}(\Phi)$,
- for all $t \in \{t_1, \ldots, t_n\}$ there exists an P s.t. $P(t) \in \mathcal{PI}^{g+}_{\Sigma}(\Phi)$,
- $\mathcal{A} = \{P(t) \in \mathcal{PI}^{g+}_{\Sigma}(\Phi) \mid t \text{ is one of } t_1, \ldots, t_m\}$, and
- $\mathcal{H} = \{U_{\mathcal{A},t} \sqsubseteq U_{\mathcal{B},t} \mid t \text{ is one of } t_1, \ldots, t_m \text{ and } U_{\mathcal{B},t} \not\preceq_\sqcap U_{\mathcal{A},t}\}$, where $U_{\mathcal{A},t} = \bigsqcap_{P(t)\in\mathcal{A}} P$ and $U_{\mathcal{B},t} = \bigsqcap_{Q(t)\in\mathcal{B}} Q$.

Proof. Let $\langle \mathcal{T}, \Sigma, U_1 \sqsubseteq U_2 \rangle$ be an abduction problem and Φ be its first-order translation.

We begin by assuming given a packed connection-minimal hypothesis $\mathcal{H}$. Then there exist concepts V_1 and V_2, and weak homomorphism ϕ verifying points 1-3 of Def. 4.1.9 while $\mathcal{H}$ verifies point 4 of the same definition for these V_1, V_2 and ϕ. W.l.o.g., we consider that V_1 is such that every node in $\mathfrak{T}_{V_1}$ is in the range of ϕ. Such a V_1 can always be obtained from a V_1 that has too much nodes by pruning the extra nodes, since they cannot have children that are in the range of ϕ. Since, by Def. 4.1.9 point 1, $\mathcal{T} \models U_1 \sqsubseteq V_1$, by Lemma 4.2.5 there exists a Skolem labeling sl_1 for $\mathfrak{T}_{V_1} = (\mathcal{V}_1, \mathcal{E}_1, v_0, l_1)$ s.t. $\mathcal{M}(sl_1) \subseteq \mathcal{PI}^{g+}_{\Sigma}(\Phi)$. Since $\mathcal{H}$ is packed, $l_1(v)$ is also maximal for each v, so that $\mathcal{M}_{\Omega_\mathsf{C}}(sl_1) = \{P(sl_1(v)) \in \mathcal{PI}^{g+}_{\Sigma}(\Phi) \mid v \in \mathcal{V}_1\}$. Note that $\mathcal{M}_{\Omega_\mathsf{C}}(sl_1)$ cannot be empty and that there are no two nodes in $\mathfrak{T}_{V_1}$ with the same Skolem label, otherwise $\mathcal{M}(sl_1) \subseteq \mathcal{PI}^{g+}_{\Sigma}(\Phi)$ would not hold since this would imply that two occurrences of Skolem terms in Φ share the same Skolem function, which is forbidden in the standard Skolemization procedure. From point 3 of Def. 4.1.9, we know that ϕ is a weak homomorphism from $\mathfrak{T}_{V_2}$ to $\mathfrak{T}_{V_1}$ and from point 2 that V_2 is a $\preceq_\sqcap$-minimal concept s.t. $\mathcal{T} \models V_2 \sqsubseteq U_2$. Hence, by Lemma 4.2.9, there also exists a Skolem labeling sl_2 for $\mathfrak{T}_{V_2} = (\mathcal{V}_2, \mathcal{E}_2, w_0, l_2)$ s.t. $sl_2(v) = sl_1(\phi(v))$ for all $v \in \mathcal{V}_2$ and $\bigvee_{v\in\mathcal{V}_2, Q\in l_2(v)} \neg Q^-(sl_2(v)) \in \mathcal{PI}^{g-}_{\Sigma}(\Phi)$. Since $\{Q(sl_2(v)) \mid v \in \mathcal{V}_2, Q \in l_2(v)\} = \mathcal{M}_{\Omega_\mathsf{C}}(sl_2)$, we define $\mathcal{B}$ as $\mathcal{M}_{\Omega_\mathsf{C}}(sl_2)$. Our choice of V_1 allows us to define $\mathcal{A}$ as $\mathcal{M}_{\Omega_\mathsf{C}}(sl_1)$ since it holds that $sl_2(v) = sl_1(\phi(v))$ and there are no nodes in $\mathcal{V}_1$ outside the range of ϕ. Thus $\mathcal{A}$ and $\mathcal{B}$ verify the first two points of Th. 4.2.11. Let us now consider any concept inclusion in $\mathcal{H}$. It is of the form $\bigsqcap l_1(\phi(w)) \sqsubseteq \bigsqcap l_2(w)$ for some $w \in \mathcal{V}_2$ and s.t. $\mathcal{T} \not\models \bigsqcap l_1(\phi(w)) \sqsubseteq \bigsqcap l_2(w)$. For every $v \in \mathcal{V}_1$, consider all $w_1, \ldots, w_k \in \mathcal{V}_2$ s.t. $\phi(w_1) = \ldots = \phi(w_k) = v$. Then, $\mathcal{H}$ contains $\{\bigsqcap l_1(v) \sqsubseteq \bigsqcap l_2(w_1), \ldots, \bigsqcap l_1(v) \sqsubseteq \bigsqcap l_2(w_k)\}$ which is equivalent to $\{\bigsqcap l_1(v) \sqsubseteq (\bigsqcap l_2(w_1)) \sqcap \ldots \sqcap (\bigsqcap l_2(w_1))\} = \{U_{\mathcal{A},sl_1(v)} \sqsubseteq U_{\mathcal{B},sl_1(v)}\}$ and since $\mathcal{T} \not\models \bigsqcap l_1(v) \sqsubseteq \bigsqcap l_2(w_i)$ for all $i \in \{1, \ldots, k\}$, also $\mathcal{T} \not\models U_{\mathcal{A},t} \sqsubseteq U_{\mathcal{B},t}$ for $t = sl_1(v)$. This means in particular that this CI is not a tautology, ensuring that $U_{\mathcal{B},t} \not\preceq_{\sqcap} U_{\mathcal{A},t}$. Thus $\mathcal{H}$ is equivalent to the constructible hypothesis built for $\mathcal{A}$ and $\mathcal{B}$ as just defined.

Now, let us consider that $\mathcal{H}$ is a constructible hypothesis obtained from a given $\mathcal{A}$ and $\mathcal{B}$ verifying the constraints from Th. 4.2.11. Then $\mathcal{A}$ is a subset of $\mathcal{PI}^{g+}_{\Sigma}(\Phi)$, thus, by Lemma 4.2.6, there is a description tree $\mathfrak{T}_1 = (\mathcal{V}, \mathcal{E}, v_0, l_1)$ and associated Skolem labeling sl s.t. $\mathcal{A} = \mathcal{M}_{\Omega_{\mathsf{C}}}(sl)$ and $\mathcal{T} \models U_1 \sqsubseteq U_{\mathfrak{T}_1}$. We define $\mathfrak{T}_2 = (\mathcal{V}, \mathcal{E}, v_0, l_2)$, where for all $v \in \mathcal{V}$, $l_2(v) = \{Q \mid Q(sl(v)) \in \mathcal{B}\}$ and ϕ as the identity over $\mathcal{V}$. Then ϕ is a weak homomorphism from $\mathfrak{T}_2$ to $\mathfrak{T}_1$ and sl can also be associated to $\mathfrak{T}_2$ and it is such that $\left(\bigvee_{v\in\mathcal{V},Q\in l_2(v)} \neg Q^-(sl(v))\right) \in \mathcal{PI}^{g-}_{\Sigma}(\Phi)$. Thus, by Lemma 4.2.9, $U_{\mathfrak{T}_2}$ is a $\preceq_{\sqcap}$-minimal concept s.t. $\mathcal{T} \models U_{\mathfrak{T}_2} \sqsubseteq U_2$. As seen in the first part of this proof, sl must be injective on $\mathcal{V}$ due to its association with $\mathfrak{T}_1$, and thus, for all $v \in \mathcal{V}$, $U_{\mathcal{A},sl(v)} = \bigsqcap l_1(v)$ and by construction of $\mathfrak{T}_2$, $U_{\mathcal{B},sl(v)} = \bigsqcap l_2(v)$. If $\mathcal{T} \models \bigsqcap l_1(v) \sqsubseteq \bigsqcap l_2(v)$, then $\mathcal{T} \models U_{\mathcal{A},sl(v)} \sqsubseteq U_{\mathcal{B},sl(v)}$. We show that this implies $U_{\mathcal{B},sl(v)} \preceq_{\sqcap} U_{\mathcal{A},sl(v)}$. Consider any t s.t. $\mathcal{T} \models U_{\mathcal{A},t} \sqsubseteq U_{\mathcal{B},t}$. Then by translation, it means that $\Phi \models \neg\, \mathtt{fo}(U_{\mathcal{A},t}) \vee \mathtt{fo}(U_{\mathcal{B},t})$ and since both concepts do not contain role restrictions, it means in particular that $\Phi \models \bigvee_{P\in U_{\mathcal{A},t}} \neg P(x) \vee Q(x)$ for all $Q \in U_{\mathcal{B},t}$. Since $P(t) \in \mathcal{PI}^{g+}_{\Sigma}(\Phi)$ for all $P \in U_{\mathcal{A},t}$, $\Phi \models Q(t)$ for all $Q \in U_{\mathcal{B},t}$ and since those are atomic ground positive implicates, for all $Q \in U_{\mathcal{B},t}$, $Q(t) \in \mathcal{PI}^{g+}_{\Sigma}(\Phi)$. Furthermore, by definition of $\mathcal{A}$ and $U_{\mathcal{A},t}$, this leads to $Q \in U_{\mathcal{A},t}$ for all $Q \in U_{\mathcal{B},t}$, hence $U_{\mathcal{B},t} \preceq_{\sqcap} U_{\mathcal{A},t}$. In the particular case that interests us, it means that $U_{\mathcal{B},sl(v)} \preceq_{\sqcap} U_{\mathcal{A},sl(v)}$ as wanted. Hence $\mathcal{H}$ is a connection-minimal hypothesis. It remains only to show that it is packed. Any tree $\mathfrak{T}'$ built from $\mathfrak{T}_1$ by extending the label of some $v \in \mathcal{V}$ must be such that $\mathcal{M}(sl_{\mathfrak{T}'}) \not\subseteq \mathcal{PI}^{g+}_{\Sigma}(\Phi)$, where $sl_{\mathfrak{T}'}$ is identical to sl but associated to $\mathfrak{T}'$, since the labels of $\mathfrak{T}_1$ are already maximal in that regard. Thus, by Lemma 4.2.5, $\mathcal{T} \not\models U_1 \sqsubseteq U_{\mathfrak{T}'}$, hence any such $U_{\mathfrak{T}'}$ cannot be used to create constructible hypotheses, proving $\mathcal{H}$ packed. □

As an illustration, consider

$$\begin{aligned} \mathcal{T} = \{&U_1 \sqsubseteq \exists R_1.(P \sqcap Q), \\ &\exists R_1.U \sqcap \exists R_1.V \sqsubseteq U_2\}. \end{aligned}$$

The negative prime implicate $\neg U^-(sk_1(sk_0)) \vee \neg V^-(sk_1(sk_0))$ corresponds to a tree and associated Skolem labeling as follows:

$$\mathfrak{T}_2 = (\{v_0, v_1, v_2\}, \{v_0 R_1 v_1, v_0 R_1 v_2\}, v_0, l_2)$$

s.t. $l_2(v_0) = \emptyset$, $l_2(v_1) = \{U\}$ and $l_2(v_2) = \{V\}$; and $sl_{\mathfrak{T}_2}(v_0) = \mathtt{sk}_0$, $sl_{\mathfrak{T}_2}(v_1) = sl_{\mathfrak{T}_2}(v_2) = \mathtt{sk}_1(\mathtt{sk}_0)$. For $U_{\mathfrak{T}_2} = \exists R_1.U \sqcap \exists R_1.V$, and $U_{\mathfrak{T}_1} = \exists R_1.(P \sqcap Q)$ the set $\{P \sqcap Q \sqsubseteq U, P \sqcap Q \sqsubseteq V\}$ is a packed connection-minimal hypothesis and the equivalent constructible hypothesis $\{P \sqcap Q \sqsubseteq U \sqcap V\}$ is the one found by applying Th. 4.2.11.

4.3 Efficiency

Run-time improvement in the implementation is done on two aspects: Inference restriction in SPASS and ontology size reduction in Java preprocessing. The first includes using SOS strategy and restricting the number of variables. The second one is based on the fact that connection-minimality does not use concept inclusions unrelated to the problem at hand. We show that such concept inclusions can be removed by using the *locality-based module*. Some of the results described in this section will also be useful to derive termination in the next section.

4.3.1 Inference Restrictions in FOL

While resolution calculus is already sufficient to generate all useful prime implicates, the clause shape allows for additional inference restrictions. Considering the set of formulas $\mathtt{fo}(\mathcal{T})$ after the translation, the sets Φ and Φ_p only contain clauses of the following shapes:

I1: $U_1(\mathtt{sk}_0)$,

I2: $\neg U_2^-(\mathtt{sk}_0)$,

I3: $\neg P_1(x) \vee P_2(x)$,

I4: $\neg P_1(x) \vee \neg P_2(x) \vee P_3(x)$,

I5: $\neg R(x, y) \vee \neg P_1(y) \vee P_2(x)$,

I6: $\neg P_1(x) \vee R(x, \mathtt{sk}(x))$, and

I7: $\neg P_1(x) \vee P_2(\mathtt{sk}(x))$,

where P_1, P_2 and P_3 are either all original literals or all duplicate literals. I abbreviate a "clause of the form **I**x" as an "**I**x:C_x-clause " for $x \in \{1, .., 7\}$ with C_x its respective clause (possibly omitted). Observe that there is exactly one **I1**:$U_1(\mathtt{sk}_0)$-clause and one **I2**:$\neg U_2^-(\mathtt{sk}_0)$-clause, both for the same constant $\mathtt{sk}_0$. Moreover, for every Skolem function $\mathtt{sk}$ occurring in Φ, there is exactly one pair of clauses where one is an **I6**:-clause and the other an **I7**:-clause where a given $\mathtt{sk} \in \mathsf{\Pi_S}$ occurs. We call them the clauses *introducing* $\mathtt{sk}$. To every Skolem function $\mathtt{sk}$, we associate the atomic concept $P_{\mathtt{sk}}$ that occurs positively in the **I7**:-clause introducing $\mathtt{sk}$.

Before the prime implicate generation, we additionally compute the *presaturation* Φ_p of the set of clauses Φ, defined as

$$\Phi_p = \Phi \cup \{\neg P(x) \vee Q(x) \mid \Phi \models \neg P(x) \vee Q(x)\}$$

where P and Q are either both original or both duplicate atomic concepts. One can easily see that the new clauses in Φ_p are **I3**:-clause and thus not introducing new clause shapes.

The presaturation can be efficiently computed before the translation, using a modern $\mathcal{EL}$ reasoner such as ELK [KKS14], which is highly optimized towards the computation of all entailed concept inclusions shaped $P \sqsubseteq Q$ (where P and Q are atomic). While the presaturation computes nothing a resolution procedure could not derive, this is useful because it allows us to derive all prime implicates of a certain depth without first deriving another one with a higher term depth. By this presaturation, I now show that all the relevant prime implicates can be computed if all inferences

R1 use at least one premise with a ground term, and

R2 derive a resolvent with at most one variable.

The first restriction **R1** means that we can use SOS resolution derivation Th. 3.2.7 [HTW21] with set of support $\{U_1(\mathtt{sk}_0), U_2^-(\mathtt{sk}_0)\}$ (the only clauses with ground terms in Φ). This restriction is possible because we only want ground implicates, and that the non-ground clauses in Φ cannot entail the empty clause ($\mathcal{EL}$ TBoxes are always consistent).

The second restriction **R2** can be proven directly by looking into the possible clause shapes and the fact that $\{U_1(\mathtt{sk}_0), U_2^-(\mathtt{sk}_0)\}$ are necessary. In fact, for $\mathcal{PI}_\Sigma^{g+}(\Phi)$ it is even possible to restrict inferences to generating only ground resolvents.

Relying on Φ_p allows to derive all ground implicates by increasing term depth, which is possible thanks to the following result.

Lemma 4.3.1. It is not necessary to use clauses of the form **I5**:$\neg R(x,y) \vee \neg P_1(y) \vee P_2(x)$ to derive $\mathcal{PI}_\Sigma^{g+}(\Phi)$ from Φ_p by resolution.

Proof. Since Φ and Φ_p are equivalent, they have the same prime implicates, that can be derived by resolution from any of them. We construct a Herbrand model for Φ from all clauses in Φ_p except the **I5**:$\neg R(x,y) \vee \neg P_1(y) \vee P_2(x)$-clauses. Then, by Lemma 4.2.4, all clauses from $\mathcal{PI}_\Sigma^{g+}(\Phi)$ will be included in this model and thus derivable by resolution from the restriction of Φ_p to non-**I5**:$\neg R(x,y) \vee \neg P_1(y) \vee P_2(x)$-clauses.

Let $\mathcal{I} = \bigcup_i \mathcal{I}_i$ for $i \in \mathbb{N}$, such that:

- $\mathcal{I}_0 = \{U_1(\mathtt{sk}_0)\}$ and,
- given $\mathcal{I}_j$,

$$\begin{aligned}\mathcal{I}_{j+1} \quad = \quad & \mathcal{I}_j \cup \{Q(t) \mid t \in (D)^{\mathcal{I}_j},\ \neg\,\mathtt{fo}(D,x) \vee Q(x) \in \Phi_p\} \\ & \cup \{Q(\mathtt{sk}(t)), R(t,\mathtt{sk}(t)) \mid t \in (P)^{\mathcal{I}_j}, \\ & \quad \neg\,\mathtt{fo}(P,x) \vee Q(\mathtt{sk}(x)) \in \Phi_p,\ \neg\,\mathtt{fo}(P,x) \vee R(x,\mathtt{sk}(x)) \in \Phi_p\}.\end{aligned}$$

This interpretation is similar to the one used in the proof of Lemma 4.2.4, but uses the clauses in Φ_p, **I5**:$\neg R(x,y) \vee \neg P_1(y) \vee P_2(x)$-clauses excepted, instead of the clauses in Φ. Thus every atom in $\mathcal{I}$ can be derived by resolution from the clauses of Φ_p that are not **I5**:$\neg R(x,y) \vee \neg P_1(y) \vee P_2(x)$-clauses.

We show that $\mathcal{I}$ is a model of Φ. By construction, $\mathcal{I} \models U_1(\mathtt{sk}_0)$ and all

- **I1**:$U_1(\mathtt{sk}_0)$,
- **I3**:$\neg P_1(x) \vee P_2(x)$,
- **I4**:$\neg P_1(x) \vee \neg P_2(x) \vee P_3(x)$,
- **I6**:$\neg P_1(x) \vee R(x,\mathtt{sk}(x))$, and
- **I7**:$\neg P_1(x) \vee P_2(\mathtt{sk}(x))$

with original predicates in Φ since they also occur in Φ_p. The clauses with duplicate predicates are also satisfied since $\mathcal{I}$ contains no duplicates at all. It remains only to show that the clauses of the form **I5**:$\neg R(x,y) \vee \neg P_1(y) \vee P_2(x)$ in Φ are true in $\mathcal{I}$. By contradiction, consider that the clause $\varphi = \neg R(x,y) \vee \neg P_1(y) \vee P_2(x)$ is not satisfied by $\mathcal{I}$. Then there must exist terms t, t' such that $R(t,t') \in \mathcal{I}$, $P_1(t') \in \mathcal{I}$ but $P_2(t) \notin \mathcal{I}$. By construction, $t' = \mathtt{sk}(t)$ for some Skolem function $\mathtt{sk}$. The only clauses with $\mathtt{sk}$ in Φ_p are the clauses introducing $\mathtt{sk}$, that we denote by $\varphi_1 = \neg P_3(x) \vee R(x, \mathtt{sk}(x))$ and $\varphi_2 = \neg P_3(x) \vee P_4(\mathtt{sk}(x))$ for some original atomic concept P_4. These clauses are the only possible cause for the presence of $R(t, \mathtt{sk}(t))$ and $P_1(\mathtt{sk}(t))$ in $\mathcal{I}_j$ for some $j \geq 1$, and thus there must be an $i < j$ s.t. $P_3(t) \in \mathcal{I}_i$. The presence of φ_1, φ_2 and φ in Φ ensures that $\Phi \models \neg P_3(x) \vee P_2(x)$ and thus that $\neg P_3(x) \vee P_2(x) \in \Phi_p$. Combined with the fact that $P_3(t) \in \mathcal{I}_i$, it means that $P_2(t) \in \mathcal{I}_{i+1} \subseteq \mathcal{I}$, a contradiction. Thus $\mathcal{I}$ is also a model of all clauses of the form **I5**:$\neg R(x,y) \vee \neg P_1(y) \vee P_2(x)$ in Φ, so it is a model of Φ and it is possible to derive all clauses in $\mathcal{PI}^{g+}_{\Sigma}(\Phi)$ from Φ_p without using the clauses of the form **I5**:$\neg R(x,y) \vee \neg P_1(y) \vee P_2(x)$. □

A direct consequence of this lemma is that, regarding derivations of $\mathcal{PI}^{g+}_{\Sigma}(\Phi)$, we only need to consider those where every inference preserves or increases the depth of terms from premises to conclusion, because the only way to decrease this depth is by using an **I5**:$\neg R(x,y) \vee \neg P_1(y) \vee P_2(x)$-clause. This allows us to prove Th. 4.3.2

Theorem 4.3.2 (Restriction **R2**)**.** Given an abduction problem and its translation Φ, every constructible hypothesis can be built from prime implicates that are inferred under restriction **R2**.

Proof. By Th. 4.2.11, it suffices to show that all clauses in $\mathcal{PI}^{g+}_{\Sigma}(\Phi) \cup \mathcal{PI}^{g-}_{\Sigma}(\Phi)$ that contain no binary predicate can be derived using only inferences of clauses with at most one variable.

Since **I1**:$U_1(\mathtt{sk}_0)$ is the only clause containing no negative literals, any clause $\varphi \in \mathcal{PI}^{g+}_{\Sigma}(\Phi)$ must be derived using the **I1**:$U_1(\mathtt{sk}_0)$-clause. Moreover **I5**:$\neg R(x,y) \vee \neg P_1(y) \vee P_2(x)$-clauses are the only ones in the input that would introduce a variable when resolved with a ground clause. By Lemma 4.3.1, we can ignore these clauses to infer $\varphi \in \mathcal{PI}^{g+}_{\Sigma}(\Phi)$. Thus, any clause in $\mathcal{PI}^{g+}_{\Sigma}(\Phi)$ can be derived by inferring only ground clauses from Φ_p, which is even more than what **R2** requires.

For $\varphi' \in \mathcal{PI}^{g-}_{\Sigma}(\Phi)$, Lemma 4.3.1 does not apply. In general, any derivation from Φ_p of a clause that contains a constant involves $U_1(\mathtt{sk}_0)$ or $\neg U_2^-(\mathtt{sk}_0)$ or both, and only **I5**:$\neg R(x,y) \vee \neg P_1(y) \vee P_2(x)$-clauses would introduce a variable into such a derivation. Let φ be the first clause with a variable that occurs as a resolvent in the derivation of φ' from Φ_p, and let φ' be without binary predicates, since it must be usable to build a constructible hypothesis following Th. 4.2.11. The premises of the inference producing φ are a ground clause and an **I5**:$\neg R(x,y) \vee \neg P_1(y) \vee P_2(x)$-clause, $\neg R(x,y) \vee \neg P_1(y) \vee P_2(x)$. We show that any occurrence of a variable in φ can be immediately eliminated by another inference, creating a new derivation for φ'. Depending on the literal resolved upon in the **I5**:$\neg R(x,y) \vee \neg P_1(y) \vee P_2(x)$-clause to obtain φ several cases occur.

- This literal cannot be $\neg R(x,y)$, or φ would be ground because both x and y would be unified with ground terms.

– If the literal resolved upon is $P_2(x)$, then y occurs in φ as its only variable, in the literals $\neg R(t,y)$ for some ground t and $\neg P_1(y)$. The literal $\neg R(t,y)$ is eliminated later in the derivation since φ' contains no binary predicate. All positive occurrences of R that can be derived are of the form $R(t', \mathtt{sk}(t'))$ for some $\mathtt{sk}$, because in the Φ_p, the only positive occurrences of R are found in **I6**:$\neg P_1(x) \vee R(x, \mathtt{sk}(x))$-clauses. Thus, to obtain a clause in $\mathcal{PI}^{g-}_{\Sigma}(\Phi)$ without roles, we need to eventually unify the variable y with a ground term of the form $\mathtt{sk}(t)$ for some $\mathtt{sk}$. Since Φ_p is Horn, we can rearrange any derivation from φ to φ' so that we first resolve upon $\neg R(t,y)$ in φ with the suitable **I6**:$\neg P_1(x) \vee R(x, \mathtt{sk}(x))$-clause, i.e., the one introducing the appropriate $\mathtt{sk}$. As a result, we obtain another ground clause with no variables, before any further inference is performed if needed.

– If the literal resolved upon is $\neg P_1(y)$, then x occurs in φ in the literals $P_2(x)$ and $\neg R(x,t)$, where t is ground. The argument unfolds as in the previous case, with the nuance that the considered $\mathtt{sk}$ function is the one s.t. $t = \mathtt{sk}(t')$ for some t'.

It follows that a derivation of φ' that does not respect **R2** can always be rearranged to eliminate occurrences of variables (and binary literals) as soon as they occur, before the next variable is introduced. The rearranged derivation respects **R2**.

□

4.3.2 Locality-based Modules in DL

Realistic ontologies may be unnecessarily large in comparison to the relevant set of axioms actually needed for the connection-minimality. If the subsumers of U_1 and subsumees of U_2 can be deduced from some smaller subset of the ontology, then we can still get our desired hypotheses. By using module extraction, it is possible to obtain such subset.

Given a signature Σ' and an TBox $\mathcal{T}$, a *module* $\mathcal{M}$ of $\mathcal{T}$ for Σ' is a subset of $\mathcal{T}$ that preserves all entailments of closed second-order formulas using only predicates from Σ. The signature Σ' from the abduction problem $\langle \mathcal{T}, \Sigma, U_1 \sqsubseteq U_2 \rangle$ that are actually useful is not known in advance. For this, we need to ensure that the module $\mathcal{M}$ preserves all subsumers of U_1 and all subsumees of U_2, as these are the necessary ingredients of connection minimality (see Def. 4.1.9). This is doable using a special kind of *locality-based modules*, as presented in [GHKS08]. More specifically, we shrink the abduction problem $\langle \mathcal{T}, \Sigma, U_1 \sqsubseteq U_2 \rangle$ into $\langle \mathcal{M}^{\bot}_{U_1} \cup \mathcal{M}^{\top}_{U_2}, \Sigma, U_1 \sqsubseteq U_2 \rangle$, where $\mathcal{M}^{\bot}_{U_1}$ is the $\bot$-*module* of $\mathcal{T}$ for the signature of U_1, and $\mathcal{M}^{\top}_{U_2}$ is the $\top$-*module* of $\mathcal{T}$ for the signature of U_2 [GHKS08]. What is useful here is that $\mathcal{M}^{\bot}_{U_1}$ is a subset of $\mathcal{T}$ s.t. $\mathcal{M}^{\bot}_{U_1} \models U_1 \sqsubseteq V$ iff $\mathcal{T} \models U_1 \sqsubseteq V$ for all concepts V, while $\mathcal{M}^{\top}_{U_2}$ is a subset of $\mathcal{T}$ that ensures $\mathcal{M}^{\top}_{U_2} \models V \sqsubseteq U_2$ iff $\mathcal{T} \models V \sqsubseteq U_2$ for all concepts V.

Definition 4.3.3. A CI α is $\emptyset$-local (resp. Δ-local) for a signature Σ if every interpretation $\mathcal{I}$ s.t. $X^{\mathcal{I}} = \emptyset$ (resp. $X^{\mathcal{I}} = \Delta$) for all $X \in (\Omega_{\mathsf{C}} \cup \Omega_{\mathsf{R}}) \setminus \Sigma$ satisfies $\mathcal{I} \models \alpha$.

Definition 4.3.4. The $\emptyset$-*module* (resp. Δ-*module*) of $\mathcal{T}$ for Σ is the smallest subset $\mathcal{M} \subseteq \mathcal{T}$ s.t. every axiom in $\mathcal{T} \setminus \mathcal{M}$ is $\emptyset$-*local* (resp. Δ-local) for $\Sigma \cup \Sigma(\mathcal{M})$, where $\Sigma(\mathcal{M})$ denotes the restriction of the signature to the symbols occurring in the TBox $\mathcal{M}$.

This means the axioms outside of the $\emptyset$-module $\mathcal{M}$ are not useful to get non-trivial entailments using the signature $\Sigma \cup \Sigma(\mathcal{M})$.

A fast approximation of $\emptyset$-modules are $\bot$-modules [2]. It is sufficient to show the following: if $\mathcal{M}'$ is the $\bot$-module of $\mathcal{T}$ for Σ, and $\mathcal{M}$ a $\emptyset$-module of $\mathcal{T}$ for Σ, then $\mathcal{M} \subseteq \mathcal{M}'$. In the same way, $\top$-modules approximate Δ-modules.

Lemma 4.3.5 ($\bot$-module)**.** For a concept U_1 and a TBox $\mathcal{T}$, the $\bot$-module $\mathcal{M}$ of $\mathcal{T}$ for $\Sigma(U_1)$ satisfies $\mathcal{T} \models U_1 \sqsubseteq V$ iff $\mathcal{M} \models U_1 \sqsubseteq V$ for all concepts V.

Proof. (sketch) Thanks to the relation between the $\bot$-module and the $\emptyset$-module, it suffices to prove that the $\emptyset$-module $\mathcal{M}$' for $\Sigma(U_1)$ satisfies the property to obtain the same result for the $\bot$-module $\mathcal{M}$. We first observe that every non-tautological axiom $U \sqsubseteq U' \in \mathcal{T}$ s.t. $\Sigma(U) \subseteq \Sigma(\mathcal{M})$ occurs in $\mathcal{M}$. Otherwise, we would have $\Sigma(U') \not\subseteq (\mathcal{M})$, and $U'^{\mathcal{I}} = \emptyset$ for an interpretation $\emptyset$-local for $\Sigma(\mathcal{M})$, while $U^{\mathcal{I}} \neq \emptyset$, and thus $\mathcal{I} \not\models U \sqsubseteq V$. Moreover, in $\mathcal{EL}$, all subsumers of U_1 can be generated by unfolding, i.e., by iteratively replacing sub-concepts U in U_1 by concepts U' s.t. $U \sqsubseteq U' \in \mathcal{T}$. By using our first observation, we obtain that any axiom $U \sqsubseteq U' \in \mathcal{T}$ that could be involved by such an unfolding operation must be included in $\mathcal{M}$. It follows then that $\mathcal{M} \models U_1 \sqsubseteq V$ iff $\mathcal{T} \models U_1 \sqsubseteq V$ for all concepts V. □

Lemma 4.3.6 ($\top$-module)**.** For a concept U_2, the $\top$-module $\mathcal{M}$ of $\mathcal{T}$ for $\Sigma(U_2)$ satisfies $\mathcal{T} \models V \sqsubseteq U_2$ iff $\mathcal{M} \models V \sqsubseteq U_2$ for all concepts V.

Proof. (sketch) Can be shown in the same way as Lemma 4.3.5. □

Lemma 4.3.5 helps preserve the subsumers of U_1 while Lemma 4.3.6 helps preserve the subsmees of U_2. So, by also looking at the definition of connection-minimality (Def. 4.1.9), replacing $\mathcal{T}$ in the abduction problem by the union of the $\bot$-module for $\Sigma(U_1)$ and the $\top$-module for $\Sigma(U_2)$ still preserves the desired hypotheses despite that the problem size decreases.

4.4 Termination

The number of prime implicates can be infinite when our TBox contains cycles. For example, given $\mathcal{T} = \{U_1 \sqsubseteq P, P \sqsubseteq \exists R.P, \exists R.Q \sqsubseteq Q, Q \sqsubseteq U_2\}$, the number of the positive and negative ground prime implicates of Φ are infinite despite that the set of constructible hypotheses is finite.

$$\begin{aligned}
\mathcal{PI}^{g+}_{\Sigma}(\Phi) &= \{U_1(\mathtt{sk}_0), P(\mathtt{sk}_0), P(\mathtt{sk}(\mathtt{sk}_0)), P(\mathtt{sk}(\mathtt{sk}(\mathtt{sk}_0))), \ldots\}, \\
\mathcal{PI}^{g-}_{\Sigma}(\Phi) &= \{\neg U_2^-(\mathtt{sk}_0), \neg Q^-(\mathtt{sk}_0), \neg Q^-(\mathtt{sk}(\mathtt{sk}_0)), \ldots\}.
\end{aligned}$$

This means, naive generation of all prime implicates in Φ will not terminate. However, if we only consider the subset-minimal ones, termination can be obtained by bounding the term depth.

[2] Interested readers may consult [GHKS08] for the exact definition

The key result of this section is that we only need prime implicates with term depths of at most $n \times m$, where $n = |\Omega_\mathsf{C}|$ and $m = |\Omega_\mathsf{R}|$ for the given signature Ω_C and Ω_R.

The proof relies on the notion of *solution tree* to summarize inferences leading to a solution. This tree looks like a description tree, but with additional labeling functions. The first label simply assigns a Skolem term to each node (similar to the Skolem labeling for the canonical model). The second label (*positive label*) assigns a set of atomic concepts. The third one (*negative label*) assigns a single atomic concept. In the leaves, each node in someway represents matches in a constructible hypothesis. A positive label takes the atomic concepts in the positive prime implicates using that term. If we take any maximal antichain of its nodes then their negative labels correspond to the predicates in a derivable negative implicate while the ground terms are immediately taken from its Skolem label. Combining the positive and negative labels of these leaves, we get a constructible hypothesis, called the *solution* of the tree.

I show that, given a solution tree with solution $\mathcal{H}$, we can construct a solution tree with solution $\mathcal{H}' \subseteq \mathcal{H}$ s.t. any two nodes lying in one path do not have both the same topmost Skolem function in their Skolem labeling and the same negative label. Both of them are bounded by the number of Skolem functions n and the number of atomic concepts m. This means, the depth of the solution tree for any desired hypotheses that is also subset-minimal is bounded by $n \times m$. This is a rather loose bound. For the academia example, the bound is $22 \times 6 = 132$ which is much higher than needed.

4.4.1 Summarizing the Inferences of Matching Prime Implicates

The key idea to establish termination is to summarize inferences leading to a matching set of prime implicates using a *solution tree*, that looks like a description tree, but we give two additional labels that help identify intermediate ground implicates that are used to derive the positive prime implicates and the negative prime implicate with matching Skolem terms. A solution tree for a hypothesis $\mathcal{H}$ is defined as a tuple $(\mathfrak{S}, \mathfrak{l}^+, \mathfrak{l}^-)$, which is a tree-shaped labeled graph $\mathfrak{S} = (\mathcal{V}, \mathcal{E}, \mathfrak{s})$ together with two additional labeling functions $\mathfrak{l}^+$ and $\mathfrak{l}^-$ that maps the vertices to the matching prime implicates used to construct the hypothesis following Th. 4.2.11. So, the leaves $v_1, \ldots, v_n$ of a solution tree are such that

- $\mathfrak{l}^+(v_i) \in \mathcal{PI}^{g+}_\Sigma(\Phi)$ is not empty for all $i \in \{1, \ldots, n\}$, and
- $\neg\mathfrak{l}^-(\mathfrak{s}(v_1)) \vee \ldots \vee \neg\mathfrak{l}^-(\mathfrak{s}(v_n)) \in \mathcal{PI}^{g-}_\Sigma(\Phi)$ up to the repeated occurrence of some literals from different nodes.

The idea is to show how to reduce a solution tree to a smaller one when the associated constructible hypothesis is not subset-minimal.

The final result is Th. 4.4.15. For this, I incrementally introduce the required notions while at the same time present their relevant properties with the proofs. To begin with, I introduce the notion of Skolem tree, which serves as some sort of skeletal structure for the solution tree.

Definition 4.4.1 (Skolem Tree). A *Skolem tree* is a labeled tree $\mathfrak{S} = (\mathcal{V}, \mathcal{E}, \mathfrak{s})$ where $\mathfrak{s}$ assigns a Skolem term to every $v \in \mathcal{V}$ s.t. $\mathfrak{s}(v_0) = \mathtt{sk}_0$ for the root

$v_0 \in \mathcal{V}$, and for every $(v, v') \in \mathcal{E}$,[3] either $\mathfrak{s}(v) = \mathfrak{s}(v')$ or $\mathfrak{s}(v') = \mathtt{sk}(\mathfrak{s}(v))$ for some Skolem term $\mathtt{sk}$.

(Anti)chains for the nodes in the Skolem trees are in terms of the ancestor relationship: given two nodes $v, v' \in \mathcal{V}$, we call v an *ancestor* of v' iff there is a path from v to v' (i.e. every node is an ancestor of itself). A *chain* is a set of nodes that occur together on a path (where adjacent nodes have immediate ancestor relationship), and an *antichain* is a set of nodes where no node is an ancestor of another. *Maximal* chains/antichains are chains/antichains that are maximal w.r.t. the subset relation.

Labeling for $\mathcal{PI}^{g+}_{\Sigma}(\Phi)$ As a means to summarize which positive prime implicates use some term t, we simply label a node in some Skolem tree by the set consisting of every atomic concept P s.t. $P(t)$ is a prime implicate. The ancestor relationship would then represent the existence of possible derivations from a prime implicate while increasing the term depth.

Definition 4.4.2 (Positive Labeling)**.** Given a Skolem tree $\mathfrak{S} = (\mathcal{V}, \mathcal{E}, \mathfrak{s})$, the *positive labeling* for $\mathfrak{S}$ is defined as the function $\mathfrak{l}^+ : \mathcal{V} \rightarrow 2^{\Omega_C}$ s.t. for every $v \in \mathcal{V}$, $\mathfrak{l}^+(v) = \{P \mid P(\mathfrak{s}(v)) \in \mathcal{PI}^{g+}_{\Sigma}(\Phi)\}$.

Lemma 4.4.3 (Positive Labeling and $\mathcal{PI}^{g+}_{\Sigma}(\Phi)$)**.** Let $v_1, v_2 \in \mathcal{V}$ be such that v_1 is an ancestor of v_2, and $\mathfrak{s}(v_1)$ be of the form $\mathtt{sk}(t)$ for some Skolem function $\mathtt{sk}$. Then,

(i) $P_{\mathtt{sk}}(\mathfrak{s}(v_1)) \in \mathcal{PI}^{g+}_{\Sigma}(\Phi)$ and

(ii) for every $P \in \mathfrak{l}^+(v_2)$, there is a derivation of $P(\mathfrak{s}(v_2))$ from $P_{\mathtt{sk}}(\mathfrak{s}(v_1))$ and clauses only of the form

- **I3**:$\neg P_1(x) \vee P_2(x)$,
- **I4**:$\neg P_1(x) \vee \neg P_2(x) \vee P_3(x)$, and
- **I7**:$\neg P_1(x) \vee P_2(\mathtt{sk}(x))$

in Φ_p.

Proof. We first show that, (i) for every $P(\mathtt{sk}(t)) \in \mathcal{PI}^{g+}_{\Sigma}(\Phi)$, also $P_{\mathtt{sk}}(\mathtt{sk}(t)) \in \mathcal{PI}^{g+}_{\Sigma}(\Phi)$, and that $P(\mathtt{sk}(t))$ can be derived from $P_{\mathtt{sk}}(\mathtt{sk}(t))$ and **I3**:$\neg P_1(x) \vee P_2(x)$- and **I4**:$\neg P_1(x) \vee \neg P_2(x) \vee P_3(x)$-clauses of Φ_p. Afterward, we prove that, (ii) if $\mathfrak{l}^+(v_2)$ is not empty, then $\mathfrak{l}^+(v_1)$ is also not empty for any ancestor v_1 of v_2. The lemma then follows by induction.

(i) We have established in the proof of Th. 4.3.2 that $P(\mathtt{sk}(t))$ can be derived from Φ_p such that every resolvent of the derivation is ground. Moreover, by Lemma 4.3.1, **I5**:$\neg R(x, y) \vee \neg P_1(y) \vee P_2(x)$-clauses do not have to be involved in the derivation. **I6**:$\neg P_1(x) \vee R(x, \mathtt{sk}(x))$-clauses can then also be ignored because they introduce positive occurrences of binary literals and only **I5**:$\neg R(x, y) \vee \neg P_1(y) \vee P_2(x)$-clauses can be used to resolve upon them. We can also rule out the **I2**:$\neg U_2^-(\mathtt{sk}_0)$-clause $\neg U_2^-(\mathtt{sk}_0)$ because it is a duplicate, that can only derive negative duplicate clauses in Φ_p if

[3] When the role R labeling an edge vRw is irrelevant, we fall back to representing this edge as the pair of nodes (v, w).

$\mathbf{I5}$:$\neg R(x,y) \vee \neg P_1(y) \vee P_2(x)$- and $\mathbf{I6}$:$\neg P_1(x) \vee R(x, \mathtt{sk}(x))$-clauses are not used, because Φ_p is Horn. This leaves $U_1(\mathtt{sk}_0)$ and the $\mathbf{I3}$:$\neg P_1(x) \vee P_2(x)$-, $\mathbf{I4}$:$\neg P_1(x) \vee \neg P_2(x) \vee P_3(x)$- and $\mathbf{I7}$:$\neg P_1(x) \vee P_2(\mathtt{sk}(x))$-clauses as the ones that are used to derive $P(\mathtt{sk}(t))$ from Φ_p. Moreover, since Φ_p is Horn, linear resolution can be used to derive $P(\mathtt{sk}(t))$, thus inferences between two resolvents are not necessary [AB70]. Inferences in such a derivation can only preserve the Skolem term occurring in the ground premise when resolving with an $\mathbf{I3}$:$\neg P_1(x) \vee P_2(x)$- or an $\mathbf{I4}$:$\neg P_1(x) \vee \neg P_2(x) \vee P_3(x)$-clause, and increase the depth of the term in the resolvent when resolving with an $\mathbf{I7}$:$\neg P_1(x) \vee P_2(\mathtt{sk}(x))$-clause. Thus $P(\mathtt{sk}(t))$ can only be derived after $P_{\mathtt{sk}}(\mathtt{sk}(t))$ has been introduced by the $\mathbf{I7}$:$\neg P_1(x) \vee P_2(\mathtt{sk}(x))$-clause introducing $\mathtt{sk}$, and from $P_{\mathtt{sk}}(\mathtt{sk}(t))$ only $\mathbf{I3}$:$\neg P_1(x) \vee P_2(x)$- or $\mathbf{I4}$:$\neg P_1(x) \vee \neg P_2(x) \vee P_3(x)$-clauses can be used to derive $P(\mathtt{sk}(t))$. Let us consider the (linear) derivation of $P(\mathtt{sk}(t))$ and remove from it all the inferences on $\mathbf{I3}$:$\neg P_1(x) \vee P_2(x)$- and $\mathbf{I4}$:$\neg P_1(x) \vee \neg P_2(x) \vee P_3(x)$-clauses upon literals where $\mathtt{sk}(t)$ occurs. The only literals that remain in the derived clause are copies of $P_{\mathtt{sk}}(\mathtt{sk}(t))$ because it is the only literal with the term $\mathtt{sk}(t)$ that can be derived, and because all other literals are resolved upon in parts of the derivation that have not been removed. Thus, it is enough to append a few factorization inferences at the end of this derivation, if at all needed, to derive $P_{\mathtt{sk}}(\mathtt{sk}(t))$. Hence $P_{\mathtt{sk}}(\mathtt{sk}(t)) \in \mathcal{PI}^{g+}_{\Sigma}(\Phi)$. The inferences from the removed parts of the derivation of $P(\mathtt{sk}(t))$, introducing only $\mathbf{I3}$:$\neg P_1(x) \vee P_2(x)$- and $\mathbf{I4}$:$\neg P_1(x) \vee \neg P_2(x) \vee P_3(x)$-clauses can be used together with $P_{\mathtt{sk}}(\mathtt{sk}(t))$ to construct a derivation of $P(\mathtt{sk}(t))$.

(ii) To prove that the ancestors of v_2 have a non-empty positive label if v_2 has a non-empty positive label, let us consider the case when v_1 is the direct parent of v_2 (the case when $v_1 = v_2$ is trivial). Let us consider once again the part of the linear derivation of $P(\mathtt{sk}(t))$ from which we created the derivation of $P_{\mathtt{sk}}(\mathtt{sk}(t))$. If we remove from it the inference(s) introducing $\mathtt{sk}$, the clause that is derived must contain only $P'(t)$ literals, where $\neg P'(t) \vee P_{\mathtt{sk}}(\mathtt{sk}(t))$ is the $\mathbf{I7}$:$\neg P_1(x) \vee P_2(\mathtt{sk}(x))$ clause that introduced $\mathtt{sk}$. These $P'(t)$ literals can be factorized following this derivation to obtain a derivation of $P'(t)$ from Φ_p. Thus v_1 has a non-empty positive label and this result also holds for any ancestor of v_2 by induction.

Finally, we have proven that for v_1 s.t. $\mathfrak{s}(v_1) = \mathtt{sk}'(t')$, there is a derivation of any $P'(\mathtt{sk}'(t')) \in \mathfrak{l}^+(v_1)$ from $P_{\mathtt{sk}'}(\mathtt{sk}'(t'))$ and the $\mathbf{I3}$:$\neg P_1(x) \vee P_2(x)$- and $\mathbf{I4}$:$\neg P_1(x) \vee \neg P_2(x) \vee P_3(x)$-clauses in Φ_p. Moreover, in the previous paragraph, assuming v_1 is the parent of v_2, we have seen that there is at least some $P'(\mathtt{sk}'(t')) \in \mathfrak{l}^+(v_1)$ from which $P_{\mathtt{sk}}(\mathtt{sk}(\mathtt{sk}'(t')))$ can be inferred by using the $\mathbf{I7}$:$\neg P_1(x) \vee P_2(\mathtt{sk}(x))$-clause introducing $\mathtt{sk}$ where $\mathtt{sk}(\mathtt{sk}'(t')) = \mathfrak{s}(v_2)$. Thus $P(\mathfrak{s}(v_2))$ can be derived from $P_{\mathtt{sk}'}(\mathfrak{s}(v_1))$ and the $\mathbf{I3}$:$\neg P_1(x) \vee P_2(x)$-, $\mathbf{I4}$:$\neg P_1(x) \vee \neg P_2(x) \vee P_3(x)$- and $\mathbf{I7}$:$\neg P_1(x) \vee P_2(\mathtt{sk}(x))$-clauses in Φ_p, and this result can be extended to any ancestor of v_2 (except the root v_0, for which $U_1(\mathtt{sk}_0)$ could be used instead of $P_{\mathtt{sk}'}(\mathfrak{s}(v_0))$, that does not exist).

□

Apparently, if two nodes lie on the same path where both have the same topmost Skolem function for their Skolem labeling, they will also have similar positive labelings.

Lemma 4.4.4 (Vertices having similar labels)**.** For any two nodes $v_1, v_2 \in \mathcal{V}$ s.t. $\mathfrak{s}(v_1) = \mathtt{sk}(t_1)$ and $\mathfrak{s}(v_2) = \mathtt{sk}(t_2)$ for the same Skolem function $\mathtt{sk}$ and some terms t_1 and t_2, if $\mathfrak{l}^+(v_1)$ and $\mathfrak{l}^+(v_2)$ are not empty, then $\mathfrak{l}^+(v_1) = \mathfrak{l}^+(v_2)$.

Proof. By Lemma 4.4.3, if $\mathfrak{l}^+(v_1)$ and $\mathfrak{l}^+(v_2)$ are not empty, then $P_{\mathtt{sk}}(\mathfrak{s}(v_1)) \in \mathfrak{l}^+(v_1)$ and $P_{\mathtt{sk}}(\mathfrak{s}(v_2)) \in \mathfrak{l}^+(v_2)$. Moreover, for every $P \in \mathfrak{l}^+(v_1)$, $P(\mathfrak{s}(v_1))$ can be derived using only $P_{\mathtt{sk}}(\mathfrak{s}(v_1))$ and **I3**:$\neg P_1(x) \vee P_2(x)$- and **I4**:$\neg P_1(x) \vee \neg P_2(x) \vee P_3(x)$-clauses in Φ_p. The **I7**:$\neg P_1(x) \vee P_2(\mathtt{sk}(x))$-clauses do not intervene because they can only increase the depth of terms.

Any such derivation can be mirrored from $P_{\mathtt{sk}}(\mathfrak{s}(v_2))$ and the **I3**:$\neg P_1(x) \vee P_2(x)$- and **I4**:$\neg P_1(x) \vee \neg P_2(x) \vee P_3(x)$-clauses in Φ_p to derive $P(\mathfrak{s}(v_2))$, thus $\mathfrak{l}^+(v_1) \subseteq \mathfrak{l}^+(v_2)$. The reverse inclusion $\mathfrak{l}^+(v_2) \subseteq \mathfrak{l}^+(v_1)$ is proved similarly. ☐

Lemma 4.4.5. If $R(t, \mathtt{sk}(t)) \in \mathcal{PI}^{g+}_{\Sigma}(\Phi)$ and $\mathtt{sk}(t)$ has a subterm of the form $\mathtt{sk}(t')$, then $R(t', \mathtt{sk}(t')) \in \mathcal{PI}^{g+}_{\Sigma}(\Phi)$.

Proof. If $R(t, \mathtt{sk}(t)) \in \mathcal{PI}^{g+}_{\Sigma}(\Phi)$, then R occurs in the **I6**:$\neg P_1(x) \vee R(x, \mathtt{sk}(x))$-clause introducing $\mathtt{sk}$, thus any (linear) derivation of $R(t, \mathtt{sk}(t))$ can be transformed into a derivation of $P_{\mathtt{sk}}(\mathtt{sk}(t))$ by replacing this **I6**:$\neg P_1(x) \vee R(x, \mathtt{sk}(x))$-clause with the **I7**:$\neg P_1(x) \vee P_2(\mathtt{sk}(x))$-clause introducing $\mathtt{sk}$ when resolving upon $P_1(t)$ (remember it is possible to ensure that the other premise is ground). Thus $P_{\mathtt{sk}}(\mathtt{sk}(t)) \in \mathcal{PI}^{g+}_{\Sigma}(\Phi)$.

In a Skolem tree with a node v_2 s.t. $\mathfrak{s}(v_2) = \mathtt{sk}(t)$ for the t and $\mathtt{sk}$ from the previous paragraph, there must be an ancestor v_1 of v_2 s.t. $\mathfrak{s}(v_1) = \mathtt{sk}(t')$ since it is a subterm of $\mathtt{sk}(t)$. Thus by Lemma 4.4.4, $P_{\mathtt{sk}}(\mathtt{sk}(t')) \in \mathcal{PI}^{g+}_{\Sigma}(\Phi)$. Moreover, a (linear) derivation of $P_{\mathtt{sk}}(\mathtt{sk}(t'))$ from Φ_p can be turned into a derivation of $R(t', \mathtt{sk}(t'))$ by applying the reverse transformation as in the previous paragraph, thus $R(t', \mathtt{sk}(t')) \in \mathcal{PI}^{g+}_{\Sigma}(\Phi)$. ☐

Labeling for $\mathcal{PI}^{g-}_{\Sigma}(\Phi)$ For the negative prime implicates, we also introduce negative labeling. The idea is almost similar to the positive one except that we need an antichain to represent a prime implicate instead of a single node representing multiple prime implicates.

Definition 4.4.6 (Negative Labeling)**.** Given a Skolem tree $\mathfrak{S} = (\mathcal{V}, \mathcal{E}, \mathfrak{s})$, a function $\mathfrak{l}^- : \mathcal{V} \to \{P^- \mid P \in \Omega_{\mathsf{C}}\}$ is called *negative labeling for* $\mathfrak{S}$ iff:

– (*root*) $\mathfrak{l}^-(v_0) = U_2$ if v_0 is the root of $\mathfrak{S}$, and

– (*non-leaf*) for all non leaf $v \in \mathcal{V}$ with children $v_1, \ldots, v_n$, there is a derivation[4] of $\neg Q_1^-(\mathfrak{s}(v_1)) \vee \ldots \vee \neg Q_n^-(\mathfrak{s}(v_n))$ for $Q_i = \mathfrak{l}^-(v_i)$ using

  - $\neg Q^-(\mathfrak{s}(v))$ for $Q = \mathfrak{l}^-(v)$,
  - $\{R(t, \mathtt{sk}(t)) \in \mathcal{PI}^{g+}_{\Sigma}(\Phi)\}$, and
  - the non-ground clauses in Φ_p.

[4]Factoring inference is not captured in the derivability of $\neg Q_1^-(\mathfrak{s}(v_1)) \vee \ldots \vee \neg Q_n^-(\mathfrak{s}(v_n))$ in (*non-leaf*). This means, the same literal in it may come from more than one leaf.

The clause $\neg Q_1^-(\mathtt{sk}(v_1)) \vee \ldots \vee \neg Q_n^-(\mathtt{sk}(v_n))$ is denoted as $\varphi_{\mathfrak{S},\mathfrak{l}^-}$, if the set $\{v_1, \ldots, v_n\}$ consists of all leaves of $\mathfrak{S}$.

Note that any such clause $\neg Q_1^-(t_1) \vee \ldots \vee \neg Q_m^-(t_m)$ is an implicate, but not necessarily a prime implicate.

Lemma 4.4.7. Given a Skolem tree $\mathfrak{S} = (\mathcal{V}, \mathcal{E}, \mathfrak{s})$, if $\mathfrak{S}$ has a negative labeling $\mathfrak{l}^-$, and the set $\{v_1, \ldots, v_m\} \subseteq \mathcal{V}$ is a maximal antichain, then

$$\neg Q_1^-(t_1) \vee \ldots \vee \neg Q_m^-(t_m)$$

is derivable from Φ_p.

Proof. (sketch) This follows directly by induction using (*root*) as the base case and the derivability in the case (*non-leaf*) as the induction step □

What we are eventually interested in from Lemma 4.4.7 is that this holds in particular for $\varphi_{\mathfrak{S},\mathfrak{l}^-}$. Conversely, for any negative ground implicate of Φ_p of the form $\neg Q_1^-(t_1) \vee \ldots \vee \neg Q_n^-(t_n)$, we can construct a Skolem tree and a negative labeling s.t. $\varphi_{\mathfrak{S},\mathfrak{l}^-}$ is the negative ground implicate, up to the repetition of literals. This Skolem tree can be constructed together with the negative labeling $\mathfrak{l}^-$ by following the derivation from $\neg U_2^-(\mathtt{sk}_0)$ to $\neg Q_1^-(t_1) \vee \ldots \vee \neg Q_n^-(t_n)$ in Φ_p. Specifically, we note that:

- Such a derivation must exist, because Φ_p is Horn and $\neg U_2^-(\mathtt{sk}_0)$ is the only negative clause and thus every derivation of a negative ground clause must use this clause.
- Moreover, starting from $\Phi_p \cup \mathcal{PI}^{g+}_\Sigma(\Phi)$ all resolvents in a derivation of a negative ground clause can be negative clauses and only $\{R(t, \mathtt{sk}(t)) \in \mathcal{PI}^{g+}_\Sigma(\Phi) \mid \mathtt{sk} \in \sqcap_\mathsf{S}, t \in \mathsf{T}_{\mathtt{sk}_0}(\sqcap_\mathsf{S})\}$ is needed because the other positive prime implicates contain original literals that are not needed to derive a clause with only duplicate literals. Such derivations can be linear.
- Following the argument used in the proof for Th. 4.3.2, we can rearrange any linear derivation of a ground negative clause so that variables and binary literals are eliminated as soon as they are introduced. This step is done by resolving first a ground clause φ and an **I5**:$\neg R(x, y) \vee \neg P_1(y) \vee P_2(x)$-clause $\neg R(x, y) \vee \neg P_1(y) \vee P_2(x)$, and then resolving an **I6**:$\neg P_1(x) \vee R(x, \mathtt{sk}(x))$-clause $\neg P_1'(x') \vee R(x', \mathtt{sk}(x'))$ and the resolvent of the previous step. These two steps amount to replacing the literal $\neg P_2(t)$ in φ by $\neg P_1'(t) \vee \neg P_1(\mathtt{sk}(t))$.

We then get the following lemma dealing with derivability of the negative prime implicates.

Lemma 4.4.8 (Negative Labeling and $\mathcal{PI}^{g-}_\Sigma(\Phi)$)**.** For any $\mathcal{B}$ s.t. $\bigvee_{Q(t)\in\mathcal{B}} \neg Q^-(t) \in \mathcal{PI}^{g-}_\Sigma(\Phi)$, we can construct a Skolem tree $\mathfrak{S}$ that has a negative labeling $\mathfrak{l}^-$ s.t. for every $Q(t) \in \mathcal{B}$, $\mathfrak{s}(v) = t$ and $\mathfrak{l}^-(v) = Q$ for a leaf v of $\mathfrak{S}$, and for all leaves v in $\mathfrak{S}$, $\mathfrak{l}^-(v)(\mathfrak{s}(v)) \in \mathcal{B}$.

By decorating the Skolem tree with the positive and negative labels, we can now define the solution tree. By taking the labels on the leaves of a solution tree, we obtain a connection-minimal hypothesis after Th. 4.2.11. In the CI's of the solution, the positive label provides the left-hand sides while the negative label gives the right-hand sides.

Definition 4.4.9 (Solution Tree)**.** A *solution tree* is a tuple $(\mathfrak{S}, \mathfrak{l}^+, \mathfrak{l}^-)$ where

- $\mathfrak{S} = (\mathcal{V}, \mathcal{E}, \mathfrak{s})$ is a Skolem tree for the terms from which some constructible hypothesis $\mathcal{H}$ is defined,
- $\mathfrak{l}^+$ is a positive labeling for $\mathfrak{S}$ such that all nodes in $\mathfrak{S}$ have a non-empty positive label, and
- $\mathfrak{l}^-$ is a negative labeling for $\mathfrak{S}$ s.t. $\varphi_{\mathfrak{S},\mathfrak{l}^-} \in \mathcal{PI}^{g-}_{\Sigma}(\Phi)$ modulo the repetition of literals.

The *solution* of the tree is a TBox equivalent to $\mathcal{H}$, which is the set $\{\bigsqcap \mathfrak{l}^+(v) \sqsubseteq \mathfrak{l}^-(v) \mid v \text{ is a leaf of } \mathfrak{S}\}$.

To simplify the later proof, w.l.o.g. any CI in the solution of the tree has only atomic concepts on its right-hand side, while the equivalent $\mathcal{H}$ may use a conjunction. This is because a CI with a conjunction on the right-hand side can be split anyway. In addition, tautologies from this solution can be removed to get a packed connection-minimal hypothesis.

4.4.2 Bounding the Skolem Term Depth

We are now ready to see how to obtain the termination result. I will show that if two vertices in a solution tree have the same labelings while having ancestor relationship, then the subtree rooted at the ancestor can be replaced by the subtree rooted at the other vertex with an additional adaptation of the Skolem labeling. This would produce a smaller solution tree (in terms of the number of nodes). Note that we actually have a weaker condition here: if a solution tree is minimal, then one cannot perform such replacement anymore. As we shall see later, the reverse is not true. Nevertheless, this is already sufficient to acquire termination at least for connection-minimal hypotheses that are also subset-minimal.

Lemma 4.4.10 (Subtree Replacement)**.** Let $(\mathfrak{S}, \mathfrak{l}^+, \mathfrak{l}^-)$ be a solution tree, where $\mathfrak{S} = (\mathcal{V}, \mathcal{E}, \mathfrak{s})$ and $v_1, v_2 \in \mathcal{V}$ be such that v_1 is an ancestor of v_2, $\mathfrak{s}(v_1) = \mathtt{sk}(t)$ and $\mathfrak{s}(v_2) = \mathtt{sk}(t')$ for some $\mathtt{sk}$, t and t', and $\mathfrak{l}^-(v_1) = \mathfrak{l}^-(v_2)$. Let $\mathfrak{S}' = (\mathcal{V}', \mathcal{E}', \mathfrak{s}')$ the result of replacing in $\mathfrak{S}$ the subtree under v_1 by the subtree under v_2, adapting the Skolem labeling $\mathfrak{s}$ to $\mathfrak{s}'$ appropriately, and let $\mathfrak{l}^{+\prime}$ and $\mathfrak{l}^{-\prime}$ be $\mathfrak{l}^+$ and $\mathfrak{l}^-$ restricted to $\mathcal{V}'$. Then, $(\mathfrak{S}', \mathfrak{l}^{+\prime}, \mathfrak{l}^{-\prime})$ is also a solution tree.

Proof. We have to show that all primed labelings are valid labelings, that the positive one is not empty for any node in $\mathfrak{S}$' and that $\varphi_{\mathfrak{S}',\mathfrak{l}^{-\prime}} \in \mathcal{PI}^{g-}_{\Sigma}(\Phi)$ modulo the repetition of literals.

The adaptation of $\mathfrak{s}$ to create $\mathfrak{s}'$ consists of replacing the subterm $\mathtt{sk}(t')$, in every term $\mathfrak{s}(v)$ for v descending from v_2 in $\mathfrak{S}$ by $\mathtt{sk}(t)$ in $\mathfrak{S}$'. That way, $\mathfrak{s}'$ is also a Skolem labeling. By Def. 4.4.2 and Lemma 4.4.4, $\mathfrak{l}^{+\prime}$ is a positive labeling for $\mathfrak{S}$' and none of its labels are empty because none of the labels of $\mathfrak{l}^+$ are empty. By Def. 4.4.6, all properties needed to ensure $\mathfrak{l}^{-\prime}$ is a negative labeling for $\mathfrak{S}$' are trivially verified for the nodes outside of the descendants of v_1 because they are the same as in $\mathfrak{S}$.

Let $w_1, \ldots, w_n$ be the children of v_1 in $\mathfrak{S}$'. We show that $\varphi = \neg Q_1^-(\mathfrak{s}(w_1)) \vee \ldots \vee \neg Q_n^-(\mathfrak{s}(w_n))$ where $Q_i = \mathfrak{l}^-(w_i)$ for $i \in \{1, \ldots, n\}$ can be derived from the non-ground clauses in Φ_p, the set $\{R(t', \mathtt{sk}(t')) \in \mathcal{PI}^{g+}_{\Sigma}(\Phi) \mid \mathtt{sk} \in \Pi_{\mathsf{S}}, t' \in \mathsf{T}_{\mathtt{sk}_0}(\Pi_{\mathsf{S}})\}$ and $\neg Q^-(\mathfrak{s}(v_1))$, where $Q = \mathfrak{l}^-(v_1)$. In $\mathfrak{S}$, the w_i nodes are the children of v_2, thus $\varphi' = \neg Q^-_1(\mathfrak{s}(v_2)) \vee \ldots \vee \neg Q^-_n(\mathfrak{s}(v_2))$ can be derived from the non-ground clauses in Φ_p, the set $\{R(t', \mathtt{sk}(t')) \in \mathcal{PI}^{g+}_{\Sigma}(\Phi) \mid \mathtt{sk} \in \Pi_{\mathsf{S}}, t' \in \mathsf{T}_{\mathtt{sk}_0}(\Pi_{\mathsf{S}})\}$ and $\neg Q^-(\mathfrak{s}(v_2))$. We can write $\mathfrak{s}(v_2)$ as $g(\mathfrak{s}(v_1))$, where g is a composition of Skolem functions, and every $\mathfrak{s}(w_i)$ can be written as $\mathtt{sk}_i(g(\mathfrak{s}(v_1)))$ for some $\mathtt{sk}_i \in \Pi_{\mathsf{S}}$. The derivation of φ' can thus be transformed into a derivation of φ by replacing $Q(t')$ by $Q(t)$ everywhere it is used, and replacing any $R(\mathfrak{s}(v_2), \mathtt{sk}_i(\mathfrak{s}(v_2))) \in \mathcal{PI}^{g+}_{\Sigma}(\Phi)$ used by $R(\mathfrak{s}(v_1), \mathtt{sk}_i(\mathfrak{s}(v_1)))$ because the latter also belongs to $\mathcal{PI}^{g+}_{\Sigma}(\Phi)$ by Lemma 4.4.5. The same argument applies to any other descendant v' of v_1 in $\mathfrak{S}$' so that the second point of Def. 4.4.6 holds for $\mathfrak{l}^{-\prime}$. Thus $\mathfrak{l}^{-\prime}$ is a negative labeling for $\mathfrak{S}$'.

If $\mathfrak{l}^{-\prime}$ is such that $\varphi_{\mathfrak{S}',\mathfrak{l}^{-\prime}}$ is not in $\mathcal{PI}^{g-}_{\Sigma}(\Phi)$ modulo the repetition of literals, then there exists a solution tree $\mathfrak{S}$" for a strict subclause of $\varphi_{\mathfrak{S}',\mathfrak{l}^{-\prime}}$ modulo the repetition of literals, i.e., where one literal of $\varphi_{\mathfrak{S}',\mathfrak{l}^{-\prime}}$ does not appear at all, that is an implicate of Φ_p and it is possible to apply the transformation from $\mathfrak{S}$ to $\mathfrak{S}$' backward from $\mathfrak{S}$". This would produce a solution forest for a strict subclause of $\varphi_{\mathfrak{S},\mathfrak{l}^-}$ modulo the repetition of literals derivable from Φ_p by Lemma 4.4.7, which is impossible since $\varphi_{\mathfrak{S},\mathfrak{l}^-} \in \mathcal{PI}^{g-}_{\Sigma}(\Phi)$ modulo the repetition of literals. Thus $\varphi_{\mathfrak{S}',\mathfrak{l}^{-\prime}} \in \mathcal{PI}^{g-}_{\Sigma}(\Phi)$ modulo the repetition of literals. □

Definition 4.4.11 (Minimal Solution Tree)**.** A solution tree $\mathfrak{S}$ for a hypothesis $\mathcal{H}$ is *minimal* if there exists no solution tree $\mathfrak{S}'$ for a hypothesis $\mathcal{H}' \subseteq \mathcal{H}$ s.t. $\mathfrak{S}'$ uses strictly less nodes.

The following lemma characterizes equal labelings by the equality of the topmost Skolem functions. This would then one of the parameter used for the bound.

Lemma 4.4.12. Let $(\mathfrak{S}, \mathfrak{l}^+, \mathfrak{l}^-)$ be a minimal solution tree where $\mathfrak{S} = (\mathcal{V}, \mathcal{E}, \mathfrak{s})$. Let $v, v' \in \mathcal{V}$ be nodes such that v' is an ancestor of v. Then, either $\mathfrak{s}(v)$ and $\mathfrak{s}(v')$ are not headed by the same Skolem term or $\mathfrak{l}^-(v) \neq \mathfrak{l}^-(v')$.

Proof. Let $(\mathfrak{S}, \mathfrak{l}^+, \mathfrak{l}^-)$ be a minimal solution tree for the hypothesis $\mathcal{H}$ where $\mathfrak{S} = (\mathcal{V}, \mathcal{E}, \mathfrak{s})$ such that v' is an ancestor of v, $\mathfrak{l}^-(v) = \mathfrak{l}^-(v')$ and $\mathfrak{s}(v) = \mathtt{sk}(t)$ and $\mathfrak{s}(v') = \mathtt{sk}(t')$ for some $\mathtt{sk}$, t and t'. By applying Lemma 4.4.10, we obtain a solution tree $\mathfrak{S}'$ with less nodes than $\mathfrak{S}$ and for a solution $\mathcal{H}'$ s.t. $\mathcal{H}' \subseteq \mathcal{H}$. Consequently, $\mathfrak{S}$ cannot be minimal. □

Example 4.4.13 (Solution tree)**.** Consider the TBox $\mathcal{T}$ from Ex. 4.1.11

$$\begin{aligned}\mathcal{T} = \{&(1) : \mathsf{Bat} \sqsubseteq \exists \mathsf{canBite}.\mathsf{Bat},\\ &(2) : \mathsf{Bat} \sqsubseteq \exists \mathsf{canBite}.\mathsf{Human},\\ &(3) : \mathsf{Bat} \sqsubseteq \exists \mathsf{canHost}.\mathsf{Rabies},\\ &(4) : \mathsf{DiseaseVector} \sqsubseteq \mathsf{Organism},\\ &(5) : \mathsf{Human} \sqsubseteq \mathsf{Organism}\}\\ &(6) : \exists \mathsf{canBite}.\mathsf{Organism} \sqcap \exists \mathsf{canHost}.\mathsf{Virus} \sqsubseteq \mathsf{DiseaseVector}\},\end{aligned}$$

where we have an observation $\mathsf{Bat} \sqsubseteq \mathsf{DiseaseVector}$ with $\mathcal{T} \not\models \mathsf{Bat} \sqsubseteq \mathsf{DiseaseVector}$. We consider the following hypotheses to illustrate the termination bound presented before.

$$\mathcal{H}_4 = \{\mathsf{Rabies} \sqsubseteq \mathsf{Virus}\}$$
$$\mathcal{H}_5 = \{\mathsf{Human} \sqsubseteq \mathsf{DiseaseVector}, \mathsf{Rabies} \sqsubseteq \mathsf{Virus}\}$$

Suppose that $\mathtt{sk}_1$,$\mathtt{sk}_2$, and $\mathtt{sk}_3$ comes out of the Skolemization of (1), (2), and (3) respectively in its first order translation $\Phi = \mathtt{FO}(\mathcal{T}, \mathsf{Bat} \sqsubseteq \mathsf{DiseaseVector})$. Fig. 4.4 shows one possible solution tree $(\mathfrak{S}, \mathfrak{l}^+, \mathfrak{l}^-)$ where, in the picture, each node v_i is described by $v_i : \mathfrak{l}^+(v_i), \mathfrak{s}(v_i), \mathfrak{l}^-(v_i)$[5] with $\mathfrak{s}$ labeling $\mathfrak{S}$ with the necessary Skolem terms.

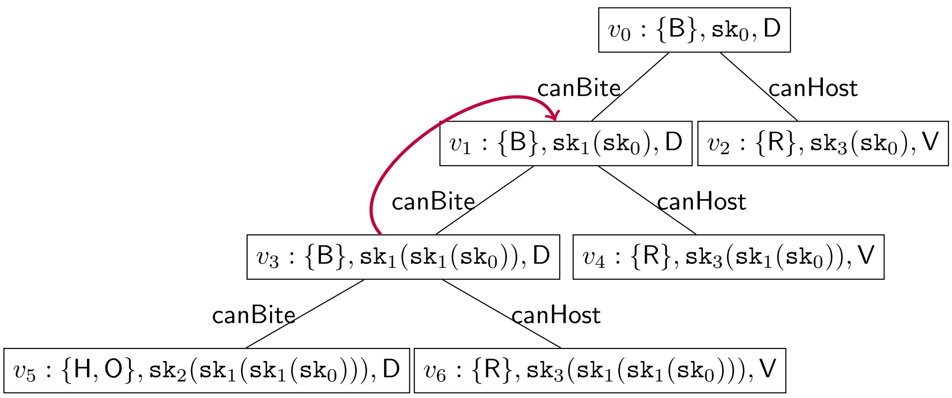

Figure 4.4: Solution tree $(\mathfrak{S}, \mathfrak{l}^+, \mathfrak{l}^-)$ for $\mathcal{H}_5$

Due to Lemma 4.4.12, the solution tree from Fig. 4.4 is not minimal:

- $\mathfrak{s}(v_1)$ and $\mathfrak{s}(v_3)$ share the same topmost skolem function $\mathtt{sk}_1$, and
- $\mathfrak{l}^-(v_1) = \mathfrak{l}^-(v_3) = \mathsf{DiseaseVector}$.

The arrow shows a subtree replacement described by Lemma 4.4.10. The subtree rooted at v_3 replaces the subtree rooted at v_1 but the labeling of the Skolem term is also adapted according to the lemma (more specifically inside the proof). In this case, in every node of the subtree rooted at v_3, all occurences of $\mathtt{sk}_1(\mathtt{sk}_1(\mathtt{sk}_0))$ (taken from $\mathfrak{s}(v_3)$) are replaced by $\mathtt{sk}_1(\mathtt{sk}_0)$ (taken from $\mathfrak{s}(v_1)$). This will result in the solution tree $(\mathfrak{S}', \mathfrak{l}^{+\prime}, \mathfrak{l}^{-\prime})$ in Fig. 4.5. From both trees, taking their solutions (see Def. 4.4.9) would give us the same hypothesis $\mathcal{H}_5 = \{\mathsf{Human} \sqsubseteq \mathsf{DiseaseVector}, \mathsf{Rabies} \sqsubseteq \mathsf{Virus}\}$[6].

Note that, according Def. 4.4.11, $(\mathfrak{S}', \mathfrak{l}^{+\prime}, \mathfrak{l}^{-\prime})$ is not minimal because of the solution tree in Fig. 4.6:

- $\mathcal{H}_4 \subsetneq \mathcal{H}_5$, and
- $\mathcal{H}_4$ is the solution of $(\mathfrak{S}'', \mathfrak{l}^{+\prime\prime}, \mathfrak{l}^{-\prime\prime})$ with less nodes.

[5]For readability, the concept names are shown using only the respective first letters.

[6]The solution of the tree gives us a constructible hypothesis but I consider only the concept Human from $\mathfrak{l}^+(v_5) = \{\mathsf{Human}, \mathsf{Organism}\}$ for illustration.

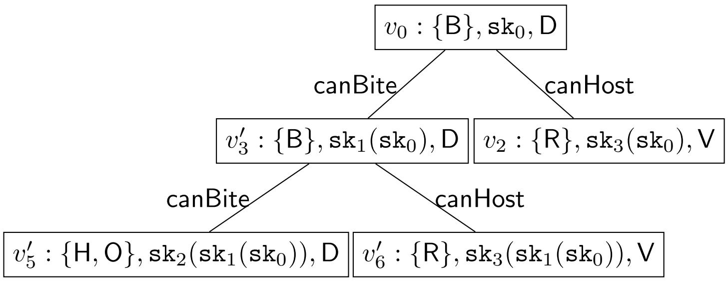

Figure 4.5: Solution tree $(\mathfrak{S}', \mathfrak{l}^{+\prime}, \mathfrak{l}^{-\prime})$ for $\mathcal{H}_5$ with less nodes

Moreover, Lemma 4.4.12 cannot be used to distinguish them.

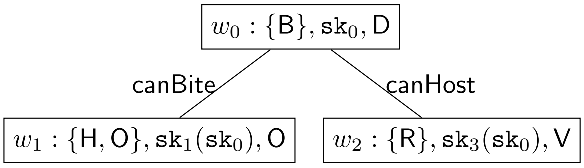

Figure 4.6: Solution tree $(\mathfrak{S}'', \mathfrak{l}^{+\prime\prime}, \mathfrak{l}^{-\prime\prime})$ for $\mathcal{H}_4$

However, Lemma 4.4.12 is already sufficient to derive a termination condition. The following corollary then relates solution trees that are minimal with their depths.

Corollary 4.4.14. Let $= (\mathfrak{S}, \mathfrak{l}^+, \mathfrak{l}^-)$ be a minimal solution tree. Then, the depth of $\mathfrak{S}$ is bounded by $n \times m$, where n is the number of Skolem functions in Φ introduced for the transformation of $\mathcal{T}$, and m is the number of atomic concepts in Φ.

Proof. Let $\mathfrak{S}$ be a minimal solution tree. By Lemma 4.4.12, on every path in $\mathfrak{S}$, there are no two nodes v, v' such that $\mathfrak{l}^-(v) = \mathfrak{l}^-(v')$, $\mathfrak{s}(v) = \mathtt{sk}(t)$ and $\mathfrak{s}(v') = \mathtt{sk}(t')$ for some $\mathtt{sk}$, t and t'. Thus in a path, for every Skolem function $\mathtt{sk}$, there can be at most m nodes on a path, each with a different negative label. However, the Skolem functions introduced during the translation of $\mathcal{T}^-$ are never needed since they do not occur in $\mathcal{PI}^{g+}_{\Sigma}(\Phi)$. The range of $\mathfrak{l}^-$ is bounded by the number of atomic concepts in Φ, that we denote m. We additionally denote by n the number of Skolem functions in Φ introduced by the translation of $\mathcal{T}$, and thus, every path in $\mathfrak{S}$ can have a length of at most $n \times m$. □

The following theorem is a direct consequence of Corollary 4.4.14, because for a subset-minimal constructible hypothesis $\mathcal{H}$, there is no constructible hypothesis $\mathcal{H}'$ s.t. $\mathcal{H}' \subsetneq \mathcal{H}$.

Theorem 4.4.15 (Term Depth Bound)**.** Given an abduction problem and its translation Φ, every subset-minimal constructible hypothesis can be built from prime implicates having a nesting depth of at most $n \times m$, where n is the number of atomic concepts in Φ, and m is the number of occurrences of existential role restrictions in $\mathcal{T}$.

4.5 Implementation and Experiments

For evaluation, we developed an implementation using SPASS as a key reasoning engine. Both of the previous sections regarding efficiency and termination are implemented accordingly. We then perform experiments to collect some basic information regarding the number of problems, successful generations, etc.

4.5.1 Implementation

We developed a prototype that computes all subset-minimal constructible hypotheses, following the technique described in the previous section. To compute the prime implicates, we used SPASS [WSH$^+$07], a first-order theorem prover that includes resolution among other calculi. Everything before and after the prime implicate are taken care of in java. These include parsing of the ontologies, preprocessing (detailed below), clausification of the abduction problems, translation to SPASS input, and parsing and processing of the SPASS output where constructible hypotheses are reconstructed and filtered-out from the non-subset-minimal ones. The Java part uses the OWL API for all of the general DL-related needs [HB11] while using the $\mathcal{EL}$ reasoner ELK specifically for the presaturation [KKS14].

Preprocessing. In this first stage, the removal of axioms not relevant to the abduction problems and presaturation are performed. First, we can remove some unrelated axioms supported by Lemma 4.3.5 and 4.3.6. Here, every connection-minimal hypothesis for $(\mathcal{T}, \Sigma, U_1 \sqsubseteq U_2)$ is also a connection-minimal hypothesis for $(\mathcal{M}^{\bot}_{U_1} \cup \mathcal{M}^{\top}_{U_2}, \Sigma, U_1 \sqsubseteq U_2)$. Presaturation is useful for termination as described in Sect. 4.4: ELK helps compute all concept inclusions of the form $P \sqsubseteq Q$ s.t. $\mathcal{M}^{\bot}_{U_1} \cup \mathcal{M}^{\top}_{U_2} \models P \sqsubseteq Q$.

Prime implicates generation. The generation uses a slightly modified version of SPASS v3.9. More specifically, for **R1**, **R2** and Th. 4.4.15, there are corresponding SPASS flags (or set of flags).

Recombination. The construction of hypotheses starts with a straightforward process of matching negative prime implicates with a set of positive ones based on their Skolem terms. It is followed by subset minimality tests to discard non-subset-minimal hypotheses based on Th. 4.4.15. This is because, the quadratic bound entails no guarantee that these hypotheses are non-subset minimal yet connection-minimal. In contrast, if SPASS terminates due to the timeout instead of the bound, then it is possible that some subset-minimal yet also connection-minimal constructible hypotheses are not found.

4.5.2 Experiments

I present three different ways of producing abduction problem $\mathcal{T}$, where in each case, we used the signature of the entire ontology for Σ. The first one `ORIGIN` simply picks a random unentailed axiom. The second one `JUSTIF` picks an entailed axiom and makes it not entailed via single-axiom removal in a specific justification. The last one `REPAIR` picks an entailed axiom but makes it not entailed via repair. The following shows this in more detail.

Definition 4.5.1 (Experimental TBox Abduction)**.** Given a TBox $\mathcal{T}$, we generate the following abduction with a randomly chosen observation $U \sqsubseteq V$.

- (`ORIGIN`) $\langle \mathcal{T}, \Sigma, U \sqsubseteq V \rangle$ with $\mathcal{T} \not\models U \sqsubseteq V$.
- (`JUSTIF`) $\langle \mathcal{J} \setminus \{\alpha\}, \Sigma, U \sqsubseteq V \rangle$ with $U \sqsubseteq V \notin \mathcal{T}$, a justification $\mathcal{J} \models U \sqsubseteq V$ in $\mathcal{T}$ and $\alpha \in \mathcal{J}$.
- (`REPAIR`) $\langle \mathcal{R}, \Sigma, U \sqsubseteq V \rangle$ with $\mathcal{T} \models U \sqsubseteq V$ and a repair $\mathcal{R}$ of $U \sqsubseteq V$ in $\mathcal{T}$.

where $\mathcal{J}$, $\mathcal{R}$, and α are also chosen at random.

`ORIGIN` is simply the basic one and here $U \sqsubseteq V$ is most likely an undesired concept inclusion such as $\mathsf{Cat} \sqsubseteq \mathsf{Dog}$ instead of a missing one like $\mathsf{Cat} \sqsubseteq \mathsf{Mammal}$. In `JUSTIF`, the TBox is smaller and its computation uses the OWL API and ELK. In `REPAIR`, $\mathcal{R}$'s are computed using a justification-based algorithm [SC03] with justifications computed as for `JUSTIF`. This usually resulted in much larger TBoxes, where more axioms would be needed to establish the entailment.

Data. There is no dataset suitable immediately to be a benchmark for this experiment, so a modification of some realistic ontologies becomes necessary. For this, the chosen one would be ontologies from the 2017 snapshot of Bioportal [MP17]. Here, most of them are expressed in a DL more expressive than $\mathcal{EL}$. First, unsupported axioms are eliminated, via the replacement of domain axioms and n-ary equivalence axioms as in [BHLS17]. Then, non-empty TBoxes with at most 50,000 axioms are selected. In the end, we have a set of 387 $\mathcal{EL}$ TBoxes with a size between 2 and 46,429 axioms, an average of 3,039 and a median of 569.

Problem generation. For each of the 387 TBoxes, we attempted to construct and translate five problem instances in each category. The generation is not always successful. In summary, there were

- 25, 28, and 25 failures in finding entailment respectively for `ORIGIN`, `JUSTIF`, and `REPAIR`
- two errors during justification generation for `JUSTIF`
- no errors during repair generation via justification for `REPAIR`, and
- five timeouts during the translation for every category.

Experimental setup In general, the experiment follows Fig. 4.2 where all experiments were run on Debian Linux (Intel Core i5-4590, 3.30 GHz, 23 GB Java heap size). A (hard) time limit of 90 seconds is imposed for each steps in Fig. 4.2. A (soft) time limit of 30 seconds is imposed for SPASS to return the so-far generated implicates. The implementation is available online .[7]

[7] https://lat.inf.tu-dresden.de/~koopmann/IJCAR-2022-Experiments.tar.gz

	$\#\mathcal{H}$	$\|\mathcal{H}\|$	$\|\alpha\|$	time (s.)
`ORIGIN`	1/8.51/1850	1/1.00/2	6/7.48/91	0.2/12.4/43.8
`JUSTIF`	1/1.50/5	1/1/1	2/4.21/32	0.2/1.1/34.1
`REPAIR`	43/228.05/6317	1/1.00/2	5/5.09/49	0.6/13.6/59.9

Table 4.1: Experiment summary (median/avg/max)

Results (SPASS). The success/failure rate for the prime implicate generation using SPASS is summarized in Table 4.2 .

	#Problem	Success	Completed
`ORIGIN`	1,925	94.7%	61.3%
`JUSTIF`	1,803	100.0%	97.2%
`REPAIR`	1,805	92.9%	57.0%

Table 4.2: Success/Completion rates for the prime implicate generation

Table 4.2 shows the percentage of problems for which SPASS does generate a prime implicate (Success) and the percentage of problems for which SPASS is forced to terminate via the soft time limit, where all hypotheses are then computed (Completed). The big size of the TBox (unsurprisingly) correlates with the number of cases where SPASS reached the soft time limit. Moreover, in many of these cases, the bound on the term depth may reach the billion rendering it impractical to compute (but actually most of them are not useful as the later reconstruction results indicate). However, the “Completed” column shows that the bound is reached before the soft time limit in most cases. The reconstruction never reached the hard time limit.

Results (Reconstruction). In addition to SPASS run-time (time, in seconds), Table 4.1, shows the median, average and maximal number of hypotheses found ($\#\mathcal{H}$), size of hypotheses in number of CIs ($|\mathcal{H}|$), size of CIs from hypotheses in the number of atomic concepts ($|\alpha|$).

We can summarize the findings as follows.

- Except for the simple `JUSTIF` problems, the number of hypotheses may become very large while solutions always contain very few (never more than 3) and possibly large axioms. The small number of hypotheses in `JUSTIF` is to be expected, since jusification generation itself drastically reduces the problem size.
- Prime implicates with deeply nested terms are not useful for hypotheses: 8/1/15 hypotheses with the largest depths 3/1/2 for `ORIGIN`/`JUSTIF`/`REPAIR`. This motivates the need for a redundancy criteria for a much earlier termination.

Chapter 5

Conclusion

Entailment Explanation via Relevance

When we have an entailment $N \models C$, any way of explaining it would arguably be most affected by which clauses are chosen to begin with. For this, I proposed a new notion of relevance based on refutations Def. 3.1.4. A clause is syntactically relevant if it occurs in all refutations, it is syntactically semi-relevant if it occurs in some refutation, and syntactically irrelevant otherwise. This notion reflects how we can choose clauses in N with respect to whether they are necessary, optional, or not needed at all in refutations. A natural follow-up issue is what it really means for them semantically. This is furthermore affected by the existence of redundant clauses in an unexpected manner. As a response to this, I additionally proposed (i) a semantic notion of relevance, Def. 3.1.11, based on the existence of conflict literals, Def. 3.1.5 and a first-order refinement of (ir)redundancy called (in)dependence Def. 3.1.10, (ii) its relationship to syntactic relevance, namely their coincidence on independent clause sets, Th. 3.3.5, and (iii) the relationship of semantic relevance to minimal unsatisfiable sets, MUSes, both for propositional, Lemma 3.1.14, and first-order logic, Lemma 3.1.15. Its interesting characteristics have also been illustrated, e.g., in Ex. 3.1.16 (even in the introduction chapter already). In practical settings, there are applications where clause sets are always independent. For example, first-order toolbox formalizations such as the toolbox for car/truck/tractor building [SKK03, FWW16] usually require that every tool is formalized by a unique predicate. Still a goal (refutation) can be proven in many ways by differently picking the tools.

The key challenge in establishing a procedure is the test for semi-relevancy. This led to the extension of the original completeness result for SOS resolution [WRC65], Th. 3.2.7. This guarantees that a clause C is syntactically semi-relevant in N iff there is an SOS resolution refutation from $(N \setminus \{C\}, \{C\})$ Lemma 3.3.2. The key in acquiring this is the refutation transformation described in Sect. 3.2.1. It requires many assumptions (e.g. a priori tree-structuredness, variable disjointness, and the existence of overall grounding substitutions) that are not principal restrictions in nature and only serve to ease the transformation. In addition, this transformation may exponentially increase or exponentially decrease the length of the deduction (as is known when changing inference orderings). In general, this gives us a semi-decision procedure for semi-relevancy test. However, this can further be an effective one when resolution constitutes a decision procedure for the considered fragment. For example, we can effectively test semi-relevancy in all fragments enjoying the bounded model property, such as the Bernays-Schönfinkel fragment [BS28].

An open problem is how one can acquire a more efficient procedure such as by incorporating a restricted resolution calculus like ordered resolution. The challenge here is that the SOS strategy may not necessarily be complete when used in conjunction with ordered resolution. So, we may start off with the fact that it is complete when the clause set is first saturated by ordered resolution. Still, this nevertheless presents us with an obstacle because a saturated clause set may already contain the empty clause, yet for the semi-relevancy test, the generalized completeness permits the set $N \setminus \{C\}$ to be unsatisfiable.

In the context of description logic, the notion of relevance adds an alternative perspective in understanding an ontology in relation to the notion of justifications and repairs. In particular, semi-relevant clauses appear to be particularly connected to laconic justifications and the computation of relevant axioms offers an alternative to existing repair methods. Nevertheless, we have only considered the syntactic notion so far. That is, we have not further explored the unique feature of our notions by considering its semantic characterization. In description logic, a similar notion of relevance is often delivered with the help of justifications. Analogous to the difference between propositional logic and first-order logic with regards to MUS, one can also inquire for which fragments are justification-based notion of relevance sufficient and which are not. In addition to the already existing research regarding laconic and precise justifications, considering how our semantic relevance works in DL context (or to a larger extent any other FOL fragments) is an interesting future work given that there are many DL fragments which are in fact also FOL fragments via translation.

Explaining Non-Entailment via Connection-Minimal Abduction for $\mathcal{EL}$

Existing works related to non-entailment explanation are dominated by abduction. Hypotheses for an abduction problem are often taken/excluded via various minimality notions. I introduced the connection-minimality notion for $\mathcal{EL}$ TBox abduction where the use of axioms in an arbitrary manner is forbidden Def 4.1.9. I presented a formal relation between the connection-minimal hypotheses in $\mathcal{EL}$ and the prime implicates of some first-order of the problem (Lemma 4.2.6, Lemma 4.2.9, and Th. 4.2.11). In addition, we developed a prototype to generate subset-minimal constructible hypotheses, a subset of connection-minimal hypotheses that are in some way representative of the hypotheses. This takes advantages SPASS to generate the prime implicates. I showed that the subset minimal ones use only first-order ground terms that are quadratically bouned Th. 4.4.14. Some set of realistic medical ontologies serves as evaluation data and the results show that its cost can be high but not prohibitive.

There are several ways to further improve efficiency. First, the current termination bound is still too generous and could be advantageously replaced by a redundancy criterion to stop earlier when we know that no useful solutions can be generated anymore. Second, it should also be possible to have stronger inference restrictions by generating only ground clauses and performing it incrementally where terms are always generated in a non-decreasing manner. Third, one can of course investigate methods that rely solely on description logic calculi.

The theoretical worst-case complexity of connection-minimal abduction is another open question. Th. 4.3.2 permits exponentially-sized clauses leading to doubly exponential number of clauses in the worst case. This would give us an 2ExpTime upper bound. However, structure-sharing and guessing may likely lower this bound or at least give us a higher efficiency.

Another class of future works are the extensional ones. First, it may be interesting to generalize this to ontology abduction where ABox is involved. This may possibly be supported by the fact that prime implicates are basically ground truths. Second, abduction with conjunctive queries also looks intriguing. One motivation of this is to understand why a particular query returns nothing (but should not) and possibly offer a fix via the hypotheses. Third, it may in some way be reasonable to allow role restrictions in the hypotheses. Last, one would obviously want to see how connection-minimality looks like for more expressive DLs such as $\mathcal{ALC}$.

Bibliography

[AB70] Robert Anderson and W.W. Bledsoe. A linear format for resolution with merging and a new technique for establishing completeness. *Journal of the ACM*, 17:525–534, July 1970.

[ABB+20a] Christian Alrabbaa, Franz Baader, Stefan Borgwardt, Patrick Koopmann, and Alisa Kovtunova. Finding small proofs for description logic entailments: Theory and practice. In Elvira Albert and Laura Kovacs, editors, *LPAR-23: 23rd International Conference on Logic for Programming, Artificial Intelligence and Reasoning*, volume 73 of *EPiC Series in Computing*, pages 32–67. EasyChair, 2020.

[ABB+20b] Christian Alrabbaa, Franz Baader, Stefan Borgwardt, Patrick Koopmann, and Alisa Kovtunova. On the complexity of finding good proofs for description logic entailments. In Stefan Borgwardt and Thomas Meyer, editors, *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*, volume 2663 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2020.

[ABB+21] Christian Alrabbaa, Franz Baader, Stefan Borgwardt, Patrick Koopmann, and Alisa Kovtunova. Finding good proofs for description logic entailments using recursive quality measures. In *Proceedings of the 28th International Conference on Automated Deduction (CADE-28), July 11–16, 2021, Virtual Event, United States*, 2021. To appear.

[ABB+22] Christian Alrabbaa, Franz Baader, Stefan Borgwardt, Raimund Dachselt, Patrick Koopmann, and Julián Méndez. Evonne: Interactive proof visualization for description logics (system description). In Jasmin Blanchette, Laura Kovács, and Dirk Pattinson, editors, *Automated Reasoning - 11th International Joint Conference, IJCAR 2022, Haifa, Israel, August 8-10, 2022, Proceedings*, volume 13385 of *Lecture Notes in Computer Science*, pages 271–280. Springer, 2022.

[ABMP98] Michael Alekhnovich, Sam Buss, Shlomo Moran, and Toniann Pitassi. Minimum propositional proof length is np-hard to linearly approximate. In Luboš Brim, Jozef Gruska, and Jiří Zlatuška, editors, *Mathematical Foundations of Computer Science 1998*, pages 176–184, Berlin, Heidelberg, 1998. Springer Berlin Heidelberg.

[AMM15] M. Fareed Arif, Carlos Mencía, and João 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*, pages 324–342, 2015.

[BB07] Martins Barinskis and Guntis Barzdins. Satisfiability model visualization plugin for deep consistency checking of OWL ontologies. In Christine Golbreich, Aditya Kalyanpur, and Bijan Parsia, editors, *Proceedings of the OWLED 2007 Workshop on OWL: Experiences and Directions, Innsbruck, Austria, June 6-7, 2007*, volume 258 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2007.

[BCM+03] Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.

[BG94] Leo Bachmair and Harald Ganzinger. Rewrite-based equational theorem proving with selection and simplification. *Journal of Logic and Computation*, 4(3):217–247, 1994. Revised version of Max-Planck-Institut für Informatik technical report, MPI-I-91-208, 1991.

[BG01] Leo Bachmair and Harald Ganzinger. Resolution theorem proving. In Alan Robinson and Andrei Voronkov, editors, *Handbook of Automated Reasoning*, volume I, chapter 2, pages 19–99. Elsevier, 2001.

[BHLS17] Franz Baader, Ian Horrocks, Carsten Lutz, and Uli Sattler. *An Introduction to Description Logic*. Cambridge University Press, 2017.

[Bie07] Meghyn Bienvenu. Prime implicates and prime implicants in modal logic. In *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence, July 22-26, 2007, Vancouver, British Columbia, Canada*, pages 379–384. AAAI Press, 2007.

[Bie08] Meghyn Bienvenu. Complexity of abduction in the $\mathcal{EL}$ family of lightweight description logics. In *Proceedings of KR 2008*, pages 220–230. AAAI Press, 2008.

[BJLM12] Anton Belov, Mikolás Janota, Inês Lynce, and João Marques-Silva. On computing minimal equivalent subformulas. In Michela Milano, editor, *Principles and Practice of Constraint Programming - 18th International Conference, CP 2012, Québec City, QC, Canada, October 8-12, 2012. Proceedings*, volume 7514 of *Lecture Notes in Computer Science*, pages 158–174. Springer, 2012.

[BK16] Fahiem Bacchus and George Katsirelos. Finding a collection of muses incrementally. In Claude-Guy Quimper, editor, *Integration of AI and OR Techniques in Constraint Programming - 13th International Conference, CPAIOR 2016, Banff, AB, Canada, May 29 - June 1, 2016, Proceedings*, volume 9676 of *Lecture Notes in Computer Science*, pages 35–44. Springer, 2016.

[BKM99] Franz Baader, Ralf Küsters, and Ralf Molitor. Computing least common subsumers in description logics with existential restrictions. In *Proceedings of IJCAI 1999*, pages 96–103. Morgan Kaufmann, 1999.

[Bla13] Jasmin Christian Blanchette. Redirecting proofs by contradiction. In Jasmin Christian Blanchette and Josef Urban, editors, *Third International Workshop on Proof Exchange for Theorem Proving, PxTP 2013, Lake Placid, NY, USA, June 9-10, 2013*, volume 14 of *EPiC Series in Computing*, pages 11–26. EasyChair, 2013.

[BM21] Jaroslav Bendík and Kuldeep S. Meel. Counting minimal unsatisfiable subsets. In Alexandra Silva and K. Rustan M. Leino, editors, *Computer Aided Verification - 33rd International Conference, CAV 2021, Virtual Event, July 20-23, 2021, Proceedings, Part II*, volume 12760 of *Lecture Notes in Computer Science*, pages 313–336. Springer, 2021.

[BN03] Franz Baader and Werner Nutt. *Basic Description Logics*, page 43–95. Cambridge University Press, USA, 2003.

[BO15] Meghyn Bienvenu and Magdalena Ortiz. Ontology-mediated query answering with data-tractable description logics. In Wolfgang Faber and Adrian Paschke, editors, *Reasoning Web. Web Logic Rules - 11th International Summer School 2015, Berlin, Germany, July 31 - August 4, 2015, Tutorial Lectures*, volume 9203 of *Lecture Notes in Computer Science*, pages 218–307. Springer, 2015.

[BO19] Camille Bourgaux and Ana Ozaki. Querying attributed dl-lite ontologies using provenance semirings. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019*, pages 2719–2726. AAAI Press, 2019.

[BOPP20] Camille Bourgaux, Ana Ozaki, Rafael Peñaloza, and Livia Predoiu. Provenance for the description logic elhr. In Christian Bessiere, editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 1862–1869. ijcai.org, 2020.

[Bor94] Alexander Borgida. On the relationship between description logic and predicate logic. In *Proceedings of the Third International Conference on Information and Knowledge Management (CIKM'94), Gaithersburg, Maryland, USA, November 29 - December 2, 1994*, pages 219–225. ACM, 1994.

[BP10] Franz Baader and Rafael Peñaloza. Axiom pinpointing in general tableaux. *J. Log. Comput.*, 20(1):5–34, 2010.

[BPS07] Franz Baader, Rafael Peñaloza, and Boontawee Suntisrivaraporn. Pinpointing in the description logic EL^+. In Joachim Hertzberg, Michael Beetz, and Roman Englert, editors, *KI 2007: Advances in Artificial Intelligence, 30th Annual German Conference on AI, KI 2007, Osnabrück, Germany, September 10-13, 2007, Proceedings*, volume 4667 of *Lecture Notes in Computer Science*, pages 52–67. Springer, 2007.

[BR00] Yacine Boufkhad and Olivier Roussel. Redundancy in random SAT formulas. In Henry A. Kautz and Bruce W. Porter, editors, *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*, pages 273–278. AAAI Press / The MIT Press, 2000.

[BS28] Paul Bernays and Moses Schönfinkel. Zum entscheidungsproblem der mathematischen logik. *Mathematische Annalen*, 99:342–372, 1928.

[BSP09] Johannes Bauer, Ulrike Sattler, and Bijan Parsia. Explaining by example: Model exploration for ontology comprehension. In Bernardo Cuenca Grau, Ian Horrocks, Boris Motik, and Ulrike Sattler, editors, *Proceedings of the 22nd International Workshop on Description Logics (DL 2009), Oxford, UK, July 27-30, 2009*, volume 477 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2009.

[CGS07] Alessandro Cimatti, Alberto Griggio, and Roberto Sebastiani. A simple and flexible way of computing small unsatisfiable cores in SAT modulo theories. In João Marques-Silva and Karem A. Sakallah, editors, *Theory and Applications of Satisfiability Testing - SAT 2007, 10th International Conference, Lisbon, Portugal, May 28-31, 2007, Proceedings*, volume 4501 of *Lecture Notes in Computer Science*, pages 334–339. Springer, 2007.

[CGS11] Alessandro Cimatti, Alberto Griggio, and Roberto Sebastiani. Computing small unsatisfiable cores in satisfiability modulo theories. *J. Artif. Intell. Res.*, 40:701–728, 2011.

[CLM+20] İsmail İlkan Ceylan, Thomas Lukasiewicz, Enrico Malizia, Cristian Molinaro, and Andrius Vaicenavicius. Explanations for negative query answers under existential rules. In Diego Calvanese, Esra Erdem, and Michael Thielscher, editors, *Proceedings of KR 2020*, pages 223–232. AAAI Press, 2020.

[CMPY22] Jieying Chen, Yue Ma, Rafael Peñaloza, and Hui Yang. Union and intersection of all justifications. In Paul Groth, Maria-Esther Vidal, Fabian M. Suchanek, Pedro A. Szekely, Pavan Kapanipathi, Catia Pesquita, Hala Skaf-Molli, and Minna Tamper, editors, *The Semantic Web - 19th International Conference, ESWC 2022, Hersonissos,* *Crete, Greece, May 29 - June 2, 2022, Proceedings*, volume 13261 of *Lecture Notes in Computer Science*, pages 56–73. Springer, 2022.

[CNS+04] Simona Colucci, Tommaso Di Noia, Eugenio Di Sciascio, Francesco M. Donini, and Marina Mongiello. A uniform tableaux-based approach to concept abduction and contraction in ALN. In Volker Haarslev and Ralf Möller, editors, *Proceedings of the 2004 International Workshop on Description Logics (DL2004), Whistler, British Columbia, Canada, June 6-8, 2004*, volume 104 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2004.

[Cos96] Yann Coscoy. A natural language explanation for formal proofs. In Christian Retoré, editor, *Logical Aspects of Computational Linguistics, First International Conference, LACL '96, Nancy, France, September 23-25, 1996, Selected Papers*, volume 1328 of *Lecture Notes in Computer Science*, pages 149–167. Springer, 1996.

[COSS13] Diego Calvanese, Magdalena Ortiz, Mantas Simkus, and Giorgio Stefanoni. Reasoning about explanations for negative query answers in DL-Lite. *J. Artif. Intell. Res.*, 48:635–669, 2013.

[CP86] Philip T. Cox and Tomasz Pietrzykowski. Causes for events: Their computation and applications. In Jörg H. Siekmann, editor, *8th International Conference on Automated Deduction, Oxford, England, July 27 - August 1, 1986, Proceedings*, volume 230 of *Lecture Notes in Computer Science*, pages 608–621. Springer, 1986.

[CS03] Koen Claessen and Niklas Sörensson. New techniques that improve mace-style finite model finding. In *Proceedings of the CADE-19 Workshop: Model Computation - Principles, Algorithms, Applications*, 2003.

[DHS06] Xi Deng, Volker Haarslev, and Nematollaah Shiri. Resolution based explanations for reasoning in the description logic $\mathcal{ALC}$. In Mamadou Tadiou Kone and Daniel Lemire, editors, *Canadian Semantic Web, CSWWS 2006, first Canadian Semantic Web Working Symposium, June 2006, Quebec, Canada*, volume 2 of *Semantic Web and Beyond: Computing for Human Experience*, pages 189–204. Springer, 2006.

[DQSP12] Jianfeng Du, Guilin Qi, Yi-Dong Shen, and Jeff Z. Pan. Towards practical ABox abduction in large description logic ontologies. *Int. J. Semantic Web Inf. Syst.*, 8(2):1–33, 2012.

[DS17] Warren Del-Pinto and Renate A. Schmidt. Forgetting-based abduction in ALC. In Patrick Koopmann, Sebastian Rudolph, Renate A. Schmidt, and Christoph Wernhard, editors, *Proceedings of the Workshop on Second-Order Quantifier Elimination and Related Topics (SOQE 2017), Dresden, Germany, December 6-8, 2017*, volume 2013 of *CEUR Workshop Proceedings*, pages 27–35. CEUR-WS.org, 2017.

[DS19] Warren Del-Pinto and Renate A. Schmidt. ABox abduction via forgetting in $\mathcal{ALC}$. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019*, pages 2768–2775. AAAI Press, 2019.

[DWM17a] Jianfeng Du, Hai Wan, and Huaguan Ma. Practical TBox abduction based on justification patterns. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pages 1100–1106, 2017.

[DWM17b] Jianfeng Du, Hai Wan, and Huaguan Ma. Practical TBox abduction based on justification patterns. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pages 1100–1106, 2017.

[DWS14] Jianfeng Du, Kewen Wang, and Yi-Dong Shen. A tractable approach to ABox abduction over description logic ontologies. In Carla E. Brodley and Peter Stone, editors, *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, pages 1034–1040. AAAI Press, 2014.

[EG95] Thomas Eiter and Georg Gottlob. The complexity of logic-based abduction. *Journal of the ACM*, 42(1):3–42, 1995.

[EKS06] Corinna Elsenbroich, Oliver Kutz, and Ulrike Sattler. A case for abductive reasoning over ontologies. In *Proceedings of the OWLED'06 Workshop on OWL: Experiences and Directions*, 2006.

[EPS18] Mnacho Echenim, Nicolas Peltier, and Yanis Sellami. A generic framework for implicate generation modulo theories. In Didier Galmiche, Stephan Schulz, and Roberto Sebastiani, editors, *Automated Reasoning - 9th International Joint Conference, IJCAR 2018, Held as Part of the Federated Logic Conference, FloC 2018, Oxford, UK, July 14-17, 2018, Proceedings*, volume 10900 of *Lecture Notes in Computer Science*, pages 279–294. Springer, 2018.

[Fie05] Armin Fiedler. Natural language proof explanation. In Dieter Hutter and Werner Stephan, editors, *Mechanizing Mathematical Reasoning, Essays in Honor of Jörg H. Siekmann on the Occasion of His 60th Birthday*, volume 2605 of *Lecture Notes in Computer Science*, pages 342–363. Springer, 2005.

[Fit90] Melvin Fitting. *First-Order Logic*, pages 97–125. Springer US, New York, NY, 1990.

[FWW16] Christof Fetzer, Christoph Weidenbach, and Patrick Wischnewski. Compliance, functional safety and fault detection by formal methods. In Tiziana Margaria and Bernhard Steffen, editors, *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*, volume 9953 of *Lecture Notes in Computer Science*, pages 626–632, 2016.

[GHKS08] Bernardo Cuenca Grau, Ian Horrocks, Yevgeny Kazakov, and Ulrike Sattler. Modular reuse of ontologies: Theory and practice. *J. Artif. Intell. Res.*, 31:273–318, 2008.

[GKT07] Todd J. Green, Gregory Karvounarakis, and Val Tannen. Provenance semirings. In Leonid Libkin, editor, *Proceedings of the Twenty-Sixth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, June 11-13, 2007, Beijing, China*, pages 31–40. ACM, 2007.

[GST16] Ofer Guthmann, Ofer Strichman, and Anna Trostanetski. Minimal unsatisfiable core extraction for SMT. In Ruzica Piskac and Muralidhar Talupur, editors, *2016 Formal Methods in Computer-Aided Design, FMCAD 2016, Mountain View, CA, USA, October 3-6, 2016*, pages 57–64. IEEE, 2016.

[Hak85] Armin Haken. The intractability of resolution. *Theoretical Computer Science*, 39:297–308, 1985. Third Conference on Foundations of Software Technology and Theoretical Computer Science.

[HB11] Matthew Horridge and Sean Bechhofer. The OWL API: a java API for OWL ontologies. *Semantic Web*, 2(1):11–21, 2011.

[HB12] Ken Halland and Katarina Britz. ABox abduction in $\mathcal{ALC}$ using a DL tableau. In *2012 South African Institute of Computer Scientists and Information Technologists Conference, SAICSIT '12*, pages 51–58, 2012.

[HK10] Anthony Hunter and Sébastien Konieczny. On the measure of conflicts: Shapley inconsistency values. *Artif. Intell.*, 174(14):1007–1026, 2010.

[HKT21a] Fajar Haifani, Patrick Koopmann, and Sophie Tourret. Abduction in EL via translation to FOL. In Renate A. Schmidt, Christoph Wernhard, and Yizheng Zhao, editors, *Proceedings of the Second Workshop on Second-Order Quantifier Elimination and Related Topics (SOQE 2021) associated with the 18th International Conference on Principles of Knowledge Representation and Reasoning (KR 2021), Online Event, November 4, 2021*, volume 3009 of *CEUR Workshop Proceedings*, pages 46–58. CEUR-WS.org, 2021.

[HKT21b] Fajar Haifani, Patrick Koopmann, and Sophie Tourret. Introducing connection-minimal abduction for el ontologies. In *XLoKR workshop*, 2021.

[HKTW20] Fajar Haifani, Patrick Koopmann, Sophie Tourret, and Christoph Weidenbach. On a notion of relevance. In Stefan Borgwardt and Thomas Meyer, editors, *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*, volume 2663 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2020.

[HKTW22a] Fajar Haifani, Patrick Koopmann, Sophie Tourret, and Christoph Weidenbach. Connection-minimal abduction in EL via translation to FOL. In *Automated Reasoning - 11th International Joint Conference, IJCAR 2022, Held as Part of the Federated Logic Conference, FloC 2022, Haifa, Israel, August 8-10, 2022, Proceedings*, 2022.

[HKTW22b] Fajar Haifani, Patrick Koopmann, Sophie Tourret, and Christoph Weidenbach. Connection-minimal abduction in el via translation to fol–technical report. *arXiv preprint arXiv:2205.08449*, 2022.

[HLP10] Thomas Hubauer, Steffen Lamparter, and Michael Pirker. Automata-based abduction for tractable diagnosis. In Volker Haarslev, David Toman, and Grant E. Weddell, editors, *Proceedings of the 23rd International Workshop on Description Logics (DL 2010), Waterloo, Ontario, Canada, May 4-7, 2010*, volume 573 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2010.

[HPS08a] Matthew Horridge, Bijan Parsia, and Ulrike Sattler. Explanation of OWL entailments in protege 4. In Christian Bizer and Anupam Joshi, editors, *Proceedings of the Poster and Demonstration Session at the 7th International Semantic Web Conference (ISWC2008), Karlsruhe, Germany, October 28, 2008*, volume 401 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2008.

[HPS08b] Matthew Horridge, Bijan Parsia, and Ulrike Sattler. Laconic and precise justifications in OWL. In *The Semantic Web - ISWC 2008, 7th International Semantic Web Conference, ISWC 2008, Karlsruhe, Germany, October 26-30, 2008. Proceedings*, pages 323–338, 2008.

[HS02] Ullrich Hustadt and Renate A. Schmidt. Using resolution for testing modal satisfiability and building models. *Journal of Automed Reasoning*, 28(2):205–232, 2002.

[HTW21] Fajar Haifani, Sophie Tourret, and Christoph Weidenbach. Generalized completeness for SOS resolution and its application to a new notion of relevance. In André Platzer and Geoff Sutcliffe, editors, *Automated Deduction - CADE 28 - 28th International Conference on Automated Deduction, Virtual Event, July 12-15, 2021, Proceedings*, volume 12699 of *Lecture Notes in Computer Science*, pages 327–343. Springer, 2021.

[Hua94] Xiaorong Huang. Reconstruction proofs at the assertion level. In Alan Bundy, editor, *Automated Deduction - CADE-12, 12th International Conference on Automated Deduction, Nancy, France, June 26 - July 1, 1994, Proceedings*, volume 814 of *Lecture Notes in Computer Science*, pages 738–752. Springer, 1994.

[Hub16] Thomas Hubauer. *Relaxed Abduction - Robust Information Interpretation for Industrial Applications*. Springer Vieweg, 2016.

[HW22] Fajar Haifani and Christoph Weidenbach. Semantic relevance. In *Automated Reasoning - 11th International Joint Conference, IJCAR 2022, Held as Part of the Federated Logic Conference, FloC 2022, Haifa, Israel, August 8-10, 2022, Proceedings*, 2022.

[Iwa97] Kazuo Iwama. Complexity of finding short resolution proofs. In Igor Prívara and Peter Ružička, editors, *Mathematical Foundations of Computer Science 1997*, pages 309–318, Berlin, Heidelberg, 1997. Springer Berlin Heidelberg.

[Jac06] Daniel Jackson. *Software Abstractions - Logic, Language, and Analysis*. MIT Press, 2006.

[JMRS17] Saïd Jabbour, Yue Ma, Badran Raddaoui, and Lakhdar Sais. Quantifying conflicts in propositional logic through prime implicates. *Int. J. Approx. Reason.*, 89:27–40, 2017.

[KDTS20] Patrick Koopmann, Warren Del-Pinto, Sophie Tourret, and Renate A. Schmidt. Signature-based abduction for expressive description logics. In Diego Calvanese, Esra Erdem, and Michael Thielscher, editors, *Proceedings of the 17th International Conference on Principles of Knowledge Representation and Reasoning, KR 2020*, pages 592–602. AAAI Press, 2020.

[KES11] Szymon Klarman, Ulle Endriss, and Stefan Schlobach. ABox abduction in the description logic $\mathcal{ALC}$. *Journal of Automated Reasoning*, 46(1):43–80, 2011.

[KK09] Hans Kleine Büning and Oliver Kullmann. Minimal unsatisfiability and autarkies. In Armin Biere, Marijn Heule, Hans van Maaren, and Toby Walsh, editors, *Handbook of Satisfiability*, volume 185 of *Frontiers in Artificial Intelligence and Applications*, pages 339–401. IOS Press, 2009.

[KKS14] Yevgeny Kazakov, Markus Krötzsch, and Frantisek Simancik. The incredible ELK - from polynomial procedures to efficient reasoning with $\mathcal{EL}$ ontologies. *J. Autom. Reason.*, 53(1):1–61, 2014.

[KKS17] Yevgeny Kazakov, Pavel Klinov, and Alexander Stupnikov. Towards reusable explanation services in protege. In Alessandro Artale, Birte Glimm, and Roman Kontchakov, editors, *Proceedings of the 30th International Workshop on Description Logics, Montpellier, France, July 18-21, 2017*, volume 1879 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2017.

[KLM06] Oliver Kullmann, Inês Lynce, and João Marques-Silva. Categorisation of clauses in conjunctive normal forms: Minimally unsatisfiable sub-clause-sets and the lean kernel. In Armin Biere and Carla P. Gomes, editors, *Theory and Applications of Satisfiability Testing - SAT 2006, 9th International Conference, Seattle, WA, USA, August 12-15, 2006, Proceedings*, volume 4121 of *Lecture Notes in Computer Science*, pages 22–35. Springer, 2006.

[Koo21a] Patrick Koopmann. Signature-based abduction with fresh individuals and complex concepts for description logics. In Zhi-Hua Zhou, editor, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*, pages 1929–1935. ijcai.org, 2021.

[Koo21b] Patrick Koopmann. Two ways of explaining negative entailments in description logics using abduction (extended version). LTCS-Report 21-03, Chair of Automata Theory, Institute of Theoretical Computer Science, Technische Universität Dresden, 2021. https://lat.inf.tu-dresden.de/research/reports.html.

[KPHS07] Aditya Kalyanpur, Bijan Parsia, Matthew Horridge, and Evren Sirin. Finding all justifications of OWL DL entailments. In Karl Aberer, Key-Sun Choi, Natasha Fridman Noy, Dean Allemang, Kyung-Il Lee, Lyndon J. B. Nixon, Jennifer Golbeck, Peter Mika, Diana Maynard, Riichiro Mizoguchi, Guus Schreiber, and Philippe Cudré-Mauroux, editors, *The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11-15, 2007*, volume 4825 of *Lecture Notes in Computer Science*, pages 267–280. Springer, 2007.

[Kul00] Oliver Kullmann. Investigations on autark assignments. *Discret. Appl. Math.*, 107(1-3):99–137, 2000.

[KUV15] Cezary Kaliszyk, Josef Urban, and Jirí Vyskocil. Lemmatization for stronger reasoning in large theories. In Carsten Lutz and Silvio Ranise, editors, *Frontiers of Combining Systems - 10th International Symposium, FroCoS 2015, Wroclaw, Poland, September 21-24, 2015. Proceedings*, volume 9322 of *Lecture Notes in Computer Science*, pages 341–356. Springer, 2015.

[LAWRS07] Tal Lev-Ami, Christoph Weidenbach, Thomas W. Reps, and Mooly Sagiv. Labelled clauses. In Frank Pfenning, editor, *Automated Deduction - CADE-21, 21st International Conference on Automated Deduction, Bremen, Germany, July 17-20, 2007, Proceedings*, volume 4603 of *LNCS*, pages 311–327. Springer, 2007.

[Lee67] Char-Tung Lee. *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.

[Lib05] Paolo Liberatore. Redundancy in logic I: CNF propositional formulae. *Artif. Intell.*, 163(2):203–232, 2005.

[LL18] Shaofan Liu and Jie Luo. FMUS2: an efficient algorithm to compute minimal unsatisfiable subsets. In Jacques D. Fleuriot, Dongming Wang, and Jacques Calmet, editors, *Artificial Intelligence and Symbolic Computation - 13th International Conference, AISC 2018, Suzhou, China, September 16-19, 2018, Proceedings*, volume 11110 of *Lecture Notes in Computer Science*, pages 104–118. Springer, 2018.

[Llo87] John Wylie Lloyd. *Foundations of Logic Programming.* Symbolic Computation. Springer, second edition, 1987. First edition appeared in 1984.

[Mar91] Pierre Marquis. Extending abduction from propositional to first-order logic. In Philippe Jorrand and Jozef Kelemen, editors, *Fundamentals of Artificial Intelligence Research, International Workshop FAIR '91, Smolenice, Czechoslovakia, September 8-13, 1991, Proceedings*, volume 535 of *Lecture Notes in Computer Science*, pages 141–155. Springer, 1991.

[McC03] William McCune. Mace4 reference manual and guide. *CoRR*, cs.SC/0310055, 2003.

[MKIM19] Carlos Mencía, Oliver Kullmann, Alexey Ignatiev, and João Marques-Silva. On computing the union of muses. In Mikolás Janota and Inês Lynce, editors, *Theory and Applications of Satisfiability Testing - SAT 2019 - 22nd International Conference, SAT 2019, Lisbon, Portugal, July 9-12, 2019, Proceedings*, volume 11628 of *Lecture Notes in Computer Science*, pages 211–221. Springer, 2019.

[MM20] João Marques-Silva and Carlos Mencía. Reasoning about inconsistent formulas. In Christian Bessiere, editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 4899–4906. ijcai.org, 2020.

[MP93] Marta Cialdea Mayer and Fiora Pirri. First order abduction via tableau and sequent calculi. *Log. J. IGPL*, 1(1):99–117, 1993.

[MP95] MARTA CIALDEA MAYER and FIORA PIRRI. Propositional Abduction in Modal Logic. *Logic Journal of the IGPL*, 3(6):907–919, 10 1995.

[MP17] Nicolas Matentzoglu and Bijan Parsia. Bioportal snapshot 30.03.2017, March 2017.

[MQHL07] Yue Ma, Guilin Qi, Pascal Hitzler, and Zuoquan Lin. Measuring inconsistency for description logics based on paraconsistent semantics. In Khaled Mellouli, editor, *Symbolic and Quantitative Approaches to Reasoning with Uncertainty, 9th European Conference, ECSQARU 2007, Hammamet, Tunisia, October 31 - November 2, 2007, Proceedings*, volume 4724 of *Lecture Notes in Computer Science*, pages 30–41. Springer, 2007.

[NdW95] Shan-Hwei Nienhuys-Cheng and Ronald de Wolf. The equivalence of the subsumption theorem and the refutation-completeness for unconstrained resolution. In Kanchana Kanchanasut and Jean-Jacques Lévy, editors, *Algorithms, Concurrency and Knowledge: 1995 Asian Computing Science Conference, ACSC '95,* *Pathumthani, Thailand, December 11-13, 1995, Proceedings*, volume 1023 of *Lecture Notes in Computer Science*, pages 269–285. Springer, 1995.

[NIIR10] Hidetomo Nabeshima, Koji Iwanuma, Katsumi Inoue, and Oliver Ray. SOLAR: an automated deduction system for consequence finding. *AI Commun.*, 23(2-3):183–203, 2010.

[NR01] Robert Nieuwenhuis and Albert Rubio. Paramodulation-based theorem proving. In Alan Robinson and Andrei Voronkov, editors, *Handbook of Automated Reasoning*, volume I, chapter 7, pages 371–443. Elsevier, 2001.

[NRW98] Andreas Nonnengart, Georg Rock, and Christoph Weidenbach. On generating small clause normal forms. In Claude Kirchner and Hélène Kirchner, editors, *Automated Deduction - CADE-15, 15th International Conference on Automated Deduction, Lindau, Germany, July 5-10, 1998, Proceedings*, volume 1421 of *Lecture Notes in Computer Science*, pages 397–411. Springer, 1998.

[NW01] Andreas Nonnengart and Christoph Weidenbach. Computing small clause normal forms. In John Alan Robinson and Andrei Voronkov, editors, *Handbook of Automated Reasoning (in 2 volumes)*, pages 335–367. Elsevier and MIT Press, 2001.

[Pfe84] Frank Pfenning. Analytic and non-analytic proofs. In Robert E. Shostak, editor, *7th International Conference on Automated Deduction, Napa, California, USA, May 14-16, 1984, Proceedings*, volume 170 of *Lecture Notes in Computer Science*, pages 394–413. Springer, 1984.

[PH17] Júlia Pukancová and Martin Homola. Tableau-based ABox abduction for the $\mathcal{ALCHO}$ description logic. In *Proceedings of the 30th International Workshop on Description Logics*, 2017.

[PH20] Júlia Pukancová and Martin Homola. The AAA ABox abduction solver. *Künstliche Intell.*, 34(4):517–522, 2020.

[PMIM17] Rafael Peñaloza, Carlos Mencía, Alexey Ignatiev, and João Marques-Silva. Lean kernels in description logics. In Eva Blomqvist, Diana Maynard, Aldo Gangemi, Rinke Hoekstra, Pascal Hitzler, and Olaf Hartig, editors, *The Semantic Web - 14th International Conference, ESWC 2017, Portorož, Slovenia, May 28 - June 1, 2017, Proceedings, Part I*, volume 10249 of *Lecture Notes in Computer Science*, pages 518–533, 2017.

[Pop73] Harry E. Pople. On the mechanization of abductive logic. In Nils J. Nilsson, editor, *Proceedings of the 3rd International Joint Conference on Artificial Intelligence. Standford, CA, USA, August 20-23, 1973*, pages 147–152. William Kaufmann, 1973.

[PS17] Rafael Peñaloza and Barıs Sertkaya. Understanding the complexity of axiom pinpointing in lightweight description logics. *Artif. Intell.*, 250:80–104, 2017.

[PSK05] Bijan Parsia, Evren Sirin, and Aditya Kalyanpur. Debugging OWL ontologies. In Allan Ellis and Tatsuya Hagino, editors, *Proceedings of the 14th international conference on World Wide Web, WWW 2005, Chiba, Japan, May 10-14, 2005*, pages 633–640. ACM, 2005.

[PW88] Christos H. Papadimitriou and David Wolfe. The complexity of facets resolved. *J. Comput. Syst. Sci.*, 37(1):2–13, 1988.

[PY84] Christos H. Papadimitriou and Mihalis Yannakakis. The complexity of facets (and some facets of complexity). *J. Comput. Syst. Sci.*, 28(2):244–259, 1984.

[Rei87] Raymond Reiter. A theory of diagnosis from first principles. *Artif. Intell.*, 32(1):57–95, 1987.

[Rob65] John Alan Robinson. A machine-oriented logic based on the resolution principle. *Journal of the ACM*, 12(1):23–41, January 1965.

[RS21] Markus N. Rabe and Christian Szegedy. Towards the automatic mathematician. In André Platzer and Geoff Sutcliffe, editors, *Automated Deduction - CADE 28 - 28th International Conference on Automated Deduction, Virtual Event, July 12-15, 2021, Proceedings*, volume 12699 of *Lecture Notes in Computer Science*, pages 25–37. Springer, 2021.

[RV01] John Alan Robinson and Andrei Voronkov, editors. *Handbook of Automated Reasoning (in 2 volumes).* Elsevier and MIT Press, 2001.

[SC03] Stefan Schlobach and Ronald Cornet. Non-standard reasoning services for the debugging of description logic terminologies. In Georg Gottlob and Toby Walsh, editors, *Proc. of the 18th Int. Joint Conf. on Artificial Intelligence (IJCAI 2003)*, pages 355–362, Acapulco, Mexico, 2003. Morgan Kaufmann.

[SKK03] Carsten Sinz, Andreas Kaiser, and Wolfgang Küchlin. Formal methods for the validation of automotive product configuration data. *Artif. Intell. Eng. Des. Anal. Manuf.*, 17(1):75–97, 2003.

[TJ07] Emina Torlak and Daniel Jackson. Kodkod: A relational model finder. In Orna Grumberg and Michael Huth, editors, *Tools and Algorithms for the Construction and Analysis of Systems, 13th International Conference, TACAS 2007, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2007 Braga, Portugal, March 24 - April 1, 2007, Proceedings*, volume 4424 of *Lecture Notes in Computer Science*, pages 632–647. Springer, 2007.

[Tou16] Sophie Tourret. *Abduction in first order logic with equality. (Prime implicate generation in equational logic).* PhD thesis, Grenoble Alpes University, France, 2016.

[WDL14] Fang Wei-Kleiner, Zlatan Dragisic, and Patrick Lambrix. Abduction framework for repairing incomplete $\mathcal{EL}$ ontologies: Complexity results and algorithms. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, pages 1120–1127. AAAI Press, 2014.

[WFK17] Rouven Walter, Alexander Felfernig, and Wolfgang Küchlin. Constraint-based and sat-based diagnosis of automotive configuration problems. *J. Intell. Inf. Syst.*, 49(1):87–118, 2017.

[WP07] Taowei David Wang and Bijan Parsia. Ontology performance profiling and model examination: First steps. In Karl Aberer, Key-Sun Choi, Natasha Fridman Noy, Dean Allemang, Kyung-Il Lee, Lyndon J. B. Nixon, Jennifer Golbeck, Peter Mika, Diana Maynard, Riichiro Mizoguchi, Guus Schreiber, and Philippe Cudré-Mauroux, editors, *The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11-15, 2007*, volume 4825 of *Lecture Notes in Computer Science*, pages 595–608. Springer, 2007.

[WRC65] Larry Wos, George A. Robinson, and Daniel F. Carson. Efficiency and completeness of the set of support strategy in theorem proving. *J. ACM*, 12(4):536–541, 1965.

[WSH+07] Christoph Weidenbach, Renate A. Schmidt, Thomas Hillenbrand, Rostislav Rusev, and Dalibor Topic. System description: SPASS Version 3.0. In Frank Pfenning, editor, *Automated Deduction - CADE-21, 21st International Conference on Automated Deduction, Bremen, Germany, July 17-20, 2007, Proceedings*, volume 4603 of *Lecture Notes in Computer Science*, pages 514–520. Springer, 2007.

[XL16] Huiyuan Xie and Jie Luo. 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*, pages 444–451. IEEE Computer Society, 2016.

[ZXZ+11] Jianmin Zhang, Weixia Xu, Jun Zhang, ShengYu Shen, Zhengbin Pang, Tiejun Li, Jun Xia, and Sikun Li. Finding first-order minimal unsatisfiable cores with a heuristic depth-first-search algorithm. In Hujun Yin, Wenjia Wang, and Victor J. Rayward-Smith, editors, *Intelligent Data Engineering and Automated Learning - IDEAL 2011 - 12th International Conference, Norwich, UK, September 7-9, 2011. Proceedings*, volume 6936 of *Lecture Notes in Computer Science*, pages 178–185. Springer, 2011.

[ZZ95] Jian Zhang and Hantao Zhang. SEM: a system for enumerating models. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, IJCAI 95, Montréal Québec, Canada, August 20-25 1995, 2 Volumes*, pages 298–303. Morgan Kaufmann, 1995.

[ZZ13] Hantao Zhang and Jian Zhang. MACE4 and SEM: A comparison of finite model generators. In Maria Paola Bonacina and Mark E. Stickel, editors, *Automated Reasoning and Mathematics - Essays in Memory of William W. McCune*, volume 7788 of *Lecture Notes in Computer Science*, pages 101–130. Springer, 2013.

Index